# Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity

Ryan Babbush,[1,*] Craig Gidney,[2] Dominic W. Berry,[3] Nathan Wiebe,[4] Jarrod McClean,[1]
Alexandru Paler,[5] Austin Fowler,[2] and Hartmut Neven[1]

[1]*Google Inc., Venice, California 90291, USA*
[2]*Google Inc., Santa Barbara, California 93117, USA*
[3]*Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia*
[4]*Microsoft Research, Redmond, Washington 98052, USA*
[5]*Institute for Integrated Circuits, Linz Institute of Technology, 4040 Linz, Austria*

We construct quantum circuits that exactly encode the spectra of correlated electron models up to errors from rotation synthesis. By invoking these circuits as oracles within the recently introduced "qubitization" framework, one can use quantum phase estimation to sample states in the Hamiltonian eigenbasis with optimal query complexity $\mathcal{O}(\lambda/\epsilon)$, where $\lambda$ is an absolute sum of Hamiltonian coefficients and $\epsilon$ is the target precision. For both the Hubbard model and electronic structure Hamiltonian in a second quantized basis diagonalizing the Coulomb operator, our circuits have T-gate complexity $\mathcal{O}(N + \log(1/\epsilon))$, where $N$ is the number of orbitals in the basis. This scenario enables sampling in the eigenbasis of electronic structure Hamiltonians with T complexity $\mathcal{O}(N^3/\epsilon + N^2 \log(1/\epsilon)/\epsilon)$. Compared to prior approaches, our algorithms are asymptotically more efficient in gate complexity and require fewer T gates near the classically intractable regime. Compiling to surface code fault-tolerant gates and assuming per-gate error rates of one part in a thousand reveals that one can error correct phase estimation on interesting instances of these problems beyond the current capabilities of classical methods using only about a million superconducting qubits in a matter of hours.

Subject Areas: Chemical Physics,
Quantum Information,
Strongly Correlated Materials

## I. INTRODUCTION

The ubiquitous problem of predicting material properties and chemical reactions from *ab initio* quantum mechanics is among the most anticipated applications of quantum computing. The limitation of most classical algorithms for modeling the physics of superconductivity and molecular electronic structure arises from the seemingly exponential growth of entanglement required to accurately capture strong correlation in systems of interacting electrons. This apparent classical intractability was referenced by Feynman in his seminal work as a key motivation for why we need quantum computers [1,2]. Fourteen years later, Lloyd formalized the concept of a universal quantum simulator [3] and demonstrated an extension for treating systems of interacting electrons in second quantization [4].

Since then, most work developing fermionic quantum simulation methods has focused on time evolution as a means of estimating Hamiltonian spectra and preparing eigenstates [5] via the quantum phase estimation algorithm [6]. Beginning with the proposal of Ref. [7], the idea that one should use phase estimation and adiabatic state preparation [8–10] to extract quantum chemistry ground-state energies became especially popular. More recently, experimental demonstrations [11–15] have focused on the development of variational algorithms [16,17], which are often [18,19], but not always [20,21], inspired by time-evolution primitives.

Performing quantum phase estimation to sample Hamiltonian spectra requires a quantum circuit to implement an operation $\mathcal{W}(H)$, which has eigenvalues that are a known (and efficient-to-compute) function of the eigenvalues of $H$. Most past work has analyzed phase estimation of circuits corresponding to dynamical Hamiltonian simulation, i.e., $\mathcal{W}(H) \approx e^{-iH\tau}$ for some duration $\tau$ [6]. We denote by $f$ the cost of implementing a primitive circuit that is repeated to realize $\mathcal{W}(H)$; e.g., a Trotter step [22] or Taylor series segment [23]. We further define $g(\epsilon)$ as the number of times that one must repeat that primitive to ensure that the error in the spectrum of $H$ encoded in the eigenphases of $\mathcal{W}(H)$ is at most $\mathcal{O}(\epsilon)$. Then, the cost of phase estimation is bounded by

*Corresponding author.
babbush@google.com

$$\mathcal{O}\left(\frac{f \cdot g(\epsilon)}{\epsilon} \|\mathcal{W}'(H)\|^{-1}\right), \qquad (1)$$

where $\|\cdot\|$ denotes the spectral norm and we have taken the derivative of the function of the eigenvalues in the operation $\mathcal{W}'(H)$. In other words, $\mathcal{W}'(H)$ has eigenvalues that are a function of the eigenvalues of $H$, and that function is the derivative of the function that gives the eigenvalues of $\mathcal{W}(H)$. For the case of dynamical time evolution,

$$\mathcal{W}(H) \approx e^{-iH\tau}, \qquad \|\mathcal{W}'(H)\|^{-1} = \|-i\tau e^{-iH\tau}\|^{-1} = \frac{1}{\tau}, \qquad (2)$$

implying that the cost of phase estimation is $\mathcal{O}(f \cdot g(\epsilon)/(\epsilon\tau))$ in this context.

Modern Hamiltonian simulation methods such as the signal processing algorithm [24] and qubitization [25] have achieved the provably optimal scaling that is possible for $g(\epsilon)$ within a query model that aims to synthesize $e^{-iH\tau}$:

$$\mathcal{O}\left(\lambda\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right). \qquad (3)$$

The definition of $\lambda$ depends on the query model; e.g., in models for which the Hamiltonian is given as a weighted sum of unitaries, $\lambda$ is the sum of the absolute values of the weightings [25]. However, $\mathcal{W}(H) \approx e^{-iH\tau}$ is not the only encoding from which one may sample the spectrum of $H$ via phase estimation. Recent papers [26,27] have advocated performing phase estimation on a quantum walk operator corresponding to $\mathcal{W}(H) = e^{i \arccos(H/\lambda)}$, which can be realized exactly as a quantum circuit without approximations beyond those required for rotation synthesis [25]. (This quantum walk operator also produces eigenvalues corresponding to $e^{-i \arccos(H/\lambda)}$, but we ignore those for simplicity of the exposition here.) Even within a black-box query model, one can achieve $g(\epsilon) = \mathcal{O}(1)$ if the goal is to implement $e^{i \arccos(H/\lambda)}$ rather than $e^{-iH/\lambda}$. Performing phase estimation on either circuit would provide the same information since the spectra of these operators are isomorphic. In this case, the cost of phase estimation is $\mathcal{O}(f \cdot \lambda/\epsilon)$, which follows from Eq. (1), $g(\epsilon) = \mathcal{O}(1)$, and

$$\mathcal{W}(H) = e^{i \arccos(H/\lambda)},$$
$$\|\mathcal{W}'(H)\|^{-1} = \left\|\frac{-ie^{i \arccos(H/\lambda)}}{\sqrt{\lambda^2 - H^2}}\right\|^{-1} \leq \lambda. \qquad (4)$$

This work develops methods with such scaling for modeling systems of correlated electrons so that $f = \mathcal{O}(N + \log(1/\epsilon))$.

Our focus on synthesizing unitaries for phase estimation, rather than time-evolution operators that could be used as variational primitives, will result in quantum circuits with millions of gates. Hence, we need quantum error correction. We focus on planar nearest-neighbor coupled arrays of qubits, which are being developed experimentally by multiple groups [28–30]. We use the surface code [31–35], as it has the highest gate threshold error rate for this geometry. Within this model of fault-tolerant quantum computation, the physical resources required for error correcting a quantum circuit are mostly determined by (i) the number of logical qubits and (ii) the number of T gates.

The focus on T gates arises because applying a single T gate consumes many logical qubits and takes significantly longer than applying any other operation [36]. Preparing a $|T\rangle$ state to enable a T gate requires hundreds of thousands of physical qubits. If the goal is to minimize the number of qubits required to execute an algorithm, it makes sense to prepare $|T\rangle$ states serially. Typically, it also takes over 100 rounds of error detection to prepare a $|T\rangle$ state, leaving plenty of time to perform Clifford gates in parallel with this preparation, meaning the execution time of the complete algorithm can be approximated as the total number of T gates multiplied by the time to prepare each $|T\rangle$ state. Thus, throughout this work, we focus on T complexity as the primary cost model. We note, however, that for all algorithms presented or discussed in this work, the T complexity is within logarithmic factors of the gate complexity.

We focus on the two most-studied models of correlated electrons: the Fermi-Hubbard model and the molecular electronic structure Hamiltonian. The Hubbard Hamiltonian is an approximate model of electrons interacting on a planar lattice which some believe may qualitatively capture the behavior of high-temperature superconductivity in cuprates [37]. The molecular electronic structure Hamiltonian is a realistic model of electrons interacting via the Coulomb potential with real kinetic energy in the presence of an external potential (which usually arises from atomic nuclei), in a finite-sized basis [38]. We focus on simulating the electronic structure Hamiltonian in a basis diagonalizing the Coulomb potential [39–41]. For both the Hubbard model and molecular electronic structure Hamiltonian, we are able to provide circuits that simulate $e^{i \arccos(H/\lambda)}$ with T complexity $\mathcal{O}(N + \log(1/\epsilon))$, where $N$ is the number of orbitals in a second-quantized representation of the system. In Tables I and II, we compare the T complexity of past quantum simulation methods for these problems.

In Theorems 1 and 2, we concisely state the T complexity and ancilla requirements of our approach to phase estimation for both the electronic structure Hamiltonian and Hubbard model Hamiltonian, respectively. Both of these theorems are established throughout the paper, but especially in Eqs. (54), (55), (61), and (62). In addition to bounding the T complexity of our algorithms, we provide explicit circuits for their construction and compile all bottleneck primitives down to surface code fault-tolerant gates (topological braiding diagrams). Therefore, the

TABLE I. Progression of lowest T complexity algorithms for implementing a phase estimation unitary encoding eigenvalues of the electronic structure Hamiltonian in second quantization. Note that $N$ is the number of spin orbitals, and $\epsilon$ is the target precision. Here and throughout the paper, $\mathcal{O}(\cdot)$ indicates an upper bound, $\tilde{\mathcal{O}}(\cdot)$ indicates an upper bound ignoring polylogarithmic factors, and $\mathcal{O}(\sim\cdot)$ indicates empirical scaling extrapolated from numerics. "Oracle T gates" refers to $f$ from Eq. (1), and "PEA queries" refers to the rest of the expression in Eq. (1). The scalings attributed to the work on general methods of Hamiltonian simulation assume that one uses the best oracles for electronic structure available at that point in time; e.g., the scaling attributed to Ref. [25] assumes the use of oracles from Ref. [42], and the scaling attributed to Ref. [26] assumes the use of oracles from Ref. [39]. While absent from this table since it did not asymptotically reduce T complexity, Ref. [43] was the first work to explicitly compile a quantum chemistry simulation to Clifford + T gates.

| Year | Reference | Basis | Algorithm | Oracle T gates | PEA queries | Total T gates |
|---|---|---|---|---|---|---|
| 2005 | Aspuru-Guzik et al. [7] | Gaussians | Trotterization | $\mathcal{O}(\text{poly}(N/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ |
| 2010 | Whitfield et al. [44] | Gaussians | Trotterization | $\mathcal{O}(N^4 \log(1/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ |
| 2013 | Wecker et al. [45] | Gaussians | Trotterization | $\mathcal{O}(N^4 \log(1/\epsilon))$ | $\mathcal{O}(N^6/\epsilon^{3/2})$ | $\mathcal{O}([N^{10}\log(1/\epsilon)]/\epsilon^{3/2})$ |
| 2014 | McClean et al. [46] | Gaussians | Trotterization | $\mathcal{O}(\sim N^2 \log(1/\epsilon))$ | $\mathcal{O}(N^6/\epsilon^{3/2})$ | $\mathcal{O}([\sim N^8 \log(1/\epsilon)]/\epsilon^{3/2})$ |
| 2014 | Poulin et al. [47] | Gaussians | Trotterization | $\mathcal{O}(N^4 \log(1/\epsilon))$ | $\mathcal{O}(\sim N^2/\epsilon^{3/2})$ | $\mathcal{O}([\sim N^6 \log(1/\epsilon)]/\epsilon^{3/2})$ |
| 2014 | Babbush et al. [48] | Gaussians | Trotterization | $\mathcal{O}(N^4 \log(1/\epsilon))$ | $\mathcal{O}(\sim N/\epsilon^{3/2})$ | $\mathcal{O}([\sim N^5 \log(1/\epsilon)]/\epsilon^{3/2})$ |
| 2015 | Babbush et al. [42] | Gaussians | Taylorization | $\tilde{\mathcal{O}}(N)$ | $\mathcal{O}([N^4 \log(N/\epsilon)]/[\epsilon \log\log(N/\epsilon)])$ | $\tilde{\mathcal{O}}(N^5/\epsilon)$ |
| 2016 | Low et al. [25] | Gaussians | Qubitization | $\tilde{\mathcal{O}}(N)$ | $\mathcal{O}((N^4/\epsilon) + [\log(N/\epsilon)]/[\epsilon \log\log(N/\epsilon)])$ | $\tilde{\mathcal{O}}(N^5/\epsilon)$ |
| 2017 | Babbush et al. [39] | Plane waves | Taylorization | $\tilde{\mathcal{O}}(N)$ | $\mathcal{O}([N^{8/3} \log(N/\epsilon)]/[\epsilon \log\log(N/\epsilon)])$ | $\tilde{\mathcal{O}}(N^{11/3}/\epsilon)$ |
| 2017 | Berry et al. [26] | Plane waves | Qubitization | $\tilde{\mathcal{O}}(N)$ | $\mathcal{O}(N^{8/3}/\epsilon)$ | $\tilde{\mathcal{O}}(N^{11/3}/\epsilon)$ |
| 2018 | Kivlichan et al. [40] | Plane waves | Trotterization | $\mathcal{O}(N^2 + N \log N \log(1/\epsilon))$ | $\mathcal{O}(\sim N^{3/2}/\epsilon^{3/2})$ | $\mathcal{O}(\sim N^{7/2}/\epsilon^{3/2})$ |
| 2018 | This paper | Plane waves | Qubitization | $\mathcal{O}(N + \log(1/\epsilon))$ | $\mathcal{O}(N^2/\epsilon)$ | $\mathcal{O}([N^3 + N^2 \log(1/\epsilon)]/\epsilon)$ |

TABLE II. Progression of lowest T complexity algorithms for implementing a unitary encoding eigenvalues of the Hubbard model for phase estimation. Here, $N$ is the number of sites, and $\epsilon$ is the target precision. As before, $\mathcal{O}(\cdot)$ indicates an upper bound, $\tilde{\mathcal{O}}(\cdot)$ indicates an upper bound ignoring polylogarithmic factors, and $\mathcal{O}(\sim\cdot)$ indicates empirical scaling extrapolated from numerics. "Oracle T gates" refers to $f$ from Eq. (1), and "PEA queries" refers to the rest of the expression in Eq. (1). The scalings attributed to the work on general methods of Hamiltonian simulation assume that one uses the best oracles for the Hubbard model available at that point in time. For instance, the scaling attributed to Refs. [25] and [26] assumes that one uses the SELECT oracles from Ref. [42], which also work for the Hubbard model. We assume that the work of Ref. [49] would use oracles from Ref. [27]. The scalings attributed to this work assume that our techniques are combined with those from Ref. [49], even though we do not focus on that strategy in our fault-tolerant analysis since those methods seem less effective for finite problem sizes near the classically intractable regime. Nonetheless, we discuss how our methods can be combined to provide the stated complexity in Sec. V D.

| Year | Reference | Algorithm | Ancillae | Oracle T gates | PEA queries | Total T gates |
|---|---|---|---|---|---|---|
| 1997 | Abrams et al. [4] | Trotterization | $\mathcal{O}(1)$ | $\mathcal{O}(N \log(1/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ | $\mathcal{O}(\text{poly}(N/\epsilon))$ |
| 2015 | Wecker et al. [50] | Trotterization | $\mathcal{O}(1)$ | $\mathcal{O}(N \log(1/\epsilon))$ | $\mathcal{O}(N^2/\epsilon^{3/2})$ | $\mathcal{O}([N^3 \log(1/\epsilon)]/\epsilon^{3/2})$ |
| 2015 | Babbush et al. [42] | Taylorization | $\mathcal{O}([\log(N/\epsilon)]/[\log\log(N/\epsilon)])$ | $\mathcal{O}(N \log(1/\epsilon))$ | $\mathcal{O}([N \log(N/\epsilon)]/[\epsilon \log\log(N/\epsilon)])$ | $\mathcal{O}([N^2 \log(N/\epsilon)] \log(1/\epsilon)/[\epsilon \log\log(N/\epsilon)])$ |
| 2016 | Low et al. [25] | Qubitization | $\mathcal{O}(\log N)$ | $\mathcal{O}(N \log(N/\epsilon))$ | $\mathcal{O}((N^2/\epsilon)/[\epsilon \log\log(N/\epsilon)])$ | $\mathcal{O}([N^2 \log(N/\epsilon)]/[\epsilon \log\log(N/\epsilon)])$ |
| 2017 | Berry et al. [26] | Qubitization | $\mathcal{O}(\log N)$ | $\mathcal{O}(N \log(N/\epsilon))$ | $\mathcal{O}(N/\epsilon)$ | $\mathcal{O}([N^2 \log(N/\epsilon)]/\epsilon)$ |
| 2017 | Poulin et al. [27] | Qubitization | $\mathcal{O}(N)$ | $\mathcal{O}(N + \log(1/\epsilon))$ | $\mathcal{O}(N/\epsilon)$ | $\mathcal{O}([N^2 \log(N/\epsilon)]/\epsilon)$ |
| 2018 | Haah et al. [49] | Qubitization | $\mathcal{O}(\log N)$ | $\mathcal{O}(N \log(N/\epsilon))$ | $\tilde{\mathcal{O}}(1/\epsilon)$ | $\tilde{\mathcal{O}}(N/\epsilon)$ |
| 2018 | Kivlichan et al. [40] | Trotterization | $\mathcal{O}(\log N)$ | $\mathcal{O}(N + \log N \log(1/\epsilon))$ | $\mathcal{O}(\sim 1/\epsilon^{3/2})$ | $\mathcal{O}(\sim N/\epsilon^{3/2})$ |
| 2018 | This paper | Qubitization | $\mathcal{O}(\log N)$ | $\mathcal{O}(N + \log(1/\epsilon))$ | $\tilde{\mathcal{O}}(1/\epsilon)$ | $\tilde{\mathcal{O}}(N/\epsilon)$ |

fault-tolerant aspect of our analysis goes further than prior estimates in the simulation literature [51], the most rigorous of which stopped at estimates of T complexity for Trotter-based electronic structure phase estimation [43] and for a variety of techniques used to effect time-evolution of the one-dimensional Heisenberg model [52]. We show that one can perform fault-tolerant phase estimation on interesting instances of both Fermi-Hubbard and molecular electronic structure beyond the capabilities of known classical algorithms using roughly $1 \times 10^6$ physical qubits in the surface code, assuming an architecture with two-qubit error rates of about one part in a thousand.

**Theorem 1.** Consider the electronic structure Hamiltonian in a basis of $N$ spin orbitals, which diagonalizes the Coulomb operator, $H = \sum_{p,q} T(p-q) a_p^\dagger a_q + \sum_p U(p) n_p + \sum_{p \neq q} V(p-q) n_p n_q$, where $\{a_p^\dagger, a_q\} = \delta_{pq}$ are fermionic raising and lowering operators. Furthermore, define $\lambda = \sum_{pq} |T(p-q)| + \sum_p |U(p)| + \sum_{p \neq q} |V(p-q)|$. Then, one can perform phase estimation to sample in the eigenbasis of $H$ with an additive error of at most $\epsilon$ in the eigenvalue using circuits with a number of T gates scaling as $24\sqrt{2}\pi N\lambda/\epsilon + \mathcal{O}((\lambda/\epsilon)\log(N/\epsilon))$ and a number of ancilla qubits scaling as $\log(\lambda^3 N^5/\epsilon^3) + \mathcal{O}(1)$.

**Theorem 2.** Consider the square planar Hubbard model with periodic boundary conditions in a basis of $N$ spin orbitals, $H = -t \sum_{\langle p,q \rangle, \sigma} a_{p,\sigma}^\dagger a_{q,\sigma} + (u/2) \sum_{p, \alpha \neq \beta} n_{p,\alpha} n_{p,\beta}$, where $\{a_{p,\alpha}^\dagger, a_{q,\beta}\} = \delta_{pq}\delta_{\alpha\beta}$ are fermionic raising and lowering operators and the $\langle p, q \rangle$ notation implies a summation over nearest-neighboring orbitals on the periodic planar lattice. Furthermore, define $\lambda = 2Nt + Nu/2$. Then, one can perform phase estimation to sample in the eigenbasis of $H$ with an additive error of at most $\epsilon$ in the eigenvalue using circuits with a number of T gates scaling as $10\sqrt{2}\pi N\lambda/\epsilon + \mathcal{O}(\lambda \log(N/\epsilon)/\epsilon)$ and a number of ancilla qubits scaling as $\log(\lambda N^3/\epsilon) + \mathcal{O}(1)$.

In Sec. II, we give an overview of the simulation strategy that we use to encode and sample eigenspectra via phase estimation. Section II A discusses how one can synthesize $e^{i \arccos(H/\lambda)}$ within the linear combinations of unitaries query model requiring two oracle circuits: SELECT and PREPARE. Section II B introduces a particularly precise variant of phase estimation which queries SELECT and PREPARE oracles to estimate spectra with a precision exceeding the typical Holevo variance. Section II C analyzes the various sources of errors that we need to consider in this algorithm and then bounds the number of times we must query SELECT and PREPARE in order to perform phase estimation.

Sections III–V focus on explicit constructions of SELECT and PREPARE. Section III introduces important primitives for both SELECT and PREPARE. In Sec. III A, we describe circuits applying controlled unitaries such as the mapping $|\ell\rangle|\psi\rangle \mapsto |\ell\rangle X_\ell|\psi\rangle$ with T complexity $\mathcal{O}(L)$, where $L$ is

the number of possible values of $\ell$. In Sec. III B, we show how to selectively apply a Majorana fermion operator, a primitive necessary for our implementation of SELECT in later sections. In Sec. III C, we use the result of Sec. III A to show a particularly efficient variety of quantum read-only memory (QROM), which we use for our PREPARE circuit. In Sec. III D, we describe a general technique for implementing PREPARE in a fashion that keeps $\lambda$ as small as possible.

Sections IV A and IV B discuss explicit constructions of SELECT and PREPARE circuits for the electronic structure Hamiltonian. Sections V A and V B discuss explicit constructions of SELECT and PREPARE circuits for the Hubbard model Hamiltonian. Sections IV C and V C focus on quantifying the number of T gates and ancillae required by the algorithms described in Sec. IV and V. These sections include investigations of the finite-size magnitude of the $\lambda$ and target precisions required to implement our algorithms for interesting problems. In Section V D, we discuss how our Hubbard model simulation techniques can be combined with recent results to achieve even lower scaling based on the locality of the Hubbard Hamiltonian.

Finally, Sec. VI discusses the compilation of these routines to surface code fault-tolerant gates and estimates the physical resources required for error-correcting these algorithms under realistic assumptions about hardware. We conclude in Sec. VII with an outlook on future directions for quantum simulating correlated electron models.

## II. PHASE ESTIMATING SPECTRA OF HERMITIAN LINEAR COMBINATIONS OF UNITARIES

The primary contribution of this paper is to demonstrate a particularly efficient method of using quantum computation to sample the spectra of correlated electron Hamiltonians. Though details of our implementation are specialized to electronic systems, our high-level simulation strategy represents a general framework for spectral estimation. While aspects of this approach were introduced recently in Refs. [26,27], the techniques involved emerged from a history of advances in Hamiltonian simulation prominently involving Szegedy quantum walks [53], the "linear combination of unitaries" (LCU) query model [54], and the method of Hamiltonian simulation known as "qubitization" [25].

Oracular methods of Hamiltonian simulation assume that information about a Hamiltonian is provided by querying "oracle" circuits [55]. These techniques aim to reduce the number of times one must query these oracles in order to effect the intended simulation to target accuracy. The techniques in this paper implement oracles from the LCU query model introduced in Ref. [54]. As the name suggests, this approach begins from the observation that any Hamiltonian can be decomposed as a linear combination of unitaries,

$$H = \sum_{\ell=0}^{L-1} w_\ell H_\ell, \qquad \text{s.t.} \quad (w_\ell \in \mathbb{R}) \wedge (w_\ell \geq 0),$$

$$H_\ell^2 = \mathbb{1}, \tag{5}$$

where $w_\ell$ are scalars and $H_\ell$ are self-inverse operators that act on qubits; e.g., $H_\ell$ could be strings of Pauli operators. The convention in this paper is that the $w_\ell$ are real and nonnegative, with any phases included in the $H_\ell$.

LCU simulation techniques are formulated in terms of queries to two oracle circuits. The first oracle circuit, the "preparation oracle," acts on an empty ancilla register of $\mathcal{O}(\log L)$ qubits and prepares a particular superposition state related to the notation of Eq. (5),

$$\text{PREPARE} \equiv \sum_{\ell=0}^{L-1} \sqrt{\frac{w_\ell}{\lambda}} |\ell\rangle\langle 0|,$$

$$\text{PREPARE} |0\rangle^{\otimes \log L} \mapsto \sum_{\ell=0}^{L-1} \sqrt{\frac{w_\ell}{\lambda}} |\ell\rangle \equiv |\mathcal{L}\rangle,$$

$$\lambda \equiv \sum_{\ell=0}^{L-1} w_\ell. \tag{6}$$

The quantity $\lambda$ is the same as that in Eq. (3) and turns out to have significant ramifications for the overall algorithm complexity. The second oracle circuit we require acts on the ancilla register $|\ell\rangle$ as well as the system register $|\psi\rangle$ and directly applies one of the $H_\ell$ to the system, controlled on the ancilla register. For this reason, we refer to the ancilla register $|\ell\rangle$ as the "selection register" and name the second oracle the "Hamiltonian selection oracle,"

$$\text{SELECT} \equiv \sum_{\ell=0}^{L} |\ell\rangle\langle \ell| \otimes H_\ell,$$

$$\text{SELECT} |\ell\rangle |\psi\rangle \mapsto |\ell\rangle H_\ell |\psi\rangle. \tag{7}$$

Note that the self-inverse nature of the $H_\ell$ operators implies that they are both Hermitian and unitary, which means they can be applied directly to a quantum state.

## A. Encoding spectra in Szegedy quantum walks using qubitization oracles

The essential simulation primitive deployed here (a quantum walk operator based on SELECT and PREPARE) was first introduced as a subroutine to the qubitization approach for Hamiltonian time evolution [25]. However, the direct use of this primitive for phase estimation was first suggested more recently in Ref. [26]. In Sec. II B and the Appendix, we go beyond existing work and prove that, as long as the eigenphase of the walk operator is bounded away from zero (so that the Hamiltonian is not frustration-free), then this algorithm can, in principle, learn as quickly

as traditional phase estimation using the Cramér-Rao bound.

We begin our discussion with the observation that the state $|\mathcal{L}\rangle$ from Eq. (6) encodes $H$ as a projection of SELECT onto $|\mathcal{L}\rangle$,

$$(\langle \mathcal{L}| \otimes \mathbb{1})\text{SELECT}(|\mathcal{L}\rangle \otimes \mathbb{1}) = \frac{1}{\lambda} \sum_\ell w_\ell H_\ell = \frac{H}{\lambda}. \tag{8}$$

This encoding is a general condition for qubitization [25], but the LCU oracles SELECT and PREPARE, as defined in Eqs. (7) and (6), are not necessarily the only constructions that meet this criterion; we refer to the broader family of circuits satisfying Eq. (8) as "qubitization oracles." With this in mind, we discuss a walk operator $\mathcal{W}$ that encodes the spectrum of $H$ as a function of the eigenphases of $\mathcal{W}$, although the spectrum of $\mathcal{W}$ differs from that of the propagator $e^{-iHt}$. One such walk operator $\mathcal{W}$ may be constructed as

$$\mathcal{W} \equiv \mathcal{R}_L \cdot \text{SELECT}, \qquad \mathcal{R}_L \equiv (2|\mathcal{L}\rangle\langle \mathcal{L}| \otimes \mathbb{1} - \mathbb{1}). \tag{9}$$

This construction takes the form of a Szegedy walk [53] since it is composed of a product of two reflection operations. The operation $\mathcal{R}_L$ is manifestly a reflection operation, and it can be seen that SELECT is a reflection operation because

$$\text{SELECT}^2 = \left( \sum_\ell |\ell\rangle\langle \ell| \otimes H_\ell \right)^2$$

$$= \sum_\ell |\ell\rangle\langle \ell| \otimes H_\ell^2 = \mathbb{1}. \tag{10}$$

Fig. 1 shows a circuit that implements $\mathcal{W}$ controlled on an ancilla.

The action of $\mathcal{W}$ partitions Hilbert space into a direct sum of two-dimensional irreducible vector spaces. Through reasoning about these eigenspaces, we can deduce the spectrum of $\mathcal{W}$ as well as the eigenvectors. In particular, we claim that the state $|\mathcal{L}\rangle|k\rangle$ and an orthogonal state $|\phi_k\rangle$ span the irreducible two-dimensional space that $|\mathcal{L}\rangle|k\rangle$ is in under the action of $\mathcal{W}$ for arbitrary eigenstate $|k\rangle$ of $H$ with eigenvalue $E_k$. This state $|\phi_k\rangle$ is formally defined to be the component of $\mathcal{W}|\mathcal{L}\rangle|k\rangle$ that is orthogonal to $|\mathcal{L}\rangle|k\rangle$, which can be simplified, using Eq. (8), to

$$|\phi_k\rangle \equiv \frac{(\mathbb{1} - |\mathcal{L}\rangle\langle \mathcal{L}| \otimes |k\rangle\langle k|) \cdot \text{SELECT}|\mathcal{L}\rangle|k\rangle}{\|(\mathbb{1} - |\mathcal{L}\rangle\langle \mathcal{L}| \otimes |k\rangle\langle k|) \cdot \text{SELECT}|\mathcal{L}\rangle|k\rangle\|}$$

$$= \frac{(\text{SELECT} - \frac{E_k}{\lambda}\mathbb{1})|\mathcal{L}\rangle|k\rangle}{\sqrt{1 - (\frac{E_k}{\lambda})^2}}. \tag{11}$$

The matrix elements of $\mathcal{W}$ can be computed for this state. The upper diagonal matrix element follows from Eq. (8),
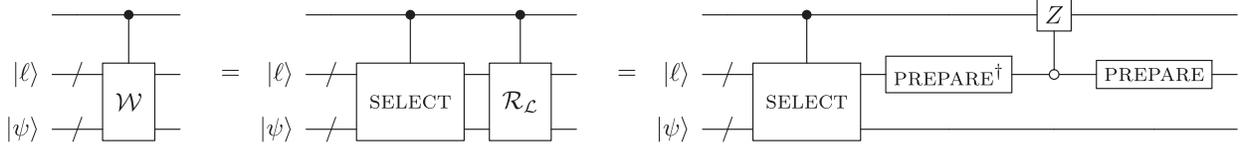
FIG. 1.   A circuit realizing the Szegedy quantum walk operator $\mathcal{W}$ controlled on an ancilla qubit. The last three gates in the circuit on the right constitute the reflection $\mathcal{R}_L$ controlled on an ancilla. Note that the $Z$ gate with the 0-control is actually controlled on the zero state of the entire $|\ell\rangle$ register and not just a single qubit. Accordingly, implementation of that controlled $Z$ has T complexity $\mathcal{O}(\log L)$, where $\log L$ is the size of the $|\ell\rangle$ register. However, that overhead is always negligible compared to the cost of the PREPARE and SELECT operators in the constructions of this paper.

$$\langle k|\langle\mathcal{L}|\mathcal{W}|\mathcal{L}\rangle|k\rangle = \langle k|\langle\mathcal{L}|\text{SELECT}|\mathcal{L}\rangle|k\rangle = \frac{E_k}{\lambda}. \quad (12)$$

The upper transition matrix element between $\langle k|\langle\mathcal{L}|$ and $|\phi_k\rangle$ is given from Eqs. (10) and (11) as

$$\langle k|\langle\mathcal{L}|\mathcal{W}|\phi_k\rangle = \langle k|\langle\mathcal{L}|\mathcal{W}\frac{(\text{SELECT}-\frac{E_k}{\lambda}\mathbb{1})}{\sqrt{1-(\frac{E_k}{\lambda})^2}}|\mathcal{L}\rangle|k\rangle$$

$$= \frac{1-(\frac{E_k}{\lambda})^2}{\sqrt{1-(\frac{E_k}{\lambda})^2}} = \sqrt{1-\left(\frac{E_k}{\lambda}\right)^2}. \quad (13)$$

Note that because phase estimation on $\mathcal{W}$ projects the system to an eigenstate of $\mathcal{W}$ and because $\mathcal{W}$ and $H$ share an eigenbasis, we are only concerned with the action of this operator for eigenstates.

Equations (12) and (13) give the first row of the action of $\mathcal{W}$. The remaining entries can be calculated in a similar way, giving the action of $\mathcal{W}$ on this irreducible two-dimensional subspace as

$$\mathcal{W} \equiv \begin{pmatrix} \frac{E_k}{\lambda} & \sqrt{1-(\frac{E_k}{\lambda})^2} \\ -\sqrt{1-(\frac{E_k}{\lambda})^2} & \frac{E_k}{\lambda} \end{pmatrix} = e^{i\arccos(E_k/\lambda)Y}, \quad (14)$$

where $Y$ is the Pauli-$Y$ operator constrained to this two-dimensional space spanned by $|\mathcal{L}\rangle|k\rangle$ and $|\phi_k\rangle$. Finally, we can see that the phases of the eigenvalues of $\mathcal{W}$ in this subspace are $\pm\arccos(E_k/\lambda)$. Whereas the work of Ref. [25] focused on transforming the evolution under $\arccos(H)$ into evolution under $H$, the more recent work of Refs. [26,27] made the simple observation that by performing phase estimation directly on $\mathcal{W}$, one can obtain the spectrum of $H$ as

$$\text{spectrum}(H) = \lambda\cos\big(\arg[\text{spectrum}(\mathcal{W})]\big), \quad (15)$$

where arg is the argument function $\arg(e^{i\phi}) = \phi$.

## B. Heisenberg-limited phase estimation of the qubitized quantum walk

Since the original work of Ref. [6], many approaches have been proposed for estimating eigenphases of a unitary operator. Whereas, in the past, iterative phase estimation approaches have been more popular in quantum simulation, here we propose using an entanglement-based approach. This approach has the virtue of requiring a number of applications of the unitary that saturates the Heisenberg limit. The ultimate precision that can be reached when one applies phase estimation by controlling a unitary when an ancilla is in $|1\rangle$ and applying the identity gate when the ancilla is in $|0\rangle$ is a Holevo variance of $\tan^2(\pi/(2^{m+1}+1))$, where the total number of applications of the unitary is $2^{m+1}-1$ and $m$ is the number of control qubits used. The Holevo variance is $\langle\cos(\hat{\phi}-\phi)\rangle^2-1$, where $\phi$ is the phase and $\hat{\phi}$ is the estimate of the phase given by the measurement. It is a convenient measure of variance for phase because it enables simple analytic results and is close to the mean-square error for narrowly peaked distributions. The states for these optimal phase measurements were given in Ref. [56]. To apply them to phase estimation of a unitary, one can take the control qubits to be in this superposition state, rather than in a uniform superposition of computational basis states.

We perform a slight optimization of that approach by applying the inverse unitary instead of the identity for the ancilla in the $|0\rangle$ state. Taking $|\phi\rangle$ to be an eigenstate of the unitary with eigenvalue $e^{i\phi}$, this means that instead of applying $|0\rangle|\phi\rangle \to |0\rangle|\phi\rangle$ and $|1\rangle|\phi\rangle \to e^{i\phi}|1\rangle|\phi\rangle$, we apply $|0\rangle|\phi\rangle \to e^{-i\phi}|0\rangle|\phi\rangle$ and $|1\rangle|\phi\rangle \to e^{i\phi}|1\rangle|\phi\rangle$. This doubles the effective phase difference and turns out to have the same complexity. As shown in Fig. 2, we accomplish the controlled inverse by removing controls from $\mathcal{W}^n$ and inserting controlled reflection operators $\mathcal{R}_L$ into the circuit, which will cause us to apply either $(\mathcal{W}^\dagger)^n$ or $\mathcal{W}^n$ depending on the state of the ancilla. We can see why this works by examining the relation

$$\mathcal{R}_L \cdot \mathcal{W}^n \cdot \mathcal{R}_L = \mathcal{R}_L{}^2 \cdot (\text{SELECT}\cdot\mathcal{R}_L)^n$$

$$= (\text{SELECT}\cdot\mathcal{R}_L)^n = (\mathcal{W}^\dagger)^n, \quad (16)$$
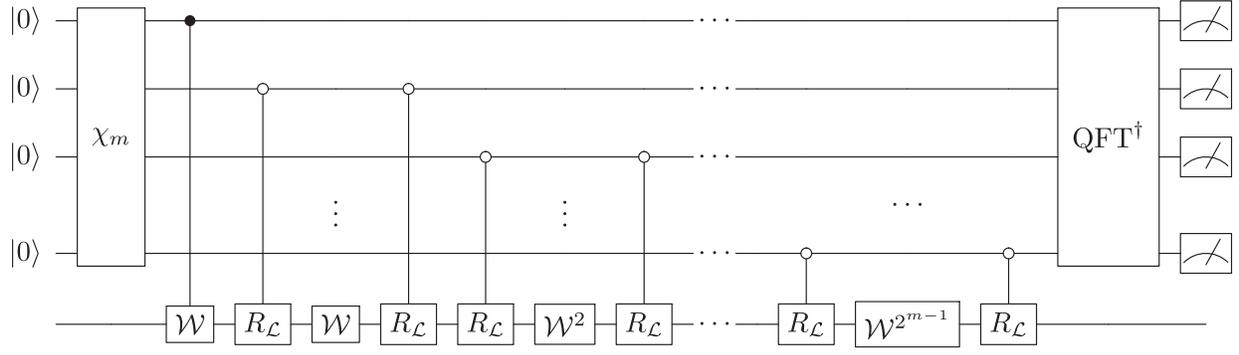
FIG. 2. Heisenberg-limited phase estimation circuit for learning the eigenphase of $\mathcal{W}$ with $m$ bits of accuracy with Holevo variance $\pi^2/2^{2(m+1)}$, where $R_{\mathcal{L}}$ is $(2|\mathcal{L}\rangle\langle\mathcal{L}| \otimes \mathbb{1} - \mathbb{1})$ and $\chi_m$ prepares the resource state from Eq. (17), which was shown to be optimal in Ref. [56]. Both $\chi_m$ and the inverse quantum Fourier transform (QFT$^\dagger$) have gate complexity $\tilde{\mathcal{O}}(m)$, which is completely negligible compared to the overall gate complexity of phase estimation, which scales as $\mathcal{O}(2^m)$. The controlled $R_{\mathcal{L}}$ and $\mathcal{W} = R_{\mathcal{L}} \cdot$ SELECT gates are implemented as shown in Fig. 1. As a consequence of Eq. (16), this circuit involves only $2^m - 1$ applications of $R_{\mathcal{L}}$ and as many applications of SELECT. Note that the first unit of $R_{\mathcal{L}} \cdot W \cdot R_{\mathcal{L}}$ is replaced by $\mathcal{W}$ controlled on the zero state of an ancilla in order to help disambiguate the outcomes of $\arccos(E_k/\lambda)$ and $\arccos(E_k/\lambda) + \pi$.

which holds for any integer $n$ as a consequence of the self-inverse nature of $R_L$ and $\mathcal{W}$. Moreover, because $\mathcal{W}^n$ always ends with an $R_L$ operation, each controlled $R_L$ can be combined in the circuit with the $\mathcal{W}^n$ to yield a complexity no greater than the complexity of just performing the $\mathcal{W}^n$ operations.

This trick will result in measuring the phase modulo $\pi$. To eliminate the $\pi$ ambiguity, an additional controlled $\mathcal{W}$ can be performed without this trick. This case is shown as the first controlled operation in Fig. 2. For $m$ control qubits, the Holevo variance is still $\tan^2(\pi/(2^{m+1} + 1))$, but the complexity is reduced by approximately half to $2^m$ applications of the unitary $\mathcal{W}$.

As seen in Fig. 2, our modified phase estimation algorithm begins with a unitary $\chi_m$, which prepares the state

$$\chi_m|0\rangle^{\otimes m} \mapsto \sqrt{\frac{2}{2^m+1}} \sum_{n=0}^{2^m-1} \sin\left(\frac{\pi(n+1)}{2^m+1}\right)|n\rangle. \quad (17)$$

To prepare this state with cost $\tilde{\mathcal{O}}(m)$, we first perform Hadamards on $m + 1$ qubits (initially in the $|0\rangle$ state) to give

$$\sqrt{\frac{1}{2^{(m+1)}}} \sum_{n=0}^{2^m-1} |n\rangle \otimes (|0\rangle + |1\rangle). \quad (18)$$

Next, we perform a series of $m$ controlled rotations, with each of the first $m$ qubits as control and qubit $m + 1$ as target. For control qubit $k$, the rotation on the target qubit $m + 1$ is $e^{i\pi 2^k Z/(2^m+1)}$. If we perform a further rotation of $e^{i\pi Z/(2^m+1)}$ on qubit $m + 1$, the resulting state is

$$\sqrt{\frac{1}{2^{(m+1)}}} \sum_{n=0}^{2^m-1} (e^{i\pi(n+1)/(2^m+1)}|n\rangle \otimes |0\rangle$$
$$+ e^{-i\pi(n+1)/(2^m+1)}|n\rangle \otimes |1\rangle). \quad (19)$$

We perform a Hadamard on qubit $m + 1$ and measure in the computational basis. Measuring $|1\rangle$ gives the state

$$i\sqrt{\frac{1}{2^m}} \sum_{n=0}^{2^m-1} \sin\left(\frac{\pi(n+1)}{2^m+1}\right)|n\rangle. \quad (20)$$

The probability of success is given by the normalization $(1 + 2^{-m})/2$. The scheme can be made unitary and deterministic via a single step of amplitude amplification. Clearly, this preparation scheme scaling as $\tilde{\mathcal{O}}(m)$ will not dominate the cost of our overall phase estimation, which scales as $\mathcal{O}(2^m)$, as we discuss in the next section.

## C. Error scaling and query complexity

Three sources of error enter our simulation: error due to performing PEA to finite precision, $\epsilon_{\text{PEA}}$; error due to approximate preparation of the Hamiltonian terms within the implementation of the PREPARE oracle, $\epsilon_{\text{PREP}}$; and the error in synthesizing the inverse QFT, $\epsilon_{\text{QFT}}$. We choose to measure error through the root-mean-square error of the estimator used within phase estimation, i.e.,

$$\Delta\phi \equiv \sqrt{\mathbb{E}[\text{dist}(\phi_{\text{est}}, \phi_{\text{true}})^2]}, \quad (21)$$

where the distance considered above is the angular distance between the estimated phase and the actual phase.

Provided phase estimation is performed on a unitary operation, the error in the estimate of the energy is at most

the error in implementing the unitary [43]. We break up the estimated phase as the sum of two contributions, $\phi_{\text{est}} = \phi + \epsilon_{\text{PREP}} + \phi_{\text{true}}$. Here, $\phi$ is a random variable with zero mean $\mathbb{E}(\phi)$ and Holevo variance $\mathbb{V}_H(\phi)$ describing the output of phase estimation, and $\epsilon_{\text{PREP}}$ represents the systematic errors in the phase that arise because of gate synthesis. In the limit of small variance, we can express this with high probability over the true phase as

$$
\begin{aligned}
\Delta\phi &\approx \sqrt{\mathbb{E}[(\phi_{\text{est}} - \phi_{\text{true}})^2]} \\
&\approx \sqrt{\mathbb{V}_H(\phi) + (\epsilon_{\text{PREP}} + \pi\epsilon_{\text{QFT}})^2} \\
&\approx \sqrt{\left(\frac{\pi}{2^{m+1}}\right)^2 + (\epsilon_{\text{PREP}} + \pi\epsilon_{\text{QFT}})^2},
\end{aligned} \tag{22}
$$

where $m$ ancilla qubits are used within the phase estimation algorithm. Note that such a division of the error is suboptimal since the cost involved in reducing the error for phase estimation is exponentially larger than that involved in increasing the accuracy of the circuit synthesis [43]; however, we take the two errors to be equal for simplicity.

Eq. (15) implies that error in the energy is at most

$$
\Delta E = \lambda\Delta\cos(\phi) \le \lambda\Delta\phi \approx \lambda\sqrt{\left(\frac{\pi}{2^{m+1}}\right)^2 + (\epsilon_{\text{PREP}} + \pi\epsilon_{\text{QFT}})^2}. \tag{23}
$$

This result suggests that we can choose to estimate the phase to a number of bits given by

$$
m = \left\lceil \log\left(\frac{\sqrt{2}\pi\lambda}{2\Delta E}\right) \right\rceil < \log\left(\frac{\sqrt{2}\pi\lambda}{\Delta E}\right), \tag{24}
$$

and the target errors can be chosen as

$$
\epsilon_{\text{PREP}} \le \frac{\sqrt{2}\Delta E}{4\lambda}, \qquad \epsilon_{\text{QFT}} \le \frac{\sqrt{2}\Delta E}{4\pi\lambda}. \tag{25}
$$

Thus, using the phase estimation procedure from Sec. II B, we need at most

$$
2^m < \frac{\sqrt{2}\pi\lambda}{\Delta E} \tag{26}
$$

queries to the SELECT oracle and at most twice as many queries to the PREPARE oracle in order to estimate spectra to within error $\Delta E$. Supposing that the circuit PREPARE can be applied at gate complexity $P$ and the circuit SELECT can be applied at gate complexity $S$, the gate complexity of our simulation [ignoring, for now, the cost of $\chi_m$ and the cost of the QFT$^\dagger$ since they scale as $\mathcal{O}(m)$] is then approximately bounded from above by

$$
\frac{\sqrt{2}\pi\lambda(S + 2P)}{\Delta E}. \tag{27}
$$

This paper discusses implementations of SELECT and PREPARE that minimize $S$ and $P$ without increasing $\lambda$.

To implement the inverse QFT that appears in Fig. 2, we use the semiclassical algorithm described in Ref. [57]. This version of the QFT requires just $m - 1$ rotation gates and $m$ Hadamards when implemented on $m$ qubits. Thus, the error in each rotation must be at most $\epsilon_{\text{QFT}}/(\pi m)$, which implies that the inverse QFT will have T complexity scaling as $\mathcal{O}(m \log(m/\epsilon_{\text{QFT}}))$. As this is an additive cost to other parts of our phase estimation algorithm with T complexity scaling as $\mathcal{O}(2^m)$, the cost of performing the QFT within the required error budget can be safely neglected.

How errors in the coefficients of the implemented Hamiltonian propagate into $\epsilon_{\text{PREP}}$ is slightly harder to bound owing to the fact that the error in the eigenphase is a nonlinear function of the error in the Hamiltonian implementation. In particular, the error can diverge for frustration-free Hamiltonians owing to the singularity of arccos. The main result, shown in the Appendix, is that PREPARE should be implemented so that if $\tilde{w}_\ell$ is the effective coefficient of $H_\ell$ in the approximately implemented Hamiltonian, then

$$
|\tilde{w}_\ell - w_\ell| \le \delta = \frac{\sqrt{2}\Delta E}{4L(1 + \frac{\Delta E^2}{8\lambda^2})}\left(1 - \frac{\|H\|^2}{\lambda^2}\right). \tag{28}
$$

## III. LOW T COMPLEXITY PRIMITIVES FOR LCU ORACLES

In this section, we introduce three circuit primitives that are helpful for implementing SELECT and PREPARE oracles with low T-gate complexity. We use these primitives for electronic structure simulation but expect them to be useful more generally. These primitives enable black-box implementations of SELECT and PREPARE for any problem with lower asymptotic complexity than prior constructions in the literature. They also have low T counts at finite size. We use these constructions extensively in Secs. IV and V of this paper.

In Sec. III A, we introduce a technique for "streaming" bits of an iterator running over a unary register. One application is that this technique can be used to coherently apply operations controlled on a register with $\log L$ qubits in superposition [e.g., the selection register in Eq. (6) and Eq. (7)] using a number of T gates scaling as $\mathcal{O}(L)$, as opposed to $\mathcal{O}(L \log L)$ as one might normally expect. However, what is even more important is the versatile way that these constructions can be applied.

In Sec. III B, we show how one can use the results of Sec. III A to implement a primitive corresponding to controlled application of a Majorana fermion operator.

This primitive is used directly in our implementation of SELECT in Secs. IV A and V A.

In Sec. III C, we show a straightforward application of the techniques in Sec. III A that allow us to develop a particularly efficient quantum data lookup, which we refer to as "quantum read-only memory" (QROM). In particular, for coherently querying a database with $L$ words, our implementation of QROM has T complexity of $4L - 4$ with no dependence on the word length, which is an asymptotic and constant-factor improvement over all prior literature. We will discuss QROM in more detail in a forthcoming work [58].

In Sec. III D, we discuss a technique for initializing a state with $L$ unique coefficients (provided by a classical database), with the number of T gates scaling as $4L + \mathcal{O}(\log(1/\epsilon))$, where $\epsilon$ is the largest absolute error that one can tolerate in the prepared amplitudes. This routine improves asymptotically over the gate complexity of prior constructions for a black box PREPARE. It also has the advantage of implementing PREPARE without increasing the value of $\lambda$ [from Eq. (6)], which has been a frequent problem with other implementations of PREPARE [39,42,59] that attempted to obtain scaling sublinear in the number of terms in the linear combinations of unitaries decomposition.

### A. Unary iteration and indexed operations

Many of the circuits in this paper rely heavily on a technique we refer to as "unary iteration." The unary iteration process gradually produces, and then uncomputes, qubits that indicate whether an index register is storing specific values (with respect to the computational basis). We call the process unary iteration because the indicator qubits are made available one by one (iteration), and they correspond to the one-hot (unary) encoding of the index register value. While these techniques were developed independently, we note that a scheme similar to unary iteration is also used for implementing SELECT operations in Ref [52]. Compared to Ref. [52], we lower the T count from $6L - 4$ to $4L - 4$ and explain how to apply the scheme to sizes $L$ that are not powers of 2.

For an index register storing an index in the interval $[0, L)$, the space overhead of converting the index register into a unary register (as in Ref. [27]) would normally be $L$ qubits. By comparison, our unary iteration technique is exponentially more efficient in space without any increased T complexity, requiring only $\log L$ ancillae. Our unary iteration has a T-count of $4L - 4$ and can be parallelized if needed without increasing the T count. Despite its efficiency, unary iteration is still the dominant source of T complexity in our algorithms. We use it for indexed operations, Majorana operators, reversible preparation of states, and database lookup, all of which have T counts that scale like $\mathcal{O}(L)$.

To explain unary iteration, we first focus on how it is used to implement a controlled indexed NOT operation:

$$|c\rangle|\ell\rangle|\psi\rangle \mapsto |c\rangle|\ell\rangle(X_\ell)^c|\psi\rangle, \qquad (29)$$

where $|c\rangle$ is the control register, $|\ell\rangle$ is the selection register, $|\psi\rangle$ is the system register, and the subscript $\ell$ on $X_\ell$ indicates that the NOT operation acts on qubit $\ell$ of the system register. From Eq. (7), it should not be surprising that this primitive is helpful for our constructions of SELECT.

A simple (but suboptimal) way to implement Eq. (29) would be to totally control the application of $X_\ell$ on all possible values that could occur in the register $|\ell\rangle$, as shown in Fig. 3. For instance, in order to apply $X_{158}$ when $|\ell\rangle = |158\rangle$, the total-control approach would place a NOT gate targeting the qubit 158 in the system register $|\psi\rangle$ but with a control on each index bit. The control's type (ON or OFF) would be determined by the binary representation of 158 ($158_{10} = 10\,011\,110_2$), so there would be a must-be-OFF control on the low bit of the index register (because the low bit of 158 in binary is 0), a must-be-ON control on the next bit (because the next bit of 158 in binary is 1), and so forth. In order to cover every case, a separate NOT gate with corresponding controls would be generated for every integer from 0 up to $L - 1$. This would produce $L$ different NOT operations, each targeting a different qubit in the target register and each having a number of controls equal to the size of the index register (i.e., $\log L$). Thus, it takes $\mathcal{O}(L \log L)$ T gates to apply Eq. (29) using this approach. Unary iteration will improve this T count to $4L - 4$.
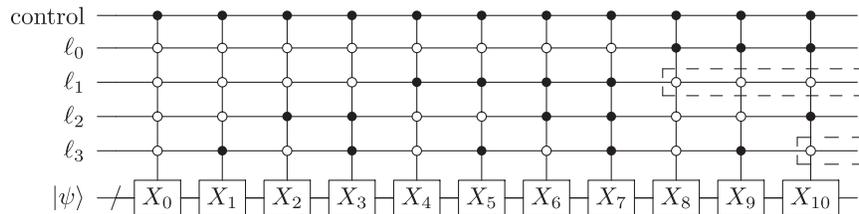


FIG. 3.    Example total control circuit for performing a controlled indexed $X_\ell$ operation, with $0 \le \ell < L = 11$. This is the (naive) starting point for producing a unary iteration circuit, before optimizations that asymptotically improve the T complexity. When indices outside the specified range do not occur, the highlighted runs of OFF-type controls reaching the right-hand side of the circuit can be removed without affecting the circuit. (There are also other controls that could be removed, but for our purposes, this would be counterproductive because it would interfere with later optimizations).
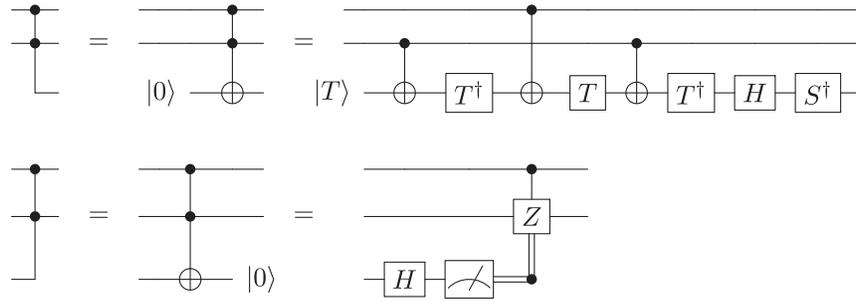
FIG. 4.   Computing and uncomputing AND operations, defined in terms of Toffoli gates and in terms of Clifford + T gates [60]. Computing an AND consumes 4 $|T\rangle$ states and is equivalent to applying a Toffoli gate to a target qubit known to be $|0\rangle$. Uncomputing an AND consumes no $|T\rangle$ states and is equivalent to applying a Toffoli gate to a target qubit guaranteed to end up in the $|0\rangle$ state. Drawing AND operations as "corners" instead of as $\oplus$ symbols is a visual cue that the target qubit will be (or was) OFF after (before) the operation. This is worth highlighting because it affects the T count of synthesizing the operation and means that the target is available for reuse as an ancilla in later operations.

Consider that the controls for the operation targeting the qubit at offset $\ell = 158$ are almost identical to the controls for the operation targeting the qubit at offset $\ell = 159$. They differ only on the low bit of the index register, where 158 requires the bit to be OFF, whereas 159 requires the bit to be ON. If we combine the $\log L - 1$ other qubits of the index register into a single representative qubit that is set if and only if those controls are met, we could use this representative qubit once for the $\ell = 158$ case and again for the $\ell = 159$ case. Using the representative qubit twice, instead of computing it twice, decreases the total amount of work done. Unary iteration is the result of taking this kind of representative-reuse idea to its natural limit.

We define our unary iteration construction by starting with a total-control circuit and then applying a fixed set of simple transformations and optimizations. Fig. 3 shows an example starting point, a total-control circuit for $L = 11$. For unary iteration, we require that the index register never store an out-of-range value $\ell \geq L$. For example, consider what occurs when the $X_{10}$ operation from Fig. 3 is not conditioned on $\ell_0$ (the least significant bit of the index register). This would cause an $X_{10}$ to be applied to the target when $\ell = 11$, but this is fine since we know $\ell \neq 11$. We use the $\ell < L$ condition to omit several controls from the circuit. For each possible $\ell$, we look at the $X_\ell$ operation and remove the control on the $b$th index bit when the following

two conditions are true: (i) the $b$th bit of $L - 1$ is not set, and (ii) setting the $b$th bit of $\ell$ would change $\ell$ into a value larger than $L - 1$. Visually, this removes "runs" of must-be-OFF controls that manage to reach the right side of the circuit as highlighted in Fig. 3.

After removing the specified controls, we expand the remaining controls into nested AND operations (the AND operation is defined in Fig. 4), always nesting so that lower controls are inside higher controls. For clarity, we consistently place the ancillae associated with an AND operation just below its lowest input qubit. The result is the "sawtooth" circuit shown in Fig. 5. By iteratively optimizing adjacent AND operations as shown in Fig. 6, the sawtooth circuit from Fig. 5 is optimized into the circuit shown in Fig. 7. This is our unary iteration circuit for $L = 11$. The optimized circuit always ends up with $L - 1$ AND computations (even when $L$ is not a power of 2), each AND takes 4 T gates to compute, and we have no other T-consuming operations in the circuit. Thus, the T count of this construction is $4L - 4$.

## B. Selective application of Majorana fermion operators

Now that we have described unary iteration, we can begin using it to construct primitives relevant for the SELECT oracle. As discussed in detail in Secs. IV and V
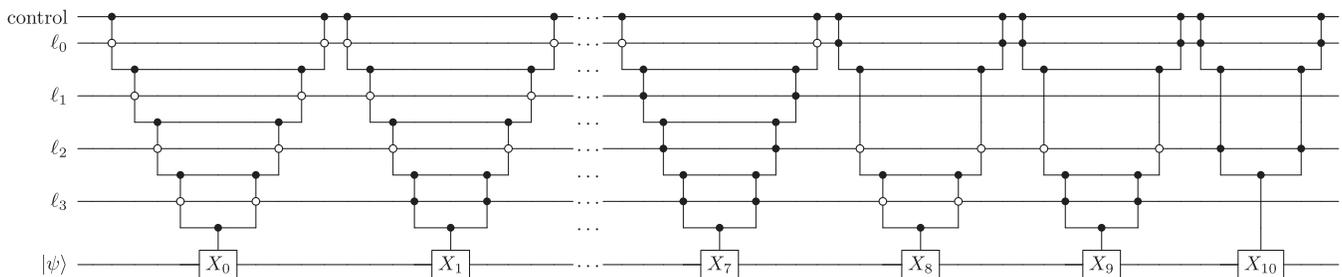


FIG. 5.   The "sawtooth" circuit resulting from removing unnecessary bits from Fig. 3 and then adding AND operations from Fig. 4 to combine the controls for performing a controlled indexed $X_\ell$ operation with $L = 11$ possible targets.
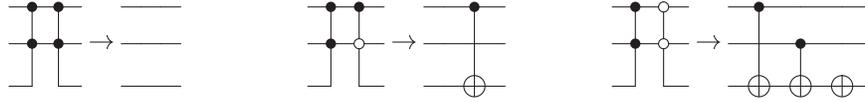
FIG. 6.   When two AND operations are adjacent, the uncomputation-and-recomputation can be replaced by CNOT and NOT operations. Each such merger saves 4 T gates.
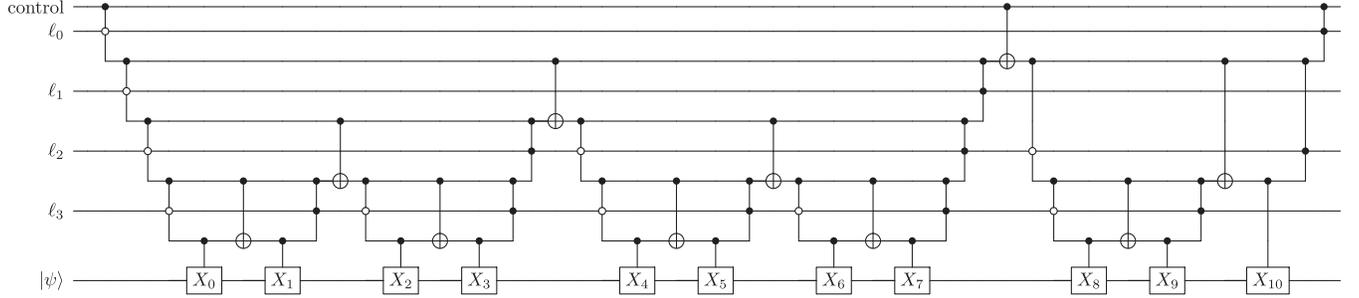


FIG. 7.   An $L = 11$ unary iteration circuit that applies $X_\ell$ to the qubit $\ell$ in the system register $|\psi\rangle$, where $\ell$ is the value stored in the selection register. The circuit is obtained by merging AND operations from Fig. 5 using the method shown in Fig. 6. It computes 10 AND operations and so has a T count of $10 \times 4 = 40 = 4L - 4$.

below, our approach for implementing SELECT will require that we have a circuit capable of selectively applying the Majorana fermion operator

$$|\ell\rangle|\psi\rangle \mapsto |\ell\rangle \left( \frac{a_\ell^\dagger - a_\ell}{i} \right)|\psi\rangle = |\ell\rangle Y_\ell \cdot Z_{\ell-1}...Z_0|\psi\rangle, \qquad (30)$$

where the last equality holds under the Jordan-Wigner transformation [61]. In this section, we describe explicit circuits that accomplish the mapping of Eq. (30).

In Sec. III A, we discuss selectively applying $X_\ell$ operations as a representative example of how one might use unary iteration. However, nothing intrinsic to the unary iteration construction requires that the indexed operation be so simple. For example, we could switch from applying $X_\ell$ to applying $Z_\ell$ halfway through the circuit. Or each $X_\ell$ could be replaced by multiple Pauli operations targeting multiple qubits. In general, each index could be associated with its own unique set of Pauli operators to be applied to various target qubits.

We can also apply transformations to our quantum unary iterators (analogous to transformations of classical iterators). Iterators can be mapped, filtered, zipped, aggregated, batched, flattened, grouped, etc. For instance, given a classical stream of bits, one can aggregate over it with the $\oplus$ operation. This produces a new iterator, which iterates over bits equal to the parity of the values so far from the original iterator. It is possible to apply this XOR-aggregation idea to the quantum unary iteration process. We can introduce an "accumulator" qubit and, as each iterated unary qubit is produced, CNOT it into the accumulator. In effect, if the index register is storing $\ell$, then the accumulator will stay OFF until the $\ell$th qubit toggles it ON.

The accumulator will then stay ON until the end of the iteration process, where it is uncomputed by a CNOT from the control qubit. By conditioning $X_\ell$ on the accumulator qubit, instead of on the original unary qubits, efficient *ranged* operations such as $|\ell\rangle|\psi\rangle \to |\ell\rangle G_\ell \cdot G_{\ell+1}\cdots G_{L-1}|\psi\rangle$ are produced. We show an example of an accumulator-based ranged operation in Fig. 8.

By using both the accumulator qubit and the original unary qubits, we can apply a ranged indexed operation and an indexed operation in a single unary iteration, which gradually sweeps over the possible target qubits. The resulting combined operation, shown in Fig. 9, is a crucial part of our SELECT circuit, effecting the transformation of Eq. (30).

### C. QROM for low T complexity data lookup

In this section, we explain how one can use the techniques of Sec. III A in order to implement a particular efficient form of what we call QROM [62], which is useful in the context of the SUBPREPARE routine, a subroutine of the PREPARE circuit described in Sec. IV B (in Fig. 16). Many quantum algorithms assume the existence of a hypothetical peripheral called "quantum random-access memory" (QRAM) [63], which allows classical or quantum data to be accessed via an index under superposition. The purpose of QROM is to read classical data indexed by a quantum register, i.e., to perform the following transformation:

$$\text{QROM}_d \cdot \sum_{\ell=0}^{L-1} \alpha_\ell |\ell\rangle|0\rangle = \sum_{\ell=0}^{L-1} \alpha_\ell |\ell\rangle|d_\ell\rangle, \qquad (31)$$
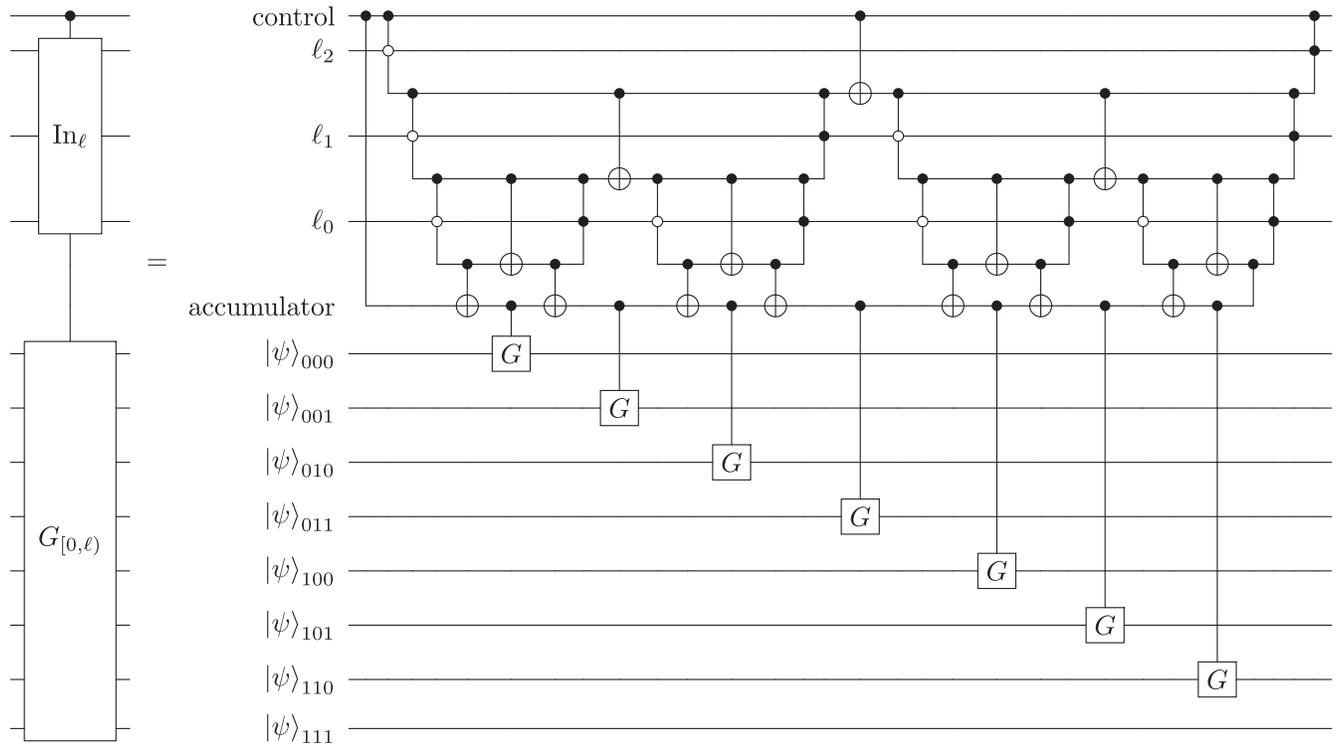
FIG. 8.    Ranged operation construction implementing $|\ell\rangle|\psi\rangle \to |\ell\rangle \prod_{k=0}^{\ell-1} G_k |\psi\rangle$. It applies the $G$ operation to a range of values, instead of to a single value, by using an accumulator. The accumulator is guaranteed to be cleared after the final CNOT targeting it (drawn as a line merging into an ancilla qubit). This occurs because (unless CONTROL is not set and the accumulator simply stays unset) exactly one of the unary bits must have been set, and we targeted the accumulator with CNOTs controlled by each of those bits in turn. Note that $G_{[p,q)}$ refers to $G$ being applied to every qubit index $k$ satisfying $p \le k < q$.
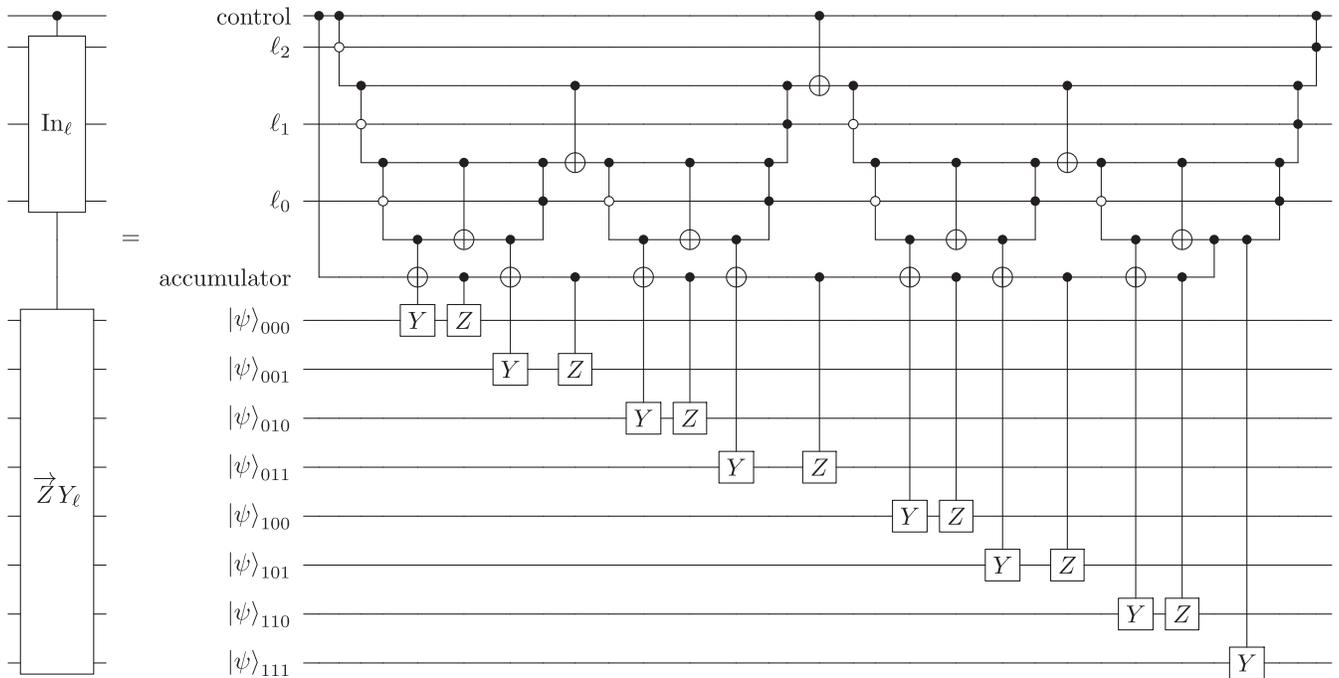
FIG. 9.    Application of a selected Majorana fermion operator, $|\ell\rangle|\psi\rangle \mapsto |\ell\rangle Y_\ell \cdot Z_{\ell-1} \cdots Z_0 |\psi\rangle$ as described in Eq. (30). This application is accomplished by performing a ranged operation (as shown in Fig. 8) and an indexed operation (similar to what is shown in Fig. 7) with a single pass through the selection register $|\ell\rangle$. It has a T count of $4L - 4$, where $L$ is the number of integer values that can be held by the selection register $|\ell\rangle$.
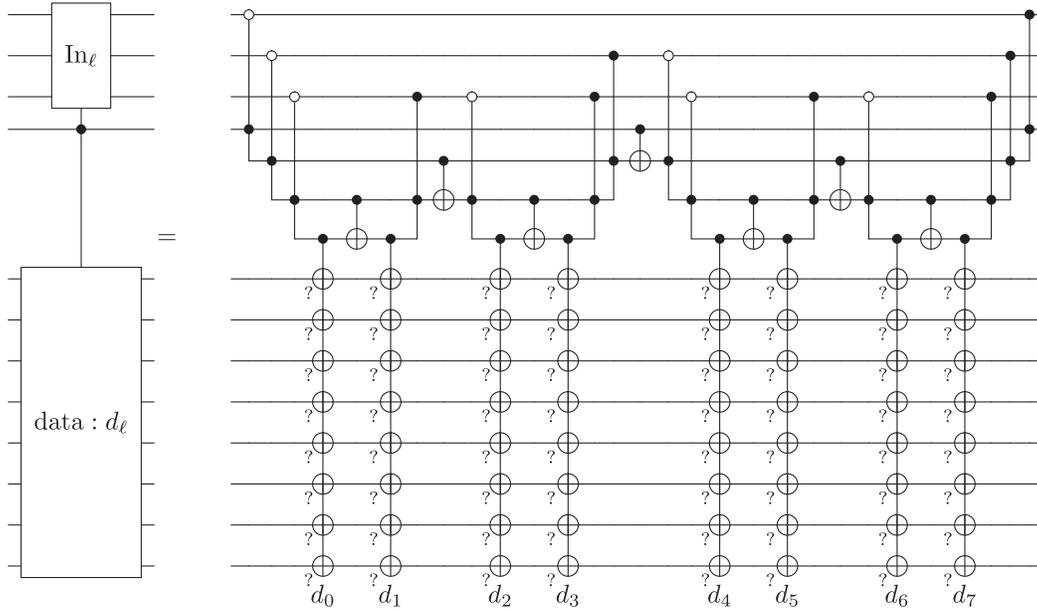
FIG. 10. Finite-sized example of the QROM database loading scheme used in our implementation of SUBPREPARE. If the index register contains $\ell$, the output register ends up storing $d_\ell$, where $\vec{d}$ is some precomputed data vector used when constructing the circuit. The top part of the circuit performs unary iteration, as described in Sec. III A. The bottom part of the circuit loads classical data associated with each possible index. The classical data are encoded into the presence or absence of CNOTs on the data lines. The "?" marks in the diagram indicate that one should decide whether or not to have a CNOT gate on each line, depending on the value of the data to load. This circuit has a T count of $4L - 4$, which is due entirely to the unary iteration. The T-gate cost of this circuit is independent of the number of bits used to store each element of the database.

where $\ell$ is the index to read, $\alpha_\ell$ is the amplitude of $|\ell\rangle$, and $d_\ell$ is the word associated with index $\ell$ in a classical list $d$ containing $L$ words. Our implementation of QROM is shown in Fig. 10. Note that our notion of QROM is unrelated to the discussion of ROM on a quantum computer in Ref. [64].

The read-only aspect of QROM makes it distinctly different from QRAM in that one can read from QROM but cannot write to it during the course of a computation. A few algorithms, such as the procedure introduced in Ref. [65], actually do require that one write to QRAM; thus, for such cases, QROM would not be appropriate. A notable difference between this paper and most previous work on QRAM [63,66–68] is that we describe the cost of QROM in terms of a fault-tolerant cost model: the number of T gates performed and the number of ancilla qubits required. Under such cost models, the "bucket brigade" QRAM design of Giovannetti *et al.* [63,66] has T complexity (and, thus, also time complexity under reasonable error-correction models) of $\mathcal{O}(L)$ regardless of the fact that it has depth $\mathcal{O}(\log L)$ because implementing it as an error-corrected circuit consumes $\mathcal{O}(L)$ T gates and $\mathcal{O}(L)$ ancillae qubits. Our implementation of QROM consumes only $4L$ T gates and $\log L$ ancillae, which is a constant-factor improvement in T count and an exponential improvement in space usage over the construction of Giovannetti *et al.*.

## D. Subsampling the coefficient oracle

In this section, we introduce a technique for initializing a state with $L$ unique coefficients (provided by a classical database) with a number of T gates scaling as $4L + \mathcal{O}(\log(1/\epsilon))$, where $\epsilon$ is the largest absolute error that one can tolerate in the prepared amplitudes. This result constitutes a general procedure for implementing PREPARE such that the cost of circuit synthesis is additive, rather than multiplicative (as in most prior schemes). In particular, it improves on the database scheme from Ref. [42] (based on the procedure of Ref. [69]), which requires a number of T gates scaling as $\mathcal{O}(L \log(L/\epsilon))$. Importantly, our scheme does not increase the value of $L$ or $\lambda$, which would usually be the case for most "on-the-fly" strategies for implementing PREPARE [23,42,59].

Generalizing the requirements of Eq. (6), we begin with the observation that it would be acceptable to have a PREPARE circuit that initializes the state

$$|\mathcal{L}\rangle \equiv \sum_{\ell=0}^{L-1} \sqrt{\frac{w_\ell}{\lambda}} |\ell\rangle |\text{temp}_\ell\rangle, \qquad (32)$$

where $|\text{temp}_\ell\rangle$ is an unspecified junk register entangled with $|\ell\rangle$. Equivalently, any pure state $|\mathcal{L}\rangle$ would suffice if

$$\langle \mathcal{L}|(|\ell\rangle\langle\ell|\otimes\mathbb{1})|\mathcal{L}\rangle = \frac{w_\ell}{\lambda} \quad \forall\, \ell \in [0, L). \quad (33)$$

Because SELECT only uses $|\ell\rangle$ to *control* operations, phase error (including entanglement with the junk register) in the state produced by PREPARE will commute across SELECT and be corrected by PREPARE$^\dagger$. However, SELECT itself necessarily introduces entanglement between its target register $|\psi\rangle$ and the index register $|\ell\rangle$ (plus associated junk register). So PREPARE$^\dagger$ will not exactly restore $|\ell\rangle$ or the junk register. Although we only specify the action of PREPARE on the $|0\rangle$ state, PREPARE will be applied to other states due to this imperfect uncomputation effect. This is accounted for by requiring that (i) qubits coming out of PREPARE$^\dagger$ are kept and fed back into the next PREPARE operation and (ii) the reflection step between PREPARE$^\dagger$ and PREPARE only affects the $|0\rangle$ state.

Given the observation that the existence of an entangled junk register is acceptable, we seek to implement a circuit that effects the transformation,

$$|0\rangle^{\otimes(1+2\mu+2\log L)} \mapsto \sum_{\ell=0}^{L-1} \sqrt{\tilde{\rho}_\ell}\, |\ell\rangle |\text{temp}_\ell\rangle, \quad (34)$$

where $\tilde{\rho}_\ell \equiv \tilde{w}_\ell/\lambda$ are probabilities characterizing the approximate Hamiltonian we are encoding. Whereas the exact Hamiltonian would be associated with probabilities $\rho_\ell \equiv w_\ell/\lambda$, the value $\tilde{\rho}_\ell$ is a $\mu$-bit binary approximation to $\rho_\ell$ such that

$$|\rho_\ell - \tilde{\rho}_\ell| = \frac{|w_\ell - \tilde{w}_\ell|}{\lambda} \leq \frac{1}{2^\mu L} \leq \frac{\delta}{\lambda}$$

$$= \frac{\sqrt{2}\Delta E}{4L\lambda(1+\frac{\Delta E^2}{8\lambda^2})}(1 - \|H\|^2/\lambda^2), \quad (35)$$

$$\mu = \left\lceil \log\left(\frac{2\sqrt{2}\lambda}{\Delta E}\right) + \log\left(1 + \frac{\Delta E^2}{8\lambda^2}\right) - \log\left(1 - \frac{\|H\|^2}{\lambda^2}\right) \right\rceil, \quad (36)$$

where the expression for $\delta$ comes from Eq. (A10) of the Appendix, and it bounds the largest acceptable deviation in the coefficients of the terms in a Hamiltonian approximating the one we mean to implement. The second log in Eq. (36) is $\mathcal{O}(1)$ because we do not take $\Delta E$ larger than $\lambda$. The Hamiltonians we consider are frustrated; thus, $\|H\|/\lambda$ is no larger than a constant (less than 1), and the third log in Eq. (36) is $\mathcal{O}(1)$ as well.

The idea behind our scheme is to create the superposition in an indirect fashion, which involves starting in a uniform superposition over an initial index $\ell$ and then using a precomputed binary representation of a probability (loaded from QROM), keep$_\ell$, to decide whether we should keep $\ell$ or swap it with a classically precomputed alternate index alt$_\ell$, which is also loaded from QROM (see Fig. 11). Specifically, our procedure creates a uniform superposition
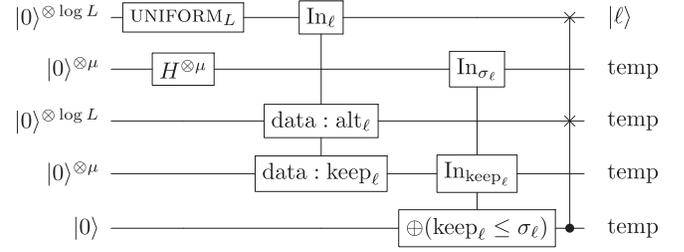


FIG. 11. Generic SUBPREPARE circuit for initializing an arbitrary state with $L$ unique amplitudes. It spans $2\mu + 2\log L + \mathcal{O}(1)$ qubits and has a T count of $4(L + \mu) + \mathcal{O}(\log L)$, where $\mu$ is calculated as in Eq. (36). Produces a state $\sum_{\ell=0}^{L-1} \sqrt{w_\ell/\lambda}\, |\ell\rangle| \text{temp}_\ell\rangle$, where "temp" is temporary garbage data that will be uncomputed when uncomputing the preparation of this state after application of SELECT. The data-loading parts of this circuit use the QROM implementation described in Fig. 10 in Sec. III C. The UNIFORM$_L$ circuit is used to initialize the initial superposition over $\ell$.

in $|\ell\rangle$ over $L$ values and then uses QROM (see Fig. 10 and Sec. III C) to load $|\text{alt}_\ell\rangle$ and $|\text{keep}_\ell\rangle$. Note that if $L$ is not a binary power, one can prepare the initial superposition using the amplitude amplification circuit discussed later in Fig. 12. The procedure described thus far prepares the state

$$\sum_{\ell=0}^{L-1} \sqrt{\frac{1}{L}}\, |\ell\rangle |\text{alt}_\ell\rangle |\text{keep}_\ell\rangle. \quad (37)$$

We then construct a circuit that coherently swaps the registers $|\ell\rangle$ and $|\text{alt}_\ell\rangle$ with probability keep$_\ell$ to create the state in Eq. (34). In order to create the state in Eq. (34) from Eq. (37), we need to introduce one additional register of size $\mu$, which we refer to as $|\sigma\rangle$. We put this entire register into a uniform superposition and then compare it to the probability represented by keep$_\ell$. If keep$_\ell \leq \sigma$, we swap registers $|\ell\rangle$ and $|\text{alt}_\ell\rangle$. Thus, after the procedure is finished, i.e., in Eq. (34), the garbage register will be in the state

$$|\text{temp}_\ell\rangle = \frac{1}{\sqrt{2^\mu L \tilde{\rho}_\ell}} \left( |\text{alt}_\ell\rangle |\text{keep}_\ell\rangle \sum_{\sigma=0}^{\text{keep}_\ell - 1} |\sigma\rangle |0\rangle \right.$$

$$\left. + \sum_{k|\text{alt}_k = \ell} |k\rangle |\text{keep}_k\rangle \sum_{\sigma=\text{keep}_k}^{2^\mu - 1} |\sigma\rangle |1\rangle \right), \quad (38)$$

where the rightmost qubit is the result of a comparison between keep$_\ell$ and $\sigma$. For Eq. (34) to give the correct state, we need $|\text{temp}_\ell\rangle$ to be normalized, which means that we require

$$\frac{\text{keep}_\ell + \sum_{k|\text{alt}_k = \ell}(2^\mu - \text{keep}_k)}{2^\mu L} = \tilde{\rho}_\ell = \frac{\tilde{w}_\ell}{\lambda}, \quad \forall\, \ell \in [0, L). \quad (39)$$
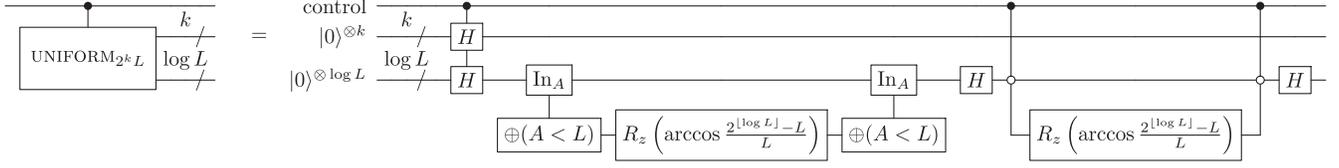
FIG. 12. A circuit that uses amplitude amplification [70,71] to conditionally and reversibly prepare the uniform superposition $(1/\sqrt{2^k L}) \sum_{\ell=0}^{2^k L - 1} |\ell\rangle$, where $L$ is odd, starting from the $|0\rangle$ state. The circuit spans $k + 2 \log L + \mathcal{O}(1)$ qubits and has a T count of $2k + 10 \log L + \mathcal{O}(\log(1/\epsilon))$. If the control is omitted, the T count drops to $8 \log L + \mathcal{O}(\log(1/\epsilon))$. If $L = 1$, the $R_z$ rotations are not needed and the T count drops to $2k$. If $L = 1$ and the control is omitted, the T count is zero.

We can find the values of keep$_\ell$ and alt$_\ell$ from Eq. (39) in a sequential way. At any step, let $\mathcal{L}$ denote the set of $\ell$ for which we have already found these values. Then, we can rewrite Eq. (39) as

$$\frac{\text{keep}_\ell + \sum_{k \notin \mathcal{L} | \text{alt}_k = \ell} (2^\mu - \text{keep}_k)}{2^\mu L}$$
$$= \tilde{\rho}_\ell - \frac{\sum_{k \in \mathcal{L} | \text{alt}_k = \ell} (2^\mu - \text{keep}_k)}{2^\mu L} = \tilde{\rho}'_\ell. \qquad (40)$$

This expression involves only known quantities on the right-hand side, which we call $\tilde{\rho}'_\ell$ for short. We show by induction that the average of $\tilde{\rho}'_\ell$ for $\ell \notin \mathcal{L}$ is $1/L$, and the $\tilde{\rho}'_\ell$ are non-negative. These are clearly true initially because then $\tilde{\rho}'_\ell = \tilde{\rho}_\ell$. Now, assume that these conditions are true at some step. If the values $\tilde{\rho}'_\ell$ are all equal for $\ell \notin \mathcal{L}$, then we can just take alt$_\ell = \ell$ and any value of keep$_\ell$, and satisfy Eq. (40) for all remaining $\ell$. Otherwise, there will be one value, $\ell_0$, where $\tilde{\rho}'_{\ell_0}$ is below the average $1/L$ and another, $\ell_1$, where $\tilde{\rho}'_{\ell_1}$ is above $1/L$. For $\ell_0$, we choose keep$_{\ell_0} = 2^\mu L \tilde{\rho}'_{\ell_0}$ and alt$_{\ell_0} = \ell_1$. Then, $\ell_0$ is added to the set $\mathcal{L}$, and the values of $\tilde{\rho}'_\ell$ are updated. According to Eq. (40), the only value of $\tilde{\rho}'_\ell$ that is updated is that for $\ell = \ell_1$, where we replace it with $\tilde{\rho}'_{\ell_1} + \tilde{\rho}'_{\ell_0} - 1/L$. This ensures that the average value of $\tilde{\rho}'_\ell$ for $\ell \notin \mathcal{L}$ is still $1/L$, and since we had $\tilde{\rho}'_{\ell_1} > 1/L$, the new value is non-negative.

A more intuitive way to understand our approach to preparation is that it is equivalent to classical alias sampling [72], which samples $\ell$ with probability $\tilde{\rho}_\ell$ by the following procedure:

(1) Select $\ell$ uniformly at random from $[0, L)$.
(2) Look up alt$_\ell$ and keep$_\ell$.
(3) Return $\ell$ with probability keep$_\ell/2^\mu$; otherwise return alt$_\ell$.

The procedure for determining the alt$_\ell$ and keep$_\ell$ is then to work backwards starting from the distribution $\tilde{\rho}_\ell$ and update this distribution by shifting probabilities from $\ell_1$ to $\ell_0$ until we obtain a uniform distribution [73].

This procedure is illustrated in Fig. 13. One starts with a histogram of the desired distribution and looks for a bar that is too small, fixes this by transferring probability from a bar that is too high, and so on until all bars have the correct height. Each probability transfer permanently solves the bar that was too low, and the remaining bars form a smaller instance of the same problem. Thus, it is not possible to get stuck in a loop or a dead end. See also the module utils/_lcu_util.py in version 0.6 of OPENFERMION [74,75] for open-source python code that performs this iterative matching process (and also handles discretizing the distribution) in $\mathcal{O}(L)$ time.

## IV. CONSTRUCTIONS FOR THE ELECTRONIC STRUCTURE HAMILTONIAN

Using the appropriate discretization into a basis of $N$ spin orbitals, the electronic structure Hamiltonian can be written as
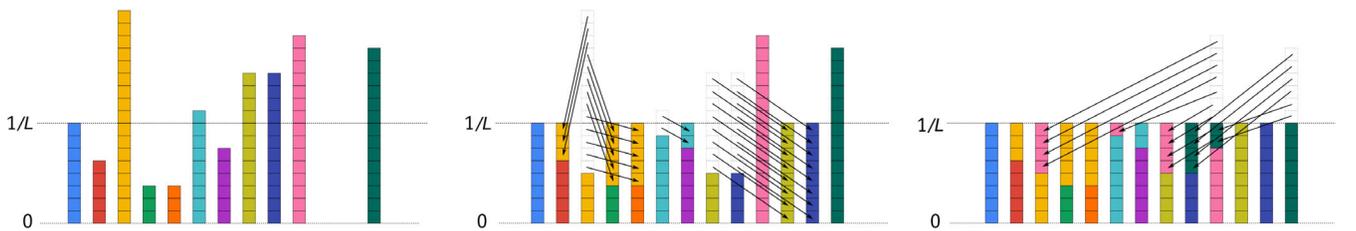


FIG. 13. Depiction of choosing alt$_\ell$ and keep$_\ell$ values for a discretized probability distribution. The left histogram is the input probability distribution, after discretization into steps of $1/(2^3 L)$. The squares must be redistributed so that every column has at most two colors and height of $1/L$ (i.e., eight squares as indicated by the dashed line) without moving the bottom square of each column. The histogram on the right satisfies these constraints, and the histogram in the middle is an intermediate distribution. Arrows indicate which boxes were moved where. The color of the top square of each column in the histogram on the right determines the value of alt$_\ell$, whereas keep$_\ell$ is determined by where the color transition is within each column.

$$H = \sum_{p,q,\sigma} T(p-q) a_{p,\sigma}^\dagger a_{q,\sigma} + \sum_{p,\sigma} U(p) n_{p,\sigma}$$
$$+ \sum_{(p,\alpha)\neq(q,\beta)} V(p-q) n_{p,\alpha} n_{q,\beta}, \tag{41}$$

where $a_{p,\sigma}^\dagger$ and $a_{p,\sigma}$ are fermionic creation and annihilation operators on spatial orbital $p \in \{0, \ldots, N/2 - 1\}$ with spin $\sigma \in \{\uparrow, \downarrow\}$, and $n_{p,\sigma} = a_{p,\sigma}^\dagger a_{p,\sigma}$ is the number operator. These operators satisfy the canonical fermionic anticommutation relations $\{a_{p,\alpha}^\dagger, a_{q,\beta}^\dagger\} = \{a_{p,\alpha}, a_{q,\beta}\} = 0$ and $\{a_{p,\alpha}^\dagger, a_{q,\beta}\} = \delta_{p,q}\delta_{\alpha,\beta}$.

Mapping to qubits under the Jordan-Wigner transformation [61,76], Eq. (41) becomes

$$H = \sum_{p\neq q,\sigma} \frac{T(p-q)}{2}(X_{p,\sigma}\vec{Z}X_{q,\sigma} + Y_{p,\sigma}\vec{Z}Y_{q,\sigma})$$
$$+ \sum_{(p,\alpha)\neq(q,\beta)} \frac{V(p-q)}{4} Z_{p,\alpha}Z_{q,\beta}$$
$$- \sum_{p,\sigma}\left(\frac{T(0) + U(p) + \sum_q V(p-q)}{2}\right) Z_{p,\sigma}$$
$$+ \sum_p \left(T(0) + U(p) + \sum_q \frac{V(p-q)}{2}\right)\mathbb{1}, \tag{42}$$

where we have introduced the notation $\vec{Z}$ that will be used throughout the paper, which we now explain. The tensor factors on which Pauli operators act can always be interpreted as some integer. For instance, $(p,\sigma)$ is mappable to an integer under a particular choice of canonical ordering in the Jordan-Wigner transformation. When placed between two Pauli operators, the notation $A_j\vec{Z}A_k$ denotes the operator $A_jZ_{j+1}\ldots Z_{k-1}A_k$. The exact mapping between a spin orbital indexed by $(p,\sigma)$ and a qubit indexed by an integer is discussed later.

The forms of Eq. (41) and Eq. (42) encompass a wide range of fermionic Hamiltonians, including the molecular electronic structure (aka "quantum chemistry") Hamiltonian in any basis that diagonalizes the Coulomb potential [39]. The particular coefficients will depend on the discretization scheme and basis functions chosen to represent the system. One such representation, derived for use in quantum simulations in Ref. [39], prescribes the coefficients

$$T(p) = \sum_\nu \frac{k_\nu^2 \cos(k_\nu \cdot r_p)}{2N},$$

$$U(p) = -\sum_{j,\nu\neq 0} \frac{4\pi\zeta_j \cos(k_\nu \cdot R_j - k_\nu \cdot r_p)}{\Omega k_\nu^2},$$

$$V(p) = \sum_{\nu\neq 0} \frac{2\pi \cos(k_\nu \cdot r_p)}{\Omega k_\nu^2}, \tag{43}$$

where each spatial orbital $p$ is associated with an orbital centroid $r_p = p(2\Omega/N)^{1/3}$, and $\Omega$ is the computational cell volume. The momentum modes are defined as $k_\nu = 2\pi\nu/\Omega^{1/3}$, with $\nu \in [-(N/2)^{1/3}, (N/2)^{1/3}]^{\otimes 3}$. When dealing with molecular potentials, $R_j$ and $\zeta_j$ are the position and charge of the $j$th nucleus, respectively.

As discussed in Ref. [39], the Hamiltonian of Eq. (43) corresponds to discretization in a basis composed of rotated plane waves known as the "plane wave dual" basis. The basis set discretization error associated with the dual basis is asymptotically equivalent to a Galerkin discretization using any other single-particle basis functions, including Gaussian orbitals [39]. Thus, Eq. (43) is a general expression of the electronic structure problem that is asymptotically equivalent to any other representation. While well suited for simulating periodic materials, despite asymptotic equivalence, this basis set is not particularly compact for the simulation of molecules. Another basis set compatible with Eq. (41) and Eq. (42), while being much more appropriate for molecules, is the so-called "Gausslet" basis [41]. Gausslets are derived from a ternary wavelet transformation [77] of Gaussian orbitals and have similar intrinsic basis set discretization errors to standard Gaussian orbitals [41].

The simulation procedures here will make use of the structure in Eq. (42). Specifically, our algorithm will make use of the fact that the Hamiltonian consists of only four types of terms—$Z_p$, $Z_pZ_q$, $X_p\vec{Z}X_q$, and $Y_p\vec{Z}Y_q$—and that there are only $3N/2$ unique values of the coefficients. Our algorithms do *not* utilize any particular structure in the dual basis Hamiltonian in Eq. (43) beyond the fact that it satisfies the form of Eq. (41). This is important since it implies that the techniques of this paper are compatible with other representations of the electronic structure Hamiltonian, such as the finite difference discretization [39], finite element methods, and Gausslet basis sets [41], which produce Hamiltonians consistent with Eq. (41) but not Eq. (43).

## A. Electronic structure Hamiltonian selection oracle

In order to implement the SELECT and PREPARE oracles for the electronic structure Hamiltonian of Eq. (41), one must first define a scheme for indexing all of the terms. For the case of the general electronic structure Hamiltonian in Eq. (41), we index terms with the registers $|\theta\rangle$, $|U\rangle$, $|V\rangle$, $|p\rangle$, $|\alpha\rangle$, $|q\rangle$, and $|\beta\rangle$. The $|p\rangle$ and $|q\rangle$ registers are little-endian binary encodings of integers going from 0 to $N/2 - 1$, thus using $\log N - 1$ qubits each; the other registers are each a single bit, which we use to specify the unitary that SELECT should apply to the system register $|\psi\rangle$.

The $|\alpha\rangle$ and $|\beta\rangle$ bits are used to specify the spins $\{\uparrow, \downarrow\}$, which, together with the spatial orbital specifications $p$ and $q$, index a spin orbital. Thus, a register set as $|p\rangle|\alpha\rangle|q\rangle|\beta\rangle$

will index a Hamiltonian term that involves action on the spin orbitals indexed by $(p, \alpha)$ and $(q, \beta)$. Next, whenever $|U\rangle = |1\rangle$, it will be the case (by construction of our circuits) that $(p, \alpha) = (q, \beta)$, and we will apply the $Z_{p,\alpha}$ terms. If $|V\rangle = |1\rangle$, we will apply the $Z_{p,\alpha} Z_{q,\beta}$ terms. If $|U\rangle |V\rangle = |0\rangle |0\rangle$ and $p < q$, it will also be the case that $\alpha = \beta$, and we will apply the $X_{p,\alpha} \vec{Z} X_{q,\alpha}$ terms; if $|U\rangle |V\rangle = |0\rangle |0\rangle$ and $p > q$, it will again be the case that $\alpha = \beta$, and we will apply the $Y_{q,\alpha} \vec{Z} Y_{p,\alpha}$ terms. Finally, the $|\theta\rangle$ register encodes whether the unitary should have a negative phase (if $|\theta\rangle = |1\rangle$). Thus, our SELECT circuit meets the following specification (where UNDEFINED means this case should not occur):

$$
\begin{aligned}
&\text{SELECT}_{\text{CHEM}} |\theta, U, V, p, \alpha, q, \beta\rangle |\psi\rangle \\
&= (-1)^\theta |\theta, U, V, p, \alpha, q, \beta\rangle \\
&\otimes \begin{cases}
Z_{p,\alpha} |\psi\rangle & U \wedge \neg V \wedge ((p, \alpha) = (q, \beta)) \\
Z_{p,\alpha} Z_{q,\beta} |\psi\rangle & \neg U \wedge V \wedge ((p, \alpha) \neq (q, \beta)) \\
X_{p,\alpha} \vec{Z} X_{q,\alpha} |\psi\rangle & \neg U \wedge \neg V \wedge (p < q) \wedge (\alpha = \beta) \\
Y_{q,\alpha} \vec{Z} Y_{p,\alpha} |\psi\rangle & \neg U \wedge \neg V \wedge (p > q) \wedge (\alpha = \beta) \\
\text{UNDEFINED} & \text{otherwise.}
\end{cases}
\end{aligned}
$$

$$(44)$$

We present our implementation of SELECT$_{\text{CHEM}}$ in Fig. 14. Our circuit relies on the subroutines that we describe in Sec. III A, which provide a method for selectively applying strings of Pauli operators to a system register of size $N$, with controls on $\log N$ qubits. Important notation for these subroutines is also defined in Sec. III A, and thus, that section is necessary for understanding the details of Fig. 14.

Since $p$ and $q$ are actually three-dimensional vectors with elements taking integer values $p \in [0, (N/2)^{1/3} - 1]$ and $\sigma \in \{\uparrow, \downarrow\}$, we should clarify how the spin orbitals $(p, \sigma)$ are mapped to an integer representing qubits. For ease of exposition, we define the following mapping function for a $D$-dimensional system,

$$
M \equiv (N/2)^{1/D}, \qquad f(p, \sigma) = \delta_{\sigma,\downarrow} M^D + \sum_{j=0}^{D-1} p_j M^j,
$$

$$(45)$$

where $D = 3$ for chemistry and $D = 2$ for the Hubbard model. The $\delta$ function behaves as one might expect: $\delta_{\uparrow,\downarrow} = 0$ and $\delta_{\downarrow,\downarrow} = 1$. Thus, it should be understood that $X_{p,\sigma}$ implies the $X$ operator acting on qubit $f(p, \sigma)$.

## B. Electronic structure coefficient preparation oracle

We see from Eq. (41) that there are only $\mathcal{O}(N)$ unique coefficients in the Hamiltonian, despite the Hamiltonian having $\mathcal{O}(N^2)$ different terms. Based on the indexing in Eq. (44) and definition in Eq. (6), our PREPARE initializes
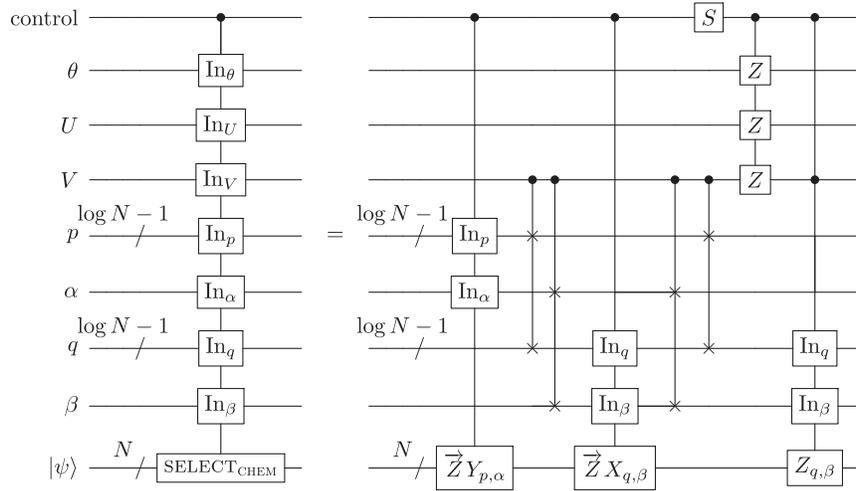


FIG. 14. SELECT$_{\text{CHEM}}$ circuit, with a T count of $12N + 8 \log N + \mathcal{O}(1)$, which implements the functionality specified by Eq. (44), conditioned on a "control" qubit. The unitaries performing Majorana and indexed operations (each requiring $4N$ T gates) are explicitly constructed in Sec. III A. As described in Fig. 9, the unitaries labeled as $\vec{Z} A_j$ apply the operation $Z_0 \ldots Z_{j-1} A_j$ to the target register, depending on the value from the input $|p\rangle$ register. These operations require an extra $\log N$ ancillae, so the overall circuit spans $N + 3 \log N + \mathcal{O}(1)$ qubits. The operation that targets the system register with $Z_q$ is a variant of Fig. 7, with the $X_\ell$ gate replaced by $Z_\ell$. All indexed operations reuse the same ancillae.

$$\text{PREPARE}_{\text{CHEM}}|0\rangle^{\otimes(3+2\log N)}$$

$$\mapsto \sum_{p,\sigma} \tilde{U}(p)|\theta_p\rangle|1\rangle_U|0\rangle_V|p,\sigma,p,\sigma\rangle$$

$$+ \sum_{p\neq q,\sigma} \tilde{T}(p-q)|\theta_{p-q}^{(0)}\rangle|0\rangle_U|0\rangle_V|p,\sigma,q,\sigma\rangle$$

$$+ \sum_{(p,\alpha)\neq(q,\beta)} \tilde{V}(p-q)|\theta_{p-q}^{(1)}\rangle|0\rangle_U|1\rangle_V|p,\alpha,q,\beta\rangle, \quad (46)$$

where the values of the coefficients and the state of $|\theta\rangle$, related to the coefficients in Eq. (42), are defined as

$$\tilde{U}(p) = \sqrt{\frac{|T(0) + U(p) + \sum_q V(p-q)|}{2\lambda}},$$

$$\tilde{T}(p) = \sqrt{\frac{|T(p)|}{\lambda}}, \qquad \tilde{V}(p) = \sqrt{\frac{|V(p)|}{4\lambda}},$$

$$\theta_p = \frac{1 - \text{sign}(-T(0) - U(p) - \sum_q V(p-q))}{2},$$

$$\theta_p^{(0)} = \frac{1 - \text{sign}(T(p))}{2}, \qquad \theta_p^{(1)} = \frac{1 - \text{sign}(V(p))}{2}.$$

$$(47)$$

The $T(p)$ coefficient inside the square root in Eq. (47) differs from the coefficient in Eq. (42) by a factor of 2 since it occurs for each type of term only once depending on whether $p < q$ or $p > q$.

To implement PREPARE, we first synthesize a unitary referred to as SUBPREPARE, which acts as follows:

$$\text{SUBPREPARE}|0\rangle^{\otimes(2+\log N)}$$

$$\mapsto \sum_{d=0}^{N-1} (\tilde{U}(d)|\theta_d\rangle|1\rangle_U|0\rangle_T + \tilde{T}(d)|\theta_d^{(0)}\rangle|0\rangle_U|0\rangle_V$$

$$+ \tilde{V}(d)|\theta_d^{(1)}\rangle|0\rangle_U|1\rangle_V)|d\rangle. \quad (48)$$

Since in this step we initialize a state on $\mathcal{O}(\log N)$ qubits, the techniques of Ref. [69] would allow one to implement SUBPREPARE with a T count of $\mathcal{O}(N\log(1/\epsilon))$. However, in Fig. 15, we show an even more efficient method for synthesizing SUBPREPARE with T-gate complexity $\mathcal{O}(N + \log(1/\epsilon))$, based on the techniques introduced in Sec. III D. Using SUBPREPARE, we can implement the entire PREPARE circuit with the same asymptotic T complexity. In our SUBPREPARE circuit, $\ell$ is really a vector of integers; thus, we use "modular vector indices" such that if $v$ is a three-dimensional vector within a rectangular space with each dimension having $M$ values, then the function application $F(v)$ should be expanded to $F(v) = F(v \mod M) = F(v_x \mod M, v_y \mod M, v_z \mod M)$, consistent with the mapping introduced in Eq. (45).

While applying SUBPREPARE to create the state in Eq. (48), we also initialize the $|\alpha\rangle$ qubit in the $|+\rangle$ state with a Hadamard. We then use the UNIFORM$_L$ circuit from Fig. 12 to initialize the $|q\rangle$ register in an equal superposition in a way that is controlled on the $|U\rangle$ ancilla qubit being in the state $|0\rangle_U$. Subsequent to this step, the state becomes
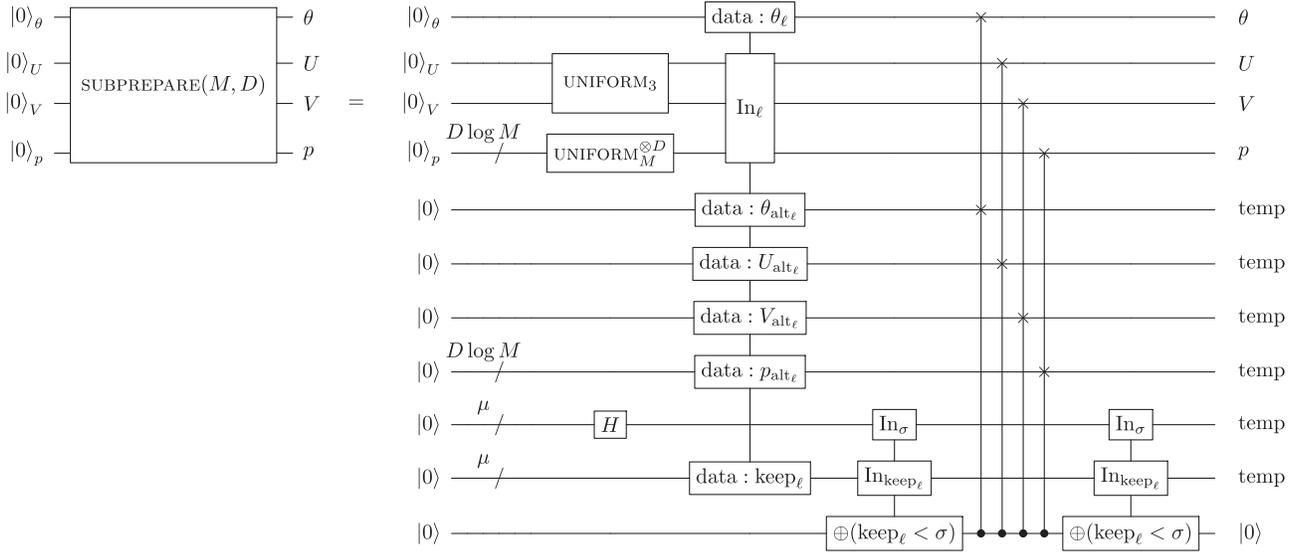


FIG. 15.    SUBPREPARE circuit for the electronic structure Hamiltonian, as in Fig. 16, with a T count of $6N + \mathcal{O}(\mu + \log N)$ and a qubit count of $2\mu + 3\log N + \mathcal{O}(1)$, where $\mu$ is defined in Eq. (36). The data-loading subroutine is implemented as in Fig. 10 and has a T count of $3 \times 4M^3 - 4 = 6N - 4$. The UNIFORM subroutine is implemented as in Fig. 12 and has a T count of $\mathcal{O}(\mu)$. The compare-and-swap operations have a negligible $\mathcal{O}(\log N)$ T count. As in Eq. (45), $D$ denotes the system dimension (usually $D = 3$), and $M$ refers to the number of values along each dimension such that $N = 2M^D$. Although we only specify the behavior of the circuit when the $U$, $V$, and $p$ qubits start in the $|0\rangle$ state, the circuit is also invoked in contexts where this is not the case.

$$\text{Eq. (48)} \mapsto \sum_{d=0}^{N/2-1} \tilde{U}(d)|\theta_d\rangle|1\rangle_U|0\rangle_V|d\rangle|+\rangle_\alpha|0\rangle^{\otimes \log N}$$

$$+ \sum_{d=0}^{N/2-1}\sum_{q=0}^{N/2-1}(\tilde{T}(d)|\theta_d^{(0)}\rangle|0\rangle_U|0\rangle_V$$

$$+ \tilde{V}(d)|\theta_d^{(1)}\rangle|0\rangle_U|1\rangle_V)|d\rangle|+\rangle_\alpha|q\rangle|0\rangle. \qquad (49)$$

The register labeled as $|d\rangle$ in Eq. (48) will ultimately become our $|p\rangle$ register, but immediately after SUBPREPARE, it is more appropriate to think of it as encoding a value $|p-q\rangle$. As we can see in Eq. (46), when $|V\rangle = |1\rangle$ and

$p=q$, it is necessarily the case that $\alpha \neq \beta$. The middle part of our PREPARE circuit is dedicated to correctly initializing this tricky part of the superposition. To do this, we use an ancilla to apply a Hadamard gate to $|0\rangle_\beta$ only when $|V\rangle = |1\rangle$ and $|d\rangle \neq |0\rangle^{\otimes(\log N-1)}$. In the event that $|V\rangle = |1\rangle$ and $|d\rangle = |0\rangle^{\otimes(\log N-1)}$, we apply a CNOT gate with an open control on $|\alpha\rangle$ which targets $|0\rangle_\beta$, thus ensuring that $|\beta\rangle \neq |\alpha\rangle$ when $p-q=0$. Then, we set $|\beta\rangle = |\alpha\rangle$ for the $U$ and $T$ part of the superposition by applying a Toffoli gate with regular control on $|\alpha\rangle$ and open control on $|V\rangle$, targeting $|\beta\rangle$. After these operations, the state can be expressed as

$$\text{Eq. (49)} \mapsto \sum_{d=0}^{N/2-1}\sum_\sigma \left( \tilde{U}(d)|\theta_d\rangle|1\rangle_U|0\rangle_V|d,\sigma,0,\sigma\rangle + \sum_{q=0}^{N/2-1} \tilde{T}(d)|\theta_d^{(0)}\rangle|0\rangle_U|0\rangle_V|d,\sigma,q,\sigma\rangle \right)$$

$$+ \sum_\alpha \left( \tilde{V}(0)|\theta_0^{(1)}\rangle|0\rangle_U|1\rangle_V|0,\alpha,q,\neg\alpha\rangle + \sum_\beta\sum_{d=1}^{N/2-1}\sum_{q=0}^{N/2-1} \tilde{V}(d)|\theta_d^{(1)}\rangle|0\rangle_U|1\rangle_V|d,\alpha,q,\beta\rangle \right). \qquad (50)$$

The final step consists of converting the $|d\rangle$ register to values representing $|p\rangle$. To do this, we must add the $|q\rangle$ register into the $|d\rangle$ register when $|U\rangle = |0\rangle$ so that $|d+q\rangle = |p-q+q\rangle = |p\rangle$. However, we also want to copy the $|d\rangle$ register into the $|q\rangle$ register when $|U\rangle = |1\rangle$; thus, prior to this operation, we also implement a Fredkin gate, which swaps $|d\rangle$ and $|q\rangle$, conditioned on $|U\rangle = |1\rangle$. After the Fredkin gate and the addition of $|d\rangle$ into $|q\rangle$,

$$\text{Eq. (50)} \mapsto \sum_{d=0}^{N/2-1}\sum_\sigma \left( \tilde{U}(d)|\theta_d\rangle|1\rangle_U|0\rangle_V|d,\sigma,d,\sigma\rangle + \sum_{q=0}^{N/2-1} \tilde{T}(d)|\theta_d^{(0)}\rangle|0\rangle_U|0\rangle_V|d+q,\sigma,q,\sigma\rangle \right)$$

$$+ \sum_\alpha \left( \tilde{V}(0)|\theta_0^{(1)}\rangle|0\rangle_U|1\rangle_V|q,\alpha,q,\neg\alpha\rangle + \sum_\beta\sum_{d=1}^{N/2-1}\sum_{q=0}^{N/2-1} \tilde{V}(d)|\theta_d^{(1)}\rangle|0\rangle_U|1\rangle_V|d+q,\alpha,q,\beta\rangle \right). \qquad (51)$$

Then, simply by relabeling $d = p - q$ whenever $|U\rangle = |0\rangle$ and $d = p$ whenever $|U\rangle = 1$, we see that our state is identical to the desired one [from Eq. (46)]. We show how to use SUBPREPARE to implement PREPARE$_{\text{CHEM}}$ in Fig. 16. The gate complexity of SUBPREPARE is $\mathcal{O}(N + \log(1/\epsilon))$, and the gate complexity of all other components of this circuit is $\mathcal{O}(\log N)$. Thus, the overall gate complexity of PREPARE is $\mathcal{O}(N + \log 1/\epsilon)$.

## C. Resources required for electronic structure simulation

The parameter $\lambda$ from Eq. (6) has significant implications for the complexity of our algorithm; as seen in Eq. (27), our circuit size will scale linearly in $\lambda$. For the case of general electronic structure, we can see from Eq. (41) that $\lambda$ is

This expression and the extremely naive assumption that all coefficients are $\mathcal{O}(1)$ would imply that $\lambda \in \mathcal{O}(N^2)$. For the case of quantum chemistry in the dual basis, i.e., Eq. (43), the work of Ref. [39] obtains the same bound:

$$\lambda \in \mathcal{O}\left(\frac{N^{7/3}}{\Omega^{1/3}} + \frac{N^{5/3}}{\Omega^{2/3}}\right) \in \mathcal{O}(N^2), \qquad (53)$$

where the last relation holds when studying electronic structure systems that grow with fixed density $N \propto \Omega$, which is the usual situation. For encoding the electronic structure Hamiltonian, we also determine that $P = 6N + \mathcal{O}(\log(N/\epsilon))$ and $S = 12N + \mathcal{O}(\log N)$ in terms of T complexity. Thus, from Eq. (27), we can conclude that the overall T complexity of our procedure is roughly

$$\lambda = \sum_{pq}|T(p-q)| + \sum_p|U(p)| + \sum_{p\neq q}|V(p-q)|. \qquad (52)$$

$$\frac{\sqrt{2}\pi\lambda(S+2P)}{\Delta E} \approx \frac{24\sqrt{2}\pi\lambda}{\Delta E}N, \qquad (54)$$
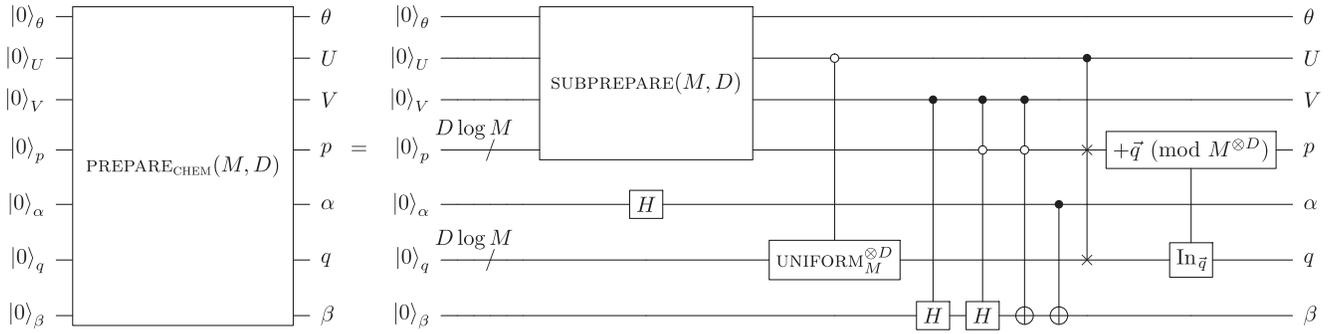
FIG. 16.    PREPARE circuit for the electronic structure Hamiltonian. It implements the unitary in Eq. (46) with a T count of $6N + \mathcal{O}(\mu + \log N)$, where $\mu$ is defined in Eq. (36). The SUBPREPARE subroutine is the dominant cost, and it is implemented as in Fig. 15. When $U$ is set by SUBPREPARE, the Fredkin gates and the addition copy $p$'s value over $q$. When $V$ is set, uniform superpositions over $q$ and $\beta$ are prepared except, if $p - q = 0$, then $\beta$ is instead set to be opposite to $\alpha$ in order to guarantee $(p, \alpha) \neq (q, \beta)$. Beware that this conditional preparation of $\beta$ doubles the weight of the $p = q$ cases (relative to the $p \neq q$ cases) and must be accounted for in the LCU coefficients given to SUBPREPARE. When neither $U$ nor $V$ is set, $\alpha$ is copied into $\beta$, and a uniform superposition over $q$ is prepared. The remaining operations contribute negligible $\mathcal{O}(\mu + \log N)$ T gates. The UNIFORM$_M^{\otimes D}$ operation prepares $D$ registers in a uniform superposition of basis states going up to $M$, as in Fig. 12. Controls on multiqubit lines are conditioned on every qubit within the line; e.g., what appears to be a Toffoli gate is actually a NOT gate with $1 + D \log M$ controls. The action of our PREPARE$_{\text{CHEM}}$ circuit is only specified in the case where the inputs are all $|0\rangle$. During actual execution, this is not the case because the effects of the SELECT operation will prevent PREPARE$_{\text{CHEM}}^\dagger$ from exactly uncomputing the $U$, $V$, $p$, $q$, $\alpha$, and $\beta$ qubits. This is expected behavior, and it is accounted for by requiring that the potentially not-uncomputed qubits be kept and used as inputs for the next PREPARE$_{\text{CHEM}}$ circuit.

which, for the electronic structure Hamiltonian, is rigorously bounded by $\mathcal{O}(N^3/\Delta E)$.

Ancilla required for our electronic structure simulation come from three sources: qubits required for our entanglement-based phase estimation [given by Eq. (24)], qubits required to store coefficient values in QROM [given by Eq. (36)], and ancilla actually required for our implementation of PREPARE and SELECT, which, for the electronic structure Hamiltonian simulation, is $5 \log N + \mathcal{O}(1)$. Putting these sources together, the total ancillae required are

$$\log\left(\frac{\sqrt{2}\pi\lambda}{2\Delta E}\right) + 2\log\left(\frac{2\sqrt{2}\lambda}{\Delta E}\right) + 5\log N + \mathcal{O}(1)$$

$$= \log\left(\frac{4\sqrt{2}\pi\lambda^3 N^5}{\Delta E^3}\right) + \mathcal{O}(1), \tag{55}$$

where the additive constant is small and can usually be neglected for problem sizes of interest. This expression gives the ancilla count in Theorem 1.

We now estimate resources required for specific problem instances. In practice, we find that $\lambda$ scales better than $\mathcal{O}(N^2)$, but exactly how much better is system dependent. For a particular material, the value of $\lambda$ can be influenced by a number of factors. These factors include the particulars of the bases used, the geometry and atomic composition of the material, and whether one scales toward continuum or thermodynamic limits.

Perhaps the simplest chemistry system that is classically intractable is a molecule without nuclei: the uniform electron gas, also known as jellium. Jellium is a system

of $\eta$ electrons with real kinetic energy and Coulomb interactions confined to a box of finite volume $\Omega$ with periodic boundary conditions. Plane waves are a near-ideal basis for the simulation of jellium; the system is naturally expressed using the discretization of Eq. (43) with a constant external potential, i.e., $\zeta_j = 0$. Jellium is an interesting system to simulate on early quantum computers due to its simplicity, classical intractability [39], historical significance tied to breakthroughs in density functional theory [78] as well as the fractional quantum Hall effect [79], and tradition as a benchmark for classical electronic structure calculations.

The phase diagram of jellium is typically parametrized in terms of the Wigner-Seitz radius, which characterizes the electron density in three dimensions as $r_s = (3\Omega/(4\pi\eta))^{1/3}$, where $\eta$ is the number of electrons. Although the ground state of jellium at high densities (metallic, $r_s \sim 1$ Bohr radii per particle) and at very low densities (insulating, $r_s \sim 100$ Bohr radii per particle) is well known, the phase diagram in the intermediate density regime is less certain [80–85]. Whereas perturbation theory performs well in the high-density regime [86,87], quantum Monte Carlo has been the most competitive simulation tool in the low- to intermediate-density regimes [88–91]. For systems with more than 50 electrons, quantum Monte Carlo simulations of jellium typically introduce a bias to control the sign problem, such as the fixed-node approximation, full-configuration quantum Monte Carlo with initiators, or auxiliary-field quantum Monte Carlo with a constrained phase bias. The systematic error from these biases is thought to be as large as half a percent in the energy

[81,88], which is on a scale similar to the energy difference between competing phases in the intermediate-density regime. Even for modest system sizes such as 50 electrons and twice as many spin orbitals, quantum simulations can offer bias-free results that cannot be obtained by quantum Monte Carlo.

We include numerics in Fig. 17 that empirically estimate a tighter bound on $\lambda$ for jellium in the classically challenging regime corresponding to $r_s = 10$ Bohr radii at half-filling $N = \lceil 2\eta \rceil$. Those numerics, shown in Fig. 17, indicate an empirical scaling of $\lambda = \mathcal{O}(\sim N^{5/3})$. If we target a chemical accuracy of $\Delta E = 0.0016$ Hartree, then from Eq. (54), we see that roughly $2 \times 10^7$ T gates would be required for jellium with 54 orbitals, $2 \times 10^8$ T gates would be required for jellium with 128 orbitals, and about a billion T gates would be required for jellium with 250 orbitals. While these numbers are promising, for small sizes, these simulations require a number of ancilla comparable to $N$. T counts and ancilla resources are tabulated for several jellium problem instances in Table III.

The dual basis of Eq. (43) is also a natural choice for periodic condensed phase systems (e.g., solids) besides jellium. Considering only this basis, there are two parameters that determine the accuracy of the simulation with respect to the true material. The first one is the number of plane waves used to discretize the cell, which determines the spacing of the quasipoints in the dual basis [39]. More plane waves equate to a finer grid and more accurate discretization. The second parameter is the size of the
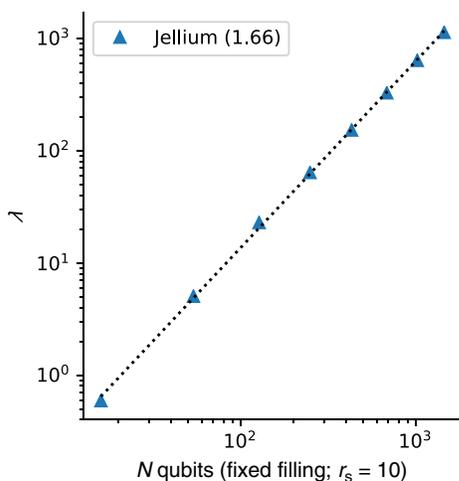
TABLE III. Resources required for quantum simulation of 3D spinful jellium in the dual basis at a Wigner-Seitz radius of 10 Bohr radii, where the cell volume is calculated assuming the system is at half-filling. The units for $\lambda$ are Hartree. The number of logical ancillae is computed using Eq. (55), and the number of T gates is computed using Eq. (54). These estimates assume (rather conservatively in comparison to classical limitations) that we should target an additive chemical accuracy of $\Delta E = 0.0016$ Hartree. These problem sizes are large enough that contemporary classical methods cannot reliably provide unbiased estimates with low enough systematic error to resolve competing phases within the fixed-size basis.

| Spin orbitals | $\lambda$ value | Logical ancilla | Total logical | T count |
|---|---|---|---|---|
| 54 | 5 | 69 | 123 | $1.8 \times 10^7$ |
| 128 | 23 | 82 | 210 | $1.9 \times 10^8$ |
| 250 | 64 | 91 | 341 | $1.1 \times 10^9$ |
| 1024 | 640 | 112 | 1136 | $4.3 \times 10^{10}$ |

supercell, which determines the error one incurs by representing an infinite system with a finite, periodic one, also known as the finite-size error. There are different ways of reducing the finite-size error for a physical system. One common method used in density functional theory utilizes Bloch's theorem to divide the sampling problem into so-called "k-points" within the first Brillouin zone [92]. The smoothness of the energy with respect to the k-points and additional symmetry provided can offer advantages in certain approaches at the cost of increased complexity, often resulting in a complex Hamiltonian representation at nonzero k-points. The origin k-point, also called the gamma point, maintains a real Hamiltonian for the appropriate basis functions. An alternative to k-point sampling is increasing the size of the supercell, which increases the relevance of the gamma point. For simplicity, here we only consider the gamma point, so the natural parameter to change is the number of repetitions of the unit cell that fixes the size of the supercell being simulated. A larger supercell tends to incur less finite-size error as the system is scaled to the thermodynamic limit.

It is clear that these two parameters are not entirely independent with respect to the accuracy of representation of the true system. For example, a much larger supercell with the same number of grid points clearly offers a coarser and less accurate representation of the true system. Moreover, in classical methods, it is common practice to extrapolate along both parameters to increase the accuracy for a given computational cost [92]. We do not introduce such complexities here but rather show empirically how the choice of these parameters influences the parameter $\lambda$, which determines the cost of our algorithms, leaving optimizations such as extrapolation schemes to future work.

Figure 18 shows the value of $\lambda$ as a function of the number of qubits being used to discretize the material cell at a fixed supercell size for several real materials.



FIG. 17. The value $\lambda$ as a function of the number of qubits $N$ for the 3D spinful dual-basis jellium Hamiltonian at a Wigner-Seitz radius of 10 Bohr radii (assuming the system is initialized at half-filling so $N = \lceil 2\eta \rceil$), which corresponds to an increasing cell volume as the number of basis functions increases. This density was chosen for study since it is in the classically challenging regime for jellium [39]. The bound is in atomic units of energy (Hartree). The number in parentheses corresponds to the best-fit exponent for the trend line (the dotted line), which suggests that $\lambda = \mathcal{O}(\sim N^{5/3})$.
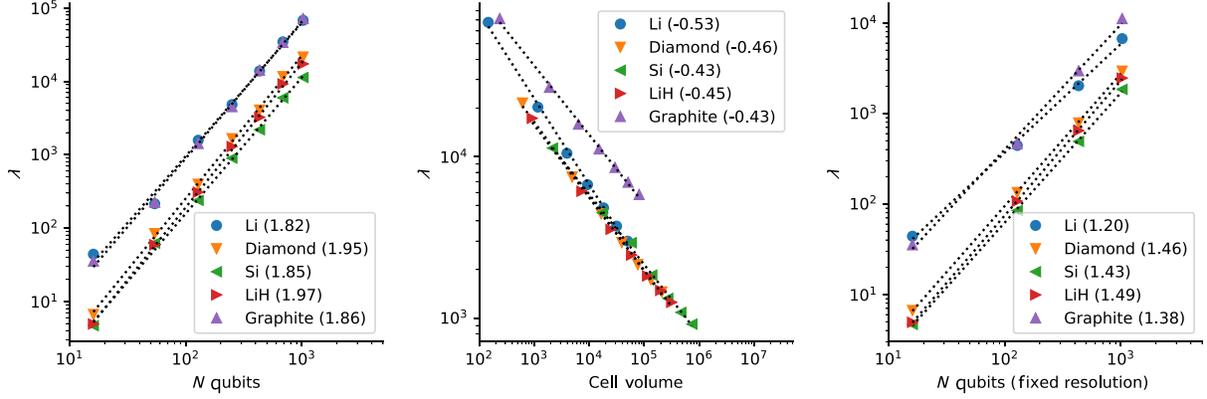
FIG. 18.    Left panel: A plot of the $\lambda$ value for one unit cell of the listed material near equilibrium bond length as a function of the number of qubits used to discretize the cell. A plane-wave dual basis is used with an equal number of points along each of the axes. The value given after the material name corresponds to the best-fit scaling for that particular material. The materials display roughly similar scaling in $\lambda$ as a function of the number of qubits. Center panel: A plot of the $\lambda$ value as a function of the cell volume for a fixed number of qubits, 1024 in this case. We observe that fixing the number of qubits while increasing the cell size tends to decrease $\lambda$ as expected. Right panel: A plot of the $\lambda$ value as a function of the number of qubits, where the supercell size is scaled proportionally to the number of points along each axis. In this case, two basis functions are used along each axis in the unit cell, which is scaled proportionally as the cell volume grows.

The number of qubits here is equal to twice the number of plane waves since spin is being treated explicitly. Equal numbers of plane waves along each of the reciprocal lattice vectors of the supercell are used, as opposed to the more common spherical energy cutoff schemes [93], as this enables the use of the plane-wave dual basis [39]. The exponent (slope on the log-log plot) of a least-squares linear regression to the data is listed alongside the material, and we see empirically that the value of $\lambda$ scales just under $\lambda = \mathcal{O}(N^2)$ in the number of basis functions while keeping the size of the supercell fixed, which matches the analytical bound rather closely. From the form of the Hamiltonian in Eq. (43), one can see that at a fixed number of plane waves, increasing the volume of the supercell $\Omega$ tends to decrease $\lambda$ such that $\lambda = \mathcal{O}(\sim\Omega^{-1/2})$, due to representing lower-frequency modes with respect to the plane-wave representation of the kinetic energy, despite increasing the total nuclei present. We show this effect empirically in the center of Fig. 18.

The first two panels of Fig. 18 leave open the question of the impact on $\lambda$ of increasing the supercell size while maintaining a constant density of dual quasipoints or a constant density of plane waves, as we expect the impact of the last two aspects to compete in some way. Empirically, this is shown in the right portion of Fig. 18, which plots the values of $\lambda$ for a fixed density of points in increasing supercell sizes. Note that this scaling is most comparable to past studies on single molecules since molecular volume tends to grow as one adds electrons. We observe that the scaling in this case is more favorable as a function of the number of qubits than simply refining the grid alone, and in all cases, it is better than $\lambda = \mathcal{O}(\sim N^{3/2})$. From Eq. (54), this numerical data would suggest that the T complexity of our overall algorithm is empirically bounded by $\mathcal{O}(\sim N^{5/2}/\Delta E)$ when the goal is simulation of real materials.

To treat molecules properly, one should further consider pseudopotentials [94], methods of extrapolation to continuum and thermodynamic limits, and embedding methods [95,96]. We leave such a thorough comparison of fault-tolerant resources required for specific instances of real materials other than jellium to future work. However, by comparing Fig. 17 and the right panel of Fig. 18, it is apparent that jellium is a reasonable proxy for other materials in that $\lambda$ values are comparable. As the rest of the simulation circuit is identical up to the particular angles of certain single-qubit rotations, one can estimate the cost of simulating these materials from our analysis of the fault-tolerant overheads required to simulate jellium in Sec. VI.

## V. CONSTRUCTIONS FOR THE HUBBARD MODEL

In this section, we describe specialized implementations of the SELECT and PREPARE oracles for simulation of the planar repulsive-interaction Fermi-Hubbard model; we then estimate the overall T complexity of simulating such models. The Hubbard model is a canonical model of a many-electron system often used to model superconductivity in cuprate superconductors. Despite its simplicity, the Hubbard model exhibits a wide range of correlated electron behavior including superconductivity, magnetism, and interaction-driven metal-insulator transitions [97].

The Hubbard model is essentially a special case of Eq. (41) when the model is restricted to a planar grid. The Hamiltonian can be expressed as

$$H = -t \sum_{\langle p,q\rangle,\sigma} a_{p,\sigma}^\dagger a_{q,\sigma} + \frac{u}{2} \sum_{p,\alpha\neq\beta} n_{p,\alpha} n_{p,\beta}, \qquad (56)$$

where the notation $\langle p, q \rangle$ implies that terms exist only between sites that are adjacent on a planar lattice with periodic boundary conditions. This Hamiltonian can be expressed under the Jordan-Wigner transformation as

$$H = -\frac{t}{2} \sum_{\langle p,q \rangle, \sigma} (X_{p,\sigma}\vec{Z}X_{p,\sigma} + Y_{p,\sigma}\vec{Z}Y_{p,\sigma})$$
$$+ \frac{u}{8} \sum_{p, \alpha \neq \beta} Z_{p,\alpha}Z_{p,\beta} - \frac{u}{4}\sum_{p,\sigma} Z_{p,\sigma} + \frac{uN}{4}\mathbb{1}. \quad (57)$$

We focus on the Hubbard model with periodic boundary conditions (which is a more typical system to study than the Hubbard model with open boundary conditions).

### A. Hubbard model Hamiltonian selection oracle

We see from Eq. (56) that there are only three unique coefficients in the Hubbard Hamiltonian: The coefficient of $-XZX$ and $-YZY$ terms is $t/2$, the coefficient of $ZZ$ terms

is $u/8$, and the coefficient of local $-Z$ terms is $u/4$. This makes the implementation of the PREPARE circuit exceptionally simple. Ultimately, we show that the PREPARE circuit for the Hubbard model can be implemented at a cost of $\mathcal{O}(\log(1/\epsilon))$. This scaling virtually guarantees that for all problem sizes of interest, the scaling of the overall algorithms will be dominated by the cost of the SELECT circuit.

We index terms in the Hubbard Hamiltonian using the registers $|U\rangle|V\rangle|p_x\rangle|p_y\rangle|\alpha\rangle|q_x\rangle|q_y\rangle|\beta\rangle$. Note that it is important for us to explicitly separate $p_x$ and $p_y$ in our construction of the Hubbard model circuits since this structure is fundamental to the efficiency of our scheme. Our indexing scheme is nearly identical to the scheme used for the arbitrary chemistry Hamiltonian in Eq. (7), but here we do not need the $\theta$ parameter since we know the sign of the parameters in advance. Thus, our SELECT circuit for the Hubbard model will meet the following specifications:

$$\text{SELECT}_{\text{HUB}}|U, V, p, \alpha, q, \beta\rangle|\psi\rangle$$

$$= |U, V, p, \alpha, q, \beta\rangle \otimes \begin{cases} -Z_{p,\alpha}|\psi\rangle & U \wedge \neg V \wedge ((p, \alpha) = (q, \beta)) \\ Z_{p,\alpha}Z_{q,\beta}|\psi\rangle & \neg U \wedge V \wedge (p = q) \wedge (\alpha = 0) \wedge (\beta = 1) \\ -X_{p,\alpha}\vec{Z}X_{q,\alpha}|\psi\rangle & \neg U \wedge \neg V \wedge (p < q) \wedge (\alpha = \beta) \\ -Y_{q,\alpha}\vec{Z}Y_{p,\alpha}|\psi\rangle & \neg U \wedge \neg V \wedge (p > q) \wedge (\alpha = \beta) \\ \text{UNDEFINED} & \text{otherwise}, \end{cases} \quad (58)$$

where, for ease of exposition, $p = p_x + p_y M$ and $q = q_x + q_y M$, consistent with the convention of Eq. (45) for $D = 2$. By exploiting translational invariance in the Hubbard model, we are able to implement SELECT$_{\text{HUB}}$ in a slightly more efficient fashion, achieving a T count of only $10N + \mathcal{O}(\log N)$. We show this more efficient implementation in Fig. 19.

### B. Hubbard model coefficient preparation oracle

Our PREPARE circuit for the Hubbard model has the following specification:

$$\text{PREPARE}_{\text{HUB}}|0\rangle^{\otimes(2+2\log N)}$$

$$\mapsto \sum_{p_x=0}^{M-1} \sum_{p_y=0}^{M-1} \left( \sqrt{\frac{u}{8\lambda}}|0\rangle_U|1\rangle_V|p_x\rangle|p_y\rangle|0\rangle_\alpha|p_x\rangle|p_y\rangle|1\rangle_\beta + \sqrt{\frac{u}{4\lambda}}\sum_{\sigma\in\{\downarrow,\uparrow\}}|1\rangle_U|0\rangle_V|p_x\rangle|p_y\rangle|\sigma\rangle|p_x\rangle|p_y\rangle|\sigma\rangle \right.$$

$$+ \sqrt{\frac{t}{2\lambda}}\sum_{\sigma\in\{\downarrow,\uparrow\}}(|0\rangle_U|0\rangle_V|p_x\rangle|p_y\rangle|\sigma\rangle|p_x+1\rangle|p_y\rangle|\sigma\rangle + |0\rangle_U|0\rangle_V|p_x\rangle|p_y\rangle|\sigma\rangle|p_x\rangle|p_y+1\rangle|\sigma\rangle)$$

$$\left. + \sqrt{\frac{t}{2\lambda}}\sum_{\sigma\in\{\downarrow,\uparrow\}}(|0\rangle_U|0\rangle_V|p_x\rangle|p_y\rangle|\sigma\rangle|p_x-1\rangle|p_y\rangle|\sigma\rangle + |0\rangle_U|0\rangle_V|p_x\rangle|p_y\rangle|\sigma\rangle|p_x\rangle|p_y-1\rangle|\sigma\rangle) \right), \quad (59)$$

where the first line above corresponds to $-Z$ and $ZZ$ terms, the second line corresponds to $-X\vec{Z}X$ terms, and the final line corresponds to the $-Y\vec{Z}Y$ terms. Note that we are looking at a Hubbard model with periodic

boundary conditions, so wherever something like $|p_x + 1\rangle$ appears, we really mean $|(p_x + 1) \bmod M\rangle$, which we omitted from the above equation for clarity. Our implementation of PREPARE begins for the Hubbard model by
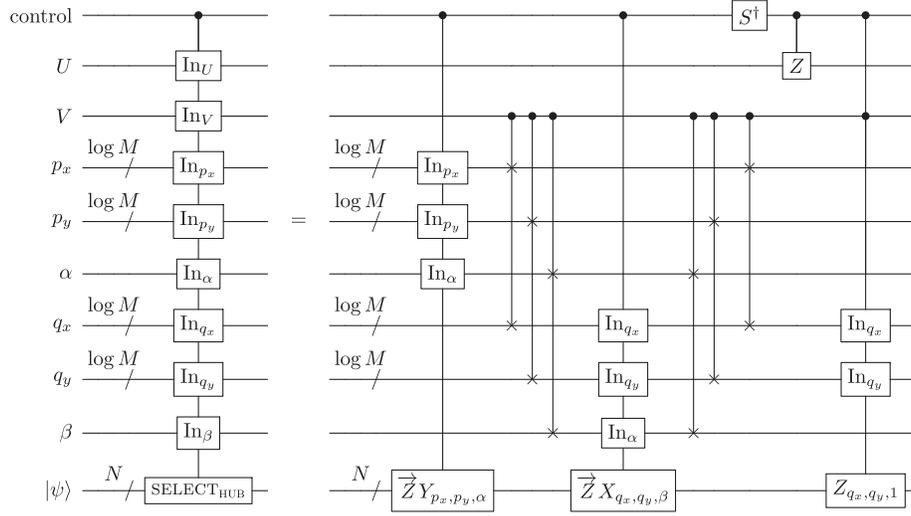
FIG. 19. A SELECT circuit for the Hubbard model, with function determined by how $p$ relates to $q$. Recall from Eq. (45) that $M = \sqrt{N/2}$ for the Hubbard model. This circuit has a T count of $10N + \mathcal{O}(\log N)$ and spans $N + 3 \log N + \mathcal{O}(1)$ qubits. If control is OFF, the circuit has no effect. Otherwise, if $|U\rangle|V\rangle = |1\rangle|0\rangle$, it will be the case that $(p, \alpha) = (q, \beta)$, and our circuit applies $-Z_{p,\alpha}$. If $|U\rangle|V\rangle = |0\rangle|1\rangle$, we again have that $p = q$, but this time, we also have that $\alpha = 0$ and $\beta = 1$, so the circuit applies $Z_{p,0}Z_{q,1}$. If $|U\rangle|V\rangle = |0\rangle|0\rangle$ and $p < q$, the circuit performs $-X_{p,\alpha}\vec{Z}X_{q,\alpha}$, but if $p > q$, the circuit performs $-Y_{p,\alpha}\vec{Z}Y_{q,\alpha}$. The larger gates in this circuit are Majorana operators described in Fig. 9 and an indexed operation explained in Fig. 7 (except that the $X_\ell$ gate is replaced by a $Z_\ell$ gate). The Majorana operators each have a T count of $4N$, but the indexed operation has no dependence on $\alpha$ and so has a T count of $2N$. All other circuit components have T counts in $\mathcal{O}(\log N)$.

initializing a two-qubit state containing the three distinct coefficients for the $U$, $V$, and $T$ terms. This is done with standard circuit synthesis techniques with a T count of $\mathcal{O}(\log(1/\epsilon))$ [69]. We then spread these coefficients over the various cases. We depict our implementation in Fig. 20.

### C. Hubbard model resources

For the case of the planar Hubbard model in Eq. (56), it is readily apparent that
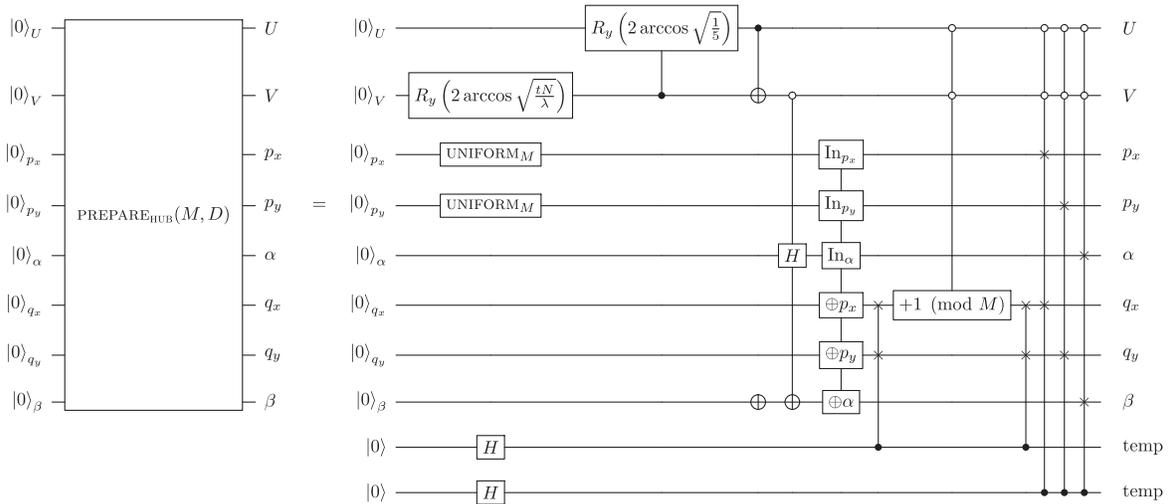
$$\lambda = 2Nt + \frac{Nu}{2} \in \mathcal{O}(N), \qquad (60)$$



FIG. 20. PREPARE$_{\text{HUB}}$ circuit with T count $\mathcal{O}(\log(N/\epsilon))$. The $R_y$ operations are used to prepare the three distinct LCU coefficients, which, including multiplicity, are $\sqrt{tN/\lambda}$ (for the $2N$ T-type terms), $\sqrt{Nu/(4\lambda)}$ (for the $N = 2M^2$ U-type terms), and $\sqrt{Nu/(8\lambda)}$ (for the $N/2 = M^2$ V-type terms). We only specify the action of this circuit in the case where the inputs are all $|0\rangle$. During the actual execution, the effects of the SELECT operation will prevent PREPARE$^\dagger$ from exactly uncomputing the $U$, $V$, $p_x$, $p_y$, and $\alpha$ qubits as well as the bottom two ancilla qubits. This is the expected behavior, and it is required that the potentially not-uncomputed qubits be kept and used as inputs for the next PREPARE$_{\text{HUB}}$ circuit.

assuming that we are dealing with the spinful model with periodic boundary conditions. We also determine that $P \in \mathcal{O}(\log(N/\epsilon))$ and that $S = 10N + \mathcal{O}(\log N)$. Thus, the total T cost of the Hubbard algorithm is

$$\frac{\sqrt{2}\pi\lambda(S + 2P)}{\Delta E} = \frac{\sqrt{2}\pi(2t + u/2)N(S + 2P)}{\Delta E}$$
$$\approx \frac{20\sqrt{2}\pi t + 5\sqrt{2}\pi u}{\Delta E}N^2. \qquad (61)$$

Ancillae required for our Hubbard model simulation come from two sources: qubits required for our entanglement-based phase estimation [see Eq. (24)] and ancillae actually required for our implementation of PREPARE and SELECT, which for the Hubbard model is $3 \log N + \mathcal{O}(1)$. Putting these sources together, the total ancillae required are

$$\log\left(\frac{\sqrt{2}\pi\lambda}{2\Delta E}\right) + 3\log N + \mathcal{O}(1) = \log\left(\frac{\sqrt{2}\pi\lambda N^3}{2\Delta E}\right) + \mathcal{O}(1), \qquad (62)$$

where the additive constant is small and can usually be neglected for problem sizes of interest. This expression gives the ancillae count in Theorem 2.

While numerically exact solutions to the Hubbard model are available for one-dimensional [98] and infinite-dimensional systems [99], no known polynomial-time scaling classical methods can provide reliable solutions to the planar model in all parts of its phase diagram [97]. For state-of-the-art approximate methods, the most challenging low-temperature phase of the model appears to be the intermediate-interaction regime due to the presence of many competing phases, around $u/t = 4$ [97]. Accordingly, we focus our analysis on this regime. If $u = 4t$, then $\lambda = 4Nt$. An interesting and classically challenging-to-obtain accuracy (beyond the agreement of state-of-the-art numerical methods [97]) for this regime would be in the vicinity of $\Delta E \approx t/100$ [100]; these choices would suggest a T complexity of approximately

$$\frac{40\sqrt{2}\pi t}{t/100}N^2 < (1.8 \times 10^4)N^2 \qquad (63)$$

and an ancilla count of approximately

$$\log\left(\frac{8\sqrt{2}\pi t N^3}{t/100}\right) < 12 + 3\log N. \qquad (64)$$

We summarize these resources for various interesting sizes of Hubbard model simulation in Table IV.

TABLE IV. Resources required for quantum simulation of a planar Hubbard model with periodic boundary conditions and spin, as in Eq. (56). The dimension of the system indicates how many sites (spatial orbitals) are on each side of the square model. The number of system qubits is thus twice the number of spatial orbitals. The number of logical ancillae is computed as Eq. (64). Finally, the number of T gates is computed using Eq. (63), which assumes that $u/t = 4$ and $\Delta E = t/100$. The first three problem sizes in the table are near the classically intractable regime.

| Dimension | Spin orbitals | Logical ancilla | Total logical | T count |
|---|---|---|---|---|
| $6 \times 6$ | 72 | 33 | 105 | $9.3 \times 10^7$ |
| $8 \times 8$ | 128 | 33 | 161 | $2.9 \times 10^8$ |
| $10 \times 10$ | 200 | 36 | 236 | $7.1 \times 10^8$ |
| $20 \times 20$ | 800 | 42 | 842 | $1.2 \times 10^{10}$ |

## D. Exploiting locality in simulations of lattice Hamiltonians

Looking forward, another way that our circuits can be applied is to accelerate the recent Lieb-Robinson simulation method of Ref. [49]. Lieb-Robinson bounds reveal an intriguing fact about local Hamiltonians: Interactions spread out in a light cone similar in form to the causal diamonds used in relativity to indicate the regions of space-time that can have an impact on an event at a point in spacetime [101]. More specifically, Lieb-Robinson bounds show that information propagates at finite speeds (up to exponentially small errors) in systems with nearest-neighbor interactions. The idea behind Ref. [49] is to exploit this structure to break up the evolution into subpieces that can be independently simulated, thus reducing the cost of simulation.

We formalize this by envisioning that we have a lattice of $N$ sites, $\Lambda$, and a Hamiltonian that consists of terms that act upon these sites, $H = \sum_{X \subseteq \Lambda} h_X$. Here, each $h_X$ is local in that if $h_X$ and $h_Y$ act on different sites in the lattice, then $[h_X, h_Y] = 0$, and $h_X$ only has support on sites that are a constant Euclidean distance away from each other. Note that this definition of locality also incorporates the terms within the Hubbard model. The final concept that we need in order to explain the method is that of distance between sites. We assume that for all $X, Y \subseteq \Lambda$, $\text{dist}(X, Y)$ yields the minimum Euclidian distance between any two points within the lattice vectors contained within sets $X$ and $Y$. For example, given a lattice 1D on 10 sites, $\text{dist}(\{3, 4, 5\}, \{8, 9, 10\}) = 3$. The following lemma (a restatement of Lemma 6 in Ref. [49]) explains the impact that the locality imposed by the Lieb-Robinson bound has on simulation.

**Lemma 3** (patching lemma). Let $\Lambda$ be a lattice on $N$ sites with a Hamiltonian $H = \sum_{X \subseteq \Lambda} h_X$, where each $h_X$ is a local bounded Hamiltonian for every $X \subseteq \Lambda$. Let $A, B, C$ be subsets of $\Lambda$, and let $H_{P_1 \ldots P_q}$ for any sequence

$P:\{1, ..., q\} \mapsto \{A, B, C\}^q$ be for integer $q \geq 1$ $H_{P_1...P_q} = \sum_{X \in P_1 \bigcup ... \bigcup P_q} h_X$ (for example, $H_{AB} = \sum_{X \subseteq A \bigcup B} h_X$). There are constants $v \geq 0$, called the Lieb-Robinson velocity, and $\mu > 0$ such that

$$\|e^{-iH_{ABC}t} - e^{-iH_{AB}t}e^{iH_Bt}e^{-iH_{BC}t}\|$$

$$\in \mathcal{O}\left(\sum_{X \subseteq (A \bigcup B \bigcup C) \setminus (A \bigcup B) \setminus C} \|h_X\| e^{vt - \mu \, \text{dist}(A,C)}\right). \quad (65)$$

Note that, in the above terminology, $(A \cup B \cup C) \setminus (A \cup B) \setminus C$ is the boundary of the sets $AB$ and $C$, meaning the set of all terms within the Hamiltonian that act on sites contained in both $A$ or $B$ and $C$. Lemma 3 is the core of the simulation method. The central idea behind the proof is to use the patching lemma recursively to break up the evolution into a product of evolution operators, each of which contains terms that act on one or two of the constituent subsets of sites in the problem. This is conceptually similar to a Trotter decomposition; however, as the error in this approximation can be made exponentially small by choosing the patches in Lemma 3 to be linearly far apart, the error can be controlled in a tighter fashion without requiring short time steps (unlike Trotter decompositions [22,102]). For example, consider regions $A$, $B$, $C$, $D$. Then, we can write

$$e^{-iH_{ABCD}t} \approx e^{-iH_{AB}t}e^{iH_Bt}e^{-iH_{BCD}t}$$

$$\approx e^{-iH_{AB}t}e^{iH_Bt}e^{-iH_{BC}t}e^{iH_Ct}e^{-iH_{CD}t}. \quad (66)$$

In order to achieve scaling that is polylogarithmic in $1/\epsilon$, the evolution of each patch needs to be simulated using a method with polylogarithmic scaling in $1/\epsilon$, such as the truncated Taylor series simulation result [23] or qubitization [49]. Our circuits can be used to optimize this result since qubitization remains the best way to simulate the evolution, and our SELECT and PREPARE circuits meet the requirements of qubitization oracles. This result is formally given as Theorem 1 of Ref. [49], a special case of which is restated below for convenience.

**Theorem 4.** Assume the preconditions of Lemma 3 and that, for every unit ball in $\Lambda$ within the Euclidean metric space $\mathbb{R}^D$, at most $\mathcal{O}(1)$ sites are contained within the ball and $h_X = 0$ if the diameter of the set $X$ is greater than 1. Additionally, let each $h_X$ be efficiently computable and have norm at most 1. Then, there exists a quantum algorithm that simulates the evolution of $H$ for time $\tau$ with accuracy $\epsilon$ that uses $\mathcal{O}(\tau N \text{polylog}(\tau N/\epsilon))$ 2-qubit gates and further has gate depth $\mathcal{O}(\tau \text{polylog}(\tau N/\epsilon))$.

We claim that our approach can be used to achieve $\tilde{\mathcal{O}}(N/\epsilon)$ scaling for simulating the Hubbard model with nearest-neighbor interactions. The Hubbard model satisfies the preconditions because each term in the Hamiltonian is local on the fermion lattice [49]. By using our constructions

for the PREPARE and SELECT circuits, we can reduce constant factors (and some log factors in T complexity) involved in the qubitized simulation while saturating the $\tilde{\mathcal{O}}(\tau N)$ scaling of Theorem 4. We then choose $\tau \in \mathcal{O}(1/\lambda)$ and apply phase estimation on the result. In order to estimate the eigenvalue to within error $\epsilon$ with high probability, we need $\mathcal{O}(\lambda/\epsilon)$ repetitions of the circuit. Thus, by multiplying the two results, we find that the overall scaling for simulating such a Hubbard model is $\tilde{\mathcal{O}}(N/\epsilon)$, as claimed.

This approach requires some follow-up work in order to determine exact T counts. Specifically, we need to implement a full qubitized simulation (rather than $e^{-i \arccos(H/\lambda)}$). This transformation is known to be achievable with a polylogarithmic-sized circuit [25]. While our work provides a highly optimized method for implementing the oracles needed in this process, more work remains to estimate constant factors associated with this simulation.

## VI. RESOURCE ANALYSIS FOR FAULT-TOLERANT IMPLEMENTATION

Throughout this work, we have focused on the number of T gates as the primary cost model of interest. The reasons for this are our focus on hardware consisting of a 2D nearest-neighbor coupled array of qubits, the intention to use the surface code [31–35], and the high relative overhead of T gates compared to all others in that context. In this section, we discuss the overhead of the complete algorithm in detail.

When using the surface code, each T gate is implemented by first preparing a magic state

$$|\text{T}\rangle \equiv \text{T}|+\rangle = \frac{|0\rangle + e^{i\pi/4}|1\rangle}{\sqrt{2}} \quad (67)$$

that is consumed during the gate. The gate is probabilistic, and 50% of the time, $\text{T}^\dagger$ is actually applied instead of T. When the gate implemented is not as desired, an S gate must be inserted to correct it. Preparing T states requires a substantial amount of time and hardware, which we describe below. In an effort to minimize the number of physical qubits required, we therefore only prepare a

TABLE V. Breakdown of the various elements that make up the Majorana operator circuit from Fig. 9 and the data lookup circuit from Fig. 10. Here, $N$ is the number of spin orbitals in the system that the circuits are being applied to.

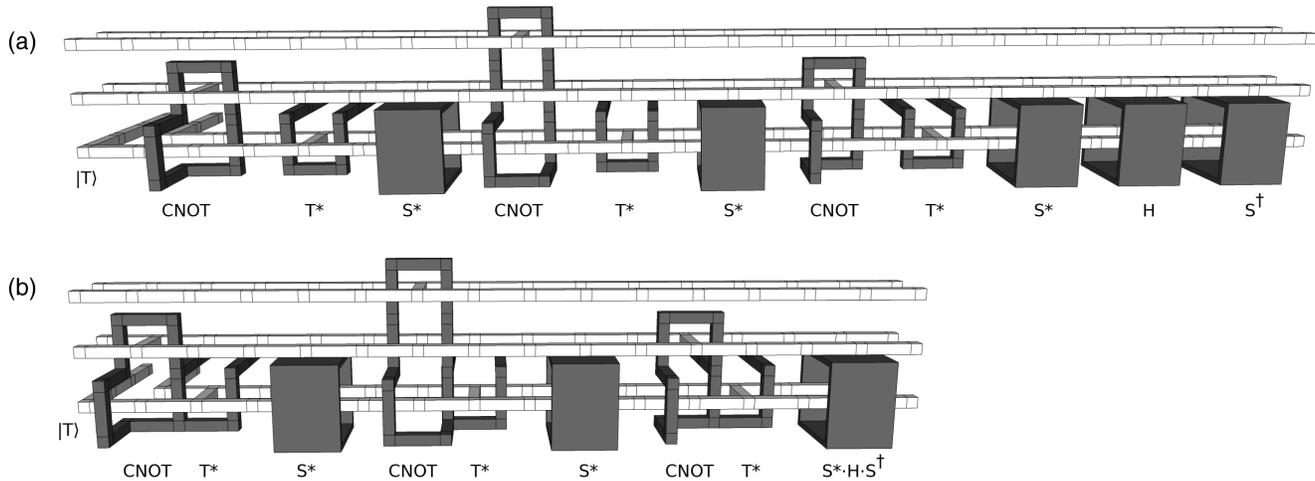| | Compute ANDS | Uncompute ANDS | Naked CNOTS | Subcircuits |
|---|---|---|---|---|
| Fig. 9 | $N - 1$ | $N - 1$ | $0.5N$ | $0.5N$ |
| Fig. 10 | $1.5N - 1$ | $1.5N - 1$ | $0.75N$ | $0.75N$ |

(a)



(b)



FIG. 21. (a) Canonical surface code AND gate computation (Fig. 4) circuit. The bottom pair of white lines (coming from the back instead of from the left) represents the injection of a $|T\rangle$ state. Each dark ring is a CNOT. Each dark U-shape labeled T* is a random T or $T^\dagger$ gate (as determined by a measurement in a location not shown in the diagram connecting to the dark U-shape). The boxes labeled S* immediately following each T* are S or $S^\dagger$ gates that are included if the random T or $T^\dagger$ gate results in the incorrect gate. The final two boxes are Hadamard and $S^\dagger$ operations, respectively. (b) Compressed version. The final three boxes can be compressed to a single box, as an arbitrary single-qubit Clifford can be performed inside using twists [104] and other techniques [105]. Distinct dark structures can be made to touch, provided this occurs in a single place [36].

single T state at a time. We assume the availability of a correlated-error minimum-weight perfect matching decoder [103] capable of keeping pace with 1 $\mu$s rounds of surface code error detection, and capable of delivering feedforward in 10–20 $\mu$s. We calculate the qubit and time overhead for physical gate error rates $p = 10^{-3}$ and $p = 10^{-4}$.

The overhead is approximated by considering only the overhead of the Majorana operator circuit from Fig. 9 and the data lookup circuit from Fig. 10. It is expected that these circuits will account for over 90% of the total algorithm overhead. These circuits break down into a number of common pieces: compute ANDs, uncompute ANDs, naked CNOTs, and active subcircuits. The number of these pieces, in terms of the number of algorithm target qubits, $N$, is shown in Table V.
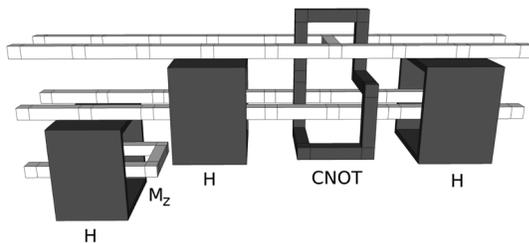


FIG. 22. Uncompute AND (Fig. 4) implemented directly as a Hadamard operation followed by a measurement on the bottom qubit. The outcome of this measurement determines if the sequence of operations on the top two qubits (a CZ implemented as a CNOT framed by Hadamard operations) are included or omitted.

A surface code implementation of compute AND in Fig. 4 is shown in Fig. 21. The regular geometric structure can be decomposed into plumbing pieces, namely, cubic volumes each containing a single, small, light-colored cube. The compressed [36] version has a depth 15 plumbing pieces. The circumference of each stringlike structure (defect) is the surface code distance $d$, and the minimum separation of defects of the same color is also $d$. In the temporal direction (left-right), each unit of $d$ is a round of error detection. In the spatial directions (plane perpendicular to temporal), each unit of $d$ corresponds to two qubits. Note that a single CNOT, after compression, takes depth 1 plumbing piece as drawn. The overhead of any algorithm ultimately needs to be expressed as some number of qubits (space) and seconds (time). A plumbing piece is a convenient device-physics and code-distance independent measure of space-time volume. As described above, the $(5d/4)^3$ cubic volume of a plumbing piece can easily be converted to qubits and seconds, given a code distance $d$ and single-round error-detection time.

A plumbing piece depth 8 surface code implementation of uncompute AND in Fig. 4 is shown in Fig. 22. This could be compressed by performing the measurement differently so that no initial Hadamard would be necessary; however, the current surface code form is more easily identified with the original abstract form, and further compression is not necessary, as the execution time of the algorithm, as we shall see, is limited by our serial preparation of T states.

An effective plumbing piece depth 5 surface code implementation of the Majorana operator active subcircuit is shown in Fig. 23. The unusual pair-of-horns structure is
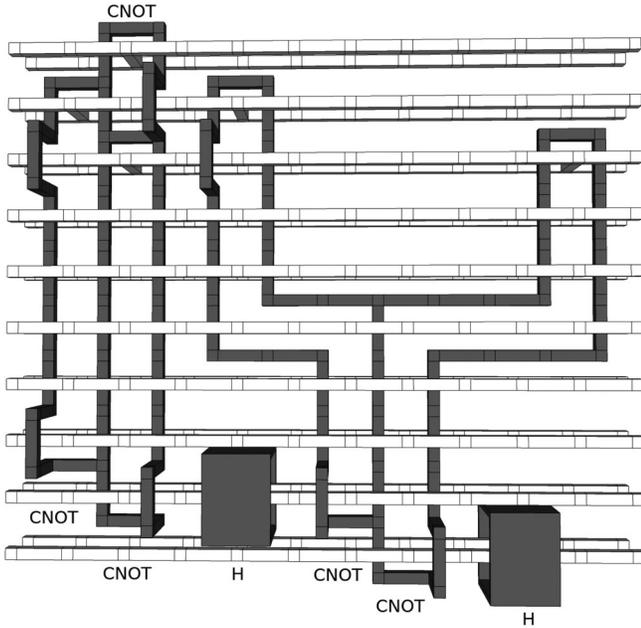
FIG. 23.   Surface code implementation of the inner loop of the Majorana operator circuit. Contrast with Fig. 9, noting that the circuit has been somewhat modified to reduce its surface code spacetime volume. In particular, the controlled-$Y$ operations from Fig. 9 have been propagated through the controlled-$Z$ operations, producing CNOT operations that are cheaper to perform. This creates phase error, which must be corrected by an S gate on the control of the entire Majorana operator. Here, we do not show initial Hadamard gates on every target qubit.

to permit an uncompute AND circuit to fit in before the final CNOT. The inner loop of the data circuit (Fig. 10) is just a pair of single-control multiple-target CNOTs, and a single

additional CNOT. This case can be implemented in plumbing piece depth 4 and is not shown.

Preparing a T state is an involved process [106,107], which is shown for discussion purposes in Fig. 24. The important features for our purposes are the fact that this can be tiled vertically (meaning in time) every six plumbing pieces, and the whole structure occupies an area of 160 plumbing pieces. A significant amount of fast classical feedforward is required, as many T gates are potentially followed by S gates, and the paths of the connections from the first (small) level of distillation to the second (large) level must be determined based on which succeed. Our assumption of a 10–20-$\mu$s latency decoder is sufficient to make this work. We are interested in the overhead of solving instances of the electronic structure and Hubbard Hamiltonians discussed in prior sections. We must choose a target inaccuracy $\epsilon$ to fix the number of data logical qubits and gates required. To first order, the dependence of the gate count on $\epsilon$ can be ignored. We choose $\epsilon = 10^{-3}$. Table VI summarizes the circuit input parameters we will study.

Given Tables V and VI, and the plumbing piece depths of the various circuit elements, we can calculate the total number of data plumbing pieces $N_{\mathrm{PP}}^{\mathrm{data}}$ and hence the code distance required to ensure no more than a 1% chance of logical error in any data plumbing piece using $p_L(d, p) \simeq 2d(50p)^{(d+1)/2} < 1/(100 N_{\mathrm{PP}}^{\mathrm{data}})$. Similarly, knowing that the compute AND circuit contains 4 T gates and that no other part of the data or Majorana operator circuits contains T gates, we can calculate the total number of T gates, $N_{\mathrm{T}}$, and hence the target T-state error rate from distillation of $1/(100 N_{\mathrm{T}})$. We also calculate the total number of T



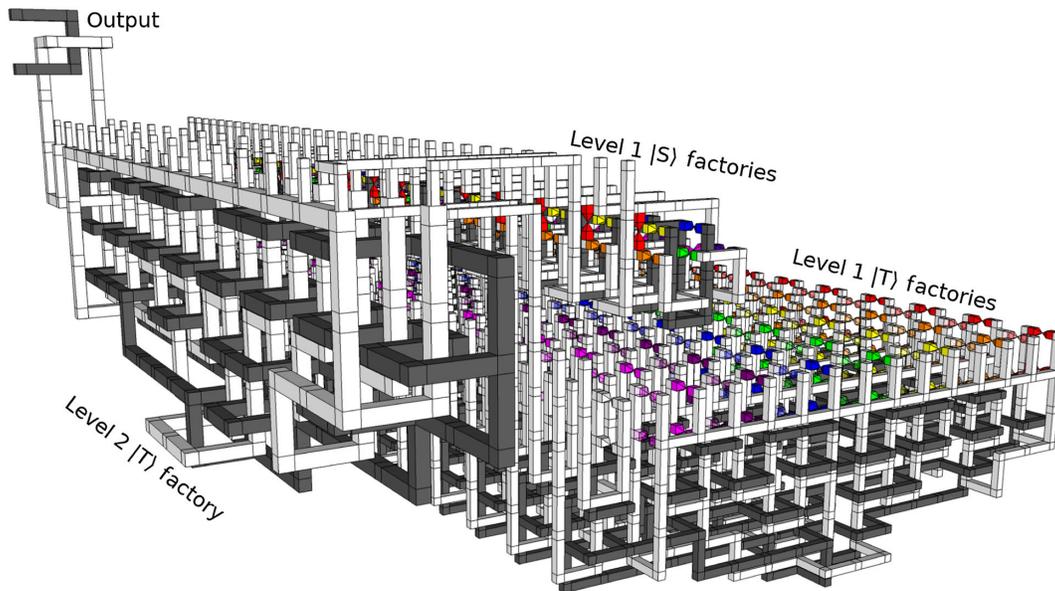FIG. 24.   Preparing a T state in the surface code. Physical $|T\rangle_0$ states are injected into the 16 lower factories in the lower rear right, which distill them into less noisy $|T\rangle_1$ states. Fifteen of the successful distillations are forwarded to the larger factory towards the front and left, which distills them into a $|T\rangle_2$ state with low enough error. The output is shown as the top-left dark U-shape.

TABLE VI. Cases for which we will calculate the fault-tolerant overhead, along with numbers relevant to estimating the non-negligible components of this overhead. Column 3 contains the number of times that the $\mathcal{W}$ oracle is queried. For the Hubbard model, we consider the system at intermediate coupling ($u/t = 4$), implying that $\lambda = 4Nt$, and we consider an accuracy of $\Delta E = t/100$. For jellium, values of $\lambda$ are provided in Table III, and we target chemical accuracy, which is defined as $\Delta E = 0.0016$ Hartree. Each call to $\mathcal{W}$ includes a call to SELECT, PREPARE, and PREPARE$^\dagger$, which in turn apply QROM and Majorana operations (which are the dominant costs of the algorithm). Columns 4 and 5 estimate the number of times the Majorana operator circuit (see Fig. 9) must be applied to the entire system or else to half of the system. For electronic structure, there are three Majorana operators of size $N$ per query to $\mathcal{W}$ (see Fig. 14). For Hubbard, there are two Majorana operators of size $N$ and one of size $N/2$ per query to $\mathcal{W}$ (see Fig. 19). Column 6 estimates the number of times QROM lookups (see Fig. 10) of size $L = 3N/2$ are performed. This does not occur in our Hubbard model circuits, but it happens twice per query to $\mathcal{W}$ in our electronic structure circuits (once in PREPARE and once in PREPARE$^\dagger$; see Fig. 15). The final column contains the maximum number of data qubits required at any point in the algorithm, which occurs while applying the Majorana operations in SELECT. This number does not include space to prepare T states, which will be discussed separately.

| System | Spin orbitals ($N$) | $\mathcal{W}$ queries | Majorana$_N$ | Majorana$_{N/2}$ | QROM$_{3N/2}$ | Max qubits |
|---|---|---|---|---|---|---|
| Hubbard model | 72 | $1.3 \times 10^5$ | $2.5 \times 10^5$ | $1.3 \times 10^5$ | 0 | 105 |
| Hubbard model | 128 | $2.3 \times 10^5$ | $4.6 \times 10^5$ | $2.3 \times 10^5$ | 0 | 161 |
| Hubbard model | 200 | $3.6 \times 10^5$ | $7.2 \times 10^5$ | $3.6 \times 10^5$ | 0 | 236 |
| Hubbard model | 800 | $1.4 \times 10^6$ | $2.8 \times 10^6$ | $1.4 \times 10^6$ | 0 | 842 |
| Electronic structure | 54 | $1.4 \times 10^4$ | $4.2 \times 10^4$ | 0 | $2.8 \times 10^4$ | 123 |
| Electronic structure | 128 | $6.3 \times 10^4$ | $1.9 \times 10^5$ | 0 | $1.3 \times 10^5$ | 210 |
| Electronic structure | 250 | $1.7 \times 10^5$ | $5.3 \times 10^5$ | 0 | $3.5 \times 10^5$ | 341 |
| Electronic structure | 1024 | $1.8 \times 10^6$ | $5.3 \times 10^6$ | 0 | $3.5 \times 10^6$ | 1136 |

distillation plumbing pieces, $N_{\text{PP}}^{\text{T}}$, and a code distance to ensure that the chance of T plumbing piece error is below $1/(100N_{\text{PP}}^{\text{T}})$. We have elected to keep algorithm error rates low to ensure that, on average, only a few repetitions are required. For both $p = 10^{-3}$ and $p = 10^{-4}$ and all algorithm instances considered, T-state distillation of the form in Fig. 24 is sufficient to achieve the target logical error rate. This information is collectively sufficient to calculate the qubit and time overheads, shown for all cases in Table VII.

The previous paragraphs described a manual overhead estimation method. There are a number of approximations that go into such an estimate, in particular, assuming that it will always be possible to route gates in 3D spacetime without overhead beyond that of where the data qubits are stored. In order to strengthen the relevance of the presented results, we have also used a software-automated approximation method. The software is an improved version of the tool from Ref. [108]. Automated overhead approximation starts from a Clifford + T representation of the circuit to be analyzed (e.g., Fig. 8) and ends with a full surface code layout, having each gate translated into a corresponding configuration of plumbing pieces. Thus, the automated estimation work flow is similar to the manual one. However, certain circuit particularities are analyzed differently, so similarities and differences between the two methods are discussed.

The Clifford + T circuit is prepared according to a worst-case scenario, based on the available hardware restrictions, plumbing-piece layout problems, and T-gate correction mechanisms. The preparation of a single distilled T state at a time is a restriction that influences the resulting surface code layout: The Clifford + T gates have to be scheduled (laid out) in such a way that the T gates will

TABLE VII. Manual calculation of qubit and time overheads of general chemistry and Hubbard circuits, assuming gate error rates of $p = 10^{-3}$ and $p = 10^{-4}$, a 2D array of nearest-neighbor coupled qubits, and a surface code error-detection cycle time of 1 $\mu$s. The execution time being estimated is the duration of one complete run of the phase estimation process.

| Problem | | Physical qubits | | Execution time (h) | |
|---|---|---|---|---|---|
| System | Spin orbitals ($N$) | $p = 10^{-3}$ | $p = 10^{-4}$ | $p = 10^{-3}$ | $p = 10^{-4}$ |
| Hubbard model | 72 | $1.4 \times 10^6$ | $4.4 \times 10^5$ | 4.6 | 2.6 |
| Hubbard model | 128 | $2.1 \times 10^6$ | $6.6 \times 10^5$ | 15 | 8.4 |
| Hubbard model | 200 | $3.2 \times 10^6$ | $8.9 \times 10^5$ | 40 | 21 |
| Hubbard model | 800 | $1.4 \times 10^7$ | $3.6 \times 10^6$ | $6.7 \times 10^2$ | $3.7 \times 10^2$ |
| Electronic structure | 54 | $1.4 \times 10^6$ | $3.9 \times 10^5$ | 0.82 | 0.43 |
| Electronic structure | 128 | $2.4 \times 10^6$ | $8.1 \times 10^5$ | 9.9 | 5.6 |
| Electronic structure | 250 | $4.4 \times 10^6$ | $1.2 \times 10^6$ | 58 | 30 |
| Electronic structure | 1024 | $2.0 \times 10^7$ | $4.8 \times 10^6$ | $2.7 \times 10^3$ | $1.4 \times 10^3$ |
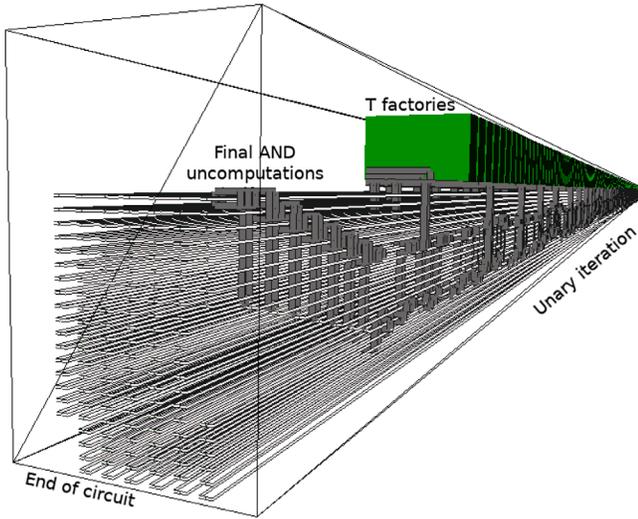
FIG. 25. Illustration of the circuit layout used for resource estimation. The figure shows a portion of a circuit towards the outputs (the time axis indicating circuit execution runs from the back to the front). The figure sketches the arrangement of the logical qubits (light gray lines), the CNOTs (dark gray lines), with the boxes (green) abstracting the multilevel distillation circuits from Fig. 24 and the bounding box (black wire frame) expressing the resources estimated to lay out the circuit.

be executed as soon as possible, but not earlier than the availability of distilled T states. The state distillation form in Fig. 24 implies that a T gate can be executed, on average, every six plumbing pieces along the time axis. Additionally, T-gate implementations are probabilistic, and S-gate corrections may be necessary. Thus, our scenario considers that all T gates are followed by the corrective S gate, resulting in a synthetic increase of the Clifford + T

circuit depth. Circuit preparation is followed by an optimization procedure, where as many Clifford gates as possible are scheduled between two subsequent T gates. The software simulates the availability of the distilled T states and places T gates whenever their execution is possible. If no T states are available, the T gates are delayed, which will later increase the approximated time overhead.

Finally, the Clifford + T circuit is translated into the surface code layout. From a resource estimation perspective, the complexity of this task is increased because the software currently only partially includes the optimization strategy presented in Fig. 21(b): Final boxes can be compressed to a single one, but distinct dark structures are not allowed to touch (for verification or debugging purposes). Due to this fact, the automated approximation uses a slightly different Clifford + T realization of the computing AND gate (cf. Fig. 4), which has the advantage of being more suitable for automatic placement in stairway-structured circuits (i.e., the arrangement of AND gates in Fig. 9). Automatic placement of those Clifford + T subcircuits results in a shorter depth of the generated surface code layouts. Overhead of the two basic circuits in units of plumbing pieces can be found in Table VIII. These data converted into qubits and time can be found in Table IX. The automatically generated estimations are comparable to the manual ones, though generally slightly higher, exceeding the manual estimates by 10%–20%. While the automated method is penalized by missing optimization strategies that are possible when analyzing circuits manually, and the need to provide explicit communication paths for long-range gates, some of this penalty is canceled by using algorithmic methods too complex to perform manually. The fact that both approximation methods lead to

TABLE VIII. Automatically generated resource estimates of the Majorana and QROM circuits (Figs. 9 and 10). The area width, height, and time columns give the dimensions of the bounding box (e.g., Fig. 25) in units of plumbing pieces. The last column is the number of plumbing pieces estimated to be actively used within the bounding box. The volume numbers do not include the volume of the T factory, but they do include idle qubits that are present in the algorithm as a whole but not the individual circuits. "Braided volume" refers to the amount of actively used volume, i.e., nonempty space with defects used to encode qubits and operations. QROM circuits are indexed like $QROM_{\frac{3}{2}N}$ because, in context, the QROM index size $L$ is always 50% larger than the number of orbitals $N$.

| System | Circuit ($N$) | T count | Area (PP$^2$) | Time (PP) | Volume (PP$^3$) | Braided volume (PP$^3$) |
|---|---|---|---|---|---|---|
| Hubbard model | Majorana$_{72}$ | 284 | $17 \times 16$ | 1840 | 500 480 | 429 624 |
| Hubbard model | Majorana$_{128}$ | 508 | $25 \times 16$ | 3252 | 1 300 800 | 1 155 817 |
| Hubbard model | Majorana$_{200}$ | 796 | $36 \times 16$ | 5080 | 2 926 080 | 2 637 504 |
| Hubbard model | Majorana$_{800}$ | 3196 | $123 \times 16$ | 20 262 | 39 875 616 | 37 451 032 |
| Electronic structure | Majorana$_{54}$ | 212 | $20 \times 16$ | 1382 | 442 240 | 365 717 |
| Electronic structure | Majorana$_{128}$ | 508 | $32 \times 16$ | 3252 | 1 665 024 | 1 473 563 |
| Electronic structure | Majorana$_{250}$ | 996 | $51 \times 16$ | 6342 | 5 175 072 | 4 685 164 |
| Electronic structure | Majorana$_{1024}$ | 4092 | $165 \times 16$ | 25 932 | 68 460 480 | 64 114 531 |
| Electronic structure | $QROM_{\frac{3}{2}54}$ | 320 | $20 \times 16$ | 2068 | 661 760 | 558 098 |
| Electronic structure | $QROM_{\frac{3}{2}128}$ | 764 | $32 \times 16$ | 4872 | 2 494 464 | 2 273 711 |
| Electronic structure | $QROM_{\frac{3}{2}250}$ | 1496 | $51 \times 16$ | 9508 | 7 758 528 | 7 272 549 |
| Electronic structure | $QROM_{\frac{3}{2}1024}$ | 6140 | $165 \times 16$ | 38 892 | 102 674 880 | 100 399 903 |

TABLE IX. Automatically generated qubit and time overheads of general chemistry and Hubbard circuits assuming gate error rates of $p = 10^{-3}$ and $p = 10^{-4}$, a 2D array of nearest-neighbor coupled qubits, and a surface code error-detection cycle time of 1 $\mu$s. The execution time being estimated is the duration of one complete run of the phase estimation process.

| Problem | | Physical qubits | | Execution time (h) | |
|---|---|---|---|---|---|
| System | $N$ spin orbitals | $p = 10^{-3}$ | $p = 10^{-4}$ | $p = 10^{-3}$ | $p = 10^{-4}$ |
| Hubbard model | 72 | $1.7 \times 10^6$ | $5.3 \times 10^5$ | 4.6 | 2.6 |
| Hubbard model | 128 | $2.4 \times 10^6$ | $7.8 \times 10^5$ | 15 | 8.4 |
| Hubbard model | 200 | $3.8 \times 10^6$ | $1.0 \times 10^6$ | 40 | 21 |
| Hubbard model | 800 | $1.5 \times 10^7$ | $4.2 \times 10^6$ | $6.7 \times 10^2$ | $3.7 \times 10^2$ |
| Electronic structure | 54 | $1.7 \times 10^6$ | $4.7 \times 10^5$ | 0.85 | 0.44 |
| Electronic structure | 128 | $2.9 \times 10^6$ | $9.5 \times 10^5$ | 10 | 5.7 |
| Electronic structure | 250 | $5.1 \times 10^6$ | $1.4 \times 10^6$ | 58 | 30 |
| Electronic structure | 1024 | $2.3 \times 10^7$ | $5.6 \times 10^6$ | $2.8 \times 10^3$ | $1.4 \times 10^3$ |

such comparable qubit and time overheads strengthens our confidence in these estimates. The time estimates, in particular, are practically identical.

The data are highly encouraging, with physical qubit counts of order a million and times in hours for all but the largest cases considered. Significant further reduction is expected to be possible. For example, the $N$ qubits in the $|\psi\rangle$ register are only operated on by the Majorana operator circuit, and this circuit targets just one of these qubits at a time. This implies that the remainder can be stored more compactly in square surface code patches while not being interacted with. This method could easily reduce the overhead of these $N$ qubits, which account for 70%–80% of the physical qubits, by a factor of 6. This result would conservatively lower overall physical qubit requirements by a factor of two.

## VII. CONCLUSION

In this work, we introduced especially efficient fault-tolerant quantum circuits for using phase estimation to estimate the spectra of electronic Hamiltonians. Unlike past work, which has focused on realizing phase estimation unitaries encoding $e^{-iH\tau}$, corresponding to time evolution under $H$ for duration $\tau$, we focused on a recent idea that one might more cheaply realize phase estimation unitaries encoding the quantum walk $e^{i \arccos(H/\lambda)}$, where $\lambda$ is a parameter closely related to the induced 1-norm of the system Hamiltonian [26,27]. We construct explicit quantum circuits for realizing this quantum walk with T complexity linear in basis size for both the planar Hubbard model and electronic structure Hamiltonians in second quantization. We showed that phase estimation projects these systems to an eigenstate and estimates the associated eigenvalue to within additive error $\epsilon$ by querying the quantum walk operator an optimal number of times, scaling as $\mathcal{O}(\lambda/\epsilon)$. To accomplish this result, we introduced general techniques that we conjecture are near optimal for streaming bits of a unary register and for implementing a quantum read-only memory. We introduced a new form of

Heisenberg-limited phase estimation specialized to linear combinations of unitaries based simulations and provided bounds on T complexity and ancilla count, which remain tight even at small finite sizes.

In addition to providing explicit Clifford + T circuits, we compiled the bottleneck components of these simulations to fault-tolerant surface code gates in order to rigorously determine the resources that would be required for error correcting interesting problems. We performed this compilation both by hand and by using automatic tools and found similar overheads in both cases. We found that classically intractable instances of jellium and the Fermi-Hubbard model could be simulated with under $1 \times 10^6$ T gates and would require about $1 \times 10^6$ physical qubits in the surface code, with two-qubit error rates on the order of $10^{-3}$. At error rates of $10^{-4}$, about an order of magnitude fewer physical qubits would be required. We also priced out simulations of realistic solid-state materials such as diamond, graphite, silicon, metallic lithium, and crystalline lithium hydride and found that only slightly more than $1 \times 10^9$ T gates and a few million physical qubits would be required.

Despite focusing on different systems, our results are most readily comparable to the previous state-of-the-art results from Ref. [43]. Even though Ref. [43] sought empirical estimates of the T complexity rather than rigorous upper bounds as we did, they estimated that approximately $10^{15}$–$10^{16}$ T gates would be required for a 108-qubit simulation of the FeMoco molecule active space. By comparison, our upper bounds on the T complexity required to solve the classically intractable electronic structure problems studied here were roughly a million times less. The low T complexity is the result of designing a lean algorithm from the ground up, with insights matched to the Hamiltonian and with innovative algorithmic subroutines. The improvements are distributed across several parts of our approach, each of which provides 1 or 2 orders of magnitude improvement. Because our simulations require only a few times more physical qubits than is

required by a single T factory, it is reasonable to expect that the simulations we outline here will become practical on the first universal fault-tolerant quantum devices, many years before the simulations discussed in Ref. [43] would be viable.

Several important directions for future research pertain to the extension of these simulation techniques to representations that would be more effective for single molecules. While the dual basis described in Ref. [39] is well suited to treating solid-state materials such as the ones explored here (e.g., jellium, solid-state silicon, graphite, diamond, lithium and lithium hydride), by combining our techniques with the "Gausslet" basis sets of Ref. [41], we should also be able to simulate single molecules with similar resolution to Gaussian orbitals—thus extending our results to systems such as FeMoco with similar overheads to those observed in this work. However, deploying the Gausslet basis functions to systems with large atomic nuclei such as iron (as in FeMoco) will require further research. If basis errors are a concern, then future work should combine results from this paper with Refs. [109] and [26] in order to determine the cost of encoding first-quantized electronic spectra in quantum circuits; in first quantization, basis errors are suppressed exponentially in the number of qubits used to represent the system.

Another remaining challenge is to compute a tighter upper bound on the number of physical qubits required by the algorithm. At the first moment that it becomes technologically possible to distill magic states, the number of physical qubits available on one machine will still be extremely limited. Getting a meaningful computation to fit at all will be difficult. Fortunately, the qubit count estimates of this paper were fairly conservative: We only explored surface code constructions that we were able to validate and implement in software; these constructions are not necessarily optimal. For example, the logical qubit representation used in lattice surgery [110] requires fewer physical qubits than the double-defect representation used in the estimates of this paper. Furthermore, there are several places in our circuits where we preferred small multiplicative improvements in T count over small additive improvements in logical qubit count. For example, we delay uncomputing QROM lookups in order to avoid recomputation, and when performing phase estimation, we minimize the number of oracle queries by using a full-size phase register instead of a single phase qubit. Since we have managed to show that with error rates of $10^{-3}$ one can solve interesting problems in chemistry using on the order of a million physical qubits within the surface code, a next natural goal would be to try to further reduce the resources required to be on the order of a 100 000 physical qubits.

## ACKNOWLEDGMENTS

## APPENDIX: PROPAGATING ERRORS FROM HAMILTONIAN COEFFICIENTS INTO PHASE ESTIMATE

In this appendix, we address the question of how accurately coefficients of the Hamiltonian must be prepared in the PREPARE oracle in order to estimate the Hamiltonian eigenvalues to precision $\epsilon$. As discussed in Sec. II A, our phase estimation scheme involves estimating the phases induced by the operator $e^{i \arccos(H/\lambda)}$. If one is near the singularity of arccos, then a small error in the Hamiltonian can have a significant impact on the phase. Let us define

$$\tilde{H} \equiv \sum_{\ell=0}^{L-1} \tilde{w}_\ell H_\ell \tag{A1}$$

for our approximate encoding of $H$. Using the state preparation technique in Sec. III D, we obtain

$$\lambda = \sum_{\ell=0}^{L-1} \tilde{w}_\ell. \tag{A2}$$

We denote by $\delta$ an upper bound on the approximation in any of the $\tilde{w}_\ell$, so

$$\delta \geq |\tilde{w}_\ell - w_\ell|. \tag{A3}$$

Next, note that the error in the eigenphase obeys

$$\begin{aligned}
\epsilon_{\text{PREP}} &\leq \|e^{i \arccos(H/\lambda)} - e^{i \arccos(\tilde{H}/\lambda)}\| \\
&\leq \|\arccos(H/\lambda) - \arccos(\tilde{H}/\lambda)\| \\
&\leq \sum_{p=0}^{\infty} \frac{(2p-1)!!}{\lambda^{2p+1}(2p+1)(2p)!!}\|H^{2p+1} - \tilde{H}^{2p+1}\|, \quad \text{(A4)}
\end{aligned}$$

where !! is the double factorial $z!! = z \cdot (z-2) \cdot (z-4)\ldots 1$ assuming $z$ is a natural number. It is straightforward to show inductively that for any $p > 0$,

$$\|H^{2p+1} - \tilde{H}^{2p+1}\| \leq (2p+1)(\max\{\|H\|, \|\tilde{H}\|\})^{2p}\|H - \tilde{H}\|. \tag{A5}$$

We then have from Eq. (A1) that

$$\|H - \tilde{H}\| \le \sum_{\ell=0}^{L-1} |w_\ell - \tilde{w}_\ell| \le L\delta. \qquad (A6)$$

We further have that

$$\max\{\|H\|, \|\tilde{H}\|\} \le \|H\| + L\delta. \qquad (A7)$$

Substituting these equations into Eq. (A4) then gives

$$\epsilon_{\text{PREP}} \le \sum_{p=0}^{\infty} \frac{(2p-1)!!}{\lambda^{2p+1}(2p)!!}(\|H\| + L\delta)^{2p}L\delta$$

$$= \frac{L\delta}{\lambda} \sum_{p=0}^{\infty} \frac{(2p-1)!!}{(2p)!!}\left(\frac{\|H\| + L\delta}{\lambda}\right)^{2p}$$

$$= \frac{L\delta}{\lambda}\left[1 - \left(\frac{\|H\| + L\delta}{\lambda}\right)^2\right]^{-1/2}. \qquad (A8)$$

This inequality can be solved for $\delta$ to give

$$\delta \ge \frac{\epsilon_{\text{PREP}}}{(1+\epsilon_{\text{PREP}}^2)L}\left(\sqrt{\lambda^2(1+\epsilon_{\text{PREP}}^2) - \|H\|^2} - \epsilon_{\text{PREP}}\|H\|\right)$$

$$\ge \frac{\epsilon_{\text{PREP}}\lambda}{(1+\epsilon_{\text{PREP}}^2)L}(1 - \|H\|^2/\lambda^2). \qquad (A9)$$

If we require that $\epsilon_{\text{PREP}} \le \sqrt{2}\Delta E/(4\lambda)$ as in Eq. (25), then this can be obtained by choosing

$$\delta = \frac{\sqrt{2}\Delta E}{4L(1 + \frac{\Delta E^2}{8\lambda^2})}(1 - \|H\|^2/\lambda^2). \qquad (A10)$$

[1] R. P. Feynman, *Simulating Physics with Computers*, Int. J. Theor. Phys. **21**, 467 (1982).

[2] R. P. Feynman, *Quantum Mechanical Computers*, Found. Phys. **16**, 507 (1986).

[3] S. Lloyd, *Universal Quantum Simulators*, Science **273**, 1073 (1996).

[4] D. S. Abrams and S. Lloyd, *Simulation of Many-Body Fermi Systems on a Universal Quantum Computer*, Phys. Rev. Lett. **79**, 2586 (1997).

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

[6] A. Y. Kitaev, *Quantum Measurements and the Abelian Stabilizer Problem*, arXiv:quant-ph/9511026.

[7] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, *Simulated Quantum Computation of Molecular Energies*, Science **309**, 1704 (2005).

[8] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem*, Science **292**, 472 (2001).

[9] L.-A. Wu, M. S. Byrd, and D. A. Lidar, *Polynomial-Time Simulation of Pairing Models on a Quantum Computer*, Phys. Rev. Lett. **89**, 057904 (2002).

[10] R. Babbush, P. J. Love, and A. Aspuru-Guzik, *Adiabatic Quantum Simulation of Quantum Chemistry*, Sci. Rep. **4**, 6603 (2014).

[11] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, *A Variational Eigenvalue Solver on a Photonic Quantum Processor*, Nat. Commun. **5**, 4213 (2014).

[12] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, A. Megrant, J. Y. Mutus, C. Neill *et al.*, *Scalable Quantum Simulation of Molecular Energies*, Phys. Rev. X **6**, 031007 (2016).

[13] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, J. M. Chow, and J. M. Gambetta, *Hardware-efficient Quantum Optimizer for Small Molecules and Quantum Magnets*, Nature (London) **549**, 242 (2017).

[14] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, *Robust Determination of Molecular Spectra on a Quantum Processor*, Phys. Rev. X **8**, 011021 (2018).

[15] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. Roos, *Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator*, Phys. Rev. X **8**, 031022 (2018).

[16] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *The Theory of Variational Hybrid Quantum-Classical Algorithms*, New J. Phys. **18**, 023023 (2016).

[17] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, *Barren Plateaus in Quantum Neural Network Training Landscapes*, arXiv:1803.11173.

[18] D. Wecker, M. B. Hastings, and M. Troyer, *Progress Towards Practical Quantum Variational Algorithms*, Phys. Rev. A **92**, 042303 (2015).

[19] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, *Quantum Simulation of Electronic Structure with Linear Depth and Connectivity*, Phys. Rev. Lett. **120**, 110501 (2018).

[20] J. Romero, R. Babbush, J. McClean, C. Hempel, P. Love, and A. Aspuru-Guzik, *Strategies for Quantum Computing Molecular Energies Using the Unitary Coupled Cluster Ansatz*, arXiv:1701.02691.

[21] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, *Low-Depth Circuit Ansatz for Preparing Correlated Fermionic States on a Quantum Computer*, arXiv:1801.01053.

[22] M. Suzuki, *Improved Trotter-Like Formula*, Phys. Lett. A **180**, 232 (1993).

[23] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Simulating Hamiltonian Dynamics with a Truncated Taylor Series*, Phys. Rev. Lett. **114**, 090502 (2015).

[24] G. H. Low and I. L. Chuang, *Optimal Hamiltonian Simulation by Quantum Signal Processing*, Phys. Rev. Lett. **118**, 010501 (2017).

[25] G. H. Low and I. L. Chuang, *Hamiltonian Simulation by Qubitization*, arXiv:1610.06546.

[26] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, *Improved Techniques for Preparing Eigenstates of Fermionic Hamiltonians*, npj Quantum Inf. **4**, 22 (2018).

[27] D. Poulin, A. Y. Kitaev, D. Steiger, M. Hastings, and M. Troyer, *Fast Quantum Algorithm for Spectral Properties*, Phys. Rev. Lett. **121**, 010501 (2018).

[28] A. Dunsworth, A. Megrant, R. Barends, Y. Chen, Z. Chen, B. Chiaro, A. Fowler, B. Foxen, E. Jeffrey, J. Kelly, P. V. Klimov, E. Lucero, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner *et al.*, *Low Loss Multi-layer Wiring for Superconducting Microwave Devices*, Appl. Phys. Lett. **112**, 063502 (2018).

[29] X. Fu, M. A. Rol, C. C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels, *An Experimental Microarchitecture for a Superconducting Quantum Processor*, Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, Massachusetts, 2017 (ACM, New York, 2017), p. 813–825.

[30] N. T. Bronn, V. P. Adiga, S. B. Olivadese, X. Wu, J. M. Chow, and D. P. Pappas, *High Coherence Plane Breaking Packaging for Superconducting Qubits*, Quantum Sci. Technol. **3**, 024007 (2018).

[31] S. B. Bravyi and A. Y. Kitaev, *Quantum Codes on a Lattice with Boundary*, arXiv:quant-ph/9811052.

[32] E. Dennis, A. Y. Kitaev, A. Landahl, and J. Preskill, *Topological Quantum Memory*, J. Math. Phys. (N.Y.) **43**, 4452 (2002).

[33] R. Raussendorf and J. Harrington, *Fault-Tolerant Quantum Computation with High Threshold in Two Dimensions*, Phys. Rev. Lett. **98**, 190504 (2007).

[34] R. Raussendorf, J. Harrington, and K. Goyal, *Topological Fault-Tolerance in Cluster State Quantum Computation*, New J. Phys. **9**, 199 (2007).

[35] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, *Surface Codes: Towards Practical Large-Scale Quantum Computation*, Phys. Rev. A **86**, 032324 (2012).

[36] A. G. Fowler and S. J. Devitt, *A Bridge to Lower Overhead Quantum Computation*, arXiv:1209.0510.

[37] J. Hubbard, *Electron Correlations in Narrow Energy Bands*, Proc. R. Soc. A **276**, 238 (1963).

[38] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular Electronic Structure Theory* (Wiley, New York, 2002).

[39] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K. -L. Chan, *Low-Depth Quantum Simulation of Materials*, Phys. Rev. X **8**, 011044 (2018).

[40] I. Kivlichan, N. Wiebe, C. Gidney, J. McClean, W. Sun, V. Denchev, A. Fowler, A. Aspuru-Guzik, and R. Babbush, *Low T Gate Trotter-Based Quantum Simulation of Correlated Electrons* (unpublished).

[41] S. R. White, *Hybrid Grid/Basis Set Discretizations of the Schrödinger Equation*, J. Chem. Phys. **147**, 244102 (2017).

[42] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, *Exponentially More Precise Quantum Simulation of Fermions in Second Quantization*, New J. Phys. **18**, 033032 (2016).

[43] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, *Elucidating Reaction Mechanisms on Quantum Computers*, Proc. Natl. Acad. Sci. U.S.A. **114**, 7555 (2017).

[44] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, *Simulation of Electronic Structure Hamiltonians Using Quantum Computers*, Mol. Phys. **109**, 735 (2011).

[45] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, *Gate-Count Estimates for Performing Quantum Chemistry on Small Quantum Computers*, Phys. Rev. A **90**, 022305 (2014).

[46] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik, *Exploiting Locality in Quantum Computation for Quantum Chemistry*, J. Phys. Chem. Lett. **5**, 4368 (2014).

[47] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, *The Trotter Step Size Required for Accurate Quantum Simulation of Quantum Chemistry*, Quantum Inf. Comput. **15**, 361 (2015).

[48] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe, *Chemical Basis of Trotter-Suzuki Errors in Chemistry Simulation*, Phys. Rev. A **91**, 022311 (2015).

[49] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, *Quantum Algorithm for Simulating Real Time Evolution of Lattice Hamiltonians*, arXiv:1801.03922.

[50] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, *Solving Strongly Correlated Electron Models on a Quantum Computer*, Phys. Rev. A **92**, 062318 (2015).

[51] N. Cody Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. V. Meter, A. Aspuru-Guzik, and Y. Yamamoto, *Faster Quantum Chemistry Simulation on Fault-Tolerant Quantum Computers*, New J. Phys. **14**, 115023 (2012).

[52] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, *Toward the First Quantum Simulation with Quantum Speedup*, arXiv:1711.10980.

[53] M. Szegedy, *Quantum Speed-up of Markov Chain Based Algorithms*, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2004), pp. 32–41.

[54] A. M. Childs and N. Wiebe, *Hamiltonian Simulation Using Linear Combinations of Unitary Operations*, Quantum Inf. Comput. **12**, 901 (2012).

[55] D. Aharonov and A. Ta-Shma, *Adiabatic Quantum State Generation and Statistical Zero Knowledge*, in *Proceedings of the 35th ACM Symposium on Theory of Computing—STOC '03* (ACM Press, New York, 2003), p. 20.

[56] A. Luis and J. Peřina, *Optimum Phase-Shift Estimation and the Quantum Description of the Phase Difference*, Phys. Rev. A **54**, 4564 (1996).

[57] R. B. Griffiths and C.-S. Niu, *Semiclassical Fourier Transform for Quantum Computation*, Phys. Rev. Lett. **76**, 3228 (1996).

[58] C. Gidney and R. Babbush, *Quantum Read-Only Memory for Implement Efficient Fault-Tolerant Quantum Oracles* (in press).

[59] R. Babbush, D. W. Berry, Y. R. Sanders, I. D. Kivlichan, A. Scherer, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, *Exponentially More Precise Quantum Simulation*

of Fermions in the Configuration Interaction Representation, Quantum Sci. Technol. **3**, 015006 (2018).

[60] C. Gidney, *Halving the Cost of Quantum Addition*, arXiv:1709.06648.

[61] R. D. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, *Simulating Physical Phenomena by Quantum Networks*, Phys. Rev. A **65**, 042323 (2002).

[62] C. Gidney, R. Babbush, M. Mohseni, and H. Neven, *Quantum Read-Only Memory for Implementing Efficient Fault-Tolerant Quantum Oracles* (in press).

[63] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum Random Access Memory*, Phys. Rev. Lett. **100**, 160501 (2008).

[64] B. C. Travaglione, M. A. Nielsen, H. M. Wiseman, and A. Ambainis, *ROM-based Computation: Quantum versus Classical*, Quantum Inf. Comput. **2**, 324 (2002).

[65] I. Kerenidis and A. Prakash, *Quantum Recommendation Systems*, arXiv:1603.08675.

[66] V. Giovannetti, S. Lloyd, and L. Maccone, *Architectures for a Quantum Random Access Memory*, Phys. Rev. A **78**, 052310 (2008).

[67] F.-Y. Hong, Y. Xiang, Z.-Y. Zhu, L.-Z. Jiang, and L.-N. Wu, *Robust Quantum Random Access Memory*, Phys. Rev. A **86**, 010306 (2012).

[68] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, *On the Robustness of Bucket Brigade Quantum RAM*, New J. Phys. **17**, 123010 (2015).

[69] V. V. Shende, S. S. Bullock, and I. L. Markov, *Synthesis of Quantum-Logic Circuits*, IEEE Trans. CAD Integrated Circuits Syst. **25**, 1000 (2006).

[70] L. K. Grover, *Synthesis of Quantum Superpositions by Quantum Computation*, Phys. Rev. Lett. **85**, 1334 (2000).

[71] P. Høyer, *Arbitrary Phases in Quantum Amplitude Amplification*, Phys. Rev. A **62**, 052304 (2000).

[72] A. Walker, *New Fast Method for Generating Discrete Random Numbers with Arbitrary Frequency Distributions*, Electron. Lett. **10**, 127 (1974).

[73] M. Vose, *A Linear Algorithm for Generating Random Numbers with a Given Distribution*, IEEE Transactions on Software Engineering **17**, 972 (1991).

[74] Code provided at www.openfermion.org.

[75] J. R. McClean, I. D. Kivlichan, K. J. Sung, D. S. Steiger, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, T. Häner, T. Hardikar, V. Havlíček, C. Huang, Z. Jiang, M. Neeley, T. O'Brien, I. Ozfidan, M. D. Radin, J. Romero, N. Rubin et al., *OpenFermion: The Electronic Structure Package for Quantum Computers*, arXiv:1710.07629.

[76] P. Jordan and E. Wigner, *Über das Paulische Äquivalenzverbot*, Z. Phys. **47**, 631 (1928).

[77] G. Evenbly and S. R. White, *Representation and Design of Wavelets Using Unitary Circuits*, Phys. Rev. A **97**, 052314 (2018).

[78] P. Hohenberg and W. Kohn, *Inhomogeneous Electron Gas*, Phys. Rev. **136**, B864 (1964).

[79] M. Stone, *Quantum Hall Effect* (World Scientific, Singapore, 1992).

[80] D. M. Ceperley and B. J. Alder, *Ground State of the Electron Gas by a Stochastic Method*, Phys. Rev. Lett. **45**, 566 (1980).

[81] B. Tanatar and D. M. Ceperley, *Ground State of the Two-Dimensional Electron Gas*, Phys. Rev. B **39**, 5005 (1989).

[82] F. H. Zong, C. Lin, and D. M. Ceperley, *Spin Polarization of the Low-Density Three-Dimensional Electron Gas*, Phys. Rev. E **66**, 036703 (2002).

[83] C. Attaccalite, S. Moroni, P. Gori-Giorgi, and G. B. Bachelet, *Correlation Energy and Spin Polarization in the 2D Electron Gas*, Phys. Rev. Lett. **88**, 256601 (2002).

[84] N. D. Drummond and R. J. Needs, *Phase Diagram of the Low-Density Two-Dimensional Homogeneous Electron Gas*, Phys. Rev. Lett. **102**, 126402 (2009).

[85] G. G. Spink, R. J. Needs, and N. D. Drummond, *Quantum Monte Carlo Study of the Three-Dimensional Spin-Polarized Homogeneous Electron Gas*, Phys. Rev. B **88**, 085121 (2013).

[86] M. Gell-Mann and K. A. Brueckner, *Correlation Energy of an Electron Gas at High Density*, Phys. Rev. **106**, 364 (1957).

[87] D. L. Freeman, *Coupled-Cluster Expansion Applied to the Electron Gas: Inclusion of Ring and Exchange Effects*, Phys. Rev. B **15**, 5512 (1977).

[88] J. J. Shepherd, G. Booth, A. Grüneis, and A. Alavi, *Full Configuration Interaction Perspective on the Homogeneous Electron Gas*, Phys. Rev. B **85**, 081103 (2012).

[89] J. J. Shepherd, G. H. Booth, and A. Alavi, *Investigation of the Full Configuration Interaction Quantum Monte Carlo Method Using Homogeneous Electron Gas Models*, J. Chem. Phys. **136**, 244101 (2012).

[90] M. T. Wilson and B. L. Gyorffy, *A Constrained Path Auxiliary-Field Quantum Monte Carlo Method for the Homogeneous Electron Gas*, J. Phys. Condens. Matter **7**, L371 (1995).

[91] M. Motta, D. E. Galli, S. Moroni, and E. Vitali, *Imaginary Time Density-Density Correlations for Two-Dimensional Electron Gases at High Density*, J. Chem. Phys. **143**, 164108 (2015).

[92] R. Martin, *Electronic Structure* (Cambridge University Press, Cambridge, England, 2004).

[93] R. Martin, L. Reining, and D. Ceperley, *Interacting Electrons* (Cambridge University Press, Cambridge, England, 2016).

[94] S. Tosoni, C. Tuma, J. Sauer, B. Civalleri, and P. Ugliengo, *A Comparison between Plane Wave and Gaussian-type Orbital Basis Sets for Hydrogen Bonded Systems: Formic Acid as a Test Case*, J. Chem. Phys. **127**, 154102 (2007).

[95] G. Knizia and G. K.-L. Chan, *Density Matrix Embedding: A Simple Alternative to Dynamical Mean-Field Theory*, Phys. Rev. Lett. **109**, 186404 (2012).

[96] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, *Hybrid Quantum-Classical Approach to Correlated Materials*, Phys. Rev. X **6**, 031045 (2016).

[97] J. P. F. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K. -L. Chan, C.-M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jiménez-Hoyos, E. Kozik, X.-W. Liu, A. J. Millis, N. V. Prokofev, M. Qin, G. E. Scuseria, H. Shi, B. V. Svistunov, L. F. Tocchio, I. S. Tupitsyn, S. R. White et al., *Solutions of the Two-Dimensional Hubbard Model: Benchmarks and Results from a Wide Range of Numerical Algorithms*, Phys. Rev. X **5**, 041041 (2015).

[98] E. H. Lieb and F. Y. Wu, *Absence of Mott Transition in an Exact Solution of the Short-Range, One-Band Model in One Dimension*, Phys. Rev. Lett. **20**, 1445 (1968).

[99] W. Metzner and D. Vollhardt, *Correlated Lattice Fermions in Infinity Dimensions*, Phys. Rev. Lett. **62**, 324 (1989).

[100] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo, *Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions*, Phys. Rev. Applied **9**, 044036 (2018).

[101] E. H. Lieb and D. W. Robinson, *The Finite Group Velocity of Quantum Spin Systems*, in *Statistical Mechanics* (Springer, New York, 1972), pp. 425–431.

[102] N. Wiebe, D. W. Berry, P. Hoyer, and B. C. Sanders, *Higher Order Decompositions of Ordered Operator Exponentials*, J. Phys. A **43**, 065203 (2010).

[103] A. G. Fowler, *Optimal Complexity Correction of Correlated Errors in the Surface Code*, arXiv:1310.0863.

[104] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, *Poking Holes and Cutting Corners to Achieve Clifford Gates with the Surface Code*, Phys. Rev. X **7**, 021029 (2017).

[105] A. G. Fowler, *Low-Overhead Surface Code Logical Hadamard*, Quantum Inf. Comput. **12**, 970 (2012).

[106] S. Bravyi and A. Y. Kitaev, *Universal Quantum Computation with Ideal Clifford Gates and Noisy Ancillas*, Phys. Rev. A **71**, 022316 (2005).

[107] B. W. Reichardt, *Quantum Universality from Magic States Distillation Applied to CSS Codes*, Quantum Inf. Process. **4**, 251 (2005).

[108] A. Paler, A. G. Fowler, and R. Wille, *Synthesis of Arbitrary Quantum Circuits to Topological Assembly: Systematic, Online and Compact*, Sci. Rep. **7**, 10414 (2017).

[109] I. D. Kivlichan, N. Wiebe, R. Babbush, and A. Aspuru-Guzik, *Bounding the Costs of Quantum Simulation of Many-Body Physics in Real Space*, J. Phys. A **50**, 305301 (2017).

[110] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, *Surface Code Quantum Computing by Lattice Surgery*, New J. Phys. **14**, 123011 (2012).