

Exponential Improvements in the Simulation of Lattice Gauge Theories Using Near-Optimal Techniques

Mason L. Rhodes^{1,2,3}, Michael Kreshchuk^{4,5}, and Shivesh Pathak^{6,*}

¹Center for Computing Research, *Sandia National Laboratories*, Albuquerque, New Mexico 87185, USA

²Center for Quantum Information and Control, *University of New Mexico*, Albuquerque, New Mexico 87131, USA

³Department of Physics and Astronomy, *University of New Mexico*, Albuquerque, New Mexico 87131, USA

⁴Physics Division, *Lawrence Berkeley National Laboratory*, Berkeley, California 94720, USA

⁵Phasecraft Inc., Washington, DC 20001, USA

⁶Quantum Algorithms and Applications Collaboratory, *Sandia National Laboratories*, Albuquerque, New Mexico 87185, USA



(Received 19 June 2024; accepted 22 November 2024; published 23 December 2024)

We report a first-of-its-kind analysis on post-Trotter simulation of U(1), SU(2), and SU(3) lattice gauge theories including fermions in arbitrary spatial dimension. We provide explicit circuit constructions as well as T-gate counts and logical qubit counts for Hamiltonian simulation. We find a reduction of up to 25 orders of magnitude in space-time volume over Trotter methods for simulations of non-Abelian lattice gauge theories relevant to the standard model. This improvement results from our algorithm having polynomial scaling with the number of colors in the gauge theory, achieved by utilizing oracle constructions relying on the sparsity of physical operators, in contrast to the exponential scaling seen in state-of-the-art Trotter methods, which employ explicit mappings onto Pauli operators. Our work demonstrates that the use of advanced algorithmic techniques leads to dramatic reductions in the cost of simulating fundamental interactions, bringing it in step with resources required for first-principles quantum simulation of chemistry.

DOI: [10.1103/PRXQuantum.5.040347](https://doi.org/10.1103/PRXQuantum.5.040347)

I. INTRODUCTION

The simulation of strongly interacting quantum many-body systems has continued to be a challenging problem in classical and quantum simulation. Simulations of quantum chemistry [1], condensed-matter physics [2], and lattice quantum chromodynamics [3,4] consume a significant portion of the world's high-performance computing resources. The difficulty of accurately and efficiently performing such simulations on classical computers inspired the development of the field of quantum computing [5] and is still considered one of its most exciting potential applications [6–8]. Substantial effort has already been devoted to efficiently simulating electronic structure from first principles, employing sophisticated algorithms based on both first- and second-quantized formulations [9–40]. Owing to its

classical hardness [41], of particular interest is utilizing quantum computers for dynamical simulations of fundamental interactions, particularly non-Abelian lattice gauge theories (LGTs).

There is a significant corpus of work on the quantum simulation of the dynamics of LGTs using Trotter product-formula techniques. Studies include simulation algorithms of Abelian and non-Abelian finite gauge groups [42–52] as well as Lie groups [53–57], from one to three spatial dimensions [47–50,54–57], constructed for both noisy intermediate-scale quantum (NISQ) hardware and fault-tolerant hardware (FT), and in some cases with rigorous resource estimates for NISQ and FT regimes [53,55,56] or even experimental realizations on contemporary NISQ hardware [50–52,57]. While these papers constitute a thorough body of literature in tandem, each result is limited, such as only including implementations for a particular gauge group, being limited to a particular spatial dimension, or only simulating pure gauge theories without fermions. One standout paper, however, by Kan and Nam [58] provides a quantum algorithm and FT resource estimates for Trotterized simulation of lattice gauge theories applicable for one to three spatial dimensions, including

*Contact author: sapatha@sandia.gov

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

fermions, and for U(1), SU(2), and SU(3) LGTs, and has been a benchmark of comparison in the quantum simulation of LGTs.

Trotter-based methods, while having simple implementations and prospectively small constant-factor overheads [59], have suboptimal asymptotic scaling in parameters such as problem size, evolution time, gauge group dimension, spatial dimension, and error. A particular challenge in LGT simulation is the exponential scaling in the number of color degrees of freedom in state-of-the-art Trotter methods [53,58], resulting in dramatic increases in gate counts from 10^{23} expensive non-Clifford operations for FT simulation of U(1) in 2+1 dimensions (2+1D), to 10^{40} for SU(2) and 10^{55} for SU(3) gauge theories [58]. As such, simulation of LGTs using post-Trotter methods that possess optimal or near-optimal dependence problem parameters [60–66]—which we collectively refer to as *qubitization* (see Appendix A)—may be advantageous over Trotter methods in certain parameter regimes. However, to date, no comprehensive study of qubitized simulation of LGTs has been carried out to address these potential advantages, with the few extant results relegated to small system sizes using the quantum eigenvalue transformation [67], light-front quantization [68,69], and quantum signal-processing techniques [60,70–72].

In this work, we build on and fully develop the algorithm for qubitized Hamiltonian simulation of LGTs, based on the proposal by Tong *et al.* [60], and work out all the necessary details for U(1), SU(2), and SU(3) gauge theories coupled to fermions in arbitrary spatial dimensions. We lay out the ground work of geometrically local fermion-to-qubit encodings and construct explicit block-encoding oracle circuits using the linear combination of unitaries (LCU) and sparse-access oracle constructions. We also estimate all relevant algorithmic parameters, including Lieb-Robinson velocities, and provide complete front-to-back FT resource estimates in terms of T-gate counts and logical qubit counts for qubitized Hamiltonian simulation.

Importantly, when comparing our results to Kan and Nam’s work on Trotterized simulation [58], we find improvements of up to 25 orders of magnitude in T-gate counts for SU(3) LGT simulation. This is primarily due to a polynomial scaling in color degrees of freedom in the qubitized simulation compared to exponential scaling in Trotterized simulation. Such efficiency was achieved by utilizing oracle constructions relying on the sparsity of physical operators rather than on their explicit mappings onto Pauli operators. We note that while Kan and Nam’s work is not representative of the most optimal Trotterized simulation algorithm in terms of gate counts, as the aforementioned corpus of work contains optimizations for specific problem instances [47–50,53–57], our work is, to the best of our knowledge, the first to have polynomial scaling in color degrees of freedom.

The contents of the paper are as follows. In Sec. II, we discuss Hamiltonian discretization using the Kogut-Susskind formalism, followed by explicit constructions for geometric localization of LGTs by means of various fermion-to-qubit mappings in Sec. III. In Sec. IV, we provide explicit constructions of circuits for sparse and LCU block encodings of U(1) gauge theory and comparison of the full interaction-picture qubitized simulation to Trotterization: we also provide two unique constructions, an optimized signed-increment-decrement (SID) circuit, and bounding of Lieb-Robinson velocities for U(1) in dimension $d > 1$. In Sec. V, we provide generalizations of the constructions to SU(2) and SU(3) gauge theories, and we conclude with a discussion of outlooks for simulation of LGTs in Sec. VI.

II. HAMILTONIAN SIMULATION

A. Kogut-Susskind Hamiltonian

Throughout this paper, we will follow the notations used by Kan and Nam [58] for the Kogut-Susskind Hamiltonian. The Hamiltonian is defined over a d -dimensional lattice where each site is denoted by a vector $\vec{n} \in \mathbb{Z}^d$. The lattice sites are labeled even or odd, depending on the sign $(-1)^{\vec{n}} = (-1)^{\sum_i n_i}$. We will take the lattice to be a hypercube with linear dimension N and total N^d sites.

Fermions are distributed on the lattice sites, with the spinor being staggered such that the fermionic and antifermionic spinors are on even and odd sites, respectively. Staggering fermions reduces the number of spurious fermionic degrees of freedom known as *doublers*, which arise due to the periodic nature of the dispersion relation of lattice fermions [73,74]. For gauge theories with various fermion colors, each site contains multiple spinors corresponding to each color degree of freedom. A link between neighboring sites is denoted by the tuple $(\vec{n}, \hat{l})$, where $\hat{l}$ indicates the direction of a nearest-neighbor site to $\vec{n}$. The links contain the gauge degrees of freedom, which mediate the interaction between the fermionic degrees of freedom.

The Hamiltonian itself contains four terms [75]:

$$H = g_M H_M + g_{GM} H_{GM} + g_E H_E + g_B H_B. \quad (1)$$

Here, the four constants are related to the physical parameters, namely, the dimension d , the lattice spacing a , the bare gauge-matter coupling constant g , and the bare fermion mass m , as

$$g_M = m, \quad g_{GM} = \frac{1}{2a}, \quad g_E = \frac{g^2}{2a^{d-2}}, \quad g_B = -\frac{1}{2a^{4-d}g^2}. \quad (2)$$

The first of the four operators is the mass term:

$$H_M = \sum_{\vec{n}} \sum_a (-1)^{\vec{n}} \psi_a^\dagger(\vec{n}) \psi_a(\vec{n}). \quad (3)$$

Here, $\psi_a(\vec{n})$ and $\psi_a^\dagger(\vec{n})$ are the creation and annihilation operators for a fermion with color degree of freedom a on site $\vec{n}$. The $(-1)^{\vec{n}}$ sign is indicative of the staggered nature of the fermions. The next three terms are the gauge-matter, electric field, and magnetic field terms:

$$H_{GM} = \sum_{\vec{n}, \hat{l}} \sum_{a,b} \psi_a^\dagger(\vec{n}) U_{ab}(\vec{n}, \hat{l}) \psi_b(\vec{n} + \hat{l}) + \text{h.c.}, \quad (4)$$

$$H_E = \sum_{\vec{n}, \hat{l}} \sum_a [E_a(\vec{n}, \hat{l})]^2, \quad (5)$$

$$H_B = \sum_{\square} \text{Tr}[P_{\square} + P_{\square}^\dagger]. \quad (6)$$

The plaquette operator $P_{\square}$ is

$$\text{Tr}[P_{\square}] = \sum_{a,b,c,e} U_{ab}(\vec{n}, \hat{i}) U_{bc}(\vec{n} + \hat{i}, \hat{j}) U_{ce}^\dagger(\vec{n} + \hat{j}, \hat{i}) U_{ea}^\dagger(\vec{n}, \hat{j}), \quad (7)$$

where $\hat{i}$ and $\hat{j}$ are links that form a closed loop starting and ending at a site $\vec{n}$.

The gauge link operator $U_{ab}(\vec{n}, \hat{l})$ and electric field operator $E_b(\vec{n}, \hat{l})^2$ depend on the gauge field under consideration. For U(1) theory, the color index can be suppressed. In this case, we further note that the Casimir operator E^2 and the electric field operator E have the same eigenfunctions, which we label as $|k\rangle$ for $k \in \mathbb{Z}$. The operators act on these basis states as follows:

$$E|k\rangle = k|k\rangle, \quad (8)$$

$$U|k\rangle = |k-1\rangle, \quad (9)$$

i.e., E and U , $U^\dagger$ act as the number and ladder operators, correspondingly [76]. For a practical calculation, the number of bosonic modes is truncated such that $k \in [-\Lambda, \Lambda]$ where $\Lambda \in \mathbb{Z}^+$ is some integer.

For the case of the non-Abelian SU(2) and SU(3) theories, the situation is more complicated. Each E_a can no longer be simultaneously diagonalized as they do not commute and instead one operates within the eigenbasis of the Casimir operator. For SU(2), the action of bosonic operators in this basis is given by

$$E^2|j, m^L, m^R\rangle = j(j+1)|j, m^L, m^R\rangle, \quad (10)$$

$$U_{ab}|j, m^L, m^R\rangle = \sum_{J=j-1/2}^{j+1/2} \sqrt{\frac{2j+1}{2J+1}} \langle J, M_L|j, m^L; 1/2, a'\rangle \langle J, M_R|j, m^R; 1/2, b'\rangle |J, M_L = m^L + a', M_R = m^R + b'\rangle. \quad (11)$$

The first two terms including J, M_L and J, M_R in the operation of U_{ab} are the Clebsch-Gordan coefficients for SU(2). Conservation of angular momentum along the z axis requires that $a', b' = -1/2$ or $1/2$ when $a, b = 1$ or 2 ,

respectively. For SU(3), a similar construction applies, with the link basis again being the eigenbasis of the SU(3) Casimir operator,

$$E^2|p, q, T_L, T_L^z, Y_L, T_R, T_R^z, Y_R\rangle = \frac{1}{3} (p^2 + q^2 + pq + 3(p+q)) |p, q, T_L, T_L^z, Y_L, T_R, T_R^z, Y_R\rangle, \quad (12)$$

$$U_{ab}|p, q, T_L, T_L^z, Y_L, T_R, T_R^z, Y_R\rangle = \sum_{(p', q')} \sum_{T_L' = |T_L - t_L|}^{T_L + t_L} \sum_{T_R' = |T_R - t_R|}^{T_R + t_R} \sqrt{\frac{\dim(p, q)}{\dim(p', q')}} \langle p', q', T_L', T_L'^z, Y_L' | p, q, T_L, T_L^z, Y_L; 1, 0, t_L, t_L^z, Y_L \rangle \\ \times \langle p', q', T_R', T_R'^z, Y_R' | p, q, T_R, T_R^z, Y_R; 1, 0, t_R, t_R^z, Y_R \rangle |p', q', T_L', T_L'^z, Y_L', T_R', T_R'^z, Y_R'\rangle, \quad (13)$$

where $p, q \in [0, \Lambda]$ are SU(3) representation labels and T , T^z , and Y are known as the isospin, the z -axis projection of the isospin, and the hypercharge, respectively. Additionally, we have $T \in [0, \Lambda]$ and $T^z \in [-\Lambda, \Lambda]$, where each are incremented in half-integer steps, and $Y \in [-\Lambda, \Lambda]$ incremented in third-integer steps. For the gauge operators, $(p', q') \in \{(p+1, q), (p-1, q+1), (p, q-1)\}$ and $\dim(p, q) = (1+p)(1+q)(1+(p+q)/2)$. Here, the isospin and hypercharge values t , t^z , and y depend on the color degrees of freedom but ultimately correspond to incrementing or decrementing the values of T , T^z , and Y , respectively, by half-integer or third-integer amounts. Finally, $\langle p', q', T', T'^z, Y' | p, q, T, T^z, Y; 1, 0, t, t^z, y \rangle$ are the Clebsch-Gordan coefficients for SU(3) in the fundamental representation. For more details on the SU(3) gauge operators, see Ref. [58, Appendix D].

B. Trotterization

Trotterized approaches to Hamiltonian simulation invoke a product formula approximating the exact time evolution operator as

$$e^{iHT} = (e^{iHT/r})^r \approx \left[\left(\prod_{j=1}^{N_H} e^{iH_j T/2r} \right) \left(\prod_{j=N_H}^1 e^{iH_j T/2r} \right) \right]^r, \quad (14)$$

where we have decomposed the full Hamiltonian into the $N_H = 4$ subterms of Eq. (1), i.e., $j \in \{M, GM, E, B\}$, and we have decomposed the evolution into $r \gg T$ Trotter steps. A number of algorithms exist that explore Trotterized approaches to simulating the dynamics of LGTs [47–58, 77]. We focus in particular on the algorithms presented in Ref. [58], as that work contains the most general estimates provided to date including U(1), SU(2), and

SU(3) LGTs in d spatial dimensions using a second-order Suzuki-Trotter formula [78]. We emphasize that while other works cited provide significantly improved algorithms to the work of Kan and Nam, explicit resource estimates are limited to particular gauge groups or spatial dimensions (such as SU(2) in 1+1D [53]), thereby lacking the generality necessary for comparison to our work.

The number of Trotter steps contributes a multiplicative prefactor to the cost of the full simulation complexity and individual operators $\exp(iH_j T/2r)$ can be realized using polylogarithmic-size circuits. In particular, when considering the complexity scaling with respect to the bosonic truncation parameter Λ , the second-order Suzuki-Trotter formula requires only $\mathcal{O}(\Lambda)$ Trotter steps [56]. On the other hand, typical qubitization approaches have a complexity scaling proportional to the Hamiltonian norm, which is dominated by the electric field term since $\|H_E\| = \mathcal{O}(\Lambda^2)$ in the electric field basis for the gauge field. Thus, the product-formula approach seems to admit a quadratic speed-up over typical qubitization approaches, but this overhead can be avoided by performing simulation in the interaction picture or by moving into a different basis for the gauge fields [45, 79]. This will be discussed further in Sec. II C. In addition to the bosonic truncation parameter, the number of Trotter steps will also scale with the total evolution time T and the precision of the approximation ϵ as $\mathcal{O}(T^{3/2}/\epsilon^{1/2})$ [56] for the second-order product formula.

In Ref. [58], a binary encoding is used for the bosons stored on each lattice edge but the Jordan-Wigner transformation, which does not maintain geometric locality of the resulting qubit Hamiltonian, is used to encode the fermions. The authors then construct explicit circuits for each of the operators $\exp(iH_j T/2r)$ with respect to these encodings for U(1), SU(2), and SU(3) LGTs. The full asymptotic gate complexities are

$$U(1) \sim \mathcal{O}(2^8 N^{3d/2} T^{3/2} \Lambda \epsilon^{-1/2} \text{polylog}(\Lambda)), \quad (15)$$

$$SU(2), SU(3) \sim \mathcal{O}(2^{8(n_c^2-1)} N^{3d/2} T^{3/2} \Lambda \epsilon^{-1/2} \text{polylog}(N^{3d/2} T^{3/2} \Lambda \epsilon^{-3/2})), \quad (16)$$

where we have explicitly included the prefactor scaling exponentially in the number of colors responsible for the increase, by several orders of magnitude, in complexity when scaling up from U(1) to SU(2) and SU(3) in Ref. [58].

This exponential prefactor is a result of the decomposition that the authors perform to block diagonalize the Hamiltonian terms in order to implement their Trotterized time evolution more efficiently. The decomposition splits the Hamiltonian into a sum over all possible parity

configurations, in which off-diagonal elements can be written as $D \cdot P$ where D is a diagonal operator and P is a Pauli string. To obtain the sum over all local quantum numbers, $D \cdot P$ can be conjugated with the possible parity operations over sums involving color indices.

Despite the exponential prefactor, the authors of Ref. [58] note that this is a superpolynomial improvement over the results of Ref. [54], which scale as $\mathcal{O}(\Lambda^{4(n_c^2-1)})$ for SU(2) and SU(3). However, we note that this is only a quadratic improvement over a naive Pauli decomposition.

Each gauge-field operator of size $(n_c^2 - 1) \log \Lambda \times (n_c^2 - 1) \log \Lambda$ can be represented as a string of $\Lambda^{2(n_c^2-1)}$ Pauli operators. Then the magnetic field term, a product of four gauge-field operators, will dominate, scaling as $\Lambda^{8(n_c^2-1)}$. We emphasize this prefactor here to compare to our algorithm later in Sec. II C, which scales polynomially in n_c .

C. Qubitization

While various qubitization-based approaches to simulating LGTs have been considered in the literature [70,71], we focus on an efficient algorithm proposed in Ref. [60] that has a near-optimal scaling in the space-time volume. We consider a d -dimensional lattice with total number of sites N^d , for which we want to carry out dynamical simulation of some geometrically local Hamiltonian H for time T with error ϵ , starting from a state with local quantum numbers between $[-\Lambda_0, \Lambda_0]$ at any gauge link.

The core of the algorithm is the Haah-Hastings-Kothari-Low (HHKL) decomposition [80], which allows us to decompose full time evolution into time evolution of local blocks $\mathcal{B}$, of size $N_B^d = \mathcal{O}(\text{polylog}(N^d T \epsilon^{-1}))$, for which we only require simulation for time $\tau = \mathcal{O}(1)$. The entire simulation of the system then requires $\mathcal{O}(T)$ such segments and $\mathcal{O}(N^d)$ such blocks. The HHKL decomposition is only valid under certain assumptions on the geometric locality of the Hamiltonian, all of which are satisfied by the LGTs under consideration here [60].

One can then consider the time evolution of the Hamiltonian restricted to the local blocks $\mathcal{B}$, which are polylogarithmic in the system size. Rather than constructing $e^{itH^{\mathcal{B}}}$ directly, one finds that for LGTs, using interaction-picture dynamics yields a particularly efficient implementation. Specifically, one can fast-forward the dynamics of the electric field term, $e^{itH_E^{\mathcal{B}}}$, as it is entirely diagonal in the link basis, regardless of the gauge theory. This removes a quadratic dependence on the bosonic cutoff in the overall scaling of the algorithm that would otherwise be necessary to block encode the electric field term. Since each of the electric field terms acts on separate links, all of these terms will commute and their simulation can be performed in parallel, so simulation of $e^{itH_E^{\mathcal{B}}}$ for each local block has only $\mathcal{O}(N_B^d \text{polylog}(\Lambda_0 t)) = \mathcal{O}(\text{polylog}(N^d T \Lambda_0 t \epsilon^{-1}))$ gate complexity.

After fast-forwarding, one can simulate the dynamics with the time-dependent interaction-picture Hamiltonian:

$$H_I^{\mathcal{B}}(t) = e^{itg_E H_E^{\mathcal{B}}} (g_M H_M^{\mathcal{B}} + g_{GM} H_{GM}^{\mathcal{B}} + g_B H_B^{\mathcal{B}}) e^{-itg_E H_E^{\mathcal{B}}}. \quad (17)$$

By making use of a truncated Dyson-series implementation [62], one can simulate the dynamics of $H_I^{\mathcal{B}}(t)$ using $\mathcal{O}(\alpha t \text{polylog}(\alpha t \epsilon^{-1}))$ queries to the time-dependent

block encoding of $H_I^{\mathcal{B}}(t)$, where $\alpha = \max_{s \in (0,t)} \|H_I^{\mathcal{B}}(s)\|$, referred to as HAM-T. Since each term H_M, H_{GM}, H_B is bounded by a constant, the total norm scales simply as N_B^d . Therefore, $\mathcal{O}(\text{polylog}(N^d T \Lambda_0 \epsilon^{-1}))$ queries to $H_I^{\mathcal{B}}(t)$ are required. The implementation of $H_I^{\mathcal{B}}(t)$ itself must scale as N_B^d , as there are only N_B^d terms in each Hamiltonian. Therefore, the final asymptotic gate cost should still be $\mathcal{O}(\text{polylog}(N^d T \Lambda_0 \epsilon^{-1}))$. Finally, given that we must simulate $\mathcal{O}(N^d)$ such blocks at each of the $\mathcal{O}(T)$ time segments, the total asymptotic cost is

$$\text{gate cost} \sim \mathcal{O}(n_c^4 N^d T \text{polylog}(N^d T \Lambda_0 \epsilon^{-1})). \quad (18)$$

The qubitized algorithm implements the gauge operators directly, meaning that for a single plaquette of the magnetic field term, we only need to sum over the n_c^4 possible color configurations at the plaquette vertices. Thus, by using qubitization, we achieve exponential improvement in algorithm complexity with respect to the number of colors compared to the Trotterized approach of Ref. [58], which has scaling $\mathcal{O}(2^{8(n_c^2-1)})$. In this work, we will establish rigorous resource estimates for qubitized simulations of LGTs for various gauge theories and dimensions.

III. GEOMETRICALLY LOCAL QUBIT ENCODINGS

As a final technical stepping stone to resource estimates, we provide explicit constructions for boson- and fermion-to-qubit encodings for arbitrary dimension d and gauge theory $SU(N)$. Importantly, we provide constructions that preserve geometric locality of the Kogut-Susskind Hamiltonian, a necessary component of efficient simulation using the qubitized algorithm involving the HHKL decomposition as described in Sec. II C.

A. Boson-to-qubit mapping

The Hilbert space of each bosonic degree of freedom (DOF) is infinite dimensional, meaning that a truncation on the space is necessary in order to achieve a tractable encoding. The trade-off between the precision of the result and level of truncation is finely tuned but a provable bound has been established in Ref. [60]. In particular, given an initial state in which the local quantum number on every link is within $[-\Lambda_0, \Lambda_0]$, evolving a truncated Hamiltonian for time T on N^d lattice sites to error ϵ can be achieved by keeping bosonic modes only in the range of $[-\Lambda, \Lambda]$, where

$$\Lambda = \Lambda_0 + \tilde{\mathcal{O}}((\chi T + 1) \text{polylog}(N/\epsilon)), \quad (19)$$

with χ being a constant depending solely on the model parameters and not on N and T . For LGTs, we have $\chi = c(4|g_B| + 2|g_{GM}|)$, with c being the number of (color-indexed) gauge operators U_{ab} in the theory. This bound has

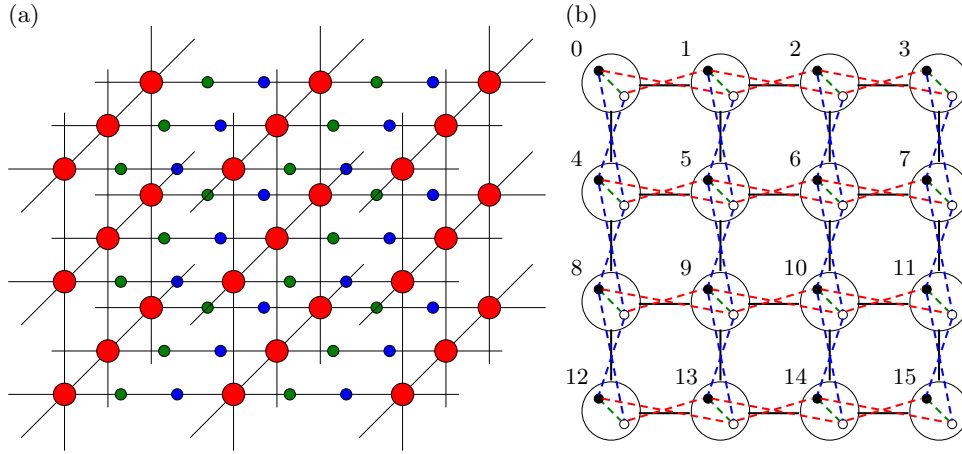


FIG. 1. The fermion-to-qubit encoding schemes. (a) The generalized superfast (GS) encoding [84] for SU(3) in three spatial dimensions on a $3 \times 3 \times 3$ cubic lattice. Each lattice vertex of degree $2d$ is assigned d qubits to store a single color DOF (large red circle) and a single edge for every vertex is decorated with $n_c - 1$ qubits (small green and blue circles) to store the remaining color DOFs. (b) The Verstraete-Cirac encoding [85] for U(1) in two spatial dimensions on a 4×4 square lattice. Each lattice site contains one physical fermion (shaded circle) and one auxiliary fermion (open circle). The sites are enumerated according to our imposed ordering and the vertical and horizontal connectivity between sites is depicted by blue and red dashed lines, respectively, between physical and auxiliary fermions.

been further improved in Ref. [81] to

$$\Lambda = \Lambda_0 + \mathcal{O}(\chi T \log(N \Lambda_0 \chi T / \epsilon)). \quad (20)$$

There are two main approaches taken for direct boson-to-qubit encodings [82]. A unary encoding requires a linear number of qubits but fewer gates to implement oracles, while a binary encoding uses only a logarithmic number of qubits but more gates to implement oracles. We opt to use the latter. The binary encoding stores each bosonic mode $\lambda \in [-\Lambda, \Lambda]$ as a binary representation of the integer λ , using a register of size $\lceil \log(\Lambda) \rceil + 1$, where the additional qubit stores the sign of the integer. Each vertex in the d -dimensional lattice has d associated edges, each of which hosts a boson. Thus, for a lattice on N^d sites, we require $dN^d(\lceil \log(\Lambda) \rceil + 1)$ qubits. Finally, we impose a consistent local ordering of the edges at a site to establish an ordering of all bosons. The full bosonic register can then be written as

$$|\lambda\rangle = |\lambda_1^1, \lambda_1^2, \lambda_1^3, \dots, \lambda_1^d, \lambda_2^1, \lambda_2^2, \dots, \lambda_N^{d-2}, \lambda_N^{d-1}, \lambda_N^d\rangle, \quad (21)$$

where the subscripts are site labels and the superscripts are edge labels at the corresponding site.

B. Fermion-to-qubit mapping

Fermions, unlike bosons, do not suffer from the infinite-dimensional Hilbert space but possess their own technical challenges to be overcome. The traditional Jordan-Wigner encoding [83] maps fermions onto qubits acted on by strings of Pauli operators within the lattice. In two or more

dimensions, one must impose an ordering of the fermions to generate the strings of Pauli operators and this leads to nonlocal operators with weights scaling as $O(N)$. However, in order to invoke the HHKL decomposition [80], we must encode the fermions in a way that preserves their geometric locality.

One such way of encoding fermions that preserves geometric locality is the generalized superfast (GS) encoding [84]. Given a d -dimensional hypercubic lattice, the GS encoding assigns d qubits to each vertex storing the first fermionic color DOF. Then, for each vertex, a single (but consistent) edge is chosen and decorated with $n_c - 1$ qubits storing the remaining color DOFs. An example of the GS encoding for SU(3) in three dimensions is shown in Fig. 1(a).

In order to map fermionic interactions onto these qubits, an associated operator basis must be chosen. To fix an operator basis, we begin with the fermionic creation and annihilation operators $\psi_j^\dagger$ and ψ_j for each fermionic mode j or, more conveniently, with a pair of Majorana operators, $\gamma_j = \psi_j + \psi_j^\dagger$ and $\bar{\gamma}_j = -i(\psi_j - \psi_j^\dagger)$. A generating set for the algebra of even operators is then given by the parity operator V_v for each vertex v and the Majorana bilinear $E_{v,v'}$ for each edge (v, v') , defined by

$$V_v = -i\gamma_v \bar{\gamma}_v = (-1)^{n_v}, \quad (22)$$

$$E_{v,v'} = -i\gamma_v \gamma_{v'}. \quad (23)$$

To obtain the qubit counterparts to V_v and $E_{v,v'}$, we first define for each vertex v a set of $2d$ independent mutually anticommuting Pauli operators $c_{v,1}, \dots, c_{v,2d}$. A local

edge ordering at each vertex is imposed such that $c_{v,j}$ corresponds to a Pauli operator originating from the j th edge of vertex v . Then, the parity and bilinear terms generating the even-parity algebra are

$$V_v = (-i)^d \prod_{j=1}^{2d} c_j, \quad (24)$$

$$E_{v,v'} = c_{v,j} c_{v',k} = -E_{v',v}, \quad (25)$$

where (v, v') corresponds to an edge between lattice vertices v and v' , while j and k correspond to the imposed edge ordering of each vertex such that the j th edge of vertex v is the k th edge of vertex v' . Importantly, this encoding uses only one auxiliary qubit per lattice site and can be easily generalized to higher dimensions and to $SU(N)$ LGTs.

An alternative locality-preserving fermionic encoding in two spatial dimensions is the Verstraete-Cirac (VC) encoding [85]. The VC encoding assigns an auxiliary fermion for each physical fermion at a site to facilitate fermionic hopping in two spatial dimensions, as opposed to one-dimensional fermionic hopping in the Jordan-Wigner encoding. There is one physical fermion, and thus one auxiliary fermion, for each color DOF in the LGT, meaning that n_c auxiliary qubits per lattice site are needed. An example of the VC encoding for $U(1)$ in two dimensions is shown in Fig. 1(b).

Again, the fermion interactions must be mapped onto qubit operations obeying the fermionic anticommutation

relations, which amounts to fixing an associated Pauli basis. Such a procedure is discussed in Ref. [85] but the resulting Hamiltonian is not provided. We construct the explicit VC-encoded Hamiltonian here. In order to do so, we need to impose an ordering of the physical and auxiliary fermions in our system, namely, the ordering shown in Fig. 1(b). We first enumerate the lattice sites using a vector $x = (k, l)$, where k is the column index and l is the row index. Then, within each site, the fermions are ordered such that they alternate between physical and auxiliary fermions for each color. The resulting basis is

$$|f\rangle = |f_1^1, \tilde{f}_1^1, f_1^2, \tilde{f}_1^2, \dots, f_1^{n_c}, \tilde{f}_1^{n_c}, f_2^1, \tilde{f}_2^1, \dots, f_N^{n_c}, \tilde{f}_N^{n_c}\rangle, \quad (26)$$

where $\tilde{f}$ indicates an auxiliary fermion, the subscripts are site labels, and the superscripts are color labels.

Next, the Jordan-Wigner transformation is performed on the composite system of physical and auxiliary fermions in order to obtain the encoded Hamiltonian. In particular, the mass term of Eq. (3) transforms as

$$H_M \rightarrow \sum_{k,l} \sum_{a=1}^{n_c} (-1)^{(k,l)} (1 - Z_{k,l}^a), \quad (27)$$

which acts only on the physical fermionic DOFs. On the other hand, the gauge-matter Hamiltonian of Eq. (4) splits into two terms. First, we have the usual horizontal-hopping term that arises in the Jordan-Wigner encoding but now acts on both physical and auxiliary fermions,

$$H_{GM}^h = \frac{1}{4} \sum_{k,l} \sum_{a,b=1}^{n_c} (X_{k+1,l} - iY_{k+1,l})^a U(k+1, l; k, l)^{ab} (X_{k,l} + iY_{k,l})^b S(k+1, l; k, l)^{ab}, \quad (28)$$

where

$$S(k+1, l; k, l)^{ab} = \tilde{Z}_{k,l}^b Z_{k,l}^{b+1} \dots Z_{k,l}^{n_c} \tilde{Z}_{k,l}^{n_c} Z_{k+1,l}^1 \tilde{Z}_{k+1,l}^1 \dots Z_{k+1,l}^{a-1} \tilde{Z}_{k+1,l}^{a-1} \quad (29)$$

is just a string of Pauli-Z operators between fermions $f_{(k,l)}^b$ and $f_{(k+1,l)}^a$. In addition to the horizontal-hopping term, we obtain a vertical-hopping term of the form

$$H_{GM}^v = \frac{1}{4} \sum_{k,l} \sum_{a,b=1}^{n_c} (X_{k,l+1} - iY_{k,l+1})^a U(k, l+1; k, l)^{ab} (X_{k,l} + iY_{k,l})^b (-1)^{l+1} \begin{cases} \tilde{Y}_{k,l}^a \tilde{X}_{k,l+1}^b, & \text{if } k \text{ is even,} \\ \tilde{X}_{k,l}^a \tilde{Y}_{k,l+1}^b, & \text{otherwise.} \end{cases} \quad (30)$$

The electric field term of Eq. (5) and the magnetic field term of Eq. (6) act purely on the gauge fields and are

therefore left unchanged under the VC encoding. Lastly, there is an additional auxiliary Hamiltonian term,

$$\begin{aligned}
H_{\text{aux}} = & \sum_{\substack{k \text{ odd} \\ l \text{ odd}}} \sum_{a=1}^{n_c} \tilde{Y}_{k,l} \tilde{Y}_{k+1,l} \tilde{Y}_{k,l+1} \tilde{Y}_{k+1,l+1} S(k+1, l; k, l) {}^{aa} S(k+1, l+1; k, l+1) {}^{aa} \\
& + \sum_{\substack{k \text{ even} \\ l \text{ odd}}} \sum_{a=1}^{n_c} \tilde{X}_{k,l} \tilde{X}_{k+1,l} \tilde{X}_{k,l+1} \tilde{X}_{k+1,l+1} S(k+1, l; k, l) {}^{aa} S(k+1, l+1; k, l+1) {}^{aa} \\
& + \sum_{\substack{k \text{ odd} \\ l \text{ even}}} \sum_{a=1}^{n_c} \tilde{Y}_{k,l} \tilde{Y}_{k+1,l} \tilde{Y}_{k,l+1} \tilde{Y}_{k+1,l+1} S(k, l; k-1, l) {}^{aa} S(k+2, l+1; k+1, l+1) {}^{aa} \\
& + \sum_{\substack{k \text{ even} \\ l \text{ even}}} \sum_{a=1}^{n_c} \tilde{X}_{k,l} \tilde{X}_{k+1,l} \tilde{X}_{k,l+1} \tilde{X}_{k+1,l+1} S(k, l; k-1, l) {}^{aa} S(k+2, l+1; k+1, l+1) {}^{aa}, \tag{31}
\end{aligned}$$

which is necessary to fix the ground state of the full Hamiltonian after the addition of the auxiliary fermions. However, as we are only interested in studying the dynamics of LGTs, provided that our initial state is an eigenstate of H_{aux} , it will only contribute an irrelevant global phase to the final state. In fact, the physical states of the LGT must be eigenstates of H_{aux} , so we will always be preparing an initial state that is an eigenstate of H_{aux} .

Finally, we note that for $U(1)$ in two spatial dimensions, the GS and VC encodings require the same number of qubits per site and have Pauli operators of equivalent weight. We first present explicit circuit constructions for $U(1)$ in two dimensions and for pedagogical reasons, we use the explicit VC-encoded Hamiltonian that we have just constructed. We defer employing the GS until we discuss generalizations to higher dimensions and arbitrary $SU(N)$ LGTs.

IV. $U(1)$ GAUGE THEORY IN TWO DIMENSIONS

A. Fast-forwarding H_E

Within the interaction picture, the only term of the Hamiltonian that is simulated without block encoding is the electric field term H_E as it is diagonal in the electric field basis and can be fast-forwarded [86]. Applying the HHKL algorithm [80], the electric field term is decomposed spatially into blocks $\mathcal{B}$ and we are only tasked with determining the circuit complexity of time-evolving $H_E^{\mathcal{B}}$, the electric field within a single block.

We efficiently simulate the time evolution from $H_E^{\mathcal{B}}$ using a general circuit for simulating diagonal operators in Fig. 2 [87, Rule 1.6]. Here, U_k is a unitary that takes an electric field eigenstate and prepares a binary representation of the associated eigenvalue on an ancilla register to p bits of precision. In general, the eigenvalues of the Casimir operators for LGTs are products of the representation indices, so one way of efficiently implementing U_k is via a Karatsuba-based integer multiplier quantum circuit

[88]. This algorithm requires $4p^2 - 3p$ Toffoli gates and is linear in p ancilla space. However, for the special case of $U(1)$, we note that given an eigenstate $|k\rangle$, we only need to prepare the eigenvalue k^2 on the ancilla register to $p = \lceil \log \Lambda^2 \rceil = \lceil 2 \log \Lambda \rceil$ bits of precision. Squaring an integer can be accomplished more efficiently than general multiplication, as noted in Ref. [28, Lemma 7], requiring only $\lceil \log \Lambda \rceil (\lceil \log \Lambda \rceil - 1)$ Toffoli gates. Additionally, $P(\theta)$ is the phase gate that performs rotations about the z axis for each $\theta \in \{-2^j t \bmod 2\pi\}_{j=0}^{p-1}$. Each phase gate can be implemented to arbitrary precision ϵ within the Clifford+T gate set using at most $4 \log(1/\epsilon)$ T gates [89]. Here, we can set $\epsilon < \min_{0 \leq j \leq p-1} \{-2^j t \bmod 2\pi\}$, meaning that the full T complexity is

$$\begin{aligned}
T[e^{-itH_E^{\mathcal{B}}}] = & \left(8 \log \Lambda (\log \Lambda - 1/2) + 8 \log \Lambda \log(1/\epsilon) \right) \\
& \times dN_{\mathcal{B}}^d. \tag{32}
\end{aligned}$$

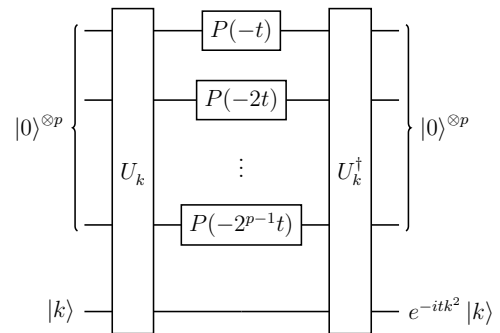


FIG. 2. A circuit for fast-forwarding the time evolution of the electric field Hamiltonian $H_E^{\mathcal{B}}$ in $U(1)$. Given an electric field eigenstate $|k\rangle$, the unitaries U_k prepare the eigenvalue k^2 on an ancilla register to p bits of precision, then a series of phase gate rotations $P(\theta)$ are performed to various angles θ , and finally the ancilla register is uncomputed.

While the scaling of this circuit holds for arbitrary Λ and utilizes the best asymptotically scaling subroutine for implementing U_k to date, the large prefactors make this approach less desirable for practical implementations. In practice, Λ can often be small, as is the case for the results presented in Ref. [58] where $\Lambda = 10$. For small Λ , an alternative circuit implementation of U_k can be realized using a quantum lookup table or, equivalently, a quantum read-only memory (QROM). Here, the possible eigenvalues k^2 are classically precomputed and stored in a database for each $k \in [-\Lambda, \Lambda]$. Upon receiving an index state $|k\rangle$, the QROM loads the associated eigenvalue k^2 into the quantum computer. The complexity of implementing a QROM is $4\Lambda - 4$ [18], meaning that this alternative approach to fast-forwarding the electric field term scales as

$$T[e^{-itH_E^B}] = \left(2(4\Lambda - 4) + 8 \log \Lambda \log(1/\epsilon)\right) \times dN_B^d. \quad (33)$$

In Fig. 3, we show the complexity of both subroutines for implementing U_k as a function of Λ . We see that for both values of ϵ , the arithmetical implementation is equal to or better than using QROM. The major difference between the two implementations, however, appears at $\Lambda \sim 40$, in that the arithmetical method is significantly better. However, in the case of interest, $\Lambda = 10$, using either methodology would yield a similar estimate for U(1) LGTs.

B. Sparse-access oracles

After fast-forwarding the evolution of H_E , the remaining Hamiltonian terms need to be block encoded in order to perform qubitized simulation of the full Hamiltonian.

In this section, we consider the sparse-access oracles, deferring the LCU block-encoding oracles to Sec. IV C.

In general, three oracles are needed to encode a sparse operator:

$$O_r |i\rangle |\ell\rangle = |i\rangle |c(i, \ell)\rangle, \quad (34)$$

$$O_c |j\rangle |\ell\rangle = |j\rangle |r(j, \ell)\rangle, \quad (35)$$

$$O_H |i\rangle |j\rangle |k\rangle = |i\rangle |j\rangle |k \oplus H_{ij}\rangle, \quad (36)$$

where $c(i, \ell)$ and $r(j, \ell)$ are functions that return the column and row index of the ℓ th nonzero matrix element in row i and column j , respectively, and H_{ij} is a binary representation of the Hamiltonian matrix-element value. Note that for Hermitian operators, such as Hamiltonians, then we have $O_r = O_c = O_F$, which we will make use of in the remainder of this work.

In order to realize these oracles for the VC-encoded Hamiltonian, we employ some useful features of the Pauli algebra to simplify the structure of the oracles. Generalizing the results of Ref. [90, Lemma S.1] to s -sparse Hamiltonians, the oracles can be rewritten in the following form suitable for LGT Hamiltonians:

$$O_F |f, \lambda\rangle |\ell\rangle = |f, \lambda\rangle |f \oplus a_\ell, \lambda_\ell^{h,v} - 1\rangle, \quad (37)$$

$$O_H |f, \lambda\rangle |f', \lambda'\rangle |z\rangle = |f, \lambda\rangle |f', \lambda'\rangle |z \oplus H_{(f, \lambda), (f', \lambda')}\rangle. \quad (38)$$

Here, the states $|f\rangle$ and $|\lambda\rangle$ store the fermionic and bosonic occupation states as in Eq. (26) and Eq. (21), respectively. Further, $|\ell\rangle$ stores a binary representation of an integer indexing the lattice sites and $|z\rangle$ is a register that will store the matrix-element value. The matrix-element value after the action of a Pauli operator can be expressed as

$$H_{(f, \lambda), (f', \lambda')} = \begin{cases} (-i)^{a_\ell \cdot b_{\ell,j}} (-1)^{b_{\ell,j} \cdot f}, & \text{if } |f', \lambda'\rangle = |f \oplus a_\ell, \lambda_\ell^{h,v} - 1\rangle, \\ 0, & \text{otherwise,} \end{cases} \quad (39)$$

where a_ℓ and $b_{\ell,j}$ are the classically precomputed length- $2N_B^d$ symplectic binary representations of a Pauli operator $P = P_1 \otimes P_2 \otimes \dots \otimes P_{N_B^d}$, defined by

$$(a_\ell)_k = \begin{cases} 0, & \text{if } P_k = I, Z, \\ 1, & \text{if } P_k = X, Y, \end{cases} \quad \text{and} \quad (b_{\ell,j})_k = \begin{cases} 0, & \text{if } P_k = I, X, \\ 1, & \text{if } P_k = Y, Z. \end{cases} \quad (40)$$

We obtain a vector for each lattice-site index ℓ and the vectors b are additionally indexed by $j \in [0, 3]$ to determine which of the weight-4 Pauli operators is applied at the site according to Eqs. (28) and (30).

We note that the LCU model makes use of an index register $|\ell\rangle$ that iterates over physical sites in a system, while

the sparse-access model requires $|\ell\rangle$ to index the locations of the nonzero matrix-element values. However, here we make use of the fact that the Pauli terms in the Hamiltonian map a fermionic state to at most one other fermionic state, so the sparsity of the Hamiltonian terms is at most N_B^d . The symplectic binary vectors a_ℓ are unique for each site and

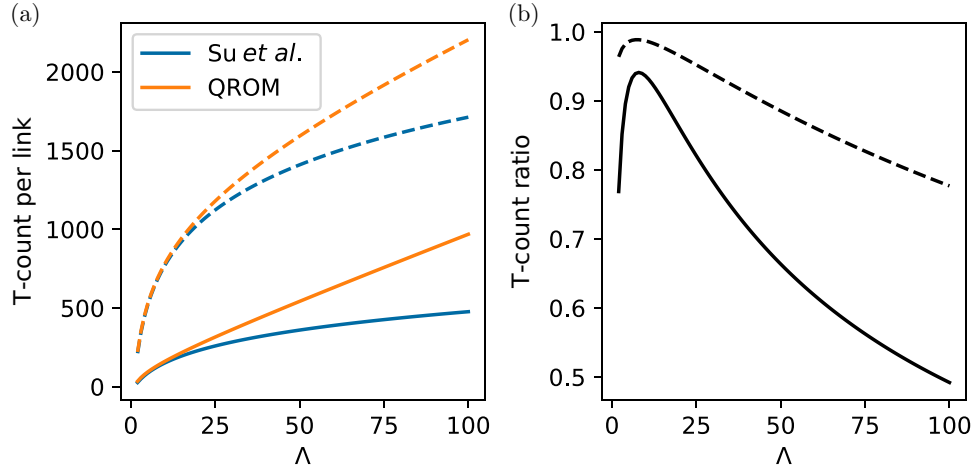


FIG. 3. The associated complexity of implementing the unitary U_k for fast-forwarding the electric field term using an arithmetic circuit of Su *et al.* [28] and QROM for U(1) gauge theory. (a) The resources per gauge link, namely, $T[e^{-iH_E^B}]/(dN_B^d)$, are shown. The solid lines correspond to $\epsilon = 10^{-1}$ and the dashed lines to $\epsilon = 10^{-8}$. (b) The ratio of the arithmetic T count to the QROM T count.

this allows us to map our lattice-site vectors $x = (k, l)$ onto an integer index ℓ using the imposed enumeration scheme in Fig. 1(b). Therefore, the sparsity index ℓ can be more naturally thought of as a lattice-site index here.

Up to now, we have primarily focused on the action of the fermionic operators. This is due to the fact that the gauge operators are quite simple in U(1) LGTs, corresponding to incrementing or decrementing the occupied bosonic mode [see Eq. (9)]. Thus, the gauge operators only determine the locations of the nonzero matrix-element values, via oracle calls to O_F , while oracle calls to O_H will be solely determined by the action of the fermionic operators. We now consider the complexities of constructing these oracles for each of the three Hamiltonian terms.

1. Mass term: H_M

Consider the mass term H_M of Eq. (27), restricted to a geometrically local HHKL block $\mathcal{B}$. For block encoding, the Hamiltonian must be rescaled such that $\|H\|_{\max} \leq 1$, i.e., the maximum matrix-element value is at most $|1|$. This amounts to scaling the Hamiltonian by $(N_B^d/2)^{-1}$. To accomplish this, we will construct oracles for the unnormalized Hamiltonian and at the end of the computation append additional ancilla qubits to the result to bit shift the encoded values.

Since the Hamiltonian H_M is diagonal, the enumerator oracle O_F is implemented via a series of controlled-NOT (CNOT) gates that copy the row index to a new register, as shown in Fig. 4. We omit the bosonic register here for simplicity, as the mass term only acts on the fermionic matter.

In order to implement the matrix-element-value oracle, O_H , we first note that the spectrum of H_M ranges from $-N_B^d/2$ to $N_B^d/2$. Additionally, the staggered fermion

structure of the lattice corresponds to incrementing or decrementing the matrix-element value on alternating sites, depending on the site occupancy. We store a p -bit representation of the matrix-element value with the circuit in Fig. 5. Controlled on the occupancy of the *physical* fermion site, the current matrix-element value is alternately incremented or decremented and the final state of the second register stores a binary representation of this value. Finally, this register must be bit shifted to ensure that $\|H_M\|_{\max} \leq |1|$ but this bit shift can be implemented by introducing a few additional ancillas.

We further optimize the circuit by defining a SID gate. The SID gate acts on signed integers encoded in a binary representation and will either increment or decrement the value of this integer depending on the state of an ancilla qubit. This allows us to reduce the number of increments and decrements by a factor of 2 by first computing the parity of two successive physical fermions in $|f\rangle$. Namely, if the parity is zero, the state remains unchanged and if the parity is one, either an increment or decrement is performed. Incrementing or decrementing is distinguished by whether the first qubit in a successive pair is one or zero, respectively, so by controlling on these registers, the appropriate operation is performed.

The qubit overhead can also be reduced by temporarily storing the parity in the second qubit of a successive

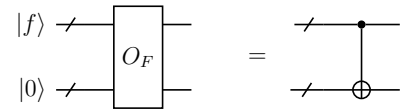


FIG. 4. The implementation of the enumerator oracle O_F for the mass term H_M . Since H_M is diagonal, the row and column indices of the nonzero matrix-element values are equivalent.

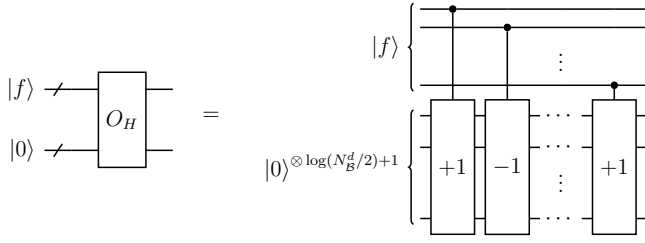


FIG. 5. The naive implementation of the matrix-element-value oracle O_H for the mass term H_M . The staggered fermion model imposes that alternating sites occupied by physical fermions amounts to incrementing (+1) or decrementing (−1) the current matrix-element-value state. The value at the end of the computation is the eigenvalue of the corresponding basis state.

pair, which then acts as the control register of the SID gate. To avoid using separate subroutines for both the increment and decrement operations, note that the decrement circuit can be obtained from the increment circuit simply by bit flipping the input state prior to incrementing, so an additional layer consisting of a multitarget CNOT gate is included. An additional qubit must be included to store the sign of the matrix-element value. After incorporating all of these techniques, the optimized oracle O_H is shown in Fig. 6 and the controlled-SID gate is shown in Fig. 7.

For a register storing the matrix-element value of size $\log(N_B^d/2)$, the controlled-SID gate requires $2 \log(N_B^d/2) + 1$ Toffoli gates. Additionally, the controlled-SID gate must be performed $N_B^d/2$ times. This brings the total T count of O_H to

$$T[O_H] = 2N_B^d(2 \log(N_B^d/2) + 1). \quad (41)$$

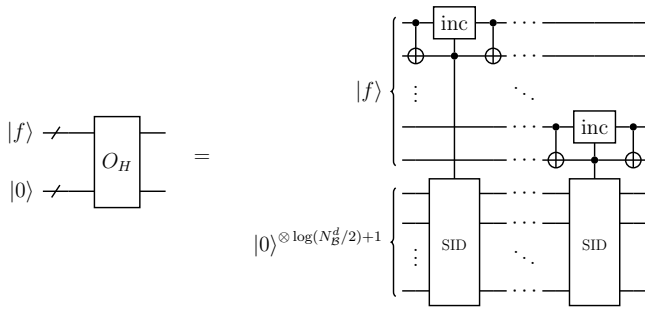


FIG. 6. The optimized implementation of the matrix-element-value oracle O_H for the mass term H_M using the controlled-SID gate. The parity of successive pairs of physical fermions is computed and temporarily stored on the second qubit of the pair (the |ctrl> qubit). If this parity is odd, an increment or decrement is performed depending on the state of the first qubit in a pair (the |inc> qubit). This circuit computes the matrix-element value using half of the calls to an increment or decrement circuit using fewer T gates and only one additional qubit.

2. Gauge-matter term: H_{GM}

For the gauge-matter terms, H_{GM}^h and H_{GM}^v , both the fermionic and bosonic operators must be considered, though each of these operators can be treated independently. Starting with the fermionic part of H_{GM}^v , the symplectic binary vectors a_ℓ of the Pauli strings are classically precomputed according to Eq. (40).

The enumerator oracle O_F is then realized using these symplectic binary vectors, which map a fermionic occupancy state onto at most one other fermionic occupancy state by tracking the action of the Pauli operators. The bosonic operators also map a bosonic state to at most one other bosonic state, the action of which is simply to increment or decrement a corresponding bosonic mode. Finding the correct final states under the action of the Hamiltonian terms will give us the correct locations of the nonzero matrix elements. The classically precomputed binary symplectic vectors are loaded in via a QROM [18], just as the eigenvalues were for the electric field term in Sec. IV A. Using these vectors, the corresponding actions of the fermionic and bosonic operators of Eq. (37) can be constructed in a quantum circuit. The explicit circuit is shown in Fig. 8.

The overall complexity of this circuit will consist of the T counts from both the QROM and the multicontrolled-SID gates. The QROM requires only $4N_B^d - 4$ T gates [18]. To obtain the remaining complexity, we promote each gate in the controlled-SID circuit of Fig. 7 by one level of control. Combining these two results, the associated T count is

$$\begin{aligned} T[O_F] &= (4N_B^d - 4) + 4N_B^d(2 \log \Lambda + 4) \\ &= 8N_B^d \log \Lambda + 20N_B^d - 4. \end{aligned} \quad (42)$$

The matrix-element-value oracle O_H takes a similar form but now relies on both of the classically precomputed symplectic binary representations, a_ℓ and $b_{\ell,j}$. The vectors a_ℓ have already been loaded in as part of the implementation of O_F and we will reuse them here. The vectors $b_{\ell,j}$ are now loaded in via a QROM with two additional qubits in the index register such that a unique vector $b_{\ell,j}$ is loaded for every possible combination of lattice site ℓ and weight-4 Pauli term $j \in [0, 3]$. This amounts to repeating the QROM subroutine 4 times, one for each unique Pauli term in the Hamiltonian of Eq. (30). The matrix-element value of Eq. (39) is then determined by computing the bit-wise AND operation between two sets of registers. The circuit construction is given in Fig. 9.

The T complexity results from a call to the QROM as well as two series of Toffoli gates to compute the bit-wise AND between the two sets of registers. Combining these two pieces, the T count is

$$T[O_H] = 4(4N_B^d - 4) + 3 + 4(4N_B^d) = 32N_B^d - 13. \quad (43)$$

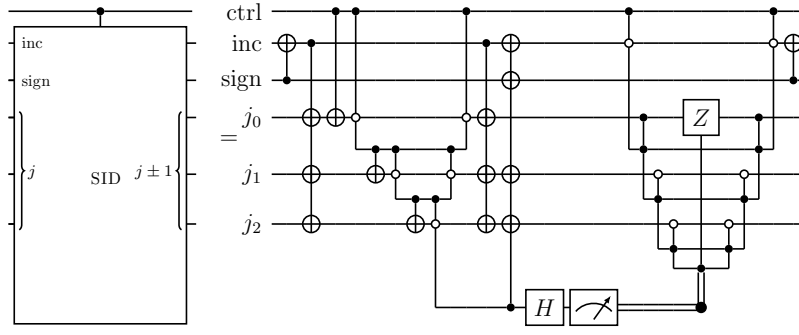


FIG. 7. The controlled-SID gate.

Finally, we note that the oracle implementations for the horizontal gauge-matter term H_{GM}^h will follow this structure exactly and can be implemented using equivalent T

counts. Not surprisingly, this is a result of the similar structure in the terms for both the vertical and horizontal gauge-matter Hamiltonians.

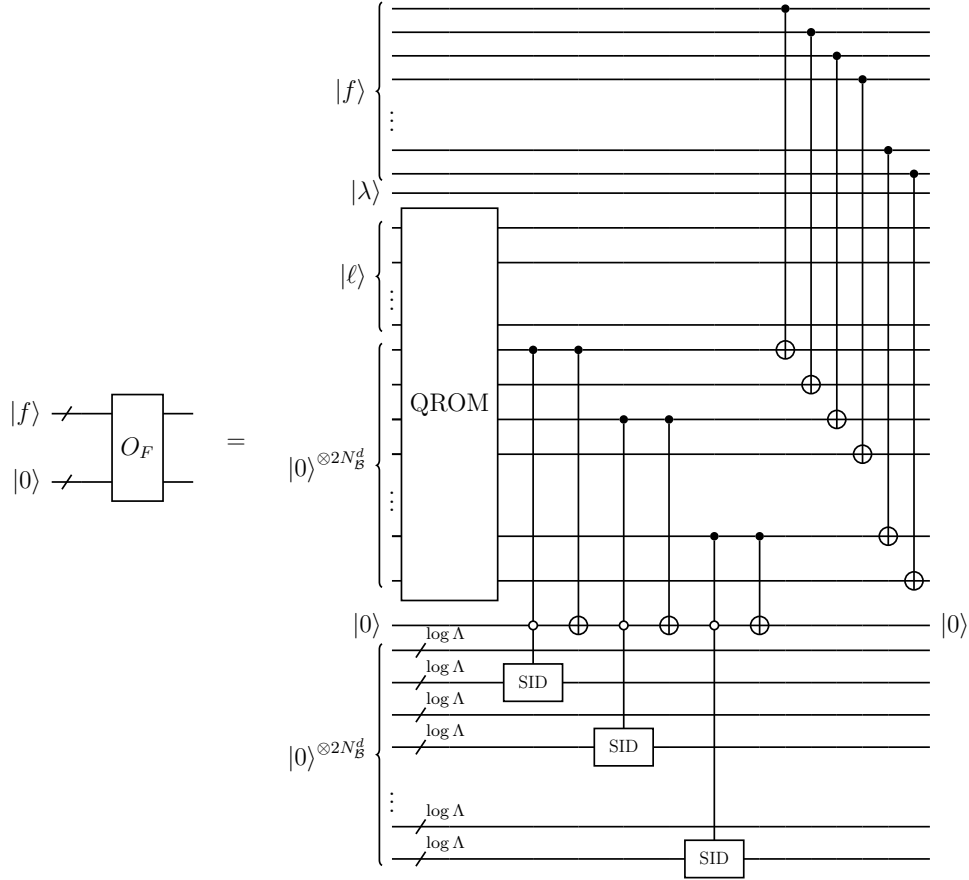


FIG. 8. The implementation of the enumerator oracle O_F for the vertical gauge-matter Hamiltonian H_{GM}^v . First, the QROM loads in the vector a_ℓ controlled on the index register $|\ell\rangle$. Then, the action of the bosonic operators is performed. Based on the ordering imposed by the VC encoding, the first nonzero bit in a_ℓ indicates the physical site that the link originates from and the third nonzero bit indicates the physical site at which the link terminates (the second and fourth nonzero bits act on auxiliary fermions and do not affect the bosonic links). Note that imposing an ordering of the vertical links, namely, that bosonic operators propagate from a starting site downward, eliminates the need for indexing on both the initial and terminal sites, so we need only control on the first nonzero bit of a_ℓ . Thus we control on every other bit of a_ℓ (corresponding to physical sites) of which there will always be two nonzero bits. To avoid performing two decrements, we introduce a flag qubit. If the flag qubit is off, then the decrement is performed on the corresponding bosonic mode λ_ℓ^v (as will be the case for the first nonzero bit in a_ℓ) and a successive CNOT flips the state of the flag qubit such that the second nonzero bit of a_ℓ does not trigger a decrement. Finally, the fermionic action is performed via a series of CNOT gates between $|f\rangle$ and $|a_\ell\rangle$, as given in Eq. (37).

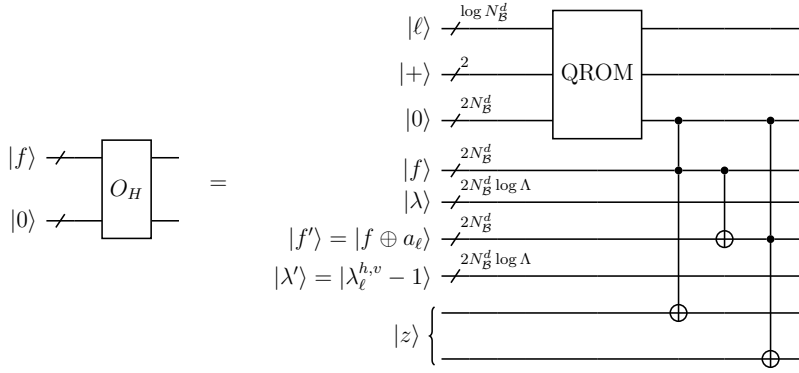


FIG. 9. The implementation of the matrix-element-value oracle O_H for the vertical gauge-matter Hamiltonian H_{GM}^v . First, the same index register $|\ell\rangle$ used in O_F is used to index the QROM, which ensures that a_ℓ and b_ℓ will correspond to the same site ℓ . An additional two qubits are prepared in the $|+\rangle$ state to index the type of Pauli operator acting on site ℓ . Then, the corresponding binary symplectic representation $b_{\ell,j}$ is loaded in. With this vector, the matrix-element value can be computed from $H_{f,f \oplus a_\ell} = (-1)^{b_{\ell,j} \cdot f} (-i)^{a_\ell \cdot b_{\ell,j}}$. To do so, the value $b_{\ell,j} \cdot f$ is computed (for each j) and stored in the first qubit of $|z\rangle$. In order to compute $a_\ell \cdot b_{\ell,j}$, an XOR is performed between f and $f \oplus a_\ell$, returning a_ℓ . Then $a_\ell \cdot b_{\ell,j}$ is computed and stored on the second qubit of $|z\rangle$. The state of $|z\rangle$ is then mapped uniquely to the matrix-element value ± 1 or $\pm i$.

3. Gauge term: H_B

For the magnetic field term H_B in $U(1)$, the trace operation of Eq. (6) can be ignored as there is only one associated color. Each Hamiltonian term is the product of four bosonic operators forming a plaquette. The implementation of the bosonic operators has been discussed in Sec. IV B 2 when considering the gauge-matter interaction, so the circuit for the magnetic field Hamiltonian will just consist of a product of four controlled-SID gates, each acting on their respective bosonic modes.

As mentioned previously, the enumerator oracle O_F is realized by the action of the Hamiltonian terms on the basis states, as this yields the locations of the nonzero matrix elements. Imposing an enumeration scheme such that the lattice-site index ℓ fixes the upper-left vertex of a plaquette uniquely identifies which bosonic modes are acted on by each plaquette term $P_{\square}^{(\ell)}$. Given that $|\ell\rangle$ is a superposition over binary encoded states, the naive way to implement O_F is via a series of multicontrolled gates in which there is a unique operation for every possible binary configuration of the controls. The Clifford+T decomposition for such a circuit is not optimal.

Instead, we make use of the unary-iteration subroutine, which converts a binary encoding into a successive preparation, and then unpreparation, of unary-encoded states that only require operations with a single control [18]. Using unary iteration, each lattice-site index is successively controlled on and the associated plaquette operator $P_{\square}^{(\ell)}$ is applied. This is accomplished via the circuit in Fig. 10.

The complexity of this oracle combines the cost of unary iteration and the cost of implementing four bosonic operators at each lattice site. The full complexity is

$$\begin{aligned} T[O_F] &= 4N_B^d + 4(4N_B^d(2 \log \Lambda + 1)) \\ &= 32N_B^d \log \Lambda + 20N_B^d. \end{aligned} \quad (44)$$

The value oracle O_H for the magnetic field term is trivial in $U(1)$, as all matrix-element values are 1.

4. Resource requirements

Given the explicit sparse-access-oracle constructions, the Hamiltonian interaction terms can be block encoded. This circuit complexity is given in Ref. [63, Lemma 48] and a circuit representation is given in Ref. [65]. In terms of our registers, the block-encoding circuit is shown in Fig. 11.

Since block encoding the Hamiltonian interaction terms is realized via two calls to O_F and O_H , as well as a series of controlled rotations, its full complexity is

$$T[H] = 2T[O_F] + 2T[O_H] + 4p(4 \log(1/\epsilon)), \quad (45)$$

where p is the number of qubits needed to store the matrix-element value and ϵ is the rotation-synthesis error to which a rotation can be approximated in the Clifford+T gate set, using the optimal bound of Ref. [89]. For $U(1)$ gauge theory, $p = \log_2(N_B^d/2)$ for storing the terms in H_M . Then, for each Hamiltonian term, their complexity scales as

$$T[H_M^B] = 4N_B^d(2 \log(N_B^d/2) + 1) + 16 \log(N_B^d/2) \log(1/\epsilon) \simeq 8N_B^d \log(N_B^d/2), \quad (46)$$

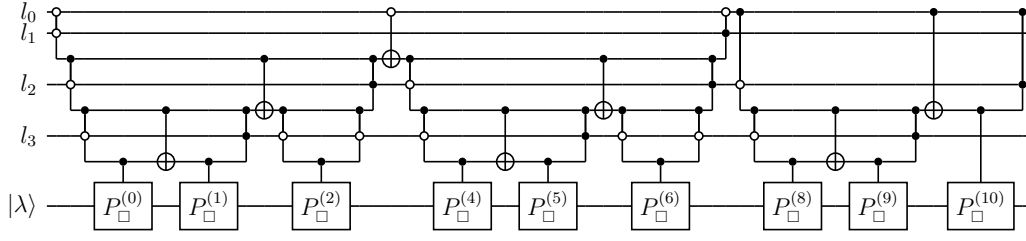


FIG. 10. The implementation of the enumerator oracle O_F for the magnetic field term H_B . The lattice sites ℓ are indexed via unary iteration [18] and the corresponding bosonic operators are applied around the associated plaquette. An example is shown here using the enumeration scheme of the 4×4 lattice in Fig. 1(b). The plaquettes are enumerated by the index of their upper-left lattice sites.

$$\begin{aligned} T[H_{GM}^B] &= 2(4N_B^d - 4 + 4N_B^d(2 \log \Lambda + 4)) + 2(4(4N_B^d - 4) + 3 + 4(4N_B^d)) + 32 \log(1/\epsilon) \\ &\simeq 16N_B^d \log \Lambda + 104N_B^d, \end{aligned} \quad (47)$$

$$\begin{aligned} T[H_B^B] &= 2(4N_B^d - 8 + 8(4N_B^d)(2 \log \Lambda + 1)) + 2N_B(2 \log \Lambda + 1) + 16 \log(1/\epsilon) \\ &\simeq 64N_B^d \log \Lambda + 72N_B^d. \end{aligned} \quad (48)$$

The reason that two calls to O_H are made, along with a series of controlled rotations, is to implement an alternative version of the matrix-element-value oracle,

$$\tilde{O}_H |0\rangle |i\rangle |j\rangle = \left(H_{ij} |0\rangle + \sqrt{1 - |H_{ij}|^2} |1\rangle \right) |i\rangle |j\rangle, \quad (49)$$

which encodes the matrix-element value in the amplitude of the signal qubit. While for U(1) it is possible to remove this overhead by implementing Eq. (49) directly, in general this approach will not be as straightforward. When matrix values are more complicated functions of the system parameters, such as those in SU(2) and SU(3) LGTs, it is more convenient to design an oracle that stores the p -bit representation of the matrix-element value, as we have done, and then convert to the

signal-qubit encoding when block encoding via the circuit in Fig. 11.

C. LCU oracles

Consider the LCU oracles for a Hamiltonian $H = \sum_{\ell=0}^{L-1} \alpha_\ell U_\ell$ with norm $\|H\| \leq \|\vec{\alpha}\|_1 = \sum_{\ell=0}^{L-1} |\alpha_\ell|$, defined in Ref. [62] as

$$\text{PREPARE } |0\rangle = \sum_{\ell=0}^{L-1} \sqrt{\frac{\alpha_\ell}{\|\vec{\alpha}\|_1}} |\ell\rangle, \quad (50)$$

$$\text{SELECT } |\ell\rangle |\psi\rangle = |\ell\rangle U_\ell |\psi\rangle. \quad (51)$$

Since the phases can be absorbed into the unitary operators U_ℓ , we can assume that all coefficients α_ℓ are positive

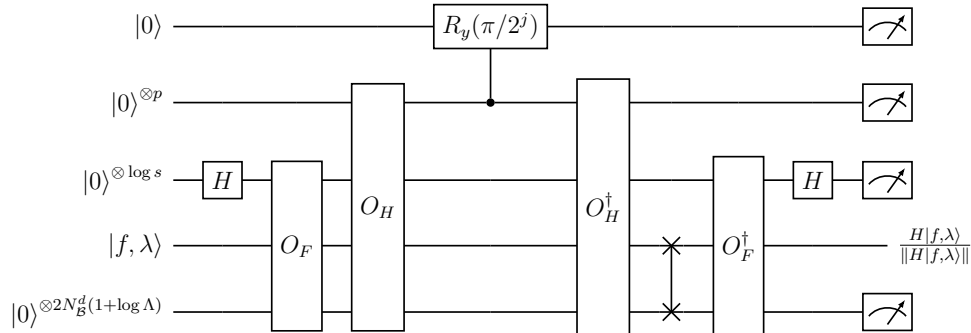


FIG. 11. A circuit to block encode the Hamiltonian via sparse-access oracles [65]. The top qubit is the signal qubit, while the second register stores the matrix-element value to p bits of precision. The third register stores a lattice-site index, the fourth register stores the Hamiltonian matrix row indices, and the fifth register stores the matrix-element column indices. Importantly, this circuit is successful only upon all measurement outcomes being 0.

and real. Thus, PREPARE generates a superposition of index states weighted by the Hamiltonian coefficients α_ℓ and SELECT applies the unitary term U_ℓ controlled on the index state $|\ell\rangle$.

1. Mass term: H_M

For the mass term H_M in $U(1)$, we first rewrite the Hamiltonian of Eq. (27) in a form more suitable to the LCU oracles,

$$H_M = \frac{1}{2} \sum_{\ell=0}^{N_B^d-1} (-1)^{\ell+1} Z(\ell), \quad (52)$$

where the lattice vector indices $x = (k, l)$ have been mapped onto scalar indices ℓ with the ordering imposed by the VC-encoding, much like the indexing scheme for the sparse-access oracles. The phases $(-1)^{\ell+1}$ can all be absorbed into the unitaries such that $\alpha_\ell = 1/2$ for all ℓ . Then, the action of the PREPARE oracle is

$$\text{PREPARE } |0\rangle = \frac{1}{\sqrt{N_B^d}} \sum_{\ell=0}^{N_B^d-1} |\ell\rangle, \quad (53)$$

corresponding to a uniform superposition over the site indices. This circuit is implemented with a single layer of Hadamard gates and thus has a trivial complexity. As we

will see, PREPARE will be trivial for each Hamiltonian interaction term in $U(1)$, making it a desirable encoding model for this problem.

On the other hand, the SELECT oracle is implemented via operations controlled on the index states $|\ell\rangle$. As mentioned previously, rather than performing a series of multiconrolled gates for all possible binary configurations of the index state, we use the unary-iteration circuit of Ref. [18]. Then, the action of SELECT is to apply $\pm Z$ to the physical fermion sites controlled on the unary-encoded index states. This is accomplished via the circuit in Fig. 12. The total T-gate count of the SELECT circuit comes from the unary iteration subroutine, so

$$T[\text{SELECT}] = 4N_B^d - 4 \quad (54)$$

and we obtain an $O(\log N_B^d)$ improvement over the sparse-access encoding of the mass term in Eq. (41).

2. Gauge-matter term: H_{GM}

The vertical gauge-matter term H_{GM}^v of Eq. (30) is also naturally suitable to the PREPARE and SELECT oracles. Again, imposing an enumeration of the lattice sites fixes the locations of the fermions that can be acted on by the gauge-matter term. This then fixes the structure of the associated Pauli operators and the overall phase of the Pauli operator, i.e., the row and column parities are uniquely determined by ℓ under a specified enumeration scheme. Then, the action of the SELECT oracle is

$$\text{SELECT } |A, B, \ell\rangle |\psi\rangle = |A, B, \ell\rangle \otimes \begin{cases} \pm X_\ell \tilde{Y}_\ell X_{\ell'} \tilde{X}_{\ell'} U(\ell'; \ell) |\psi\rangle, & A \wedge B \wedge \ell \text{ is even,} \\ \pm i Y_\ell \tilde{Y}_\ell X_{\ell'} \tilde{X}_{\ell'} U(\ell'; \ell) |\psi\rangle, & A \wedge \neg B \wedge \ell \text{ is even,} \\ \mp i X_\ell \tilde{Y}_\ell Y_{\ell'} \tilde{X}_{\ell'} U(\ell'; \ell) |\psi\rangle, & \neg A \wedge B \wedge \ell \text{ is even,} \\ \pm Y_\ell \tilde{Y}_\ell Y_{\ell'} \tilde{X}_{\ell'} U(\ell'; \ell) |\psi\rangle, & \neg A \wedge \neg B \wedge \ell \text{ is even,} \\ \pm X_\ell \tilde{X}_\ell X_{\ell'} \tilde{Y}_{\ell'} U(\ell'; \ell) |\psi\rangle, & A \wedge B \wedge \ell \text{ is odd,} \\ \pm i Y_\ell \tilde{X}_\ell X_{\ell'} \tilde{Y}_{\ell'} U(\ell'; \ell) |\psi\rangle, & A \wedge \neg B \wedge \ell \text{ is odd,} \\ \mp i X_\ell \tilde{X}_\ell Y_{\ell'} \tilde{Y}_{\ell'} U(\ell'; \ell) |\psi\rangle, & \neg A \wedge B \wedge \ell \text{ is odd,} \\ \pm Y_\ell \tilde{X}_\ell Y_{\ell'} \tilde{Y}_{\ell'} U(\ell'; \ell) |\psi\rangle, & \neg A \wedge \neg B \wedge \ell \text{ is odd,} \end{cases} \quad (55)$$

where ℓ' is a function of ℓ determined *a priori* from the lattice enumeration and the phase of each term also has a known dependence on ℓ . Here, A and B are single-qubit index registers the states of which determine which type of Pauli operators are applied. Much like the mass term, since the phases are absorbed into the Pauli terms, then $\alpha_\ell = 1/4$ for all ℓ . The PREPARE oracle takes the

form

$$\text{PREPARE } |0\rangle = \frac{1}{2\sqrt{N_B}} \sum_{\ell=0}^{N_B^d-1} |+\rangle_A |+\rangle_B |\ell\rangle, \quad (56)$$

which is again just the uniform superposition state and can be implemented trivially.

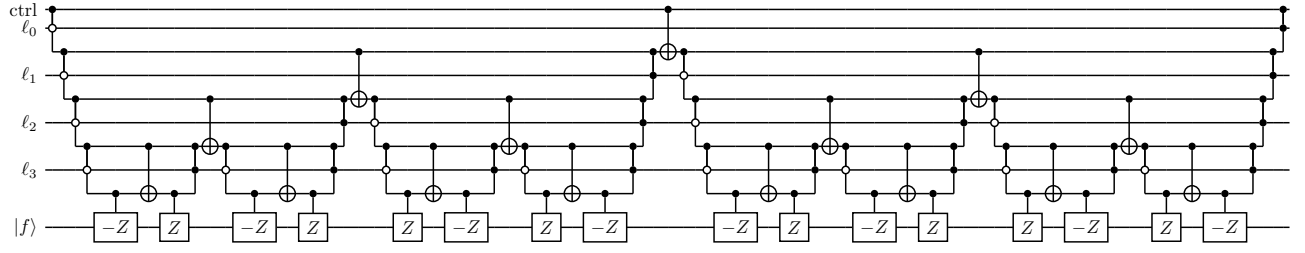


FIG. 12. The implementation of SELECT for the mass term H_M using unary iteration [18]. Controlled on the iteratively produced unary-encoded site indices $|\ell\rangle$, the controlled- $\pm Z$ gates are performed on the corresponding physical fermion sites.

To implement SELECT, unary iteration is again used to successively generate the index states $|\ell\rangle$ and the additional index qubits A and B will not be iterated over in unary iteration but, instead, will index the Pauli operator applied at each site. Thus, unary iteration must be repeated over the lattice sites 4 times, once for each configuration of $|A, B\rangle$. The first iteration of the SELECT oracle is shown in Fig. 13.

The full complexity is once again determined from both unary iteration as well as the cost of implementing the controlled unitary operations at each lattice site. However, while the fermionic unitaries can be implemented trivially, to implement the gauge operator we need to implement a controlled-SID gate once again. The complexity is thus

$$T[\text{SELECT}] = 4(4N_B^d - 4) + 3 + 4N_B^d(2 \log \Lambda + 1) = 8N_B^d \log \Lambda + 20N_B^d - 13. \quad (57)$$

As for the horizontal-hopping term $\hat{H}_{GM}^h$, as given in Eq. (28), it takes the same form as the vertical-hopping term in the LCU formalism. The only difference is that the Pauli strings in the SELECT operation can be of higher weight for SU(2) and SU(3), but are of lower weight for U(1). In either case, these do not contribute any additional T

gates. The total T-gate counts for the combined vertical and horizontal hoppings are double that of the horizontal hoppings and we require one additional control qubit D , which determines the hopping direction—vertical or horizontal.

3. Gauge term: H_B

The magnetic field term H_B for U(1) amounts to decrementing, or incrementing, the bosonic modes on the links forming a plaquette in the lattice, as was the case for the sparse-access oracles. Once again, the lattice structure plays a crucial role in the construction of these oracles, as a site index ℓ will allow us to uniquely determine the plaquette and thus the bosonic modes that need to be modified. In particular,

$$\text{SELECT} |\ell\rangle |\psi\rangle = U^h(\ell + 1; \ell) U^v(\ell + N_B^d + 1; \ell + 1) U^{h\dagger}(\ell + N_B^d; \ell + N_B^d + 1) U^{v\dagger}(\ell; \ell + N_B^d) |\psi\rangle, \quad (58)$$

where $U^h(f, i)$ ($U^v(f, i)$) acts on the horizontal (vertical) links oriented from site i to site f . The circuit for implementing SELECT will actually be identical to the circuit implementing the sparse-access oracle O_F for the magnetic field term in U(1), though an additional control qubit is needed for the LCU implementation.

The PREPARE oracle does not require T gates and only involves Hadamards, as we only need to prepare the uniform superposition over the index register $|\ell\rangle$,

$$\text{PREPARE} |0\rangle = \frac{1}{\sqrt{N_B^d}} \sum_{\ell=0}^{N_B^d-1} |\ell\rangle. \quad (59)$$

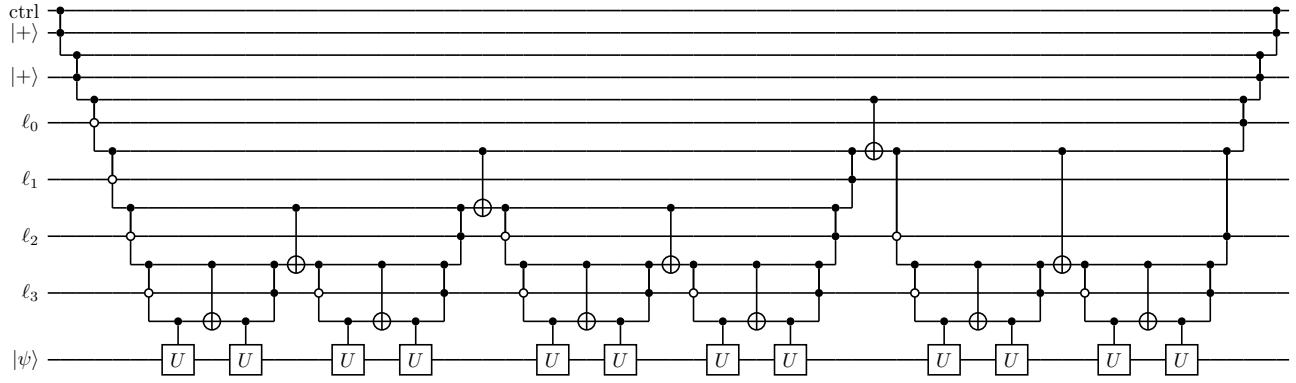


FIG. 13. The partial implementation of SELECT for the gauge-matter term H_{GM} . This example shows the first configuration for $|A, B\rangle = |11\rangle$ and thus will apply $U = \pm X_\ell \tilde{Y}_\ell X_{\ell'} \tilde{X}_{\ell'} U(\ell'; \ell)$ or $U = \pm X_\ell \tilde{X}_\ell X_{\ell'} \tilde{Y}_{\ell'} U(\ell', \ell)$ according to Eq. (55), depending on whether ℓ is an even or odd site. This circuit will then be repeated for the remaining three configurations of $|A, B\rangle$ to apply the rest of the terms.

4. Resource requirements

Block encoding the Hamiltonian using LCU oracles is accomplished via the circuit shown in Fig. 14, with a T count of

$$T[H] = 2T[\text{PREPARE}] + T[\text{SELECT}], \quad (60)$$

such that

$$T[H_M^B] = 4N_B^d - 4 \simeq 4N_B^d, \quad (61)$$

$$T[H_{GM}^B] = 2(4(4N_B^d - 4) + 3 + N_B^d(4(2 \log \Lambda + 1))) \simeq 16N_B^d \log \Lambda + 40N_B^d, \quad (62)$$

$$T[H_B^B] = 4N_B^d - 4 + 8N_B^d(4(2 \log \Lambda + 1)) + 2N_B(2 \log \Lambda + 1) \simeq 32N_B^d \log \Lambda + 36N_B^d. \quad (63)$$

D. Resource comparison

Here, we compute the total resource requirements for a two-dimensional U(1) LGT Hamiltonian simulation using the interaction-picture simulation method, HHKL block decomposition, and the costs for electric field circuits and block-encoding circuits determined in Secs. IV B 4 and IV C 4. We conclude this section by comparing with the cost of Trotterized Hamiltonian simulation developed by Kan and Nam [58].

We first establish the resource requirements for the total block encoding of H_{M+GM+B}^B . To prepare a linear superposition over our three block encodings, H_M^B , H_{GM}^B , and H_B^B , with constants g_M , g_{GM} , and g_B , we use two ancilla registers containing a single-qubit rotation on each ancilla and controlled application of each of the three block-encoded operators therein. The costs for doubly controlled applications of the sparse block encodings are as follows:

$$T[CC - H_M^B] = 16N_B^d + 4N_B^d(2 \log(N_B^d/2) + 5) + 16 \log(N_B^d/2) \log(1/\epsilon), \quad (64)$$

$$T[CC - H_{GM}^B] = 40N_B^d + 2(4N_B^d + 4 + 4N_B^d(2 \log \Lambda + 6)) + 2(4(4N_B^d - 4) + 5 + 32N_B^d) + 32 \log(1/\epsilon), \quad (65)$$

$$T[CC - H_B^B] = 2(4N_B^d - 4 + 32N_B^d(2 \log \Lambda + 1) + 2N_B(2 \log \Lambda + 1)) + 16 \log(1/\epsilon), \quad (66)$$

$$Q = 6N_B^d + 4N_B^d \log \Lambda + 2 \log N_B^d + 2, \quad (67)$$

where Q is the number of qubits required, and similarly for the LCU encodings:

$$T[CC - H_M^B] = 4N_B^d + 4, \quad (68)$$

$$T[CC - H_{GM}^B] = 2(4(4N_B^d + 4) + 3 + 4N_B^d(2 \log \Lambda + 1)), \quad (69)$$

$$T[CC - H_B^B] = 4N_B^d + 4 + 32N_B^d(2 \log \Lambda + 1) + 2N_B(2 \log \Lambda + 1), \quad (70)$$

$$Q = N_B^d + 2N_B^d \log \Lambda + 2 \log N_B^d + 2. \quad (71)$$

The total cost of block encoding is then

$$T[H_{M+GM+B}^B] \equiv T[CC - H_M^B] + T[CC - H_{GM}^B] + T[CC - H_B^B] + 6 \log(1/\epsilon). \quad (72)$$

We note that the addition of the two extra control qubits just yields a constant overhead for LCU due to the presence of unary logic, whereas significant overheads are present for the sparse-access model. This is because the circuits for the sparse-access models involve CNOT gates for logic, which get elevated to multicontrolled Toffoli gates once two additional control lines are added.

We present these resource estimates in Fig. 15, alongside the cost of implementing the fast-forwarded term $e^{i\tau H_E^B}$. We find that the LCU block encoding beats the sparse encoding by a factor of approximately 3 in T counts. Further, the cost of simulating the electric term is an order of magnitude smaller than either block encoding. This would indicate that the LCU encoding significantly outpaces the sparse encoding in practical applications. However, as mentioned in Sec. II on qubitized simulation, the oracle that is queried in the interaction-picture simulation is HAM-T, which requires multiple queries to the electric field term but only a single query to the block encoding. We will see shortly that the overhead of multiple queries eliminates any T-gate reduction from LCU. We additionally find an improvement by a factor of 2–3 in the qubit counts in the LCU block encoding relative to the sparse encoding.

In implementing HAM-T, we note by Ref. [92, Theorem 7] that it is computed using $\lceil \log_2 \mathcal{M} \rceil$ queries to controlled applications of $e^{iH_E^B/(2\alpha)}$, where $\alpha = \max_{s \in [0, T]} \|H_I^B(s)\|$ defined in Eq. (17), and a single application of the block encoding $T[H_{M+GM+B}^B]$. Here, $\mathcal{M} = 16(\alpha + \alpha_E)/\epsilon$, where $\alpha_E = g_E \|H_E^B\|$. Noting the coefficients in the Hamiltonian, we can express the norms as $\alpha = 2(g_M + g_{GM} + |g_B|)N_B^d$, $\alpha_E = 2g_E \Lambda^2 N_B^d$. The total cost for HAM-T is then

$$T[\text{HAM-T}^B] = T[e^{-iH_E^B}] \log_2 \frac{16(\alpha + \alpha_E)}{\epsilon} + T[H_{M+GM+B}^B]. \quad (73)$$

This requires an additional $\lceil \log_2 \mathcal{M} \rceil$ qubits.

We then make use of Ref. [92, Lemma 6] to obtain the T-gate cost for the simulation of e^{iH^B} from HAM-T. In total, we will require α segments of simulation, each of which require one call to $e^{iH_E^B/(2\alpha)}$ and $3K$ calls to HAM-T where $K = \lceil -1 + \{2 \ln(2\alpha/\epsilon)/[\ln \ln(2\alpha/\epsilon) + 1]\} \rceil$. No additional qubits are required. The total simulation cost for a single HHKL block to unit time is

$$T[e^{iH^B}] = \alpha \left(T[e^{iH_E^B}] + \lceil -1 + \frac{2 \ln(2\alpha/\epsilon)}{\ln \ln(2\alpha/\epsilon) + 1} \rceil T[\text{HAM-T}^B] \right). \quad (74)$$

The final cost is then just a multiplicative factor of the above expression:

$$T_{\text{qubitization}} = \frac{N^d T}{N_B^d} T[e^{iH^B}]. \quad (75)$$

The only remaining unknown parameter is the size of the HHKL blocks, which is determined by the Lieb-Robinson velocity of the system and the block-simulation time, namely, $N_B = v_{\text{LR}}$. We document our approach to bounding the Lieb-Robinson velocity in Appendix B.

For our direct comparison to Trotterization, we take the physical parameters cited in Ref. [58] for the case

of simulating heavy-ion collisions on a two-dimensional U(1) gauge theory: $a = 0.1$, $N = 100$, $T = 10$, $\Lambda = 10$, and $m = g = 10$. The Lieb-Robinson velocity and the subsequent block-size calculation yield $N_B \geq 53$. We consider $\epsilon = 10^{-8}$, as per Kan and Nam, but also include comparison with lower-accuracy simulations $\epsilon = 10^{-1}, 10^{-3}, 10^{-5}$, as the accuracy scaling is worse for the Trotterization than for qubitized simulation. The simulation costs are shown in Fig. 16.

Qubitized simulation outperforms Trotterization for essentially all accuracies, with the crossover in costs occurring at very low accuracies $\epsilon < 10^{-1}$. Both qubitized methods perform between 2 and 4 orders of magnitude

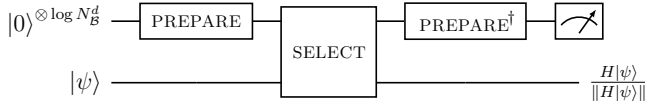


FIG. 14. The block-encoding circuit in the LCU model [65,91].

better in T-gate counts, with larger improvements at higher accuracies. However, we note that there is practically no difference between LCU and sparse implementations of the block encoding: this seems contradictory to our previous results for the block encoding, showing approximately 3 times improvements in LCU over sparse block encodings. The closing of the gap, *per se*, can be understood by looking at HAM-T, which requires $\log_2(16(\alpha + \alpha_E)/\epsilon) \sim 50$ applications of the electric field term, which becomes comparable in cost to the block encodings themselves. As such, the improvement in the total cost of implementing HAM-T, the primary query oracle, is diminished, as the electric field implementation is agnostic to the block-encoding method.

We also report the computational volume, defined as a the T-gate count multiplied by the logical qubit counts. We see that there are additional improvements here for the qubitized algorithm, where the Trotter model requires 5×10^5 qubits, while the sparse and LCU models, respectively, take 1×10^5 and 5×10^4 qubits. Overall, we see an improvement by 6 orders of magnitude in computational volume between the Trotterized and qubitized methods. We have also included a qubitized simulation without using HHKL, where $N_B = N$. We find that this naive qubitized simulation outperforms Trotterization and is slightly worse than HHKL, as expected since we have only four HHKL blocks in this parameter range.

For a richer comparison between the techniques, we also compare the resource estimates for different values of other physical parameters. The qubitized algorithm purports to be optimal in space-time volume, wherein we may look at different T and N . The case for T is quite straightforward, so we exclude it. As for N , the case is more complex, since the real benefit of the qubitized approach is when the HHKL block size is sufficiently small, becoming polylogarithmic in N . It is clear that for the above parameters, this may not be the case and so we may not be getting the greatest benefits of the technique.

We also vary g , m , and a , as they affect v_{LR} , as well as the commutators for Trotterization. We find, however, that changing g and m does not significantly affect v_{LR} ; e.g., a variation by 2 orders of magnitude in either takes $v_{LR} \geq 53 \rightarrow v_{LR} \geq 42$. The values do change the resource estimates; e.g., changing $m = g = 10$ to $m = g = 0.1$ increases the T-gate counts in Trotterization, almost independently of all other parameter variations, by 3 orders of magnitude. However, the same applies for qubitization, meaning that the relative improvement in using either method is nearly unchanging with g and m . The remaining parameter is a , which highly influences v_{LR} and also the commutators for Trotterization.

We report the required resources in terms of T-gate counts and computational volume, as well as the relative improvement for qubitization over Trotterization in space-time volume in Table I at a fixed $m = g = 10$, while varying $\epsilon \in \{10^{-3}, 10^{-1}\}$, $N \in \{100, 1000\}$, and $a \in \{10^{-2}, 10^{-1}, 1\}$. For the sake of readability, we have only included the LCU results here. We see a general trend, namely, that increasing a is the primary improvement, indicating that Trotterization is preferred for small a . Further, as ϵ decreases, the improvements increase, indicating that

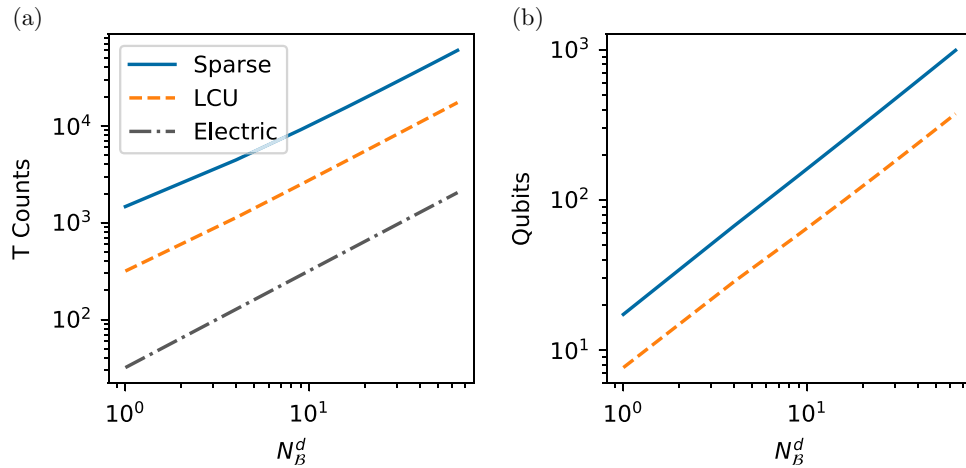


FIG. 15. The total (a) T counts and (b) qubit counts needed to block encode the Hamiltonian. The LCU model achieves an approximately 3 times improvement over the sparse model in the T count and an approximately 3 times improvement in qubit count. Here, we present results for $\Lambda = 5$ and $\epsilon = 10^{-3}$.

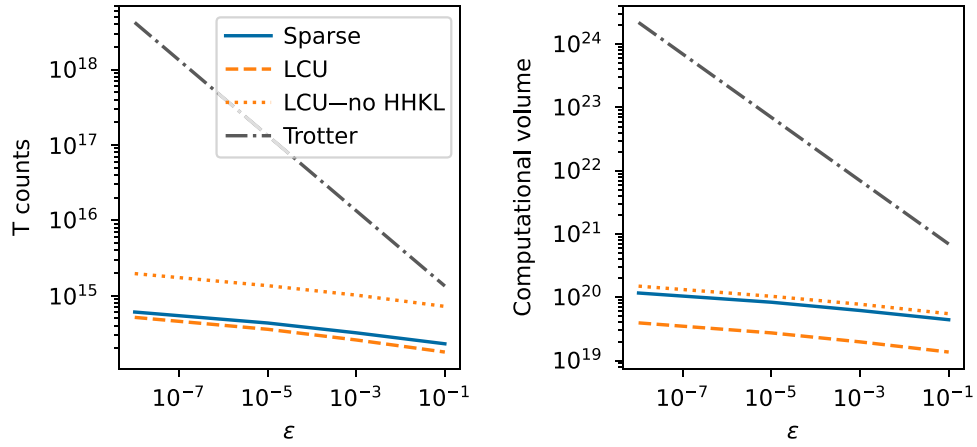


FIG. 16. A comparison of the resource estimates for Hamiltonian simulation of a two-dimensional U(1) lattice gauge theory using the methods in this work, labeled “LCU” and “sparse,” and from the Trotterized simulation of Kan and Nam [58] for the case of heavy-ion collision. (a) T-count comparisons. (b) The computational volume, defined as the T counts multiplied by the qubit counts. Costs are shown for different values of accuracy of the simulation, ϵ . Here, we take the parameters given by Kan and Nam ($a = 0.1$, $N = 100$, $T = 10$, and $\Lambda = 10$) and $m = g = 10$.

at high accuracies, the qubitized simulation outperforms Trotterization.

We note, however, that the expected improvements in qubitization with respect to N are not seen and an opposite trend is observed. Second-order Trotterization scales as $\mathcal{O}(N^{3/2})$, naive qubitization as $\mathcal{O}(N^2)$, and HHKL as $\mathcal{O}(N)$, up to logarithmic factors. For qubitization, the $\mathcal{O}(N^2)$ scaling is a product of the $\mathcal{O}(N)$ Hamiltonian norm appearing as a factor in query complexity and the $\mathcal{O}(N)$ block-encoding cost. Therefore, the HHKL scaling is only $\mathcal{O}(N)$, *asymptotically*, in the regime in which the sizes of blocks simulated via qubitization are much smaller than the system size, which is *not* the case for U(1). As such, we see that the U(1) scaling is closer to the naive qubitization scaling and hence scales worse with N than Trotterization.

If one were able to find tighter bounds for v_{LR} , this might not be the case.

As it stands, we find that the Trotterization scheme of Kan and Nam [58] will outperform our qubitized implementation for simulations with asymptotically small a , large ϵ , and large N . Physically, a dictates the length scale of physical interest and N the system size. While we do not comment on the quantitative regimes in which qubitization outperforms Trotterization, we note that resources pertaining to a particular simulation task can be computed from our resource-estimate expressions. A quantitative analysis would be of particular interest to U(1) gauge theory since in 3+1 dimensions (3+1D) and 2+1D, U(1) has massless photon interactions, for which finite-size corrections scale only polynomially with the system size [93], in contrast

TABLE I. A comparison of Trotterization- and HHKL-based simulation using an LCU block encoding for U(1) lattice gauge theory in two dimensions for a heavy-ion simulation. Various accuracies ϵ , linear system sizes N , and lattice spacings a are shown. Bounds on the Lieb-Robinson velocity v_{LR} , T-gate and qubit counts are derived and a final improvement in space-time volume of the computation (T counts times logical qubit counts) is reported.

ϵ	N	a	v_{LR}	T_{Trotter}	$T_{\text{qubitization}}$	Q_{Trotter}	$Q_{\text{qubitization}}$	Improvement
10^{-3}	10^2	10^0	1.0×10^1	3.2×10^{15}	4.7×10^{12}	5.1×10^5	8.1×10^4	4.4×10^3
		10^{-1}	5.3×10^1	1.3×10^{16}	2.6×10^{14}	5.1×10^5	7.6×10^4	3.5×10^2
		10^{-2}	4.2×10^2	1.1×10^{17}	7.8×10^{15}	5.1×10^5	7.7×10^4	9.8×10^1
	10^3	10^0	1.0×10^1	3.2×10^{18}	4.7×10^{14}	5.1×10^7	8.1×10^6	4.4×10^4
		10^{-1}	5.3×10^1	1.3×10^{19}	2.6×10^{16}	5.1×10^7	7.7×10^6	3.5×10^3
		10^{-2}	4.2×10^2	1.1×10^{20}	1.7×10^{19}	5.1×10^7	7.6×10^6	4.4×10^1
10^{-1}	10^2	10^0	1.0×10^1	3.2×10^{14}	3.1×10^{12}	5.1×10^5	8.1×10^4	6.9×10^2
		10^{-1}	5.3×10^1	1.3×10^{15}	1.8×10^{14}	5.1×10^5	7.7×10^4	5.1×10^1
		10^{-2}	4.2×10^2	1.1×10^{16}	5.6×10^{15}	5.1×10^5	7.7×10^4	1.4×10^1
	10^3	10^0	1.0×10^1	3.2×10^{17}	3.1×10^{14}	5.1×10^7	8.1×10^6	6.9×10^3
		10^{-1}	5.3×10^1	1.3×10^{18}	1.8×10^{16}	5.1×10^7	7.7×10^6	5.1×10^2
		10^{-2}	4.2×10^2	1.1×10^{19}	1.3×10^{19}	5.1×10^7	7.6×10^6	6.0×10^0

to $SU(N)$ theories, which scale exponentially. As such, the quantitative assessment of “small a , large N ” would be particularly important in determining the benefit of qubitization over Trotterization for $U(1)$ LGTs.

V. GENERALIZATION TO NON-ABELIAN GAUGE THEORIES AND HIGHER DIMENSIONS

The explicit circuits for block encoding the Kogut-Susskind Hamiltonian that we have provided can be readily adapted to higher space-time dimensions and higher-order non-Abelian LGTs, such as $SU(2)$ and $SU(3)$. While the VC encoding [85] is sufficient for $U(1)$ LGTs in two dimensions, it is more resource efficient to consider the GS [84] encoding for $SU(2)$ and $SU(3)$ in higher dimensions. The advantage of the GS encoding is that it is easily amenable to changing lattice geometry, number of colors, and spatial dimension and uses only one extra qubit per lattice site, in contrast to the $\mathcal{O}(n_c)$ extra qubits required in the VC encoding. Additionally, its structure and ordering follow closely that of the VC encoding, so the circuit constructions that we employ will follow the same logic.

Using the GS construction, we have already seen that considering a d -dimensional cubic lattice on N^d sites for simulating an $SU(n_c)$ LGT will require $(d + n_c - 1)N_B^d$ qubits to store the fermionic degrees of freedom. Unlike fermions, the bosonic interactions are always local, so there is no need to consider any special form of encoding. Instead, the bosonic encoding will scale directly in terms of the spatial dimension and the specific LGT. For d spatial dimensions, each lattice site will store d bosons, corresponding to the d ordered edges, and for $SU(n_c)$, each bosonic state will have $n_c^2 - 1$ registers, each of size at most $\log \Lambda$, indexing the local quantum numbers. Thus, the total number of qubits needed to store the wave function is

$$Q = (d + n_c - 1)N^d + dN^d(n_c^2 - 1) \log \Lambda. \quad (76)$$

Rather than providing full resource estimates for both the sparse- and LCU-encoding models as for the $U(1)$ case, here we restrict ourselves to providing resource requirements in only the sparse-access model, for three reasons. First, as noted in the $U(1)$ case, the difference in resource requirements between the sparse and LCU models for full simulation is insignificant with respect to the dominant electric field term, making either encoding model a valid candidate. Second, the gauge-field operators $U_{ab}(\vec{n}, \hat{l})$ are conventionally written in a form suitable to the sparse-access model [see Eqs. (11) and (13)]. In the LCU model, one must determine a sequence of elementary gates to implement these operators directly on basis states, which is not only nontrivial but will be unique to each LGT considered and thus does not generalize well to $SU(n_c)$. We remark that a possible implementation of the gauge-field operator could be realized via a modification to the

Clebsch-Gordan transform [94]. One would need to adjust the quantum circuit to transform both the left and right ends of a lattice link and include the dimensional normalization prefactor. Additionally, the generalization of the Clebsch-Gordan transform to $SU(n_c)$ relies on using qudits, so one would need to construct the Clebsch-Gordan transform for $SU(3)$ on qubits. Third, provided that one could find an implementation for SELECT, then the resource requirements could be determined straightforwardly from the expressions given in the $U(1)$ case, as the complexity of the unary-iteration subroutine would scale directly with the spatial dimension and number of colors in the LGT. One would only need to account for the cost of implementing the controlled- U_{ab} gates.

The logic of our algorithm remains relatively unchanged, with only modifications to the operators being applied in the corresponding LGT. The mass term generalizes straightforwardly, scaling with the number of sites and colors. In the sparse-access model, only O_H contributes to the complexity scaling, which requires a controlled-SID gate for each lattice site and each color DOF. Additionally, the size of the registers being incremented or decremented will scale with the type of LGT. Thus, for $SU(n_c)$ in d spatial dimensions, we have

$$T[O_H(H_M)] = 4n_c N^d (2 \log(N^d + n_c - 2) + 1). \quad (77)$$

The remaining Hamiltonian terms all have contributions from the gauge-field operators, which will depend more strictly on the specific LGT being considered. At this point, we will restrict to $SU(2)$ and $SU(3)$ LGTs. The simplest of the gauge terms is the electric field term, which again will be fast-forwarded. In $SU(2)$, the Casimir operator is given by Eq. (10) and in $SU(3)$ by Eq. (12).

The cost of implementing these operators can again be determined using the circuit for fast-forward evolution in Fig. 2. For $SU(2)$, either the QROM or Karatsuba integer multiplication can be used, as before, to implement a unitary U_k that prepares the eigenvalues of the Casimir operators, depending on the bosonic truncation parameter Λ . Here, we provide resource estimates using a QROM, as it outperforms integer multiplication for physically relevant values of Λ .

On the other hand, for $SU(3)$, there are now two representation indices characterizing the action of the Casimir operator, meaning that a QROM will scale as $\mathcal{O}(\Lambda^2)$. The alternative approach of implementing the integer arithmetic on the fly will also require more work. In addition to the Karatsuba multiplication algorithm [88], we also require algorithms for integer addition [95] and integer division [96]. There is again a trade-off in optimal complexity depending on Λ . Numerically, for $\Lambda \lesssim 90$, the QROM is more efficient. The associated complexities are then given by

$$\text{SU}(2): T[e^{-itH_E^B}] = 2(4\Lambda - 4) + 4 \log(\Lambda(\Lambda + 1)) \log(1/\epsilon), \quad (78)$$

$$(\text{QROM}) \text{SU}(3): T[e^{-itH_E^B}] = 2(2\Lambda^2 - 4) + 4 \log(\Lambda(\Lambda + 3)) \log(1/\epsilon), \quad (79)$$

$$(\text{arithmetic}) \text{SU}(3): T[e^{-itH_E^B}] = 2(99 \log^2(\Lambda(\Lambda + 3)) - 60 \log(\Lambda(\Lambda + 3)) - 16) + 4 \log(\Lambda(\Lambda + 3)) \log(1/\epsilon). \quad (80)$$

The oracle constructions for the gauge-field operators $U_{ab}(\vec{n}, \hat{l})$ are also required for both the gauge-matter and magnetic field interaction terms. For SU(2), the gauge-field operators are given in Eq. (11) and for SU(3), in Eq. (13).

In the sparse-access model, we generalize the bosonic part of the circuit constructions for O_F and O_H in Figs. 8 and 9. The fermionic part of the gauge-matter interaction will follow the same logic as in the U(1) case. A QROM will load in the binary symplectic vectors associated with the particular Pauli representation chosen. Then, the nonzero final states are given by $|f \oplus a_\ell\rangle$, just as before. The only difference is that now, in addition to their being a unique symplectic vector for every site, there will also be a unique vector for every color configuration, meaning that the QROM contains $n_c^2 N_B^d$ elements in the database.

In order to implement the gauge-field operators, though, we must be a bit more careful than in the U(1) case. For the U(1) LGTs, we were able to implement the gauge-field operators directly onto the bosonic registers, as they corresponded to increments or decrements of occupancies of individual bosonic modes. On the other hand, for SU(2) and SU(3), the gauge-field operators require preparing a superposition of final states.

For our particular method, we make use of a space-time trade-off. Rather than storing a superposition of final states on a single register, we store each final state in its own register, increasing the spatial resource requirements by a constant-prefactor overhead. The number of additional registers will depend on the specific LGT, but for SU(2), one additional register is required and for SU(3) (in the worst case), 11 additional registers are required, each of size $O(\log \Lambda)$. With this approach, computing the final states amounts to performing only controlled increments and controlled decrements on the state indices. Additionally, with each nonzero final state stored in its own register, then the unique matrix-element values for each state can be computed in O_H on these separate registers.

While we make use of the approach involving additional state registers for computing resource requirements, we also provide some alternative, less intuitive, approaches

that one could employ. First, rather than storing each entire final state, we can include an $O(1)$ size register of indicator qubits that determine the final state that we are in. In SU(2), for example, the action of the gauge-field operator would map $|j\rangle \mapsto \{|j\rangle|0\rangle, |j\rangle|1\rangle\}$, where the state of the indicator qubit corresponds to the states $|j \pm 1/2\rangle$. However, since computing the matrix-element values in O_H requires arithmetic, the explicit circuit construction would require additional controlled operations on the indicator qubit to prepare the correct binary representation of the state prior to doing the arithmetic. Another alternative approach applicable only to SU(2) involves decomposing the gauge field operators into sums of one-sparse terms, which are easy to encode via the sparse-access oracles [90]. For the magnetic field term H_B , though, this amounts to exchanging the sums of the gauge-field operators and the trace operation over the plaquettes, which is only possible in SU(2), as the isospin parameters in SU(3) are color dependent.

Using the approach to store each entire final state, for the bosonic part in SU(2) the nonzero final states $|J, M_L, M_R\rangle$ are constructed via a series of increments or decrements, similar to the U(1) case. However, for SU(2), we now need six increments or decrements for each gauge-field operator with fixed color indices, three for each of the $|j \pm 1/2\rangle$ final states. Thus, in Fig. 8, we can replace each controlled-SID gate with six controlled-SID gates (three to each of the two registers storing the two final states). The rest of the logic will follow exactly as in the U(1) case.

For SU(3), the generalization is similar. The isospin and hypercharges will be incremented or decremented according to the fixed color indices. However, unlike SU(2), the SU(3) gauge-field operators have sums over T'_L and T'_R in addition to the sum over the representation indices. While T'_L and T'_R are color dependent, $t_L, t_R \in \{0, 1/2\}$, so in the worst case, we obtain at most four additional nonzero final states. Thus, there are a total of 12 nonzero final states, each one requiring either seven or eight controlled-SID gates. So each controlled-decrement gate in Fig. 8 will now be replaced with 88 controlled-SID gates. The resulting complexities for the gauge-matter and magnetic field term then become

$$\text{SU}(2): T[O_F(H_{GM})] = 4(4N^d) - 4 + 96N^d(2 \log \Lambda + 4), \quad (81)$$

$$\text{SU}(3): T[O_F(H_{GM})] = 4(9N^d) - 4 + 3168N^d(2 \log \Lambda + 4). \quad (82)$$

While the implementation of O_H was trivial for the gauge fields in U(1), it is not so for SU(2) and SU(3) and we must include additional subroutines to the circuit in Fig. 9 to obtain the bosonic matrix-element values. In both SU(2) and SU(3), this amounts to computing the Clebsch-Gordan coefficients and the dimension-normalization prefactor. While each of these amplitudes is known *a priori*, allowing us to use a QROM to classically precompute and load these values in, we will see that it is in fact more efficient to do the arithmetic on the fly. For SU(2), there are four Clebsch-Gordan coefficients $c_{\Delta j, \Delta m}$ indexed by the change in angular momentum $\Delta j = J - j$ and the change in the z component of the angular momentum, $\Delta m = M - m$. However, each of these coefficients is a function of j and m . Since $j \in [0, \Lambda]$ (with half-integer steps) and $m \in [-\Lambda, \Lambda]$, there are $16\Lambda^2$ elements in the classical database. On the other hand, each of the Clebsch-Gordan coefficients is a simple function of j and m and can be readily computed

using the integer-arithmetic subroutines from the electric field term in addition to a subroutine for computing the square root [97]. The dimension-normalization prefactor can also be readily computed in this way.

For SU(3), the same logic applies, though now the arithmetic is a bit more involved, but still more efficient than implementing a QROM for a database of size $\mathcal{O}(\Lambda^5)$. The SU(3) Clebsch-Gordan coefficients can be computed from the product of the isoscalar factors I_{ab} and the SU(2) Clebsch-Gordan coefficients, with Δj and Δm replaced by ΔT and ΔT^z . In contrast to SU(2), there are now 15 Clebsch-Gordan coefficients, each of which is a function of the representation indices p and q , the isospin T , the hypercharge Y , and the z component of the isospin, T^z (for a more detailed review, see Ref. [58, Appendix D]).

Including these additional costs in the circuits of Fig. 9, we obtain the resource requirements for the gauge-matter term of the Hamiltonian in both SU(2) and SU(3):

$$\text{SU(2): } T[O_H(H_{GM})] = 4(4N^d) - 4 + 3 + 8(d+1)N^d + 2(71 \log^2 \Lambda - 16 \log \Lambda - 32), \quad (83)$$

$$\text{SU(3): } T[O_H(H_{GM})] = 4(9N^d) - 4 + 3 + 8(d+2)N^d + 12(263 \log^2 \Lambda - 28 \log \Lambda - 164). \quad (84)$$

Finally, the magnetic term is realized as the product of four gauge-field operators around a plaquette in the lattice, with the additional constraint that the color indices contract to singlets at each vertex, denoted by the trace operation. Thus the nonzero final states will be given by the tensor product of incremented or decremented basis

states resulting from the action of the gauge-field operators. The complexity is 4 times as costly as implementing the encoding oracle for a single gauge-field term and we can use unary iteration to index over the lattice plaquettes. The number of plaquettes in a d -dimensional hypercubic lattice is

$$N_{\square} = \binom{d}{2} N^2 (N+1)^{d-2}. \quad (85)$$

The associated complexities are

$$\text{SU(2): } T[O_F(H_B)] = 4(2^4 N_{\square}) - 4 + 384N^d(2 \log \Lambda + 1), \quad (86)$$

$$\text{SU(2): } T[O_H(H_B)] = 8(71 \log^2 \Lambda - 16 \log \Lambda - 32), \quad (87)$$

$$\text{SU(3): } T[O_F(H_B)] = 4(3^4 N_{\square}) - 4 + 12 \cdot 672 N^d(2 \log \Lambda + 1), \quad (88)$$

$$\text{SU(3): } T[O_H(H_B)] = 48(263 \log^2 \Lambda - 28 \log \Lambda - 164). \quad (89)$$

With each of the oracle complexities, we can give the full block-encoding complexities for each Hamiltonian term. Since we are actually interested in a block encoding of $H_M + H_{GM} + H_B$, we need two additional levels of control

on each individual encoding circuit as well as two single-qubit rotations to implement an LCU encoding of our block encodings. The doubly controlled block-encoding complexities for SU(2) are

$$T[CC - H_M] = 32N^d + 16N^d(2\log(N^d) + 5) + 16\log(N^d)\log(1/\epsilon), \quad (90)$$

$$T[CC - H_{GM}] = 48(d+1)N^d + 2[4(4N^d) + 4 + 96N^d(2\log\Lambda + 8) + 4(4(4N^d) - 4) + 5 + 24(d+1)N^d + 2(684\log^2\Lambda - 432\log\Lambda + 16)] + 32p\log(1/\epsilon), \quad (91)$$

$$T[CC - H_B] = 2[4(2^4N_\square) + 4 + 384N^d(2\log\Lambda + 1) + 8(684\log^2\Lambda - 432\log\Lambda + 16)] + 16p\log(1/\epsilon) \quad (92)$$

and those for SU(3) are

$$T[CC - H_M] = 48N^d + 24N^d(2\log(N^d + 1) + 5) + 16\log(N^d + 1)\log(1/\epsilon), \quad (93)$$

$$T[CC - H_{GM}] = 48(d+2)N^d + 2[4(4N^d) + 4 + 12\,672N^d(2\log\Lambda + 8) + 4(4(9N^d) - 4) + 5 + 24(d+2)N^d + 12(2988\log^2\Lambda - 2028\log\Lambda + 148)] + 32p\log(1/\epsilon), \quad (94)$$

$$T[CC - H_B] = 2[4(3^4N_\square) + 4 + 384N^d(2\log\Lambda + 1) + 48(2988\log^2\Lambda - 2028\log\Lambda + 148)] + 16p\log(1/\epsilon), \quad (95)$$

where $p = 64$ corresponds to double-precision accuracy for the matrix-element values.

In Fig. 17, we show a comparison of the different components of HAM-T, similar to Fig. 15, for SU(2) and SU(3). We see that for SU(2), the block-encoding cost is about 25 times the cost of the electric field term, in concordance with the U(1) calculations. This means that for HAM-T, the electric field term still dominates and hence the method of block encoding is not particularly important. Proceeding with only the sparse encoding will still yield good resource estimates. As for SU(3), the ratio is closer to 1000 and so the block encoding is finally the dominant cost in HAM-T. Accordingly, improvements in block-encoding techniques, such as LCU, may yield improvements in SU(3) simulation relative to the numbers that we will report with our sparse encodings.

In Tables II and III, we provide resource comparisons for three-dimensional simulations of SU(2) and SU(3) LGTs between our qubitized simulation, without HHKL and using the sparse block encoding, to the Trotterized implementation of Kan and Nam [58]. We find for SU(2), at a minimum, an improvement in the computational volume by 10 orders of magnitude and for SU(3), a minimum improvement by 23 orders of magnitude.

We remark that the main contribution to the large improvement over the results of Ref. [58] is a result of the particular representation used for the gauge-field operators U_{ab} . Kan and Nam map the explicit gauge-field terms of Eqs. (11) and (13) to a block-diagonal representation of the operator that consists of approximately 10^{12} terms for SU(2) and approximately 10^{21} terms for SU(3). This operation dominates their complexity, as the number of terms dictates the number of calls to their arithmetic oracle but allows them to exponentiate the gauge-field operators for Trotterization. We do not encounter this problem using the sparse-access block-encoding model, because

the sparse-access oracles do not require any structure on the individual Hamiltonian terms, so we do not need a representation in which individual terms are Hermitian or unitary. As discussed in Secs. II B and II C, the overhead of the block-diagonalization procedure of Ref. [58] is largely related to the exponential scaling with respect to n_c , while the sparse-encoding model only has a polynomial dependence on n_c .

VI. CONCLUSION AND OUTLOOK ON THE SIMULATION OF LGTs

In this work, we have carried out a careful analysis of qubitized simulation of LGTs. We have built on and fully developed the qubitized Hamiltonian-simulation algorithm proposed by Tong *et al.* [60] for U(1), SU(2), and SU(3) gauge theories with matter fields in arbitrary spatial dimension. In doing so, we have worked out the details of various locality-preserving mappings of these models onto qubits [84,85,98], constructed block encodings for the corresponding qubit Hamiltonians based on sparse and LCU oracles, and used the block encodings to construct a full front-to-back analysis of the qubitized simulation algorithm, including estimates of Lieb-Robinson velocities for U(1) gauge theory and a novel SID circuit.

Additionally, we have carried out resource estimates (T-gate counts and qubit counts) for the entire simulation cost within the interaction picture and compared our numbers to those obtained in Ref. [58], wherein Trotterized time evolution has been considered. For the case of simulating the SU(3) model in three spatial dimensions, our algorithm leads to computational-volume requirements that are 25 orders of magnitude lower compared to Trotterization. A key ingredient for reaching such an improvement has been leveraging the sparsity of color gauge-field operators, which is possible in both the sparse and LCU models of

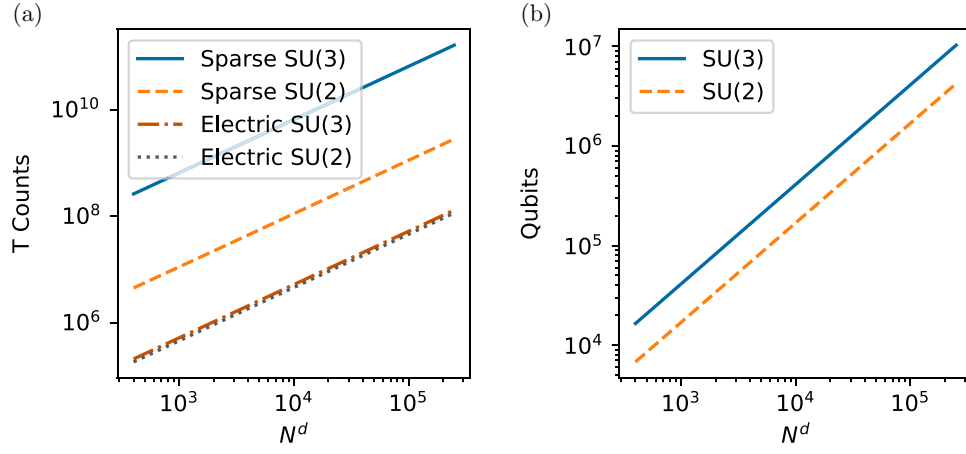


FIG. 17. The total (a) T counts and (b) qubit counts needed to block encode the Hamiltonian for SU(2) and SU(3) gauge theories. Also shown are the electric field costs, which are necessary to compute the cost of HAM-T.

block encoding. The resulting algorithms have polynomial dependence on the number of colors n_c , an exponential improvement over state-of-the-art Trotterized simulation techniques [53,58]. Given the significant improvements found in this work, one may wonder what additional improvements can be made and what the results mean in the context of other approaches to the simulation of LGTs on quantum devices. In terms of additional improvements to the algorithm in this work, we note that the dominant contribution to the cost of the algorithm for small gauge groups such as U(1) and SU(2) come from the fast-forwarding of the diagonal electric field term. Given its asymptotically optimal scaling with the Hamiltonian norm, significant improvements in this direction are unlikely. However, in the case of SU(3) gauge theories, and more complex gauge groups that arise in grand-unification theories such as SU(5) [99] or SO(10) [100], the sparse block-encoding of the magnetic plaquette term

is the dominant cost. It is likely that progress can be made here in reducing resource requirements by making use of LCU implementations utilizing efficient circuits for Schur and Clebsch-Gordan transformations [94].

We note, however, that the efficient Trotterized simulation of LGTs is by no means a lost cause. While a naive look at our reported improvements for SU(3) would indicate that Trotterization would always fall behind qubitization, independent of system size, error, and lattice spacing, the overheads for Trotterization come mainly from a costly decomposition of the plaquette term in Ref. [58] and the expensive arithmetic therein, required for maintaining gauge invariance. Recent work [42–45] has indicated that these costs can be radically improved with two steps: first, working in the magnetic field basis, and, second, discretizing the gauge group, e.g., going from U(1) to $\mathbb{Z}_N$. In doing so, these methods can implement the plaquette term in Trotterization while maintaining gauge invariance

TABLE II. A comparison of Trotterization and qubitized simulation using a sparse block encoding for SU(2) lattice gauge theory in three dimensions for a heavy-ion simulation. Various accuracies ϵ , linear system sizes N , and lattice spacings a are shown. T-gate and qubit counts are derived and a final improvement in the space-time volume of the computation (T counts times logical qubit counts) is reported. Note that we use naive qubitization here, with no HHKL decomposition.

ϵ	N	a	T_{Trotter}	$T_{\text{qubitization}}$	$\mathcal{Q}_{\text{Trotter}}$	$\mathcal{Q}_{\text{qubitization}}$	Improvement
10^{-3}	10^3	10^0	1.2×10^{36}	1.4×10^{26}	10^{11}	3.4×10^{10}	2.9×10^{10}
		10^{-1}	3.9×10^{37}	2.2×10^{26}	10^{11}	2.4×10^{10}	7.1×10^{11}
		10^{-2}	1.2×10^{39}	9.6×10^{26}	10^{11}	2.4×10^{10}	5.1×10^{12}
	10^2	10^0	3.9×10^{31}	1.0×10^{20}	10^8	2.4×10^7	1.5×10^{12}
		10^{-1}	1.2×10^{33}	1.5×10^{20}	10^8	2.4×10^7	3.1×10^{13}
		10^{-2}	3.9×10^{34}	6.9×10^{20}	10^8	2.4×10^7	2.2×10^{14}
10^{-1}	10^3	10^0	1.2×10^{35}	6.8×10^{25}	10^{11}	2.4×10^{10}	7.3×10^9
		10^{-1}	3.9×10^{36}	1.0×10^{26}	10^{11}	2.4×10^{10}	1.5×10^{11}
		10^{-2}	1.2×10^{39}	4.4×10^{26}	10^{11}	2.4×10^{10}	1.1×10^{13}
	10^2	10^0	3.9×10^{30}	4.7×10^{19}	10^8	2.4×10^7	3.3×10^{11}
		10^{-1}	1.2×10^{32}	7.2×10^{19}	10^8	2.4×10^7	6.9×10^{12}
		10^{-2}	3.9×10^{33}	3.2×10^{20}	10^8	2.4×10^7	4.9×10^{13}

TABLE III. A comparison of Trotterization and qubitized simulation using a sparse block encoding for SU(3) lattice gauge theory in three dimensions for a heavy-ion simulation. Various accuracies ϵ , linear system sizes N , and lattice spacings a are shown. T-gate and qubit counts are derived and a final improvement in the space-time volume of the computation (T counts times logical qubit counts) is reported. Note that we use naive qubitization here, with no HHKL decomposition.

ϵ	N	a	T_{Trotter}	$T_{\text{qubitization}}$	Q_{Trotter}	$Q_{\text{qubitization}}$	Improvement
10^{-3}	10^3	10^0	1.0×10^{50}	2.0×10^{27}	2.6×10^{11}	6.0×10^{10}	2.1×10^{23}
		10^{-1}	3.2×10^{51}	2.9×10^{27}	2.6×10^{11}	6.0×10^{10}	4.6×10^{24}
		10^{-2}	1.0×10^{53}	1.2×10^{28}	2.6×10^{11}	6.0×10^{10}	3.5×10^{25}
	10^2	10^0	3.2×10^{45}	1.6×10^{21}	2.6×10^8	6.0×10^7	8.4×10^{24}
		10^{-1}	1.0×10^{47}	2.3×10^{21}	2.6×10^8	6.0×10^7	1.8×10^{26}
		10^{-2}	3.2×10^{48}	1.0×10^{22}	2.6×10^8	6.0×10^7	1.3×10^{27}
10^{-1}	10^3	10^0	1.0×10^{49}	1.7×10^{27}	2.6×10^{11}	6.0×10^{10}	2.5×10^{22}
		10^{-1}	3.2×10^{50}	2.5×10^{27}	2.6×10^{11}	6.0×10^{10}	5.4×10^{23}
		10^{-2}	1.0×10^{52}	1.0×10^{28}	2.6×10^{11}	6.0×10^{10}	4.1×10^{24}
	10^2	10^0	3.2×10^{44}	1.3×10^{21}	2.6×10^8	6.0×10^7	1.0×10^{24}
		10^{-1}	1.0×10^{46}	1.9×10^{21}	2.6×10^8	6.0×10^7	2.1×10^{25}
		10^{-2}	3.2×10^{47}	8.4×10^{21}	2.6×10^8	6.0×10^7	1.6×10^{26}

without expensive arithmetic. Even with these improvements, Trotterization would still be asymptotically worse than qubitized simulation but extrapolating from electronic structure simulation results [33], it is likely that Trotterization with these improvements would beat qubitization for small N and large ϵ .

Trotterization in this regime has another distinct advantage, namely, that one may be able to numerically compute tight, and perhaps even state-dependent, Trotterization bounds [101,102]. It is therefore likely that such Trotterized simulation methods would outpace qubitization for small simulations with low accuracy requirements. On the other hand, if one required high accuracies and large simulation cells—and, in particular, if one were unwilling to incur errors from discretizing the gauge group—the qubitized simulation would certainly be better.

Other recent developments in the simulation of lattice gauge theories can be used to improve both the qubitized and Trotterized simulation methods. In particular, recent results using hybrid architectures, wherein different hardware is used to simulate the bosonic and fermionic degrees of freedom [103,104], allow for significant improvements in the simulation of lattice gauge theories. For example, in such an encoding scheme, the implementation of any unitary in the plaquette term becomes trivial, having zero non-Clifford cost. This may lead to an improvement in the cost of qubitized simulation of 5–6 orders of magnitude for SU(3) relative to the numbers that we have reported.

Two other related points of interest in the simulation of lattice gauge theories are the choice of model and the topic of gauge invariance. In terms of considering different physical models, one natural direction to explore is investigating the peculiarities of mapping various lattice fermions (Wilson [105], overlap [106], domain-wall [107], etc.) onto qubits. More broadly, as numerous formulations of LGTs emerge (purely bosonic [52,108,109], prepotential

[53,110], loop-string-hadron [53,111,112], orbifold [113], etc.), investigating their compatibility with state-of-the-art simulation techniques appears to be a crucial direction of research from the algorithmic perspective. Regarding gauge invariance in the simulation, the qubitized algorithm presented in this work can break gauge invariance up to an error included in the ϵ simulation error of the dynamics. If gauge invariance breaking is absolutely intolerable, then there are a few potential approaches to addressing it. First, one may choose to use discretized gauge theory approaches wherein simulation can be done without incurring gauge invariance breaking [42–45]. Second, one may choose to use specially adapted polynomials for quantum signal processing in the context of qubitized simulation that preserve gauge invariance, an open research problem. Third, one may forego addressing this question at the level of the logical algorithm and instead address it in within the context of the error-correcting code that a prospective fault-tolerant device would make use of [114]. Lastly, one may use an approach such as discrete gauge group Trotterization [42–45], which has built-in gauge invariance under the discrete group.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. S.P. was supported by the National Nuclear Security Administration’s Advanced Simulation and Computing Program. We have benefited from a number of useful discussions with colleagues, who we would like to acknowledge (in alphabetical order): Andrew Baczewski, Riley Chien, Andrew Landahl, Benjamin Morrison, and Stefan Seritan. Sandia National Laboratories is a multimission laboratory managed and operated by

National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract No. DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

APPENDIX A: THE NOTION OF QUBITIZATION

In the quantum simulation literature, the notion of *qubitization* has been repeatedly used for several different concepts. To avoid potential confusion, we discuss those concepts below:

- (1) In the original work [62], *qubitization* referred to the process of constructing a special form of block encoding, the *Szegedy walk operator* W_H , from a general block encoding U_H . The action of U_H on a state of the form $|0^m\rangle|\lambda\rangle$, where $|\lambda\rangle$ is an eigenvector of H with eigenvalue λ , can be written as

$$U_H|0^m\rangle_a|\lambda\rangle_s = \lambda|0^m\rangle_a|\lambda\rangle_s + \sqrt{1 - \lambda^2}|\perp^\lambda\rangle_{as}, \quad (\text{A1})$$

where the subscripts s and a indicate the system- and ancillary-qubit registers. As compared to U_H , the quantum walk operator W_H has an additional property that for each λ , the vector $|\perp^\lambda\rangle_{as}$ factorizes as $|\perp^\lambda\rangle_{as} = |\perp^\lambda\rangle_a|\lambda\rangle_s$:

$$W_H|0^m\rangle_a|\lambda\rangle_s = \left(\lambda|0^m\rangle_a - \sqrt{1 - \lambda^2}|\perp^\lambda\rangle_a \right) |\lambda\rangle_s, \quad (\text{A2})$$

where the minus sign between the two terms is conventional and has been added to match the definition used in Ref. [62].

The form of Eq. (A2) allows one to express its consecutive applications to the eigenstates of H in terms of Chebyshev polynomials and to readily utilize the machinery of quantum signal processing (QSP) [61] for implementing polynomial functions of H using controlled calls to W_H .

- (2) *Qubitization* is also frequently used as a term for the entire procedure of Ref. [62] for simulating the dynamics governed by time-independent Hamiltonians, which rely on QSP and controlled calls to W_H .

It is, however, important to note that simulating time evolution is not the only use case of W_H in quantum simulation. For example, Szegedy walk operator is utilized in a highly efficient implementation of quantum phase estimation from Ref. [115].

Such a construction is possible due to the fact that the spectrum of W_H contains eigenvalues that are simple functions of the Hamiltonian eigenvalues:

$$\text{spec } W_H \supset \{e^{\pm i \arccos \lambda}\}. \quad (\text{A3})$$

- (3) *Qubitization* is sometimes used as an umbrella term for all post-Trotter methods relying on block encoding. This is largely due to the fact that the approach of Ref. [62] provides the most asymptotically efficient way of simulating time-independent Hamiltonians, with complexity $\tilde{O}(\|H\|t + \log 1/\epsilon)$. Such terminology can be quite confusing, since near-optimal methods may use neither W_H nor QSP. For example, one can simulate time evolution with complexity $\tilde{O}(\|H\|t + \log 1/\epsilon)$ by implementing the real and imaginary parts of the time-evolution operator via quantum singular-value transformation (QSVT), which are then added up using the LCU formalism [63,65]. While QSVT is closely related to QSP, in this approach the construction of W_H is not required.
- (4) The notion of *qubitization* has also been used to denote the process of mapping physical degrees of freedom in discretized formulations of quantum field theory onto qubits [116,117]. This arguably most confusing convention is not related in any way to (1)–(3) above.

The present authors did not agree on whether the usage of the term *qubitization* was permissible in the paper, as the truncated Dyson-series algorithm [92,118] is neither related to QSP nor does it employ the Szegedy walk operator. Similarly, the notion of “Trotter” decomposition is used throughout the paper as an umbrella term for methods based on product formulas. The decisions were made by vote.

APPENDIX B: BOUNDING THE LIEB-ROBINSON VELOCITY

Here, we establish an upper bound on the Lieb-Robinson velocity v_{LR} , which is needed to lower bound the linear size of the HHKL block for the simulation. To accomplish this, we employ the algorithm of Ref. [119], which establishes the tightest known Lieb-Robinson bounds for strictly local systems.

The algorithm begins by constructing the *commutativity graph* G of the Hamiltonian, in which the vertices of G correspond to terms of the Hamiltonian, and two vertices, v_1 and v_2 , are connected by an edge if $[v_1, v_2] \neq 0$. Then, one can write the associated Green's function of the commutativity graph, $G_{ij}(t)$, which is a solution to the differential

equation

$$\dot{G}_{ij}(t) = \sum_{k: \langle ij \rangle \in G} H_{ik} G_{kj}(t), \quad (\text{B1})$$

where $H_{ij} = 2\sqrt{|h_i||h_j|}\delta(\langle ij \rangle \in G)$ and h_i are the coefficients of the Hamiltonian term at site i . While a full solution is needed to bound the norm of the commutator for observables over disjoint regions, i.e., to obtain a Lieb-Robinson bound, the simpler task of obtaining the Lieb-Robinson velocity v_{LR} only requires the matrix H . In particular, assuming translation invariance, we can work in the Fourier-integral representation of the Green's function $G_{ij}(t) := G_{I\alpha;J\beta}(t)$, defined by

$$G_{ij}(t) = \int_{-\pi}^{\pi} \frac{d^d k}{(2\pi)^d} G_{\alpha\beta}^{(\vec{k})}(t) e^{i\vec{k} \cdot (\vec{r}_I - \vec{r}_J)}, \quad (\text{B2})$$

where $G_{\alpha\beta}^{(\vec{k})}(t)$ is a solution to the differential equation

$$\dot{G}_{\alpha\beta}^{(\vec{k})}(t) = \sum_{\gamma=1}^I H_{\alpha\gamma}^{(\vec{k})} G_{\gamma\beta}^{(\vec{k})}(t). \quad (\text{B3})$$

Here, $I, J \in [1, I]$ label unit cells of the lattice and $H_{\alpha\beta}^{(\vec{k})}$ is the Fourier transform of $H_{I\alpha;J\beta}$,

$$H_{\alpha\beta}^{(\vec{k})} = \sum_J H_{I\alpha;J\beta} e^{-i\vec{k} \cdot (\vec{r}_I - \vec{r}_J)}. \quad (\text{B4})$$

The task at hand is thus to construct $H_{\alpha\beta}^{(\vec{k})}$, from which we can bound the Lieb-Robinson velocity using its eigenvalues. We now describe a simple approach to accomplishing this task whenever the strictly local Hamiltonian can be represented entirely in terms of Pauli operators or Majorana fermion operators, a structure present in many physical Hamiltonians of interest.

One might wonder how this approach will be possible for the Kogut-Susskind Hamiltonian, given that the gauge operators do not evoke a Pauli or Majorana representation. According to Ref. [60, Lemma 13], the on-site electric field terms cannot change the Lieb-Robinson velocity, so they can be excluded from the commutativity graph. This allows us to additionally exclude the magnetic field terms from the commutativity graph, as the gauge operators only fail to commute with the electric field terms. The gauge operators will, however, remain in the gauge-matter interaction as a result of the fermionic anticommutation relations between terms, but the gauge operators will only contribute a constant prefactor from their norm, so we can write the commutativity graph purely in terms of fermionic operators. We can now construct $H^{(\vec{k})}$ for our strictly local Hamiltonian expressed only as Pauli operators or Majorana fermions.

To do so, we note that we can encode any Pauli or Majorana operator on n qubits as a binary string of length $2n$, namely, a vector $v \in \mathbb{F}_2^{2n}$. The first n elements of v index the presence of an X or γ operator and the second n elements indicate the presence of a Z or $\bar{\gamma}$ operator, depending on the representation. Now consider the binary field extension $\mathbb{F}_2(x, x^{-1})^{2n}$ such that for vectors $v, w \in \mathbb{F}_2^{2n}$, we have $v + xw \in \mathbb{F}_2(x, x^{-1})^{2n}$. The variables x and x^{-1} allow us to represent translations of unit cells in the lattice, so for higher dimensions, additional extension variables, and their inverses, must be included.

The associated commutation relations of this representation are determined by $\sigma^\dagger \Lambda \sigma \pmod{2} \in \mathbb{F}_2(x, x^{-1})^{2n}$, where σ is a $2 \times n$ matrix composed of the binary representation vectors for the n terms in a unit cell of the lattice and

$$\Lambda = \mathbb{1} \text{ (Majorana)} \quad \text{and} \quad \Lambda = \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix} \text{ (Pauli)}. \quad (\text{B5})$$

Also, in this representation, $\dagger$ corresponds to taking a transpose of the matrix and replacing x with x^{-1} , or vice versa.

The next step is to define a function $f^{(\vec{k})} : \mathbb{F}_2(x, x^{-1})^{2n} \rightarrow \mathbb{C}^{2n}$ that maps $x \mapsto e^{ik}$ or, in general, $x^\ell \mapsto e^{ik\ell}$. We can then evaluate $f^{(\vec{k})}(\sigma^\dagger \Lambda \sigma \pmod{2})$ for a given wave vector k . The last piece needed to construct the coefficient matrix $H^{(\vec{k})}$ is a vector storing the coefficients of the Hamiltonian terms. In particular, let $\vec{c} = \sqrt{2}(\sqrt{h_1}, \sqrt{h_2}, \dots, \sqrt{h_n})$. Then, combining all of this together, we have

$$H^{(\vec{k})} = (\vec{c} \otimes \vec{c}^\dagger) \odot f^{(\vec{k})}(\sigma^\dagger \Lambda \sigma \pmod{2}), \quad (\text{B6})$$

where $\odot$ indicates element-wise multiplication.

At this point, we are ready to bound the Lieb-Robinson velocity, v_{LR} . According to Ref. [119], we have

$$v_{\text{LR}} \leq \min_{\kappa_0 > 0} \frac{\omega(i\vec{k}_0)}{\kappa_0}, \quad (\text{B7})$$

where $\omega(\vec{k})$ is the largest eigenvalue of $H^{(\vec{k})}$ and $\vec{k}_0 = \text{sgn}(\vec{r})\kappa_0$. Finally, $\vec{r}$ varies over all possible sign configurations for unit translations of the unit cell.

We now use this technique to bound the Lieb-Robinson velocity for the U(1) LGT in $d = 2$. We use the Majorana representation for simplicity, in which our fermionic terms of the Kogut-Susskind Hamiltonian become

$$H_M = \frac{g_M}{2} \sum_x (\mathbb{1} + i\gamma_x \bar{\gamma}_x), \quad (\text{B8})$$

$$\begin{aligned} H_{GM} = & \frac{g_{GM}}{4} \sum_{x, \hat{n}} \gamma_x (U - U^\dagger) \gamma_{x+\hat{n}} + i\gamma_x (U + U^\dagger) \bar{\gamma}_{x+\hat{n}} \\ & - i\bar{\gamma}_x (U + U^\dagger) \gamma_{x+\hat{n}} + \bar{\gamma}_x (U - U^\dagger) \bar{\gamma}_{x+\hat{n}}. \end{aligned} \quad (\text{B9})$$

We construct our binary representation vectors in the field extension $\mathbb{F}_2(x, x^{-1}, y, y^{-1})^{2n}$, which allows us to construct

$$\sigma = \begin{pmatrix} 1 & 1+x & 1 & x & 0 & 1+y & 1 & y & 0 \\ 1 & 0 & x & 1 & 1+x & 0 & y & 1 & 1+y \end{pmatrix}. \quad (\text{B10})$$

Using the coefficients $g_M/2$ and $g_{GM}/4$, we can construct the coefficient vector $\vec{c}$ and employ Eq. (B6) to obtain $H^{(\vec{k})}$. Finally, we compute the largest eigenvalue and map $\vec{k} = (k_x, k_y) \mapsto i\kappa_0$. For the U(1) LGT in $d = 2$, $\text{sgn}(\vec{r})$ yields the same eigenvalue for all possible translations of the unit cell. Then, employing Eq. (B7), we minimize the eigenvalue over κ_0 , where we have fixed the parameters $g_M = m$ and $g_{GM} = 1/2a$ and used the parameter values $m = 10$ and $a = 0.1$ reported in Ref. [58]. This results in

$$v_{\text{LR}} \leq 53. \quad (\text{B11})$$

-
- [1] W. A. de Jong, E. Bylaska, N. Govind, C. L. Janssen, K. Kowalski, T. Müller, I. M. B. Nielsen, H. J. J. van Dam, V. Veryazov, and R. Lindh, Utilizing high performance computing for chemistry: Parallel computational chemistry, *Phys. Chem. Chem. Phys.* **12**, 6896 (2010).
 - [2] A. Kononov and A. Schleife, Pre-equilibrium stopping and charge capture in proton-irradiated aluminum sheets, *Phys. Rev. B* **102**, 165401 (2020).
 - [3] C. DeTar and S. Gottlieb, Lattice quantum chromodynamics comes of age, *Phys. Today* **57**, 45 (2004).
 - [4] K. Nagata, Finite-density lattice QCD and sign problem: Current status and open problems, *Prog. Part. Nucl. Phys.* **127**, 103991 (2022).
 - [5] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
 - [6] C. W. Bauer, *et al.*, Quantum simulation for high-energy physics, *PRX Quantum* **4**, 027001 (2023).
 - [7] C. W. Bauer, Z. Davoudi, N. Klco, and M. J. Savage, Quantum simulation of fundamental particles and forces, *Nat. Rev. Phys.* **5**, 420 (2023).
 - [8] B. Fauseweh, Quantum many-body simulations on digital quantum computers: State-of-the-art and future challenges, *Nat. Commun.* **15**, 2123 (2024).
 - [9] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, *Science* **309**, 1704 (2005).
 - [10] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Simulation of electronic structure Hamiltonians using quantum computers, *Mol. Phys.* **109**, 735 (2011).
 - [11] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, Improving quantum algorithms for quantum chemistry, *Quantum Info. Comput.* **15**, 1 (2015).
 - [12] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2014).
 - [13] 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).
 - [14] 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).
 - [15] 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 Info. Comput.* **15**, 361 (2015).
 - [16] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe, Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation, *Phys. Rev. A* **91**, 022311 (2015).
 - [17] 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).
 - [18] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, *Phys. Rev. X* **8**, 041015 (2018).
 - [19] 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).
 - [20] 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).
 - [21] 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).
 - [22] R. Babbush, D. W. Berry, J. R. McClean, and H. Neven, Quantum simulation of chemistry with sublinear scaling in basis size, *npj Quantum Inf.* **5**, 92 (2019).
 - [23] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization, *Quantum* **3**, 208 (2019).
 - [24] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, H. Neven, and R. Babbush, Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization, *Quantum* **4**, 296 (2020).
 - [25] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* **120**, 12685 (2020).
 - [26] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, *PRX Quantum* **2**, 030305 (2021).
 - [27] Y. Su, H.-Y. Huang, and E. T. Campbell, Nearly tight Trotterization of interacting electrons, *Quantum* **5**, 495 (2021).
 - [28] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, Fault-tolerant quantum simulations of chemistry in first quantization, *PRX Quantum* **2**, 040332 (2021).

- [29] J. J. Goings, A. White, J. Lee, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers, *Proc. Natl. Acad. Sci.* **119**, e2203533119 (2022).
- [30] N. C. Rubin, D. W. Berry, F. D. Malone, A. F. White, T. Khattar, A. E. DePrince, S. Siculo, M. Kühn, M. Kaicher, J. Lee, and R. Babbush, Fault-tolerant quantum simulation of materials using Bloch orbitals, *PRX Quantum* **4**, 040303 (2023).
- [31] D. W. Berry, N. C. Rubin, A. O. Elnabawy, G. Ahlers, A. E. DePrince III, J. Lee, C. Gogolin, and R. Babbush, Quantum simulation of realistic materials in first quantization using non-local pseudopotentials, *ArXiv:2312.07654*.
- [32] I. Loaiza, A. M. Khah, N. Wiebe, and A. F. Izmaylov, Reducing molecular electronic Hamiltonian simulation cost for linear combination of unitaries approaches, *Quantum Sci. Technol.* **8**, 035019 (2023).
- [33] N. C. Rubin, D. W. Berry, A. Kononov, F. D. Malone, T. Khattar, A. White, J. Lee, H. Neven, R. Babbush, and A. D. Baczewski, Quantum computation of stopping power for inertial fusion target design, *PNAS* **121**, e2317772121 (2024).
- [34] J. D. Watson, J. Bringewatt, A. F. Shaw, A. M. Childs, A. V. Gorshkov, and Z. Davoudi, Quantum algorithms for simulating nuclear effective field theories, *ArXiv:2312.05344*.
- [35] W. Du and J. P. Vary, Multinucleon structure and dynamics via quantum computing, *Phys. Rev. A* **108**, 052614 (2023).
- [36] D. Liu, W. Du, L. Lin, J. P. Vary, and C. Yang, An efficient quantum circuit for block encoding a pairing Hamiltonian, *ArXiv:2402.11205*.
- [37] W. Du and J. P. Vary, Systematic many-fermion Hamiltonian input scheme and spectral calculations on quantum computers, *ArXiv:2402.08969*.
- [38] P. Wang, W. Du, W. Zuo, and J. P. Vary, Nuclear scattering via quantum computing, *Phys. Rev. C* **109**, 064623 (2024).
- [39] S. Pathak, A. E. Russo, S. K. Seritan, and A. D. Baczewski, Quantifying T-gate-count improvements for ground-state-energy estimation with near-optimal state preparation, *Phys. Rev. A* **107**, L040601 (2023).
- [40] S. Pathak, A. E. Russo, S. Seritan, A. B. Magann, E. Bobrow, A. J. Landahl, and A. D. Baczewski, Requirements for building effective Hamiltonians using quantum-enhanced density matrix downfolding, *ArXiv:2403.01043*.
- [41] S. P. Jordan, H. Krovi, K. S. M. Lee, and J. Preskill, BQP-completeness of scattering in scalar quantum field theory, *Quantum* **2**, 44 (2018).
- [42] E. J. Gustafson, H. Lamm, F. Lovelace, and D. Musk, Primitive quantum gates for an SU(2) discrete subgroup: Binary tetrahedral, *Phys. Rev. D* **106**, 114501 (2022).
- [43] E. J. Gustafson, H. Lamm, and F. Lovelace, Primitive quantum gates for an SU(2) discrete subgroup: Binary octahedral, *Phys. Rev. D* **109**, 054503 (2024).
- [44] E. J. Gustafson, Y. Ji, H. Lamm, E. M. Murairi, S. O. Perez, and S. Zhu, Primitive quantum gates for an SU(3) discrete subgroup: $\sigma(36 \times 3)$, *Phys. Rev. D* **110**, 034515 (2024).
- [45] H. Lamm, Y.-Y. Li, J. Shu, Y.-L. Wang, and B. Xu, Block encodings of discrete subgroups on quantum computer, *Phys. Rev. D* **110**, 054505 (2024).
- [46] B. Assi and H. Lamm, Digitization and subduction of SU(N) gauge theories, *Phys. Rev. D* **110**, 074511 (2024).
- [47] E. Zohar, A. Farace, B. Reznik, and J. I. Cirac, Digital lattice gauge theories, *Phys. Rev. A* **95**, 023604 (2017).
- [48] J. Bender, E. Zohar, A. Farace, and J. I. Cirac, Digital quantum simulation of lattice gauge theories in three spatial dimensions, *New J. Phys.* **20**, 093001 (2018).
- [49] H. Lamm, S. Lawrence, and Y. Y. and, General methods for digital quantum simulation of gauge theories, *Phys. Rev. D* **100**, 034518 (2019).
- [50] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, Lattice renormalization of quantum simulations, *Phys. Rev. D* **104**, 094519 (2021).
- [51] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, Improved Hamiltonians for quantum simulations of gauge theories, *Phys. Rev. Lett.* **129**, 051601 (2022).
- [52] G. Pardo, T. Greenberg, A. Fortinsky, N. Katz, and E. Zohar, Resource-efficient quantum simulation of lattice gauge theories in arbitrary dimensions: Solving for Gauss's law and fermion elimination, *Phys. Rev. Res.* **5**, 023077 (2023).
- [53] Z. Davoudi, A. F. Shaw, and J. R. Stryker, General quantum algorithms for Hamiltonian simulation with applications to a non-Abelian lattice gauge theory, *Quantum* **7**, 1213 (2023).
- [54] T. Byrnes and Y. Yamamoto, Simulating lattice gauge theories on a quantum computer, *Phys. Rev. A* **73**, 022328 (2006).
- [55] S. V. Mathis, G. Mazzola, and I. Tavernelli, Toward scalable simulations of lattice gauge theories on quantum computers, *Phys. Rev. D* **102**, 094501 (2020).
- [56] A. F. Shaw, P. Lougovski, J. R. Stryker, and N. Wiebe, Quantum algorithms for simulating the lattice schwinger model, *Quantum* **4**, 306 (2020).
- [57] A. Ciavarella, N. Klco, and M. J. Savage, Trailhead for quantum simulation of SU(3) Yang-Mills lattice gauge theory in the local multiplet basis, *Phys. Rev. D* **103**, 094501 (2021).
- [58] A. Kan and Y. Nam, Lattice quantum chromodynamics and electrodynamics on a universal quantum computer, *ArXiv:2107.12769*.
- [59] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, *Phys. Rev. X* **11**, 011020 (2021).
- [60] Y. Tong, V. V. Albert, J. R. McClean, J. Preskill, and Y. Su, Provably accurate simulation of gauge theories and bosonic systems, *Quantum* **6**, 816 (2022).
- [61] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [62] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [63] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2019).
- [64] L. Lin and Y. Tong, Near-optimal ground state preparation, *Quantum* **4**, 372 (2020).

- [65] L. Lin, Lecture notes on quantum algorithms for scientific computation, [ArXiv:2201.08309](#).
- [66] C. F. Kane, S. Hariprakash, N. S. Modi, M. Kreshchuk, and C. W. Bauer, Block encoding by signal processing, [arXiv:2408.16824](#).
- [67] C. F. Kane, N. Gomes, and M. Kreshchuk, Nearly optimal state preparation for quantum simulations of lattice gauge theories, [Phys. Rev. A **110**, 012455 \(2024\)](#).
- [68] M. Kreshchuk, W. M. Kirby, G. Goldstein, H. Beauchemin, and P. J. Love, Quantum simulation of quantum field theory in the light-front formulation, [Phys. Rev. A **105**, 032418 \(2022\)](#).
- [69] W. M. Kirby, S. Hadi, M. Kreshchuk, and P. J. Love, Quantum simulation of second-quantized Hamiltonians in compact encoding, [Phys. Rev. A **104**, 042607 \(2021\)](#).
- [70] S. Hariprakash, N. S. Modi, M. Kreshchuk, C. F. Kane, and C. W. Bauer, Strategies for simulating time evolution of Hamiltonian lattice field theories, [ArXiv:2312.11637](#).
- [71] L. W. Anderson, M. Kiffner, T. O’Leary, J. Crain, and D. Jaksch, Solving lattice gauge theories using the quantum Krylov algorithm and qubitization, [ArXiv:2403.08859](#).
- [72] C. F. Kane, S. Hariprakash, N. S. Modi, M. Kreshchuk, and C. W. Bauer, Block encoding by signal processing, [ArXiv:2408.16824](#).
- [73] J. Kogut and L. Susskind, Hamiltonian formulation of Wilson’s lattice gauge theories, [Phys. Rev. D **11**, 395 \(1975\)](#).
- [74] E. Zohar and M. Burrello, Formulation of lattice gauge theories for quantum simulations, [Phys. Rev. D **91**, 054506 \(2015\)](#).
- [75] Following the literature on quantum simulation [58, 60], we use simplified notations for the Kogut-Susskind Hamiltonian. Complete expressions for the 2+1D and 3+1D cases can be found in Refs. [120,121].
- [76] Electric field digitization in Eq. (8) is analogous to digitization of scalar fields considered in Refs. [122–127].
- [77] S. Harmalkar, H. Lamm, and S. Lawrence, Quantum simulation of field theories without state preparation, [ArXiv:2001.11490](#).
- [78] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, [J. Math. Phys. **32**, 400 \(1991\)](#).
- [79] I. D’Andrea, C. W. Bauer, D. M. Grabowska, and M. Freytsis, New basis for Hamiltonian SU(2) simulations, [Phys. Rev. D **109**, 074501 \(2024\)](#).
- [80] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, Quantum algorithm for simulating real time evolution of lattice Hamiltonians, [SIAM J. Comput. **52**, FOCS18 \(2021\)](#).
- [81] B. Peng, Y. Su, D. Claudino, K. Kowalski, G. H. Low, and M. Roetteler, Quantum simulation of boson-related Hamiltonians: Techniques, effective Hamiltonian construction, and error analysis, [ArXiv:2307.06580](#).
- [82] N. P. Sawaya, T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi, Resource-efficient digital quantum simulation of d -level systems for photonic, vibrational, and spin- s Hamiltonians, [npj Quantum Inf. **6**, 49 \(2020\)](#).
- [83] P. Jordan and E. Wigner, Über das Paulische äquivalenzverbot, [Z. Phys. **47**, 631 \(1928\)](#).
- [84] K. Setia, S. Bravyi, A. Mezzacapo, and J. D. Whitfield, Superfast encodings for fermionic quantum simulation, [Phys. Rev. Res. **1**, 033033 \(2019\)](#).
- [85] F. Verstraete and J. I. Cirac, Mapping local Hamiltonians of fermions to local Hamiltonians of spins, [J. Stat. Mech.: Theory Exp. **2005**, P09012 \(2005\)](#).
- [86] S. Gu, R. D. Somma, and B. Şahinoğlu, Fast-forwarding quantum evolution, [Quantum **5**, 577 \(2021\)](#).
- [87] A. M. Childs, Ph.D. thesis, Massachusetts Institute of Technology (Department of Physics), 2004.
- [88] A. Parent, M. Roetteler, and M. Mosca, Improved reversible and quantum circuits for Karatsuba-based integer multiplication, [ArXiv:1706.03419](#).
- [89] N. J. Ross and P. Selinger, Optimal ancilla-free Clifford+T approximation of z -rotations, [Quantum Inf. Comput. **16**, 901 \(2016\)](#).
- [90] W. M. Kirby and P. J. Love, Variational quantum eigensolvers for sparse Hamiltonians, [Phys. Rev. Lett. **127**, 110503 \(2021\)](#).
- [91] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, [ArXiv:1202.5822](#).
- [92] G. H. Low and N. Wiebe, Hamiltonian simulation in the interaction picture, [ArXiv:1805.00675](#).
- [93] X. Y. Xu, Y. Qi, L. Zhang, F. F. Assaad, C. Xu, and Z. Y. Meng, Monte Carlo study of lattice compact quantum electrodynamics with fermionic matter: The parent state of quantum phases, [Phys. Rev. X **9**, 021022 \(2019\)](#).
- [94] D. Bacon, I. L. Chuang, and A. W. Harrow, Efficient quantum circuits for Schur and Clebsch-Gordan transforms, [Phys. Rev. Lett. **97**, 170502 \(2006\)](#).
- [95] C. Gidney, Halving the cost of quantum addition, [Quantum **2**, 74 \(2018\)](#).
- [96] H. Thapliyal, T. S. S. Varun, E. Munoz-Coreas, K. A. Britt, and T. S. Humble, in *2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)* (IEEE, Bhopal, India, 2017), p. 123.
- [97] S. Sultana and K. Radecka, in *2011 18th IEEE International Conference on Electronics, Circuits, and Systems* (IEEE, Beirut, Lebanon, 2011), p. 141.
- [98] K. Setia and J. D. Whitfield, Bravyi-Kitaev Superfast simulation of electronic structure on a quantum computer, [J. Chem. Phys. **148**, 164104 \(2018\)](#).
- [99] H. Georgi and S. L. Glashow, Unity of all elementary-particle forces, [Phys. Rev. Lett. **32**, 438 \(1974\)](#).
- [100] H. Fritzsch and P. Minkowski, Unified interactions of leptons and hadrons, [Annals Phys. **93**, 193 \(1975\)](#).
- [101] B. Şahinoğlu and R. D. Somma, Hamiltonian simulation in the low-energy subspace, [npj Quantum Inf. **7**, 119 \(2021\)](#).
- [102] C. Yi and E. Crosson, Spectral analysis of product formulas for quantum simulation, [npj Quantum Inf. **8**, 37 \(2022\)](#).
- [103] D. González-Cuadra, T. V. Zache, J. Carrasco, B. Kraus, and P. Zoller, Hardware efficient quantum simulation of non-Abelian gauge theories with qudits on Rydberg platforms, [Phys. Rev. Lett. **129**, 160501 \(2022\)](#).
- [104] T. V. Zache, D. González-Cuadra, and P. Zoller, Fermion-qudit quantum processors for simulating lattice gauge theories with matter, [Quantum **7**, 1140 \(2023\)](#).

- [105] K. G. Wilson, Confinement of quarks, *Phys. Rev. D* **10**, 2445 (1974).
- [106] H. Neuberger, Exactly massless quarks on the lattice, *Phys. Lett. B* **417**, 141 (1998).
- [107] D. B. Kaplan, A method for simulating chiral fermions on the lattice, *Phys. Lett. B* **288**, 342 (1992).
- [108] E. Zohar and J. I. Cirac, Eliminating fermionic matter fields in lattice gauge theories, *Phys. Rev. B* **98**, 075119 (2018).
- [109] E. Zohar and J. I. Cirac, Removing staggered fermionic matter in U(N) and SU(N) lattice gauge theories, *Phys. Rev. D* **99**, 114511 (2019).
- [110] E. Zohar, J. I. Cirac, and B. Reznik, Cold-atom quantum simulator for SU(2) Yang-Mills lattice gauge theory, *Phys. Rev. Lett.* **110**, 125304 (2013).
- [111] I. Raychowdhury and J. R. Stryker, Loop, string, and hadron dynamics in SU(2) Hamiltonian lattice gauge theories, *Phys. Rev. D* **101**, 114502 (2020).
- [112] Z. Davoudi, I. Raychowdhury, and A. Shaw, Search for efficient formulations for Hamiltonian simulation of non-Abelian lattice gauge theories, *Phys. Rev. D* **104**, 074505 (2021).
- [113] A. J. Buser, H. Gharibyan, M. Hanada, M. Honda, and J. Liu, Quantum simulation of gauge theory via orbifold lattice, *J. High Energy Phys.* **2021**, 34 (2021).
- [114] L. Spagnoli, A. Roggero, and N. Wiebe, Fault-tolerant simulation of lattice gauge theories with gauge covariant codes, *ArXiv:2405.19293*.
- [115] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, *Phys. Rev. X* **8**, 041015 (2018).
- [116] X.-Y. Huang, L. Yu, X. Lu, Y. Yang, D.-S. Li, C.-W. Wu, W. Wu, and P.-X. Chen, Qubitization of bosons, *ArXiv:2105.12563*.
- [117] A. Alexandru, P. F. Bedaque, A. Carosso, M. J. Cervia, and A. Sheng, Qubitization strategies for bosonic field theories, *Phys. Rev. D* **107**, 034503 (2023).
- [118] D. W. Berry, A. M. Childs, Y. Su, X. Wang, and N. Wiebe, Time-dependent Hamiltonian simulation with L^1 -norm scaling, *Quantum* **4**, 254 (2020).
- [119] Z. Wang and K. R. Hazzard, Tightening the Lieb-Robinson bound in locally interacting systems, *PRX Quantum* **1**, 010303 (2020).
- [120] L. Susskind, Lattice fermions, *Phys. Rev. D* **16**, 3031 (1977).
- [121] A. Crippa, S. Romiti, L. Funcke, K. Jansen, S. Kühn, P. Stornati, and C. Urbach, Towards determining the (2+1)-dimensional quantum electrodynamics running coupling with Monte Carlo and quantum computing methods, *ArXiv:2404.17545*.
- [122] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum computation of scattering in scalar quantum field theories, *Quantum Inf. Comput.* **14**, 1014 (2014).
- [123] N. Klco and M. J. Savage, Digitization of scalar fields for quantum computing, *Phys. Rev. A* **99**, 052335 (2019).
- [124] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, Digital quantum computation of fermion-boson interacting systems, *Phys. Rev. A* **98**, 042312 (2018).
- [125] T. Farrelly and J. Streich, Discretizing quantum field theories for quantum simulation, *ArXiv:2002.02643*.
- [126] A. C. Y. Li, A. Macridin, S. Mrenna, and P. Spentzouris, Simulating scalar field theories on quantum computers with limited resources, *Phys. Rev. A* **107**, 032603 (2023).
- [127] M. Hanada, J. Liu, E. Rinaldi, and M. Tezuka, Estimating truncation effects of quantum bosonic systems using sampling algorithms, *Mach. Learn. Sci. Tech.* **4**, 045021 (2023).