

Primitive quantum gates for dihedral gauge theoriesM. Sohaib Alam^{1,2,3,*} Stuart Hadfield^{2,3,†} Henry Lamm^{4,‡} and Andy C. Y. Li^{4,§}

(SQMS Collaboration)

¹*Rigetti Computing, Berkeley, California 94701, USA*²*Quantum Artificial Intelligence Laboratory (QuAIL),**NASA Ames Research Center, Moffett Field, California 94035, USA*³*USRA Research Institute for Advanced Computer Science (RIACS),
Mountain View, California 94043, USA*⁴*Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*

(Received 12 April 2022; accepted 9 May 2022; published 2 June 2022)

We describe the simulation of dihedral gauge theories on digital quantum computers. The non-Abelian discrete gauge group D_N —the dihedral group—serves as an approximation to $U(1) \times \mathbb{Z}_2$ lattice gauge theory. In order to carry out such a lattice simulation, we detail the construction of efficient quantum circuits to realize basic primitives including the non-Abelian Fourier transform over D_N , the trace operation, and the group multiplication and inversion operations. For each case the required quantum resources scale linearly or as low-degree polynomials in $n = \log N$. We experimentally benchmark our gates on the Rigetti Aspen-9 quantum processor for the case of D_4 . The estimated fidelity of all D_4 gates was found to exceed 80%.

DOI: [10.1103/PhysRevD.105.114501](https://doi.org/10.1103/PhysRevD.105.114501)**I. INTRODUCTION**

A promising area for quantum advantage is simulating the dynamics of nonperturbative quantum field theories [1–5]. In order to propagate for a time t , one requires the unitary operator $U(t) = e^{-iHt}$ which in general may be challenging to efficiently implement on a quantum computer. Different quantum algorithms exist for approximating $U(t)$, in particular, Trotter-Suzuki product formulas [3,4,6–14], quantum walks [15], Taylor series approximations [16], and quantum signal processing [17,18], as well as more recent variational approaches [19–21]. While each of these algorithms differs in how to approximate $U(t)$, fundamentally these methods all require implementing operations derived from the Hamiltonian H as quantum circuits [22]. Thus, a small set of primitive operations should be required for all of them. In the case of gauge theories, the Kogut-Susskind Hamiltonian H_{KS} [23] is the most common Hamiltonian discussed in the literature for quantum simulations. Using H_{KS} , initial comparisons between a few quantum algorithms were performed for the Schwinger model [24].

For efficient digital simulations, the local lattice degrees of freedom must be truncated. For fermionic degrees of freedom, this is relatively easy [25–27]. Further proposals discuss how to map lattice fermions (e.g., Wilson and staggered) to these encodings [28] or use gauge symmetry to eliminate them [29,30]. The question of gauge boson digitization is murkier, with many proposals [10,31–66] that make complicated tradeoffs. Digitizing reduces symmetries—either explicitly or through finite-truncations [33]. Furthermore, the utility of a given scheme depends upon spacetime dimensionality [67]. Care must be taken, as the regulated theory may not have the original theory as its continuum limit [68–73].

One promising digitization method is the approximation of gauge theories by discrete subgroups [10,40–44,65,74]. Replacing the continuous group by a discrete subgroup was explored in the early days of Euclidean lattice field theory as a resource reduction procedure, with most studies focusing on the theories in $(3 + 1)$ dimensions with the Wilson action. The viability of the \mathbb{Z}_N subgroups replacing $U(1)$ were studied in [75,76]. Further studies of the crystal-like discrete subgroups of $SU(N)$ were performed [40,41,65,77–79], including with fermions [80,81]. Alongside this work, theoretical studies revealed that such discrete subgroup approximations correspond to effective field theories of continuous groups where a mass is given to the gauge fields through the Higgs mechanism [82–86]. The result of this

* malam@usra.edu† stuart.hadfield@nasa.gov‡ hlyamm@fnal.gov§ cli@fnal.gov

mass is that the discrete subgroup fails to well approximate the continuous group below a certain lattice spacing a_f (or equivalently beyond a certain coupling β_f).

In lattice calculations, one performs calculations at fixed lattice spacing $a = a(\beta)$ which shrinks as $\beta \rightarrow \infty$ for asymptotically free theories. To control extrapolation errors in taking $a \rightarrow 0$, one simulates in the *scaling regime* of $a \ll m_{\text{IR}}^{-1}$ where m_{IR} is the infrared mass scale of the physics of interest. We will consider the start of the scaling regime as occurring at a_s . Thus, the approximation error from using discrete subgroups should be small provided that $a_s \gtrsim a_f$ or equivalently that $\beta_s \lesssim \beta_f$. For the Wilson action, β_f are known. In the case of $U(1)$ in $(3+1)$ - d with $\beta_s = 1$, $\mathbb{Z}_{n>5}$ satisfies $\beta_f > \beta_s$. For non-Abelian groups, only a finite set of crystal-like subgroups exist. $SU(2)$ has three: the binary tetrahedral $\mathbb{B}\mathbb{T}$, the binary octahedral $\mathbb{B}\mathbb{O}$, and the binary icosahedral $\mathbb{B}\mathbb{I}$. While $\mathbb{B}\mathbb{T}$ has $\beta_f = 2.24(8)$ in $3+1d$, $\mathbb{B}\mathbb{O}$ and $\mathbb{B}\mathbb{I}$ have $\beta_f = 3.26(8)$ and $\beta_f = 5.82(8)$ respectively [41], above $\beta_s = 2.2$. Hence, $\mathbb{B}\mathbb{O}$ and $\mathbb{B}\mathbb{I}$ appear useful for $SU(2)$.

For $SU(3)$ (the theory underlying QCD) with $\beta_s \approx 6$ in $(3+1)$ - d , all five crystal-like subgroups have $\beta_f < \beta_s$, with the largest, the 1080-element Valentiner group,¹ \mathbb{V} having $\beta_f = 3.935(5)$ [41]. Thus the discrete approximation is inadequate when using the Wilson action. By extending \mathbb{V} to include the midpoints between elements of \mathbb{V} , one can increase $\beta_f \approx 7$ [90]. However this requires more qubits and sacrifices gauge symmetry. This gauge violation is dangerous on noisy quantum computers [91–93]. An alternative approach to decrease a_f introduces additional terms into the lattice action [41,76,79,87,88,94–97], although only in [41,79] were Monte Carlo calculations undertaken for $SU(3)$. In [41] it was shown that such modified actions of \mathbb{V} could reach into the scaling regime, finding calculations could be undertaken at $a > 0.08$ fm without the effects of a_f being seen. This suggests that \mathbb{V} can reproduce $SU(3)$ in the scaling region with a modified action, such that practical quantum computations of $SU(3)$ could be performed.

Non-Abelian gauge theories have a number of novel features not seen in Abelian ones, and thus studies of Abelian theories like $U(1)$ or \mathbb{Z}_N may be unrepresentative of the full complexity of lattice gauge theories. Unfortunately, even the smallest crystal-like subgroup of a non-Abelian theory, $\mathbb{B}\mathbb{T}$ requires 6 qubits per register and is thus beyond current hardware. To reduce this cost to be more inline with near-term devices, in this work we study a class of discrete groups that are not crystal-like subgroups of a single continuous group. The binary dihedral groups D_N have $2N$ elements and are each an extension of \mathbb{Z}_N by an additional \mathbb{Z}_2 subgroup giving $D_N \simeq \mathbb{Z}_N \rtimes \mathbb{Z}_2$. In the limit of $N \rightarrow \infty$ this becomes $U(1) \times \mathbb{Z}_2$. D_3 and D_4 have

previously been investigated for simulation on quantum computers [10,22]. Having 6 and 8 elements respectively, they both require 3 qubits per register. Unfortunately in both $(2+1)$ - d and $(3+1)$ - d , these two groups have $a_s < a_f$ with the standard Wilson action and thus either a modified action or larger group is required to minimize the discrete group approximation error. For larger N , the necessary primitive gates are unknown, and within this work we will derive a set of such gates for D_{2^n} gauge theories which naturally map onto qubit devices.

Since we are interested in finding gauge theories that could be simulated on near-term quantum devices, it behooves us to study not just $(3+1)$ - d but also consider $(2+1)$ - d theories. Using classical lattice simulations, we have determined that in both spacetimes, while $\beta_f > \beta_s$ for D_8 , it is only slightly larger, so either D_9 (which we did not simulate) or D_{10} would be desirable to have simulations with sufficiently small a (See Fig. 1). The dependence of a_f on β_f within the scaling regime is exponential, so a slightly larger group can have dramatically smaller errors. Since D_{2^n} theories can more efficiently be implemented in qubits, we believe that the 5-qubit D_{16} should be the ultimate target for quantum hardware of the near-future, with D_4 and D_8 as important stepping stones to it. After this, $\mathbb{B}\mathbb{T}$ would be a natural next step.

In this paper, we construct quantum circuits implementing the four primitive gates (inversion, multiplication, trace, and Fourier) required to simulate the D_{2^n} theories. A Trotterized time evolution circuit can be built using these gates. Although Trotterized evolution on quantum processors is

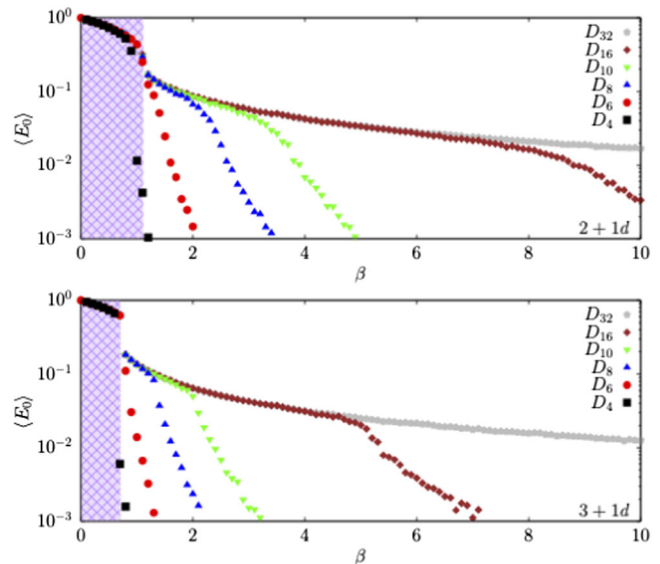


FIG. 1. Euclidean calculations of lattice energy density $\langle E_0 \rangle$ as measured by the expectation value of the plaquette as a function of Wilson coupling beta on 4^d lattices for different D_N groups (top) $d = 2 + 1$ dimensions and (bottom) $d = 3 + 1$ dimensions. The shaded region corresponds to couplings outside the scaling regime for the $U(1) \times \mathbb{Z}_2$ theory.

¹Sometimes called $S(1080)$ [41,79,87,88] or $\Sigma_{3 \times 360}$ [89].

unfeasible at present due to the limited two-qubit gate fidelities, we benchmark the primitive gates for D_4 individually on the Rigetti Aspen-9 QPU to evaluate if simulations are practical on near-term quantum processors. We perform process tomography to measure the process fidelities of the trace and Fourier gates. Since process tomography is experimentally costly for a six-qubit gate, the fidelity of the multiplication gate is estimated by the fraction of the correct bit-strings produced for all possible pairs of input bit-strings. The process fidelity of the inversion gate is reported based on the benchmark result of the CCPHASE gate [98].

This paper is organized as follows. In Sec. II the Euclidean action lattice formalism is briefly reviewed and its connection to the Hamiltonian formulation is elucidated. Section III presents an overview of the four primitive gates required for implementing the group operations necessary for lattice gauge theories on quantum computers. This is followed by quantum circuit constructions for these gates for D_{2^n} gauge theories: the inversion gate in Sec. IV, the multiplication gate in Sec. V, the trace gate in Sec. VI, and the Fourier transform gate in Sec. VII. Benchmarking results for our D_4 gates on the Rigetti Aspen-9 QPU are found in Sec. VIII. We conclude and discuss future work in Sec. IX.

II. LATTICE FIELD THEORY

In order to understand the origin of the various primitive gates requires from simulating lattice gauge theory, it is useful to review the connection between the Kogut-Susskind Hamiltonian [23] and the Euclidean Wilson action. We summarize the derivation of [99] that begins with the anisotropic Wilson action in Euclidean time $\tau = it$ defined on a spacetime lattice:

$$S_E = -\beta_t \sum_t \text{Re Tr } U_t - \beta_s \sum_s \text{Re Tr } U_s \quad (1)$$

where $i = t, s$ refers to temporal and spatial plaquettes U_i formed from gauge links. We introduce anisotropy between the lattice spacings by using different bare couplings on the spatial and temporal plaquettes:

$$\beta_t(a, a_0) = \frac{a}{g_t^2(a, a_0)a_0}, \quad \beta_s(a, a_0) = \frac{a_0}{g_s^2(a, a_0)a}. \quad (2)$$

The renormalized anisotropic parameter $\xi \equiv a/a_0$ is used to denote the physical change in the lattice spacings caused by tuning the bare parameters β_t, β_s . To approach the Hamiltonian limit ($a_0 \rightarrow 0$), it becomes natural to introduce two new couplings, $g_H^2 = g_s g_t$, and the speed of light, $c = g_s g_t^{-1}$.

The connection to the Hamiltonian is via the transfer matrix, $T(a, a_0)$ which takes a state at time $\tau, |\tau\rangle$, to $|\tau + 1\rangle$. T is related to S_E by the partition function Z :

$$Z = \int D U e^{-S_E} = \text{Tr } T(a_0)^N \quad (3)$$

where N counts time slices. It follows that the matrix elements of $T(a_0)$ are [99]

$$\begin{aligned} \langle \tau + 1 | T(a_0) | \tau \rangle &= e^{\frac{\beta_s}{2} \sum_s \text{Re Tr } U_s} e^{\beta_t \sum_{\{\tau, \tau+1\}} \text{Re Tr } U_t} e^{\frac{\beta_s}{2} \sum_s \text{Re Tr } U_s} \\ &\equiv T_V^{1/2} T_K T_V^{1/2}, \end{aligned} \quad (4)$$

where we have used the second-order Trotterization. Since $T(a_0) \equiv e^{-a_0 H(a, a_0)}$, we desire $T(a_0)$ in terms of variables on one time slice. While this is trivial for U_s, U_t couples links at two times $U_{ij}(\tau), U_{ij}(\tau + 1)$. To proceed, we gauge-fix in the temporal gauge, $U_{0i} = \mathbb{1}$, yielding

$$S_K = -\beta_t \sum_{\{\tau, \tau+1\}} \text{Re Tr } U_{ij}(\tau) U_{ij}^\dagger(\tau + 1). \quad (5)$$

The next step is to express $T(a_0)$ in terms of operators. The link operator is simply $\hat{U}_{ij} |\tau\rangle = U_{ij} |\tau\rangle$. For T_K , we need an operator that evolves a link via $R_{ij}(g) |\tau\rangle = |\tau'\rangle$, where $U_{ij} \rightarrow g U_{ij}$. This operator has the property of $R_{ij}(g) R_{ij}(h) = R_{ij}(gh)$ and can define a conjugate momentum to \hat{U}_{ij} by performing a rotation on $U_{ij}(\mathbf{x}, \tau + 1)$. With this, we write

$$T_K = \prod_{\{ij\}} T_{K,ij} = \prod_{\{ij\}} \left[\int Dg R_{ij}(g) e^{\beta_t \text{Re Tr } g} \right], \quad (6)$$

where the product is over all spatial links $U_{ij}(\tau)$. Any group element equals $g = e^{i\omega \cdot \lambda}$ where λ_i are the adjoint generators, and $R_{ij}(g) = e^{i\omega \cdot l_{ij}}$ in terms of the generators l_{ij} of that representation. Defining $\prod_\alpha (D\omega^\alpha) J(\omega)$ as the invariant group measure with a Jacobian J , $T_K(a_0)$ is

$$T_{K,ij} = \int \prod_\alpha (D\omega^\alpha) J(\omega) e^{i\omega \cdot l_{ij}} e^{\beta_t \text{Tr } \cos(\omega \cdot \lambda)}. \quad (7)$$

Summing over all character functions of the group, this integral is can be performed analytically, requiring the Fourier transform over the group. This was used in [22] and seems to be a viable procedure when the group is finite. On the other hand, when dealing with continuous groups, the summation contains infinite character functions and is thus computationally impractical.

To remedy this obstacle in continuous groups, one expands T_K to $O(\omega^2)$ leaving Gaussian integrals. Neglecting the overall normalization, $T(a_0)$ is

$$T(a_0) = e^{\frac{\beta_s}{2} \sum_s \text{Re Tr } \hat{U}_s} e^{-\beta_t^{-1} \sum_{\{ij\}} l_{ij}^2} e^{\frac{\beta_s}{2} \sum_s \text{Re Tr } \hat{U}_s}. \quad (8)$$

Regardless of whether one approximates T_K or not, the integral corresponds to the Fourier transform of the gauge group. This transform which diagonalizes the kinetic energy is why we need a primitive quantum Fourier transform gate for the given group.

From Eq. (8), we use the definition of $T(a_0)$ to obtain a lattice Hamiltonian. However, since l_{ij} and \hat{U}_{ij} do not commute, rearranging this expression into a single exponential requires application of the Baker-Campbell-Hausdorff (BCH) formula, yielding:

$$H(a, a_0) = \frac{1}{c(a, a_0)a} \left(g_H^2(a, a_0) \sum_{\{ij\}} l_{ij}^2 - g_H^{-2}(a, a_0) \sum_s \text{Re Tr} \hat{U}_s - \frac{1}{24} \frac{1}{c^2(a, a_0)\xi^2} \sum_{\{ij\}, s} (g_H^2(a, a_0) [2l_{ij}^2, [l_{ij}^2, \text{Re Tr} \hat{U}_s]] - g_H^{-2}(a, a_0) [\text{Re Tr} \hat{U}_s, [l_{ij}^2, \text{Re Tr} \hat{U}_s]]) + \dots \right). \quad (9)$$

Taking the $a_0 \rightarrow 0$ limit of $T(a_0)$:

$$\mathcal{T}(\tau) \equiv \lim_{a_0 \rightarrow 0, N \rightarrow \infty} T(a_0)^N, \quad (10)$$

the BCH terms vanish and one obtains $H_{\text{KS}} \equiv -\frac{1}{\tau} \log(\mathcal{T}(\tau))$ [23]:

$$H_{\text{KS}} = \frac{1}{c(a)a} \left(g_H^2(a) \sum_{\{ij\}} l_{ij}^2 - \frac{1}{g_H^2(a)} \sum_s \text{Re Tr} U_s \right). \quad (11)$$

This is a common starting point for the evolution of lattice gauge theories on quantum computers. From this, we see that in order to simulate these gauge theories, there are a number of basic, group-dependent gates that can be used to simulate the two terms of Eq. (11). Along with the quantum Fourier transform for the kinetic term, for the potential term we need to be able to introduce a phase $\propto \text{Re Tr} U_s$. This is the origin of the need for the trace gate, the multiplication gate (needed to compute the plaquette U_s from the four links forming it), and the inversion gate (needed to uncompute the plaquette).

III. OVERVIEW OF BASIC GATES

In Ref. [22], the basic set of gates requires for a general gauge group G were given. This construction begins with the defining for G a qubit G -register by identifying each group element with a computational basis state $|g\rangle$, $g \in G$. For pure-gauge Hamiltonians, a set of useful primitive gates defined on the G -register are: inversion, multiplication, trace, and the quantum Fourier transform.

The inversion gate acts on a single G -register mapping each group element to its inverse. This is defined in the fiducial basis by

$$\mathbf{U}_{-1}|g\rangle = |g^{-1}\rangle. \quad (12)$$

The group (matrix) multiplication gate acts on two G -registers and is defined by

$$\mathbf{U}_{\times}|g\rangle|h\rangle = |g\rangle|gh\rangle. \quad (13)$$

Here we have defined \mathbf{U}_{\times} as implementing in-place left multiplication, in the sense that the content of the second register was multiplied on the left. Left multiplication suffices for a minimal set as right multiplication can be implemented using two applications each of \mathbf{U}_{-1} and \mathbf{U}_{\times} [22], although fidelity of simulations may be improved by directly implementing right multiplication as well.

The trace of a plaquette appears in H_{KS} , and so to perform this operation we combine the matrix multiplication gate with a single-register trace gate:

$$\mathbf{U}_{\text{Tr}}(\theta)|g\rangle = e^{i\theta \text{Re Tr} g}|g\rangle. \quad (14)$$

In our construction, the final gate required on the G -register is the Fourier transform gate \mathbf{U}_F . This gate acts on a G -register to rotate into the Fourier basis:

$$\mathbf{U}_F \sum_{g \in G} f(g)|g\rangle = \sum_{\rho \in \hat{G}} \hat{f}(\rho)_{ij} |\rho, i, j\rangle. \quad (15)$$

The second sum is taken over ρ , the representations of G , and \hat{f} denotes the Fourier transform of f . This gate diagonalizes what will be the ‘kinetic’ part of the Trotterized time-evolution operator. After application of the gate, the register is no longer a G -register but a \hat{G} -register.

In the subsequent section we consider quantum circuit implementations of these gates for the dihedral group $D_N = \{g = s^m r^k | s^2 = r^N = e\}$ generated by a reflection s and rotation r ; we review the important properties of D_N in the Appendix A. Following [22], the $2N$ group elements $s^m r^k$, $m \in \{0, 1\}$, $k \in \{0, \dots, N-1\}$, are encoded using standard binary in the qubit computational basis states $|m\rangle|k\rangle$, where the register $|k\rangle$ uses $\lceil \log_2 N \rceil$ qubits. We may variously refer to the $|m\rangle$ as the s -qubit or the reflection qubit, and the $|k\rangle$ as the r -register or the rotation register. In this paper, we focus exclusively on the case $N = 2^n$, so that in all we need $n+1$ qubits to encode all the elements of D_N .

IV. INVERSION GATE

Here we describe how to construct a circuit realizing the inversion gate $\mathcal{U}_{-1}|g\rangle = |g^{-1}\rangle$ for D_N . First, consider the case of a general discrete gauge group G . As observed in [22], if we have access to both the multiplication gate \mathcal{U}_\times and its reversed (adjoint) circuit $\mathcal{U}_\times^\dagger$, then we can implement \mathcal{U}_{-1} using an ancillary G -register initialized to the group identity element $|e\rangle$. We discuss construction of the multiplication gate in Sec. V. We can then implement \mathcal{U}_{-1} using the sequence of operations

$$|g\rangle|e\rangle \xrightarrow{\mathcal{U}_\times^\dagger} |g\rangle|g^{-1}\rangle \xrightarrow{\text{SWAP}} |g^{-1}\rangle|g\rangle \xrightarrow{\mathcal{U}_\times} |g^{-1}\rangle|e\rangle, \quad (16)$$

at which point the ancillary register has been returned to $|e\rangle$ and can be reused or discarded. We note that the SWAP may be performed virtually by simply switching (relabeling) the top and bottom registers in the circuit for \mathcal{U}_\times . Hence the cost of this implementation of \mathcal{U}_{-1} is at most twice that of \mathcal{U}_\times . Note that the property that the ancilla register is initialized and returned to a fixed state and can be used to further simplify the circuits for $\mathcal{U}_\times^\dagger$ so that fewer gates are required than for the general case. In any case, the use of ancillary G -register means this implementation requires at least $\log|G|$ additional qubits, with $\log|G| = n + 1$ for D_N .

Alternatively, one may use the properties of D_N (see Appendix A) to derive more specific constructions requiring fewer ancilla qubits and lower circuit depth. For this we use that the inverse of an element $s^m r^k$ is given by

$$(s^m r^k)^{-1} = s^m r^{(N-k)(1-m)+mk} \quad (17)$$

As a result, given the qubit encoding $s^m r^k \rightarrow |m\rangle|k\rangle$ described above, the effect of \mathcal{U}_{-1} is to change the state of the register $|0\rangle|k\rangle \rightarrow |0\rangle|N-k\rangle$, and leave $|1\rangle|k\rangle$ unmodified. Therefore, controlled on the state of the left-most qubit, we need to compute the 2's complement of the register $|k\rangle$. The 2's complement of an n -bit binary number is defined as its complement with respect to 2^n , so that the sum of the number and its complement equals $N = 2^n \pmod{0}$. It can be obtained by first taking its 1's complement, i.e., flipping all the 0s to 1s and 1s to 0s, and then adding 1 to the resulting integer.

Hence, controlled on $m = 0$, we apply a Pauli X gate (i.e., a CNOT) to each qubit in $|r\rangle$, followed by the increment operation. Treating the register $|r\rangle$ as an integer mod N , the increment operation can be implemented using simplified versions of standard quantum circuits for addition; various constructions with different tradeoffs in terms of size, depth, and number of ancilla qubits can be found in the literature, see in particular [[100] Table 1]. In terms of circuit depth, a straightforward modification of the constructions of [100,101] yields quantum circuits for the controlled-increment operation using $O(n)$ Toffoli gates

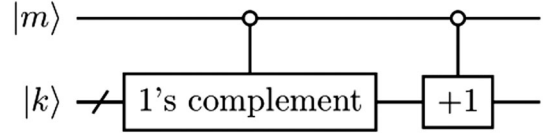


FIG. 2. Schematic quantum circuit implementation of the inversion gate \mathcal{U}_{-1} . The controlled 1's complement operation can be implemented with n CNOT gates. The controlled increment gate can be implemented with $O(n)$ Toffoli gates using a constant number of additional ancilla qubits [100,101].

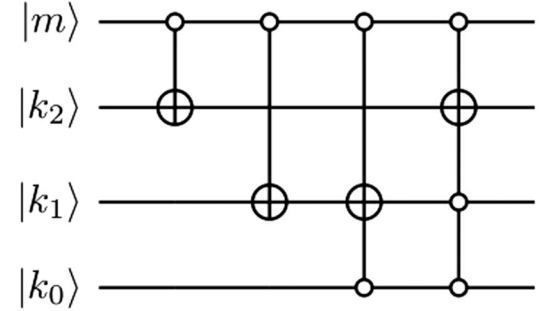


FIG. 3. Example: A simplified implementation of the inversion gate \mathcal{U}_{-1} for D_8 . The first two gates correspond to the 1's complement operation, and the last two to incrementation; an inner pair of CNOTs has canceled out.

and as few as 1 additional ancilla qubits. A schematic circuit for the inversion gate is shown in Fig. 2, and an example implementation is shown in Fig. 3.

V. MULTIPLICATION GATE

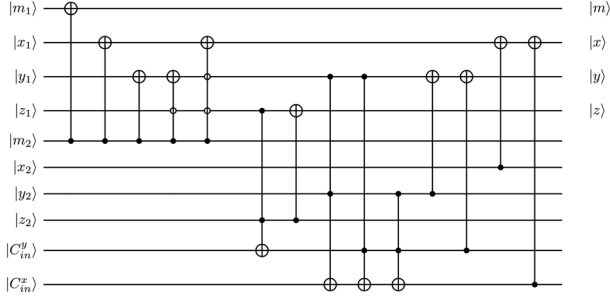
Here we describe how to build a circuit realizing the multiplication gate $\mathcal{U}_\times|g\rangle|h\rangle = |g\rangle|gh\rangle$ for the dihedral group. We employ the following group multiplication rule for elements of D_N

$$s^{m_1} r^{k_1} \cdot s^{m_2} r^{k_2} = s^{m_1+m_2} r^{Nm_2+(-1)^{m_2}k_1+k_2}, \quad (18)$$

which implies that it suffices to construct a circuit that performs either addition or subtraction depending on whether $m_2 = 0$ or $m_2 = 1$. Therefore, the task of realizing the D_N multiplication gate reduces to performing conditional binary arithmetic on qubits.

In the case where $m_2 = 0$, we must add k_1 and k_2 , whereas in the case where $m_2 = 1$, we must add the two's complement of k_1 and k_2 . The construction of the circuit to compute the two's complement is given in the section on the inversion gate IV, except that now we control on the value of the leading qubit being 1 instead of 0. Having conditionally prepared the two's complement of k_1 , we must then perform binary addition to complete the computation of Eq. (18).

A variety of proposed quantum algorithms for addition and multiplication exist in the literature [102,103] with

FIG. 4. A multiplication gate circuit for D_8 theory.

different resource tradeoffs, see, e.g., [[104] Table 1]. One approach is to use the classical full-adder, which takes inputs A and B , the two bits to be added, and C_{in} , the carry-in bit from the previous bit addition, and outputs the sum S and the carry-out C_{out} . In Reed-Muller form, these are given by

$$\begin{aligned} S &= A \oplus B \oplus C_{\text{in}} \\ C_{\text{out}} &= AB \oplus AC_{\text{in}} \oplus BC_{\text{in}}. \end{aligned} \quad (19)$$

If we choose to over-write one of the registers with the sum, say the register containing the A bits in the convention above, then we can compute S at every step using 2 CNOTs, one controlled on the value of B and the other on the value of C_{in} , with the target being A . Similarly, we can compute C_{out} and write out its value to an ancillary qubit at every step using 3 CCNOT gates. Therefore, for a D_{2^n} gauge theory, using this scheme we would require 2 CNOTs to compute the sum and 3 CCNOTs to compute the carry outs for each of $n - 2$ bits, in addition to $n - 1$ ancillary qubits to hold the value of the carries. We would only need 1 CNOT to compute the sum of the least significant bit, and 1 CCNOT to compute the carry-out for this bit. We also do not need to compute the carry-out of the most significant bit. Assuming 1 CCNOT \sim 6 CNOTs, in all this adds a cost of $20n - 31$ CNOTs in addition to the circuit to compute the two's complement in order to implement the multiplication gate.

An example implementation for the D_8 multiplication gate is shown in Fig. 4.

VI. TRACE GATE

Here we describe how to construct quantum circuits realizing the trace gate $\mathfrak{U}_{\text{Tr}}(\theta)|g\rangle = e^{i\theta\text{Re}(\text{Tr}g)}|g\rangle$ for D_N . Observe that, unlike the other basic gates we consider, this family of gates is parametrized by a real number θ . We describe both a straightforward implementation that scales with N , and is in principle exact, as well as a more complicated implementation that scales polynomially with $n := \log_2 N$, but generally comes with some degree of approximation error; we refer to these as *direct* and *ancilla-assisted* implementations, respectively. Though its implementation cost has worse asymptotic scaling, the direct construction may be especially useful when N is relatively

small, such as the important cases for near-term experiments described in Sec. I.

Here $\text{Tr}g$ corresponds to the matrix trace in the fundamental representation. We let H_{Tr} denote the diagonal Hamiltonian defined as $H_{\text{Tr}}|g\rangle = \text{Re}(\text{Tr}(g))|g\rangle$ such that $\mathfrak{U}_{\text{Tr}}(\theta)|g\rangle = e^{i\theta\text{Re}(\text{Tr}(g))} = e^{i\theta H_{\text{Tr}}}$.

For D_N , in the fundamental (two-dimensional) representation for each group element $g = s^m r^k$ we have

$$\rho(g) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^m \begin{pmatrix} \omega & 0 \\ 0 & \bar{\omega} \end{pmatrix}^k, \quad \text{where } \omega = e^{2\pi i/N}. \quad (20)$$

Clearly $\rho(g)$ is always traceless when $m = 1$. When $m = 0$ we have $\text{Tr}(g) := \text{Tr}(\rho(g)) = \omega^k + \bar{\omega}^k = 2 \cos(2\pi k/N)$, and so for each N the trace values uniformly sample from one period of $2 \cos(x)$. Hence for D_N we observe $\text{Re Tr}(g) = \text{Tr}(g)$. Therefore we have

$$H_{\text{Tr}} = |0\rangle\langle 0| \otimes \sum_{k=0}^{N-1} 2 \cos(2\pi k/N) |k\rangle\langle k|. \quad (21)$$

Direct implementation: We first consider implementation of $\mathfrak{U}_{\text{Tr}}(\theta)$ by directly simulating evolution under the Hamiltonian H_{Tr} for time θ . Diagonal Hamiltonians on n qubits can be uniquely written as $H = \sum_{\alpha} a_{\alpha} Z_{\alpha}$, where $Z_{\alpha} = Z_{\alpha_1} \dots Z_{\alpha_n}$ denotes a tensor product of Pauli Z operators indexed by subsets of qubits $\alpha \subset [n]$, with coefficients given by $a_{\alpha} = \frac{1}{2^n} \text{Tr}[Z_{\alpha} H] \in \mathbb{R}$ [105]. We may apply this decomposition to H_{Tr} , or, to take advantage of its tensor structure, to only the second factor on the right-hand side of Eq. (21), as desired; in general, however, the number of non-zero terms in such a decomposition is proportional to N . Nevertheless, for moderate N this decomposition yields an straightforward implementation of $\mathfrak{U}_{\text{Tr}}(\theta)$. For the latter case, i.e., writing $H_{\text{Tr}} = |0\rangle\langle 0| \otimes \sum_{\alpha} a_{\alpha} Z_{\alpha}$, we have

$$\mathfrak{U}_{\text{Tr}}(\theta) = \prod_{\alpha=0}^N e^{i\theta a_{\alpha} |0\rangle\langle 0| \otimes Z_{\alpha}} = \prod_{\alpha=0}^N \Lambda_{m=0}(e^{i\theta a_{\alpha} Z_{\alpha}}), \quad (22)$$

where $\Lambda_{m=0}(e^{i\theta a_{\alpha} Z_{\alpha}})$ denotes the controlled unitary implementing $e^{i\theta a_{\alpha} Z_{\alpha}}$ conditioned on the first qubit being zero, and we used the fact that diagonal terms mutually commute. Each controlled rotation can be implemented with $O(n)$ basic gates consisting of CNOTs and single-qubit gates [105,106]; however the number of rotations may be proportional to $N = 2^n$. An advantage of this approach is that if qubit rotations can be implemented exactly then so can $\mathfrak{U}_{\text{Tr}}(\theta)$. If we tolerate approximation error in $\mathfrak{U}_{\text{Tr}}(\theta)$ the gate costs can be further reduced [107].

For example, consider D_4 , for which Eq. (20) gives $\rho(g) = X^m (iZ)^k$, which has trace 2 for $g = e$ and -2 for $g = r^2$, else 0. Hence we have the Hamiltonian $H_{\text{Tr}} = 2|000\rangle\langle 000| - 2|010\rangle\langle 010|$ which we may

write as $H_{\text{Tr}} = \frac{1}{2}(Z_{k_1} + Z_m Z_{k_1} + Z_{k_1} Z_{k_0} + Z_m Z_{k_1} Z_{k_0})$, or with control as $H_{\text{Tr}} = |0\rangle\langle 0| \otimes (Z_{k_1} + Z_{k_1} Z_{k_0}) = 2|0\rangle\langle 0| \otimes Z_{k_1} \otimes |0\rangle\langle 0|$. So for D_4 we see that we can implement U_{Tr} exactly with a double-controlled Z rotation, or a controlled Z and controlled ZZ rotation, or a combination of Z , two ZZ , and a ZZZ rotations.

Ancilla-enabled implementation: On the other hand, when N becomes large it is desirable to have a quantum circuit for the trace gate with resource costs that scales polynomially with n as opposed to $N = 2^n$. This can be accomplished if we accept tradeoffs such as the use of ancilla qubit registers and some degree of approximation error. Here the basic idea is that we use the ancilla registers as scratchpad space for quantum arithmetic circuits that coherently compute the trace value for each group element, upon which we apply controlled rotation gates to achieve the desired phase kickback. Clearly, for real numbers any finite size ancilla register will lead to some degree of approximation error, in general, in the computed values and resulting phases. This error may be systematically reduced by employing larger ancilla registers and circuits that utilize higher precision numbers; we leave a detailed analysis of these time, space, and precision tradeoffs for future work.

Let us first consider restricting the required trigonometric quantities to the first quadrant $0 \leq 2\pi\ell/N < \pi/2$ which will simplify construction of the resulting quantum circuits. In particular, many approximations for computing numerical functions come with guaranteed precision only over such a bounded interval, and moreover some trigonometric algorithms proceed by computing values of \cos and \sin simultaneously. Observe that for each group element $|g\rangle = |mk_{n-1}\dots k_1 k_0\rangle$ we have $\text{Tr}(g) = 2(1-m)\cos(2\phi k/N)$, and so the periodicity of the cosine function implies that the bit k_{n-1} controls the sign of the coefficient and the bit k_{n-2} controls the ‘‘phase,’’ i.e., explicitly $\text{Tr}(0k_{n-1}0k_{n-3}\dots k_1 k_0) = 2(-1)^{k_{n-1}}\cos(2\pi k'/N)$ and $\text{Tr}(0k_{n-1}1k_{n-3}\dots k_1 k_0) = -2(-1)^{k_{n-1}}\sin(2\pi k'/N)$, where k' is the integer given by the bits $k_{n-3}\dots k_1 k_0$. (Note that a similar treatment of the first 3 bits may be employed in the direct case above for the D_4 example.)

Assume for the moment we can implement the desired quantum arithmetic modules for computing fixed-precision trigonometric functions to b bits of accuracy after the decimal point [102]. Then \mathcal{U}_{Tr} can be implemented as follows:

- (1) (Compute classical functions.) Append a sufficiently large ancilla register $|00\dots 0\rangle$ and reversibly compute (in superposition) the transformation for each basis state $|g\rangle = |mk_{n-1}k_{n-2}k'\rangle$

$$\begin{aligned} &|mk_{n-1}k_{n-2}k'\rangle|0\dots 0\rangle \\ &\rightarrow |mk_{n-1}k_{n-2}k'\rangle|\sin(\widetilde{2\pi k'/N})\rangle|\cos(\widetilde{2\pi k'/N})\rangle|sp\rangle, \end{aligned}$$

where \tilde{x} denotes a b -bit binary approximation of a quantity $0 \leq x < 1$. The remaining scratchpad register, $|sp\rangle$, denotes intermediate classical values which will be used to facilitate uncomputation. We discuss how this may be implemented below.

- (2) (Phase kickback.) Given a b -bit quantity $0 \leq x < 1$ we can implement $|x\rangle \rightarrow e^{i\theta x}|x\rangle$ (up to an irrelevant global phase) using a controlled $R_Z(\theta 2^{-j})$ gate applied to each j th bit of $|x\rangle$, $j = 1, \dots, b$, such that the number of such gates is b . The single-qubit Z rotation gate is defined as $R_Z(\phi) = e^{-i\phi Z/2}$. Similarly, we can implement $|x\rangle \rightarrow e^{i2\theta x}|x\rangle$ applying instead controlled $R_Z(\theta 2^{1-j})$ gates. Hence we apply two high-level unitaries that kickback a phase of $\theta \text{Tr}(g)$ to each basis state, as schematically depicted in Fig. 5:
- (3) (Uncomputation.) As the operation of the second step is diagonal, we can restore the ancilla qubits to $|00\dots 0\rangle$ then discard for reuse by applying the reverse of the circuit in Step 1. Hence each input basis state is taken to the desired state

$$|g\rangle = |mk\rangle \rightarrow e^{i\theta 2(1-m)\cos(2\pi k/N)}|mk\rangle = e^{i\theta \text{Tr}(g)}|g\rangle.$$

The cost of Steps 1 and 3 depend on the arithmetic subroutine used and the number b of bits of accuracy in the \sin and \cos registers. This cost dominates that of Step 2 which depends linearly on b . Here we have assumed the ancilla qubits are restored to $|00\dots 0\rangle$ for reuse; we note that the allocation and uncomputation of ancilla qubit resources may often be significantly optimized within the context of an overall algorithm [102].

Computing the trigonometric functions: Methods for computing the \cos and \sin functions using quantum arithmetic circuits are discussed in [102,103,108,109]. Different approaches come with different tradeoffs in terms of the number of qubits, number and types of basic gates, and required numerical accuracy for a given application.

The approach of [[108] Sec. V and App. 2] requires only addition and multiplication operations, and simultaneously computes both \sin and \cos using repeated squaring via the approximation

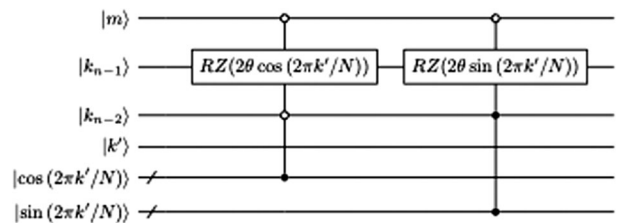


FIG. 5. Schematic for phase kickback of $\text{Tr}(g)$ using registers containing the required \cos and \sin values. Here the two schematic gates represent one multicontrolled- Z rotation for each bit in the \sin/\cos registers, respectively, i.e., the gate $R_Z(\theta 2^{1-j})$ is applied for each j th bit, $j = 1, \dots, b$, encoding the 2^{-j} bit.

$$e^{i\theta a} = \cos(a) + i \sin(a) = (e^{i\theta a/R})^R \\ \simeq (1 - i\theta a/R - (\theta a/R)^2/2)^R \quad (23)$$

such that the quantity $1 - i\theta a/R - (\theta a/R)^2/2$ is computed by storing separately its real and imaginary parts, for a suitable $R = 2^r \gg 1$ selected with respect to the accuracy parameter b . Repeatedly squaring this quantity (requiring only r operations) then yields the desired sin and cos approximations. Roughly, the error in these approximations goes as $1/R$ for sufficiently many bits of accuracy in the sin and cos registers as well as the intermediate quantities (cf. [[108], Prop. 1 and 2]).

Alternatively, the approach of [[103] Appendix D] uses piecewise polynomial approximations implemented via controlled Horner polynomial evaluation (such that each degree d polynomial approximation requires $d + 1$ additions and multiplications), while the approach of [109] employs nontrivial quantum submodules for approximately computing square roots. In these approaches care must be taken to ensure the desired accuracy is achieved.

Hence the ancilla-assisted approaches yield quantum circuits with resource costs scaling as low-degree polynomials in $n = \log N$ and the accuracy bits b . The specific cost in terms of gates and ancillas depends on these parameters and the particular quantum arithmetic circuits employed as subroutines. As stated the direct approach is much simpler for moderate N ; we show an explicit quantum circuit for this implementation of $\mathcal{U}_{\text{Tr}}(\theta)$ for D_4 and its compilation to hardware gates in Fig. 10 below.

VII. FOURIER GATE

The standard n -qubit quantum Fourier transform [110], a critical component of Shor's prime factoring algorithm, corresponds to the Abelian group \mathbb{Z}_{2^n} . Quantum circuits implementing Fourier transforms over a variety of non-Abelian groups have been considered in [111–114], though there remains important groups for which efficient QFT circuits are not known [115].²

Here we consider the explicit construction of quantum circuits for the QFT on D_N . Our construction employs the standard QFT as a subroutine. We note that the more general construction of [111] for efficient circuits for QFTs over metacyclic groups also includes D_N .

The Fourier transform of a representation of some finite group G is defined as

²We note that an efficient quantum circuit for the QFT of a group G does not entail an efficient quantum algorithm for the corresponding hidden subgroup problem (HSP) for G , an important class of problems that includes both the prime factoring and graph isomorphism problems [115]. Subexponential time quantum algorithms for the HSP on D_N are given in [116,117] using the standard QFT rather than the D_N one considered here.

$$\hat{f}(\rho) = \sqrt{\frac{d_\rho}{N}} \sum_{g \in G} f(g) \rho(g), \quad (24)$$

where $N = |G|$, d_ρ is the dimensionality of the representation ρ , and f is a function over G . The inverse transform is given by

$$f(g) = \frac{1}{\sqrt{N}} \sum_{\rho \in \hat{G}} \sqrt{d_\rho} \text{Tr}(\hat{f}(\rho) \rho(g^{-1})), \quad (25)$$

where the dual \hat{G} is the set of all irreducible representations (irrep) of G . Note, if there exists a subgroup $H \subset G$ and elements $\{g_i\}_{i=1}^n$ such that we can write $G = \cup_{i=1}^n g_i H$, i.e., a *left transversal* of H exists in G , then

$$\sum_{g \in G} f(g) \rho(g) = \sum_{i=1}^n \sum_{h \in H} f(g_i h) \rho(g_i h) \\ = \sum_{i=1}^n \rho(g_i) \sum_{h \in H} f_i(h) \rho(h) = \sum_{i=1}^n \rho(g_i) \hat{f}_i(\rho|_H) \quad (26)$$

where we have defined $f_i(h) = f(g_i h)$, $\rho|_H$ denotes the restriction of the representation ρ to the subgroup H , and \hat{f}_i represents the Fourier transform of the function f_i . Using this, we can compute the Fourier transform \hat{f} on the representation ρ in a recursive manner for the series of subgroups H_1, \dots, H_n that form a chain $G \supset H_1 \supset \dots \supset H_n = id$, using “adapted bases” such that $\rho|_{H_i}$ can be written as a direct sum of irreps of H_i .

Let $\alpha: G \rightarrow \mathbb{C}$ and $\alpha_i(g) \equiv \alpha(g_i g)$, where $g_i, g \in G$. Then, we may similarly construct quantum Fourier transforms (QFTs) via the following series of operations

$$|\psi\rangle = \sum_{g \in G} \alpha(g) |g\rangle = \sum_{i=1}^n \sum_{h \in H} \alpha(g_i h) |g_i\rangle |h\rangle \\ = \sum_{i=1}^n |g_i\rangle \left(\sum_{h \in H} \alpha_i(h) |h\rangle \right) \xrightarrow{F_H} \sum_{i=1}^n |g_i\rangle \left(\sum_{\tilde{h} \in \tilde{H}} \hat{\alpha}_i(\tilde{h}) |\tilde{h}\rangle \right) \\ \xrightarrow{U} \sum_{\tilde{g} \in \tilde{G}} \hat{\alpha}(\tilde{g}) |\tilde{g}\rangle = |\tilde{\psi}\rangle \quad (27)$$

where F_H denotes the Fourier transform over the subgroup H , and U denotes a change of basis from $T \otimes B_H$ to B_G , where T denotes the coset representatives $\{g_i\}_{i=1}^n$ and B_H (B_G) denotes the Fourier basis of the group H (respectively G). In our encoding of D_N elements $s^m r^k \rightarrow |g\rangle = |m\rangle |k\rangle$, $|k\rangle$ encode the basis elements of \mathbb{Z}_N , while $|\tilde{k}\rangle$ denote the Fourier basis of \mathbb{Z}_N . Then, we have $F_H: |m\rangle |k\rangle \rightarrow |m\rangle |\tilde{k}\rangle$. Likewise, denoting the Fourier basis of D_N by $|\tilde{g}\rangle$, the final transformation is $U: |m\rangle |\tilde{k}\rangle \rightarrow |\tilde{g}\rangle$. Determining U is often the more nontrivial part of any such QFT algorithm.

For even N , the group D_N has the following four 1-dimensional irreps using $m \in \{0, 1\}$ and $k \in \{0 \dots N - 1\}$:

- (i) $\rho_A: r^k \rightarrow 1, sr^k \rightarrow 1$
 - (ii) $\rho_B: r^k \rightarrow 1, sr^k \rightarrow -1$
 - (iii) $\rho_C: s^m r^k \rightarrow 1$ for even k ; $s^m r^k \rightarrow -1$ for odd k
 - (iv) $\rho_D: r^k \rightarrow 1, sr^k \rightarrow -1$ for even k ; and $r^k \rightarrow -1$, and $sr^k \rightarrow 1$ for odd k ,
- and $\frac{N-2}{2}$ 2-dimensional irrep:

$$\phi^{(l)}(s^m r^k) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^m \begin{pmatrix} e^{i2\pi l/N} & 0 \\ 0 & e^{-i2\pi l/N} \end{pmatrix}^k, \quad (28)$$

$$U: \left| mN + p\frac{N}{2} + x \right\rangle \rightarrow \begin{cases} |mN + p\frac{N}{2} + x\rangle, & 1 < x < \frac{N}{2} \\ (e^{i\pi\frac{N}{2}})^{pm} \frac{1}{\sqrt{2}} \sum_{j=0}^1 (e^{i\pi})^{jm} |jN + p\frac{N}{2} + x\rangle, & x = 0 \end{cases} \quad (29)$$

where $m, p \in \{0, 1\}$ are the 2 most significant bits, while $x \in \{0, \dots, \frac{N}{2} - 1\}$ specifies the state of the remaining part of the register. The complete circuit for \mathcal{U}_F for D_N ($N = 2^n$ for some n) is given in Fig. 6. There, we use the operation $\Phi(\omega)|u\rangle|v\rangle = \omega^{uv}|u\rangle|v\rangle$, with $\omega = e^{i\pi\frac{N}{2}}$. In general, if u takes on n_1 values and v takes on n_2 values, then Φ can be compiled using $\Theta(\lceil \log(n_1) \rceil \lceil \log(n_2) \rceil)$ gates. In our case however, m and p only take on two values each, and we can therefore compile this operation using a single CCPHASE gate and an ancillary qubit, as shown in Fig. 7. With this formulation, we reduce the gate costs for D_4 from five entangling gates in [22] to two.

Upon the execution of the Fourier gate, the four 1-dimensional irreps of D_N (with $N = 2^n$) are encoded into the following basis states

$$\begin{aligned} \rho_A &\rightarrow |00\rangle|0\rangle^{\otimes n-1}, & \rho_B &\rightarrow |10\rangle|0\rangle^{\otimes n-1} \\ \rho_C &\rightarrow |01\rangle|0\rangle^{\otimes n-1}, & \rho_D &\rightarrow |11\rangle|0\rangle^{\otimes n-1} \end{aligned}$$

while the matrix entries $\rho_{ij}^{(l)}$, the i th row and j th column of the 2-dimensional irreps indexed by $l \in [1, \frac{N}{2}]$ and given by Eq. (28) are encoded into the remaining computational

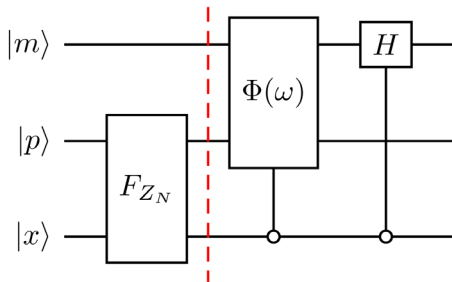


FIG. 6. Circuit computing \mathcal{U}_F of the D_N group. The first part of the circuit computes the Fourier transform F_{Z_N} over $Z_N = \{r^0, \dots, r^{N-1}\}$, while the latter part performs a change of basis $|g_i\rangle|\tilde{h}\rangle \rightarrow |\tilde{g}\rangle$, implementing the unitary transform given in Eq. (29).

where $1 \leq l < \frac{N}{2}$. D_N has a cyclic subgroup $Z_N = \{r^0, \dots, r^{N-1}\}$ for which the QFT is well known [110], and for which the elements $\{e, s\}$ provide a left transversal in D_N . Our encoding of D_N elements into qubits, $s^m r^k \rightarrow |m\rangle|k\rangle$, and the existence of the QFT over Z_N provide all the steps in Eq. (27) to compute the QFT over D_N except the last one involving a change of basis. This nontrivial step is provided by [111]

basis states as $\rho_{ij}^{(l)} \rightarrow |ij\rangle|l\rangle$. If so desired, one could rearrange the representations to appear in a different order, e.g., the first four computational basis states $|\dots 00\rangle, |\dots 01\rangle, |\dots 10\rangle, |\dots 11\rangle$ encoding the four 1-dimensional irreps, the next four encoding the matrix entries of the $l = 1$ 2-dimensional irrep and so on. In principle, the amplitudes of any two basis states $|s\rangle$ and $|s'\rangle$ could be exchanged by using an ancillary qubit $|t\rangle$ and applying an $(n+1)$ -qubit controlled operation $C^{n+1}(s) = |s\rangle\langle s| \otimes X + \sum_{s \neq s'=0}^{2^{n+1}-1} |s'\rangle\langle s'| \otimes \mathbb{I}$ on a single target qubit $|t\rangle$, followed by at most $n+1$ CNOTs controlled on $|t\rangle$ to change $|s\rangle$ to $|s'\rangle$.

However, for us, this is unnecessary since to apply the kinetic gate, we only ever need to apply the Fourier gate to transform to the momentum basis, and thereupon apply a diagonal operator, followed by the inverse of the Fourier gate to move back to position basis. In Appendix B, we prove that the Fourier gate diagonalizes the kinetic gate for D_N theory (even N), satisfying our requirement.

VIII. EXPERIMENTAL RESULTS

In this section, we discuss experimental results from running realizations of the circuits described above on the Rigetti Aspen-9 QPU, which features 32 transmon qubits with a square-octagon topology [118–120] (see Fig. 8). The Rigetti stack [121] allows us to use the Quil language [122]

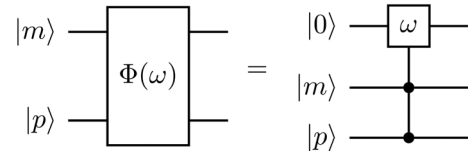


FIG. 7. Circuit for implementing $\Phi(\omega)$. This is given by a simple application of a CCPHASE gate targeted on an ancillary qubit, obtaining a phase kick-back on the qubits $|k\rangle$ and $|i\rangle$, on which it is controlled.

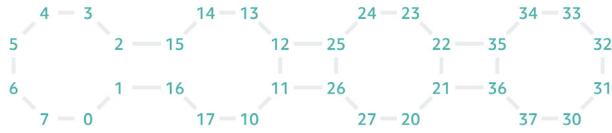


FIG. 8. The Rigetti Aspen-9 QPU consists of 32 transmon qubits arranged in a square-octagon lattice. Single-qubit gates $\{\text{RZ}(\theta), \text{RX}(k\pi/2)\}$ are available for the qubits denoted by the integers. Native two-qubit gates (CPHASE, CZ, XY) can be applied between the connected qubits with the linkage denoted by gray lines.

to program the Aspen-9 device, and its associated optimizing compiler Quilc [123] to compile fundamental gates into its native gateset $\{\text{RZ}(\theta), \text{RX}(k\pi/2), \text{CPHASE}, \text{CZ}, \text{XY}\}$. A recently realized native gate CCPHASE [98] is also accessible using the Quil language. We report the process fidelities of the Fourier, inversion and trace gates for D_4 theory. The multiplication gate for D_4 involves a 6-qubit circuit, performing process tomography on which is experimentally costly. Instead, we compute the fraction of correct bitstrings the gate produces for all possible pairs of input bitstrings, and report this as the *accuracy* of this operation as a proxy to its fidelity. We find all the gates to have greater than 80% fidelity or accuracy.

A. D_4 multiplication gate

Concretely, for D_4 theory, we can use the encoding $sr^j \rightarrow |s\rangle|j\rangle = |a\rangle|bc\rangle$ as in [22] to specify an element of D_4 as $s^a r^{2b+c}$. We compute the product $|abc\rangle = |(a_1 b_1 c_1) \cdot (a_2 b_2 c_2)\rangle$, using the multiplication gate $\mathcal{U}_\times |a_1 b_1 c_1\rangle |a_2 b_2 c_2\rangle = |a_1 b_1 c_1\rangle |abc\rangle$. Whether we perform subtraction or addition, the rightmost bit will simply be given by $c = c_1 \oplus c_2$. The relation is similar for the leftmost bit (a).

For the second rightmost bit (b), we must first mod-2 sum both bits involved in the product, $b_1 \oplus b_2$. However, we must also account for the carry from (to) the mod-2 addition (subtraction) of the rightmost bit. Depending on whether we perform addition or subtraction, the appropriate carry is either $c_1 c_2$ or $c_1 \bar{c}_2$ respectively. Thus, in all, we have the two following rules.

For $m_2 = 0$ (addition), we obtain

$$a = a_1 \oplus a_2, \quad b = b_1 \oplus b_2 \oplus c_1 c_2, \quad c = c_1 \oplus c_2 \quad (30)$$

For $m_2 = 1$ (subtraction), we obtain the product

$$a = a_1 \oplus a_2, \quad b = b_1 \oplus b_2 \oplus c_1 \bar{c}_2, \quad c = c_1 \oplus c_2. \quad (31)$$

In circuit form, this is provided in Fig. 9. We implement this on the Rigetti Aspen-9 QPU, whose lattice topology is shown in Fig. 8. In order to minimize the number of SWAPs necessary to compile the circuit onto the native hardware, we use a 6-qubit sub-lattice consisting of the identifications

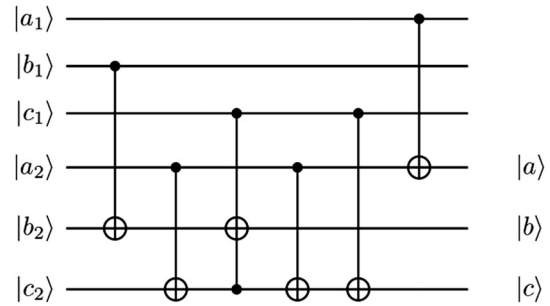


FIG. 9. Multiplication gate circuit for D_4 theory.

$(a_1, b_1, c_1, a_2, b_2, c_2) = (22, 30, 35, 21, 37, 36)$. This identification ensures only nearest-neighbor interactions in the implementation of the gate. In addition to 2-qubit gates such as CPHASE [120] and XY [119], the Rigetti hardware also allows the use of 3-qubit gates [98]. This can be used to compile the Toffoli gate with a single application of the CCPHASE gate, up to a few single-qubit gates.

In order to benchmark the multiplication gate, we start with each possible pair of 3-bitstrings, apply the multiplication gate, and obtain the fraction of correct bitstrings that we measure as output from a total of 10,000 shots. Using only 2-qubit gates to compile the Toffoli in Fig. 9, we can obtain some depth reduction by identifying a CNOT followed by a SWAP operation with a single XY gate (upto single-qubit gates) as described in [124]. Using this approach, the average fraction of correct output bitstrings, over all possible input pairs of 3-bitstrings, is found to be $\sim 0.19(6)$ where the standard deviation is reported in parenthesis. However, if we use the native CCPHASE gate to compile the Toffoli in the multiplication gate, the average fraction of correct output bitstrings goes up to $\sim 0.89(18)$. If we instead take the majority vote of 200 successive shots, we boost the average fraction of correct output bitstrings even more to $\sim 0.91(15)$.

B. D_4 trace gate and Fourier gate

To benchmark the fidelities of \mathcal{U}_F and $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ of the D_4 theory, we carry out quantum process tomography (QPT) [110]. To minimize SWAP gates due to Aspen-9 connectivity, we swap the qubit ordering for a linear connectivity and implement the circuits on the qubits (17, 10, 11) of Aspen-9. We also allow the qubit ordering to be different at the beginning and at the end of the circuits as shown in Fig. 10(a) and (c). The circuits are compiled into the native gate set $\{\text{RZ}(\theta), \text{RX}(k\pi/2), \text{CPHASE}, \text{CZ}, \text{XY}\}$ by Quilc, and need 2 and 4 two-qubit CPHASE gate, respectively as seen in Fig. 10(b) and (d).

QPT measures the process fidelities of \mathcal{U}_F and $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ to be 0.920 and 0.857. The process infidelity is dominated by the error of the two-qubit CPHASE gates which are calibrated to be around 2% to 3% at the time of the experiments. The χ matrices measured with 8000 shots

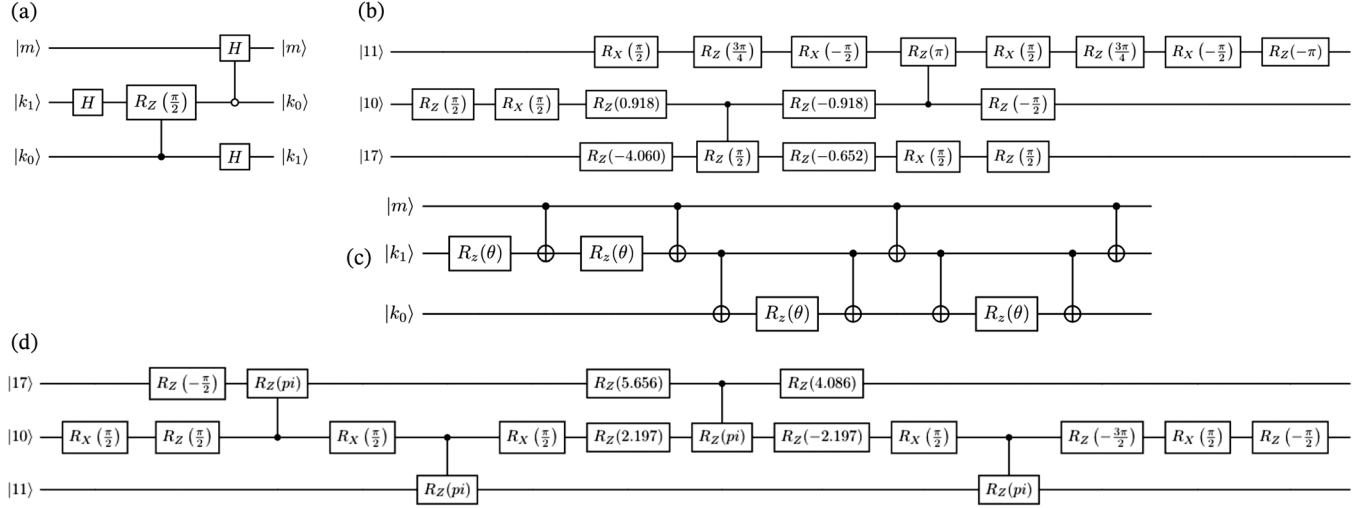


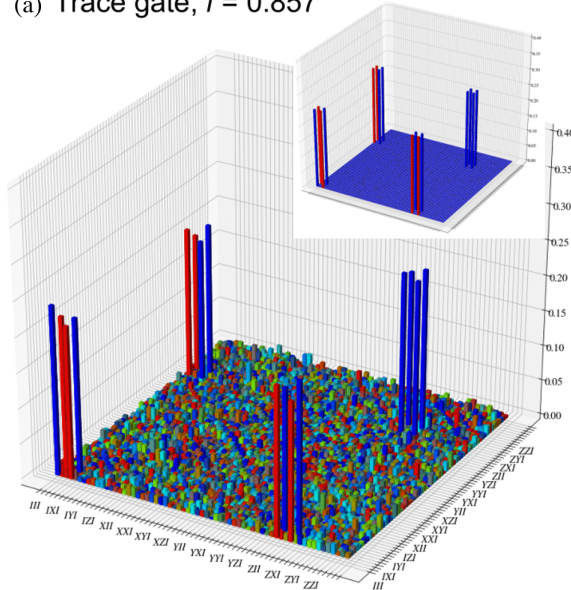
FIG. 10. \mathcal{U}_F and $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ for D_4 theory and their compiled versions for Aspen-9 QPU with qubits (11, 10, 17). We use the Quilc compiler to convert the circuits to versions using only native gates of Aspen-9 QPU. (a) D_4 Fourier gate (b) Compiled Fourier gate for Aspen-9 (c) D_4 trace gate (d) Compiled trace gate for Aspen-9.

are shown in Fig. 11 with the inset ideal matrices. Readout error mitigation is implemented by modeling the readout error as a classical stochastic process characterized by a confusion matrix, which can be determined by preparing all bit strings $|000\rangle, |001\rangle, \dots, |111\rangle$ and measuring the output. Any distribution is then postprocessed by inverting the confusion matrix to mitigate the readout error. More details of the readout error mitigation can be found in Refs. [125,126].

C. D_4 inversion gate

As described in Sec. IV, in order to construct the inversion gate, we need to apply the 2's complement (neglecting the leading bit) of the *rotation* register controlled on the value of the *reflection* qubit being 0. The only nontrivial operations the D_4 inversion gate therefore has are $|001\rangle \rightarrow |011\rangle$ and $|011\rangle \rightarrow |001\rangle$. This operation can be implemented using a single $\text{CCPHASE}(\pi)$ gate [98], with a

(a) Trace gate, $f = 0.857$



(b) Fourier gate, $f = 0.920$

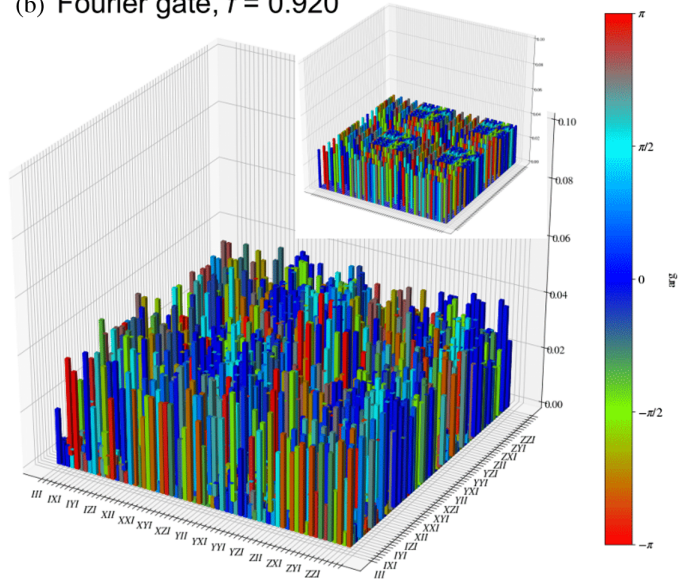
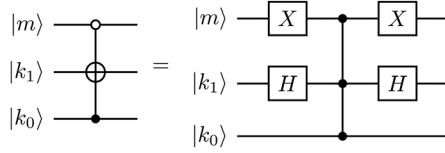


FIG. 11. The χ matrices associated with $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ and \mathcal{U}_F . The process fidelity is determined by $f = \text{Tr}(\chi_{\text{target}}^\dagger \chi)$, where the target χ_{target} is computed by noiseless simulator (see insets). $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ has a lower fidelity $f = 0.857$ compared to that of \mathcal{U}_F being $f = 0.920$ since $\mathcal{U}_{\text{Tr}}(\theta = \pi/2)$ consists of two more CZ gates. While the process tomography involves pairs of all 4^3 Pauli operators, to avoid overcrowding, we only display the labels of every four of the operators in the figures above. (a) Trace gate, $f = 0.857$ (b) Fourier gate, $f = 0.920$.

FIG. 12. Inversion gate for D_4 theory.

few additional single-qubit gates, as shown in Fig. 12. The process fidelity of the CCPHASE gate is computed to be $\sim 87.1\%$ on the Aspen-9 sublattice (10,11,12) (see Fig. 8) using cycle benchmarking [127].

IX. CONCLUSIONS

In this paper, we have shown how to construct quantum circuits for the simulation of arbitrary D_{2^n} gauge theories. The operations were shown to reduce to simpler ones such as computing the two's complement, or binary arithmetic, and therefore benefit from the wide variety of techniques used to implement such operations. The Fourier gate was shown to assume a particularly simple form. All these operations were shown to scale as $O(n)$, or as a low-degree polynomial in n in the case of the trace gate, providing an exponential advantage over classical state vector simulation. Experimentally, we found the success rate of the various operations for D_4 theory to be greater than or equal to roughly 80% on Rigetti's Aspen-9 quantum processor. These findings provide encouragement that large scale lattice simulations of gauge theories are within reach.

Looking to the future, several directions warrant mention. The first would be to extend the construction of primitive gates to gauge theories beyond D_{2^n} , in particular to the crystal-like subgroups of $SU(N)$ theories. The second would be to perform a detailed resource analysis both on the individual gates and algorithms for state preparation [128,129] and extracting physical observables [130,131] from simulations on specific architecture. Another followup would investigate the performance of these gates and their combinations on current devices.

ACKNOWLEDGMENTS

We would like to thank the Rigetti team for useful feedback and assistance with running experiments on the Aspen-9 processor, particularly Alex Hill, Mark Hodson, Bram Evert, Nicolas Didier, and Matt Reagor. We are grateful for support from NASA Ames Research Center. This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Superconducting Quantum Materials and Systems Center (SQMS) under Contract No. DE-AC02-07CH11359. S.H., and also M. S. A. for work done after the initial posting of this paper on arXiv, were supported by the NASA Academic Mission Services, Contract No. NNA16BD14C. Fermilab is

operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

APPENDIX A: ALGEBRAIC PROPERTIES OF DIHEDRAL GROUPS

Here, we note a few important properties of D_N , the *dihedral group* of symmetries of a regular N -sided polygon, which is generated by two elements: r (a rotation) and s (a reflection) such that $r^N = s^2 = e$, the identity element. Here N can be any positive integer, and it follows that each D_N is isomorphic to the semidirect product of cyclic groups $\mathbb{Z}_N \rtimes \mathbb{Z}_2$. Each of the $2N$ elements of D_N can be uniquely expressed as $s^m r^k$, where $m \in \{0, 1\}$ and $k \in \{0, 1, \dots, N-1\}$. The two generators satisfy the property $srs = r^{-1} = r^{N-1}$, or equivalently $sr = r^{N-1}s$, which in geometric terms means that a mirror reflection of a rotation gives a rotation in the opposite direction. Observe that this implies

$$sr^{k+1} = r^{N-1}sr^k = \dots = r^{N-(k+1)}s \quad (\text{A1})$$

so that by induction we have $sr^k = r^{N-k}s$ for $k \in \{0, \dots, N-1\}$. These properties can be summarized as

$$s^m r^k = r^{Nm+(-1)^m k} s^m. \quad (\text{A2})$$

Through a similar calculation we also have

$$r^k s^m = s^m r^{Nm+(-1)^m k}. \quad (\text{A3})$$

Using the above we find the product rule of Eq. (18):

$$\begin{aligned} s^{m_1} r^{k_1} \cdot s^{m_2} r^{k_2} &= s^{m_1} s^{m_2} r^{Nm_2+(-1)^{m_2} k_1} r^{k_2} \\ &= s^{m_1+m_2} r^{Nm_2+(-1)^{m_2} k_1+k_2}. \end{aligned} \quad (\text{A4})$$

From Eq. (A2) we the inverse of sr^k to be

$$(sr^k)^{-1} = r^{N-k} s = sr^k \quad (\text{A5})$$

while the inverse of r^k is simply $(r^k)^{-1} = r^{N-k}$ so that in general, the inverse of a D_N element is given by Eq. (17), i.e., $(s^m r^k)^{-1} = s^m r^{(N-k)(1-m)+mk}$.

APPENDIX B: PROOF THAT FOURIER GATE DIAGONALIZES D_N (EVEN N) KINETIC GATE

Given $M_{ij} = \text{Re}[\text{Tr}(\rho^\dagger(g_i)\rho(g_j))]$ and the matrix T with entries $T_{ij} = \exp \beta M_{ij}$, here we show that FTF^\dagger , where F is the unitary matrix corresponding to the non-Abelian Fourier transform, is diagonal. Moreover, we provide an explicit form of this diagonal matrix for arbitrary D_N (even N).

Note that M is dependent on the representation we use. There are 4 1D irreps of the D_N (even N) group,

- (i) $\rho_A: r^k \rightarrow 1, sr^k \rightarrow 1.$
- (ii) $\rho_B: r^k \rightarrow 1, sr^k \rightarrow -1.$
- (iii) $\rho_C: s^m r^k \rightarrow 1$ for even $k; s^m r^k \rightarrow -1$ for odd $k.$
- (iv) $\rho_D: r^k \rightarrow 1, sr^k \rightarrow -1$ for even $k; and $r^k \rightarrow -1, and $sr^k \rightarrow 1$ for odd $k,$$$

where $m \in \{0, 1\}$ and $k \in \{0 \dots N-1\}$, and $\frac{N}{2}-1$ 2D irreps,

$$\phi_{(l)}(s^m r^k) = X^m \begin{pmatrix} e^{i2\pi lk/N} & 0 \\ 0 & e^{-i2\pi lk/N} \end{pmatrix} \quad (\text{B1})$$

with $1 \leq l < N/2$ and $0 \leq k < N$. We work with the $l=1$ 2D irreps, and denote $\phi(s^m r^k) \equiv \phi_{(1)}(s^m r^k)$ in what follows for simplicity. Then, letting $i \equiv (m', k') = Nm' + k', j \equiv (m, k) = Nm + k$, it is clear that M_{ij} can be nonzero only when $m = m'$. In this case, we see that

$$\phi^\dagger(s^m r^{k'}) \phi(s^m r^k) = \begin{pmatrix} e^{i2\pi(k-k')/N} & 0 \\ 0 & e^{-i2\pi(k-k')/N} \end{pmatrix} \quad (\text{B2})$$

so that $M_{i;j} = M_{(m',k');(m,k)} = 2\delta_{m,m'} \cos[2\pi(\frac{k'-k}{N})]$. Therefore,

$$T_{ij} \equiv T_{(m',k');(m,k)} = e^{2\delta_{m,m'}\beta \cos[2\pi(k'-k)/N]}. \quad (\text{B3})$$

Now the Fourier matrix is built out of the inequivalent irreps of the D_N group, and can be represented as

$$F = \begin{pmatrix} \rho_a \\ \rho_b \\ \rho_c \\ \rho_d \\ \phi_{00(1)} \\ \phi_{01(1)} \\ \phi_{10(1)} \\ \phi_{11(1)} \\ \vdots \\ \phi_{00(N/2-1)} \\ \phi_{01(N/2-1)} \\ \phi_{10(N/2-1)} \\ \phi_{11(N/2-1)} \end{pmatrix} \quad (\text{B4})$$

so that we have

$$\begin{aligned} F_{0i} &= \rho_a = [1]^{2N} \\ F_{1i} &= \rho_b = [1]^N [-1]^N \\ F_{2i} &= \rho_c = [1, -1]^N \\ F_{3i} &= \rho_d = [1, -1]^{N/2} [-1, 1]^{N/2} \end{aligned} \quad (\text{B5})$$

where $[a]^{m_1}[b]^{m_2}$ denotes an $m_1 + m_2$ -dimensional row vector with the first m_1 entries equaling a , and the next m_2 entries equaling b , and $[a, b]^m$ denotes a $2m$ -dimensional row vector with entries alternating between a and b . Let $F' = FT$ and $\rho'_a, \dots, \rho'_d, \phi'_{00(1)}, \dots, \phi'_{11(N/2-1)}$ denote its rows.

We then have

$$\begin{aligned} (FT)_{0;(m,k)} &= \sum_{m',k'} F_{0;(m',k')} T_{(m',k');(m,k)} \\ &= N + \sum_{k'=0}^{N-1} e^{2\beta \cos[2\pi(k'-k)/N]} \\ &= N + \sum_{k'=0}^{N-1} e^{2\beta \cos(2\pi k'/N)} \end{aligned} \quad (\text{B6})$$

where in the last step we have repeatedly used the identity $\cos(\frac{2\pi(N-k)}{N}) = \cos(\frac{2\pi k}{N})$. The last expression in Eq. (B6) is independent of (m, k) , so that

$$\rho'_a = \left(N + \sum_{k'=0}^{N-1} e^{2\beta \cos(2\pi k'/N)} \right) \rho_a. \quad (\text{B7})$$

Similarly,

$$\begin{aligned} (FT)_{1;(m,k)} &= F_{1;(m',k')} T_{(m',k');(m,k)} \\ &= \sum_{k'} e^{2\delta_{0,m}\beta \cos[2\pi(k'-k)/N]} \\ &\quad - \sum_{k'} e^{2\delta_{1,m}\beta \cos[2\pi(k'-k)/N]} \\ &= (1-2m) \left(\sum_{k'} e^{2\beta \cos(2\pi k'/N)} - N \right) \end{aligned} \quad (\text{B8})$$

and since $(\rho_b)_{(m,k)} = (1-2m)$, we have

$$\rho'_b = \left(\sum_{k'} e^{2\beta \cos(2\pi k'/N)} - N \right) \rho_b \quad (\text{B9})$$

Through very similar calculations, we find

$$\begin{aligned}
(FT)_{2;(m,k)} &= \sum_{m'} \left[\sum_{k' \text{ odd}} (-1) e^{2\delta_{m,m'} \beta \cos [2\pi(k'-k)/N]} \right. \\
&\quad \left. + \sum_{k' \text{ even}} e^{2\delta_{m,m'} \beta \cos [2\pi(k'-k)/N]} \right] \\
&= \sum_{k'} (-1)^{k'} e^{2\beta \cos (2\pi k'/N)} \quad (\text{B10})
\end{aligned}$$

and, for the last of the 1D irreps,

$$(FT)_{3;(m,k)} = (-1)^m \sum_{k'} (-1)^{k'} e^{2\beta \cos (2\pi k'/N)} \quad (\text{B11})$$

so that

$$\rho'_{c,d} = \left(\sum_{k'} (-1)^{k'} e^{2\beta \cos (2\pi k'/N)} \right) \rho_{c,d}. \quad (\text{B12})$$

Next, for the 2D irreps, we have

$$\begin{aligned}
\phi_{00(l)}(s^m r^k) &= (1-m) e^{i2\pi l k/N} \\
\phi_{01(l)}(s^m r^k) &= m e^{-i2\pi l k/N} \\
\phi_{10(l)}(s^m r^k) &= m e^{i2\pi l k/N} \\
\phi_{11(l)}(s^m r^k) &= (1-m) e^{-i2\pi l k/N}. \quad (\text{B13})
\end{aligned}$$

We will make use of the identities

$$\sum_{k'=0}^{N-1} \cos\left(\frac{2\pi l k'}{N}\right) = \sin(\pi l) \left(\cos(\pi l) \cot\left(\frac{\pi l}{N}\right) + \sin(\pi l) \right) \quad (\text{B14})$$

$$\begin{aligned}
\sum_{k'=0}^{N-1} \sin\left(\frac{2\pi l k'}{N}\right) &= \frac{1}{2} \left(\cos\left(\frac{\pi l}{N}\right) - \cos\left(\frac{\pi l(2N-1)}{N}\right) \right) \\
&\quad \times \operatorname{cosec}\left(\frac{\pi l}{N}\right) \quad (\text{B15})
\end{aligned}$$

both of which vanish for $l \in \mathbb{Z}$. Therefore,

$$\sum_{k'=0}^{N-1} e^{\pm i2\pi l k'/N} = 0 \quad (\text{B16})$$

for $1 \leq l < N/2$. We will also make use of the identity

$$\sum_{k'=0}^{N-1} e^{\pm i2\pi l k'/N} = \sum_{k'=0}^{N-1} e^{\pm i2\pi l (k'+k)/N} \quad (\text{B17})$$

for $k = 0, \dots, N-1$.

Now, analyzing the transformation of the 2D irreps as we did before for the 1D irreps, we have

$$\begin{aligned}
\phi'_{00(l)}(s^m r^k) &= \sum_{(m',k')} \phi_{00(l)}(s^{m'} r^{k'}) T_{(m',k');(m,k)} \\
&= \sum_{m',k'} (1-m') e^{i2\pi l k'/N} e^{2\beta \delta_{m,m'} \cos [2\pi(k'-k)/N]} \\
&= \begin{cases} \sum_{k'} e^{i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} & , m = 0 \\ \sum_{k'} e^{i2\pi l k'/N} & , m = 1 \end{cases} \\
&= (1-m) e^{i2\pi l k/N} \sum_{k'} e^{i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} \\
&= \left(\sum_{k'} e^{i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} \right) \phi_{00(l)} \quad (\text{B18})
\end{aligned}$$

where in the second to last equality, we have used Eqs. (B16) and (B17). Repeating essentially the same arguments, we also obtain

$$\begin{aligned}
\phi'_{01(l)}(s^m r^k) &= \left(\sum_{k'} e^{-i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} \right) \phi_{01(l)} \\
\phi'_{10(l)}(s^m r^k) &= \left(\sum_{k'} e^{i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} \right) \phi_{01(l)} \\
\phi'_{11(l)}(s^m r^k) &= \left(\sum_{k'} e^{-i2\pi l k'/N} e^{2\beta \cos (2\pi k'/N)} \right) \phi_{11(l)} \quad (\text{B19})
\end{aligned}$$

Altogether, we have shown that $\rho'_i \propto \rho_i$ for all irreducible representations ρ_i . Finally, using the Schur orthogonality relations which state that for two inequivalent irreps of some finite group G , $\phi: G \rightarrow U_n(\mathbb{C})$ and $\rho: G \rightarrow U_m(\mathbb{C})$, we have

$$\langle \phi_{kl}, \rho_{ij} \rangle = 0, \quad \langle \phi_{kl}, \phi_{ij} \rangle = \delta_{ik} \delta_{jl} \quad (\text{B20})$$

and so we find

$$\begin{aligned}
FTF^\dagger = \text{Diag} \left\{ \left(N + \sum_{k'=0}^{N-1} e^{2\beta \cos(2\pi k'/N)} \right), \left(\sum_{k'} e^{2\beta \cos(2\pi k'/N)} - N \right), \left[\sum_{k'} (-1)^{k'} e^{2\beta \cos(2\pi k'/N)} \right]^2, \right. \\
\left[\left(\sum_{k'} e^{i2\pi k'/N} e^{2\beta \cos(2\pi k'/N)} \right), \left(\sum_{k'} e^{-i2\pi k'/N} e^{2\beta \cos(2\pi k'/N)} \right), \right. \\
\left. \left. \left(\sum_{k'} e^{i2\pi k'/N} e^{2\beta \cos(2\pi k'/N)} \right), \left(\sum_{k'} e^{-i2\pi k'/N} e^{2\beta \cos(2\pi k'/N)} \right) \right]_{l=1}^{\frac{N}{2}-1} \right\} \quad (\text{B21})
\end{aligned}$$

where, $[f_1(l), \dots, f_k(l)]_{l=1}^m$ denotes mk entries with increasing values of l . The expression above is valid for even N . We do not consider the odd N case in our paper, though the analysis may be carried out similarly.

-
- [1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
- [2] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
- [3] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for quantum field theories, *Science* **336**, 1130 (2012).
- [4] S. P. Jordan, H. Krovi, K. S. Lee, and J. Preskill, BQP-completeness of scattering in scalar quantum field theory, *Quantum* **2**, 44 (2018).
- [5] N. Klco, A. Roggero, and M. J. Savage, Standard model physics and the digital quantum revolution: Thoughts about the interface, [arXiv:2107.04769](https://arxiv.org/abs/2107.04769).
- [6] 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).
- [7] L. García-Álvarez, J. Casanova, A. Mezzacapo, I. L. Egusquiza, L. Lamata, G. Romero, and E. Solano, Fermion-Fermion Scattering in Quantum Field Theory with Superconducting Circuits, *Phys. Rev. Lett.* **114**, 070502 (2015).
- [8] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for fermionic quantum field theories, [arXiv:1404.7115](https://arxiv.org/abs/1404.7115).
- [9] A. Hamed Moosavian and S. Jordan, Faster quantum algorithm to simulate fermionic quantum field theory, *Phys. Rev. A* **98**, 012332 (2018).
- [10] 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).
- [11] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, Quantum algorithm for simulating real time evolution of lattice Hamiltonians, *SIAM J. Comput.* **250** (2018).
- [12] W. Du, J. P. Vary, X. Zhao, and W. Zuo, Quantum simulation of nuclear inelastic scattering, *Phys. Rev. A* **104**, 012611 (2021).
- [13] 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).
- [14] E. Campbell, Random Compiler for Fast Hamiltonian Simulation, *Phys. Rev. Lett.* **123**, 070503 (2019).
- [15] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, *Quantum Inf. Comput.* **12** (2012).
- [16] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, *Phys. Rev. Lett.* **114**, 090502 (2015).
- [17] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [18] G. H. Low and I. L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [19] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, Variational fast forwarding for quantum simulation beyond the coherence time, *npj Quantum Inf.* **6**, 82 (2020).
- [20] J. Gibbs, K. Gili, Z. Holmes, B. Commeau, A. Arrasmith, L. Cincio, P. J. Coles, and A. Sornborger, Long-time simulations with high fidelity on quantum hardware, [arXiv:2102.04313](https://arxiv.org/abs/2102.04313).
- [21] Y.-X. Yao, N. Gomes, F. Zhang, T. Iadecola, C.-Z. Wang, K.-M. Ho, and P. P. Orth, Adaptive variational quantum dynamics simulations, *PRX Quantum* **2**, 030307 (2021).
- [22] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS Collaboration), General methods for digital quantum simulation of gauge theories, *Phys. Rev. D* **100**, 034518 (2019).
- [23] J. Kogut and L. Susskind, Hamiltonian formulation of Wilson's lattice gauge theories, *Phys. Rev. D* **11**, 395 (1975).
- [24] A. F. Shaw, P. Lougovski, J. R. Stryker, and N. Wiebe, Quantum algorithms for simulating the lattice Schwinger model, *Quantum* **4**, 306 (2020).
- [25] P. Jordan and E. P. Wigner, About the Pauli exclusion principle, *Z. Phys.* **47**, 631 (1928).
- [26] S. Bravyi and A. Y. Kitaev, Fermionic quantum computation, *Ann. Phys. (Amsterdam)* **298**, 210 (2002).
- [27] Y.-A. Chen and A. Kapustin, Bosonization in three spatial dimensions and a 2-form gauge theory, *Phys. Rev. B* **100**, 245127 (2019).
- [28] C. Muschik, M. Heyl, E. Martinez, T. Monz, P. Schindler, B. Vogell, M. Dalmonte, P. Hauke, R. Blatt, and P. Zoller,

- U(1) Wilson lattice gauge theories in digital quantum simulators, *New J. Phys.* **19**, 103020 (2017).
- [29] E. Zohar and J.I. Cirac, Eliminating fermionic matter fields in lattice gauge theories, *Phys. Rev. B* **98**, 075119 (2018).
- [30] 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).
- [31] E. Zohar, J.I. Cirac, and B. Reznik, Simulating Compact Quantum Electrodynamics with Ultracold Atoms: Probing Confinement and Nonperturbative Effects, *Phys. Rev. Lett.* **109**, 125302 (2012).
- [32] 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).
- [33] E. Zohar, J.I. Cirac, and B. Reznik, Quantum simulations of gauge theories with ultracold atoms: Local gauge invariance from angular momentum conservation, *Phys. Rev. A* **88**, 023617 (2013).
- [34] E. Zohar and M. Burrello, Formulation of lattice gauge theories for quantum simulations, *Phys. Rev. D* **91**, 054506 (2015).
- [35] E. Zohar, J.I. Cirac, and B. Reznik, Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices, *Rep. Prog. Phys.* **79**, 014401 (2016).
- [36] E. Zohar, A. Farace, B. Reznik, and J.I. Cirac, Digital lattice gauge theories, *Phys. Rev. A* **95**, 023604 (2017).
- [37] N. Klco, J.R. Stryker, and M.J. Savage, $SU(2)$ non-Abelian gauge field theory in one dimension on digital quantum computers, *Phys. Rev. D* **101**, 074512 (2020).
- [38] A. Ciavarella, N. Klco, and M.J. Savage, A trailhead for quantum simulation of $SU(3)$ Yang-Mills lattice gauge theory in the local multiplet basis, *Phys. Rev. D* **103**, 094501 (2021).
- [39] J. Liu and Y. Xin, Quantum simulation of quantum field theories as quantum chemistry, *J. High Energy Phys.* **12** (2020) 011.
- [40] D. C. Hackett, K. Howe, C. Hughes, W. Jay, E. T. Neil, and J. N. Simone, Digitizing gauge fields: Lattice Monte Carlo results for future quantum computers, *Phys. Rev. A* **99**, 062341 (2019).
- [41] A. Alexandru, P. F. Bedaque, S. Harmalkar, H. Lamm, S. Lawrence, and N. C. Warrington (NuQS Collaboration), Gluon field digitization for quantum computers, *Phys. Rev. D* **100**, 114501 (2019).
- [42] A. Yamamoto, Real-time simulation of $(2+1)$ -dimensional lattice gauge theory on qubits, *Prog. Theor. Exp. Phys.* **2021**, 013B06 (2021).
- [43] J. F. Haase, L. Dellantonio, A. Celi, D. Paulson, A. Kan, K. Jansen, and C. A. Muschik, A resource efficient approach for quantum and classical simulations of gauge theories in particle physics, *Quantum* **5**, 393 (2021).
- [44] T. Armon, S. Ashkenazi, G. García-Moreno, A. González-Tudela, and E. Zohar, Photon-Mediated Stroboscopic Quantum Simulation of a \mathbb{Z}_2 Lattice Gauge Theory, *Phys. Rev. Lett.* **127**, 250501 (2021).
- [45] A. Bazavov, S. Catterall, R. G. Jha, and J. Unmuth-Yockey, Tensor renormalization group study of the non-Abelian higgs model in two dimensions, *Phys. Rev. D* **99**, 114507 (2019).
- [46] A. Bazavov, Y. Meurice, S.-W. Tsai, J. Unmuth-Yockey, and J. Zhang, Gauge-invariant implementation of the Abelian Higgs model on optical lattices, *Phys. Rev. D* **92**, 076003 (2015).
- [47] J. Zhang, J. Unmuth-Yockey, J. Zeiher, A. Bazavov, S. W. Tsai, and Y. Meurice, Quantum Simulation of the Universal Features of the Polyakov Loop, *Phys. Rev. Lett.* **121**, 223201 (2018).
- [48] J. Unmuth-Yockey, J. Zhang, A. Bazavov, Y. Meurice, and S.-W. Tsai, Universal features of the Abelian Polyakov loop in $1+1$ dimensions, *Phys. Rev. D* **98**, 094511 (2018).
- [49] J. F. Unmuth-Yockey, Gauge-invariant rotor Hamiltonian from dual variables of 3D $U(1)$ gauge theory, *Phys. Rev. D* **99**, 074502 (2019).
- [50] 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).
- [51] M. Kreshchuk, S. Jia, W. M. Kirby, G. Goldstein, J. P. Vary, and P. J. Love, Simulating hadronic physics on NISQ devices using basis light-front quantization, *Phys. Rev. A* **103**, 062601 (2021).
- [52] I. Raychowdhury and J. R. Stryker, Solving Gauss's law on digital quantum computers with loop-string-hadron digitization, *Phys. Rev. Research* **2**, 033039 (2020).
- [53] I. Raychowdhury and J. R. Stryker, Loop, string, and hadron dynamics in $SU(2)$ Hamiltonian lattice gauge theories, *Phys. Rev. D* **101**, 114502 (2020).
- [54] 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).
- [55] U.-J. Wiese, Towards quantum simulating QCD, in *Proceedings of the 24th International Conference on Ultra-Relativistic Nucleus-Nucleus Collisions (Quark Matter 2014)* (Darmstadt, Germany, 2014); [*Nucl. Phys.* **A931**, 246 (2014)].
- [56] D. Luo, J. Shen, M. Highman, B. K. Clark, B. DeMarco, A. X. El-Khadra, and B. Gadway, A framework for simulating gauge theories with dipolar spin systems, *Phys. Rev. A* **102**, 032617 (2020).
- [57] R. C. Brower, D. Berenstein, and H. Kawai, Lattice gauge theory for a quantum computer, *Proc. Sci.*, LATTICE2019 (2019) 112 [arXiv:2002.10028].
- [58] S. V. Mathis, G. Mazzola, and I. Tavernelli, Toward scalable simulations of Lattice Gauge Theories on quantum computers, *Phys. Rev. D* **102**, 094501 (2020).
- [59] H. Singh, Qubit $O(N)$ nonlinear sigma models, arXiv:1911.12353.
- [60] H. Singh and S. Chandrasekharan, Qubit regularization of the $O(3)$ sigma model, *Phys. Rev. D* **100**, 054505 (2019).
- [61] A. J. Buser, T. Bhattacharya, L. Cincio, and R. Gupta, Quantum simulation of the qubit-regularized $O(3)$ -sigma model, *Phys. Rev. D* **102**, 114514 (2020).
- [62] T. Bhattacharya, A. J. Buser, S. Chandrasekharan, R. Gupta, and H. Singh, Qubit Regularization of Asymptotic Freedom, *Phys. Rev. Lett.* **126**, 172001 (2021).
- [63] J. a. Barata, N. Mueller, A. Tarasov, and R. Venugopalan, Single-particle digitization strategy for quantum

- computation of a ϕ^4 scalar field theory, *Phys. Rev. A* **103**, 042410 (2021).
- [64] M. Kreshchuk, S. Jia, W.M. Kirby, G. Goldstein, J.P. Vary, and P.J. Love, Light-front field theory on current quantum computers, *Entropy* **23**, 597 (2021).
- [65] Y. Ji, H. Lamm, and S. Zhu (NuQS), Gluon field digitization via group space decimation for quantum computers, *Phys. Rev. D* **102**, 114513 (2020).
- [66] E. Gustafson, Prospects for simulating a qudit based model of $(1+1)d$ scalar QED, *Phys. Rev. D* **103**, 114505 (2021).
- [67] E. Zohar, Quantum simulation of lattice gauge theories in more than one space dimension—requirements, challenges, methods, *Philos. Trans. A. Math. Phys. Eng. Sci.* **380**, 20210069 (2021).
- [68] P. Hasenfratz and F. Niedermayer, Asymptotic freedom with discrete spin variables?, in *Proceedings of the 2001 Europhysics Conference on High Energy Physics (EPS-HEP 2001)* (Budapest, Hungary, 2001); *Proc. Sci., HEP2001* (**2001**) 229 [arXiv:hep-lat/0112003].
- [69] S. Caracciolo, A. Montanari, and A. Pelissetto, Asymptotically free models and discrete non-Abelian groups, *Phys. Lett. B* **513**, 223 (2001).
- [70] P. Hasenfratz and F. Niedermayer, Asymptotically free theories based on discrete subgroups, in *Lattice field theory. Proceedings of the 18th International Symposium, Lattice 2000* (Bangalore, India, 2000); [*Nucl. Phys. Proc. Suppl.* **94**, 575 (2001)].
- [71] A. Patrascioiu and E. Seiler, Continuum limit of two-dimensional spin models with continuous symmetry and conformal quantum field theory, *Phys. Rev. E* **57**, 111 (1998).
- [72] R. Krmar, A. Gendiar, and T. Nishino, Phase diagram of a truncated tetrahedral model, *Phys. Rev. E* **94**, 022134 (2016).
- [73] S. Caracciolo, A. Montanari, and A. Pelissetto, Asymptotically free models and discrete non-Abelian groups, *Phys. Lett. B* **513**, 223 (2001).
- [74] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, Lattice renormalization of quantum simulations, *Phys. Rev. D* **104**, 094519 (2021).
- [75] M. Creutz, L. Jacobs, and C. Rebbi, Monte Carlo study of Abelian lattice gauge theories, *Phys. Rev. D* **20**, 1915 (1979).
- [76] M. Creutz and M. Okawa, Generalized actions in $Z(p)$ lattice gauge theory, *Nucl. Phys.* **B220**, 149 (1983).
- [77] G. Bhanot and C. Rebbi, Monte Carlo simulations of lattice models with finite subgroups of $SU(3)$ as gauge groups, *Phys. Rev. D* **24**, 3319 (1981).
- [78] D. Petcher and D. H. Weingarten, Monte Carlo calculations and a model of the phase structure for gauge theories on discrete subgroups of $SU(2)$, *Phys. Rev. D* **22**, 2465 (1980).
- [79] G. Bhanot, $SU(3)$ lattice gauge theory in four-dimensions with a modified wilson action, *Phys. Lett.* **108B**, 337 (1982).
- [80] D. H. Weingarten and D. N. Petcher, Monte Carlo integration for lattice gauge theories with fermions, *Phys. Lett.* **99B**, 333 (1981).
- [81] D. Weingarten, Monte Carlo evaluation of hadron masses in lattice gauge theories with fermions, *Phys. Lett.* **109B**, 57 (1982).
- [82] J. B. Kogut, $1/n$ expansions and the phase diagram of discrete lattice gauge theories with matter fields, *Phys. Rev. D* **21**, 2316 (1980).
- [83] J. Romers, Discrete gauge theories in two spatial dimensions, Ph.D. thesis, Master's thesis, Universiteit van Amsterdam, 2007.
- [84] E. H. Fradkin and S. H. Shenker, Phase diagrams of lattice gauge theories with Higgs fields, *Phys. Rev. D* **19**, 3682 (1979).
- [85] D. Harlow and H. Ooguri, Symmetries in quantum field theory and quantum gravity, *Commun. Math. Phys.* **383**, 1669 (2021).
- [86] D. Horn, M. Weinstein, and S. Yankielowicz, Hamiltonian approach to $Z(N)$ lattice gauge theories, *Phys. Rev. D* **19**, 3715 (1979).
- [87] H. Flyvbjerg, Group space decimation: A way to simulate the 1080 element subgroup of $SU(3)$?, *Nucl. Phys.* **B243**, 350 (1984).
- [88] H. Flyvbjerg, Internal space decimation for lattice gauge theories, *Nucl. Phys.* **B240**, 481 (1984).
- [89] C. Hagedorn, A. Meroni, and L. Vitale, Mixing patterns from the groups $\Sigma(n\varphi)$, *J. Phys. A* **47**, 055201 (2014).
- [90] P. Lisboa and C. Michael, Discrete subsets of $SU(3)$ for lattice gauge theory, *Phys. Lett.* **113B**, 303 (1982).
- [91] J. R. Stryker, Oracles for Gauss's law on digital quantum computers, *Phys. Rev. A* **99**, 042301 (2019).
- [92] J. C. Halimeh and P. Hauke, Reliability of Lattice Gauge Theories, *Phys. Rev. Lett.* **125**, 030503 (2020).
- [93] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS Collaboration), Suppressing coherent gauge drift in quantum simulations, arXiv:2005.12688.
- [94] R. C. Edgar, $Z(N)$ lattice gauge models with generalized actions, *Nucl. Phys.* **B200**, 345 (1982).
- [95] M. Fukugita, T. Kaneko, and M. Kobayashi, Phase structure and duality of $Z(N)$ lattice gauge theory with generalized actions in four space-time dimensions, *Nucl. Phys.* **B215**, 289 (1983).
- [96] D. Horn, M. Karliner, E. Katznelson, and S. Yankielowicz, Phase structure of $U(1)$ models with mixed actions, *Phys. Lett.* **113B**, 258 (1982).
- [97] C. Ayala and M. Baig, Strong coupling expansions in pure lattice gauge theory mixed actions, *Ann. Phys. (N.Y.)* **198**, 1 (1990).
- [98] A. D. Hill, M. J. Hodson, N. Didier, and M. J. Reagor, Realization of arbitrary doubly-controlled quantum phase gates, arXiv:2108.01652.
- [99] M. Creutz, *Quarks, Gluons and Lattices*, Cambridge Monographs on Mathematical Physics (Cambridge University Press, Cambridge, England, 1985).
- [100] T. Häner, M. Roetteler, and K. M. Svore, Factoring using $2n+2$ qubits with toffoli based modular multiplication, *Quantum Inf. Comput.* **17**, 0673 (2017).
- [101] C. Gidney, Constructing large increment gates (2015), <https://algassert.com/circuits/2015/06/12/Constructing-Large-Increment-Gates.html>.

- [102] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, Quantum algorithms and circuits for scientific computing, *Quantum Inf. Comput.* **16**, 197 (2016).
- [103] T. Häner, M. Roetteler, and K. M. Svore, Optimizing quantum circuits for arithmetic, [arXiv:1805.12445](https://arxiv.org/abs/1805.12445).
- [104] Y. Takahashi, S. Tani, and N. Kunihiro, Quantum addition circuits and unbounded fan-out, *Quantum Inf. Comput.* **10**, 0872 (2010).
- [105] S. Hadfield, On the representation of Boolean and real functions as Hamiltonians for quantum computing, *ACM Trans. Quantum Comput.* **2**, 1 (2021).
- [106] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [107] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, Efficient quantum circuits for diagonal unitaries without ancillas, *New J. Phys.* **16**, 033040 (2014).
- [108] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the poisson equation, *New J. Phys.* **15**, 013021 (2013).
- [109] S. Wang, Z. Wang, W. Li, L. Fan, G. Cui, Z. Wei, and Y. Gu, Quantum circuits design for evaluating transcendental functions based on a function-value binary expansion method, *Quantum Inf. Process.* **19**, 1 (2020).
- [110] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, England, 2010).
- [111] P. Hoyer, Efficient quantum transforms, [arXiv:quant-ph/9702028](https://arxiv.org/abs/quant-ph/9702028).
- [112] R. Beals, Quantum computation of Fourier transforms over symmetric groups, in *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, El Paso, Texas, 1997), pp. 48–53, [10.1145/258533.258548](https://doi.org/10.1145/258533.258548).
- [113] M. Püschel, M. Rötteler, and T. Beth, Fast quantum Fourier transforms for a class of non-Abelian groups, in *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes* (Springer, New York, 1999), pp. 148–159.
- [114] C. Moore, D. Rockmore, and A. Russell, Generic quantum Fourier transforms, *ACM Trans. Algorithms (TALG)* **2**, 707 (2006).
- [115] A. M. Childs and W. Van Dam, Quantum algorithms for algebraic problems, *Rev. Mod. Phys.* **82**, 1 (2010).
- [116] O. Regev, A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, [arXiv:quant-ph/0406151](https://arxiv.org/abs/quant-ph/0406151).
- [117] G. Kuperberg, Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem, [arXiv:1112.3333](https://arxiv.org/abs/1112.3333).
- [118] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Didier, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, Demonstration of a parametrically activated entangling gate protected from flux noise, *Phys. Rev. A* **101**, 012302 (2020).
- [119] D. M. Abrams, N. Didier, M. P. Johnson, Blake R. da Silva, and C. A. Ryan, Implementation of XY entangling gates with a single calibrated pulse, *National electronics review* **3**, 744 (2020).
- [120] M. Reagor *et al.*, Demonstration of universal parametric entangling gates on a multi-qubit lattice, *Sci. Adv.* **4**, eaa03603 (2018).
- [121] P. J. Karalekas, N. A. Tezak, E. C. Peterson, C. A. Ryan, M. P. da Silva, and R. S. Smith, A quantum-classical cloud platform optimized for variational hybrid algorithms, *Quantum Sci. Technol.* **5**, 024003 (2020).
- [122] R. S. Smith, M. J. Curtis, and W. J. Zeng, A practical quantum instruction set architecture, [arXiv:1608.03355](https://arxiv.org/abs/1608.03355).
- [123] R. S. Smith, E. C. Peterson, M. G. Skilbeck, and E. J. Davis, An open-source, industrial-strength optimizing compiler for quantum programs, [arXiv:2003.13961](https://arxiv.org/abs/2003.13961).
- [124] N. Schuch and J. Siewert, Natural two-qubit gate for quantum computation using the XY interaction, *Phys. Rev. A* **67**, 032301 (2003).
- [125] E. Peters, A. C. Y. Li, and G. N. Perdue, Perturbative readout error mitigation for near term quantum computers, [arXiv:2105.08161](https://arxiv.org/abs/2105.08161).
- [126] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, Scalable mitigation of measurement errors on quantum computers, *PRX Quantum* **2**, 040326 (2021).
- [127] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, Characterizing large-scale quantum computers via cycle benchmarking, *Nat. Commun.* **10**, 5347 (2019).
- [128] S. Harmalkar, H. Lamm, and S. Lawrence (NuQS Collaboration), Quantum simulation of field theories without state preparation, [arXiv:2001.11490](https://arxiv.org/abs/2001.11490).
- [129] E. J. Gustafson and H. Lamm, Toward quantum simulations of \mathbb{Z}_2 gauge theory without state preparation, *Phys. Rev. D* **103**, 054507 (2021).
- [130] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS Collaboration), Parton physics on a quantum computer, *Phys. Rev. Research* **2**, 013272 (2020).
- [131] T. D. Cohen, H. Lamm, S. Lawrence, and Y. Yamauchi, Quantum algorithms for transport coefficients in gauge theories, *Phys. Rev. D* **104**, 094514 (2021).