Quantum Simulation of Dihedral Gauge Theories

M. Sohaib Alam,[1,*] Stuart Hadfield,[2,3,†] Henry Lamm,[4,‡] and Andy C. Y. Li[4,§]
(SQMS collaboration)

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

(Dated: August 31, 2021)

We describe the simulation of dihedral gauge theories on digital quantum computers. The nonabelian 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 nonabelian 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 fidelity of all D_4 gates was found to exceed 80%.

I. INTRODUCTION

A promising area of potential quantum advantage is the simulation of the dynamics of nonperturbative quantum field theories [1–5]. In order to propagating for a time t, one requires the unitary operator of $\mathcal{U}(t) = e^{-iHt}$ 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 $\mathcal{U}(t)$, fundamentally these methods all require one to implement operations derived from the Hamiltonian H as quantum circuits [22]. Thus, a small set of basic gates 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 of the quantum algorithms was 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 straightforward [25–27]. Further proposals discuss how to map lattice fermions (e.g. Wilson and staggered) onto 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 digitization 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].

On promising digitization method is the approximation of continuous 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 of the studies focusing on the theories in $3+1$ dimensions. 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]. Along side this work, theoretical studies revealed that such discrete subgroup approximations correspond to effective field theories of continuous groups where a mass m_f is given to the gauge fields through the Higgs mechanism [82–86]. The result of this mass is that the discrete subgroup fails to well approximate the continuous group below a certain lattice spacing $a_f \sim m_f^{-1}$ (or equivalently beyond a certain coupling β_f).

In lattice calculations, one performs calculations at fixed lattice spacing $a = a(\beta)$ which shrinks as $\beta \to \infty$ for asymptotically free theories. To control extrapolation errors in taking $a \to 0$, one simulates in the *scaling regime* of $a \ll m_{IR}^{-1}$ where m_{IR} is the infrared mass scale of the physics of interest. We will consider the start of the scaling regime as occuring at a_s. Thus, the approximation error from using discrete subgroups should be small provided

* sohaib@rigetti.com
† stuart.hadfield@nasa.gov
‡ hlamm@fnal.gov
§ cli@fnal.gov

This document was prepared by SQMS Collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359.

that $a_s \gtrsim a_f$ or equivalently that $\beta_s \leq \beta_f$. In the case of $U(1)$ in $3+1d$ with $\beta_s = 1$, $\mathbb{Z}_{n>5}$ satisfies $\beta_f > \beta_s$. For nonabelian groups, only a finite set of crystal-like subgroups exist. $SU(2)$ has three: the binary tetrahedral $\mathbb{BT}$, the binary octahedral $\mathbb{BO}$, and the binary icosahedral $\mathbb{BI}$. While $\mathbb{BT}$ has $\beta_f = 2.24(8)$ in $3+1d$, $\mathbb{BO}$ and $\mathbb{BI}$ have $\beta_f = 3.26(8)$ and $\beta_f = 5.82(8)$ respectively [41], above $\beta_s = 2.2$. Hence, $\mathbb{BO}$ and $\mathbb{BI}$ appear useful for $SU(2)$.

For $SU(3)$ (the gauge theory underlying QCD) with $\beta_s \approx 6$ in $3+1d$, 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 alone. By extending $\mathbb{V}$ to include the midpoints between elements of $\mathbb{V}$, one can increase $\beta_f \approx 7$ [90]. However this require more qubits and sacrifices gauge symmetry. This loss of gauge symmetry is dangerous on noisy quantum computers [91–93]. An alternative approach to decrease a_f was to introduce 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 suggest 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.

Nonabelian 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 nonabelian theory, $\mathbb{BT}$ 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 $N \to \infty$ 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 $2+1d$ and $3+1d$, 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 $3+1d$ but also consider $2+1d$ theories. Using classical lattice simulations, we have determined that in both spacetimes, either D_9 (which we did not simulate) or D_{10} is the first group such that $a_f < a_s$ (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, $\mathbb{BT}$ would be a natural next step.

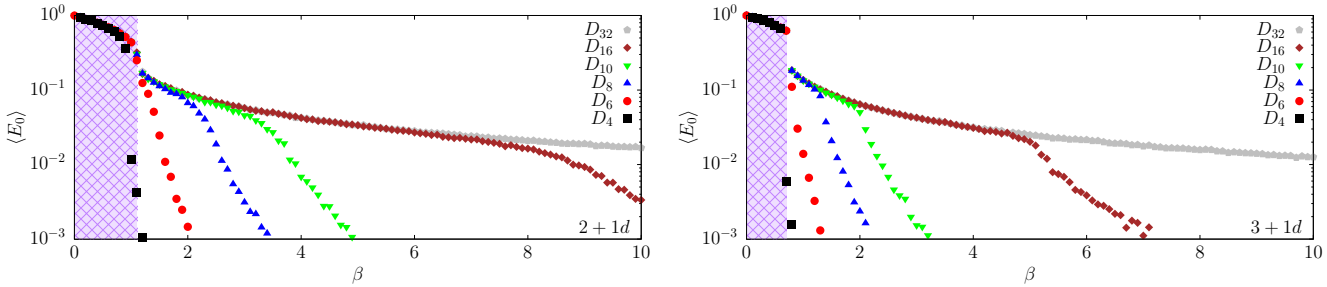


FIG. 1: Lattice energy density $\langle E_0 \rangle$ as a function of Wilson coupling β for different D_N groups in (left) 2+1 dimensions and (right) 3+1 dimensions. The shaded region corresponds to couplings outside the scaling regime for the $U(1) \times \mathbb{Z}_2$ theory.

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. Sec. 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 sections where quantum circuit constructions are given for these gates for the 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. Experimental benchmarking results for our D_4 gates on the Rigetti Aspen-9 quantum processor are found in Sec. VIII. We conclude and discuss future work in Sec. IX.

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

II. LATTICE FIELD THEORY

It is useful to review the connection between the Kogut-Susskind Hamiltonian [23] and the Euclidean Wilson action. We summarize the derivation of [98] that begins with the anisotropic Wilson action in Euclidean time $\tau = it$ defined on a spacetime lattice:

$$S_E = -\beta_t \sum_t \mathrm{Re\,Tr}\, U_t - \beta_s \sum_s \mathrm{Re\,Tr}\, U_s \tag{1}$$

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

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

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 ($a_0 \to 0$), 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 obtained through the transfer matrix, $T(a, a_0)$ which takes a state at time τ, $|\tau\rangle$, to $|\tau + 1\rangle$. T is related to S_E through the partition function Z:

$$Z = \int DU e^{-S_E} = \mathrm{Tr} T(a_0)^N \tag{3}$$

where N counts time slices. It follows that the matrix elements of $T(a_0)$ are [98]

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

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

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

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 evolves a link via

$$R_{ij}(g)|\tau\rangle = |\tau'\rangle, \quad \text{where } U_{ij} \to g U_{ij}. \tag{6}$$

This operator has the group property of $R_{ij}(g)R_{ij}(h) = R_{ij}(gh)$ and can be used to define a conjugate momentum to $\hat{U}_{ij}$ by performing a rotation on $U_{ij}(\mathbf{x}, \tau + 1)$. With this, we write

$$T_K = \prod_{\{ij\}} T_{K,ij} = \prod_{\{ij\}} \left[\int Dg R_{ij}(g) e^{\beta_t \mathrm{\,Re\,Tr} g} \right], \tag{7}$$

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} for that representation. Defining $\prod_\alpha (D\omega^\alpha) J(\omega)$ as the invariant group measure with a Jacobian J, one can rewrite $T_K(a_0)$ as

$$T_{K,ij} = \int \prod_\alpha (D\omega^\alpha) J(\omega) e^{i\omega\cdot l_{ij}} e^{\beta_t \mathrm{Tr}\cos(\omega\cdot\lambda)}. \tag{8}$$

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 over infinite character functions and 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, the final transfer matrix $T(a_0)$ is

$$T(a_0) = e^{\frac{\beta_s}{2}\sum_s \mathrm{Re\,Tr}\hat{U}_s} e^{-\beta_t^{-1}\sum_{\{ij\}} l_{ij}^2} e^{\frac{\beta_s}{2}\sum_s \mathrm{Re\,Tr}\hat{U}_s}. \tag{9}$$

From this, we can use the definition $T(a_0) \equiv e^{-a_0 H(a,a_0)}$ to obtain a lattice Hamiltonian. However, since l_{ij} and $\hat{U}_{ij}$ are non-commuting operators, rearranging this expression into a single exponential requires application of the Baker-Campbell-Hausdorff (BCH) formula. Using this, $H(a,a_0)$ is found to be

$$\begin{aligned} H(a,a_0) =& \frac{1}{c(a,a_0)a}\Bigg(g_H^2(a,a_0)\sum_{\{ij\}} l_{ij}^2 - g_H^{-2}(a,a_0)\sum_s \mathrm{Re\,Tr}\hat{U}_s \\ &- \frac{1}{24}\frac{1}{c^2(a,a_0)\xi^2}\sum_{\{ij\},s}\Big(g_H^2(a,a_0)[2l_{ij}^2,[l_{ij}^2,\mathrm{Re\,Tr}\hat{U}_s]] - g_H^{-2}(a,a_0)[\mathrm{Re\,Tr}\hat{U}_s,[l_{ij}^2,\mathrm{Re\,Tr}\hat{U}_s]]\Big) + \dots\Bigg). \end{aligned} \tag{10}$$

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

$$\mathcal{T}(\tau) \equiv \lim_{a_0\to 0, N\to\infty} T(a_0)^N, \tag{11}$$

the BCH terms will go to zero in the Hamiltonian and one is left with the Kogut-Susskind Hamiltonian [23], $H_{KS} \equiv -\frac{1}{\tau}\log(\mathcal{T}(\tau))$

$$H_{KS} = \frac{1}{c(a)a}\left(g_H^2(a)\sum_{\{ij\}} l_{ij}^2 - \frac{1}{g_H^2(a)}\sum_s \mathrm{Re\,Tr}\,U_s\right). \tag{12}$$

This is the common starting point for the real-time 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. (12).

III. OVERVIEW OF BASIC GATES

Given a gauge group G, we define 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

$$\mathfrak{U}_{-1}|g\rangle = \left|g^{-1}\right\rangle. \tag{13}$$

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

$$\mathfrak{U}_{\times}|g\rangle|h\rangle = |g\rangle|gh\rangle. \tag{14}$$

Here we have defined $\mathfrak{U}_{\times}$ as implementing in-place left multiplication, in the sense that the content of the second register was multiplied on the left. Left multiplication suffices as right multiplication can be implemented using two applications each of $\mathfrak{U}_{-1}$ and $\mathfrak{U}_{\times}$ [22].

The trace of a plaquette appears in our gauge Hamiltonian, and so to perform this operation we combine the matrix multiplication gate with a single-register trace gate:

$$\mathfrak{U}_{\mathrm{Tr}}(\theta)|g\rangle = e^{i\theta\,\mathrm{Re\,Tr}\,g}|g\rangle. \tag{15}$$

In our construction, the final gate required on the G-register is the Fourier transform gate $\mathfrak{U}_F$. This gate acts on a G-register to rotate into the Fourier basis. In general it is defined by

$$\mathfrak{U}_F\sum_{g\in G} f(g)|g\rangle = \sum_{\rho\in\hat{G}} \hat{f}(\rho)_{ij}|\rho,i,j\rangle. \tag{16}$$

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 Appendix A. Following [22], the $2N$ group elements $s^m r^k$, $m \in \{0, 1\}$, $k \in \{0, \ldots, N-1\}$, 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 $U_{-1}|g\rangle = |g^{-1}\rangle$ 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 $U_\times$ and its reversed (adjoint) circuit $U_\times^\dagger$, then we can implement U_{-1} using an ancillary G-register initialized to the group identity element $|e\rangle$. We discuss construction of the multiplication gate in Sect. V. We can then implement U_{-1} using the sequence of operations

$$|g\rangle|e\rangle \xrightarrow{U_\times^\dagger} |g\rangle|g^{-1}\rangle \xrightarrow{SWAP} |g^{-1}\rangle|g\rangle \xrightarrow{U_\times} |g^{-1}\rangle|e\rangle, \tag{17}$$

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 (relabelling) the top and bottom registers in the circuit for $U_\times$. Hence the cost of this implementation of the inversion gate is at most twice that of the multiplication gate. Note that the property that the ancilla register is initialized and returned to a fixed state can be used to further simplify the circuits for $U_{mult}^\dagger$ so that fewer gates are required than for the general case. In any case, the use of ancilliary 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 $s^m r^k \in D_N$ is given by

$$\left(s^m r^k\right)^{-1} = s^m r^{Nm+(-1)^m k}. \tag{18}$$

As a result, given the qubit encoding $s^m r^k \to |m\rangle|k\rangle$ described above, the effect of U_{-1} is to change the state of the register $|0\rangle|k\rangle \to |0\rangle|N-k\rangle$, 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 $N = 2^n$. 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.

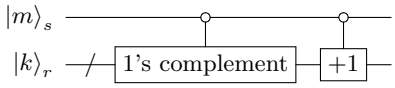


FIG. 2: Schematic quantum circuit implementation of the inversion gate U_{-1}. The controlled 1's complement operation can be implemented with n CNOT gates. The controlled increment gate can be implemented with $O(n)$ Tofolli gates using a constant number of additional ancilla qubits [99, 100].

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 [99, 100] yields quantum circuits for the controlled-increment operation using $O(n)$ Tofolli gates and as few as 1 additional ancilla qubits. An example circuit is shown in Fig. 3.

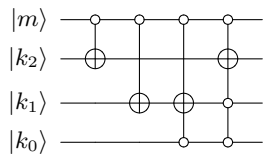


FIG. 3: Example: A simplified implementation of the inversion gate U_{-1} for D_8. The first two gates correspond to the 1's complement operation, and the last two to incrementation; an inner pair of CNOTs has canceled out.

V. MULTIPLICATION GATE

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

$$s^{m_1}r^{k_1} \cdot s^{m_2}r^{k_2} = s^{m_1+m_2}r^{Nm_2+(-1)^{m_2}k_1+k_2}, \tag{19}$$

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.

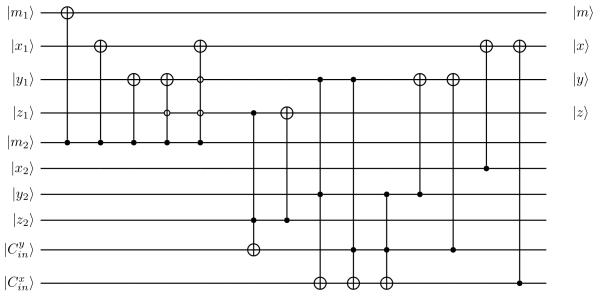


FIG. 4: A multiplication gate circuit for D_8 theory

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. (19).

A variety of proposed quantum algorithms for addition and multiplication exist in the literature [101, 102] with different resource tradeoffs, see, e.g., [103, 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

$$\begin{aligned} S &= A \oplus B \oplus C_{in} \\ C_{out} &= AB \oplus AC_{in} \oplus BC_{in}. \end{aligned} \tag{20}$$

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 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 $U_{\mathrm{Tr}}(\theta)\,|g\rangle = e^{i\theta\,\mathrm{Re}\,\mathrm{Tr}\,g}\,|g\rangle$ for D_N. Observe that, unlike the other basic gates we consider, this family of gates is parameterized 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 advantageous for cases where N is small or moderate in size.

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

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

$$\rho(g) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^m \begin{pmatrix} \omega & 0 \\ 0 & \omega \end{pmatrix}^k, \qquad \text{where} \quad \omega = e^{2\pi i/N}. \tag{21}$$

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

Therefore we have

$$H_{\mathrm{Tr}} = |0\rangle\langle 0| \otimes \sum_{\ell=0}^{N-1} 2\cos(2\pi\ell/N)\,|\ell\rangle\langle\ell|\,. \tag{22}$$

Direct implementation: We first consider implementation of U_{Tr} 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 $Z_\alpha = Z_{\alpha_1} \ldots Z_{\alpha_j}$ denotes a tensor product of Pauli Z operators indexed by subsets of qubits $\alpha \subset [n]$, with coefficients given by $a_\alpha = \mathrm{Tr}[Z_\alpha H] \in \mathbb{R}$ [104]. 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. (22), 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 $U_{\mathrm{Tr}}(\theta)$. For the latter case, i.e., writing $H_{\mathrm{Tr}} = |0\rangle\langle 0| \otimes \sum_\alpha a_\alpha Z_\alpha$, we have

$$U_{\mathrm{Tr}}(\theta) = \prod_{j=0}^{N} e^{i\theta a_\alpha |0\rangle\langle 0| \otimes Z_\alpha} = \prod_{\ell=0}^{N} \Lambda_{m=0}(e^{i\theta a_\alpha Z_\alpha}), \tag{23}$$

where $\Lambda_{m=0}(e^{i\theta a_\alpha Z_\alpha})$ denotes the controlled unitary that implements $e^{i\theta a_\alpha Z_\alpha}$ conditioned on the first qubit being zero, and we used the fact that diagonal terms mutually commute. Each such controlled rotation can be implemented with $O(n)$ basic gates consisting of CNOTs and single-qubit gates [105]; however the number of such 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 U_{Tr}. If we tolerate approximation error in U_T the number of terms can be further reduced [106].

For example, consider D_4, for which Eq. (21) gives $\rho(g) = X^a(iZ)^\ell$, which has trace 2 for $g = e$ and -2 for $g = r^2$, else 0. Hence we have the Hamiltonian $H_{\mathrm{Tr}} = 2\,|000\rangle\langle 000| - 2\,|010\rangle\langle 010|$ which we may write as $H_{\mathrm{Tr}} = \frac{1}{2}(Z_{k_1} + Z_m Z_{k_1} + Z_{k_1} Z_{k_0} + Z_m Z_{k_1} Z_{k_0})$, or with control as $H_{\mathrm{Tr}} = |0\rangle\langle 0| \otimes (Z_{k_1} + Z_{k_1} Z_{k_0}) = 2\,|0\rangle\langle 0| \otimes Z_{k_1} \otimes |0\rangle\langle 0|$. 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, as 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's first consider restricting the required trigonometric quantities to the first first quadrant $0 \le 2\pi\ell/N < \pi/2$ 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 $|g\rangle = |mk_{n-1} \ldots k_1 k_0\rangle$ we have $\mathrm{Tr}(g) = 2(1-m)\cos(2\phi k/N)$, 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 $Tr(0k_{n-1}0k_{n-3}\dots k_1k_0) = 2(-1)^{k_{n-1}}\cos(2\pi k'/N)$ and $Tr(0k_{n-1}1k_{n-3}\dots k_1k_0) = -2(-1)^{k_{n-1}}\sin(2\pi k/N)$, where k' is the integer given by the bits $k_{n-3}\dots k_1k_0$. (Note that a similar treatment of the first 3 bits may be employed in the direct case above, as indicated in 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 [101]. Then U_{Tr} can be implemented as follows:

1. (Compute classical functions.) Append a sufficiently large ancilla register $|00\dots0\rangle$ and reversibly compute (in superposition) the transformation for each basis state $|g\rangle = |mk_{n-1}k_{n-2}k'\rangle$

$$|mk_{n-1}k_{n-2}k'\rangle\,|0\dots0\rangle \to |mk_{n-1}k_{n-2}k'\rangle\,|\widetilde{\sin(2\pi k'/N)}\rangle\,|\widetilde{\cos(2\pi k'/N)}\rangle\,|scratchpad\rangle\,,$$

   where $\widetilde{x}$ denotes a b-bit binary approximation of a quantity $0 \le x < 1$. The remaining $|scratchpad\rangle$ register 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 $0 \le x < 1$ we can implement $|x\rangle \to e^{i\theta x}|x\rangle$ (up to an irrelavant global phase) using a controlled $R_Z(\theta 2^{-j})$ gate applied to each jth 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 $|x\rangle \to e^{i2\theta x}|x\rangle$ applying instead controlled $R_Z(\theta 2^{1-j})$ gates. Hence we apply two high-level unitaries that kickback a phase of $\theta Tr(g)$ to each basis state:

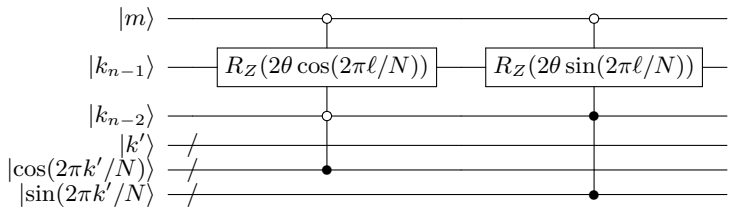


FIG. 5: Schematic for phase kickback of $\mathrm{Tr}(g)$ using registers containing the required cos and sin values. Here the two schematic gates represent one multicontrolled-Z rotation for each bit in the sin/cos registers, respectively, i.e., the gate $R_Z(\theta 2^{1-j})$ is applied for each jth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

3. (Uncomputation.) As the operation of the second step is diagonal, we can restore the ancilla qubits to $|00\dots0\rangle$ then discard for reuse by applying the reverse of the circuit in Step 1. Hence each input basis state is taken to

$$|g\rangle = |mk\rangle \to e^{i\theta 2(1-m)\cos(2\pi\ell_b/N)}\,|mk\rangle = e^{i\theta Tr(g)}\,|g\rangle$$

   as desired.

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 linearly on b. Here we've assumed the ancilla qubits are restored to $|00\dots0\rangle$ 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 [101].

Computing the trigonometric functions: Methods for computing the cos and sin functions using quantum arithmetic circuits are discussed in [101, 102, 107, 108]. 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 [107, Sec. 5 and App. 2] requires only addition and multiplication operations, and simultaneously computes both sin and cos using repeated squaring via the approximation

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

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. [107, Prop. 1 and 2]).

Alternatively, the approach of [102, App. 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 [108] employs nontrivial quantum submodules for approximately computing square roots. In these approaches care must be taken to ensure the desired overall gate 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 the D_4 trace gate and its compilation to hardware gates in Fig. 10 below.

VII. FOURIER GATE

The standard n-qubit quantum Fourier transform [109], a critical component, for instance, of Shor's prime factoring algorithm, corresponds to the abelian group $\mathbb{Z}_{2^n}$. More general quantum circuits implementing Fourier transforms over a variety of nonabelian groups have been considered in [110–113], though there remains important groups for which efficient QFT circuits are not known [114].[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 [110] for efficient circuits for QFTs over metacyclic groups also includes the dihedral case.

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

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

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

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

where $\hat{G}$, the *dual* of G, is the set of all irreducible representations of G and $|G| = |\hat{G}|$. Note that 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

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

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 if we have a series of subgroups $H_1, \ldots, H_n$ that form a chain $G \supset H_1 \supset \cdots \supset H_n = id$, using "adapted bases" such that $\rho|_{H_i}$ can be written as a direct sum of irreducible representations of H_i.

---

[2] We note that in general an efficient quantum circuit for the QFT of a group G does not necessarily 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 [114]. Subexponential time quantum algorithms for the HSP on dihedral groups are given in [115, 116] using the standard (abelian) QFT rather than the dihedral one considered here.

We may similarly construct quantum Fourier transforms (QFTs) if we can perform the following series of operations

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

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 $s^m r^k \to |g\rangle = |m\rangle|k\rangle$, $|k\rangle$ encode the basis elements of $\mathbb{Z}_N$, while $|\tilde{k}\rangle$ denote the Fourier basis of $\mathbb{Z}_N$. Then, we have $F_H : |m\rangle|k\rangle \to |m\rangle|\tilde{k}\rangle$. Likewise, denoting the Fourier basis of D_N by $|\tilde{g}\rangle$, the final transformation is $U : |m\rangle|\tilde{k}\rangle \to |\tilde{g}\rangle$. Determining U is often the more non-trivial part of any such QFT algorithm.

For even N, the group D_N has the following four 1-dimensional irreducible representations:

- $\rho_A : r^k \to 1$, $sr^k \to 1$, $\forall k \in \{0 \dots N-1\}$
- $\rho_B : r^k \to 1$, $sr^k \to -1$, $\forall k \in \{0 \dots N-1\}$
- $\rho_C : s^m r^k \to 1$, if $k \in \{0 \dots N-1\}$ is even, and $s^m r^k \to -1$ if $k \in \{0 \dots N-1\}$ is odd, $\forall m \in \{0,1\}$
- $\rho_D : r^k \to 1$ if k is even, and $r^k \to -1$ if k is odd; $sr^k \to -1$ if k is even, and $sr^k \to 1$ if k is odd,

and $\frac{N-2}{2}$ 2-dimensional irreducible representations of the form

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

where $1 \le l < \frac{N}{2}$. D_N has a cyclic subgroup $Z_N = \{r^0, \dots, r^{N-1}\}$ for which the QFT is well known [109], and for which the elements $\{e, s\}$ provide a left transversal in D_N. Our encoding of D_N elements into qubits, $s^m r^k \to |m\rangle|k\rangle$, 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 non-trivial step is provided by [110]

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

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 the D_N ($N = 2^n$ for some n) Fourier transform 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 2 values each, and we can therefore compile this operation using a single CCPHASE gate and an ancillary qubit, as shown in Fig. 7.

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

- $\rho_A \to |00\rangle|0\rangle^{\otimes n-1}$
- $\rho_B \to |10\rangle|0\rangle^{\otimes n-1}$
- $\rho_C \to |01\rangle|0\rangle^{\otimes n-1}$
- $\rho_D \to |11\rangle|0\rangle^{\otimes n-1}$

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

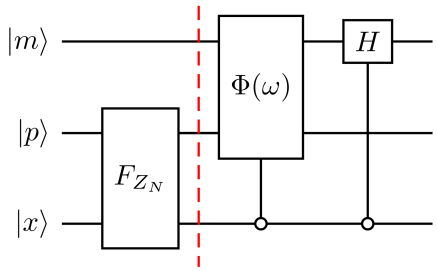


FIG. 6: Circuit computing the Fourier transform of the D_N ($N = 2^n$ for some n) group. The first part of the circuit computes the Fourier transform $F_{\mathbb{Z}_N}$ over the cyclic subgroup $H = \mathbb{Z}_N = \{r^0, \ldots, r^{N-1}\}$, while the latter part performs a change of basis $|g_i\rangle|\tilde{h}\rangle \to |\tilde{g}\rangle$, implementing the unitary transform given in Eq. (29).

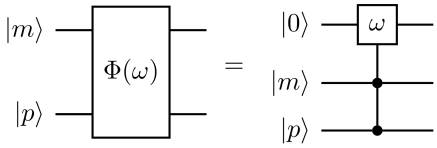


FIG. 7: Circuit for implementing $\Phi(\omega)$. This is given by a simple application of a $CCPHASE$ gate targeted on an ancillary qubit, obtaining a phase kick-back on the qubits $|k\rangle$ and $|i\rangle$, on which it is controlled.

- $\rho_{00}^{(l)} \to |00\rangle|l\rangle$
- $\rho_{01}^{(l)} \to |11\rangle|l\rangle$
- $\rho_{10}^{(l)} \to |10\rangle|l\rangle$
- $\rho_{11}^{(l)} \to |01\rangle|l\rangle$

If so desired, one could rearrange the representations to appear in a different order, e.g. the first four computational basis states $|\ldots 00\rangle$, $|\ldots 01\rangle$, $|\ldots 10\rangle$, $|\ldots 11\rangle$ encoding the four 1-dimensional irreducible representations, the next four encoding the matrix entries of the $l = 1$ 2-dimensional irreducible representation 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 our purpose, 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_4 theory (with similar reasoning applied for arbitrary D_{2^n} theories), 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 [117–119] (see Fig. 9). The Rigetti stack [120] allows us to use the Quil language [121] to program the Aspen-9 device, and its associated optimizing compiler Quil-C [122] to compile primitive gates such as Hadamard and CNOT into its native gateset $\{RZ(\theta), RX(k\pi/2), CPHASE, CZ, XY\}$. 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 or roughly equal to 85% fidelity or accuracy.

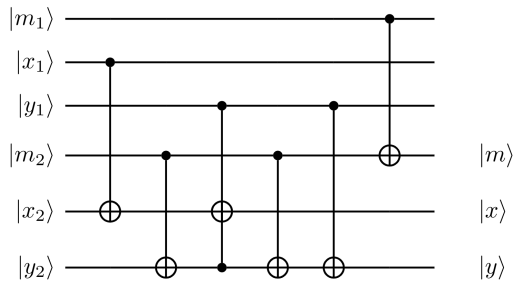


FIG. 8: Multiplication gate circuit for D_4 theory

A. D_4 Multiplication Gate

Concretely, for D_4 theory, we can use the encoding $sr^j \to |s\rangle|j\rangle = |a\rangle|bc\rangle$ as in [22] to specify an element of D_4 as $s^a r^{2b+c}$. We compute the product $|abc\rangle = |(a_1b_1c_1)\cdot(a_2b_2c_2)\rangle$, using the multiplication gate $U_\times|a_1b_1c_1\rangle|a_2b_2c_2\rangle = |a_1b_1c_1\rangle|abc\rangle$. Whether we perform subtraction or addition, the right-most bit will simply be given by $c = c_1 \oplus c_2$. Similarly for the left-most bit (a).

For the second-right-most 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 right-most bit. Depending on whether we perform addition or subtraction, the appropriate carry is either c_1c_2 or $c_1\bar{c}_2$ respectively. Thus, in all, we have the two following rules.

For the case $m_2 = 0$ (addition), we obtain the product

$$\begin{aligned} a &= a_1 \oplus a_2 \\ b &= b_1 \oplus b_2 \oplus c_1c_2 \\ c &= c_1 \oplus c_2 \end{aligned} \tag{30}$$

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

$$\begin{aligned} a &= a_1 \oplus a_2 \\ b &= b_1 \oplus b_2 \oplus c_1\bar{c}_2 \\ c &= c_1 \oplus c_2 \end{aligned} \tag{31}$$

In circuit form, this is provided in Fig. (8). We implement this on the Rigetti Aspen-9 QPU, whose lattice topology is shown in Fig. 9. 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 $(a_1, b_1, c_1, a_2, b_2, c_2) = (22, 30, 35, 21, 37, 36)$. This identification ensures only nearest-neighbor interactions in the implementation of the gate. In addition to 2-qubit gates such as CPHASE [119] and XY [118], the Rigetti hardware also allows the use of 3-qubit CCPHASE gates [123]. 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. 8, 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 ~ 0.19 (with standard deviation $\Delta \sim 0.06$). 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 ~ 0.89 (with standard deviation $\Delta \sim 0.18$). If we instead take the majority vote of 200 successive shots, we boost the average fraction of correct output bitstrings even more to ~ 0.91 (with standard deviation $\Delta \sim 0.15$).

B. D_4 trace gate and Fourier gate

We carry out Quantum Process Tomography (QPT) [109] to benchmark the fidelities of the Fourier gate and the trace gate $U_{\mathrm{Tr}}(\theta = \pi/2)$ for D_4 theory on the Rigetti Aspen-9 QPU as shown in fig. 9. To minimize the number of

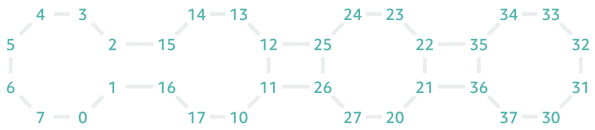
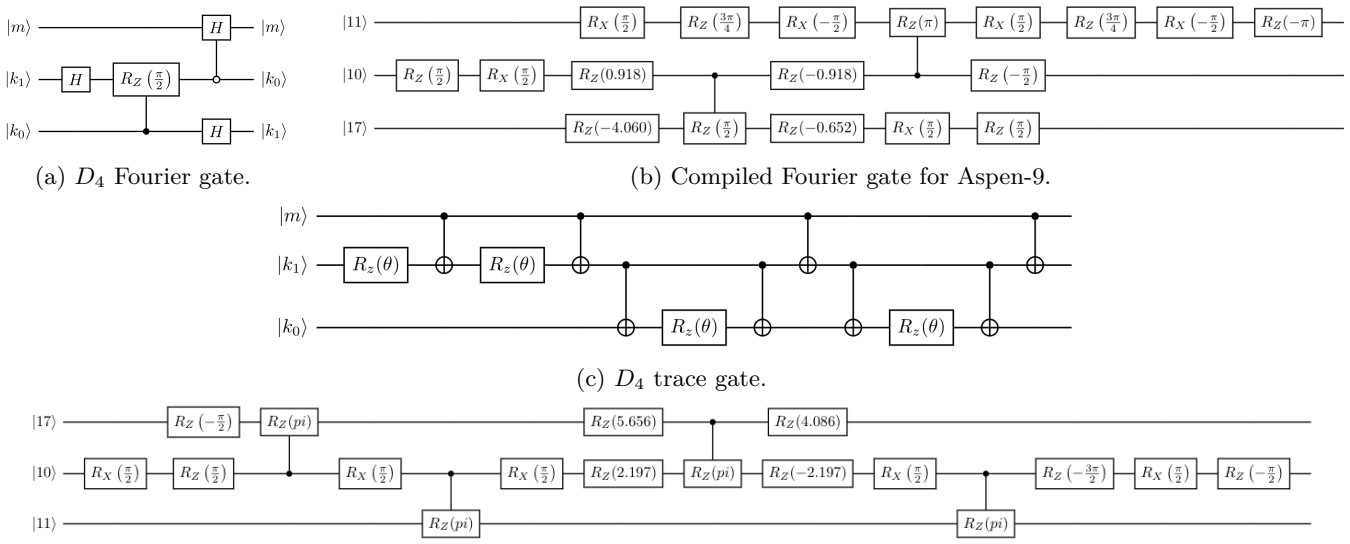


FIG. 9: Rigetti Aspen-9 lattice



(a) D_4 Fourier gate.

(b) Compiled Fourier gate for Aspen-9.

(c) D_4 trace gate.

(d) Compiled trace gate with $\theta = \pi/2$ for Aspen-9.

FIG. 10: Fourier gate and Trace gate 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 the compiled versions consisting of only native gates of Aspen-9 QPU. For the trace gate, $\theta = \pi/2$ is chosen for compilation.

SWAP gates required by the Aspen-9 connectivity, we swap the qubit ordering for a linear connectivity and implement the circuits using qubits (17, 10, 11) on 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 {RX, RZ, CPHASE(ϕ)} by the optimizing compiler Quilc [125], and need 2 and 4 two-qubit CPHASE gate, respectively as shown in fig. 10(b) and (d).

QPT measures the process fidelities of the Fourier gate and $U_{\rm Tr}(\theta = \pi/2)$ to be 0.920 and 0.857. The process infidelity is mainly originated from 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 are shown in fig. 11 with the ideal matrices inserted. 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 $|000\rangle$, $|001\rangle$, ..., $|111\rangle$ and measuring the output. Any distribution is then post-processed by inverting the confusion matrix to mitigate the readout error. More details and a scalable approach for measurement involving more qubits can be found in ref. [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 non-trivial operations the D_4 inversion gate therefore has are $|001\rangle \to |011\rangle$ and $|011\rangle \to |001\rangle$. This operation can be implemented using a single CCPHASE(π) gate [123], with a 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 sub-lattice $(10, 11, 12)$ (see Fig. 9) using cycle benchmarking [127].

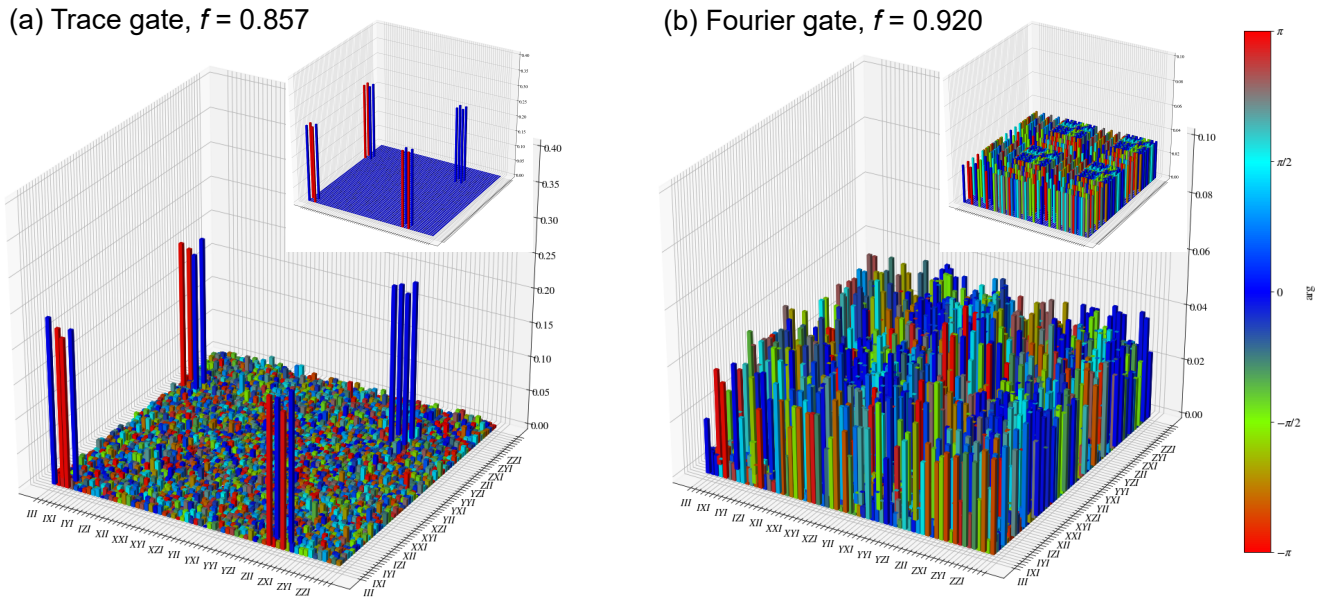


FIG. 11: The χ matrices associated with $U_{\mathrm{Tr}}(\theta = \pi/2)$ and the Fourier gate. The process fidelity is determined by $f = \mathrm{Tr}(\chi_{\mathrm{target}}^{\dagger}\chi)$, where the target χ_{target} computed by noiseless simulator is inserted to the figure. The trace gate has a lower fidelity $f = 0.857$ compared to that of the Fourier gate being $f = 0.920$ since the trace gate consists of two more two-qubit CZ gates.

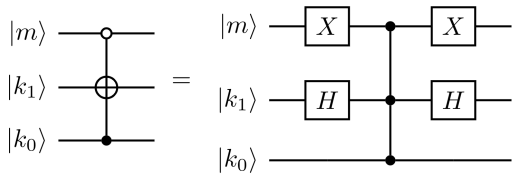


FIG. 12: Inversion gate for D_4 theory.

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 is within reach.

Looking to the future, several directions of further study 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 more 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 number DE-AC02-07CH11359. S.H. was supported by the NASA Academic Mission Services, Contract No. NNA16BD14C. Fermilab is operated by Fermi Research Alliance, LLC under contract number DE-AC02-07CH11359 with the United States Department of Energy.

[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), arXiv:1111.3633 [quant-ph].
[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), arXiv:1703.00454 [quant-ph].
[5] N. Klco, A. Roggero, and M. J. Savage, Standard model physics and the digital quantum revolution: Thoughts about the interface, arXiv preprint arXiv:2107.04769 (2021).
[6] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum Computation of Scattering in Scalar Quantum Field Theories, Quant. Inf. Comput. **14**, 1014 (2014), arXiv:1112.4833 [hep-th].
[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), arXiv:1404.2868 [quant-ph].
[8] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum Algorithms for Fermionic Quantum Field Theories (2014), arXiv:1404.7115 [hep-th].
[9] A. Hamed Moosavian and S. Jordan, Faster Quantum Algorithm to simulate Fermionic Quantum Field Theory, Phys. Rev. **A98**, 012332 (2018), arXiv:1711.04006 [quant-ph].
[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), arXiv:1804.02082 [quant-ph].
[11] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, Quantum algorithm for simulating real time evolution of lattice Hamiltonians (2018).
[12] W. Du, J. P. Vary, X. Zhao, and W. Zuo, Quantum Simulation of Nuclear Inelastic Scattering (2020), arXiv:2006.01369 [nucl-th].
[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 Information & Computation **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 Information **6**, 1 (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 (2021), arXiv:2102.04313 [quant-ph].
[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, arXiv preprint arXiv:2011.00622 (2020).
[22] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS), General Methods for Digital Quantum Simulation of Gauge Theories, Phys. Rev. **D100**, 034518 (2019), arXiv:1903.08807 [hep-lat].
[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), arXiv:2002.11146 [quant-ph].
[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, Annals of Physics **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), arXiv:1807.07081 [cond-mat.str-el].
[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), arXiv:1612.08653 [quant-ph].
[29] E. Zohar and J. I. Cirac, Eliminating fermionic matter fields in lattice gauge theories, Phys. Rev. B **98**, 075119 (2018), arXiv:1805.05347 [quant-ph].
[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), arXiv:1905.00652 [quant-ph].
[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), arXiv:1204.6574 [quant-ph].
[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), arXiv:1211.2241 [quant-ph].
[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. **A88**, 023617 (2013), arXiv:1303.5040 [quant-ph].
[34] E. Zohar and M. Burrello, Formulation of lattice gauge theories for quantum simulations, Phys. Rev. **D91**, 054506 (2015), arXiv:1409.3085 [quant-ph].
[35] E. Zohar, J. I. Cirac, and B. Reznik, Quantum Simulations of Lattice Gauge Theories using Ultracold Atoms in Optical Lattices, Rept. Prog. Phys. **79**, 014401 (2016), arXiv:1503.02312 [quant-ph].
[36] E. Zohar, A. Farace, B. Reznik, and J. I. Cirac, Digital lattice gauge theories, Phys. Rev. **A95**, 023604 (2017), arXiv:1607.08121 [quant-ph].
[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), arXiv:1908.06935 [quant-ph].
[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 (2021), arXiv:2101.10227 [quant-ph].
[39] J. Liu and Y. Xin, Quantum simulation of quantum field theories as quantum chemistry (2020), arXiv:2004.13234 [hep-th].
[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), arXiv:1811.03629 [quant-ph].
[41] A. Alexandru, P. F. Bedaque, S. Harmalkar, H. Lamm, S. Lawrence, and N. C. Warrington (NuQS), Gluon field digitization for quantum computers, Phys.Rev.D **100**, 114501 (2019), arXiv:1906.11213 [hep-lat].
[42] A. Yamamoto, Real-time simulation of (2+1)-dimensional lattice gauge theory on qubits, PTEP **2021**, 013B06 (2021), arXiv:2008.11395 [hep-lat].
[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 (2020), arXiv:2006.14160 [quant-ph].
[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 (2021), arXiv:2107.13024 [quant-ph].
[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. **D92**, 076003 (2015), arXiv:1503.08354 [hep-lat].
[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), arXiv:1803.11166 [hep-lat].
[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. **D98**, 094511 (2018), arXiv:1807.09186 [hep-lat].
[49] J. F. Unmuth-Yockey, Gauge-invariant rotor Hamiltonian from dual variables of 3D $U(1)$ gauge theory, Phys. Rev. D **99**, 074502 (2019), arXiv:1811.05884 [hep-lat].
[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 (2020), arXiv:2002.04016 [quant-ph].
[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 (2020), arXiv:2011.13443 [quant-ph].
[52] I. Raychowdhury and J. R. Stryker, Solving Gauss's Law on Digital Quantum Computers with Loop-String-Hadron Digitization (2018), arXiv:1812.07554 [hep-lat].
[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), arXiv:1912.06133 [hep-lat].
[54] Z. Davoudi, I. Raychowdhury, and A. Shaw, Search for Efficient Formulations for Hamiltonian Simulation of non-Abelian Lattice Gauge Theories (2020), arXiv:2009.11802 [hep-lat].
[55] U.-J. Wiese, Towards Quantum Simulating QCD, *Proceedings, 24th International Conference on Ultra-Relativistic Nucleus-Nucleus Collisions (Quark Matter 2014): Darmstadt, Germany, May 19-24, 2014*, Nucl. Phys. **A931**, 246 (2014), arXiv:1409.7414 [hep-th].
[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 (2019), arXiv:1912.11488 [quant-ph].
[57] R. C. Brower, D. Berenstein, and H. Kawai, Lattice Gauge Theory for a Quantum Computer, PoS **LATTICE2019**, 112 (2019), arXiv:2002.10028 [hep-lat].
[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), arXiv:2005.10271 [quant-ph].
[59] H. Singh, Qubit $O(N)$ nonlinear sigma models (2019), arXiv:1911.12353 [hep-lat].
[60] H. Singh and S. Chandrasekharan, Qubit regularization of the $O(3)$ sigma model, Phys. Rev. D **100**, 054505 (2019), arXiv:1905.13204 [hep-lat].
[61] A. J. Buser, T. Bhattacharya, L. Cincio, and R. Gupta, Quantum simulation of the qubit-regularized O(3)-sigma model (2020), arXiv:2006.15746 [quant-ph].
[62] T. Bhattacharya, A. J. Buser, S. Chandrasekharan, R. Gupta, and H. Singh, Qubit regularization of asymptotic freedom (2020), arXiv:2012.02153 [hep-lat].

[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 (2020), arXiv:2012.00020 [hep-th].

[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 (2020), arXiv:2009.07885 [quant-ph].

[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), arXiv:2005.14221 [hep-lat].

[66] E. Gustafson, Prospects for Simulating a Qudit Based Model of (1+1)d Scalar QED, Phys. Rev. D **103**, 114505 (2021), arXiv:2104.10136 [quant-ph].

[67] E. Zohar, Quantum Simulation of Lattice Gauge Theories in more than One Space Dimension – Requirements, Challenges, Methods (2021), arXiv:2106.04609 [quant-ph].

[68] P. Hasenfratz and F. Niedermayer, Asymptotic freedom with discrete spin variables?, *Proceedings, 2001 Europhysics Conference on High Energy Physics (EPS-HEP 2001): Budapest, Hungary, July 12-18, 2001*, PoS **HEP2001**, 229 (2001), arXiv:hep-lat/0112003 [hep-lat].

[69] S. Caracciolo, A. Montanari, and A. Pelissetto, Asymptotically free models and discrete nonAbelian groups, Phys. Lett. **B513**, 223 (2001), arXiv:hep-lat/0103017 [hep-lat].

[70] P. Hasenfratz and F. Niedermayer, Asymptotically free theories based on discrete subgroups, *Lattice field theory. Proceedings, 18th International Symposium, Lattice 2000, Bangalore, India, August 17-22, 2000*, Nucl. Phys. Proc. Suppl. **94**, 575 (2001), arXiv:hep-lat/0011056 [hep-lat].

[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. Krcmar, 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, Physics Letters B **513**, 223 (2001).

[74] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, Lattice Renormalization of Quantum Simulations (2021), arXiv:2107.01166 [hep-lat].

[75] M. Creutz, L. Jacobs, and C. Rebbi, Monte Carlo Study of Abelian Lattice Gauge Theories, Phys. Rev. **D20**, 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. **D24**, 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. **D22**, 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), [,631(1981)].

[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 (2018), arXiv:1810.05338 [hep-th].

[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 Σ (nφ), Journal of Physics A: Mathematical and Theoretical **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. **A99**, 042301 (2019), arXiv:1812.01617 [quant-ph].

[92] J. C. Halimeh and P. Hauke, Reliability of lattice gauge theories (2019), arXiv:2001.00024 [cond-mat.quant-gas].

[93] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS), Suppressing Coherent Gauge Drift in Quantum Simulations (2020), arXiv:2005.12688 [quant-ph].

[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, Annals Phys. **198**, 1 (1990).

[98] M. Creutz, *Quarks, gluons and lattices*, Cambridge Monographs on Mathematical Physics (Cambridge Univ. Press, Cambridge, UK, 1985).
[99] C. Gidney, Constructing large increment gates (2015).
[100] T. Häner, M. Roetteler, and K. M. Svore, Factoring using 2n+ 2 qubits with toffoli based modular multiplication, arXiv preprint arXiv:1611.07995 (2016).
[101] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, Quantum algorithms and circuits for scientific computing, Quantum Information & Computation **16**, 197 (2016).
[102] T. Häner, M. Roetteler, and K. M. Svore, Optimizing quantum circuits for arithmetic, arXiv preprint arXiv:1805.12445 (2018).
[103] Y. Takahashi, S. Tani, and N. Kunihiro, Quantum addition circuits and unbounded fan-out, Quantum Information & Computation **10**, 0872 (2010).
[104] S. Hadfield, On the representation of Boolean and real functions as Hamiltonians for quantum computing, arXiv:1804.09130 (2018).
[105] 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, Physical review A **52**, 3457 (1995).
[106] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, Efficient quantum circuits for diagonal unitaries without ancillas, New Journal of Physics **16**, 033040 (2014).
[107] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the poisson equation, New Journal of Physics **15**, 013021 (2013).
[108] 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 Information Processing **19**, 1 (2020).
[109] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
[110] P. Hoyer, Efficient quantum transforms, arXiv preprint quant-ph/9702028 (1997).
[111] R. Beals, Quantum computation of Fourier transforms over symmetric groups, in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (Citeseer, 1997) pp. 48–53.
[112] 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, 1999) pp. 148–159.
[113] C. Moore, D. Rockmore, and A. Russell, Generic quantum Fourier transforms, ACM Transactions on Algorithms (TALG) **2**, 707 (2006).
[114] A. M. Childs and W. Van Dam, Quantum algorithms for algebraic problems, Reviews of Modern Physics **82**, 1 (2010).
[115] O. Regev, A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, arXiv preprint quant-ph/0406151 (2004).
[116] G. Kuperberg, Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem, arXiv preprint arXiv:1112.3333 (2011).
[117] 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).
[118] D. M. Abrams, N. Didier, M. P. Johnson, Blake R .and da Silva, and C. A. Ryan, Implementation of XY entangling gates with a single calibrated pulse, Nature Electronics **3**, 744 (2020).
[119] M. Reagor *et al.*, Demonstration of universal parametric entangling gates on a multi-qubit lattice, Sci. Adv. **4**, eaao3603 (2018).
[120] 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 Science and Technology **5**, 024003 (2020).
[121] R. S. Smith, M. J. Curtis, and W. J. Zeng, A practical quantum instruction set architecture (2016), arXiv:1608.03355 [quant-ph].
[122] R. S. Smith, E. C. Peterson, M. G. Skilbeck, and E. J. Davis, An open-source, industrial-strength optimizing compiler for quantum programs (2020), arXiv:2003.13961 [quant-ph].
[123] A. D. Hill, M. J. Hodson, N. Didier, and M. J. Reagor, Realization of arbitrary doubly-controlled quantum phase gates (2021), arXiv:2108.01652 [quant-ph].
[124] N. Schuch and J. Siewert, Natural two-qubit gate for quantum computation using the XY interaction, Phys. Rev. A **67**, 032301 (2003).
[125] R. S. Smith, E. C. Peterson, M. G. Skilbeck, and E. J. Davis, An open-source, industrial-strength optimizing compiler for quantum programs, Quantum Science and Technology **5**, 044001 (2020).
[126] E. Peters, A. C. Y. Li, and G. N. Perdue, Perturbative readout error mitigation for near term quantum computers (2021), arXiv:2105.08161 [quant-ph].
[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, Nature Communications **10**, 5347 (2019).
[128] S. Harmalkar, H. Lamm, and S. Lawrence (NuQS), Quantum Simulation of Field Theories Without State Preparation (2020), arXiv:2001.11490 [hep-lat].
[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), arXiv:2011.11677 [hep-lat].
[130] H. Lamm, S. Lawrence, and Y. Yamauchi (NuQS), Parton physics on a quantum computer, Phys. Rev. Res. **2**, 013272 (2020), arXiv:1908.10439 [hep-lat].

[131] T. D. Cohen, H. Lamm, S. Lawrence, and Y. Yamauchi, Quantum algorithms for transport coefficients in gauge theories (2021), arXiv:2104.02024 [hep-lat].

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. Each of the $2N$ elements of D_N can be uniquely expressed as $s^m r^k$, where $m \in \{0,1\}$ and $k \in \{0,1,\ldots,N-1\}$. 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 is like a rotation in the opposite direction. Hence it follows that D_N is isomorphic to the semidirect product of cyclic groups $\mathbb{Z}_N \rtimes \mathbb{Z}_2$ for each N.

Observe that $s = r^N s$, and further that if we assert $sr^k = r^{N-k}s$, then

$$\begin{aligned} sr^{k+1} &= sr^k r \\ &= r^{N-k} sr \\ &= r^{N-k} r^{N-1} s \\ &= r^N r^{N-k-1} s \\ &= r^{N-(k+1)} s \end{aligned} \tag{A1}$$

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

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

Through a very similar calculation, we also have

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

Using the above, we find the product rule given in Eq. (19):

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

Using Eq. (A2), we find that the inverse of sr^k is given by

$$\left(sr^k\right)^{-1} = r^{-k} = r^{N-k}s = sr^k \tag{A5}$$

while the inverse of r^k is simply $\left(r^k\right)^{-1} = r^{N-k}$ so that in general, the inverse of a D_N element is given by Eq. (18), i.e., $\left(s^m r^k\right)^{-1} = s^m r^{-(-1)^m k} = s^m r^{Nm-(-1)^m k}$.

Appendix B: Proof that Fourier gate diagonalizes D_4 Kinetic gate

Given $M_{ij} = \mathrm{Re}\left[\mathrm{Tr}\left(\rho^\dagger(g_i)\rho(g_j)\right)\right]$ and the matrix T with entries $T_{ij} = \exp \beta M_{ij}$, we want to show that $FTF^\dagger$, where F is the unitary matrix corresponding to the nonabelian Fourier transform, is diagonal.

Note that M is dependent on the representation we use. There are a total of 4 1D irreducible representations of the D_4 group, which map each of the elements $|000\rangle,\ldots,|111\rangle$ respectively to

1. $\rho_a = (1,1,1,1,1,1,1,1)$
2. $\rho_b = (1,1,1,1,-1,-1,-1,-1)$
3. $\rho_c = (1,-1,1,-1,1,-1,1,-1)$
4. $\rho_d = (1,-1,1,-1,-1,1,-1,1)$

and a single 2D irreducible representation, which upto a change of basis, is specified by mapping elements $|abc\rangle$ onto

$$\phi(g_{abc}) = \left[\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right]^a \left[\begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}\right]^{2b+c} \tag{B1}$$

For the 2D irrep, we note that M_ij is non-zero only when $i = j$ or when g_i and g_j are each others' inverses. Given the encoding for the elements of the D_4 group into qubits as described in the main text, this means that the only off-diagonal non-zero elements of M are given at $(000, 010)$, $(001, 011)$, $(100, 110)$, and $(101, 111)$. In matrix form, this looks like

$$M = \begin{pmatrix} 2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -2 & 0 & 0 & 0 & 0 \\ -2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 \\ 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 \end{pmatrix} \tag{B2}$$

where the non-zero factor of ± 2 appears since $\mathrm{Tr}(\mathbb{I}_{2x2}) = 2$. Now the Fourier matrix is built out of the inequivalent irreducible representations of the D_4 group, and can be represented as

$$F = \begin{pmatrix} \rho_a \\ \rho_b \\ \rho_c \\ \rho_d \\ \phi_{00} \\ \phi_{01} \\ \phi_{10} \\ \phi_{11} \end{pmatrix} \tag{B3}$$

Let $F' = FT$ and $\rho'_a, dots, \rho'_d, \phi'_{00}, \ldots, \phi'_{11}$ denote its rows. We find that

$$\rho'_a = (\exp(\beta d) + \exp(-\beta d) + 6)\,\rho_a \tag{B4}$$

$$\rho'_{b,c,d} = (\exp(\beta d) + \exp(-\beta d) - 2)\,\rho_{b,c,d} \tag{B5}$$

$$\phi'_{ij} = (\exp(\beta d) - \exp(-\beta d))\,\phi_{ij} \tag{B6}$$

Finally, using the Schur orthogonality relations which state that for two inequivalent irreducible unitary representations of some finite group G, $\phi : G \to U_n(\mathbb{C})$ and $\rho : G \to U_m(\mathbb{C})$, we have

- $\langle \phi_{kl}, \rho_{ij} \rangle = 0$,
- $\langle \phi_{kl}, \phi_{ij} \rangle = \delta_{ik}\delta_{jl}$

we find

$$\begin{aligned} FTF^\dagger = \mathrm{Diag}\,(&\exp(\beta d) + \exp(-\beta d) + 6, \exp(\beta d) + \exp(-\beta d) - 2, \\ &\exp(\beta d) + \exp(-\beta d) - 2, \exp(\beta d) + \exp(-\beta d) - 2, \\ &\exp(\beta d) - \exp(-\beta d), \exp(\beta d) - \exp(-\beta d), \\ &\exp(\beta d) - \exp(-\beta d), \exp(\beta d) - \exp(-\beta d))) \end{aligned} \tag{B7}$$