

**Continuous-time quantum walks on dynamic graphs**

Rebekah Herrman\*

*Department of Mathematical Sciences, The University of Memphis, Memphis, Tennessee 38152-3520, USA*

Travis S. Humble†

*Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-6015, USA*

(Received 18 February 2019; published 8 July 2019)

Continuous-time quantum walks (CTQWs) on static graphs provide efficient methods for search and sampling as well as a model for universal quantum computation. We consider an extension of CTQWs to the case of dynamic graphs, in which an ordered sequence of graphs governs free evolution of the quantum walk. We then consider how perfect state transfer during the quantum walk can be used to design dynamic graphs that implement a universal set of quantum logic gates. We give explicit examples for a complete logical basis, and we validate implementations using numerical simulations for quantum teleportation and addition circuits. Finally, we discuss the potential for realizing CTQWs on dynamic graphs using actively controlled quantum optical waveguides.

DOI: [10.1103/PhysRevA.100.012306](https://doi.org/10.1103/PhysRevA.100.012306)**I. INTRODUCTION**

Quantum walks offer a unique paradigm for using quantum mechanics to perform computation [1], where a walk may represent either the discrete or continuous-time propagation of a quantum state over a graph [2,3]. In a continuous-time quantum walk (CTQW), free evolution of an  $N$ -dimensional quantum state under a Hamiltonian is represented by probability amplitudes assigned to each vertex in a graph on  $N$  vertices. The CTQW was originally envisioned as a method for sampling decision trees [4] and later applied to a variety of search and sampling problems on  $d$ -dimensional lattices, searches on balanced trees, as well as quantum navigation of networks [5–10]. Moreover, Childs has shown that CTQWs on time-independent graphs offer a novel model for universal quantum computation [11,12], while Qiang *et al.* have described how efficient implementations of CTQWs may be useful for comparing the broader computational power of quantum computing to conventional computing models [13].

In a typical CTQW, the Hamiltonian is interpreted as the connectivity of the underlying graph on which the quantum state evolves. The graph connectivity determines the evolution of the quantum state and specific graphs have been found to demonstrate well-defined quantum walk behaviors. For example, perfect state transfer occurs in a quantum walk when the amplitude assigned to a subset of vertices transfers with unit probability to a distinct vertex set within a well-defined period of evolution [14]. Kendon and Tamon have surveyed perfect state transfer for a number of several specific graphs including the singleton graph  $K_1$ , the complete graph on two vertices  $K_2$ , the path graph on three vertices, and the cycle on four vertices

$C_4$  [15]. Perfect state transfer has also been shown to exist for graphs on more vertices, including certain graph products, weighted join graphs, and quotient graphs [16–18].

The versatility of CTQWs across many known types of graphs motivates our consideration for how quantum walks may behave on dynamic graphs. We define a dynamic graph as a well-defined sequence of static graphs in which the CTQW evolution changes at specific transition times. In the dynamic graphs discussed below, we use perfect state transfer under the component static graphs to demonstrate how more complex unitary processes can be realized. We provide explicit realizations of quantum walks on dynamic graphs for realizing a complete set of computational gates, and we then illustrate how compositions of multiple walks correspond to examples of quantum circuits. This formalism establishes a connection between CTQWs on dynamic graphs and the gates found in the conventional quantum circuit model. We also provide a connection between this model of computation and the development of tunable optical waveguides for performing continuous-time quantum walks.

Our approach to quantum walks on dynamic graphs shares similarities with Childs' model for universal quantum computation [11,12]. Both approaches draw on the use of unweighted and relatively sparse graphs to formalize state transfer as well as the composition of such graphs to describe more complex operations. However, the models differ in the types of underlying graphs as Childs relies on strictly static graphs while we employ dynamic graphs. Another closely related model is the hybrid quantum walk proposed by Underwood and Feder, which combines concepts from both continuous and discrete walk models [19]. In that work, a series of weighted adjacency matrices corresponding to distinct graphs are used to propagate a quantum state. They refer to this model as a discontinuous quantum walk, where free evolution is again based on widgets that control propagation dynamics. Underwood and Feder emphasize the use of a dual-railing

\*rherrman@memphis.edu

†humblets@ornl.gov

encoding to represent individual qubits and the interleaving of continuous and discrete quantum walks to perform computation. By comparison, we design quantum walks on dynamic graphs to implement a sequence of continuous-time evolutions that perform quantum logic using perfect state transfer in the native vertex space. Du *et al.* considered the task of designing a quantum walk to implement a single-qubit  $X$  gate by walking on a single static, weighted graph [20], whereas our work develops implementations for a complete gate set using dynamics graphs. Chakraborty *et al.* have explored spatial search using CTQW on time-ordered sequences of random graphs, for which they demonstrated a threshold for the optimal run time using Grover's algorithm [21], while our work uses deterministic, time-ordered sequences to carry out discrete logic gates.

The paper is organized as follows. Following a review of CTQWs on static graphs in Sec. II, we describe quantum walks on disconnected graphs in Sec. III and dynamic graphs in Sec. IV. Using this formalism, we design a series of quantum walks that implement elementary logic gates in Sec. V, and we demonstrate how the sparsely connected dynamic graphs may be composed to correspond with gate-based circuits in Sec. VI. We offer a discussion on these results in Sec. VII, where we establish a connection between dynamic quantum walks and current approaches to designing quantum computing hardware based on optical waveguides and ion trap technologies.

## II. CONTINUOUS-TIME QUANTUM WALKS

Consider an undirected graph  $G = (V, E)$  with a canonically labeled vertex set  $V = \{0, 1, \dots, N-1\}$  of  $N$  vertices and an edge set  $E = \{(i, j) : i \sim j\}$ . We allow no multiedges in the graph, i.e., there can be at most one edge incident with any two vertices. However, we do allow for a self-loop on a vertex  $v \in V$  if and only if there does not exist  $u \in V$  such that  $u \neq v$  and  $u \sim v$ . Additionally, the edges of  $G$  are undirected. Let  $B_G = \{|j\rangle : \forall j \in V\}$  be a linearly independent basis for the complex vector space  $\mathbb{C}^N$  with the inner product  $\langle j|k\rangle = \delta_{jk}$ . Graphs  $G$  and  $G' = (V', E')$  have the same basis if  $V = V'$ . The Hamiltonian for the graph  $G$  is denoted as  $H_G$  and is defined as the adjacency matrix of the graph as given by the edge set  $E$ . The adjacency matrix  $A$  of  $G$  is a 0–1 valued  $N \times N$  matrix such that for  $u, v \in V(G)$ , if  $u \sim v$ ,  $A_{u,v} = A_{v,u} = 1$ , and 0 otherwise. We will use the convention that if a vertex  $v$  is not adjacent to any other vertices then  $A_{v,v} = 1$ , a convention also used in studies of classical random walks. The resulting real-valued adjacency matrix  $A$  is symmetric about the main diagonal.

We define the quantum state of a graph  $G$ , or graph state for short, as a normalized vector  $|\psi_G\rangle \in B_G$  such that

$$|\psi_G\rangle = \sum_{j \in V} c_j |j\rangle, \quad (1)$$

with  $c_j \in \mathbb{C}$  and

$$\langle \psi_G | \psi_G \rangle = \sum_{j \in V} |c_j|^2 = 1. \quad (2)$$

For a continuous-time quantum walk, the graph state transforms with respect to time  $\tau$  under the Schrödinger

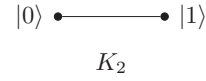


FIG. 1. The  $K_2$  graph supports perfect state transfer between the two vertices labeled by the single-qubit computational basis states  $|0\rangle$  and  $|1\rangle$ .

equation,

$$i\hbar \frac{\partial |\psi_G(\tau)\rangle}{\partial \tau} = H_G |\psi_G(\tau)\rangle, \quad (3)$$

where  $\hbar$  is Planck's constant divided by  $2\pi$ . When the Hamiltonian is constant over the interval  $[t_0, t]$ , the formal solution to Eq. (3) is given by the propagation operator,

$$U_G(t, t_0) = e^{-iH_G(t-t_0)/\hbar}, \quad (4)$$

such that

$$|\psi_G(t)\rangle = U_G(t, t_0) |\psi_G(t_0)\rangle, \quad (5)$$

where the boundary condition  $|\psi_G(t_0)\rangle$  is the state at time  $t_0$ . The propagation operator  $U_G$  is unitary since  $H_G$  is Hermitian. We say that a graph  $G$  admits perfect state transfer between unique vertices  $u, v \in V(G)$  at time  $t$  if

$$U_G(t, 0)|u\rangle = a|v\rangle, \quad (6)$$

where  $a \in \mathbb{C}$  such that  $|a| = 1$ .

There are several well-known examples that illustrate perfect state transfer using CTQW on static graphs. The singleton graph  $K_1$  has vertex set  $V = \{0\}$  and an empty edge set  $E = \emptyset$ . As the lone vertex  $|0\rangle$  is adjacent to no other vertices during the CTQW, we represent the unitary dynamics by a self-loop. The  $K_1$  Hamiltonian is then represented in its eigenbasis as

$$H_{K_1} = \lambda_1 |0\rangle\langle 0|, \quad (7)$$

where  $\lambda_1$  is the real-valued energy eigenvalue, and the normalized state,

$$|\psi_{K_1}(t_0)\rangle = c_0(t_0)|0\rangle, \quad (8)$$

has  $|c_0| = 1$  for all time such that

$$|\psi_{K_1}(t)\rangle = e^{-i\nu_1 t} |0\rangle, \quad (9)$$

where  $\nu_1 = \lambda_1/\hbar$  is the frequency.

As a second example, the complete graph on two vertices  $K_2$  shown in Fig. 1 has vertex set  $V = \{0, 1\}$  and edge set  $E = \{(0, 1)\}$ . We specify the Hamiltonian for  $K_2$  as the free evolution operator over degenerate basis states, i.e.,  $\langle 0|H_{K_2}|0\rangle = \langle 1|H_{K_2}|1\rangle$ , which offers a natural representation of a qubit in a degenerate eigenbasis. Setting this eigenenergy to zero, the Hamiltonian is represented as

$$H_{K_2} = \lambda_2(|0\rangle\langle 1| + |1\rangle\langle 0|), \quad (10)$$

where the eigenvalue  $\lambda_2$  defines the energy scale and the characteristic frequency  $\nu_2 = \lambda_2/\hbar$ . The time propagator for  $K_2$  may be decomposed by series expansion as

$$U_{K_2}(t, t_0) = \cos[\nu_2(t-t_0)]\mathbb{I}_2 - i \sin[\nu_2(t-t_0)]H_{K_2}, \quad (11)$$

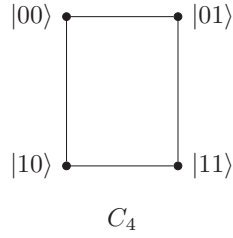


FIG. 2. The  $C_4$  graph supports perfect state transfer between vertices labeled by the two-qubit computational states.

where  $\mathbb{I}_N$  is the  $N$ -dimensional identity matrix. The  $K_2$ -graph state evolves as

$$|\psi_{K_2}(t)\rangle = (c_0 \cos[\nu_2(t-t_0)] - ic_1 \sin[\nu_2(t-t_0)])|0\rangle + (c_1 \cos[\nu_2(t-t_0)] - ic_0 \sin[\nu_2(t-t_0)])|1\rangle, \quad (12)$$

which is capable of perfect state transfer up to a trivial phase factor for propagation time  $t = \frac{\pi}{2\nu_2}$  [15].

As a final example, the cycle graph  $C_4$  shown in Fig. 2 has a vertex set  $V = \{0, 1, 2, 3\}$ , edge set  $E = \{(0, 1), (0, 2), (1, 3), (2, 3)\}$ , and Hamiltonian

$$H_{C_4} = \lambda_4(|0\rangle\langle 1| + |0\rangle\langle 2| + \text{H.c.}), \quad (13)$$

where H.c. denotes the Hermitian conjugate,  $\lambda_4$  is the energy scale, and  $\nu_4 = \lambda_4/\hbar$  defines the characteristic frequency. The propagation operator may be decomposed as

$$U_{C_4}(t, t_0) = \mathbb{I}_4 + \frac{1}{2} \cos(2\nu_4(t-t_0))H_{C_4}^2 - \frac{i}{2} \sin(2\nu_4(t-t_0))H_{C_4} \quad (14)$$

to yield the state  $|\psi_{C_4}(t)\rangle$  with coefficients in the nodal basis as

$$c_0(t) = \frac{1}{2}[c_0(1 + \cos(2t)) - i \sin(2t)(c_1 + c_2) + c_3(-1 + \cos(2t))], \quad (15)$$

$$c_1(t) = \frac{1}{2}[-i \sin(2t)(c_0 + c_3) + c_1(1 + \cos(2t)) + c_2(-1 + \cos(2t))], \quad (16)$$

$$c_2(t) = \frac{1}{2}[-i \sin(2t)(c_0 + c_3) + c_1(-1 + \cos(2t)) + c_2(1 + \cos(2t))], \quad (17)$$

and

$$c_3(t) = \frac{1}{2}[c_0(-1 + \cos(2t)) - i \sin(2t)(c_1 + c_2) + c_3(1 + \cos(2t))]. \quad (18)$$

Perfect state transfer in  $C_4$  is a special instance of the case of an  $N$ -dimensional hypercube [15], which has been shown by Christandl *et al.* to be capable of perfect state transfer for all  $N$  at time  $t = \frac{\pi}{2\nu_N}$  [22,23]. For the remainder of our presentation, we will simplify the analysis to the case that  $\nu_N = 1$  for  $k = 1, 2, 4$  and we will set  $\hbar = 1$  for convenience.

### III. QUANTUM WALKS ON DISCONNECTED GRAPHS

We now consider quantum walks on disjoint graphs  $G_1$  and  $G_2$  with  $G_j = (V_j, E_j)$ , where the disjoint union  $G = G_1 + G_2$  has vertex set  $V = V_1 \cup V_2$  and edge set  $E = E_1 \cup E_2$ . We require that  $G_1$  and  $G_2$  are disconnected graphs, termed components of the graph  $G$ , and that  $V_1 \cap V_2 = \emptyset$ . The basis for the disjoint union  $G$  is  $B_G = B_{G_1} \oplus B_{G_2}$  and a composite quantum state for  $G$  takes the form,

$$|\psi_G\rangle = |\psi_{G_1}\rangle \otimes |\psi_{G_2}\rangle, \quad (19)$$

with  $\oplus$  the direct sum and  $\otimes$  the Kronecker product. The corresponding Hamiltonian is defined as  $H(G) = H_{G_1} \oplus H_{G_2}$ , which yields decoupled equations of motion,

$$i \frac{\partial |\psi_{G_j}(t)\rangle}{\partial t} = H_{G_j} |\psi_{G_j}(t)\rangle, \quad j = 1, 2, \quad (20)$$

and a composite time propagator,

$$U_{G_1+G_2}(t, t_0) = e^{-iH_{G_1}(t-t_0)} \otimes e^{-iH_{G_2}(t-t_0)}. \quad (21)$$

The graph state of  $G$  is modeled by two disconnected states  $|\psi_{G_1}\rangle \in B_{G_1}$  and  $|\psi_{G_2}\rangle \in B_{G_2}$  and

$$|\psi_G(t)\rangle = U_{G_1}(t, t_0)|\psi_{G_1}(t_0)\rangle \otimes U_{G_2}(t, t_0)|\psi_{G_2}(t_0)\rangle. \quad (22)$$

As an example, consider the empty graph on  $N$  vertices  $\bar{K}_N$ , which is the complement of the complete graph  $K_N$  and expressed as the union,

$$\bar{K}_N = \bigcup_{j=0}^{N-1} K_1^{(j)}, \quad (23)$$

where  $K_1^{(j)}$  is the singleton graph with vertex label  $j$ . The composite Hamiltonian is the direct sum of  $N$  singleton Hamiltonians,

$$H_{\bar{K}_N} = \bigoplus_{j=0}^{N-1} H_{K_1^{(j)}}, \quad (24)$$

and the quantum state is defined as the Kronecker product of the individual states,

$$|\psi_{\bar{K}_N}\rangle = \bigotimes_{j=0}^{N-1} |\psi_{K_1^{(j)}}\rangle, \quad (25)$$

in the basis formed from the union of these graphs,

$$B_{\bar{K}_N} = \bigcup_{j=0}^{N-1} B_{K_1^{(j)}}. \quad (26)$$

The graph state for the  $j$ th singleton graph is now given as

$$U_{K_1^{(j)}}(t, t_0)|j\rangle = e^{-i\nu_1^{(j)}t}|j\rangle, \quad (27)$$

with  $\nu_1^{(j)}$  the energy eigenvalue of the  $j$ th vertex. We will assume that the vertices are indistinguishable and therefore  $\nu_1^{(j)} = \nu_1$  for all  $j$ . Thus, the Hamiltonian of these  $N$  disjoint identical vertices,

$$H_{\bar{K}_N} = \bigoplus_{j=0}^{N-1} \lambda_1 |j\rangle\langle j| = \lambda_1 \mathbb{I}_N, \quad (28)$$

is proportional to the  $N$ -dimensional identity operator  $\mathbb{I}_N$  over the basis  $B_{\tilde{K}_N}$ . This yields an  $N$ -fold Kronecker sum of  $K_1$  states with the form of Eq. (9).

As a second example, consider the disjoint union  $K_2 + K_1$  with Hamiltonian

$$H_{K_2+K_1} = H_{K_2} \oplus H_{K_1}, \quad (29)$$

represented as

$$H_{K_2+K_1} = \begin{pmatrix} 0 & \lambda_2 & 0 \\ \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_1 \end{pmatrix}. \quad (30)$$

The composite state of this disjoint graph propagates as

$$U_{K_2+K_1}(t, t_0)|\psi_{K_2+K_1}\rangle = U_{K_2}(t, t_0)|\psi_{K_2}(t_0)\rangle \oplus U_{K_1}(t, t_0)|\psi_{K_1}(t_0)\rangle, \quad (31)$$

and may be recast as

$$\begin{aligned} U_{K_2+K_1}(t, t_0)|\psi_{K_2+K_1}\rangle &= [c_1 \cos(v_2(t - t_0)) - ic_1 \sin(v_2(t - t_0))]|0\rangle \\ &\quad + [c_1 \cos(v_2(t - t_0)) - ic_0 \sin(v_2(t - t_0))]|1\rangle \\ &\quad + c_2 e^{-iv_1 t}|2\rangle. \end{aligned} \quad (32)$$

#### IV. QUANTUM WALKS ON DYNAMIC GRAPHS

We next consider quantum walks on dynamic graphs, in which a dynamic graph  $\mathcal{G} = \{(G_\ell, t_\ell)\}$  is a set of graphs  $G_\ell = (V_\ell, E_\ell)$  with associated propagation times  $t_\ell < t_{\ell+1}$  for  $\ell \in \mathbb{Z}$ . In subsequent discussion, we will consider the case that only the edge sets change while the vertex sets stay constant, i.e.,  $V_\ell = V$ , such that the bases for all  $G_\ell$  are the same. However, the case of changing vertex sets is equally valid as this represents the growth and reduction of the underlying Hilbert space, for example, through the addition or removal of ancillary vertices.

The Hamiltonian of a dynamic graph  $\mathcal{G}$  is expressed as the weighted sum,

$$H_{\mathcal{G}} = \sum_{\ell=0}^{L-1} H_{G_\ell} \Pi_\ell(t), \quad (33)$$

where transitions between graphs are modulated by the functions  $\Pi_\ell(t)$ . We consider the explicit case that the  $\ell$ th transition function takes the form of the  $\ell$ th rectangle function:

$$\Pi_\ell(t) = \begin{cases} 1 & t_\ell < t < t_{\ell+1}, \\ 0 & \text{otherwise} \end{cases}, \quad (34)$$

with  $[t_0, t_L]$  the interval over which the entire walk is defined. The dynamics is then expressed as a sequence of propagations through the series of Schrödinger equations,

$$i \frac{\partial |\psi_{G_\ell}(\tau)\rangle}{\partial \tau} = H_{G_\ell} |\psi_{G_\ell}(\tau)\rangle, \quad t_\ell < \tau < t_{\ell+1}. \quad (35)$$

As the set of discontinuities is countable, the function is still Riemann integrable, and this system of equations yields the

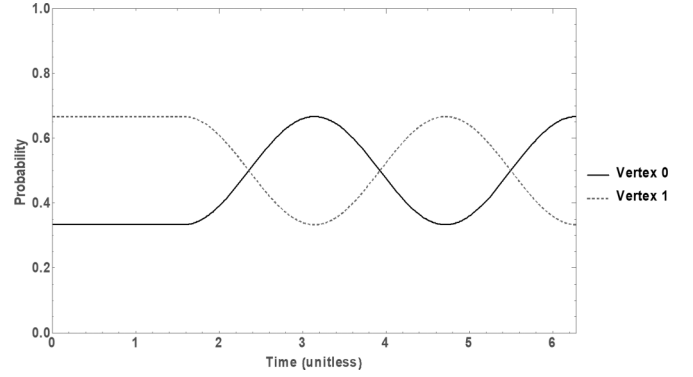


FIG. 3. The time-dependent probability densities of two vertices, 0 and 1, as the state propagates under  $K_1 + K_1$  for  $t = \frac{\pi}{2}$  units of time before switching to  $K_2$  and propagating for an additional time  $t = \frac{3\pi}{2}$ . In this example, the initial state  $\sqrt{\frac{1}{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ .

composite propagation operator,

$$U_{\mathcal{G}}(t_L, t_0) = \prod_{\ell=0}^{L-1} e^{-iH_{G_\ell}(t_{\ell+1}-t_\ell)}, \quad (36)$$

which is understood to be a product ordered from right to left with increasing index. The quantum state of the dynamic graph  $\mathcal{G}$  is then defined under this operator transform as

$$|\psi_{\mathcal{G}}(t)\rangle = U_{\mathcal{G}}(t_L, t_0)|\psi_{\mathcal{G}}(t_0)\rangle, \quad (37)$$

with initial condition  $|\psi_{\mathcal{G}}(t_0)\rangle \in B_{\mathcal{G}}$  and  $\langle\psi_{\mathcal{G}}(t_0)|\psi_{\mathcal{G}}(t_0)\rangle = 1$ .

As a simple example of a quantum walk on a dynamic graph, consider the case of two disjoint  $K_1$  graphs switched to a bipartite  $K_2$ . The dynamic graph is expressed as  $\mathcal{G} = \{(K_1 + K_1, t_0), (K_2, t_1)\}$ , where  $t_0$  and  $t_1$  denote the transition times. Taking the initial quantum state as a superposition over the nodal basis, Fig. 3 plots the time-dependent probability for each basis state with respect to the propagation time. Initially under the  $K_1 + K_1$ , the probability remains constant until the transition time  $t_1$ , after which the Hamiltonian switches to  $K_2$  and creates an edge between vertices. This leads to the oscillations in probability as expected by Eq. (11). Figure 4 is

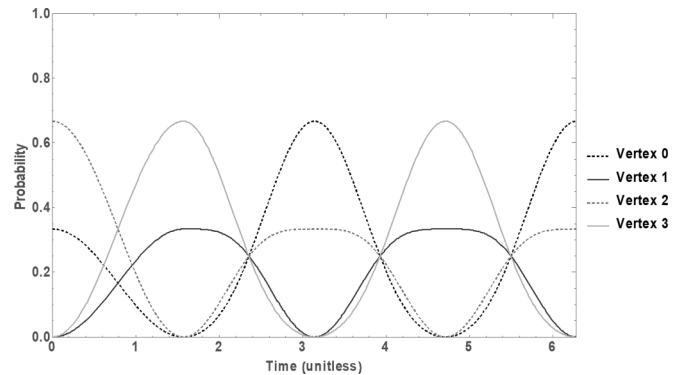


FIG. 4. The time-dependent probability density of four vertices, 0, 1, 2, and 3, as the state propagates under  $K_2 + K_2$  for time  $t = \frac{\pi}{2}$  followed by  $C_4$  for time  $t = \frac{3\pi}{2}$ . In this example, the initial state is  $\sqrt{\frac{1}{3}}|00\rangle + \sqrt{\frac{2}{3}}|10\rangle$ .

an example of two  $K_2$  graphs allowed to propagate on their own and then connected as a  $C_4$  and allowed to propagate again.

**V. QUANTUM WALKS FOR ELEMENTARY GATES**

The formalism of quantum walks on dynamic graphs may be used to realize one- and two-qubit gates within the quantum circuit model by identifying the quantum walk on a graph of  $|V| = N = 2^n$  vertices with a corresponding  $n$ -qubit circuit. Let the vertex label  $v \in V$  map to the computational basis state  $|v_1, v_2, \dots, v_n\rangle$  with  $v_i$  the  $i$ th coefficient in the binary expansion of the  $n$ -bit, non-negative integer  $v$ . We demonstrate several explicit examples of how few-qubit quantum gates can be realized using perfect state transfer limited to  $K_1$ ,  $K_2$ , and  $C_4$  graphs. We limit our CTQWs to those on  $K_1$ ,  $K_2$ , and  $C_4$  because the periods are all multiples of  $\pi$  and achieve perfect state transfer at times  $\frac{k\pi}{2}$  for  $k \in \mathcal{N}$ . In fact, we use the  $K_1^{(i)}$  graph exclusively to add a phase factor to the  $i$ th qubit. We add the phase factors for sake of completeness in some of the CTQWs, but for implementation purposes, the phase factor may be omitted if desired by removing the appropriate CTQWs. We show that in some instances, such as the  $Z$  gate, the realization of gate logic within the quantum walk model requires additional vertices whereas other gates, such as CNOT and CCNOT, are straightforward to realize.

The realization of elementary gates from the circuit model provides a constructive approach to demonstrate the completeness of quantum walks on dynamics graphs. While the quantum walk formalism can naturally represent any unitary of the form  $\exp(iAt)$ , we have imposed the restriction that the Hermitian matrix  $A$  must represent the connectivity of the dynamic graph and that these graphs should be limited to a small number of vertices. By demonstrating that a complete basis of elementary gates can be constructed under these restrictions, we can then invoke the Solovay-Kitaev theorem to establish universality. The Solovay-Kitaev theorem establishes the feasibility of approximating an arbitrary unitary transformation when only a limited subset of such transformation may be accessed [24]. We demonstrate an explicit realization for a universal set of gates, including the Pauli,  $H$ ,  $T$  and CNOT gates described below, from which it follows that sequences of these gates of length  $O(\log^c(1/\epsilon))$  may approximate an arbitrary unitary within precision  $\epsilon$  for constant  $c \approx 3.97$  [24].

**A. Single-qubit gates**

The Pauli gates provide a set of single-qubit operations represented in the computational basis as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (38)$$

We can implement these gates exactly using perfect state transfer within a dynamic graph. For example, the  $X$  gate may be implemented on two graph vertices using a quantum walk on  $K_2$ . For simplicity, we assume the vertices are labeled 0 and 1 and that the graph state is initially prepared as  $c_0|0\rangle + c_1|1\rangle$ . The walk under  $K_2^{(0,1)}$  for a period of  $\frac{3\pi}{2}$  prepares the state  $i(c_1|0\rangle + c_0|1\rangle)$ . The resulting global phase factor of  $i$  may be removed by evolving under  $K_1^{(0)} + K_1^{(1)}$  for a second period of



FIG. 5. A dynamic graph representation of the  $X$  gate consists of two graphs and the associated propagation times. This sequence of CTQW executes the logical bit flip operation on the graph state.

$\frac{\pi}{2}$ , and we include these dynamics in our definition of the  $X$  gate. The dynamic graph for the  $X$  gate is defined as

$$\mathcal{G}_X = \left\{ \left( G_{K_2}^{(0,1)}, \frac{3\pi}{2} \right), \left( G_{K_1}^{(0)} + G_{K_1}^{(1)}, \frac{\pi}{2} \right) \right\}, \quad (39)$$

and Fig. 5 provides a graphical representation. When the target pair of vertices is embedded in a larger graph state, it is understood that all other nodes evolve disjointly from the above dynamic graph.

The  $Z$  gate may be implemented using a  $K_1$  and  $C_4$  defined on five vertices. Notice that  $|001\rangle$  must propagate as a singleton for  $\pi$  units of time to flip the sign of the coefficient, however,  $|000\rangle$  needs to propagate as a  $C_4$  in the same time frame in order to keep its original sign. We maintain a clear correspondence with the circuit model by using a graph on eight vertices which represent the full Hilbert space for three qubits. Three of these vertices will propagate as singletons for the entirety of the walk. For example, given the initial state  $c_0|0\rangle + c_1|1\rangle$  for a graph of  $|V| = 8$  vertices, the dynamic graph representing the  $Z$  gate is defined as

$$\mathcal{G}_Z = \left\{ \left( G_{C_4}^{(0,2,4,6)} + G_{K_1}^{(1)} + G_{K_1}^{(3)} + G_{K_1}^{(5)} + G_{K_1}^{(7)}, \pi \right) \right\}. \quad (40)$$

A graphical representation of the walk for the  $Z$  gate is shown in Fig. 6. Note this dynamic flips the signs of  $|011\rangle$ ,  $|101\rangle$ , and  $|111\rangle$  in addition to  $|001\rangle$

A  $Y$  gate may be derived from the commutation relations for the Pauli operators and implemented by performing the  $X$  and  $Z$  gates in series. An additional phase shift of  $i$  is

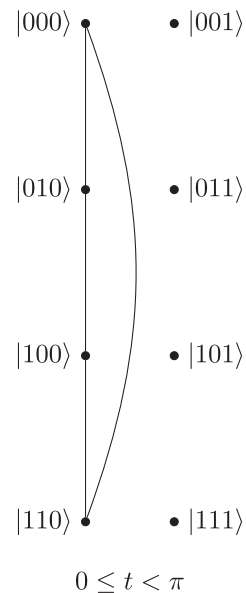


FIG. 6. A graphical representation of the  $Z$  gate using CTQW on  $\mathcal{G}_Z$ .



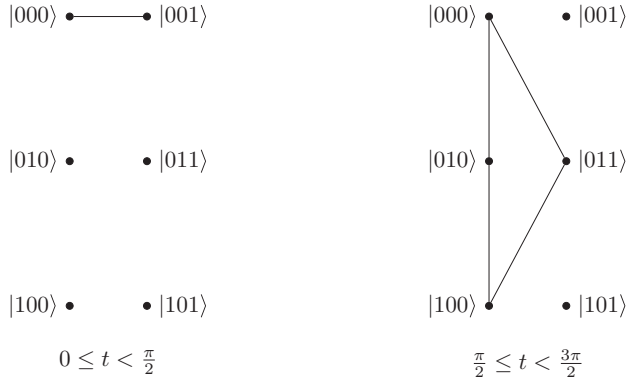


FIG. 7. A graphical representation of the  $Y$  gate using CTQW on  $\mathcal{G}_Y$ .

required and this may be recovered by evolving all vertices under disjoint singletons for  $t = 3\pi/2$ . Of course, reversing the order in which the  $X$  and  $Z$  gates are performed would change the necessary phase shift,  $-iY$ . Alternatively, the  $Y$  transformation may be implemented by propagating vertices  $|000\rangle$  and  $|001\rangle$  under  $K_2$  for  $\pi/2$  units of time, then allowing vertex  $|001\rangle$  to propagate as a singleton for  $\pi$  units while simultaneously allowing  $|000\rangle$  to propagate as a  $C_4$  to three new vertices. The dynamic graph for the latter  $Y$  operation is given as

$$\mathcal{G}_Y = \left\{ \left( G_{K_2}^{(0,1)} + G_{K_1}^{(2)} + G_{K_1}^{(3)} + G_{K_1}^{(4)}, \frac{\pi}{2} \right), \left( G_{K_1}^{(1)} + G_{C_4}^{(0,2,3,4)}, \pi \right) \right\}, \quad (41)$$

which is illustrated in Fig. 7. Completing the Pauli group, we note that the identity gate may be implemented using a

number of different dynamic graphs. This includes assigning every vertex to propagate under the singleton graph for  $t = 2\pi$ , connecting pairs of vertices as  $K_2$  graphs for  $t = 2\pi$ , or connecting four vertices as a  $C_4$  and propagating for  $t = \pi$ . The best choice for implementation is likely to be determined by other scheduling concerns.

The single-qubit Hadamard gate is defined in the computational basis as

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (42)$$

and may be implemented using a series of  $C_4$  and  $K_2$  graphs. The Hadamard gate may be performed with only five vertices, but we again use eight vertices to establish a clear correspondence with three qubits in the circuit model. Consider the initial state  $c_0|0\rangle + c_1|1\rangle$  embedded in a Hilbert space represented by  $|V| = 8$  nodes. Figure 8 illustrates the dynamic graph for the  $H$  gate, defined as

$$\begin{aligned} \mathcal{G}_H = \{ & (G_{C_4}^{(0,2,4,6)} + G_{K_1}^{(1)} + G_{K_1}^{(3)} + G_{K_1}^{(5)} + G_{K_1}^{(7)}, 3\pi/2), \\ & (G_{K_2}^{(0,7)} + G_{K_2}^{(1,6)} + G_{K_2}^{(2,5)} + G_{K_2}^{(3,4)}, \pi/4), \\ & (G_{C_4}^{(0,2,4,6)} + G_{K_1}^{(1)} + G_{K_1}^{(3)} + G_{K_1}^{(5)} + G_{K_1}^{(7)}, 3\pi/2), \\ & (G_{K_2}^{(0,1)} + G_{K_2}^{(2,3)} + G_{K_2}^{(4,5)} + G_{K_2}^{(6,7)}, \pi/2), \\ & (G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_1}^{(2)} + G_{K_1}^{(3)} + G_{K_1}^{(4)} \\ & + G_{K_1}^{(5)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, 3\pi/2) \}. \end{aligned} \quad (43)$$

We show in the Appendix that the CTQW defined by Eq. (43) implements the logical transformation for the Hadamard gate.

The  $T$  gate is defined as

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \quad (44)$$

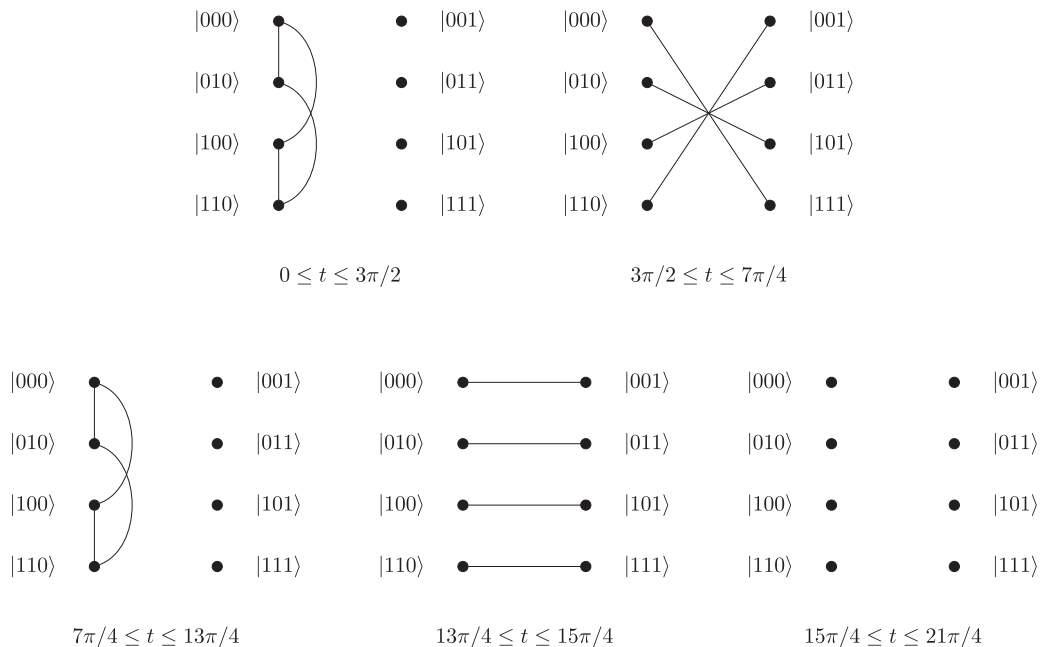


FIG. 8. A graphical representation of the  $H$  gate using CTQW on  $\mathcal{G}_H$ .

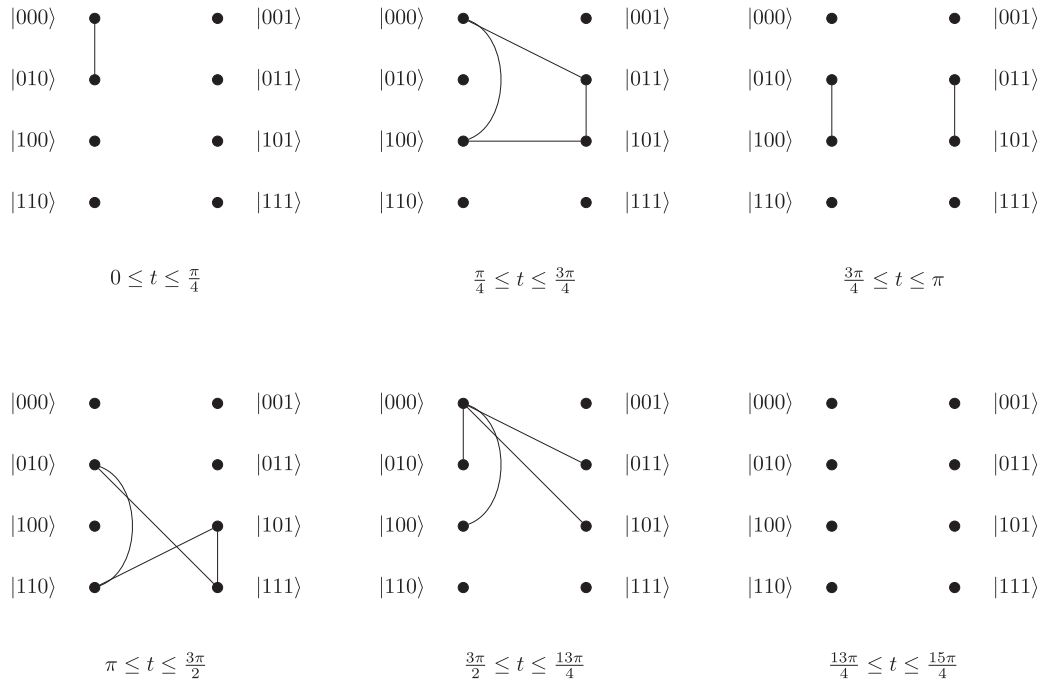


FIG. 9. A graphical representation of the  $T$  gate using  $\mathcal{G}_T$ .

and may be implemented using  $K_1$ ,  $K_2$ , and  $C_4$  graphs, along with the star graph on five vertices. A star graph is a connected graph  $G$  on  $n$  vertices such that exactly one vertex has degree  $n - 1$  and all other vertices have degree one. Figure 9 illustrates the dynamic graph used for the  $T$  gate, which is written as

$$\begin{aligned}
 \mathcal{G}_T = & \left\{ \left( G_{K_2}^{(0,2)} + G_{K_1}^{(1)} + G_{K_1}^{(3)} + G_{K_1}^{(4)} + G_{K_1}^{(5)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, \frac{\pi}{4} \right), \right. \\
 & \left( G_{C_4}^{(0,3,4,5)} + G_{K_1}^{(1)} + G_{K_1}^{(2)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, \frac{\pi}{2} \right), \\
 & \left( G_{K_2}^{(2,4)} + G_{K_2}^{(3,5)} + G_{K_1}^{(0)} + G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, \frac{\pi}{4} \right), \\
 & \left( G_{C_4}^{(2,5,6,7)} + G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_1}^{(3)} + G_{K_1}^{(4)}, \frac{\pi}{2} \right), \\
 & \left( G_{S_5}^{(0,2,3,4,5)} + G_{K_1}^{(1)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, \frac{7\pi}{4} \right), \\
 & \left. \left( G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_1}^{(2)} + G_{K_1}^{(3)} + G_{K_1}^{(4)} \right. \right. \\
 & \left. \left. + G_{K_1}^{(5)} + G_{K_1}^{(6)} + G_{K_1}^{(7)}, \frac{\pi}{4} \right) \right\}. \tag{45}
 \end{aligned}$$

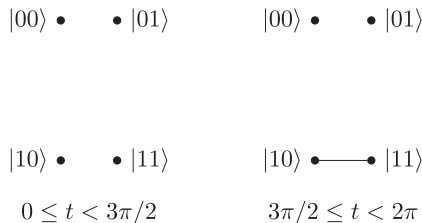


FIG. 10. A graphical representation of the CNOT gate using CTQW on  $\mathcal{G}_{\text{CNOT}}$ .

We show in the Appendix that the CTQW defined by Eq. (45) implements the logical transformation for the  $T$  gate.

### B. Multiqubit gates

Quantum walks on dynamics graphs may also be used to construct multiqubit gates. For example, the two-qubit CNOT gate,

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{46}$$

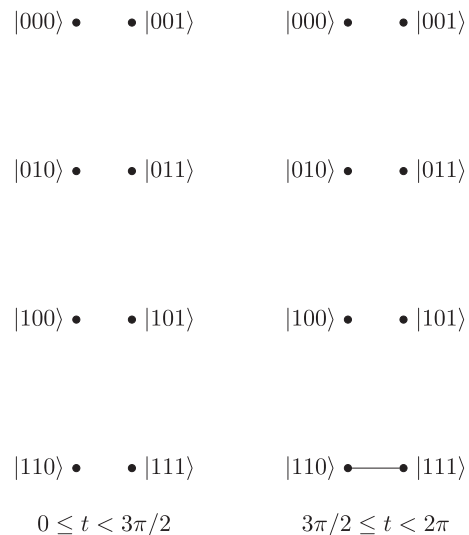


FIG. 11. A graphical representation of the CCNOT (Toffoli) gate using CTQW on  $\mathcal{G}_{\text{CCNOT}}$ .

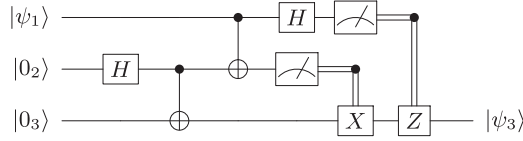


FIG. 12. The circuit model representation of quantum teleportation uses three qubits and a series of elementary gates.

can be realized using a quantum walk on four vertices that span the space of the control and target qubits. Let vertices 0 and 1 propagate as singletons for time  $2\pi$  while allowing vertices 2 and 3 to propagate under  $K_2$  as shown in Fig. 10,

$$\mathcal{G}_{\text{CNOT}} = \left\{ \left( G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_1}^{(2)} + G_{K_1}^{(3)}, \frac{3\pi}{2} \right), \left( G_{K_1}^{(0)} + G_{K_1}^{(1)} + G_{K_2}^{(2,3)}, \frac{\pi}{2} \right) \right\}. \quad (47)$$

The three-qubit CCNOT, or Toffoli, gate is constructed similarly but now using  $|V| = 8$  vertices that represent the two control qubits and one target qubit. The implementation of the Toffoli gate is identical to the CNOT gate but with four additional vertices allowed to propagate as singletons for  $2\pi$  units of time. It is used in both the carry and sum subcircuits in the quantum adder circuit. It is also reversible, meaning its effects may be reversed using other operations. Figure 11 illustrates the dynamic graph for the Toffoli gate.

### C. Measurement and initialization

We model measurement of the quantum state on a graph  $G$  as a projection onto a subspace of the basis  $B_G$ . In establishing a correspondence with the qubit-encoded circuit model, we decompose the labels of the basis according to a binary expansion,

$$|j\rangle = \sum_{i=1}^m j_i 2^{m-i}, \quad (48)$$

with  $j_i \in \{0, 1\}$  and  $m = \log_2 |V|$ . In this binary representation, a quantum state  $|\psi_G\rangle \in B_G$  can be expressed as

$$|\psi_G\rangle = \sum_{j \in V} c_j |j_1, \dots, j_m\rangle, \quad (49)$$

and measuring the  $i$ th qubit to have a fixed value  $\bar{j}_i$  corresponds to projecting the state onto a subset of nodes in the graph, i.e.,

$$|\bar{j}_i\rangle \langle \bar{j}_i | \psi_G\rangle = \sum_{j \in V} c_j |j_1, \dots, \bar{j}_i, \dots, j_m\rangle. \quad (50)$$

The probability to observe node  $j$  is given as

$$\text{Prob}(j) = |\langle \bar{j}_i | \psi_G\rangle|^2 = \sum_{j \in V, j_i = \bar{j}_i} |c_j|^2 \leq 1. \quad (51)$$

We may use measurement as part of a deterministic initialization method, in which the projective outcome is transformed into the desired initial state. This requires conditional operations based on the decoded output from the measurement, from which the necessary series of single-qubit gates are applied to graph. For projections into the label basis, these feed-forward operations consist of products of the Pauli operators flipping the label state to a fiducial starting label, e.g., the vertex 0.

## VI. QUANTUM WALKS FOR QUANTUM CIRCUITS

We complete our analysis by providing explicit examples of how quantum walks on dynamic graphs realize circuits within gate-model computing. These examples highlight the differences in the representation of the logic as well as the resources required to achieve the desired unitary transformations. In our examples, CTQWs are performed in series and the number of vertices needed to implement each circuit is equal to the largest of the number of vertices needed to perform the CTQW equivalent for each logic gate. For the sake of completeness, we also explicitly indicate singleton vertices that add global phase evolution to select vertices in

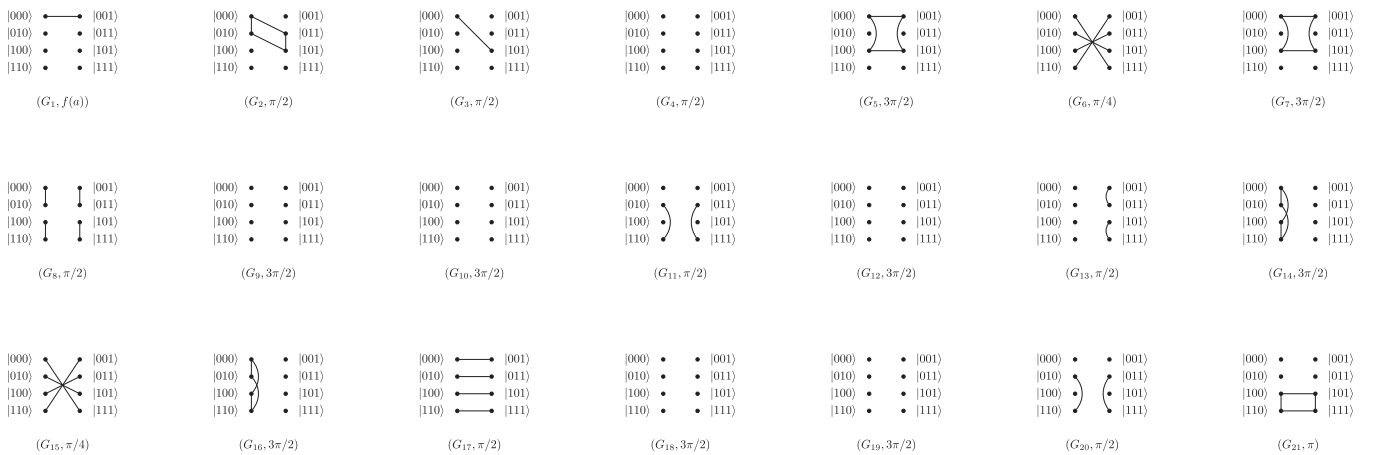


FIG. 13. In this graphical representation of quantum teleportation, each graph is labeled as  $(G_\ell, \tau_\ell)$  with  $\tau_\ell$  the propagation time in the  $\ell$ th graph. The time  $f(a) = \arcsin(\sqrt{a})$  is the state specific time required to rotate  $|000\rangle$  to  $\sqrt{1-a}|000\rangle + \sqrt{a}|001\rangle$ . From left to right, the first four graphs rotate the state while the next five graphs correspond to the  $H$  gate on the second qubit. The following four graphs represent a pair of CNOT gates. The next five graphs correspond with an  $H$  gate on the first qubit. Assuming a measurement outcome  $(b_1 = 1, b_2 = 1)$ , the remaining graphs implement the  $X$  and  $Z$  gates needed to complete teleportation.



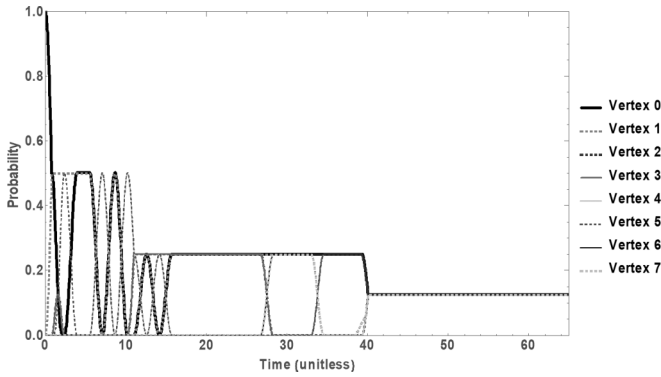


FIG. 14. The population dynamics for state preparation and quantum teleportation using CTQW on the dynamic graph shown in Fig. 13. This example corresponds to the case of measurement outcomes  $b_1 = 1$  and  $b_2 = 1$  for qubits 1 and 2, respectively, and completes the protocol by applying the necessary recovery operations  $X$  and  $Z$ .

order to clearly demonstrate where each gate is used in the implementation. Future optimizations may remove such singleton CTQWs that sum to  $2\pi$  from actual implementations.

### A. Quantum teleportation circuit

In quantum teleportation, a qubit of information is transferred from one logical element to another as shown in Fig. 12. In the circuit model description, three qubits are initially prepared in the state  $|000\rangle$ . The first element is prepared in the state  $|\psi_1\rangle$  by applying the necessary single-qubit transformation. The remaining elements are prepared in a two-qubit entangled state by applying the Hadamard gate to the second element followed by the CNOT gate acting on the second and third elements. A second CNOT gate entangles the first and second qubits. A final Hadamard gate is applied to the first, after which measurements performed on elements 1 and 2 generate binary values  $b_1$  and  $b_2$ , respectively. The effect of these measurements is to project element 3 into the state  $X_3^{b_1} Z_3^{b_2} |\psi_3\rangle$ , which may be transformed to the original state of element 1 with knowledge of  $(b_1, b_2)$ .

The implementation of quantum teleportation using CTQW on a dynamic graph is shown in Fig. 13, and it begins with a graph on eight vertices. Initialization of these vertices is realized through a projective measurement and, depending on the measurement outcome, a sequence of  $X$  operations to populate the 0 vertex. We then approximate an arbitrary unitary operation to prepare the input superposition

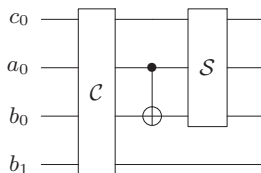


FIG. 15. A quantum circuit for the addition of two one-bit numbers, where the carry circuit  $\mathcal{C}$  and the sum circuit  $\mathcal{S}$  are defined in Figs. 16 and 17. Note that since we only have one carry operation, this is our last carry, and thus is not reversed.

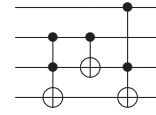


FIG. 16. The carry subcircuit  $\mathcal{C}$  used in Fig. 15.

state  $|\psi\rangle = \sqrt{1-a}|0\rangle + \sqrt{a}|1\rangle$  for  $a \in \mathcal{C}$  where  $|a| = 1$ . The number of vertices needed to represent an arbitrary  $|\psi\rangle$  depends on the desired state, but this single-qubit unitary can be constructed using the universal basis described above. A Hadamard transform is then applied to vertices 0 and 7 using Eq. (43) followed by a pair of CNOT transforms using Eq. (47) acting on vertices  $\{2, 3, 6, 7\}$  and  $\{0, 1, 2, 3\}$ , respectively. The output from this series of CTQWs prepares the graph state,

$$|\psi\rangle = \frac{1}{2}(-\sqrt{1-a}|0\rangle + \sqrt{1-a}|1\rangle + \sqrt{1-a}|2\rangle - \sqrt{1-a}|3\rangle - \sqrt{a}|4\rangle - \sqrt{a}|5\rangle + \sqrt{a}|6\rangle + \sqrt{a}|7\rangle), \quad (52)$$

and a partial projective measurement on the first two bits of the label representation generates the four possible teleported states. The population dynamics of the QTC CTQW are shown in Fig. 14.

### B. Quantum adder

As a second example, we consider a quantum addition circuit for summing two positive integers such that the input  $|a, b\rangle \rightarrow |a, a \oplus b\rangle$  [25]. This variant of in-place addition takes two inputs encoded in registers  $a$  and  $b$  with the binary representations  $a = a_{n-1}a_{n-2}\dots a_1a_0$  and  $b = b_{n-1}b_{n-2}\dots b_1b_0$ . An additional bit  $b_{n+1} = 0$  is added to register  $b$  to give a size  $n+1$ . A third workspace register  $c$  of size  $n-1$  is used in this implementation to store carry values with initialization  $c_i = 0 \forall i$ , while the final carry value is stored in the bit  $b_{n+1}$ . The circuit is composed from two subcircuits for carry and sum operations denoted as  $\mathcal{C}$  and  $\mathcal{S}$ , respectively, and the subcircuits for  $\mathcal{C}$  and  $\mathcal{S}$  are specified in Figs. 16 and 17, respectively. The carry operation uses a Toffoli gate with the second and third qubit as controls and the fourth qubit as the target. This is followed by a CNOT gate on the second and third qubits before another Toffoli gate on the first, second, and fourth qubits. The reverse carry  $\mathcal{RC}$  circuit undoes the carry computation by applying the gates in the reverse order. The last carry bit in the computation is not reversed but stored as  $b_{n+1}$ . The sum subcircuit denoted as  $\mathcal{S}$  in Fig. 17 takes three qubits as input, in which a CNOT is applied to the second and third qubits followed by a Toffoli gate performed with the first two qubits being the controls and the third qubit as the target. In Fig. 15, we show the demonstrated instance of one-bit inputs, i.e.,  $n = 1$ , for which the reverse carry subcircuit is unnecessary. For this example, carry bits are also unnecessary but we include the single carry bit  $c_0$  to confirm generality.

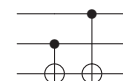


FIG. 17. The sum subcircuit  $\mathcal{S}$  used in Fig. 15.

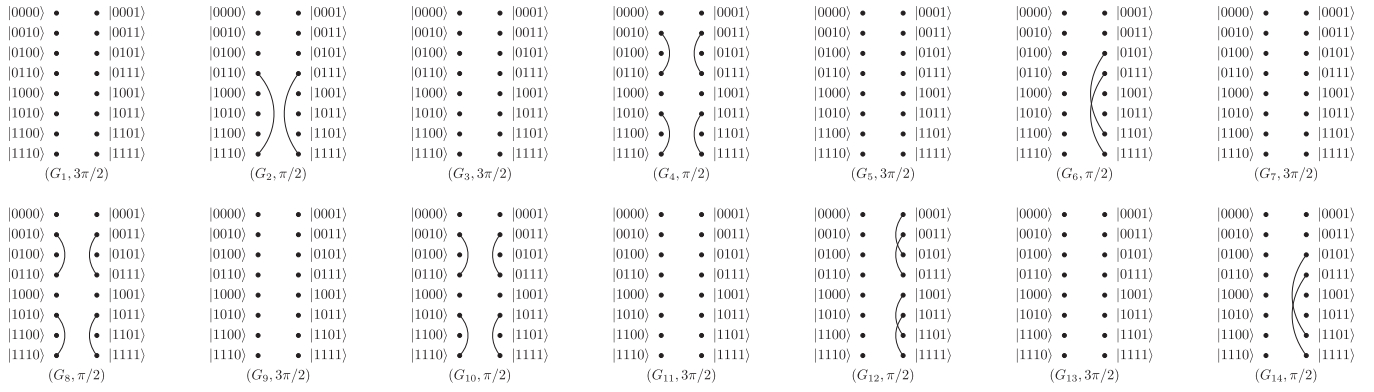


FIG. 18. In this graphical representation of a one-bit quantum adder circuit, each graph is labeled as  $(G_\ell, \tau_\ell)$  with  $\tau_\ell$  the propagation time in the  $\ell$ th graph.

We reduce the gate sequences in the quantum addition circuit into the dynamic graph shown in Fig. 18. Our reduction uses the CTQWs for CNOT and CCNOT gates described in Sec. V and sequentially orders them according to the gate specification in Figs. 15–17. In order to verify the correctness of the reduction, we have used numerical simulation to determine the quantum state generated by the CTQW on the dynamic graph shown in Fig. 18. Numerical simulation of the CTQW requires a memory space that is exponential in the number of qubits, i.e.,  $2^{3n+1}$ . Implementing the quantum adder circuit for  $n = 1$  requires a dynamic graph on 16 vertices.

We show results from a specific simulation with  $|a_0\rangle = |1\rangle$  and  $|b_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  in Fig. 19. We plot the time-dependent population of the vertices that represent the joint state of the computational registers. The carry register is initialized to  $|c_0\rangle = |0\rangle$  and the resulting computational output is  $|b_1, b_0, a_0, c_0\rangle = \frac{1}{\sqrt{2}}(|0, 0, 1, 0\rangle + |1, 1, 1, 0\rangle)$ , where the  $a_0$  and  $c_0$  registers remain in their initial states, and the sum  $a_0 + b_0$  is stored in the  $b_0$  and  $b_1$ . As shown in Fig. 19, our CTQW simulations verify that the dynamic graph yields the expected output states, which corresponds to a uniform superposition of the vertex labels 6 and 10.

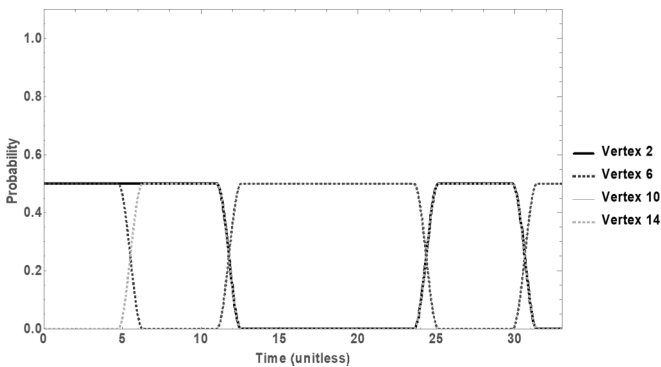


FIG. 19. The population dynamics of the CTQW for quantum addition of inputs  $|a\rangle = |1\rangle$  and  $|b\rangle = \frac{1}{\sqrt{2}}(|0, 0\rangle + \frac{1}{\sqrt{2}}|0, 1\rangle)$ . Numerical simulations of the CTQW on the dynamic graph shown in Fig. 18 calculates exactly the amplitudes of each vertex and the final state is  $|b_1, b_0, a_0, c_0\rangle = \frac{1}{\sqrt{2}}(|0, 1, 1, 0\rangle + |1, 0, 1, 0\rangle)$ , which corresponds to a uniform superposition of the vertices 6 and 10.

## VII. DISCUSSION

Continuous-time quantum walks offer a versatile paradigm for quantum computing, in which the edges between vertices in a graph serve to model the connectivity between basis states. We have defined a dynamic graph as a time-ordered sequence of changing connectivity through which the state of a continuous quantum walk can be tailored to perform computation and, in particular, we have provided constructions of continuous-time quantum walks on dynamic graphs that implement a diverse set of gates taken from the quantum circuit computational model. Our realizations of the single-qubit Pauli, Hadamard, and  $T$  gates, and the CNOT and Toffoli gates, as well as measurement and initialization, form a complete set of primitive operations that can be composed to approximate an arbitrary unitary operator. We were able to implement these gates with a complete basis set of at most eight sparsely connected nodes, allowing the small graph motif to extend across any algorithm. We have presented implementations of the bit-wise addition operation and quantum teleportation to demonstrate composition of quantum walks and shown how some reduction in the composite dynamic graph can be realized by eliminating redundancies.

An important distinction in our formulation of continuous-time quantum walks is the condition that the Hamiltonian represent the connectivity of the underlying basis states. Although we permit variations in this Hamiltonian, the restriction has several side effects on the computational model. For example, our design for some single-qubit gates taken from the circuit model require graphs with more than two vertices. These additional vertices are effectively ancilla used to store temporarily intermediate states of the walk. This unique representation may afford opportunities for optimizing quantum logic by better understanding the transformation of an input state to its output form. Similarly, multiqubit gates such as CNOT and Toffoli are trivial to implement by using the starkly different periods for perfect state transfer. Algorithmic methods that take advantage of these otherwise idle vertices may provide more compact representations of logical transformations.

We have restricted designs of the current quantum walks to small and relatively simple graphs, e.g.,  $K_2$  and  $C_4$ . These designs are appealing because they require less complex in-

teractions between the physical elements, but the ability to realize these designs will depend on technological constraints as well as algorithmic requirements. In particular, perfect state transfer has been implemented recently in a photonic processor [26]. Chapman *et al.* used a linear array of evanescently coupled waveguides to realize nearest-neighbor coupling and transfer the polarized state of one photon to another. The underlying tight-binding Hamiltonian provides an approximation to the connectivity graph underlying a continuous-time quantum walk defined within the space of the single-photon Fock states. The approximation is controlled by the spectra of the coupled waveguides, which must be nonuniform in their geometry for (almost) perfect state transfer using a linear coupled chain [22,23]. The geometrical constraints imposed by linear chains have been overcome by a recent demonstration of continuous-time quantum walks in two-dimensional waveguide arrays [27]. Tang *et al.* demonstrated control of the coupling between waveguide in a two-dimensional array by fabricating specific distance between the channels. We anticipate that these capabilities may be applied to vary the coupling along the waveguide length and, consequently, develop a physical realization of a dynamic graph. These adaptations may require relaxations of our model, including modifying the sharp transitions induced by the rectangle function with more gradual transitions. As a second possible implementation, we note that the Mølmer-Sørensen gate commonly used in ion trap technology enables highly tunable connectivity between multiple qubits for a specific Hamiltonian [28,29], and it would be interesting to apply our principles of continuous-time quantum walks on dynamics graphs to these systems as well.

**ACKNOWLEDGMENTS**

This work was supported in part by the US Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI) as well as the Department of Energy, Office of Science Early Career Research Program and the Mathematical Sciences Graduate Internship program of the National Science Foundation.

This manuscript has been authored in part by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy (DOE). The U.S. government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

**APPENDIX**

We demonstrate that the dynamic graph representing Eq. (43) implements the Hadamard transform by showing explicitly the graph state prepared under the sequence of CTQWs. We first note that the CTQW on each element  $G_\ell$  in a dynamic graph can be evaluated numerically for the

designated propagation time  $t_\ell$ . For  $\mathcal{G}_H$ , we have

$$U_{G_0} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i \end{pmatrix},$$

$$U_{G_1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-i}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{-i}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{-i}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{-i}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{pmatrix},$$

$$U_{G_2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$U_{G_3} = \begin{pmatrix} 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \end{pmatrix},$$

$$U_{G_4} = \begin{pmatrix} i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i \end{pmatrix}.$$

By multiplying the resulting matrices in order, we construct an explicit numerical representation for the CTQW under the dynamic graph  $\mathcal{G}_H$  as

$$U_{\mathcal{G}_H} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}.$$

It is then apparent from this numerical representation that the CTQW for  $\mathcal{G}_H$  is equivalent to applying the circuit-model operator  $H_1 \otimes H_2 \otimes H_3$  on the three-qubit Hilbert space.

We provide a similar proof that the dynamic graph representing Eq. (45) implements the  $T$  gate by showing explicitly the graph state prepared under the sequence of CTQWs. We first note that

$$U_{G_0} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-\frac{i\pi}{4}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-i}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} \end{pmatrix},$$

$$U_{G_1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i \end{pmatrix},$$

$$U_{G_2} = \begin{pmatrix} e^{-\frac{i\pi}{4}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-\frac{i\pi}{4}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} \end{pmatrix},$$

$$U_{G_3} = \begin{pmatrix} -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix},$$

$$U_{G_4} = \begin{pmatrix} 0 & 0 & \frac{i}{2} & \frac{i}{2} & \frac{i}{2} & \frac{i}{2} & 0 & 0 \\ 0 & e^{\frac{i\pi}{4}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{i}{2} & 0 & \frac{3}{4} & \frac{-1}{4} & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 \\ \frac{i}{2} & 0 & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 \\ \frac{i}{2} & 0 & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} & 0 & 0 \\ \frac{i}{2} & 0 & \frac{-1}{4} & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{\frac{i\pi}{4}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{\frac{i\pi}{4}} \end{pmatrix},$$

$$U_{G_5} = \begin{pmatrix} -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i \end{pmatrix}.$$

Thus, as  $\mathcal{G}_T$  is the product of the above matrices, we have

$$U_{\mathcal{G}_T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{\frac{i\pi}{4}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{2} & 0 & \frac{-e^{-\frac{i\pi}{4}}}{\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{-1}{2} & 0 & \frac{e^{-\frac{i\pi}{4}}}{\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{e^{-\frac{i\pi}{4}}}{\sqrt{2}} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{-e^{-\frac{i\pi}{4}}}{\sqrt{2}} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{i\pi}{4}} & 0 \end{pmatrix}.$$

- 
- [1] Y. Aharonov, L. Davidovich, and N. Zagury, *Phys. Rev. A* **48**, 1687 (1993).  
[2] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).  
[3] S. E. Venegas-Andraca, *Quantum Inf. Proc.* **11**, 1015 (2012).  
[4] E. Farhi and S. Gutmann, *Phys. Rev. A* **58**, 915 (1998).  
[5] S. Aaronson and A. Ambainis, in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, 2003), pp. 200–209.  
[6] A. M. Childs and J. Goldstone, *Phys. Rev. A* **70**, 022314 (2004).  
[7] J. K. Gamble, M. Friesen, D. Zhou, R. Joynt, and S. N. Coppersmith, *Phys. Rev. A* **81**, 052313 (2010).  
[8] E. Sánchez-Burillo, J. Duch, J. Gómez-Gardeñes, and D. Zueco, *Sci. Rep.* **2**, 605 (2012).  
[9] P. Philipp, L. Tarrataca, and S. Boettcher, *Phys. Rev. A* **93**, 032305 (2016).  
[10] S. Li and S. Boettcher, *Phys. Rev. A* **95**, 032301 (2017).  
[11] A. M. Childs, *Phys. Rev. Lett.* **102**, 180501 (2009).  
[12] A. M. Childs, D. Gosset, and Z. Webb, *Science* **339**, 791 (2013).  
[13] X. Qiang, T. Loke, A. Montanaro, K. Aungkunsiri, X. Zhou, J. L. O'Brien, J. B. Wang, and J. C. F. Matthews, *Nat. Commun.* **7**, 11511 (2016).  
[14] A. Kay, *Int. J. Quantum Inf.* **8**, 641 (2010).

- [15] V. M. Kendon and C. Tamon, *J. Comput. Theor. Nanosci.* **8**, 422 (2011).
- [16] G. Coutinho and C. Godsil, *Linear and Multilin. Alg.* **64**, 235 (2016).
- [17] R. J. Angeles-Canul, R. M. Norton, M. Opperman, C. C. Paribello, M. C. Russell, and C. Tamon, *Quantum Inf. Comput.* **10**, 325 (2010).
- [18] R. Bachman, E. Fredette, J. Fuller, M. Landry, M. Opperman, C. Tamon, and A. Tollefson, *Quantum Inf. Comput.* **12**, 293 (2012).
- [19] M. S. Underwood and D. L. Feder, *Phys. Rev. A* **82**, 042304 (2010).
- [20] Y.-M. Du, L.-H. Lu, and Y.-Q. Li, *Sci. Rep.* **8**, 7117 (2018).
- [21] S. Chakraborty, L. Novo, S. Di Giorgio, and Y. Omar, *Phys. Rev. Lett.* **119**, 220503 (2017).
- [22] M. Christandl, N. Datta, A. Ekert, and A. J. Landahl, *Phys. Rev. Lett.* **92** 187902 (2004).
- [23] M. Christandl, N. Datta, T. C. Dorlas, A. Ekert, A. Kay, and A. J. Landahl, *Phys. Rev. A* **71**, 032312 (2005).
- [24] C. M. Dawson and M. A. Nielsen, *Quantum Inf. Comput.* **6**, 81 (2006).
- [25] V. Vedral, A. Barenco, and A. Ekert, *Phys. Rev. A* **54**, 147 (1996).
- [26] R. J. Chapman, M. Santandrea, Z. Huang, G. Corrielli, A. Crespi, M.-H. Yung, R. Osellame, and A. Peruzzo, *Nat. Commun.* **7**, 11339 (2016).
- [27] H. Tang *et al.*, *Sci. Adv.* **4**, 3174 (2018).
- [28] A. Sørensen and K. Mølmer, *Phys. Rev. Lett.* **82**, 1971 (1999).
- [29] A. Sørensen and K. Mølmer, *Phys. Rev. A* **62**, 1835 (2000).