

Analysis of the trade-off between spatial and temporal resources for measurement-based quantum computation

Jisho Miyazaki,¹ Michal Hajdušek,^{1,2} and Mio Murao^{1,3}

¹*Department of Physics, Graduate School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan*

²*Singapore University of Technology and Design, 20 Dover Drive, Singapore*

³*Institute for Nano Quantum Information Electronics, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo*

(Received 29 September 2014; published 4 May 2015)

In measurement-based quantum computation (MBQC), elementary quantum operations can be more parallelized than the quantum circuit model by employing a larger Hilbert space of graph states used as the resource. Thus MBQC can be regarded as a method of quantum computation where the temporal resource described by the depth of quantum operations can be reduced compared to the quantum circuit model by using the extra spatial resource described by graph states. To analyze the trade-off relationship of the spatial and temporal resources, we consider a method to obtain quantum circuit decompositions of general unitary transformations represented by MBQC on graph states with a certain underlying geometry called generalized flow. We present a method to translate any MBQC with generalized flow into quantum circuits without extra spatial resource. We also show an explicit way to unravel acausal gates without postselection that appear in the quantum circuit decomposition derived by a translation method [V. Danos and E. Kashefi, *Phys. Rev. A* **74**, 052310 (2006)] and that represents an effect of the reduction of the temporal resource in MBQC. Finally, by considering a way to deterministically simulate these acausal gates, we investigate a general framework to analyze the trade-off between the spatial and temporal resources for quantum computation.

DOI: [10.1103/PhysRevA.91.052302](https://doi.org/10.1103/PhysRevA.91.052302)

PACS number(s): 03.67.Lx, 03.67.Ac, 03.65.Ta

I. INTRODUCTION

Measurement-based quantum computation (MBQC) originally proposed in [1] is a framework for quantum computation in which unitary transformations are implemented by measuring qubits of multipartite entangled states. The multipartite entangled states used as resources in MBQC are characterized by graphs specifying to which pairs of qubits entangling operations have been performed to prepare the state and are called graph states [2,3]. The total number of qubits in the graph state is larger than the number of qubits to which unitary transformations are applied. The graph state extends the “work space” for quantum computation, although the action on the work space is limited to single-qubit operations (measurements). Thus they can be regarded as a *spatial* resource for quantum computation.

In MBQC, the choice of a graph state and measurements, which is referred to as a *measurement pattern*, specifies the implemented unitary transformation. The choice of measurements depends on the outcomes of previous measurements in order to counter nondeterministic state transformations caused by these measurements. Such measurements are called feed-forward measurements. The temporal order of measurements should be carefully chosen to guarantee deterministic implementation of unitary transformations. The quantum depth of a measurement pattern is determined as the minimum number of steps required for preparing a graph state and for performing the feed-forward measurements when any measurements that are not temporally ordered can be performed in a single step. Quantum depth of a quantum circuit which does not include classically controlled operations depending on measurement outcomes is defined as the number of elementary gates included in the longest dependent sequence of gates in that circuit. Thus the depth can be regarded as a *temporal* resource for quantum computation.

For several algorithms, including the approximate quantum Fourier transformation [4], it has been shown that MBQC requires smaller quantum depth than a variation of the quantum circuit model without classically controlled operations depending on measurement outcomes [5,6]. This advantage of MBQC originates from the constant-time implementability of any sequence of Clifford gates [7,8] due to the extended work space by using ancillary qubits of graph states and the feed-forward measurements. In this case, we can see that the spatial resource (ancillary qubits) is used for reducing the temporal resource (the quantum depth).

Flow [9] and generalized flow (or *gflow* for short) [10] are ordering relations on a graph that guarantee the existence of a proper ordering of the measurements required for a deterministic implementation of a unitary transformation by a measurement pattern, irrespective of the choices of measurement angles. If flow or *gflow* exists on a graph, it determines the ordering of measurements, and thus gives an upper bound for the quantum depth of measurement patterns on the graph. These upper bounds are called the depth of flow and *gflow*, respectively. The depth of *gflow* on a graph is lower than the depth of flow on the same graph, since *gflow* is a generalization of flow. There are graphs which have *gflow* but do not have flow for the same reason.

Given a quantum circuit decomposition of a unitary transformation, we can construct a measurement pattern with depth of flow equal to or less than the quantum depth of the original circuit [9]. This implies that the depth of flow, and so the depth of *gflow*, already takes into account the constant-time implementability of Clifford gates.

In order to study how the depth of flow and *gflow* are related to the constant-time implementability of Clifford gates, it would be helpful to construct a method to write a circuit decomposition with no ancillary qubits representing the same unitary transformation implemented by a measurement pattern

with flow or gflow. Since this compact circuit decomposition does not utilize extra work space, it includes sequences of Clifford gates that contribute to the increase of quantum depth of the circuit but not to the depth of flow and gflow. Thus translations of a unitary represented by MBQC to that of a quantum circuit provide a clue for understanding the trade-off relation between the spatial and temporal resources.

To date, there are three methods to translate a measurement pattern into a compact quantum circuit proposed by [9,11,12]. In [9], a translation method called the star-pattern transformation (SPT) applicable to measurement patterns on the graph states with flow is presented. If we ignore the depth for implementing Clifford gates, the depth of the resulting circuit coincides with the depth of flow of the original measurement pattern. If we use the SPT to convert a measurement pattern on a graph with gflow but without flow into a quantum circuit, the translation fails and we cannot avoid obtaining an *acausal* circuit with ill-defined two-qubit gates simultaneously acting in two different steps of time. In [11] and [12], the authors investigated translation methods applicable also for graphs with gflow but no flow. The method proposed in [11] based on category theory translates any measurement pattern with gflow into compact circuits and is applicable to a more general class of measurement patterns with no gflow. The depth of resulting circuits, however, is not analyzed and does not necessarily coincide with the depth of gflow, even if the depth for implementing sequences of Clifford gates is assumed to be constant on the circuit.

If acausal gates are allowed to be used in the quantum circuit model, its computational power can be greatly enhanced [13,14]. The authors of [10] suggest that the acausal circuits obtained by applying the SPT for measurement patterns of MBQC may efficiently implement the unitary transformation represented by the original measurement pattern. We further expect that from a viewpoint of the trade-off between spatial and temporal resources of computation, a measurement pattern with gflow reduces the quantum depth by deterministically simulating acausal gates by utilizing the extra work space.

In this paper, we propose a method to translate a measurement pattern on a graph state with gflow into a compact quantum circuit by generalizing the SPT. Based on this translation method, we clarify the relation between the depth of gflow and constant-time implementability of Clifford gates. We investigate the properties of graphs with gflow and the entanglement structure of the graph states defined by these graphs, and construct the translation method. We show the existence of path covers on graphs with gflow, which was previously shown only for graphs with flow [15]. Local unitary transformations on certain sets of qubits are used for simplifying the entanglement structure of the graph state.

We also show a relation between the circuit decomposition obtained by our method and the acausal circuits obtained by directly applying the SPT for measurement patterns on graph states with gflow but no flow. An operation represented by an acausal two-qubit gate simultaneously acting on two different time positions of the acausal circuit is defined to be consistent with a unitary transformation implemented by the measurement pattern. Finally, we discuss how MBQC compresses the quantum depth in connection with the acausal circuit representation.

This paper is organized as follows. In Sec. II, we review MBQC and the properties of a graph corresponding to a graph state used as a resource for deterministic MBQC. We also reformulate the SPT on graphs with flow. In Sec. III, we show several graph-theoretical properties of gflow. In Sec. IV, we present the translation method from a measurement pattern to a quantum circuit using a transformation of a graph with gflow to a graph with flow. In Sec. V, the quantum circuit obtained by the method in the previous section is further transformed to parallelize non-Clifford gates. In Sec. VI, we introduce the SPT for graphs with gflow and formally define an acausal circuit. In Sec. VII, we present another translation method from a measurement pattern with gflow to a quantum circuit via an acausal circuit. In Sec. VIII, we discuss the relation between the acausal circuit and the compression of quantum depth.

II. PRELIMINARIES

A. Graph and graph states

For a given graph $G = (V, E)$ with the vertex set V and the edge set $E \subset \{\{u, v\} | u, v \in V, u \neq v\}$, we choose a set of input vertices $I \subset V$ and a set of output vertices $O \subset V$, corresponding to the qubits used to encode the input state and decode the output state, respectively. The triplet (G, I, O) is called an open graph.

If U is a subset of V , U^C represents the complement of U . We say vertices u and v are connected if $\{u, v\} \in E$ and denote it by $u \sim v$ ($u \sim_G v$, for specifying a graph G). A neighborhood of vertex v on G is a set of vertices that are connected to v on G , and is denoted by $N_G(v)$. $\text{Odd}_G(V_0)$ [$\text{Even}_G(V_0)$] represents the odd (even) neighborhood of $V_0 \subset V$ on G , i.e., the set of vertices that are connected to the odd (even) number of vertices in V_0 . For example, $\text{Odd}_G(\{u\})$ is just the neighborhood of vertex u . $V_0 \oplus V_1$ represents the symmetric difference between vertex sets V_0 and V_1 defined by $V_0 \oplus V_1 = (V_0 \cup V_1) \setminus (V_0 \cap V_1)$. Note that $\text{Odd}(V_0 \oplus V_1) = \text{Odd}(V_0) \oplus \text{Odd}(V_1)$ [16]. Vertex sets V_0 and V_1 are said to be linearly independent in a vertex set V_2 , when $(V_0 \oplus V_1) \cap V_2 \neq \emptyset$. Similarly, a set of vertex sets $\{V_n\}_{n \in \Lambda}$ is said to be a basis of V' if $|\Lambda| = |V'|$ and for any subset $\Gamma \subset \Lambda$, $\bigoplus_{n \in \Gamma} V_n \cap V' \neq \emptyset$. These relations are understood as a linear independence in the vertex space, where the symmetric difference corresponds to \mathbb{F}_2 addition [16].

The Pauli matrices are denoted by capital letters X, Y , and Z in this paper. The eigenstates of Z (the computational basis) corresponding to eigenvalues 1 and -1 are represented by $|0\rangle$ and $|1\rangle$, respectively. The eigenstates of X corresponding to eigenvalues 1 and -1 are represented by $|+\rangle$ and $|-\rangle$, respectively. We define a general controlled-unitary (CU) transformation on a set of qubits specified by a set of indices S' where a qubit specified by index $v \in S'$ is a controlled qubit and a unitary transformation U is applied on the rest of qubits specified by a set $\mathcal{S} := S' \setminus v$ only when the controlled qubit is in $|1\rangle$, otherwise no transformation is applied, namely,

$$\text{CU}_{v;\mathcal{S}} := |0\rangle\langle 0|_v \otimes \mathbb{I} + |1\rangle\langle 1|_v \otimes U. \quad (1)$$

If $\mathcal{S} = \{u\}$, $\text{CU}_{v;\mathcal{S}}$ is also represented by $\text{CU}_{v;u}$. In particular, $\text{CZ}_{v;u}$ and $\text{CX}_{v;u}$ represent the controlled- Z (CZ) gate and controlled-NOT (CNOT) gate, respectively.

A quantum state corresponding to an open graph is called an open-graph state and is constructed in the following way. First we prepare a qubit system on each vertex of the graph G . Each qubit is labeled by the index of the corresponding vertex. All qubits with indices in I^C are prepared in the $|+\rangle$ state, whereas the qubits with the indices in I are prepared in a joint input state $|\phi\rangle$. Next, CZ gates are applied to all pairs of qubits corresponding to adjacent vertices, namely, the qubits with indices connected by edges E of G . We denote a unitary transformation U acting on qubits of the graph states by \tilde{U} in order to distinguish it from a unitary transformation acting on logical qubits of the corresponding circuit. Then an open-graph state $|G\rangle_\phi$ of an open graph G with an input state $|\phi\rangle_I$ is represented by

$$|G\rangle_\phi = \tilde{E}_G |\phi\rangle_I |+\rangle_{I^C},$$

where

$$\tilde{E}_G = \prod_{\{u,v\} \in E} \tilde{CZ}_{u,v}.$$

This state is stabilized by $\tilde{K}_v := \tilde{X}_v \tilde{Z}_{N(v)}$ ($v \in I^C$), namely, $\tilde{K}_v |G\rangle_\phi = |G\rangle_\phi$.

B. Flow

After preparing the open-graph state, the unitary transformation is implemented by performing projective measurements on each qubit in O^C . The measurement operators are defined by $\{|\pm_{\alpha_v}\rangle\langle\pm_{\alpha_v}|\}$, where

$$|\pm_{\alpha_v}\rangle := (|0\rangle \pm e^{i\alpha_v}|1\rangle)/\sqrt{2},$$

and $\alpha_v \in [0, 2\pi)$ represents the measurement angle depending on the vertex $v \in O^C$. If we obtain a measurement result “−” on a qubit, we adjust the measurement angles of future measurements so that the quantum computation proceeds as if we had obtained the result “+”. Therefore, the unitary transformation implemented by a deterministic MBQC on an open graph (G, I, O) is proportional to

$$\bigotimes_{u \in O^C} \langle +_{\alpha_u} | \tilde{E}_G | + \rangle_{I^C}. \quad (2)$$

The dependency relation of the measurement angles determines the ordering of measurements. Flow [9] is an ordering relation guaranteeing deterministic computation, and is a pair $(f, <)$ of a function $f : O^C \rightarrow I^C$ and a partial order $<$ satisfying the following conditions:

- (i) [f-1] $u < f(u)$
- (ii) [f-2] $u \in N[f(u)]$
- (iii) [f-3] $\forall v \in N[f(u)], u = v$ or $u < v$.

The vertex $f(u)$ is referred to as a “correcting vertex” [10], since measurement angles of vertices in $f(u) \cup N[f(u)] \setminus u$ are corrected according to the result of a measurement on u in the MBQC with flow. This is why there must be an ordering from u to $f(u) \cup N[f(u)] \setminus u$.

Graph-theoretical properties of flow are analyzed in [15]. A path cover is an important property of a graph with flow for understanding the correspondence with the circuit model. It is defined by the following.

Definition 1. Path cover [15]: Let (G, I, O) be an open graph. A collection P_f of (possibly trivial) directed paths in G is a path cover of (G, I, O) if

- (i) each $v \in V(G)$ is contained in exactly one path (i.e., the paths cover G and they are vertex disjoint);
- (ii) each path in P_f is either disjoint from I or intersects I only at its initial point;
- (iii) each path in P_f intersects O only at its final point.

For any open graph with flow, a unique path cover P_f is defined by

$$v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n \in P_f \Leftrightarrow v_n \in O \wedge f(v_i) = v_{i+1} (\forall i). \quad (3)$$

Note that this definition of a path cover is more restrictive compared to the notion commonly used in graph theory where a path cover is a set of disjoint paths on a directed graph and does not necessarily connect vertices in I and O [16].

C. Circuits and measurement patterns with flow

A *measurement pattern* consists of the open graph (G, I, O) , the ordering of measurements, and the measurement angles $\alpha_v (v \in O^C)$, which may depend on the outcomes of previous measurements. The measurement pattern determines how to prepare the qubits, how to entangle them, and how to perform measurements.

Star-pattern transformation (SPT) is a method [9] (see also Ref. [17] in this context) to translate a unitary transformation implemented by a measurement pattern with flow to a circuit decomposition, in such a way that each measurement in the measurement pattern corresponds to an elementary gate in the circuit. In this section, we reformulate this method to be easily extendible for measurement patterns with gflow.

The procedure of the SPT is divided into three parts:

- (i) We regard each path in P_f as a wire that represents a Hilbert space of a qubit \mathbb{C}^2 in the circuit.

The wire in the circuit corresponding to the path including vertex v on the graph is also labeled by v . A wire labeled by a flow image $f(v)$ of a vertex v is identical to the wire labeled by v .

- (ii) We place a J gate, $J(\alpha_v)$, defined by

$$J(\alpha_v) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{-i\alpha_v} \\ 1 & -e^{-i\alpha_v} \end{pmatrix},$$

on wire v if qubit v is measured at an angle α_v .

These J gates must be placed so that $J(\alpha_u)$ acts before $J(\alpha_v)$ if $u < v$ on the graph. We sometimes have to specify not only wires but also the position in the wire on which a gate acts. This position indicates the timing when the gate acts on the qubit represented by the wire. The position between $J(\alpha_{f^{-1}(v)})$ and $J(\alpha_v)$ in the wire v is labeled by $v!$. If v is a starting vertex of a path cover, the label $v!$ represents the position before the gate $J(\alpha_v)$ in the wire v , (see Fig. 1).

- (iii) If there is an edge between vertices u and v , we place $CZ_{u!,v!}$. The ordering of multiple CZ gates corresponding to nonpath edges incident from the same vertex is determined by the partial ordering of the vertices on the other side of the edges corresponding to the CZ gates.

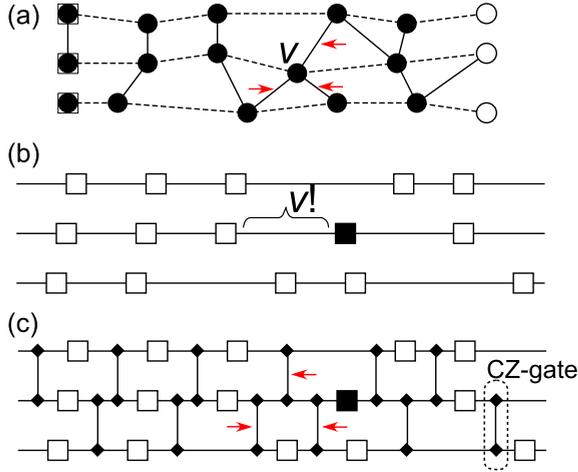


FIG. 1. (Color online) (a) An open graph with flow. Circles and lines represent the vertices and edges, respectively. The boxed vertices represent inputs and the white vertices are outputs by following the notation of MBQC presented in [9]. The set of dashed edges represents the path cover, and the vertex incident with the edges pointed by arrows is labeled by v . (b) A quantum circuit corresponding to the open graph given by (a) obtained by process (ii) of SPT (Sec. II C). The three wires correspond to the path cover, and boxes represents the J gates. The black box represents a particular J gate, $J(\alpha_v)$, assigned for the vertex labeled by v . The position $v!$ denotes a region on the wire between $J(\alpha_v)$ and $J(\alpha_{f^{-1}(v)})$. (c) A quantum circuit representing the measurement pattern on the open graph given by (a). The CZ gates pointed by arrows correspond to the edges pointed by arrows on the graph.

We define a binary relation \prec_p on the positions in the circuit by $u! \prec_p v!$ if and only if

- (a) $\exists p \in P_f$ such that $(u \rightarrow v) \in p$, or
- (b) $\exists p \in P_f, \exists w \in N(v) \setminus \{u\}$ such that $(u \rightarrow w) \in p$, or
- (c) $\exists w \in V$ such that $u! \prec_p w!$ and $w! \prec_p v!$.

This binary relation can be defined on any circuit corresponding to a measurement pattern on a graph with a path cover. The relation \prec_p must be a partial order to have a consistent gate sequence. If the path cover P_f is defined by flow according to Eq. (3), then $u! \prec_p v! \Leftrightarrow u < v$ holds. This is shown by Lemma 9 and Theorem 10 of Ref. [15] (note that $u! \prec_p v!$ if and only if there is an “influencing walk” from u to v defined in Definition 7 of Ref. [15]). There is a consistent gate sequence on any circuit corresponding to a measurement pattern on a graph with flow because $<$ is a partial order. Conversely, if an open graph with a path cover does not have flow, the binary relation \prec_p on the corresponding circuit is not a partial order (again from Lemma 9 and Theorem 10 of Ref. [15]), and the gate sequence is not well defined as a quantum circuit.

By reversing this method, we obtain a measurement-pattern representation of a unitary transformation from a circuit representation whose elementary gates are given by J gates and CZ gates [17].

D. gflow

Gflow is defined as follows:

Definition 2. (See [10], Definition 3.) Let (G, I, O) be an open graph. Let $g : O^c \rightarrow 2^{I^c}$ be a function on nonoutput vertices to the power set of noninput vertices, and let $<$ be a

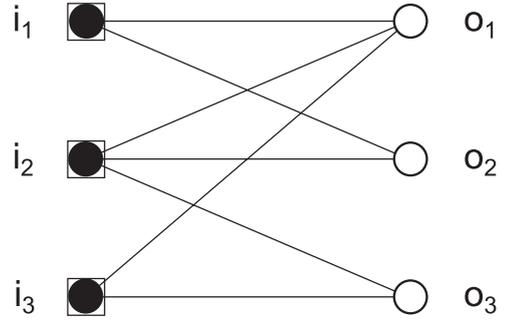


FIG. 2. An open graph with gflow but no flow. The input and output vertices are $I = \{i_1, i_2, i_3\}$ and $O = \{o_1, o_2, o_3\}$, respectively. The last layer $V_{<}^0$ with regard to the maximally delayed gflow is O , and the second-to-last layer $V_{<}^1$ is I . The values of the maximally delayed gflow function g are $g(i_1) = \{o_1, o_3\}$, $g(i_2) = \{o_1, o_2, o_3\}$, $g(o_3) = \{o_1, o_2\}$. The modified gflow (g_v, \prec_v) that satisfies Eqs. (11) and (12) is $g_v(i_1) = g(i_1) = \{o_1, o_3\}$, $g_v(i_2) = g(i_2) \oplus g(o_3) = \{o_2\}$, $g_v(i_3) = g(i_3) \oplus g(o_3) = \{o_3\}$.

strict partial order on vertices. The pair (g, \prec) is a gflow of the open graph if it satisfies the following three conditions:

- (i) [g-1] $\forall v \in g(u), u < v$.
- (ii) [g-2] $u \in \text{Odd}[g(u)]$.
- (iii) [g-3] $\forall v \in \text{Odd}[g(u)], u = v$ or $u < v$.

Gflow is a generalization of flow in the sense that g can take a set of vertices, whereas the flow function f can take only one vertex. The set $g(u)$ is referred to as a “correcting set” [10] for u , since measurement angles of vertices in $g(u) \cup \text{Odd}[g(u)] \setminus u$ are corrected according to the result of a measurement on u in the measurement pattern with gflow. There are graphs that have gflow but do not have flow. In this case, the SPT does not lead to a well-defined circuit. For example, an open graph presented in Fig. 2 has a path cover, so we can define wires of the circuit for this graph. However, we cannot assign all CZ gates in a way obeying a well-defined ordering of gates [Fig. 15(a)].

The strict partial order of gflow induces a temporal ordering that the sequence of measurements and corrections must follow. Vertices that do not have ordering between each other are said to be in the same layer.

Definition 3. Layers [18]: Let (G, I, O) be an open graph with gflow (g, \prec) . Layers $V_{<}^k$ of this gflow are defined as

$$V_{<}^0 \equiv \max_{<} V(G),$$

$$V_{<}^k \equiv \max_{<} V(G) \setminus \cup_{i < k} V_{<}^i \quad (k > 0),$$

where the maximization in terms of the relation $<$ is defined by $\max_{<} X \equiv \{u \in X \mid \forall v \in X, \neg(u < v)\}$.

When there are $d_{<} + 1$ layers $V_{<}^0, \dots, V_{<}^{d_{<}}$, $d_{<}$ is called *depth of the gflow*. Measurements of qubits corresponding to the vertices belonging to the same layer can be performed simultaneously. The depth of the gflow represents the number of rounds of simultaneous measurements required according to the gflow.

Definition 4. Delay: A gflow (g, \prec) is more delayed than (g', \prec') if and only if

$$\forall k, \quad |\cup_{i=0}^k V_{<}^i| \geq |\cup_{i=0}^k V_{<}^i|, \quad (4)$$

and there is a number specified by k with which the inequality (4) becomes strict.

In general, gflow is not unique and neither is the depth of gflow. The gflow with minimal depth on an open graph is called maximally delayed gflow.

Maximally delayed gflow has the following properties:

$$\begin{aligned} V_{\prec}^0 &= O, \\ V_{\prec}^1 &= \{v \in O^C \mid \exists S_v \subset O, \text{Odd}(S_v) \cap O^C = \{v\}\}. \end{aligned}$$

These properties are used extensively in our analysis.

III. PATH COVERS FOR GFLOW

In this section, we show the existence of a path cover on graphs with gflow. Paths of the path cover will be regarded as the wires in the circuit decomposition similarly to the cases of graphs with flow presented in Sec. II C. We prove the existence of the path cover by construction. The first step is to find a matching between the output and the penultimate layer. The following lemmas, which are similar to the lemmas presented in Sec. 2.3.2 of Ref. [19], are used for the proof.

Lemma 1. Let (G, I, O) be an open graph with gflow, with its layers $\{V_{\prec}^i\}_{i=0, \dots, d}$ ($V_{\prec}^0 = O$) defined by maximally delayed gflow (g, \prec) . For all subsets $V \subset V_{\prec}^1$, there is a subset $R_V \subset O$ and a gflow (g_V, \prec_V) satisfying the following four conditions:

- (i) [R-a] $|R_V| = |V|$.
- (ii) [R-b] $\{R_V \cap g(v)\}_{v \in V}$ becomes a basis of R_V .
- (iii) [R-c] There is a perfect matching between R_V and V , and the edges of the matching are chosen from real gflow edges of (g_V, \prec_V) , where a real gflow edge refers to the edge $(x, y) \in E$ satisfying $y \in g(x)$ (Fig. 3).
- (iv) [R-d] $\text{Odd}[g_V(v)] \cap (V_{\prec}^1 \setminus V) = \emptyset$ ($\forall v \in V$).

Proof. The proof is by induction with respect to $|V|$. The statement holds for the case of $|V| = |\{v\}| = 1$, if we choose $(g_V, \prec_V) = (g, \prec)$ and R_V to be any vertex $\{r_1\}$ in $g(v)$. We assume that there exists a subset R_V that satisfies conditions [R-a] to [R-d] for any V with $|V| \leq m$. Let $V_n = \{v_1, v_2, \dots, v_n\}$ ($\forall n \leq m+1$). From the assumption, there is a subset $R_{V_m} \subset O$ that satisfies conditions [R-a] to [R-d]. We denote the gflow described in [R-c] and [R-d] by (g_{V_m}, \prec_{V_m}) . From [R-b], there exists a vertex set $U_{m+1} \subset V_m$ such that

$$R_{V_m} \cap g(v_{m+1}) = R_{V_m} \cap \bigoplus_{v \in U_{m+1}} g(v). \quad (5)$$

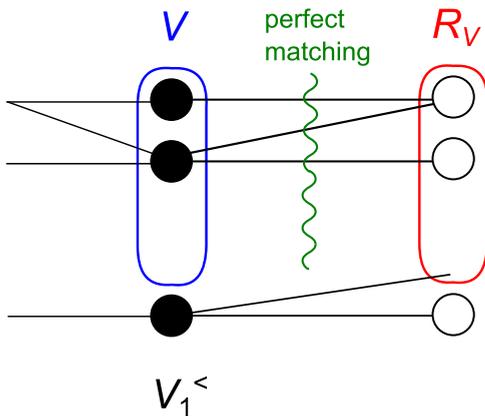


FIG. 3. (Color online) Perfect matching between $V \subset V_{\prec}^1$ and $R_V \subset O$ (Lemma 1).

Let us define a subset O_{m+1} of O by

$$O_{m+1} := g(v_{m+1}) \oplus \bigoplus_{v \in U_{m+1}} g(v). \quad (6)$$

By definition, $O_{m+1} \cap R_{V_m} = \emptyset$. We define a function $g_{V_{m+1}} : O^C \rightarrow 2^{I^C}$ by

$$g_{V_{m+1}}(v) := \begin{cases} O_{m+1} & \text{for } v = v_{m+1} \\ g_{V_m}(v) & \text{otherwise.} \end{cases} \quad (7)$$

For any $l \leq m+1$,

$$\begin{aligned} \text{Odd}[g_{V_{m+1}}(v_l)] \setminus O &= \text{Odd}(O_l) \setminus O \\ &= \{\text{Odd}[g(v_l)] \setminus O\} \oplus \bigoplus_{v \in U_l} \text{Odd}[g(v)] \setminus O \\ &= \{v_l\} \oplus \bigoplus_{v \in U_l} \{v\} \\ &= \{v_l\} \cup U_l \end{aligned}$$

holds. For the case $n = m+1$, $v_{m+1} \in \text{Odd}[g_{V_{m+1}}(v_{m+1})]$ implies the existence of an odd number of edges between v_{m+1} and $g_{V_{m+1}}(v_{m+1})$. We choose a vertex $r_{m+1} (\sim v_{m+1})$ from $g_{V_{m+1}}(v_{m+1})$ for the matching with v_{m+1} . This vertex is not included in R_{V_m} because $O_{m+1} \cap R_{V_m} = \emptyset$. For later use, we define a function h on the vertices inductively by

$$h(v_{m+1}) = r_{m+1}. \quad (8)$$

The domain of this function becomes O^C after the induction is finished.

Now we have to define a partial ordering $\prec_{V_{m+1}}$ for function $g_{V_{m+1}}$. Because $\text{Odd}[g_{V_{m+1}}(v_{m+1})] \setminus O = v_{m+1} \cup U_{m+1}$, $v_{m+1} \prec_{V_{m+1}} U_{m+1}$ must hold. This is allowed if $v_{m+1} \notin \text{Odd}[g_{V_m}(v)]$ for any vertex $v \in U_{m+1}$, which is guaranteed by the assumption [R-d]. Thus, $g_{V_{m+1}}$ and the ordering inductively defined by

$$v \prec_{V_{m+1}} u \Leftrightarrow (v = v_{m+1} \wedge u \in U_{m+1} \cup O) \vee (v \prec_{V_m} u) \quad (9)$$

is a gflow on the open graph (G, I, O) .

Now we define

$$R_{V_{m+1}} := R_{V_m} \cup \{r_{m+1}\}. \quad (10)$$

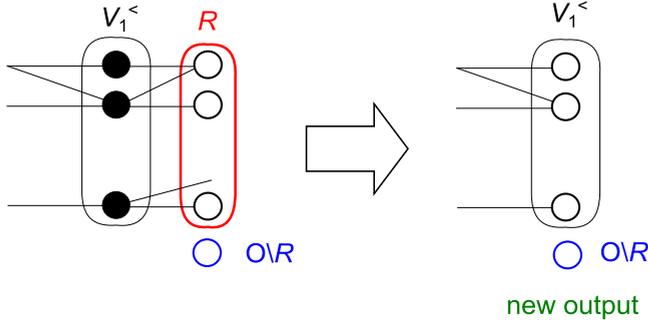
The gflow $(g_{V_{m+1}}, \prec_{V_{m+1}})$ and $R_{V_{m+1}}$ satisfy [R-c] because v_{m+1} is connected to $r_{m+1} \in O_{m+1} \setminus R_{V_m} \subset O \setminus R_{V_m}$ and $r_{m+1} \in O_{m+1} = g_{V_{m+1}}(v_{m+1})$. The condition [R-d] follows from the definition of $(g_{V_{m+1}}, \prec_{V_{m+1}})$.

It remains to show [R-b] since [R-a] is trivial. By assumption, $\{R_{V_m} \cap g(v)\}_{v \in V_m}$ is the basis of R_{V_m} . This implies that a union of $\{R_{V_{m+1}} \cap g(v)\}_{v \in V_m}$ and $\{r_{m+1}\} = R_{V_{m+1}} \cap g_{V_{m+1}}(v_{m+1})$ forms the basis of $R_{V_{m+1}}$. Then from the definition of $g_{V_{m+1}}(v_{m+1})$, $\{R_{V_{m+1}} \cap g(v)\}_{v \in V_{m+1}}$ becomes the basis of $R_{V_{m+1}}$.

Thus the statement of the lemma holds for $|V| = m+1$, which concludes the proof.

We give a particular name to the gflow constructed by this lemma for convenience.

Definition 5. Matching gflow: If $V = V_{\prec}^1 = \{v_1, v_2, \dots, v_n\}$ ($n = |V_{\prec}^1|$), we call the gflow (g_{V_n}, \prec_{V_n}) , which is constructed inductively by Eqs. (5)–(7), (9), and (10), a matching gflow of


 FIG. 4. (Color online) Removing R from O (Lemma 2).

(G, I, O) , and denote it by (g_V, \prec_V) . We call the function $h : V_{\prec}^1 \rightarrow O$ defined by Eq. (8) a successor function of (g_V, \prec_V) .

Lemma 1 guarantees the existence of a matching between V_{\prec}^1 and a suitable subset R_V of O . The following lemma helps to find the matching between other layers by reducing R_V .

Lemma 2. Let (G, I, O) be an open graph with gflow, with its layers $\{V_{\prec}^i\}_{i=0, \dots, d}$ ($V_{\prec}^0 = O$) defined by maximally delayed gflow (g, \prec) . If the subset $R \subset O$ satisfies [R-a] and [R-b] of Lemma 1 with $V = V_{\prec}^1$, then an open graph $(G \setminus R, I \setminus R, V_{\prec}^1 \cup [O \setminus R])$ has maximally delayed gflow with the same ordering \prec .

Proof. We define an reduced open graph $(G', I', O') = (G \setminus R, I \setminus R, V_{\prec}^1 \cup [O \setminus R])$ by reducing R from the original open graph (G, I, O) as presented in Fig. 4, and construct a gflow on this open graph. Because $\{R \cap g(v)\}_{v \in V_{\prec}^1}$ is a basis of R , for all $v \in V(G') \setminus O'$, there is a subset $V_v \subset V_{\prec}^1$ such that

$$g(v) \cap R = \bigoplus_{u \in V_v} g(u) \cap R.$$

$$\begin{aligned} V_{\prec_m}^1 &= \{v \in G' \setminus O' \mid \exists S_v \subset O', \text{Odd}_{G'}(S_v) \setminus O' = \{v\}\} \\ &= \{v \in G \setminus (O \cup V_{\prec}^1) \mid \exists S_v \subset (O \cup V_{\prec}^1) \setminus R, \text{Odd}_G(S_v) \setminus (O \cup V_{\prec}^1) = \{v\}\} \\ &\subset \{v \in G \setminus (O \cup V_{\prec}^1) \mid \exists S_v \subset (O \cup V_{\prec}^1), \text{Odd}_G(S_v) \setminus (O \cup V_{\prec}^1) = \{v\}\} \\ &= V_{\prec}^2. \end{aligned}$$

Since V_{\prec}^2 is the first layer in (G', I', O') with respect to (g', \prec) , $|V_{\prec}^2| \leq |V_{\prec_m}^1|$ holds. Thus we have $V_{\prec}^2 = V_{\prec_m}^1$, and we can show the equality of all layers (i.e., identity of \prec_m and \prec) by induction.

If we consecutively apply Lemma 2 with a subset to be removed chosen according to Lemma 1 (by taking $V = V_{\prec}^1$), the resulting set of edges used for the matchings forms a path cover.

Theorem 1. If an open graph (G, I, O) has gflow, then there exists a path cover. Each edge of the paths can be chosen from real gflow edges of some gflow on (G, I, O) .

The path cover defined in this way is not necessarily unique. There is an arbitrariness in choosing the subset R and the matching between R and V_{\prec}^1 .

We note that Lemma 2 solely implies that $|V_{\prec}^k| \leq |O|$ for all k , since the number of output vertices is not changed by the use of Lemma 2 (i.e., $|O'| = |V_{\prec}^1 \cup (O \setminus R)| = |O|$). This result implies that the depth of gflow is upper bounded by $|V|/|O|$.

We define another gflow function $g' : O' \rightarrow 2^{I'}$ by

$$g'(v) := g(v) \setminus R \oplus \bigoplus_{u \in V_v} g(u) \setminus R.$$

It can be checked that $v \prec w [\forall w \in g'(v)]$ holds for the original partial order of gflow (g, \prec) , due to $\bigoplus_{u \in V_v} g(u) \subset O$. The following calculation shows that $v \prec w [\forall w \in \text{Odd}[g'(v)]]$ also holds:

$$\begin{aligned} &\text{Odd}_{G'}[g'(v)] \\ &= \text{Odd}_G \left[g(v) \setminus R \oplus \bigoplus_{u \in V_v} g(u) \setminus R \right] \setminus R \\ &= \text{Odd}_G \left(g(v) \setminus R \oplus \bigoplus_{u \in V_v} [g(u) \cap R] \oplus g(u) \right) \setminus R \\ &= \text{Odd}_G \left\{ g(v) \setminus R \oplus [g(u) \cap R] \oplus \bigoplus_{u \in V_v} g(u) \right\} \setminus R \\ &= \text{Odd}_G \left[g(v) \oplus \bigoplus_{u \in V_v} g(u) \right] \setminus R \\ &= \left\{ \text{Odd}_G[g(v)] \oplus \text{Odd}_G \left[\bigoplus_{u \in V_v} g(u) \right] \right\} \setminus R \\ &= v \oplus V_v \oplus (\text{subset of } O) \setminus R \\ &= v \oplus (\text{subset of } O'). \end{aligned}$$

It remains to show that (g', \prec) is maximally delayed in G' . Let (g_m, \prec_m) be the maximally delayed gflow of (G', I', O') ; then

IV. TRANSLATION FROM MBQC INTO QUANTUM CIRCUIT

The path cover we have constructed in Sec. III is used as a wire of the corresponding circuit decomposition. We divide the unitary transformation represented by a measurement pattern into a step-by-step unitary transformation implemented between each of the layers $V_{\prec}^i \rightarrow V_{\prec}^{i-1}$. For simplicity of notation, we define the following two multiqubit gates:

$$\text{CZ}_{v,S} := \prod_{u \in S} \text{CZ}_{v,u}$$

and

$$\text{CX}_{v,S} := \prod_{u \in S} \text{CX}_{v,u}.$$

Lemma 3. Let (G, I, O) be an open graph with maximally delayed gflow (g, \prec) . Let us remove all the edges inside O from G and denote the resulting open graph by (G', I, O) ,

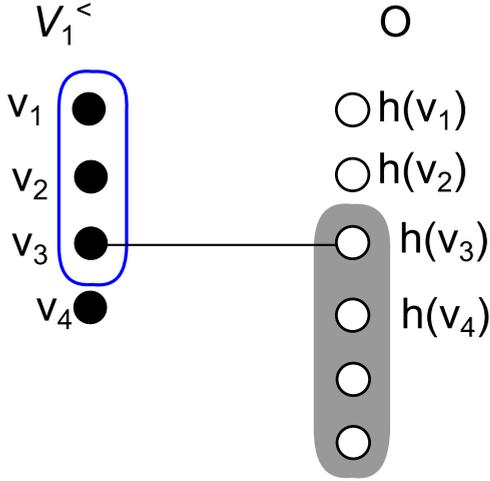


FIG. 5. (Color online) Properties of the gflow $(g_V, <_V)$. There is an edge between v_i and $h(v_i)$. A gflow image $g_V(v_3)$ is in the shaded region [Eq. (11)]. $\text{Odd}_{G'}[g_V(v_i)]$ is in the circled region [Eq. (12)].

namely,

$$\prod_{\{u,v\} \in E, u,v \in O} \tilde{\text{CZ}}_{u,v} |G'\rangle = |G\rangle.$$

Then the state

$$\tilde{\text{CX}}_{h(v_n);g_V(v_n) \oplus h(v_n)} \cdots \tilde{\text{CX}}_{h(v_1);g_V(v_1) \oplus h(v_1)} |G'\rangle$$

is also an open-graph state, where $(g_V, <_V)$ is a matching gflow of (G, I, O) , and h is the successor function of $(g_V, <_V)$. This graph does not have any edges between R_V and the outside of $V_<^1$. The subgraph of this graph consisting of $V_<^1$ and R_V has flow if the input set is $V_<^1$ and the output set is R_V .

Proof. By construction, the gflow $(g_V, <_V)$ has the properties given by

$$g_V(v_i) \subset O \setminus \{h(v_1), \dots, h(v_{i-1})\}, \quad (11)$$

$$\text{Odd}_{G'}[g_V(v_i)] \subset \{v_1, \dots, v_i\}, \quad (12)$$

where $\text{Odd}_{G'}$ represents the odd neighborhood on the graph G' (see Fig. 5). Let us denote the set $[g_V(v_i) \oplus h(v_i)] \cup \text{Odd}_{G'}[g_V(v_i) \oplus h(v_i)]$ by $W(v_i)$. The open-graph state $|G'\rangle$ is a stabilizer state of the operator

$$\begin{aligned} \tilde{K}[g_V(v_i)] &= \tilde{X}_{g_V(v_i) \oplus h(v_i)} \tilde{Z}_{\text{Odd}_{G'}[g_V(v_i) \oplus h(v_i)]} \\ &= \tilde{X}_{g_V(v_i) \oplus h(v_i)} \tilde{Z}_{\text{Odd}_{G'}[g_V(v_i) \oplus h(v_i)] \oplus N_{G'}[h(v_i)]} \\ &= \tilde{X}_{g_V(v_i) \oplus h(v_i)} \tilde{Z}_{\text{Odd}_{G'}[g_V(v_i)]} \tilde{Z}_{N_{G'}[h(v_i)]}. \end{aligned} \quad (13)$$

The stabilizer $\tilde{K}[g_V(v_i)]$ is a multiqubit local unitary transformation acting on the vertices in $W(v_i)$. It follows that the controlled version of the stabilizer $\tilde{\text{CK}}[g_V(v_i)]_{h(v_i);W(v_i)}$ also stabilizes $|G'\rangle$ (the proof is given in Appendix B), namely,

$$|G'\rangle = \tilde{\text{CK}}[g_V(v_i)]_{h(v_i);W(v_i)} |G'\rangle. \quad (14)$$

We define a sequence of open graphs $\{G_i\}_{i=0, \dots, n}$ ($n = |V_<^1|$) inductively by

$$|G_i\rangle := \tilde{F}_i |G_{i-1}\rangle, \quad (15)$$

where $G_0 := G'$ and

$$\tilde{F}_i := \tilde{\text{CZ}}_{h(v_i); \text{Odd}_{G'}[g_V(v_i)]} \tilde{\text{CZ}}_{h(v_i); N_{G'}[h(v_i)]}. \quad (16)$$

We denote the sets of edges that correspond to $\tilde{\text{CZ}}_{h(v_i); N_{G'}[h(v_i)]}$ and to $\tilde{\text{CZ}}_{h(v_i); \text{Odd}_{G'}[g_V(v_i)]}$ by E_i^{erase} and E_i^{create} , respectively. Equation (15) indicates that G_i is obtained by creating edges E_i^{create} after erasing edges E_i^{erase} from G_{i-1} . Note that the sets of edges $\{E_i^{\text{erase}} \cup E_i^{\text{create}}\}_{i=1, \dots, n}$ are disjoint to each other. An equivalent definition of G_i is given by

$$|G_i\rangle := \tilde{F}_i \tilde{F}_{i-1} \cdots \tilde{F}_1 |G_0\rangle. \quad (17)$$

We show that G_i can be alternatively defined by

$$|G_{i-1}\rangle = \tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} |G_i\rangle. \quad (18)$$

Since any two CZ gates commute,

$$\begin{aligned} &\tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} |G_i\rangle \\ &= \tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} \tilde{F}_i \cdots \tilde{F}_1 |G_0\rangle \\ &= \tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} \tilde{F}_{i-1} \cdots \tilde{F}_1 \tilde{F}_i |G_0\rangle. \end{aligned} \quad (19)$$

The CNOT gates $\tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)}$ act on vertices $h(v_i)$ (the control side) and $g_V(v_i) \oplus h(v_i)$ (the target side), which are all included in $O \setminus \{h(v_1), \dots, h(v_{i-1})\}$, and there is no CZ gate in $\tilde{F}_{i-1} \cdots \tilde{F}_1$ that acts on vertices in this set. Thus, Eq. (18) is obtained by

$$\begin{aligned} &\tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} \tilde{F}_{i-1} \cdots \tilde{F}_1 \tilde{F}_i |G_0\rangle \\ &= \tilde{F}_{i-1} \cdots \tilde{F}_1 \tilde{\text{CX}}_{h(v_i); g_V(v_i) \oplus h(v_i)} \tilde{F}_i |G_0\rangle \\ &= \tilde{F}_{i-1} \cdots \tilde{F}_1 \tilde{\text{CK}}[g_V(v_i)]_{h(v_i); W(v_i)} |G_0\rangle \\ &= \tilde{F}_{i-1} \cdots \tilde{F}_1 |G_0\rangle \\ &= |G_{i-1}\rangle. \end{aligned}$$

For example, if $|G_i\rangle$ in the left hand side of Eq. (18) is given by the open-graph state presented in Figs. 6(a) and 6(b), then the another description given by the right hand side of Eq. (18) is presented in Figs. 6(c) and 6(d).

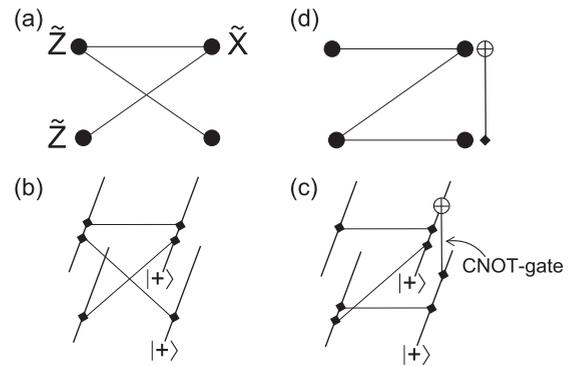


FIG. 6. (a) An open-graph state. The left two vertices are input qubits and the right two are the outputs. A product of the Pauli \tilde{X} and two Pauli \tilde{Z} operators described in the figure form a stabilizer of the graph state. (b) The circuit description of the same graph state. The four lines from the bottom left to the top right represent physical qubits on the vertices. (They are not the wires corresponding to the path cover.) (c) The equivalent circuit description including a CNOT gate. (d) The graph state obtained by applying a CNOT gate. This state is equivalent to the graph state represented by (a).

The open-graph state $|G'\rangle$ is then represented as

$$\begin{aligned} |G'\rangle &= \widetilde{\text{CX}}_{h(v_1);g_V(v_1)\oplus h(v_1)}|G_1\rangle \\ &= \widetilde{\text{CX}}_{h(v_1);g_V(v_1)\oplus h(v_1)}\widetilde{\text{CX}}_{h(v_2);g_V(v_2)\oplus h(v_2)}|G_2\rangle \\ &= \widetilde{\text{CX}}_{h(v_1);g_V(v_1)\oplus h(v_1)}\cdots\widetilde{\text{CX}}_{h(v_n);g_V(v_n)\oplus h(v_n)}|G_n\rangle, \end{aligned} \quad (20)$$

using Eq. (18). Note that the CNOT gates act inside the set of the output vertices. The edges incident from R_V to O , namely, those edges in E_i^{create} for some i , are

$$h(v_i) \sim v,$$

such that $v \in \text{Odd}_{G'}[g_V(v_i)]$. Since

$$N_{G_n}[h(v_i)] = \{v \in \text{Odd}_{G'}[g_V(v_i)]\} = \text{Odd}_{G'}[g_V(v_i)],$$

h turns out to be a flow from V_{\prec}^1 to R_V with the same partial order to \prec_V on $V_{\prec}^1 \cup R_V$.

Let (G, I, O) be an open graph with gflow, with its layers $\{V_{\prec}^i\}_{i=0,\dots,d}$ ($V_{\prec}^0 = O$) defined by maximally delayed gflow (g, \prec) . We inductively define a sequence of open graphs $\{(G^i, I^i, O^i)\}_{i=0,\dots,d}$ by

$$\begin{aligned} (G^0, I^0, O^0) &:= (G, I, O), \\ (G^i, I^i, O^i) &:= (G^{i-1} \setminus R^{i-1}, I^{i-1} \setminus R^{i-1}, V_{\prec}^i \cup [O^{i-1} \setminus R^{i-1}]), \end{aligned} \quad (21)$$

where set R^{i-1} to be removed from G^{i-1} is chosen according to Lemma 2 by taking $G = G^{i-1}$ and $V = V_{\prec}^i = \{v_1^i, v_2^i, \dots, v_{n(i)}^i\}$ ($n(i) := |V_{\prec}^i|$). Note that $I^d = O^d$.

We now have two sets of graphs: $\{G^i\}_{i=1,\dots,d}$, where d denotes the depth of maximally delayed gflow, and $\{G_i\}_{i=1,\dots,n}$, where $n = |V_{\prec}^1|$ is defined by Eq. (15). The sequence with superscripts is constructed by reducing output vertices starting from G , while the sequence with subscripts is constructed by erasing and creating edges incident to output vertices on G .

Let us denote the matching gflow for (G^i, I^i, O^i) and its successor function by (g^i, \prec^i) and h^i , respectively. The following theorem shows how to translate a measurement pattern with gflow into a circuit decomposition.

Theorem 2. A circuit decomposition of a unitary transformation implemented by a measurement pattern with gflow on open graph (G, I, O) is given by

$$U^0 U_{spt}^1 U^1 U_{spt}^2 \cdots U^{d-1} U_{spt}^d U^d, \quad (22)$$

where

$$\begin{aligned} U^i &= \prod_{u \sim_{G^i} v, u, v \in O^i} \text{CZ}_{u;v} \text{CX}_{h^i(v_1^{i+1});g^i(v_1^{i+1})\oplus h^i(v_1^i)} \\ &\cdots \text{CX}_{h^i(v_{n(i+1)}^{i+1});g^i(v_{n(i+1)}^{i+1})\oplus h^i(v_{n(i+1)}^{i+1})}, \end{aligned} \quad (23)$$

and

$$\begin{aligned} U_{spt}^i &= J(\alpha_{v_{n(i)}^i}) \text{CZ}_{v_{n(i)}^i; \text{Odd}_{G^i}[g^i(v_{n(i)}^i)]} J(\alpha_{v_{n(i)-1}^i}) \\ &\cdots \text{CZ}_{v_2^i; \text{Odd}_{G^i}[g^i(v_2^i)]} J(\alpha_{v_1^i}). \end{aligned} \quad (24)$$

Here the graph $G^{i'}$ is given by erasing all edges inside O^i from G^i .

Proof. We denote $n(1)$ by n , v_i^1 by v_i , h^0 by h , and g^0 by g_V , to inherit notations from Definition 5 and the proof of Lemma 3. Other types of labels do not appear in this proof.

We show that the circuit decomposition of the unitary transformation implemented by the last step $V_{\prec}^1 \rightarrow O^0$ is given by $U^0 U_{spt}^1$. The map represented by Eq. (2) and implemented by the measurement pattern on the graph (G, I, O) is

$$\begin{aligned} &\bigotimes_{v \in O^C} \langle +_{\alpha_v} | G \rangle \\ &= \prod_{u \sim_G v, u, v \in O} \widetilde{\text{CZ}}_{u;v} \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G' \rangle \quad (25) \\ &= \prod_{u \sim_G v, u, v \in O} \widetilde{\text{CZ}}_{u;v} \widetilde{\text{CX}}_{h(v_1);g_V(v_1)\oplus h(v_1)} \\ &\cdots \widetilde{\text{CX}}_{h(v_n);g_V(v_n)\oplus h(v_n)} \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G_n \rangle \\ &= \widetilde{U}^0 \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G_n \rangle, \end{aligned} \quad (26)$$

where \widetilde{U}^0 acts only on the output vertices. The graph G_n is defined by Eq. (15), and Eq. (26) holds from Eq. (27). The typical shape of the graph G_n is presented in Fig. 7. By performing the SPT for flow in the last step from V_{\prec}^1 to R^0 on G_n (this is always possible since Lemma 3 states that this part of G_n has flow), the map given by Eq. (27) becomes

$$U^0 U_{spt}^1 \bigotimes_{v \in O^{1C}} \langle +_{\alpha_v} | G^1 \rangle. \quad (28)$$

Note that $U^0 U_{spt}^1$ is a unitary transformation acting on the qubits represented by the wires labeled by vertices in O^1 . If we apply the SPT on the graph represented by Fig. 7, for example, we obtain a circuit presented in Fig. 9(a) as U_{spt}^1 . Equation (28) shows that the unitary transformation implemented by the measurement pattern on the open graph (G, I, O) can be decomposed into the unitary transformation implemented by the measurement pattern on the open graph (G^1, I^1, O^1) followed by $U^0 U_{spt}^1$.

Since the maximally delayed gflow is preserved by reducing open graphs (Lemma 2), the above argument is valid even if we replace (G, I, O) with (G^i, I^i, O^i) , that is,

$$\bigotimes_{v \in O^{iC}} \langle +_{\alpha_v} | G^i \rangle = U^i U_{spt}^{i+1} \bigotimes_{v \in O^{i+1C}} \langle +_{\alpha_v} | G^{i+1} \rangle \quad (29)$$

hold for $i = 0, \dots, d-1$. If $i = d$, the open graph (G^d, I^d, O^d) has the only layer $V_{\prec}^d = I^d = O^d$ and there is no measurement

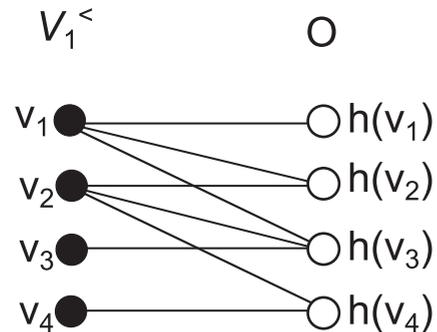


FIG. 7. A typical shape of the graph G_n . A vertex $h(v_i)$ is connected to vertices v_j where $j \leq i$.

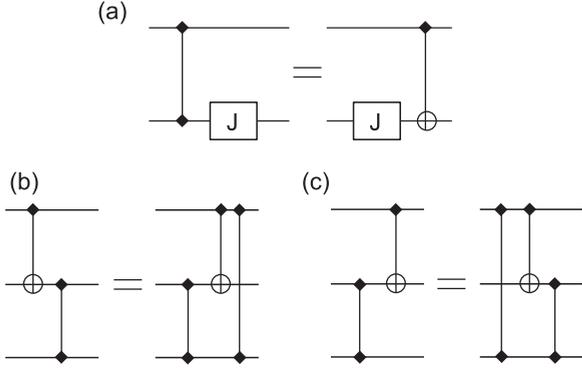


FIG. 8. Circuit identities used for the translation.

in the corresponding pattern. In this case, U^d consists only of a set of CZ gates, which is equivalent to the unitary transformation implemented by the measurement pattern on (G^d, I^d, O^d) . A concatenated application of Eq. (29) on $\otimes_{v \in O^c} \langle +_{\alpha_v} | G \rangle$ leads to

$$\begin{aligned}
 \bigotimes_{v \in O^c} \langle +_{\alpha_v} | G \rangle &= U^0 U_{spt}^1 \bigotimes_{v \in O^1} \langle +_{\alpha_v} | G^1 \rangle \\
 &= U^0 U_{spt}^1 U^1 U_{spt}^2 \bigotimes_{v \in O^2} \langle +_{\alpha_v} | G^2 \rangle \\
 &= \dots \\
 &= U^0 U_{spt}^1 U^1 U_{spt}^2 \dots U^{d-1} U_{spt}^d U^d. \quad (30)
 \end{aligned}$$

V. PARALLELIZING J GATES

Using the circuit identity presented in Fig. 8(a), U_{spt}^i is transformed into

$$\begin{aligned}
 U_{spt}^i &= CX_{v_{n(i)}^i; \text{Odd}_{G^i}[g^i(v_{n(i)}^i)]} \dots CX_{v_2^i; \text{Odd}_{G^i}[g^i(v_2^i)]} \\
 &\times \bigotimes_{j=0, \dots, n(i)} J(\alpha_{v_j^i}),
 \end{aligned}$$

which has the parallelized form for J gates. For example, if we apply this transformation to the circuit presented in Fig. 9(a) to parallelize J gates, we obtain the circuit presented in Fig. 9(b). The total unitary transformation U implemented by the measurement pattern is now written in the form

$$U = C_{\text{lifford}}^0 J^1 C_{\text{lifford}}^1 J^2 \dots C_{\text{lifford}}^{d-1} J^d C_{\text{lifford}}^d, \quad (31)$$

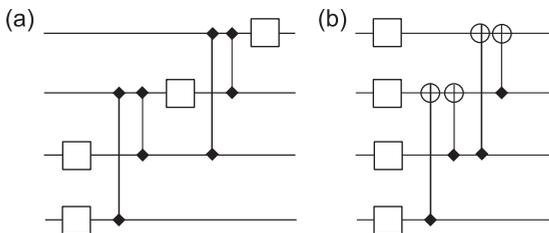


FIG. 9. (a) A circuit decomposition obtained by applying the SPT on the graph represented by Fig. 7. (b) An equivalent circuit decomposition obtained after applying the circuit identity shown in Fig. 8(a).

where each C_{lifford}^k ($k = 0, \dots, d$) consists of two-qubit Clifford gates and each J^k ($k = 1, \dots, d$) consists of parallel J gates.

The circuit representation of U given by Eq. (31) shows that the quantum depth calculated by gflow is lower bounded by the depth calculated by a quantum circuit model that implements all Clifford gates in a constant number of steps [5]. Each of the C_{lifford}^k ($k = 0, \dots, d$) is implemented in constant time by this version of the quantum circuit model. Each unitary transformation J^k ($k = 1, \dots, d$) is also implemented in constant time because all J gates act on different wires and thus they are parallelized. Therefore, the total unitary transformation U represented by Eq. (31) is implemented in $c * d$ steps by this quantum circuit model, where c denotes a constant.

In [19], a quantum circuit representing a unitary transformation implemented by a measurement pattern on a graph with flow is transformed so that the single-qubit elementary gates are parallelized. Our method is a generalization of that method for graphs with gflow.

VI. STAR-PATTERN TRANSFORMATION FOR GFLOW

We define a generalization of the SPT on the graph with gflow but no flow in this section. A straightforward application of the SPT on a graph with a path cover but without flow does not lead to a well-defined circuit, as we have noted in Sec. II C. The restriction that the target side and the control side of a CZ gate must act on the same time slice in the circuit prohibits us to write a well-defined circuit for gflow.

We formally define an *acausal CZ gate* as a two-qubit gate acting on two time slices in a circuit. We denote such an acausal CZ gate by $\text{aCZ}_{u!;v!}$, where the positions $u!$ and $v!$ on which the gate acts may be in different time slices. We pose two assumptions on the acausal CZ gate. First, if the positions $u!$ and $v!$ are regarded to be in the same time slice on the circuit, then $\text{aCZ}_{u!;v!}$ implements the same map as $\text{CZ}_{u!;v!}$. Second, if several acausal CZ gates are acting on the same position $v!$, the map implemented by these gates does not depend on the ordering of the acausal CZ gates. The latter assumption originates from the commutativity of the CZ gates acting on the same physical qubit v on the graph.

We refer to a quantum circuit decomposition composed of J gates, CZ gates, and acausal CZ gates as an *acausal circuit decomposition* [10] [see Fig. 10(c)]. The SPT on the graph with gflow but no flow is defined to be a procedure to write an acausal circuit representation for a measurement pattern on the graph. The procedure starts from applying the same processes (i) and (ii) presented for the SPT for flow in Sec. II C. These processes are applicable because there is a path cover on any graph with gflow. Positions are also defined similarly. The position between $J(\alpha_{h^{-1}(v)})$ and $J(\alpha_v)$ in the wire v is labeled by $v!$, where h is the successor function. If v is a starting vertex of the path cover, the label $v!$ represents the position before the gate $J(\alpha_v)$ in wire v . If v is one of the last vertices of the path cover, the label $v!$ represents the position just before the output in wire v . The procedure ends with placing acausal CZ gates in the circuit instead of the ordinary CZ gates in the process (iii) presented in Sec. II C. It is always possible to place $\text{aCZ}_{u!;v!}$ for any pair of positions $u!$ and $v!$.

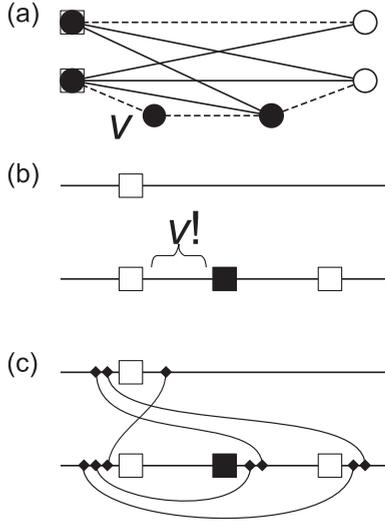


FIG. 10. (a) An open graph with gflow but no flow. The set of dashed edges represents the path cover and a vertex is labeled by v . (b) A quantum circuit given by process (ii) of the SPT (Sec. II C) applied on the graph given by (a). The two wires correspond to the path cover, and boxes represent the J gates. The gate $J(\alpha_v)$ is represented by the black box. The position $v!$ is the region of the wire between $J(\alpha_v)$ and $J(\alpha_{h^{-1}(v)})$. (c) A quantum circuit representing a measurement pattern on the graph given by (a).

VII. TRANSFORMING ACAUSAL CIRCUITS

Although direct application of the SPT on a graph without flow leads to an acausal circuit, our translation method presented in Sec. IV and the SPT have two common aspects. One is the correspondence between a path cover and a wire of the circuit. Another is the correspondence between each measurement and a J gate. Since our method translates a graph with gflow into a well-defined circuit, we expect that the acausal circuit obtained by the SPT should be transformable into a well-defined one by taking a suitable circuit transformation. In this section, we present this circuit transformation.

We define an acausal CZ gate using ancilla qubits and postselection to be consistent with the unitary transformation implemented by a measurement pattern with gflow. In [20], an acausal gate is identified with a circuit simulating the effect of a closed timelike curve (CTC) (see Fig. 12). This circuit, including ancilla qubits and postselection of the measurement results, is proposed by Bennett and Schumacher [21] and by Svetlichny [22] to simulate the disordered time effect of CTC by quantum circuits and is called the BSS-type CTC. In a similar manner, we define acausal gates by using ancilla qubits and postselection.

Lemma 4. Consider an acausal circuit obtained by directly applying the SPT on a graph with gflow. Define an acausal CZ gate $\text{aCZ}_{u!;v!}$ acting on positions $u!$ and $v!$ by

$$\text{aCZ}_{u!;v!} := \langle +|_{u'} \langle +|_{v'} \text{CZ}_{u!;u'} \text{CZ}_{u';v'} \text{CZ}_{v';v!} | + \rangle_{u'} | + \rangle_{v'} \rangle, \quad (32)$$

where $|+\rangle_{u'}$ and $|+\rangle_{v'}$ represent the initial states of the ancilla qubits, and $\langle +|_{u'}$ and $\langle +|_{v'}$ represent a postselected measurement branch for a projective measurement described by $\{|\pm\rangle_{i'}\langle \pm|\}$, for $i = u, v$, respectively. We postselect the

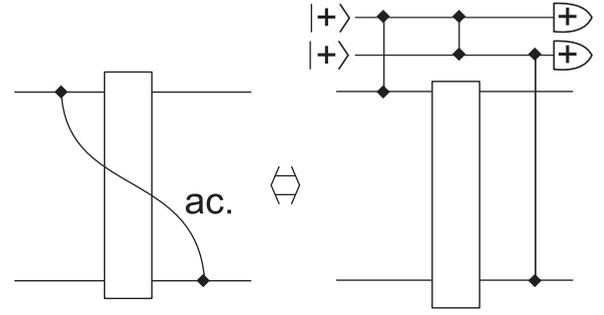


FIG. 11. The definition of an acausal CZ gate. Two ancilla qubits are initially prepared in $|+\rangle$ states, and are postselected to be in the $|+\rangle$ state at the final measurements.

measurement result “+”. (See Fig. 11.) Then the acausal circuit represents a unitary transformation equivalent to the one implemented by the measurement pattern on the graph.

Proof. Rewriting the acausal CZ gates according to Eq. (32) is always possible. This is because all of the CZ gates appearing in Eq. (32) commute with each other, and there are no other gates which define an ordering of gates on the ancilla qubits u' and v' .

Using the definition of an acausal CZ gate given by Eq. (32), the acausal circuit can be transformed into a well-defined one. The circuit is composed of three parts: preparation of initial ancilla states, a circuit consisting of J gates and CZ gates, and final measurements. The circuit in the second part can be transformed to a measurement pattern by performing the inverse transformation of the SPT [17]. Thus we obtain the corresponding open graph (G', I', O') with flow given by

$$\begin{aligned} V(G') &= V(G) \cup_{\{u,v\} \notin P_h} \{u', v'\}, \\ E(G') &= E(G) \cup_{\{u,v\} \notin P_h} \{E_{uu'} E_{u'v'} E_{v'v}\} \setminus \cup_{\{u,v\} \notin P_h} E_{uv}, \\ I' &= I \cup_{\{u,v\} \notin P_h} \{u', v'\}, \\ O' &= O \cup_{\{u,v\} \notin P_h} \{u', v'\}, \end{aligned}$$

where P_h is defined by substituting the successor function h instead of the flow function used in Eq. (3). There are trivial I/O (input/output) paths from the newly added vertices to themselves since they are included both in the input set and in the output. The original path cover on the open graph G and these trivial paths construct a path cover on the open graph G' . Let us denote the unitary transformation implemented by (G', I', O') by $U_{G'}$. Then, from $U_{G'} \propto \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G' \rangle$, we have

$$\begin{aligned} & \bigotimes_{\{u,v\} \notin P_h} \langle +|_{u'} \langle +|_{v'} U_{G'} | + \rangle_{u'} | + \rangle_{v'} \rangle \\ & \propto \bigotimes_{\{u,v\} \notin P_h} \langle +|_{u'} \langle +|_{v'} \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G' \rangle | + \rangle_{u'} | + \rangle_{v'} \rangle. \quad (33) \end{aligned}$$

Since all the projectors commute, we can perform the projector $|+\rangle\langle +|$ on the ancilla qubits first. Using the relation

$$\begin{aligned} \tilde{\text{CZ}}_{u;v} & \propto \langle +|_{u'} \langle +|_{v'} \tilde{\text{CZ}}_{u;u'} \tilde{\text{CZ}}_{u';v'} \\ & \quad \times \tilde{\text{CZ}}_{v';v} | + \rangle_{u'} | + \rangle_{v'} \rangle \quad (\forall \{u, v\} \notin P_h), \end{aligned}$$

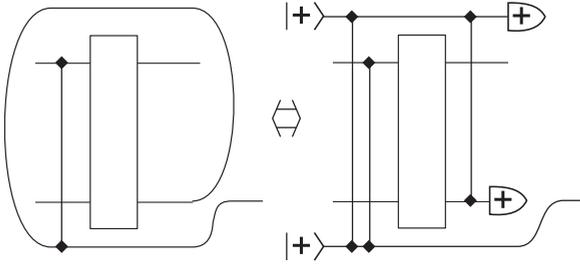


FIG. 12. The equivalent circuit representations for the acausal CZ gates given by Fig. 11. The proof of the equivalence of the circuits presented in the right-hand side and in the right-hand side circuit of Fig. 11 is given in Appendix A.

we have

$$\begin{aligned}
& \bigotimes_{\{u,v\} \notin P_h} \langle +|_{u'} \langle +|_{v'} \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G' \rangle | + \rangle_{u'} | + \rangle_{v'} \\
& \propto \bigotimes_{\{u,v\} \notin P_h} \langle +|_{u'} \langle +|_{v'} \bigotimes_{v \in O^C} \langle +_{\alpha_v} | \widetilde{CZ}_{u;u'} \widetilde{CZ}_{u';v'} \\
& \quad \times \widetilde{CZ}_{v';v} | G' \cap P_h \rangle | + \rangle_{u'} | + \rangle_{v'} \\
& \propto \bigotimes_{v \in O^C} \langle +_{\alpha_v} | G \rangle,
\end{aligned}$$

namely, the graph G' now returns to the original graph G . Thus if the ancilla qubits of $U_{G'}$ are prepared in $|+\rangle$ and the final measurements postselect the final states to be $|+\rangle$, then the unitary transformation implemented by the measurement pattern on (G, I, O) is also implemented in this postselected way.

The definition of an acausal gate by Eq. (32) is equivalent to the BSS-type CTC [21,22] (Fig. 12). The CZ gate appearing on the left-side circuit of Fig. 12 acts on two qubits, where one of the qubits has returned from a future to its past. Acausal CZ gates defined by the circuit presented on the left-side picture of Fig. 12 also appear in Ref. [20].

Although the acausal circuit can always be transformed into an ordinary circuit without acausal gates in this way, it cannot be implemented deterministically in general since the circuit includes postselection of measurement results. However, we will show that it is possible to transform the circuit to a deterministically implementable ordinary circuit by taking further transformations.

Let us denote an acausal circuit obtained by applying the SPT on an open graph G by C_{spt}^G . Since the CNOT gates in the right-hand side of Eq. (18) are acting on output qubits, the circuit identity

$$CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i-1} = C_{spt}^{G_i}, \quad (34)$$

where the CNOT gates are placed just before the output of the circuit, holds for the sequence of open graphs $\{G_i\}$ up to a normalization factor. Further, there exists a sequence of circuit transformations obtainable from one circuit to another, as shown in the following lemma.

Lemma 5. An acausal circuit $C_{spt}^{G_i}$ is obtained from a circuit represented by $CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i-1}$ by performing circuit transformations appropriately chosen from the circuit identities for acausal gates presented in Fig. 13.

Proof. We first replace the CNOT gates appearing on the left-hand side of Eq. (34) with acausal CNOT gates. We shift

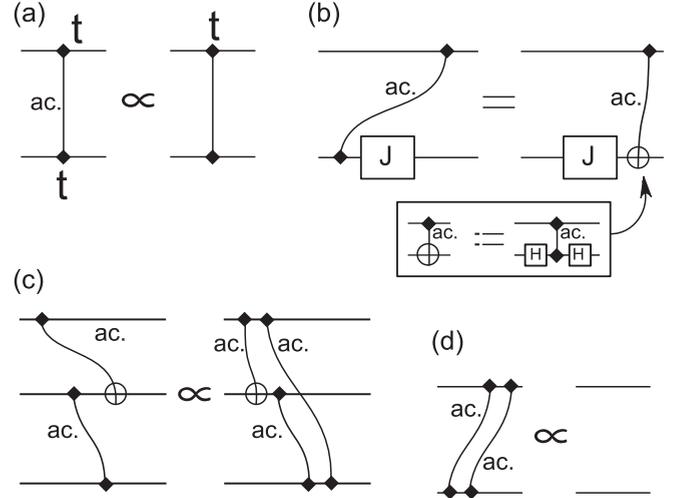


FIG. 13. The circuit identity satisfied by the acausal gates. The acausal gates are labeled “ac.” The acausal CNOT gate is defined as the acausal CZ gate sandwiched by Hadamard gates. The proof of these identities is given in Appendix A.

the position of the target of these acausal CNOT gates backwards in time, until all the acausal CNOT gates are changed to acausal CZ gates by the circuit identity presented in Fig. 13(b). In this proof, if \mathcal{S} is a set of vertices, $\mathcal{S}!$ denotes the set of positions $\{v! | v \in \mathcal{S}\}$.

The target of an acausal CNOT gate acting on v [$v \in g_V(v_i) \oplus h(v_i)$] first hits the acausal CZ gates corresponding to nonpath edges incident from the vertex v . These acausal CZ gates are represented by

$$aCZ_{v!; N(v) \setminus h^{-1}(v)!}.$$

We consider two processes for circuit transformations.

Process 1. By commuting the acausal CNOT gate and these acausal CZ gates, new acausal CZ gates are created between $h(v_i)!$ and $N(v) \setminus h^{-1}(v)!$, using the circuit identity presented in Fig. 13(c). These newly created acausal CZ gates are

$$aCZ_{h(v_i)!; N(v) \setminus h^{-1}(v)!}.$$

Process 2. Further commuting the acausal CNOT gate with $J(\alpha_v)$, it changes to an acausal CZ gate using the circuit identity presented in Fig. 13(b). The resulting acausal CZ gate is

$$aCZ_{h(v_i)!; h^{-1}(v)!}.$$

By Process 1 and Process 2, the acausal CNOT gate is transformed to

$$aCZ_{h(v_i)!; N(v) \setminus h^{-1}(v)!} aCZ_{h(v_i)!; h^{-1}(v)!} = aCZ_{h(v_i)!; N(v)!}.$$

If we perform these transformations for all of the CNOT gates appearing on the left-hand side of Eq. (34), the CNOT gates transform,

$$\begin{aligned}
& \prod_{v \in g_V(v_i) \oplus h(v_i)} aCZ_{h(v_i)!; N(v)!} \\
& = \prod_{v \in g_V(v_i)} aCZ_{h(v_i)!; N(v)!} aCZ_{h(v_i)!; N[h(v_i)!]} \\
& \propto aCZ_{h(v_i)!; \text{Odd}[g_V(v_i)]!} aCZ_{h(v_i)!; N[h(v_i)]!} \quad (35)
\end{aligned}$$

$$\propto aCZ_{h(v_i)!; \text{Odd}[g_V(v_i)]!} v_i! aCZ_{h(v_i)!; N[h(v_i)] \setminus v_i!} \quad (36)$$

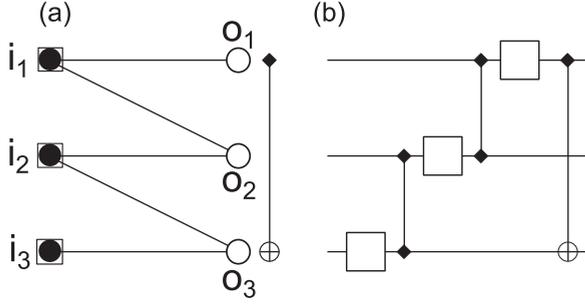


FIG. 14. (a) An open graph transformed according to Eq. (18) from the graph given by Fig. 2. The extra CNOT gate must act on the qubits o_1 and o_2 . (b) The circuit decomposition obtained by our method in Sec. IV.

Equation (35) holds since an even number of acausal CZ gates acting on the same pairs of positions are canceled by the circuit identity presented in Fig. 13(d), and since $\bigoplus_{v \in g_V(v_i)} N(v) = \text{Odd}[g_V(v_i)]$. Equation (36) holds since v_i is in $\text{Odd}[g_V(v_i)]$ and also in $N[h(v_i)]$. The last term $aCZ_{h(v_i)!, N[h(v_i)] \setminus v_i}!$ cancels all (acausal) CZ gates incident from $h(v_i)!$. New acausal CZ gates are created by the first term $aCZ_{h(v_i)!, \text{Odd}[g_V(v_i)] \setminus v_i}!$. These transformations directly correspond to the transformation from G_{i-1} to G_i given by Eq. (15).

Now we present how to transform acausal circuits into ordinary circuits by using Lemma 5. An example is shown in Fig. 15 where an acausal circuit obtained from the open graph presented in Fig. 2 is transformed to an ordinary circuit presented in Fig. 14(b).

Lemma 6. Any acausal circuit obtained by directly applying the SPT on any graph with gflow is transformed to the circuit presented in Theorem 2, by performing suitable circuit

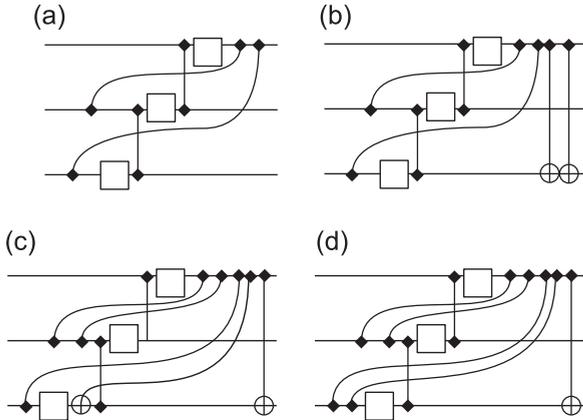


FIG. 15. (a) The acausal circuit representation for a unitary transformation implemented by the measurement pattern on the open graph presented in Fig. 2. We choose the edges (i_2, o_1) and (i_3, o_1) as the acausal CZ gates. (b) Applying $\mathbb{I} = \text{CNOT} \cdot \text{CNOT}$ on the control qubit, $h(i_1) = o_1$, and the target qubit, $g_V(i_1) \setminus h(i_1) = \{o_3\}$. (c) Shifting the position of the acausal CNOT gate before the J gate through the CZ gate. (d) Shifting the position of the acausal CNOT gate through the J gate. It is changed to an acausal CZ gate. By canceling the pairs of acausal CZ gates according to the circuit identity given in Fig. 13(d), this acausal circuit is transformed to the circuit presented in Fig. 14(b).

transformations chosen from the circuit identities presented in Fig. 13 and a circuit identity

$$CX_{u,v}CX_{u,v} = \mathbb{I} \quad (\forall u, v). \quad (37)$$

Proof. All acausal CZ gates on C_{spt}^G which originate from CZ gates on pairs of output qubits on the graph state can be changed into ordinary CZ gates by the circuit identity presented in Fig. 13(a). These changes can be expressed by a transformation,

$$C_{spt}^G \rightarrow \prod_{u \sim_G v, u, v \in O} CZ_{u,v} C_{spt}^{G'}, \quad (38)$$

where graph G' is defined in Lemma 3.

Next we start from $C_{spt}^{G'} = C_{spt}^{G_0}$ and inductively transform $C_{spt}^{G_{i-1}}$ to $CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i}$. Using the circuit identity by Eq. (37), the circuit is transformed to

$$C_{spt}^{G_{i-1}} \rightarrow (CX_{h(v_i); g_V(v_i) \oplus h(v_i)})^2 C_{spt}^{G_{i-1}}. \quad (39)$$

Using Lemma 5, the circuit is further transformed to

$$(CX_{h(v_i); g_V(v_i) \oplus h(v_i)})^2 C_{spt}^{G_{i-1}} \rightarrow CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i}. \quad (40)$$

By combining transformations given in Eqs. (39) and (40), the circuit is transformed to

$$C_{spt}^{G_{i-1}} \rightarrow CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i}. \quad (41)$$

Starting from $C_{spt}^{G'}$, inductive application of transformations from $C_{spt}^{G_{i-1}}$ to $CX_{h(v_i); g_V(v_i) \oplus h(v_i)} C_{spt}^{G_i}$ results in a transformation,

$$C_{spt}^{G'} \rightarrow CX_{g_V(v_1) \oplus h(v_1)} \cdots CX_{h(v_n); g_V(v_n) \oplus h(v_n)} C_{spt}^{G_n}. \quad (42)$$

Since there exists flow between the last two layers of G_n , all acausal CZ gates inside this region of $C_{spt}^{G_n}$ can be changed into ordinary CZ gates [Fig. 13(a)] by the circuit identity presented in Fig. 13(a). This is expressed by the transformation

$$C_{spt}^{G_n} \rightarrow U_{spt}^0 C_{spt}^{G^1}, \quad (43)$$

where graphs $\{G^i\}_{i=0, \dots, d}$ are defined by Eq. (22).

In total, the circuit is transformed to

$$C_{spt}^G \rightarrow U^0 U_{spt}^1 C_{spt}^{G^1}. \quad (44)$$

If we start transformations from $C_{spt}^{G^1}$, the circuit is transformed to

$$C_{spt}^{G^1} \rightarrow U^i U_{spt}^{i+1} C_{spt}^{G^{i+1}}, \quad (45)$$

for $i = 0, \dots, d-1$. Thus, analogously to Eq. (30), any C_{spt}^G can be transformed to

$$\begin{aligned} C_{spt}^G &\rightarrow U^0 U_{spt}^1 C_{spt}^{G^1} \\ &\rightarrow U^0 U_{spt}^1 U_{spt}^2 C_{spt}^{G^2} \\ &\rightarrow \cdots \\ &\rightarrow U^0 U_{spt}^1 U_{spt}^2 \cdots U_{spt}^{d-1} U_{spt}^d U^d. \end{aligned}$$

The next theorem follows immediately from Lemma 6.

Theorem 3. All acausal gates appearing on an acausal circuit obtained by directly applying the SPT on any graph with gflow can be canceled to give a circuit that is deterministically implementable.

Despite the fact that we have originally defined the acausal circuit including ancillas and postselection, Theorem 3 shows that all acausal gates cancel in this case and deterministic implementation of a unitary transformation is possible.

VIII. DEPTH COMPRESSION AND ACAUSAL CIRCUITS

In Sec. V, we have shown that the unitary transformation implemented by the measurements on qubits in a single layer of gflow is written as parallelized J gates followed by a sequence of Clifford gates. Any unitary transformation that is written in this form is implemented in a constant quantum depth by the measurement pattern. This is in contrast to a variation of the quantum circuit model where classically controlled operations depending on measurement outcomes are not included. Generally, in such a model, the quantum depth depends on the system size.

In this section, we show how the acausal circuit obtained by directly applying the SPT expresses the depth compression by extending the definition of the temporal ordering of gates. In this section, the acausal gates are regarded as shorthands for circuits implementing gate teleportation [23]. With the aid of ancilla qubits and postselection, this circuit can be interpreted to have a power equivalent to sending quantum states back into the past, from where the computation continues again. The condition of postselection is circumvented by applying suitable correction operators depending on the outcomes of the Bell measurement in gate teleportation.

A. Gate teleportation

Let us describe how to parallelize unitary transformations and probabilistically compress the quantum depth by using a postselection version of the gate-teleportation protocol [23]. We first review the gate-teleportation protocol for this purpose. Consider a one-qubit circuit representing two unitary transformations U and U' applied on a single Hilbert space labeled A in a sequence [see Fig. 16(a)]. The unitary transformation U'_A must be applied after the gate U_A for implementing a unitary transformation $U'U$ on a state on A .

With the aid of spatial resources, namely, ancilla qubits, we construct another circuit where U and U' are applied on different qubits *in parallel*, but the *ordered sequence* of unitary transformations $U'U$ is still implemented. Consider a circuit that has three qubits labeled by A , B , and C , depicted in Fig. 16(b). The initial state of qubits B and C is prepared in a maximally entangled state, $|\Psi\rangle_{BC} := CZ_{BC}|+\rangle_B|+\rangle_C$. After performing unitary transformation U on qubit A , qubits A and B are measured in a basis $\{CZ|s_A\rangle_A|s_B\rangle_B\}_{s_A, s_B \in \{+, -\}}$. This measurement is called the Bell measurement. A Pauli correction operator $P_C(s_A, s_B)$ depending on the measurement outcome is applied on qubit C , where $P_C(+, +) = \mathbb{I}_C$ (do nothing), $P_C(+, -) = X_C$, $P_C(-, -) = Y_C$, and $P_C(-, +) = Z_C$. Finally, the unitary transformation U' is performed on qubit C . In short, after U_A , we perform a quantum teleportation from A to C , followed by U'_C . An initial state $|\phi\rangle_A|\Psi\rangle_{BC}$ is transformed to

$$U'_C P_C(s_A, s_B) \langle s_A | \langle s_B | U_A | \Psi \rangle_{BC} | \phi \rangle_A = \frac{1}{2} U'_C U_C | \phi \rangle_C, \quad (46)$$

namely, apart from the difference on the Hilbert space where the output state is obtained, the circuit presented in Fig. 16(b)

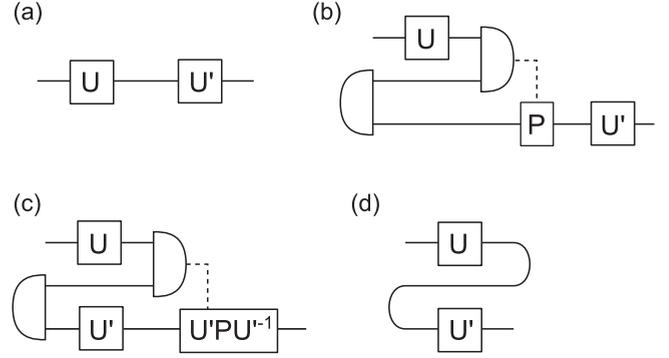


FIG. 16. (a) A circuit to implement a unitary transformation $U'U$. The unitary transformations U and U' may consist of sequences of gates. (b) A circuit obtained by inserting a teleportation between U and U' presented in (a). The half ellipsoids at the left and right ends of the wires represent a preparation of a maximally entangled state and a Bell measurement, respectively. The white box labeled “P” represents the Pauli correction operator. This is a circuit representation of the sequence of operations presented in Eq. (46). (c) The circuit representation of the sequence of operations presented in Eq. (47). (d) A circuit obtained by assuming postselection in the Bell measurement in the circuit presented in (c). The curved wire is an abbreviation of the postselected teleportation, similar to the curved wire presented in Fig. 12. This abbreviation is motivated by the fact that quantum states can be sent “back in time” by postselected teleportation.

implements the same unitary transformation as the circuit presented in Fig. 16(a).

In this form, we still have to perform U'_C after U_A , since $P'_C(s_A, s_B)$ must be performed after the outcome (s_A, s_B) is obtained by the measurement on qubits A and B . However, we can rewrite the left-hand side of (46) as

$$U'_C P_C(s_A, s_B) U_C^{-1} \langle s_A | \langle s_B | CZ_{AB} U'_C U_A | \Psi \rangle_{BC} | \phi \rangle_A. \quad (47)$$

On the right-hand side of Eq. (47), U_A and U'_C is applied in parallel, before the correction operator $U'_C P_C(s_A, s_B) U_C^{-1}$ [see Fig. 16(c)]. In general, the quantum depth for implementing $U'_C P_C(s_A, s_B) U_C^{-1}$ may be greater than that for implementing U'_C . Thus the quantum depth is not necessarily reduced in spite of the unitary transformations U and U' being parallelized.

If the measurement result is postselected to give $(s_A, s_B) = (+, +)$, there is no need for the correction, and the quantum depth for implementing $U'U$ is reduced by parallelizing U and U' . In this postselected branch, the teleportation represents a map that can be interpreted to send a quantum state “back in time,” in the same way as in the interpretation of the BSS-type CTC [21,22]. The argument of this interpretation is given as follows. Consider a case where $U = U' = \mathbb{I}$ and we perform a measurement in an arbitrary basis $\{|\varphi\rangle_m\}_{m \in M}$ on the output state obtained in C *before the input state $|\phi\rangle_A$ is prepared*. The probability to obtain measurement result m' is

$$\begin{aligned} & 4 \| \langle + | \langle + | \langle + | \langle + | CZ_{AB} | \phi \rangle_A \langle \varphi |_{m'} | \Psi \rangle_{BC} \|^2 \\ &= 4 \| \langle \varphi |_{m'} \langle + | \langle + | \langle + | \langle + | CZ_{AB} | \phi \rangle_A | \Psi \rangle_{BC} \|^2 \\ &= \| \langle \varphi |_{m'} | \phi \rangle_A \|^2, \end{aligned} \quad (48)$$

in the postselected branch. This equality implies that the probability distribution of the measurement performed on C before the preparation of state $|\phi\rangle_A$ is equal to the probability

distribution of the measurement performed on state $|\phi\rangle_A$. This is equivalent to saying that the state $|\phi\rangle_A$ is sent back in time, since the basis of the measurement is arbitrary. From this perspective, the teleportation protocol can be interpreted to reduce the quantum depth by sending the quantum state into the past and allowing the computation to continue again from there.

Using the notation introduced for representing the BSS-type CTC, a circuit for sending a quantum state back in time should be depicted by a curved wire, as represented in Fig. 16(d). Note that this expression indicates that the Hilbert spaces described by qubits A and C are considered to be the Hilbert spaces of an identical qubit at two different *temporal* positions, instead of the Hilbert spaces of the two *spatially* different qubits existing at the same time.

The definition of the partial ordering of gates included in a circuit with such curved wires must be extended from that defined on an ordinary circuit. In a circuit without curved wires, the partial ordering $g_1 \leq g_2$ between gates g_1 and g_2 applied on the same qubit is defined when g_2 is performed *after* g_1 . The quantum depth of a circuit without curved wires is then defined as the maximum number of gates included in any sequence of gates g_1, g_2, \dots, g_n such that $g_i \leq g_{i+1}$. To define the partial ordering on gates included in circuits with curved wires, we have to reconsider how to define a situation to describe when a gate g_2 is performed *after* g_1 . Consider a circuit depicted in Fig. 17. Three gates g_1, g_2 , and g_3 are performed sequentially, and the state is sent back from the time represented by a dotted line to the time represented by a dashed line, and then gates g'_2, g'_3 , and g_4 are applied in a sequence. Partial orderings $g_1 \leq g_2 \leq g_3$ and $g'_2 \leq g'_3 \leq g_4$ should be defined as usual. Let us assume the time represented by the dotted line is just after g_3 and just before g_4 , and the time represented by the dashed line is just after g_1 and just before g'_2 . Thus partial orderings $g_1 \leq g'_2$ and $g_3 \leq g_4$ are defined by using the time represented by the dashed and the dotted lines as intermediaries, respectively. However, there should be no partial ordering between g_3 and g'_2 . Extending the definition of partial ordering in this way, the quantum depth of a circuit with curved wires is defined in the same way as that for a circuit without curved wires.

Parallelization of unitary transformations is to remove the partial ordering between the unitary transformations by using curved wires. From this perspective, the teleportation protocol reduces the quantum depth by extending the definition of temporal ordering of gates.

Of course, postselection is probabilistic. The teleportation protocol for sending the quantum state back in time or, equivalently, the operation represented by the circuit with curved wires cannot be deterministically implemented in quantum mechanics in general. However, in some cases, we can deterministically *simulate* it by applying appropriate correction operators. If the correction $U'_C P'_C(s) U'^{-1}_C$ is more efficiently implementable than applying U' or U , it is possible to reduce the quantum depth without assuming postselection. For example, if U' consists of a sequence of Clifford gates, $U'_C P'_C(s) U'^{-1}_C$ is equivalent to a Pauli operator whose quantum depth is one. Thus the quantum depth for implementing the circuit obtained by parallelizing J gates as in Sec. V can be reduced to $c * d$, where c and d denote a constant and the depth of gflow, respectively. As we will see in the next section, there

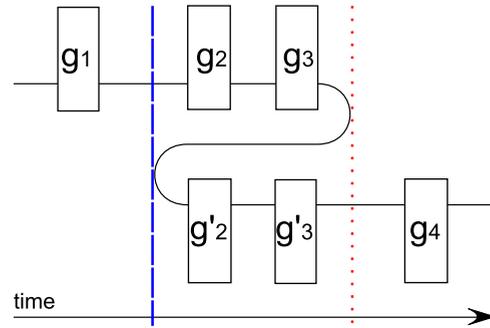


FIG. 17. (Color online) A circuit which has extended temporal ordering of gates. The white boxes labeled by g_i represent quantum gates. At the time represented by a dotted line, the state is sent back in to the time represented by a dashed line. The partial ordering of gates in this circuit is defined by $g_1 \leq g_2 \leq g_3 \leq g_4$ and $g_1 \leq g'_2 \leq g'_3 \leq g_4$.

are non-Clifford unitary transformations that change certain Pauli correction operators into other Pauli operators, and these transformations are relevant to the quantum depth of patterns with gflow. Note that the gate-teleportation protocol reduces the quantum depth by using additional ancillary systems and classically controlled operations.

B. Depth compression described by acausal gates

In this section, we apply the mechanism of depth compression presented in the previous section to the acausal circuits obtained by directly applying the SPT on measurement patterns with gflow. It is possible to reduce the quantum depth of certain types of gate sequences including J gates, which are not Clifford in general. The Pauli Z operator changes to the Pauli X operator by the conjugate action of a J gate. This allows us to reduce the quantum depth of a circuit decomposition obtained by applying the SPT on the measurement pattern implemented in the last two layers of G_n defined in Eq. (18) [e.g., the circuit presented in Fig. 9(a)]. Note that the open graph constituting the vertices of the last two layers of G_n has a gflow with its depth of one.

The acausal CZ gates describe the depth compression of circuits of this type through its equivalent description by the curved wires. Consider again the circuit depicted in Fig. 9(a). The acausal circuit description of this circuit presented in Fig. 18(a) is equivalent to the circuit with curved wires presented in Fig. 18(b), which can be deterministically simulated by performing suitable Pauli corrections as presented in Fig. 18(c). The circuits presented in Figs. 18(a)–18(c) show that J gates are applied in parallel by considering the extended temporal ordering of the gates. Thus, circuits with acausal CZ gates not only represent the unitary transformation implemented by the original pattern, but also capture the extended temporal ordering of gates after parallelizing gates by the gate-teleportation protocol.

This example shows that a class of unitary transformations implemented by acausal circuits and circuits with curved wires in constant quantum depth can be deterministically implemented by measurement patterns and circuits simulating the curved wires by the gate-teleportation protocol, also in

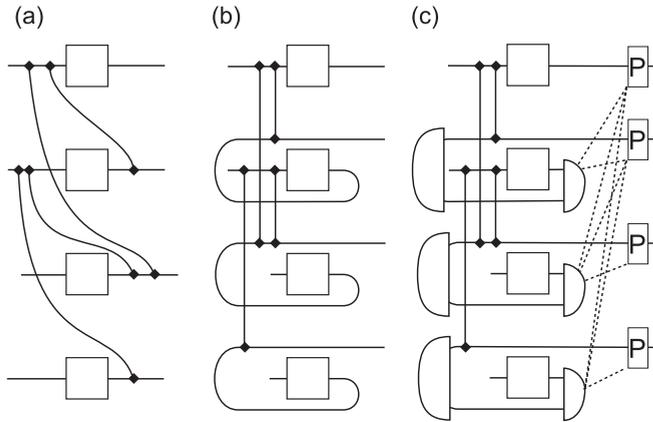


FIG. 18. (a) An acausal circuit description of the circuit decomposition presented in Fig. 9(a). (b) An equivalent circuit with curved wires obtained after representing the acausal CZ gates by using a circuit to represent the BSS-type CTC. (c) A circuit to implement the unitary transformation described by the circuit presented in (a) and (b) deterministically, by employing Pauli corrections depending on the measurement outcomes of the Bell measurements. The white boxes labeled by “P” represent Pauli operators. (The sequence of CZ gates should be parallelized by other gate teleportations not depicted here.)

constant quantum depth. Of course, this does not necessarily hold for all unitary transformations, since it may take a large quantum depth for applying correction operators in gate-teleportation protocols in general. The examples presented in Figs. 18 and 19 indicate several unitary transformations and their acausal circuit representations are given by simply applying the direct SPT on measurement patterns with gflow.

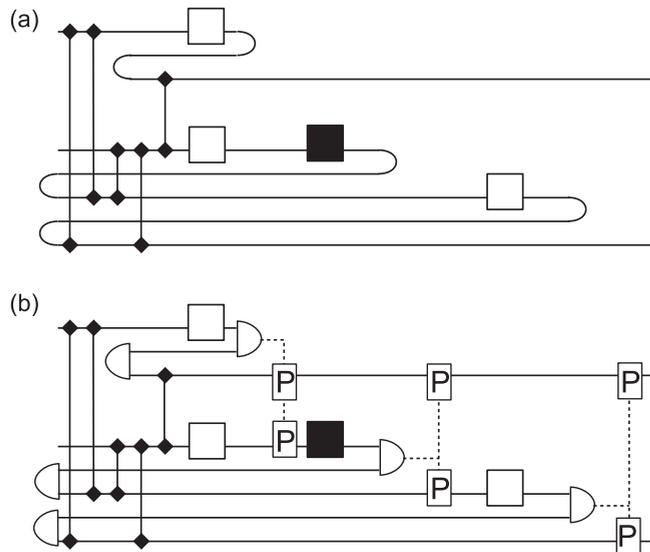


FIG. 19. (a) An acausal circuit equivalent to the one presented in Fig. 10(c) obtained after representing the acausal CZ gates by using a circuit to represent the BSS-type CTC. (b) A circuit to implement the unitary transformation described by the circuit presented in (a) deterministically, by employing Pauli corrections depending on the outcomes of the Bell measurements. The white boxes labeled “P” represent Pauli correction operators.

IX. CONCLUDING REMARKS

In this work, we have constructed two methods to translate a unitary transformation implemented by a measurement pattern on a graph with gflow to its circuit representation to analyze the trade-off relationship of the spatial and temporal resources in MBQC. The first method is a generalization of the method presented in Ref. [19] and is also an extended version of the method proposed in Ref. [12]. We have divided an open graph with gflow into layers, and transformed each layer into an open graph with flow followed by a sequence of CNOT gates. The unitary transformation implemented by the measurements on each layer is thus written by the part obtained by the SPT and the sequence of CNOT gates. We also transformed the SPT part into a circuit consisting of a sequence of Clifford gates and parallelized J gates. The resulting circuit exhibits that the depth of gflow corresponds to the depth calculated by a special version of quantum circuit model where any sequence of Clifford gates is regarded to be implemented in constant depth.

In the second translation method, a measurement pattern is translated via an acausal circuit including acausal gates obtained by directly applying the SPT on a graph with gflow but no flow. We defined these acausal gates in terms of postselection. Based on this definition, all acausal gates can be canceled by taking appropriate transformations of the circuit. This leads to a well-defined ordinary circuit representation for the measurement pattern, which is equivalent to the one obtained by the first translation method.

Finally, we have shown how the acausal circuits obtained by directly applying the SPT on measurement patterns with gflow express the depth compression by introducing an extended definition of the temporal ordering of gates in their equivalent circuit representation with curved wires by using a gate-teleportation protocol, appropriate correction operators depending on the outcomes of the Bell measurements are required.

We conjecture that in order to lift the condition of postselection entirely from the acausal circuits, it is sufficient to perform a single layer of Pauli corrections per layer of J gates according to the measurement outcomes for implementing the acausal CZ gates acting astride the J gates, as depicted in Figs. 18(c) and 19. This is because any part of an acausal circuit corresponding to a single layer of gflow can be rewritten into a circuit constituting of a single layer of J gates and a sequence of Clifford gates, and it suffices to perform a single layer of Pauli corrections for parallelizing all the Clifford gates included in the sequence. If this conjecture is true, the quantum depth of the gate-teleportation protocol to implement a gate sequence corresponding to the measurements on qubits in a single layer of gflow becomes constant. This implies that all acausal circuits obtained by directly applying the SPT on measurement patterns with gflow can be simulated by the gate-teleportation protocol with the depth equal to the depth of gflow.

Our formulation presents a way to understand the trade-off relationship between the temporal and spatial resources in quantum computation in terms of the extended temporal ordering of the acausal circuits. The temporal resource, or quantum depth, of quantum computation represented by MBQC is reduced from the ordinary quantum circuit model by extending

the definition of the temporal ordering of gates. The spatial resource, or an ancilla system, is required for simulating the extended temporal ordering by the gate-teleportation protocol. If our conjecture is true, this mechanism of the trade-off relationship explains the depth compression of MBQC. We leave the rigorous study of this conjecture for future works.

ACKNOWLEDGMENTS

We thank D. Markham, K. Nakago, and E. Pius for helpful discussions. This work was supported by the Project for Developing Innovation Systems of the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. M.M. and M.H. acknowledge support from JSPS by KAKENHI (Grants No. 23540463, No. 23240001, No. 26330006, and No.23-01770). M.H. also acknowledges support from Singapore’s National Research Foundation and Ministry of Education. This material is based on research funded by the Singapore National Research Foundation under NRF Award No. NRF-NRFF2013-01. J.M. is supported by the Leading Graduate Course for Frontiers of Mathematical Sciences and Physics. The authors also gratefully acknowledge the ELC project [Grant-in-Aid for Scientific Research on Innovative Areas MEXT KAKENHI (Grant No. 24106009)] for encouraging the research presented in this paper.

APPENDIX A: PROOF OF THE CIRCUIT IDENTITIES IN FIG. 13 AND THE EQUIVALENCE OF THE CIRCUITS IN FIGS. 11 AND 12

Once the acausal gates shown in Fig. 13 are replaced by ordinary CZ gates and CNOT gates, the circuit identities presented in Fig. 13 are easily seen to hold. We prove that these identities also hold for acausal gates. The transformation method of circuits is depicted in Figs. 20–24. There are two identities commonly used in the translation method.

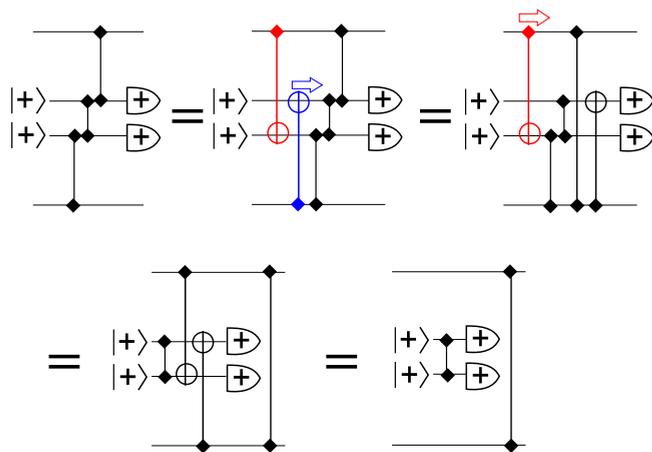


FIG. 20. (Color online) A proof of the circuit identity presented in Fig. 13(a). The first equality is by Eq. (A1) and the last equality is by Eq. (A2). The second and the third equalities are by Eq. (A3). The shifts of the position of the CNOT gates in the second and third equalities are allowed because the acausal gate acts at a single time. The probability to obtain the correct measurement result is always one-quarter.

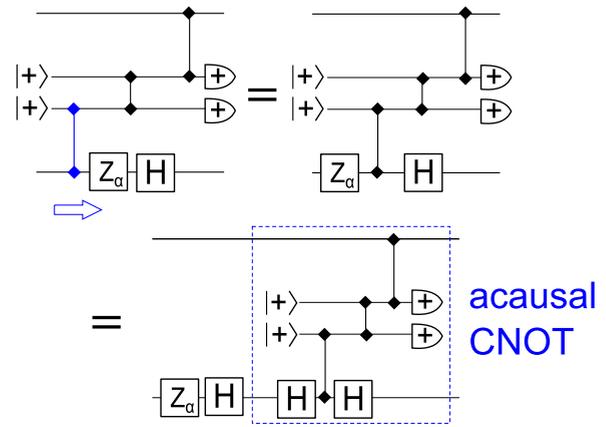


FIG. 21. (Color online) A proof of the circuit identity presented in Fig. 13(b). The J gate, J_α , is decomposed into a phase gate Z_α and Hadamard gate H .

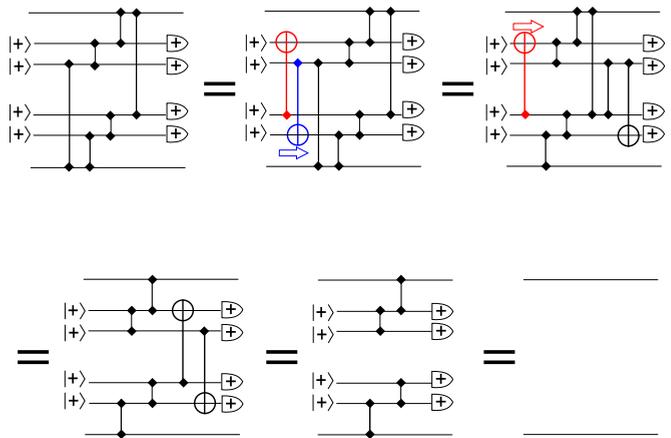


FIG. 22. (Color online) A proof of the circuit identity presented in Fig. 13(d). The first equality is by Eq. (A1) and the last equality is by Eq. (A2). The second and the third equalities are by Eq. (A3). These circuit translations can be carried out even if the acausal CZ gates act on a single wire.

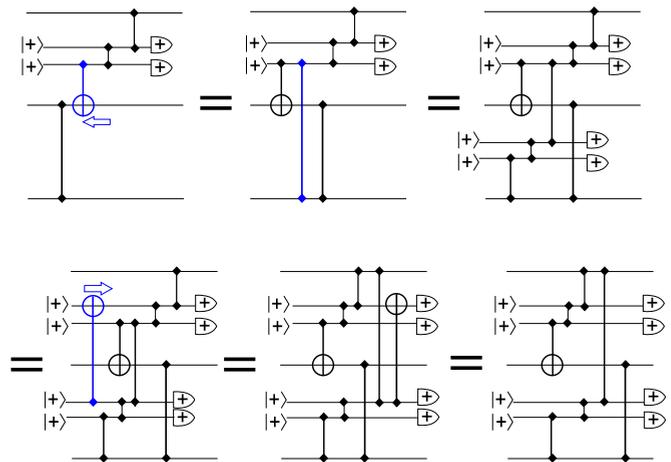


FIG. 23. (Color online) Preparation for a proof of the circuit identity presented in Fig. 13(c). The first and the fourth equalities are by Eq. (A3). The third equality is by Eq. (A1) and the last equality is by Eq. (A2). An ordinary CZ gate is changed into an acausal CZ gate at the second equality.

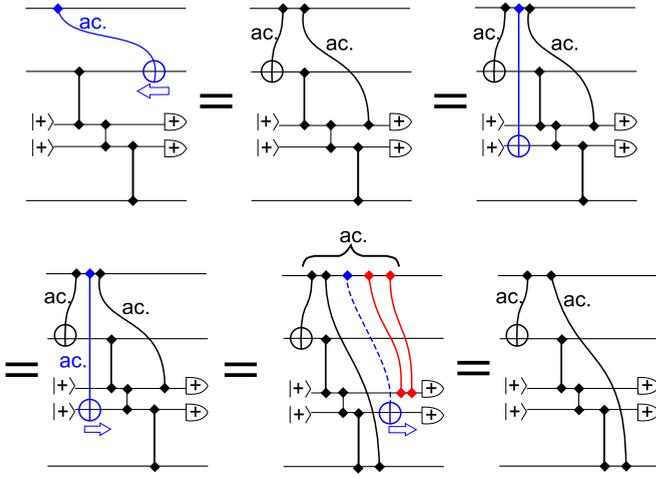


FIG. 24. (Color online) A proof of the circuit identity presented in Fig. 13(c). The first and the fourth equalities are by the circuit identity proved in Fig. 23. The second equality is by Eq. (A1) and the dashed acausal CNOT gate is erased by Eq. (A2) in the last equality. An ordinary CNOT gate is changed into an acausal CNOT gate at the third equality. Two acausal CZ gates appearing on the top right of the second to last circuit are canceled by the circuit identity presented in Fig. 13(d) in the last equality.

First, we can add and remove CNOT gates when the target qubit is prepared in $|+\rangle$ or postselected to the state $|+\rangle$, namely,

$$|+\rangle_B |\text{rest}\rangle_A = CX_{A;B} |+\rangle_B |\text{rest}\rangle_A, \quad (\text{A1})$$

$$\langle +|_B \langle \text{rest}|_A = \langle +|_B \langle \text{rest}|_A CX_{A;B}, \quad (\text{A2})$$

where $|\text{rest}\rangle$ represents the state outside system B but including system A .

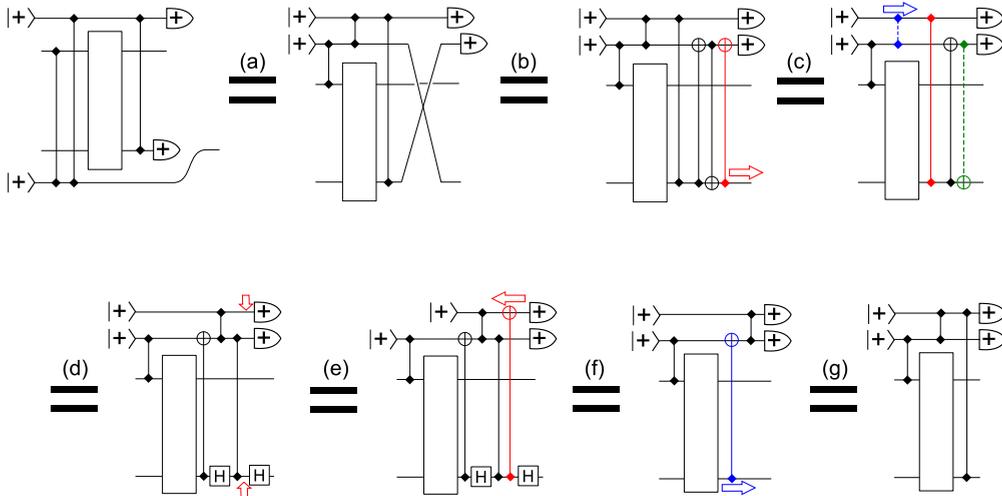


FIG. 25. (Color online) A proof of the equivalence of the circuits presented in Figs. 11 and 12. (a) The ordering of wires is changed. (b) The swap gate is decomposed into a sequence of three CNOT gates. (c) The last CNOT gate is erased by Eq. (A2). (d) Shifting the position of the short CZ gate (represented by a dashed line) forward in time. The longest CZ gate is canceled, when the short one passes through the target part of the CNOT gate. The last CNOT gate (represented by a dashed line) is decomposed into a CZ gate and two Hadamard gates. (e) Inserting a CNOT gate by Eq. (A1) at the positions pointed by the vectors. (f) Shifting the inserted CNOT gate backwards until its target part touches the state $|+\rangle$ and is erased by Eq. (A1). The last CZ gate is canceled when the CNOT gate passes through the short CZ gate. Two Hadamard gates also cancel each other after the CZ gate is canceled. (g) Shifting the position of the CNOT gate until it collides with the postselected state $|+\rangle$ and is erased by Eq. (A2).

Second, we use the following identity relation when shifting positions of CNOT gates through CZ gates:

$$CZ_{A;B} CX_{C;B} = CX_{C;B} CZ_{A;C} CZ_{A;B}. \quad (\text{A3})$$

This identity relation appears when the acausal CNOT gate presented in Fig. 13(c) is replaced by an ordinary CNOT gate.

The circuit presented in Fig. 12 is transformed into our definition of an acausal CZ gate given in Fig. 11. We again use Eqs. (A1) and (A2) and describe the transformation shown in Fig. 25.

APPENDIX B: CONTROLLED STABILIZER

We present a proof of Eq. (14): the controlled operator $\tilde{C}\tilde{K}[g_V(v_i)]_{h(v_i);W(v_i)}$ of $\tilde{K}[g_V(v_i)]$ defined by Eq. (14) also stabilizes the open-graph state $|G'\rangle_\phi$.

First we prove that the operator $\tilde{K}[g_V(v_i)]$ ($v_i \in V_\downarrow^1$) defined by

$$\tilde{K}[g_V(v_i)] = \tilde{X}_{g_V(v_i) \oplus h(v_i)} \tilde{Z}_{\text{Odd}_{G'}[g_V(v_i) \oplus h(v_i)]} \quad (\text{B1})$$

is a stabilizer of $|G'\rangle_\phi$. From the anticommutation relation $XZ + ZX = 0$,

$$(-1)^N \tilde{K}[g_V(v_i)] = \prod_{u \in g_V(v_i) \oplus h(v_i)} \tilde{X}_u \tilde{Z}_{N(u)},$$

where

$$N = |\{(v_1, v_2) \in E | v_1, v_2 \in g(u)\}|$$

holds and the right-hand side is a stabilizer of $|G'\rangle_\phi$. Since there are no edges between any pair of vertices in $g_V(v_i) \oplus h(v_i)$ on G' , $N = 0$ in this case. It follows that $\tilde{K}[g_V(v_i)]$ is a stabilizer of $|G'\rangle_\phi$.

Let us exchange the order of $\tilde{C}\tilde{X}$ included in $\tilde{C}\tilde{K}[g_V(v_i)]_{h(v_i);W(v_i)}$ and $\tilde{C}\tilde{Z}$ constructing the open-graph state,

by adding extra CZ gates in a way presented in Fig. 8(c). For any $\widetilde{CX}_{h(v_i);u} [u \in g_V(v_i) \oplus h(v_i)]$,

$$\begin{aligned}\widetilde{CX}_{h(v_i);u} \widetilde{E}_{G'} &= \widetilde{E}_{G'} \widetilde{CZ}_{h(v_i);N_{G'}(u)} \widetilde{CX}_{h(v_i);u} \\ &= \widetilde{CZ}_{h(v_i);N_{G'}(u)} \widetilde{E}_{G'} \widetilde{CX}_{h(v_i);u}\end{aligned}$$

holds. Summing up for all $u \in g_V(v_i) \oplus h(v_i)$, we have

$$\begin{aligned}\widetilde{CX}_{h(v_i);g_V(v_i) \oplus h(v_i)} \widetilde{E}_{G'} \\ &= \prod_{u \in g_V(v_i) \oplus h(v_i)} \widetilde{CZ}_{h(v_i);N_{G'}(u)} \widetilde{E}_{G'} \widetilde{CX}_{h(v_i);g_V(v_i) \oplus h(v_i)} \\ &= \widetilde{CZ}_{h(v_i);\text{Odd}_{G'}[g_V(v_i) \oplus h(v_i)]} \widetilde{E}_{G'} \widetilde{CX}_{h(v_i);g_V(v_i) \oplus h(v_i)}.\end{aligned}$$

The newly created CZ gates are identical to those appearing in $\widetilde{CK}[g_V(v_i)]_{h(v_i);W(v_i)}$. Thus if we apply $\widetilde{CK}[g_V(v_i)]_{h(v_i);W(v_i)}$ on $|G'\rangle_\phi = E_{G'}|+\rangle_{I^c}|\phi\rangle_I$, it yields

$$\begin{aligned}\widetilde{CK}[g_V(v_i)]_{h(v_i);W(v_i)} \widetilde{E}_{G'}|+\rangle_{I^c}|\phi\rangle_I \\ = \widetilde{E}_{G'} \widetilde{CX}_{h(v_i);g_V(v_i) \oplus h(v_i)}|+\rangle_{I^c}|\phi\rangle_I,\end{aligned}\quad (\text{B2})$$

where CZ gates are canceled. Since $g_V(v_i) \oplus h(v_i) \cap I = \emptyset$, the target side of the CNOT gates in Eq. (B2) all act on $|+\rangle$ states. From Eq. (A1), these CNOT gates are eliminated to derive

$$\widetilde{CK}[g_V(v_i)]_{h(v_i);W(v_i)}|G'\rangle_\phi = |G'\rangle_\phi.$$

-
- [1] R. Raussendorf and H. J. Briegel, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [2] M. Hein, J. Eisert, and H. J. Briegel, *Phys. Rev. A* **69**, 062311 (2004).
- [3] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Van den Nest, and H. J. Briegel, in *Proceedings of the International School of Physics “Enrico Fermi” on “Quantum Computers, Algorithms and Chaos,”* edited by G. Casati *et al.* (IOS, Amsterdam, 2006), Vol. 162.
- [4] R. Cleve and J. Watrous, in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, 2000), p. 526.
- [5] D. E. Browne, E. Kashefi, and S. Perdrix, in *Theory of Quantum Computation: Communication, and Cryptography*, edited by W. van Dam *et al.* (Springer, Berlin Heidelberg, 2011), Vol. 6519, p. 35.
- [6] P. Høyer and R. Špalek, *Theor. Comput.* **1**, 81 (2005).
- [7] R. Raussendorf, D. E. Browne, and H. J. Briegel, *Phys. Rev. A* **68**, 022312 (2003).
- [8] M. Mhalla and S. Perdrix, [arXiv:quant-ph/0412071](https://arxiv.org/abs/quant-ph/0412071).
- [9] V. Danos and E. Kashefi, *Phys. Rev. A* **74**, 052310 (2006).
- [10] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix, *New J. Phys.* **9**, 250 (2007).
- [11] R. Duncan and S. Perdrix, in *Proceedings of the 37th International Colloquium, ICALP 2010, Bordeaux, Part II*, edited by S. Abramsky *et al.*, Lecture Notes in Computer Science Vol. 6199 (Springer, Berlin, 2010), pp. 285–296.
- [12] R. Dias da Silva and Ernesto F. Galvão, *Phys. Rev. A* **88**, 012319 (2013).
- [13] S. Lloyd, L. Maccone, R. Garcia-Patron, V. Giovannetti, and Y. Shikano, *Phys. Rev. D* **84**, 025007 (2011).
- [14] S. Aaronson and J. Watrous, *Proc. R. Soc. A* **465**, 631 (2009).
- [15] N. de Beaudrap, *Phys. Rev. A* **77**, 022328 (2008).
- [16] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)* (Springer, New York, 2005).
- [17] A. Broadbent and E. Kashefi, *Theor. Comput. Sci.* **410**, 26 (2009).
- [18] M. Mhalla and S. Perdrix, in *Automata, Languages and Programming*, edited by L. Aceto *et al.* (Springer, Berlin Heidelberg, 2008), Vol. 5125, p. 857.
- [19] R. Dias da Silva, E. Pius, and E. Kashfi, [arXiv:1301.0351](https://arxiv.org/abs/1301.0351).
- [20] R. Dias da Silva, Ernesto F. Galvão, and E. Kashefi, *Phys. Rev. A* **83**, 012316 (2011).
- [21] C. H. Bennett and B. Schumacher (unpublished). See also <http://web.archive.org/web/20070206131550/http://www.research.ibm.com/people/b/bennetc/QUPONBshort.pdf>
- [22] G. Svetlichny, *Intl. J. Theor. Phys.* **50**, 3903 (2011).
- [23] D. Gottesman and I. L. Chuang, *Nature (London)* **402**, 390 (1999).