

Minimally universal parity quantum computing

Isaac D. Smith^{1,*}, Berend Klaver^{1,2}, Hendrik Poulsen Nautrup¹, Wolfgang Lechner^{1,2,3}, and Hans J. Briegel¹

¹*University of Innsbruck, Department of Theoretical Physics, Technikerstr. 21A, Innsbruck A-6020, Austria*

²*Parity Quantum Computing GmbH, A-6020 Innsbruck, Austria*

³*Parity Quantum Computing Germany GmbH, 20095 Hamburg, Germany*



(Received 22 April 2025; accepted 25 August 2025; published 8 September 2025)

In parity quantum computing, multiqubit logical gates are implemented by single-qubit rotations on a suitably encoded state involving auxiliary qubits. Consequently, there is a correspondence between qubit count and the size of the native gate set. One might then wonder, what is the smallest number of auxiliary qubits that still allows for universal parity computing? Here, we demonstrate that the answer is 1 if the number of logical qubits is even, and 2 otherwise. Furthermore, we present a sufficient condition for a given parity gate set to be universal. This leads to a variety of different universal parity gate sets corresponding to different numbers of auxiliary qubits and more generally contributes to the understanding of which entangling gates are required to augment the set of single-qubit unitaries to perform universal quantum computing. As a consequence, we obtain (i) minimal implementations of the parity framework on, e.g., a triangular lattice, (ii) hardware-specific implementations of the parity flow framework on, e.g., a heavy-hex lattice, and (iii) novel universal resources for measurement-based quantum computation.

DOI: [10.1103/9q8k-5378](https://doi.org/10.1103/9q8k-5378)

I. INTRODUCTION

In the near-term era of quantum computation, the number of physical qubits is low, and the fidelity of, in particular, multiqubit gates is relatively poor. Consequently, much work has gone into developing different gate sets and new techniques in order to reduce the required resources for specific quantum circuits and architectures.

The novelty of the parity quantum computing framework [1,2] consists in the trade-off between physical qubit number and the ease of performing multiqubit rotations. Specifically, a logical state on n qubits is encoded to a new state using k additional qubits, which allows multiqubit logical rotations to be implemented via single-qubit rotations on the latter. In effect, the parity encoding assigns to each auxiliary qubit certain information pertaining to some subset of the logical qubits, which defines the support of the logical rotation. This framework originates in the quantum annealing community, with each $Z \otimes Z$ term of an Ising Hamiltonian being mapped to a separate additional qubit, and has since found application in quantum optimization [3–6].

As each auxiliary qubit, often called a “parity” qubit, corresponds to a specific multiqubit rotation, there is a direct correspondence between the number of qubits and the native gate set implementable in the framework. In Ref. [2], a universal gate set was presented consisting of single-qubit Pauli

X and Z rotations for each of the n logical qubits as well as $Z \otimes Z$ rotations between each pair of logical qubits, which is to say, one parity qubit for every two-qubit rotation involved in the computation. Accordingly, this gate set can be described by a generating set containing $\frac{1}{2}n(n+3)$ Hamiltonians, all of which are single- and two-body Pauli strings.

Recently, it was demonstrated that the minimal possible generating set containing only Pauli strings and that permits universal computation contains just $2n+1$ elements [7]. Due to the correspondence between elements of the generating set and the number of physical qubits in the parity framework, it is prudent to ask how much can we minimize the number of parity qubits while still ensuring universality?

In this work, we answer this question via two main results. In the first, we present a sufficient condition for universality based on the properties of the set of subsets of logical qubits whose parities are mapped to parity qubits by the parity encoding. This condition pertains to both the size of each subset as well as their mutual intersections. As a consequence, we obtain parity encodings for various numbers of auxiliary qubits that all permit universal computation. Included in these encodings are the minimal cases involving a single parity qubit when the number of logical qubits n is even, and two parity qubits when n is odd. The second main result demonstrates the *impossibility* of performing parity quantum computing with less than two parity qubits in the latter case, indicating that, in the parity quantum computing framework, the lower bound of Ref. [7] can only be obtained in the case of n even.

There are a number of implications of these results. Since any generating set that contains a subset that satisfies the conditions outlined above is also universal by default, our results provide a method for demonstrating universality for parity encodings tailored to different physical layouts. For example, we demonstrate below that the generating set sufficient

*Contact author: isaac.smith@uibk.ac.at

conditions can lead to a variety of possible arrangements on a triangular lattice using different numbers of parity qubits. Furthermore, due to the recent development of the parity flow framework [8], which effectively performs parity quantum computation without the use of auxiliary qubits, our results produce a range of options for performing quantum algorithms in a manner suited to specific hardware constraints but without the need for swap gates. In particular, we present a universal circuit ansatz tailored to nearest-neighbor interactions on a heavy-hex layout typical of devices such as those currently provided by IBM Quantum [9]. Finally, as demonstrated in Ref. [10], there is a connection between parity quantum computation and measurement-based quantum computation (MBQC) [11–14], which allows us to leverage the generating set sufficient conditions in defining families of universal resource states [15] for the latter computing framework.

The remainder of this paper is structured as follows: In Sec. II, we present a brief introduction to the relevant information from quantum control theory relevant for understanding universal generating sets for quantum computation as well as the parity quantum computation framework. In Sec. III, we provide our main results, namely Theorems 1 and 2, with the former presenting the sufficient conditions used throughout the rest of this work. In Sec. IV, we investigate some implications of these results for implementing the parity quantum computation and the parity flow frameworks on different two-dimensional layouts of physical qubits. We conclude in Sec. V with a discussion of possible further implications of our results. The proofs of the main results are given in the Appendixes.

II. BACKGROUND

In this work, we are concerned with universal quantum computing within the parity computing framework. Accordingly, we require an understanding of what constitutes universal quantum computing, as well as how this looks in the context of parity quantum computing. The concepts related to the former are drawn primarily from the field of quantum control theory; the reader is directed to, e.g., Refs. [16–19] for further information.

A. Universal quantum computing

For our present purposes, a quantum computer is taken to be a closed, finite-dimensional quantum system with state space described by a Hilbert space $\mathcal{H}$. A quantum computation then consists of evolving an initial state $|\psi_{\text{in}}\rangle \in \mathcal{H}$, which represents the input to the computation, to a final state $|\psi_{\text{out}}\rangle \in \mathcal{H}$, which represents the logical output. Quantum theory tells us that the evolution taking $|\psi_{\text{in}}\rangle$ to $|\psi_{\text{out}}\rangle$ can be described by a (special) unitary U_{comp} such that

$$|\psi_{\text{out}}\rangle = U_{\text{comp}} |\psi_{\text{in}}\rangle. \quad (1)$$

Moreover, we know from Schrödinger's equation that U_{comp} can be specified via reference to a (traceless, time-independent) Hamiltonian H and an evolution time t by

$$U_{\text{comp}} = e^{-iH_{\text{comp}}t}. \quad (2)$$

From the perspective of quantum control theory, the operator H_{comp} is considered to describe how the system is controlled, leading to the evolution described by U_{comp} .

Typically, the overall evolution U_{comp} is constructed from a sequence of component evolutions, that is,

$$U_{\text{comp}} = e^{-iH_{j_k}t_k} \dots e^{-iH_{j_2}t_2} e^{-iH_{j_1}t_1}, \quad (3)$$

where the H_{j_i} are all (traceless, time-independent) Hermitian operators drawn from a fixed set $\{H_j\}_j$ and the t_i are positive real values. The set $\{H_j\}_j$ represents all the different ways the system can be controlled and may be determined by, e.g., specific experimental considerations.

For a given system to act as a *universal* quantum computer, we need to be able to select a sequence of controls, i.e., a sequence of operators from $\{H_j\}_j$ and times for which they are applied, to produce U_{comp} for any pair of states $|\psi_{\text{in}}\rangle$ and $|\psi_{\text{out}}\rangle$. That is, we must be able to solve Eq. (3) for every $U_{\text{comp}} \in \mathfrak{su}(N)$, where $\mathfrak{su}(N)$ denotes the Lie group of $N \times N$ special unitary matrices (N denotes the dimension of the Hilbert space $\mathcal{H}$). Equivalently, we must be able to solve

$$e^{-iH_{\text{comp}}t} = e^{-iH_{j_k}t_k} \dots e^{-iH_{j_2}t_2} e^{-iH_{j_1}t_1} \quad (4)$$

for all $t \in \mathbb{R}_{\geq 0}$ and all $iH_{\text{comp}} \in \mathfrak{su}(N)$, where $\mathfrak{su}(N)$ denotes the Lie algebra of $N \times N$ traceless, skew-Hermitian matrices [20].

The consideration of the Lie algebra $\mathfrak{su}(N)$ (and subalgebras thereof) is common when treating problems within quantum control theory, and several tools from Lie algebra theory will be useful for the questions of universality in which we are interested. In brief, a *Lie algebra* $\mathfrak{g}$ is a (real) subspace of the space of $m \times m$ complex matrices $M_m(\mathbb{C})$ equipped with a Lie bracket $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ that satisfies certain properties (see, e.g., Ref. [20]). In the cases pertinent to this work, the Lie bracket is given by the matrix commutator, that is, for any $A, B \in M_m(\mathbb{C})$,

$$[A, B] := AB - BA. \quad (5)$$

Using the Lie bracket, it is possible to define a so-called adjoint map associated with each element A of the Lie algebra defined as

$$\begin{aligned} \text{ad}_A : \mathfrak{g} &\rightarrow \mathfrak{g} \\ B &\mapsto [A, B]. \end{aligned} \quad (6)$$

Below, it will also be useful to denote the r -fold composition of the adjoint map $\text{ad}_A(\cdot)$ by $\text{ad}_A^{(r)}(\cdot)$, i.e.,

$$\text{ad}_A^{(r)}(B) := \underbrace{[A, [A, [A, \dots [A, B]]]]}_r. \quad (7)$$

Of particular importance for treating expressions such as that appearing on the right-hand side of Eq. (4), is the following formula, which is defined for any (matrix) Lie algebra $\mathfrak{g}$. For any $A, B \in \mathfrak{g}$, the formula states

$$e^A e^B e^{-A} = e^{B + \sum_{r=1}^{\infty} \frac{1}{r!} \text{ad}_A^{(r)}(B)}. \quad (8)$$

To see how this relates to Eq. (4), let us suppose for a moment that U_{comp} can be produced by just two component evolutions, given by applying H_{j_1} for a time of t_1 followed by applying

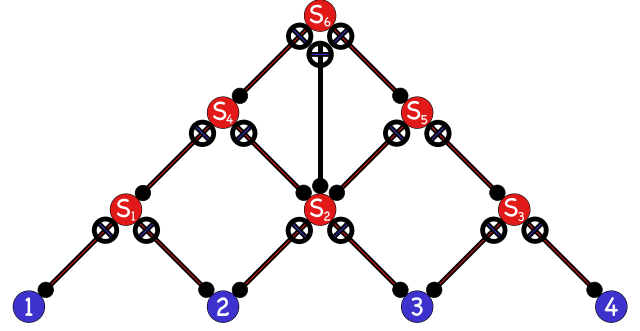
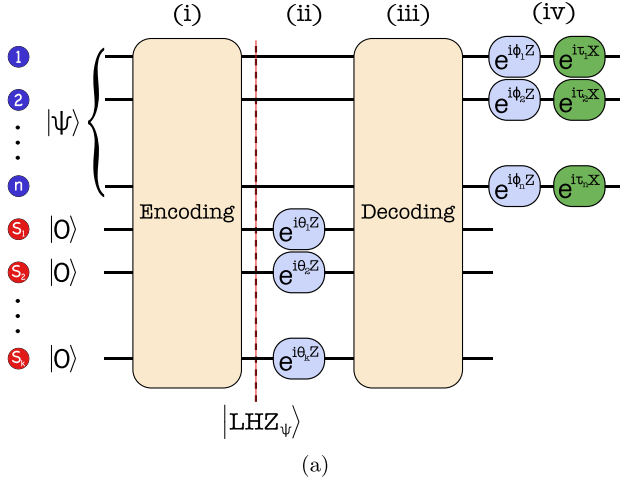


FIG. 1. Parity quantum computing consists of four phases, depicted in panel (a), which are repeated for the duration of the computation. Initially, the state of the n base qubits (in blue) represents the current logical state of the computation $|\psi\rangle$. The k parity qubits (in red) are each initially prepared in the state $|0\rangle$ and are disentangled from the logical qubits. In phase (i), an encoding procedure is applied to the $n + k$ qubits to produce the encoded state denoted $|\text{LH}\psi\rangle$. In phase (ii), single-qubit Z rotations are applied to each of the k parity qubits, and in phase (iii) a decoding procedure is performed through which the base qubits enter a new logical state and the parity qubits return to state $|0\rangle$. In the final phase, further single-qubit rotations are applied, completing the universal gate set. Panel (b) depicts the base and parity qubits for $n = 4$ and $k = 6$ arranged on a square lattice along with a unitary encoding or decoding procedure consisting of a sequence of CNOT gates between nearest neighbors. For the encoding procedure, the CNOTs are applied in order from bottom to top. The parity sets are $S_1 = \{1, 2\}$, $S_2 = \{2, 3\}$, $S_3 = \{3, 4\}$, $S_4 = \{1, 3\}$, $S_5 = \{2, 4\}$, and $S_6 = \{1, 4\}$.

H_{j_2} for a time of t_2 , meaning that

$$e^{-iH_{\text{comp}}t} = e^{-iH_{j_2}t_2} e^{-iH_{j_1}t_1} e^{iH_{j_2}t_2}. \quad (9)$$

By taking $A = -iH_{j_2}t_2$ and $B = -iH_{j_1}t_1$ in Eq. (8) allows us to write

$$-iH_{\text{comp}}t = -iH_{j_1}t_1 + \sum_{r=1}^{\infty} \frac{1}{r!} \text{ad}_{-iH_{j_2}t_2}^{(r)}(-iH_{j_1}t_1). \quad (10)$$

In other words, we are able to write iH_{comp} as a real linear combination of $iH_{j_1}t_1$, $iH_{j_2}t_2$, and sequences of nested commutators between them. For the general case where more than just two controls are used, the situation is analogous, where now the sequences of nested commutators may include more than two distinct elements.

We are thus able to state what we mean by universal quantum computation: the set of controls $\{H_j\}_j$ is *universal* if linear combinations of elements of $\mathcal{G} := \{iH_j\}_j$ as well as nested commutators of elements of $\mathcal{G}$ generate all of $\mathfrak{su}(N)$. Defining

$$\mathcal{G}^{\text{ad}^{(r)}} := \{\text{ad}_{G_1} \cdots \text{ad}_{G_r}(G_{r+1}) : G_1, \dots, G_{r+1} \in \mathcal{G}\}, \quad (11)$$

we can state this more formally as $\{H_j\}_j$ is *universal* if

$$\text{span}_{\mathbb{R}} \left\{ \mathcal{G} \bigcup_{r=1}^{\infty} \mathcal{G}^{\text{ad}^{(r)}} \right\} = \mathfrak{su}(N). \quad (12)$$

Below, we will largely work with the set $\mathcal{G}$ rather than $\{H_j\}_j$ (i.e., with the skew-Hermitian operators rather than the Hermitian ones). The elements of $\mathcal{G}$ are called *generators* and $\mathcal{G}$ itself will be called a *generating set*.

B. Parity quantum computing

Let us turn to the specific quantum computing framework, that of parity quantum computing [1,2,21], the universality of which we are interested in investigating. In particular, we will see that the parity framework comes with a native family of generating sets, which will be the objects under consideration below.

Parity quantum computing typically proceeds by iteratively applying four phases: (i) an encoding phase where the current logical state is embedded in a larger Hilbert space, (ii) the application of physical single-qubit rotations on the extended space that result in logical multiqubit rotations, (iii) a decoding procedure to obtain the new logical state on the original space, and (iv) the application of physical single-qubit rotations that result in logical single-qubit rotations. These four phases are depicted in Fig. 1(a).¹ The generating set associated with this framework, called the *parity generating set* and denoted $\mathcal{G}_{\text{parity}}$ below, features in phases (ii) and (iv); accordingly, we focus on those phases in the following. More information on the parity computing framework can be found in, e.g., Refs. [1,2,21].

Let $|\psi\rangle$ be an n -qubit state representing the current logical state of the computation. At the commencement of phase (i), n physical qubits, called *base qubits* throughout, are in the state $|\psi\rangle$ while k additional physical qubits, called *parity qubits*,

¹It should be noted that there are several proposals for parity quantum computing in which a full decoding is not required in phase (iii), with the logical single-qubit Pauli- X rotations implemented in a nonlocal fashion. As these proposals still use the same logical gate set, the specific details of these different proposals are not relevant here.

are each prepared in the $|0\rangle$ computational basis state. An encoding unitary U_{enc} is then applied to all $n + k$ qubits to produce the state

$$|\text{LHZ}_\psi\rangle = U_{\text{enc}} |0\rangle^{\otimes k} |\psi\rangle. \quad (13)$$

The encoding unitary U_{enc} consists of a product of CNOT gates between base and parity qubits, arranged in such a way that the k parity qubits encode parity information of $|\psi\rangle$ (whence the name parity qubits). For example, for $n = 2$ and $k = 1$, the state

$$|\psi\rangle = \sum_{i,j=0}^1 \alpha_{ij} |ij\rangle \quad (14)$$

is mapped to the state

$$|\text{LHZ}_\psi\rangle = \sum_{i,j=0}^1 \alpha_{i,j} |i \oplus j\rangle |ij\rangle \quad (15)$$

by the unitary U_{enc} consisting of the product of two CNOT gates each with one of the base qubits as control and the parity qubit as target (the notation “ $\oplus$ ” denotes modulo 2 addition). For larger values of n and k , U_{enc} can be defined similarly; see, e.g., Fig. 1(b) for a depiction of the encoding for $n = 4$ and $k = 6$. We would like to emphasize that the Hamiltonian corresponding to the unitary U_{enc} is *not* included in the generating set considered below because it does not directly enact a logical operation but rather facilitates the implementation of the logical multiqubit rotations via single-qubit rotations on the encoded parity qubits.

For our purposes, a useful perspective of the encoded state $|\text{LHZ}_\psi\rangle$ arises from stabilizer theory. Let us identify each of the n base qubits with a label $i \in \{1, \dots, n\}$ and each of the k parity qubits with a label given by a (nonempty) set $S_j \subseteq \{1, \dots, n\}$, $j = 1, \dots, k$. This labeling convention also extends to unitary operations performed on data and parity qubits; that is, Z_i denotes the single-qubit Pauli-Z rotation on base qubit i while Z_{S_j} denotes a single-qubit Pauli-Z rotation on the parity qubit labeled by S_j .² The idea is that a parity qubit labeled by S_j encodes parity information of the base qubits whose labels are contained in the set S_j . Let us define $\mathbb{P} := \{S_j : j = 1, \dots, k\}$. Then, for any $|\psi\rangle$, the state $|\text{LHZ}_\psi\rangle$ satisfies the following equation for each $S_j \in \mathbb{P}$:

$$Z_{S_j} \bigotimes_{i \in S_j} Z_i |\text{LHZ}_\psi\rangle = |\text{LHZ}_\psi\rangle. \quad (16)$$

where Z denotes the Pauli-Z operator (we denote the Pauli-X and Pauli-Y operators similarly by X and Y), the label S_j indicates the relevant parity qubit, and the labels $i \in S_j$ indicate the relevant base qubits. As a consequence of this, we have that

$$e^{-iZ_{S_j}t} |\text{LHZ}_\psi\rangle = e^{-i(\bigotimes_{i \in S_j} Z_i)t} |\text{LHZ}_\psi\rangle. \quad (17)$$

This means that, in effect, single-qubit Z-rotations performed on the parity qubits in the encoded state enact a multiqubit

rotation on the logical state. These rotations are precisely the operations pertaining to phase (ii) of the computational process. Regarding the generating sets we ultimately aim to consider, let us define

$$\mathcal{G}_{\mathbb{P}} := \left\{ i \bigotimes_{q \in S_j} Z_q | S_j \in \mathbb{P} \right\}. \quad (18)$$

Phase (iii) of the computation consists of the decoding phase where the logical state is mapped back to the n base qubits. There are a couple of equivalent ways in which this decoding procedure can occur: for example, one can apply the decoding unitary $U_{\text{dec}} = U_{\text{enc}}^\dagger$ as done in the original proposal for universal parity quantum computing [2]; otherwise decoding can proceed by measuring each parity qubit in the Pauli X-basis and performing suitable corrections for certain measurement outcomes as suggested in Ref. [22]. The latter procedure has close links to measurement-based quantum computation (MBQC) [11–14] as was shown in Ref. [10] (see Appendix D for a brief introduction to MBQC and an overview of the correspondence between the two frameworks). As the decoding phase has no bearing on the generating sets for the computation, we need not specify a particular decoding method here.

In the final phase, phase (iv), single-qubit rotations are applied to each of the base qubits, which are now disentangled from the parity qubits after the decoding step in phase (iii). Explicitly, we may apply $e^{-iX_j t}$ and $e^{-iZ_j t'}$ for each $j \in \{1, \dots, n\}$ and any $t, t' \in \mathbb{R}_{\geq 0}$. Accordingly, we define the generating set pertaining to this phase as

$$\mathcal{G}_{\text{s.q.}} := \{iX_j, iZ_j | j = 1, \dots, n\}, \quad (19)$$

where the subscript “s.q.” stands for “single-qubit.” Thus, the entire parity generating set is

$$\mathcal{G}_{\text{parity}} = \mathcal{G}_{\text{s.q.}} \cup \mathcal{G}_{\mathbb{P}}. \quad (20)$$

In what follows, we will always consider $\mathcal{G}_{\text{parity}}$ to contain $\mathcal{G}_{\text{s.q.}}$ for the number of base qubits n , and investigate the consequences for universality as the set $\mathcal{G}_{\mathbb{P}}$ (equivalently $\mathbb{P}$) varies. In the original demonstration of the universality of the parity computing framework [2], the set $\mathbb{P}$ contained all pairs of labels from $\{1, \dots, n\}$, which we denote for later reference by

$$\mathbb{P}_{\text{pairs}} := \{\{i, j\} | i, j = 1, \dots, n \text{ s.t. } i < j\}. \quad (21)$$

III. MINIMAL UNIVERSAL PARITY GENERATING SETS

Using the notation introduced above, we can now state the question we are interested in: for which $\mathbb{P}$ is $|\mathbb{P}|$ minimized while maintaining that

$$\text{span}_{\mathbb{R}} \left\{ \mathcal{G}_{\text{parity}} \bigcup_{r=1}^{\infty} \mathcal{G}_{\text{parity}}^{\text{ad}^{(r)}} \right\} = \mathfrak{su}(2^n)? \quad (22)$$

There are a couple of features of the generating set $\mathcal{G}_{\text{parity}}$ that are pertinent to the above question. First, this generating set consists entirely of Pauli strings, i.e., contains only terms consisting of products of Pauli operators. That is, defining

$$\mathcal{P}_n := \{P_1 \otimes P_2 \otimes \dots \otimes P_n | P_i \in \{I, X, Y, Z\}\}, \quad (23)$$

²Note that, in the literature, the notation of a unitary indexed by a set is defined to be the tensor product of many copies of that unitary, one for each element in the set. We do not employ this notation here.

we have that $\mathcal{G}_{\text{parity}} \subset i\mathcal{P}_n$. We use the notation $i\mathcal{P}_n$ to denote the set obtained by multiplying every element of $\mathcal{P}_n$ by the imaginary unit i (note also that $\mathcal{P}_n$ is the set of Pauli strings, not the Pauli group on n qubits). The inclusion of $\mathcal{G}_{\text{parity}}$ in $i\mathcal{P}_n$ is significant in part due to the fact that elements of $i\mathcal{P}_n$ exhibit nice commutation relations: any two elements of $i\mathcal{P}_n$ either commute or anticommute, with the latter operator being proportional to an element of $i\mathcal{P}$. Noting that $i\mathcal{P}_n^* := i\mathcal{P}_n \setminus \{iI^{\otimes n}\}$ forms a basis of the space of traceless, skew-Hermitian operators, it follows that, to show that Eq. (22) holds, it suffices in this case to demonstrate that³

$$\mathcal{G}_{\text{parity}} \bigcup_{r=1}^{\infty} \mathcal{G}_{\text{parity}}^{\text{ad}^{(r)}} = i\mathcal{P}_n^*. \quad (24)$$

A second noteworthy feature of $\mathcal{G}_{\text{parity}}$ pertains to the commutation relations present in the subset $\mathcal{G}_{\text{s.q.}}$. In particular, each element of $\mathcal{G}_{\text{s.q.}}$ anticommutes with precisely one other element: iX_j anticommutes with $iZ_{j'}$ if and only if $j = j'$, while iX_j and $iX_{j'}$ (respectively iZ_j and $iZ_{j'}$) commute for all $j, j' \in \{1, \dots, n\}$. Since we ultimately consider sequences of nested commutations between elements in $\mathcal{G}_{\text{s.q.}}$ (as well as $\mathcal{G}_{\mathbb{P}}$), these relations play a role in establishing the results presented below.

To begin to understand the possible limits on the minimal choices of $\mathbb{P}$, let us consider what is already known regarding generating sets consisting of Pauli strings. In Ref. [7], it was demonstrated that any universal generating set $\mathcal{G} \subset i\mathcal{P}_n^*$ is such that $|\mathcal{G}| \geq 2n + 1$ and moreover that generating sets exist that obtain this bound.⁴ Since

$$|\mathcal{G}_{\text{parity}}| = |\mathcal{G}_{\text{s.q.}}| + |\mathbb{P}| \quad (25)$$

and $|\mathcal{G}_{\text{s.q.}}| = 2n$, there is, at least in principle, nothing preventing the possibility of choosing $\mathbb{P}$ such that $|\mathbb{P}| = 1$. The results presented in the remainder of this section demonstrate when this is and is not possible.

Our first result provides a sufficient condition on $\mathbb{P}$ that ensures the universality of $\mathcal{G}_{\text{parity}}$:

Theorem 1. Let $n \geq 2$ and $\mathcal{G}_{\text{s.q.}}$ be as above. For any $1 \leq k \leq n - 1$ and sets $S_1, \dots, S_k$ such that

- (1) $|S_j|$ is even for all $j \in \{1, \dots, k\}$,
- (2) $\bigcup_{j=1}^k S_j = \{1, \dots, n\}$,
- (3) if $k \geq 2$, then
 - (i) $S_i \cap S_j = \emptyset$ for all $1 \leq j \leq k$ such that $j \neq i - 1, i, i + 1$, and
 - (ii) $S_i \cap S_{i+1} = \{s_i\}$ for all $i \leq k - 1$, with $s_i \in \{1, \dots, n\}$ all distinct.

³To be fully precise, the following notion of equality should be considered element-wise equality of the two sets up to a real constant. That is, all the elements in the left-hand set will be of the form $i2^r P$ while those in the right-hand set are of the form iP . This real constant is unimportant in light of the real span taken in Eq. (22), so we continue with the abuse of notation here.

⁴This conclusion can also be drawn from results presented in Ref. [23].

For $\mathbb{P} = \{S_j | j = 1, \dots, k\}$ and $\mathcal{G}_{\text{parity}} = \mathcal{G}_{\text{s.q.}} \cup \mathcal{G}_{\mathbb{P}}$, we have that

$$\mathcal{G}_{\text{parity}} \bigcup_{r=1}^{\infty} \mathcal{G}_{\text{parity}}^{\text{ad}^{(r)}} = i\mathcal{P}_n^*. \quad (26)$$

The proof is given in Appendix B. In particular, it makes use of a mapping between $i\mathcal{P}_n^*$ and the symplectic space $\mathbb{F}_2^{2N}$ which is presented in Appendix A.

There are a number of consequences of this result for the parity computing framework. First of all, for any even n , taking $k = 1$ and $S = \{1, \dots, n\}$ satisfies the conditions of the theorem. Similarly, for odd n , it is possible to take $k = 2$ and define S_1 and S_2 that satisfy the required conditions (for example, taking $S_1 = \{1, \dots, n - 1\}$ and $S_2 = \{n - 1, n\}$ suffices). Accordingly, we have the following corollary:

Corollary 1. For $n \geq 2$, universal parity quantum computing is possible with just

- (1) a single parity qubit, if n is even, and
- (2) two parity qubits, if n is odd.

In the case of even n , the generating set $\mathcal{G}_{\text{parity}}$ has $2n + 1$ elements, so by the results of Ref. [7] it is provably minimal. For the odd- n case, *a priori* there is still a chance that we may be able to find a single element of $i\mathcal{P}_n^*$ which extends $\mathcal{G}_{\text{s.q.}}$ to a universal generating set. The following result demonstrates that this is not the case:

Theorem 2. Let $n \geq 2$ be odd and $\mathcal{G}_{\text{s.q.}}$ be as above. For any $iP \in i\mathcal{P}_n^*$, $\mathcal{G} := \mathcal{G}_{\text{s.q.}} \cup \{iP\}$ is such that

$$\mathcal{G} \bigcup_{r=1}^{\infty} \mathcal{G}^{\text{ad}^{(r)}} \subsetneq i\mathcal{P}_n^*. \quad (27)$$

The proof is given in Appendix C and again makes use of the mapping mentioned above. This result demonstrates two things. First, it completes the claim made earlier that for n even and odd, the minimal number of parity qubits required for universal computation are 1 and 2, respectively. In the former case, the resultant generating set has size $2n + 1$, which we know is optimal, whereas the requirement of at least $2n + 2$ generators in the latter case demonstrates that not every generating set of $2n$ Pauli strings can be extended to a universal set by simply appending a single additional Pauli string.

There are two final observations worth making here. The first is that, for any $\mathbb{P}$ that contains a subset of sets that satisfies the conditions of Theorem 1, the generating set $\mathcal{G}_{\text{s.q.}} \cup \mathcal{G}_{\mathbb{P}}$ is universal. For example, if we consider the set $\mathbb{P}_{\text{pairs}}$ that was used in the original proof of universality of the parity framework [2], we see that it contains the sets

$$\{\{i, i + 1\} | i = 1, \dots, n - 1\}, \quad (28)$$

which satisfy the conditions of Theorem 1. This provides an alternative proof of the universality of $\mathcal{G}_{\text{s.q.}} \cup \mathcal{G}_{\mathbb{P}_{\text{pairs}}}$ to that given in Ref. [2] and also demonstrates that, from the point of view of universality, there is redundancy in the set of parities $\mathbb{P}_{\text{pairs}}$.

The second observation is that, even though there are sets $\mathbb{P}$ satisfying the conditions of Theorem 1 which correspond to generating sets with greater than $2n + 1$ elements, they can still be considered to be minimal from a certain perspective. Specifically, they are minimal in the sense that by removing

any single element of $\mathbb{P}$, the generating set then fails to be universal. Again this contrasts with the case of $\mathbb{P}_{\text{pairs}}$ from which it is possible to omit elements and still maintain universality.

IV. IMPLICATIONS

The parity sets that satisfy the sufficient conditions of Theorem 1 are distinct from those considered to date in the literature. In this section we elucidate certain implications of using such parity sets within the parity quantum computing framework. In particular, we first demonstrate that, for the generating sets $\mathcal{G}_{\text{parity}}$ where $|\mathbb{P}| = 1$ if n is even and $|\mathbb{P}| = 2$ if n is odd, it is possible to implement the rotation of *any* n -qubit Pauli string in constant depth. Thereafter, we focus on the ability to perform parity quantum computing using nearest-neighbor interactions between physical qubits arranged on a triangular lattice and on the ability to natively implement the recently developed parity flow framework [8] suited to the specific connectivity of current quantum devices, such as those provided by IBM Quantum [9]. Finally, we briefly outline the implications for universal measurement-based quantum computing.

A. Implementing Pauli string rotations in constant depth

One pertinent question to ask when considering the universal generating sets related to Theorem 1 is how well they scale in terms of compiling specific unitaries. In this subsection, we provide a partial answer to this question for the minimal possible universal sets, that is, for the minimal sizes of $\mathbb{P}$.

Consider the set $\mathcal{G}_{\text{parity}}$ with

$$\mathbb{P} = \begin{cases} \{iZ_1 \otimes \cdots \otimes Z_n\}, & n = 0 \bmod 2 \\ \{iZ_1 \otimes \cdots \otimes Z_j, iZ_j \otimes \cdots \otimes Z_n\}, & n = 1 \bmod 2 \end{cases} \quad (29)$$

for some even number $j \in \{2, \dots, n-1\}$. As a consequence of the results presented in Sec. III, we know that, for any even $n \in \mathbb{N}_{\geq 2}$, $\mathcal{G}_{\text{parity}}$ defined in this way is universal. Apart from the dependence on n being even or odd, the *size* of $\mathbb{P}$ does not depend on n . This is in contrast with other choices of parity set such as $\{iZ_j \otimes Z_{j+1} | j = 1, \dots, n-1\}$. This lack of independence of $|\mathbb{P}|$ on n may seem innocuous, but it has an interesting consequence for implementing unitaries of the form $e^{i\theta P}$ for $P \in \mathcal{P}_n^*$. Specifically, *any* such rotation can be implemented in constant depth using the generating set $\mathcal{G}_{\text{parity}}$ with $\mathbb{P}$ as above.

This claim follows from a part of the proof of Theorem 1 (namely Lemma B2), which we elaborate upon below. Before doing so, let us comment on what “depth” means in this context. Clearly, as n increases, the size of $\mathcal{G}_{\text{s.q.}}$ does also. Accordingly, the value $R < \infty$ for which

$$\mathcal{G}_{\text{parity}} \bigcup_{r=1}^R \mathcal{G}_{\text{parity}}^{\text{ad}^{(r)}} = i\mathcal{P}_n^* \quad (30)$$

will increase as n increases. The value R corresponds to the minimum value such that there always exists a sequence $G_1, \dots, G_{R'} \in \mathcal{G}_{\text{parity}}$ with $R' \leq R$ such that

$$\text{ad}_{G_1} \cdots \text{ad}_{G_{R'-1}}(G_{R'}) = \frac{1}{2^{R'-1}} iP \quad (31)$$

for each $iP \in i\mathcal{P}_n^*$. However, what this value R *does not* take into account is that most of the elements G_r in the sequence will come from $\mathcal{G}_{\text{s.q.}}$, that is, they are single-qubit operators and hence can be implemented in parallel in a circuit. In light of this, we can define the *sequence depth* of implementing iP as the minimum number of multiqubit Pauli-string rotations from the parity set $\mathcal{G}_{\text{P}}$ required to generate iP by nested commutation. Let us consider the corresponding quantity

$$\text{seq-depth}(iP) := \min_{\substack{G_1, \dots, G_{R'} \\ \text{s.t. (31) holds}}} |\{G_r \in \mathcal{G}_{\text{P}} | r = 1, \dots, R'\}|. \quad (32)$$

We have the following:

Proposition 1. Let $\mathcal{G}_{\text{parity}}$ be as in Eq. (29). For any $iP \in i\mathcal{P}_n^*$,

$$\text{seq-depth}(iP) \leq \begin{cases} 3, & n = 0 \bmod 2 \\ 6, & n = 1 \bmod 2. \end{cases} \quad (33)$$

The proof follows as a corollary of Lemma B2. The proof of Lemma B2 is constructive in the sense that, for each $iP \in i\mathcal{P}_n^*$, it produces $G_1, \dots, G_{R'} \in \mathcal{G}_{\text{parity}}$ that satisfy Eq. (31). For the even- n case, these sequences are given explicitly below.

Before giving the sequences, let us consider the question: how does one actually go from such a sequence to the circuit implementing $e^{i\theta P}$? One possible answer is given by Eq. (8) presented earlier. As demonstrated in, e.g., Ref. [7], if the sequence $G_1, \dots, G_{R'}$ satisfies Eq. (31) then

$$e^{\frac{\pi}{4} G_1} \cdots e^{\frac{\pi}{4} G_{R'-1}} e^{\theta G_{R'}} e^{-\frac{\pi}{4} G_{R'-1}} \cdots e^{-\frac{\pi}{4} G_1} = e^{i\theta P}. \quad (34)$$

Pursuant to Proposition 1, even though this product contains $2R' - 1$ terms, and R' grows with n , at most 5 (11) of them are entangling unitaries for all even (odd) n . Thus, the implementation of $e^{i\theta P}$ is constant also in circuit depth, if each entangling gate and each parallel implementation of single-qubit gates are considered to have constant depth.⁵

The sequences for a given $iP \in i\mathcal{P}_n^*$ have a different structure depending on whether n is even or odd and on whether the Pauli string $P = P_1 \otimes \cdots \otimes P_n$ has an even or odd number of tensor factors. We focus on the even- n case here; the odd case can be obtained by applying Lemma B2 twice. Let us write $\mathbf{Z} := Z_1 \otimes \cdots \otimes Z_n$ for the single-parity operator. Let us write $\{j_1, \dots, j_l\} \subseteq \{1, \dots, n\}$ for the subset of labels j for which $P_j \neq I$. Similarly, we write $\{k_1, \dots, k_{n-l}\} := \{1, \dots, n\} \setminus \{j_1, \dots, j_l\}$. If l is odd, then the sequence

$$G'_1, \dots, G'_{l+2} = i\mathbf{Z}, \quad iX_{j_1}, \dots, iX_{j_l}, i\mathbf{Z} \quad (35)$$

is such that

$$\text{ad}_{G'_1} \cdots \text{ad}_{G'_{l+1}}(G'_{l+2}) \propto iP'_1 \otimes \cdots \otimes P'_n, \quad (36)$$

where $P'_j = X_j Z_j$ for each $j \in \text{supp}(P)$ and $P'_j = I$ otherwise, that is, the Pauli string on the right-hand side has the same support as P . Accordingly, there exists a sequence $G''_1, \dots, G''_l$ consisting only of elements of $\mathcal{G}_{\text{s.q.}}$ such that

$$\text{ad}_{G''_1} \cdots \text{ad}_{G''_l}(iP'_1 \otimes \cdots \otimes P'_n) \propto iP. \quad (37)$$

⁵There exist measurement-based implementations of all the entangling gates considered here that have constant depth (see, e.g., Ref. [24]).

The sequence $G_1, \dots, G_r$ is then taken to be $G'_1, \dots, G'_l, G'_1, \dots, G'_{l+2}$.

In the case, where l is even, the sequence

$$\begin{aligned} G'_1, \dots, G'_{3n-3l+5} = & iX_{j_1}, iZ, iX_{k_1}, \dots, iX_{k_{n-l}}, \\ & iZ_{k_1}, \dots, iZ_{k_{n-l}}, iZ, iX_{j_1}, \\ & iX_{k_1}, \dots, iX_{k_{n-l}}, iZ \end{aligned} \quad (38)$$

is such that

$$\text{ad}_{G'_1} \cdots \text{ad}_{G'_{3n-3l+4}} (G'_{3n-3l+5}) \propto i\hat{P}_1 \otimes \cdots \otimes \hat{P}_n, \quad (39)$$

where $\hat{P}_j = Z_j$ if $j \in \text{supp}(P)$ and $\hat{P}_j = I$ otherwise. Similarly to above, it means that $\hat{P} = \hat{P}_1 \otimes \cdots \otimes \hat{P}_n$ has the same support as P , so there again exist $G''_1, \dots, G''_l \in \mathcal{G}_{\text{s.q.}}$ such that

$$\text{ad}_{G''_1} \cdots \text{ad}_{G''_l} (i\hat{P}_1 \otimes \cdots \otimes \hat{P}_n) \propto iP. \quad (40)$$

The sequence $G_1, \dots, G_r$ is again taken to be the concatenation of the two sequences.

If we substitute these sequences into Eq. (34) and abbreviate notation to highlight the splitting into local Clifford rotations and global rotations, we get, in the case where P has even support,

$$e^{i\theta P} = U_{\text{l.c.}} e^{i\frac{\pi}{4}Z} V_{\text{l.c.}} e^{i\theta Z} V_{\text{l.c.}}^\dagger e^{-i\frac{\pi}{4}Z} U_{\text{l.c.}}^\dagger, \quad (41)$$

where $U_{\text{l.c.}}$ and $V_{\text{l.c.}}$ are products of local Clifford operations. In the case where P has odd support, we get that $e^{i\theta P}$ is produced by the product of unitaries

$$\hat{U}_{\text{l.c.}} e^{i\frac{\pi}{4}Z} \hat{V}_{\text{l.c.}} e^{i\frac{\pi}{4}Z} \hat{W}_{\text{l.c.}} e^{i\theta Z} \hat{W}_{\text{l.c.}}^\dagger e^{-i\frac{\pi}{4}Z} V_{\text{l.c.}}^\dagger e^{-i\frac{\pi}{4}Z} U_{\text{l.c.}}^\dagger, \quad (42)$$

where $\hat{U}_{\text{l.c.}}$, $\hat{V}_{\text{l.c.}}$, and $\hat{W}_{\text{l.c.}}$ are also all products of local Clifford operations. The analogous sequences for the case where n is odd are longer, which is due to the fact that, in such cases, $\mathbb{P}$ contains two elements (this is the source of the factor of 2 difference between the sequence depths in Proposition 1).

B. Arrangement on a triangular lattice

One advantage of the parity framework is that long-range multiqubit logical operations are performed by single-qubit physical rotations on an encoded state, which was moreover prepared using nearest-neighbor interactions. For example, when computing using the layout depicted in Fig. 1(b), it is possible to implement the logical operation $R_{Z_1 \otimes Z_4}(\theta)$ on the non-neighboring qubits 1 and 4, by performing the rotation $R_{Z_{\{1,4\}}}(\theta)$ on the parity qubit related to the set $\{1, 4\}$. As the figure demonstrates, the encoded state permitting this can be prepared by arranging the physical base and parity qubits on a square lattice and then by applying a sequence of CNOT gates between nearest neighbors.

Here, we consider what conclusions can be drawn from Theorem 1 for designing possible layouts for the parity framework. In particular, we focus on arranging the physical base and parity qubits on a triangular lattice, while still requiring that the encoding unitary U_{enc} be implemented via nearest-neighbor CNOT gates. In Fig. 2, we portray several examples of possible layouts for $n = 10$ base qubits corresponding to different choices of the sets $S_1, \dots, S_k$ from Theorem 1.

There are a couple of benefits of performing parity computing using the layouts presented here. First, the layout in, e.g.,

Fig. 2(a) can be extended to any number of base qubits and in each case the corresponding unitary encoding has constant circuit depth. This contrasts with the circuit depth for the encoding unitary for the layouts of the form depicted in Fig. 1(b), corresponding to the parity set $\mathbb{P}_{\text{pairs}}$, which scales with n (although a constant-depth encoding using measurements does exist [22]).

Second, the layouts in Fig. 2 may be advantageous for physical device design in certain architectures. For example, in each of the layouts depicted, the topology ensures that it is possible to draw a path from the perimeter of the lattice to every two-qubit gate without crossing any other two-qubit gate. In the context of, e.g., superconducting qubits, this could allow for chip design where no reliance on so-called air-bridge crossovers [25,26] for the required control lines for the implementation of two-qubit gates, thereby simplifying the fabrication process. Moreover, in such a context it is often common to implement two-qubit gates between a control qubit with high frequency and a target qubit with low frequency [27,28]. Since the layouts in Fig. 2 all have fixed orientations of CNOTs (certain qubits are only ever targets and others are only ever controls), this suggests a natural distribution of high- and low-frequency qubits throughout the chip.

C. Minimal parity flow

So far, we have presented the parity computing framework as encoding an n -qubit state as an $(n+k)$ -qubit state. As developed in the parity flow formalism of Ref. [8], it is in fact possible to perform parity quantum computing on n qubits by appropriately tracking how the parity information changes “in place,” that is, by allowing a given physical qubit to act as a parity qubit at various times throughout the circuit.

Let us consider an explanatory example. Recall from above that, for the case $n = 2$ and $k = 1$, the encoded state for the parity computation with logical state $|\psi\rangle = \sum_{i,j=0}^1 \alpha_{ij} |i\rangle |j\rangle$ is

$$|\text{LHZ}_\psi\rangle = \sum_{i,j=0}^1 \alpha_{ij} |i \oplus j\rangle |i\rangle |j\rangle. \quad (43)$$

By applying $R_Z(\theta)$ to the parity qubit and then decoding (either unitarily or via measurements), we obtain the logical state $R_{Z \otimes Z}(\theta) |\psi\rangle$. Equivalently, we could avoid the use of the parity qubit altogether by considering instead the state

$$(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X) |\psi\rangle = \sum_{i,j=0}^1 \alpha_{ij} |i\rangle |i \oplus j\rangle \quad (44)$$

instead of $|\text{LHZ}_\psi\rangle$. In this case, we can arrive at the same final logical state by applying the rotation $I \otimes R_Z(\theta)$ followed by the CNOT gate $(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X)$.⁶ By inspecting Eqs. (43) and (44), we see that the parity information that is mapped to the auxiliary qubit in the former case is encoded in the second base qubit in the latter. In Ref. [8], this line

⁶This example, as well as its extension of implementing a $Z \otimes Z \otimes \cdots \otimes Z$ rotation by a sequence of CNOTs and a single Z rotation has appeared in a number of places in the literature; an inexhaustive list includes the Refs. [24,29–32].

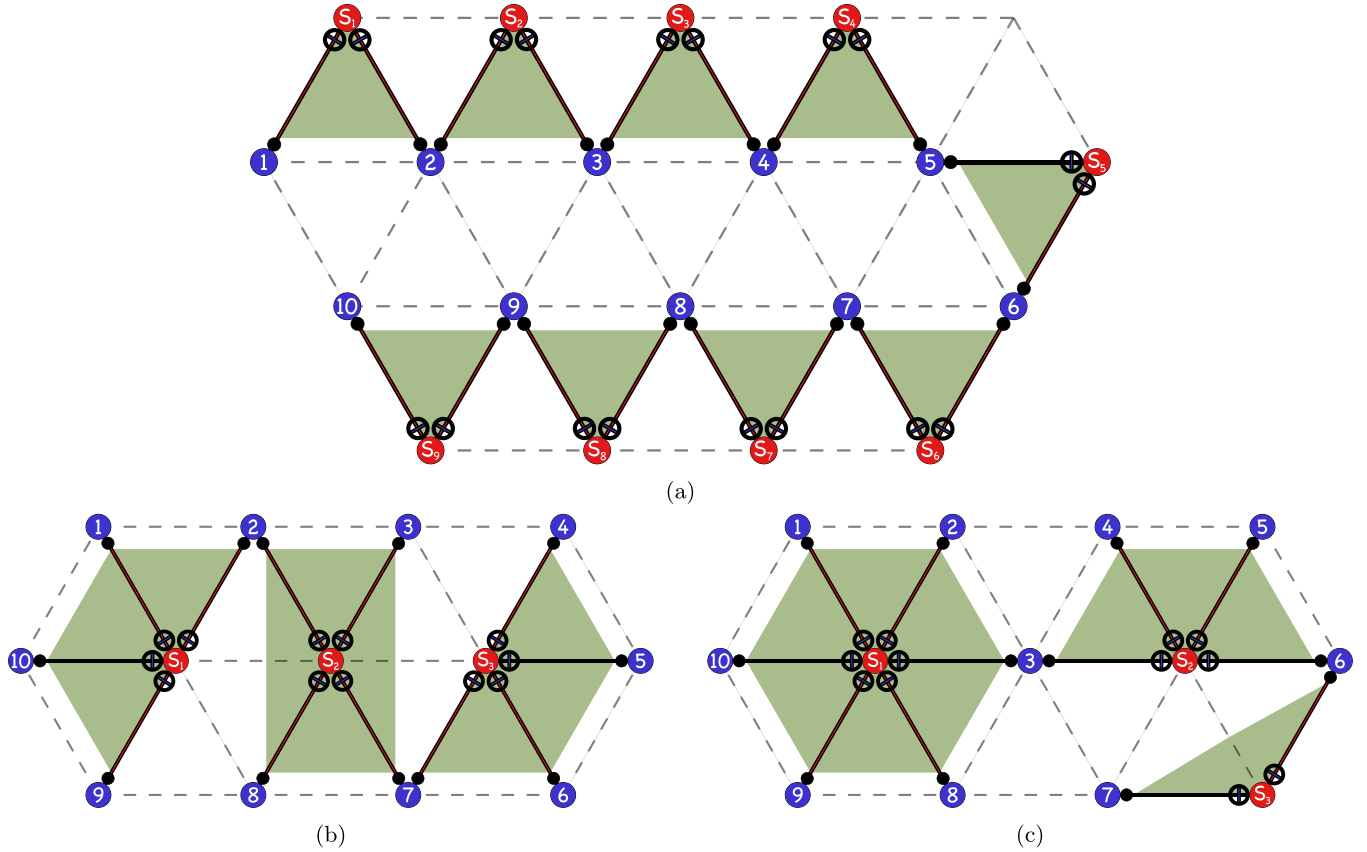


FIG. 2. This figure depicts several possibilities for performing universal parity quantum computing using 10 base qubits and varying numbers of parity qubits arranged on a triangular lattice. In each case, the base qubits are shown in blue, the parity qubits in red and the green shaded indicate which base qubits pertain to each parity set, i.e., each blue base qubit touching a corner of a green shape indicates that that qubit features in the parity set for that shape. For example, in panel (a) all the sets S_i have two parities, namely $S_i = \{i, i + 1\}$; in panel (b), each parity set has four elements with $S_1 = \{1, 2, 9, 10\}$, $S_2 = \{2, 3, 7, 8\}$, and $S_3 = \{4, 5, 6, 7\}$; in panel (c), the parity sets each have differing numbers of elements with $S_1 = \{1, 2, 3, 8, 9, 10\}$, $S_2 = \{3, 4, 5, 6\}$, and $S_3 = \{6, 7\}$. Each of these choices of sets satisfies the conditions of Theorem 1, and hence each layout supports universal parity quantum computing.

of reasoning is extended to the full parity encoding [i.e., for $\mathbb{P} = \mathbb{P}_{\text{pairs}}$ as in Fig. 1(b)], where the resource benefits for implementing certain algorithms were also shown. Figure 3(a) depicts an example of how the parity information changes for the parity computation using the parity encoding shown in Fig. 1(b) using $n = 4$ physical qubits.

We know from Theorem 1 above that smaller parity sets than $\mathbb{P}_{\text{pairs}}$ still allow for universal computation. In the formalism where parity sets are mapped to additional physical qubits, this resulted in a lower total qubit count. But what does it mean in the parity flow formalism where n qubits are used in any case? In essence, different choices for $\mathbb{P}$ correspond to different choices of the number and orientation of the CNOTs comprising the unitaries that transport the parity information (i.e., the unitaries between the filled circles in the circuits depicted in Fig. 3). For example, Figs. 3(b) and 3(c) depict the unitaries and corresponding flow of parity information for the minimal parity set examples for $n = 4$ and $n = 5$ qubits, respectively. As can be seen, there is a reduced CNOT cost for implementing the parity flow for the minimal set $\mathbb{P}$ for $n = 4$ base qubits as compared to the original case with $\mathbb{P}_{\text{pairs}}$.

To obtain a circuit ansatz allowing universal computation from each of the circuit structures presented in Fig. 3, we can

associate the different elements of the generating sets $\mathcal{G}_{\text{parity}}$ to the different components of the circuit. Just as with the original parity quantum computing framework, we can obtain a universal computation by inserting single-qubit rotations in the appropriate locations. For example, by placing single-qubit Z- and X-rotations at the location of every solid blue circle in Figs. 3(a)–3(c), we cover the elements of $\mathcal{G}_{\text{parity}}$ corresponding to $\mathcal{G}_{\text{s,q}}$, while placing a single-qubit Z rotation at the location of every red circle covers the elements corresponding to $\mathcal{G}_{\mathbb{P}}$. Accordingly, the key differences between the different parity sets in the parity flow picture amount to differences in the two-qubit gate placement of the corresponding circuits.

Similarly to the previous subsection, it is possible to leverage Theorem 1 and the differences in circuit structure between different parity sets to find favorable implementations of the parity flow formalism. For example, let us consider the arrangement of physical qubits on a heavy-hex lattice as in Fig. 3(d). The qubits marked with both blue and red represent the qubits that represent both parity and base qubits at various times throughout the circuit (those in blue only ever represent base qubits). These base or parity dual qubits are the target of several CNOT gates with each of the neighboring qubits acting as controls. As the corresponding parity sets for this layout

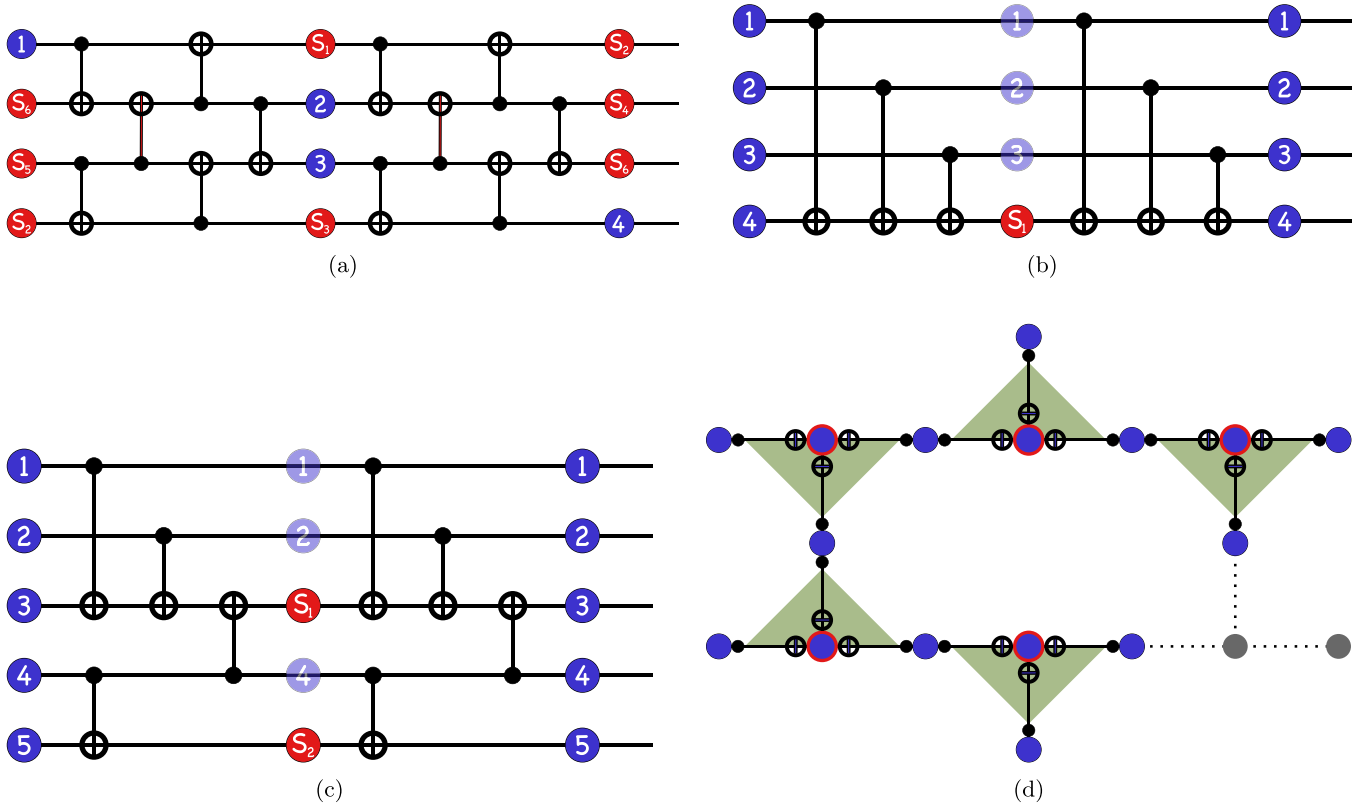


FIG. 3. In the parity flow framework of Ref. [8], the parity information of the n base qubits is tracked through time in an n -qubit circuit. A circuit ansatz permitting universal computation can be obtained from panels (a)–(c) by inserting single-qubit Z rotations at every location containing a red circle, and single-qubit X rotations at every location containing a solid blue circle (in most cases, “certain unitaries” means the single-qubit X -rotation itself but, depending on the parity flow, extra CNOTs may be required—see Ref. [8]). The faded blue circles indicate the parity information for the corresponding qubit at the given point in the circuit, but no rotations are required to be placed there for universality. In panel (a), the flow of parity information is depicted for the computation corresponding to the parity encoding shown in Fig. 1(b) for $n = 4$ and $k = 6$. As can be seen, all four base qubits and all six parity sets $S_1, \dots, S_6$ are present at least once in the circuit fragment shown [the parity sets are the same as in Fig. 1(b)]. In panel (b), the flow of parities for the minimal number of parity sets for $n = 4$ qubits is shown, i.e., with $S_1 = \{1, 2, 3, 4\}$. In panel (c), the flow of parities for the minimal number of parity sets for $n = 5$ is shown, namely with $S_1 = \{1, 2, 3, 4\}$ and $S_2 = \{4, 5\}$. In panel (d), the parity flow formalism is mapped to a heavy-hex layout, typical of the quantum devices provided by IBM Quantum [9]. Each parity set contains four elements indicated by the four qubits touching or contained within each green triangle. The qubits colored both blue and red are those that represent both base and parity qubits at various times throughout the circuit, while all blue qubits only represent base qubits. The gray qubits and dotted lines indicate part of the heavy-hex lattice not involved in the computation.

satisfy the conditions of Theorem 1, we know that universal computation can be performed by placing single-qubit Z and X gates at appropriate locations in the circuit, in the manner explained above. The heavy-hex layout considered here is of practical relevance as it is the common layout across a range of current quantum devices made available by IBM Quantum [9]. Moreover, it is in fact possible to directly map the circuits arising in the parity flow formalism to equivalent circuits using the native gates supported by the IBM Quantum platform, namely by replacing all CNOTs with CZ gates and by changing all X rotations to Z rotations and vice versa for the dual base and parity qubits.

D. Resources for MBQC

As mentioned in Sec. II B, one method to perform the decoding phase of a parity computation via single-qubit Pauli X measurements on each of the parity qubits, followed by

conditional corrections based on whether a positive (no correction required) or negative (correction required) outcome is obtained [22]. In Ref. [10], it was demonstrated that, in such a case, parity quantum computation corresponds to measurement-based quantum computation (MBQC) using specific classes of bipartite graph states and measurements in the YZ -plane of the Bloch sphere. Accordingly, a further consequence of the results presented here is that we can define novel classes of universal resources for MBQC to complement those already known in the literature (see, e.g., Ref. [15]), which we elaborate upon below.

Formally, a universal resource for MBQC is a family of states, which we denote by Ψ , such that for any state $|\gamma\rangle$ on n qubits there exists a state $|\varphi\rangle \in \Psi$ on $m \geq n$ qubits such that $|\gamma\rangle$ can be obtained deterministically from $|\varphi\rangle$ via local operations and classical communication [15]. In this context, the local operations and classical communication (LOCC) are taken to be the single-qubit projective measurements and

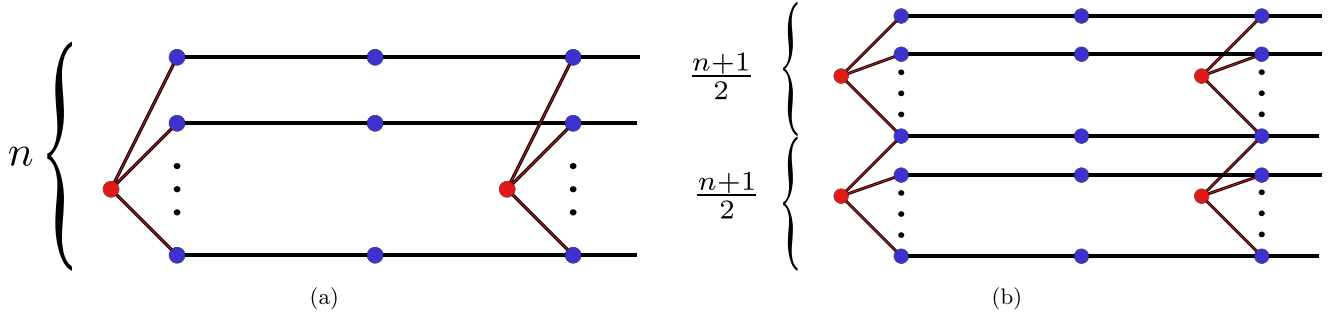


FIG. 4. Since parity quantum computing can be viewed as a specific type of measurement-based quantum computation (MBQC), the sufficient conditions from Theorem 1 in the main text allow us to derive different families of universal resources for MBQC. In this figure, graph states are depicted relating to the minimal possible generating sets for parity quantum computing, treating the two cases of an even number of base qubits [panel (a)] and an odd number of qubits [panel (b)] separately. In each case, vertices correspond to qubits prepared in the $|+\rangle$ state and edges correspond to cz gates between them. In panel (a), n different linear cluster states (with blue vertices) are connected via the red vertices, which correspond to the parity qubits under the mapping from parity quantum computing to MBQC. In panel (b), the n different linear cluster states are connected via two lots of red vertices, each of which connect to $\frac{n+1}{2}$ linear cluster states, with one in common. In both cases, universal MBQC is performed by measuring each red vertex in the YZ plane of the Bloch sphere and all blue vertices in the XY plane of the Bloch sphere. The measurement order is depicted by the left-to-right ordering of vertices.

classical feed-forwarding of measurement results upon which MBQC is based (a brief introduction to MBQC and the requirements for deterministic computation in that framework is given in Appendix D).

So, since our aim is to construct a universal resource for MBQC related to the generating sets presented above, we also need to consider different generating sets $\mathcal{G}_{\text{parity}}$ for different numbers of qubits n . As Theorem 1 identifies various different choices of $\mathbb{P}$ and hence different $\mathcal{G}_{\text{parity}}$, there are several different ways one can construct such universal resources; to simplify the discourse, we focus on the choices for $\mathbb{P}$ containing the minimal number of elements. Even with this restriction, there are typically various choices for how to choose $\mathbb{P}$ when n is odd (i.e., for $n > 3$)—in such a case, we make the further choice to take $\mathbb{P}$ to consist of S_1 and S_2 such that $|S_1| = |S_2| = \frac{n+1}{2}$. Let us then define the following notation that makes the n dependence explicit: let

$$\mathbb{P}_*^{(n)} := \begin{cases} \{\{1, 2, \dots, n\}\} & \text{if } n \bmod 2 = 0 \\ \{\{1, 2, \dots, \frac{n+1}{2}\}, \{\frac{n+1}{2}, \dots, n\}\} & \text{if } n \bmod 2 = 1, \end{cases} \quad (45)$$

and then let us define $\mathcal{G}_{\text{parity}*}^{(n)} := \mathcal{G}_{\text{s.q.}}^{(n)} \cup \mathcal{G}_{\mathbb{P}_*^{(n)}}^{(n)}$, where $\mathcal{G}_{\text{s.q.}}^{(n)} = \{iX_1, iZ_1, \dots, iX_n, iZ_n\}$ is the same as in Eq. (19) but with the extra superscript allowing reference to different numbers of qubits when necessary. Note also that $\mathbb{P}_*^{(n)}$ is the same as that considered in Sec. IV A, with $j = \frac{n+1}{2}$.

From the results of Theorem 1, we know that $\mathcal{G}_{\text{parity}*}^{(n)}$ is a universal generating set for each n . Accordingly, for any given n -qubit state $|\gamma\rangle$, there exists a sequence of rotations of elements of $\mathcal{G}_{\text{parity}*}^{(n)}$ that produces a unitary U_γ taking $|+\rangle^{\otimes n}$ to $|\gamma\rangle$. To construct a universal resource for MBQC, it suffices to produce a family of graph states that can implement U_γ for any n and any $|\gamma\rangle$.

In fact, we can proceed by combining the graph states that support computations corresponding to $\mathcal{G}_{\text{s.q.}}^{(n)}$ with those that support computations corresponding to $\mathcal{G}_{\mathbb{P}_*^{(n)}}^{(n)}$. Explicitly, since the rotations of elements of $\mathcal{G}_{\text{s.q.}}^{(n)}$ are tensor products of single-

qubit rotations, the corresponding graph states are tensor products of linear cluster states, which are known to support single-qubit rotations via measurements in the XY -plane of the Bloch sphere (see, e.g., Fig. 2 of [14]). In particular, to implement a Z rotation followed by an X rotation, it suffices to use a three-qubit cluster state:

$$\langle +_{\phi_1 + \tau_2} | CZ_{1,2} CZ_{2,3} | + + + \rangle_{123} = e^{-i\tau_2 X} e^{-i\phi_1 Z} | + \rangle_3, \quad (46)$$

where $\langle +_{\phi_1} |$ ($\langle +_{\tau_2} |$) denotes the positive projector for the measurement of the operator $e^{-i\phi_1 Z} X e^{i\phi_1 Z}$ applied to the first qubit ($e^{-i\tau_2 Z} X e^{i\tau_2 Z}$ applied to the second qubit), and where the output of the computation is given as the state of qubit 3. The tensor products of linear cluster states supporting sequences of rotations of elements from $\mathcal{G}_{\text{s.q.}}^{(n)}$ are depicted in Fig. 4 (the subgraphs containing blue vertices only).

As identified in Ref. [10], the graph states supporting rotations of elements of $\mathcal{G}_{\mathbb{P}}^{(n)}$ are bipartite graph states with the qubits related to parity sets forming one partition and the n qubit associated with the logical state forming the other. For the choice $\mathbb{P}_*^{(n)}$ that we consider here, that means that the corresponding graph states consist of $n+1$ qubits if n is even and $n+2$ if n is odd. Rotations of elements of $\mathcal{G}_{\mathbb{P}_*^{(n)}}^{(n)}$ are then implemented by performing measurements in the YZ plane of the Bloch sphere on the corresponding parity qubits; for example, for n even (i.e., $\mathbb{P}_*^{(n)} = \{S_1\}$) we have that

$$\begin{aligned} \langle 0_{\theta_{S_1}} | \prod_{j \in S_1} CZ_{S_1, j} | + \rangle_{S_1} | + \dots + \rangle_{1, \dots, n} \\ = e^{-i\theta_{S_1} Z_1 \otimes \dots \otimes Z_n} | + \dots + \rangle_{1, \dots, n}, \end{aligned} \quad (47)$$

where $\langle 0_{\theta_{S_1}} |$ is the positive projector associated with the operator $e^{-i\theta_{S_1} X} Z e^{i\theta_{S_1} X}$ applied to the parity qubit and where the computational output is a state on the n -qubits making up the partition corresponding to the base (i.e., nonparity) qubits. These bipartite graph states are depicted as subgraphs of the graphs in Fig. 4 comprising the red vertices and their nearest neighbors.

The universal resource is then constructed by combining the linear cluster states and bipartite graph states together in

an appropriate way. Specifically, consider n linear clusters states of length $2l$ arranged as in Fig. 4. Then, we identify the n qubits in every second column with the n qubits that form the nonparity partition of a copy of the bipartite graph states above, producing the graph states shown in Fig. 4(a) for the case of n even and in Fig. 4(b) for the case of n odd. Let us denote the resultant graph states as $|G_{n,l}\rangle$ and let $\Psi_{\text{parity}} := \{|G_{n,l}\rangle | n, l \in \mathbb{N}\}$.

That the family Ψ_{parity} is a universal resource follows from the fact that $\mathcal{G}_{\text{parity}^*}^{(n)}$ is a universal generating set for all n . By measuring each red vertex in the YZ plane and each blue qubit in the XY plane (and performing the appropriate corrections when necessary—see Appendix D), the final output state is

$$\underbrace{\prod_{a=1}^l \prod_{b=1}^n e^{-i\tau_a X_b} e^{-i\phi_a Z_b} \prod_{S_c \in \mathbb{P}_*^{(n)}} e^{-i\theta_a S_c} \otimes_{d \in S_c} Z_d}_{U_{\theta, \phi, \tau}} |+\rangle^{\otimes n} \quad (48)$$

where the angles θ are given by the measurements $e^{-i\theta X} Z e^{i\theta X}$ on the red vertices, the angles τ are given by the measurements $e^{-i\tau X} X e^{i\tau X}$ on the blue vertices neighboring the red vertices, and the angles ϕ are given by the measurements $e^{-i\phi Z} X e^{i\phi Z}$ on the non-neighbors of the red vertices. The universality of $\mathcal{G}_{\text{parity}^*}^{(n)}$ ensures that, for large enough l and appropriately chosen θ , τ , and ϕ , we get $U_{\theta, \phi, \tau} = U_\gamma$ meaning that the output state is indeed $|\gamma\rangle$ as required.

As mentioned earlier, the different choices of $\mathbb{P}$ that satisfy Theorem 1 all lead to universal resources for MBQC via an analogous construction to that presented above. To conclude this section, let us briefly comment on one other choice of $\mathbb{P}$, namely $\mathbb{P}_{\text{pairs}}$ [recall Eq. (21)], which contains the subset $\{|i, i+1\rangle | i = 1, \dots, n-1\}$ that satisfies Theorem 1. If we construct graph states corresponding to $\mathcal{G}_{\text{parity}}$ for $\mathbb{P}_{\text{pairs}}$, we obtain graph states that have previously been identified as well-suited for implementing the quantum approximate optimization algorithm (QAOA) (see Ref. [33] for the original QAOA and, e.g., Ref. [34] for the QAOA-suitable graph states). This is consistent with the focus on QAOA present in several works using the parity quantum computing framework (see, e.g., Refs. [35,36]).

V. DISCUSSION

Much of the focus in quantum computation research in the near-term will center around optimizing implementations of certain algorithms to minimize the total resource requirements while satisfying hardware constraints on connectivity. The generating sets presented here establish both the limits of how far the total qubit count can be minimized for the parity quantum computing framework as well as further avenues for implementing connectivity-aware universal computation.

To conclude this work, we briefly outline further implications of our results that have not been covered above and point out possible directions for future research. We comment on (i) the implications for compilation strategies for IBM Quantum architectures, (ii) the implications for noise-mitigation strategies for the parity framework that derive from purification protocols in the context of MBQC, and (iii) the possible con-

nections to other areas of quantum information theory, such as quantum cellular automata.

The implications for compilation for IBM Quantum computers arise from the results presented above for Pauli flow [recall Fig. 3(d)] in conjunction with the compilation algorithm presented in Ref. [7]. In the latter work, the algorithm PAULICOMPILER was developed which, given a generating set $\mathcal{G} \subset i\mathcal{P}_n^*$ and a target Pauli string P as input, outputs a sequence of elements from $\mathcal{G}$ that produce P via nested commutation. In other words, PAULICOMPILER specifies the unitary circuit using gates specified by $\mathcal{G}$ whose logical effect is to implement the rotation $e^{-i\theta P}$. As the generating sets considered in this work are also strictly contained in $i\mathcal{P}_n^*$, they can be used with the PAULICOMPILER (possibly with some preprocessing). In particular, using the PAULICOMPILER with the generating sets permitting parity flow computations on the heavy-hex lattice may lead to efficient compilation strategies for IBM Quantum devices, which typically use such a heavy-hex layout [9].

As discussed in Sec. IV D above, there are connections between the parity quantum computing framework and MBQC. One consequence of this is that purification protocols developed for graph states in the latter context (see, e.g., Refs. [37,38]) can be readily adapted to the former, as a method of mitigating the effects of noisy implementations of gates in current devices. The properties of these purification protocols, such as their success probability and maximal reachable fidelity, depend on the properties of the graph states being purified. Consequently, the different possible implementations of universal parity quantum computing corresponding to the different generating sets considered here (and hence also to their corresponding graph states) would exhibit different levels of noise mitigation. However, as demonstrated in Ref. [7], different Pauli-string generating sets also exhibit compilation rates for a given unitary, which indicates a possible compilation rate vs noise mitigation rate trade-off for different parity quantum computations. This trade-off is currently a topic of active research.

Finally, there are connections to other areas of quantum computation and information that are also worth highlighting. As demonstrated in Ref. [39], there are connections between MBQC and Clifford quantum cellular automata (CQCA). In the latter context, a Clifford operation satisfying certain properties (translation invariance and locality-preservation—see Ref. [39]) is repeatedly applied, interspersed with single-qubit rotations. Since the parity flow framework similarly consists of repeatedly applying a unitary comprised of a sequence of CNOT gates and single-qubit rotations, and, moreover, since connections exist between the parity framework and MBQC, it would be an interesting endeavor to clarify the similarities and differences between the parity flow and the CQCA pictures.

ACKNOWLEDGMENTS

We thank Maxime Cautrès for early discussions regarding a question related to Theorem 1 and Michael Fellner, Anette Messinger, and Christophe Goeller for discussions regarding parity quantum computing. This project was funded in whole or in part by the Austrian Science Fund (FWF) [DK-ALM

W1259-N27, SFB BeyondC F7102, SFB BeyondC F7108-N38, WIT9503323, START Grant No. Y1067-N27 and I 6011]. This work was supported by the Austrian Research Promotion Agency (FFG Project No. FO999896208). This project was funded within the QuantERA II Programme that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 101017733. This work was also co-funded by the European Union (ERC, QuantAI, Project No. 101055129).

Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

DATA AVAILABILITY

No data were created or analyzed in this study.

APPENDIX A: MAPPING PAULI STRINGS TO SYMPLECTIC VECTORS

In the main text, several of the results made use of specific properties of the set of Pauli strings and their relation to the Lie algebra $\mathfrak{su}(2^n)$. In fact, these properties allow us to map any discourse regarding nested commutation of Pauli strings to a different setting, namely, that of the symplectic space $\mathbb{F}_2^{2n}$. One advantage of doing so is that, in many cases, the questions at hand reduces to a question of linear algebra, which can be simpler to work with. In this appendix, we provide an abridged presentation of this mapping compared to that given in the supplementary material of [7]. The reader is referred to that work for further details.

Recall that $\mathcal{P}_n := \{P_1 \otimes \cdots \otimes P_n | P_i \in \{I, X, Y, Z\}\}$, $\mathcal{P}_n^* := \mathcal{P}_n \setminus \{I^{\otimes n}\}$, and $\mathfrak{su}(2^n) = \text{span}_{\mathbb{R}}\{i\mathcal{P}_n^*\}$. Recall also that, for any $A, B \in i\mathcal{P}_n^*$, the commutator $[A, B]$ is either the zero operator $\mathbf{0}$ if A and B commute, or is proportional, up to some real

scalar, to some element of $i\mathcal{P}_n^*$ otherwise. However, for the purposes of demonstrating universality, the proportionality up to a real scalar can safely be ignored since ultimately we are taking linear combinations over $\mathbb{R}$ in any case. Consequently, all the relevant information regarding $[A, B]$ can be encoded in a binary value: 0 if they commute and 1 otherwise. This is a key aspect of the mapping.

The next observation we need is that the single-qubit Pauli operators satisfy $Y = iXZ$. Accordingly, we can rewrite any element $A \in i\mathcal{P}_n^*$ as

$$A = i^{y_A+1} \left(\prod_{j=1}^n X_j^{a_j} \right) \left(\prod_{j=1}^n Z_j^{a_{n+j}} \right), \quad (\text{A1})$$

where (i) $y_A \in \mathbb{N}_0$ is the number of tensor factors of A that are a Y , and (ii) $\mathbf{a} \in \mathbb{F}_2^{2n}$ is a binary vector such that $a_j = 1$ and $a_{n+j} = 0$ if the j th tensor factor of A is a X , $a_j = 0$ and $a_{n+j} = 1$ if the j th tensor factor of A is a Z , and $a_j = a_{n+j} = 1$ if the j th tensor factor of A is a Y . This is the mapping that we consider: each element $A \in i\mathcal{P}_n^*$ is mapped to a binary vector $\mathbf{a} \in \mathbb{F}_2^{2n}$. This is a common mapping used in the context of, e.g., stabilizer quantum mechanics, quantum error correction, and classically simulating Clifford circuits efficiently [19,40–42].

So, we have a mapping from $i\mathcal{P}_n^*$ to $\mathbb{F}_2^{2n}$, but so far the latter simply has the structure of a vector space rather than of a symplectic space as promised above. Let us recall that a symplectic vector space is a vector space V over a field $\mathbb{F}$ equipped with a symplectic bilinear form $\Lambda : V \times V \rightarrow \mathbb{F}$. A symplectic bilinear form is a mapping which is (i) linear in each of its two arguments, (ii) satisfies $\Lambda(\mathbf{v}, \mathbf{v}) = 0$ for all $\mathbf{v} \in V$, and (iii) if $\Lambda(\mathbf{v}, \mathbf{w}) = 0$ for all $\mathbf{w} \in V$, then $\mathbf{v} = \mathbf{0}$. In the present context, the symplectic form encodes the commutation relation between elements of $i\mathcal{P}_n^*$. For example, for $A, B \in i\mathcal{P}_n^*$, we can use the notational convention in Eq. (A1) to write

$$[A, B] = i^{y_A+y_B+2} \left[\left(\prod_{j=1}^n X_j^{a_j} \right) \left(\prod_{j=1}^n Z_j^{a_{n+j}} \right) \left(\prod_{j=1}^n X_j^{b_j} \right) \left(\prod_{j=1}^n Z_j^{b_{n+j}} \right) - \left(\prod_{j=1}^n X_j^{b_j} \right) \left(\prod_{j=1}^n Z_j^{b_{n+j}} \right) \left(\prod_{j=1}^n X_j^{a_j} \right) \left(\prod_{j=1}^n Z_j^{a_{n+j}} \right) \right] \quad (\text{A2})$$

$$= i^{y_A+y_B+2} \left[(-1)^{\sum_{j=1}^n a_{n+j}b_j} - (-1)^{\sum_{j=1}^n a_jb_{n+j}} \right] \left(\prod_{j=1}^n X_j^{a_j+b_j} \right) \left(\prod_{j=1}^n Z_j^{a_{n+j}+b_{n+j}} \right). \quad (\text{A3})$$

The term in the square brackets in the final line encodes all the information about the commutation relation between A and B : if A, B commute, then $\sum_{j=1}^n a_jb_{n+j} = \sum_{j=1}^n a_{n+j}b_j \pmod{2}$, which can be equivalently written as $\sum_{j=1}^n a_jb_{n+j} + a_{n+j}b_j = 0 \pmod{2}$, otherwise $\sum_{j=1}^n a_jb_{n+j} + a_{n+j}b_j = 1 \pmod{2}$. This expression can be written succinctly using the vectors $\mathbf{a}, \mathbf{b}$ as

$$\mathbf{a}^\top \begin{bmatrix} \mathbf{0} & I_n \\ I_n & \mathbf{0} \end{bmatrix} \mathbf{b}, \quad (\text{A4})$$

where in $\mathbf{0}$ in the matrix denotes an $n \times n$ block consisting entirely of zeros while I_n denotes a block containing the $n \times n$

identity matrix. This matrix is precisely how we define the relevant symplectic bilinear form for $\mathbb{F}_2^{2n}$: for all $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2n}$, we define

$$\Lambda(\mathbf{a}, \mathbf{b}) := \mathbf{a}^\top \begin{bmatrix} \mathbf{0} & I_n \\ I_n & \mathbf{0} \end{bmatrix} \mathbf{b}. \quad (\text{A5})$$

In the following, we will denote the matrix also by Λ .

Let us take stock of what we have established so far with the mapping between $i\mathcal{P}_n^*$ and the symplectic space $\mathbb{F}_2^{2n}$. Every element $A \in i\mathcal{P}_n^*$ is mapped to a unique element of $\mathbb{F}_2^{2n}$. Note that no element of $i\mathcal{P}_n^*$ is mapped to the zero element $\mathbf{0} \in \mathbb{F}_2^{2n}$ (this is convenient for considering commuting operators, as

we will see in a moment). Moreover, every other element of $\mathbb{F}_2^{2n}$ is uniquely associated with an element of $i\mathcal{P}_n^*$ and vice versa. Consequently, the task of generating all of $i\mathcal{P}_n^*$ as stated in the main text, reduces to generating all of $\mathbb{F}_2^{2n}$ under the mapping. For any $A, B \in i\mathcal{P}_n^*$ with corresponding vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2n}$, the operator $[A, B]$ is mapped to $\mathbf{a}^\top \Lambda \mathbf{b}(\mathbf{a} + \mathbf{b}) \in \mathbb{F}_2^{2n}$. That is, if A, B commute, they get mapped to $\mathbf{0} \in \mathbb{F}_2^{2n}$ since $\mathbf{a}^\top \Lambda \mathbf{b} = 0$, otherwise they get mapped to $\mathbf{a} + \mathbf{b}$ (note that the vector addition is element-wise and binary). This indicates the utility of this mapping: commutation of operators in the setting of $i\mathcal{P}_n^*$ corresponds to simple vector addition in $\mathbb{F}_2^{2n}$.

To conclude this section, let us establish some notation for later use. Below, we will often refer to a symplectic basis of $\mathbb{F}_2^{2n}$. A symplectic basis of $\mathbb{F}_2^{2n}$ is a basis $\{\mathbf{v}_1, \dots, \mathbf{v}_{2n}\}$ such that for each $\mathbf{v}_i$ there is precisely one $\mathbf{v}_j$ such that $\mathbf{v}_i^\top \Lambda \mathbf{v}_j = 1$. We will write any symplectic basis to which we refer using the suggestive notation $\{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_1, \dots, \mathbf{z}_n\}$ where

$$\begin{aligned} \mathbf{x}_i^\top \Lambda \mathbf{z}_j &= \delta_{ij}, \\ \mathbf{x}_i^\top \Lambda \mathbf{x}_j &= \mathbf{z}_i^\top \Lambda \mathbf{z}_j = 0 \quad \forall i, j. \end{aligned} \quad (\text{A6})$$

This notation is suggestive as these constraints mirror the commutation relations of the single-qubit operators in $\mathcal{G}_{\text{s.q.}}$, but in general we do not enforce that the $\mathbf{x}_j \in \mathbb{F}_2^{2n}$ are the images of the $iX_j \in \mathcal{G}_{\text{s.q.}} \subset i\mathcal{P}_n^*$ and similarly for the $\mathbf{z}_j$.⁷ It will be convenient below to make use of the following notation, where $S \subseteq \{1, \dots, n\}$:

$$\begin{aligned} \mathbf{z}^S &:= \sum_{j \in S} \mathbf{z}_j, \\ \mathbf{x}^S &:= \sum_{j \in S} \mathbf{x}_j. \end{aligned} \quad (\text{A7})$$

This notation is also suggestive, since, if the $\mathbf{z}_j$ are the images of the $iZ_j \in \mathcal{G}_{\text{s.q.}}$ then $\mathbf{z}^S$ is the image of $i \bigotimes_{j \in S} Z_j \in \mathcal{G}_{\mathbb{P}}$. If $S = \emptyset$, then we define $\mathbf{x}^S = \mathbf{z}^S = \mathbf{0} \in \mathbb{F}_2^{2n}$.

Let $\{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_1, \dots, \mathbf{z}_n\} \subset \mathbb{F}_2^{2n}$ be a fixed symplectic basis. We can write any $\mathbf{v} \in \mathbb{F}_2^{2n}$ uniquely (with respect to this basis) as

$$\mathbf{v} = \sum_{i=1}^n \alpha_i \mathbf{x}_i + \beta_i \mathbf{z}_i, \quad (\text{A8})$$

where $\alpha_i, \beta_i \in \mathbb{F}_2$ for all $i = 1, \dots, n$. Let us define

$$\mathcal{X}(\mathbf{v}) := \{i \in [n] \mid \alpha_i \neq 0\}, \quad (\text{A9})$$

$$\mathcal{Z}(\mathbf{v}) := \{i \in [n] \mid \beta_i \neq 0\}, \quad (\text{A10})$$

$$\text{wt}(\mathbf{v}) := |\mathcal{X}(\mathbf{v}) \cup \mathcal{Z}(\mathbf{v})|. \quad (\text{A11})$$

The latter quantity is the equivalent notion in the symplectic space picture of the usual definition of the weight of a Pauli string (i.e., the number of nonidentity tensor factors).

⁷Note that any symplectic basis of $\mathbb{F}_2^{2n}$ can be mapped to the set of elements that are the images of $\mathcal{G}_{\text{s.q.}}$ by a symplectic transform. This corresponds to a Clifford operation at the level of $i\mathcal{P}_n^*$.

In direct parallel to the adjoint map defined for general Lie algebras [recall Eq. (6)], let us define, for each $\mathbf{a} \in \mathbb{F}_2^{2n}$, the map $\text{ad}_{\mathbf{a}}(\cdot)$ via

$$\text{ad}_{\mathbf{a}}(\mathbf{b}) := \mathbf{a}^\top \Lambda \mathbf{b}(\mathbf{a} + \mathbf{b}). \quad (\text{A12})$$

When a set $G \subset \mathbb{F}_2^{2n}$ is specified, we define (again, directly paralleling the notation in the main text) the quantity

$$G^{\text{ad}^{(r)}} := \{\text{ad}_{\mathbf{v}_1} \text{ad}_{\mathbf{v}_2} \cdots \text{ad}_{\mathbf{v}_r}(\mathbf{v}_{r+1}) \mid \mathbf{v}_1, \dots, \mathbf{v}_{r+1} \in G\} \subseteq \mathbb{F}_2^{2n} \quad (\text{A13})$$

in order to be able to make statements about $G \bigcup_{r=1}^{\infty} G^{\text{ad}^{(r)}}$.

As we will be considering sequences of adjoint maps, typically for elements from a choice of symplectic basis of $\mathbb{F}_2^{2n}$, it will be convenient to have compact notation for representing both the sequence of elements as well as the corresponding compositions of adjoint maps. For $S \subseteq \{1, \dots, n\}$ with elements $i_1, \dots, i_{|S|}$, we define

$$\mathbf{x}_S := \mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots, \mathbf{x}_{i_{|S|}}, \quad (\text{A14})$$

$$\text{ad}_{\mathbf{x}_S}(\cdot) := \text{ad}_{\mathbf{x}_{i_1}} \text{ad}_{\mathbf{x}_{i_2}} \cdots \text{ad}_{\mathbf{x}_{i_{|S|}}}(\cdot), \quad (\text{A15})$$

and similarly for $\mathbf{z}_S$ and $\text{ad}_{\mathbf{z}_S}(\cdot)$. Note the differences in notation to Eq. (A7) above: the set S in the superscript indicates a sum of elements while the set S in the subscript indicates a sequence.⁸ In the cases where the latter notation is used, the specific ordering of the elements of S will not matter; the important factor is that each element appears in the sequence precisely once.

These notational conventions provide a compact way of writing certain elements and adjoint sequences, which will be useful below. For example, for any $\mathbf{v} \in \mathbb{F}_2^{2n}$ written as in Eq. (A8) with respect to a given symplectic basis, we can write

$$\mathbf{v} = \mathbf{x}^{\mathcal{X}(\mathbf{v})} + \mathbf{z}^{\mathcal{Z}(\mathbf{v})}. \quad (\text{A16})$$

Furthermore, if $S \subseteq T \subseteq \{1, \dots, n\}$, we can write

$$\text{ad}_{\mathbf{x}_S}(\mathbf{z}^T) \equiv \text{ad}_{\mathbf{x}_{i_1}} \cdots \text{ad}_{\mathbf{x}_{i_{|S|}}}(\mathbf{z}^T) = \mathbf{x}^S + \mathbf{z}^T. \quad (\text{A17})$$

APPENDIX B: PROOF OF THEOREM 1

In this section, we present the proof of Theorem 1 in the main text. We make use of the mapping and notation presented in Appendix A. In the statement of Theorem 1, we start with the set $\mathcal{G}_{\text{s.q.}}$ and consider what sets $\mathcal{G}_{\mathbb{P}}$ ensure that $\mathcal{G}_{\text{parity}} := \mathcal{G}_{\text{s.q.}} \cup \mathcal{G}_{\mathbb{P}}$ are universal. Under the mapping outlined above, $i\mathcal{P}_n^*$ becomes $\mathbb{F}_2^{2n} \setminus \mathbf{0}$, $\mathcal{G}_{\text{s.q.}}$ becomes a symplectic basis of $\mathbb{F}_2^{2n}$, and $\mathcal{G}_{\mathbb{P}}$ becomes $\{\mathbf{z}^S \mid S \in \mathbb{P}\}$. With these changes, Theorem 1 can be stated equivalently as follows:

Theorem B1. Let $n \in \mathbb{N}_{\geq 2}$ and let $G' \subset \mathbb{F}_2^{2n}$ be a symplectic basis with elements $\{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_1, \dots, \mathbf{z}_n\}$ which satisfy Eq. (A6). Let $S_1, \dots, S_k$ for some $1 \leq k \leq n-1$ be such that

- (1) $|S_i|$ is even for all $i \in \{1, \dots, k\}$,
- (2) $\bigcup_{i=1}^k S_i = [n]$,

⁸We would like to highlight that the use of sets as sub- and superscripts appears in the notation in distinct ways in this work. For example, using a set S as a subscript on a lower-case $\mathbf{z}$ representing a vector in $\mathbb{F}_2^{2n}$ is *not* the same as a subscript on the capital Z , representing a Pauli Z rotation on a parity qubit.

(3) if $k \geq 2$, then

(i) $S_i \cap S_j = \emptyset$ for all $1 \leq j \leq k$ such that $j \neq i, i+1$, and

(ii) $S_i \cap S_{i+1} = \{s_i\}$ for all $i \leq k-1$, with the $s_i \in [n]$ all distinct.

Then, for $G := G' \cup \{z^{S_i} | i = 1, \dots, k\}$, we have that

$$G \bigcup_{r=1}^{\infty} G^{\text{ad}^{(r)}} = \mathbb{F}_2^{2n}. \quad (\text{B1})$$

Before proving the theorem, we state and prove two lemmas that are used in the proof of the theorem. The first allows us to reduce the task of showing that $G \bigcup_{r=1}^{\infty} G^{\text{ad}^{(r)}} = \mathbb{F}_2^{2n}$ even further by making use of the properties of the symplectic basis G' , while the second abstracts some of the structure present in the proof of the theorem.

Lemma B1. Let $G' = \{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_1, \dots, \mathbf{z}_n\} \subseteq \mathbb{F}_2^{2n}$ be a symplectic basis satisfying Eq. (A6) and let $T \subseteq \{1, \dots, n\}$. Then for any $\mathbf{w} \in \mathbb{F}_2^{2n}$ that can be written as

$$\mathbf{w} = \sum_{i \in T} \alpha_i \mathbf{x}_i + \beta_i \mathbf{z}_i \quad (\text{B2})$$

for $\alpha_i, \beta_i \in \{0, 1\}$ are not both zero for each $i \in T$, there exists a sequence of elements $\mathbf{u}_1, \dots, \mathbf{u}_r \in G'$ for some r

such that

$$\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_r}(\mathbf{z}^T) = \mathbf{w}. \quad (\text{B3})$$

Proof. By assumption, $\mathbf{w}$ is such that $\mathcal{X}(\mathbf{w}), \mathcal{Z}(\mathbf{w}) \subseteq T$ and moreover that $T \setminus \mathcal{Z}(\mathbf{w}) \subseteq \mathcal{X}(\mathbf{w})$ (the latter constraint comes from the fact that α_i and β_i cannot both be 0). Let $\mathbf{u}_1, \dots, \mathbf{u}_r$ be the sequence $\mathbf{z}_{T \setminus \mathcal{Z}(\mathbf{w})}, \mathbf{x}_{\mathcal{X}(\mathbf{w})}$. Using the Eq. (A6), we then have

$$\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_r}(\mathbf{z}^T) = \text{ad}_{\mathbf{z}_{T \setminus \mathcal{Z}(\mathbf{w})}} \text{ad}_{\mathbf{x}_{\mathcal{X}(\mathbf{w})}}(\mathbf{z}^T) \quad (\text{B4})$$

$$= \text{ad}_{\mathbf{z}_{T \setminus \mathcal{Z}(\mathbf{w})}}(\mathbf{x}^{\mathcal{X}(\mathbf{w})} + \mathbf{z}^T) \quad (\text{B5})$$

$$= \mathbf{x}^{\mathcal{X}(\mathbf{w})} + \mathbf{z}^T + \mathbf{z}^{T \setminus \mathcal{Z}(\mathbf{w})} \quad (\text{B6})$$

$$= \mathbf{x}^{\mathcal{X}(\mathbf{w})} + \mathbf{z}^{\mathcal{Z}(\mathbf{w})} \quad (\text{B7})$$

$$= \mathbf{w}, \quad (\text{B8})$$

as required. $\blacksquare$

Lemma B2. Let $A, B, C \subseteq [n]$ be such that $A, B \neq \emptyset, A \subseteq B, |B| = 0 \bmod 2$, and $|B \cap C| = 1$ if $C \neq \emptyset$ (C is allowed to be empty, while A and B are not). Let $\bar{A} := B \setminus A$ and, if $A \neq \emptyset$ and $|A| = 0 \bmod 2$, let $a \in A$ be a distinguished element such that $a \notin A \cap C$ and define $\tilde{A} := \bar{A} \cup \{a\}$. Let us define the sequence

$$\mathbf{p}_1, \dots, \mathbf{p}_l := \begin{cases} \mathbf{x}_A, \mathbf{z}_{A \setminus (B \cap C)}, \mathbf{z}^B, \mathbf{x}_{A \setminus (B \cap C)}, \mathbf{z}^B, \mathbf{x}_{B \cap C}, \mathbf{z}^C & \text{if } A \cap C \neq \emptyset, |A| = 1 \bmod 2 \\ \mathbf{x}_A, \mathbf{z}_A, \mathbf{z}^B, \mathbf{x}_{A \cup (B \cap C)}, \mathbf{z}_{B \cap C}, \mathbf{z}^B, \mathbf{x}_{B \cap C}, \mathbf{z}^C & \text{if } A \cap C = \emptyset, |A| = 1 \bmod 2 \\ \mathbf{x}_a, \mathbf{z}^B, \mathbf{x}_{\bar{A}}, \mathbf{z}_{\bar{A}}, \mathbf{z}^B, \mathbf{x}_{\tilde{A} \cup (B \cap C)}, \mathbf{z}_{B \cap C}, \mathbf{z}^B, \mathbf{x}_{B \cap C}, \mathbf{z}^C & \text{if } A \cap C \neq \emptyset, |A| = 0 \bmod 2 \\ \mathbf{x}_a, \mathbf{z}^B, \mathbf{x}_{\bar{A}}, \mathbf{z}_{\bar{A} \setminus (B \cap C)}, \mathbf{z}^B, \mathbf{x}_{\tilde{A} \setminus (B \cap C)}, \mathbf{z}^B, \mathbf{x}_{B \cap C}, \mathbf{z}^C & \text{if } A \cap C = \emptyset, |A| = 0 \bmod 2, A \neq \emptyset. \end{cases} \quad (\text{B9})$$

where any element with a subscript or superscript that is an empty set is simply removed from the sequence. Then

$$\text{ad}_{\mathbf{p}_1} \cdots \text{ad}_{\mathbf{p}_{l-1}}(\mathbf{p}_l) = \mathbf{z}^{A \cup [C \setminus (B \cap C)]}. \quad (\text{B10})$$

Proof. First note that each of the four cases in Eq. (B9) have the same right-hand end of the sequence, namely $\mathbf{z}^B, \mathbf{x}_{B \cap C}, \mathbf{z}^C$. If C is empty, this shortens to $\mathbf{z}^B$. When C is nonempty, we will use that, due to the requirement that $|B \cap C| = 1$, $(\mathbf{z}^B)^\top \Lambda \mathbf{x}_{B \cap C} = 1$, so we can write

$$\text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{B \cap C}}(\mathbf{z}^C) = \mathbf{x}^{B \cap C} + \mathbf{z}^{(B \cup C) \setminus (B \cap C)}. \quad (\text{B11})$$

As the expression above on the right reduces to $\mathbf{z}^B$ for $C = \emptyset$, we can treat both the cases when C is empty and when it is nonempty simultaneously. Furthermore, let us note that, if $|A| = 1 \bmod 2$, then $(\mathbf{z}^B)^\top \Lambda \mathbf{x}^A = (\mathbf{z}^A)^\top \Lambda \mathbf{x}^A = 1$, and if $|A| = 0 \bmod 2$, then $(\mathbf{z}^B)^\top \Lambda \mathbf{x}^A = 1$. Finally, note that if $A \cap C \neq \emptyset$, then $A \cap C = B \cap C$.

For the case where $A \cap C \neq \emptyset$ and $|A| = 1 \bmod 2$, we have that

$$\text{ad}_{\mathbf{p}_1} \cdots \text{ad}_{\mathbf{p}_{l-1}}(\mathbf{p}_l) = \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_{A \setminus (B \cap C)}} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{A \setminus (B \cap C)}} \times (\mathbf{x}^{B \cap C} + \mathbf{z}^{(B \cup C) \setminus (B \cap C)}) \quad (\text{B12})$$

$$= \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_{A \setminus (B \cap C)}} \text{ad}_{\mathbf{z}^B}(\mathbf{x}^A + \mathbf{z}^{(B \cup C) \setminus (B \cap C)}) \quad (\text{B13})$$

$$= \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_{A \setminus (B \cap C)}}(\mathbf{x}^A + \mathbf{z}^C) \quad (\text{B14})$$

$$= \text{ad}_{\mathbf{x}_A}(\mathbf{x}^A + \mathbf{z}^{A \cup C}) \quad (\text{B15})$$

$$= \mathbf{z}^{A \cup C} \quad (\text{B16})$$

$$\equiv \mathbf{z}^{A \cup [C \setminus (B \cap C)]}, \quad (\text{B17})$$

where the equivalence in the last line arises from the fact that $A \cap C = B \cap C \neq \emptyset$. For the case where $A \cap C = \emptyset$ and $|A| = 1 \bmod 2$, we have that

$$\text{ad}_{\mathbf{p}_1} \cdots \text{ad}_{\mathbf{p}_{l-1}}(\mathbf{p}_l) = \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_A} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{A \cup (B \cap C)}} \text{ad}_{\mathbf{z}_{B \cap C}} \times (\mathbf{x}^{B \cap C} + \mathbf{z}^{(B \cup C) \setminus (B \cap C)}) \quad (\text{B18})$$

$$= \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_A} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{A \cup (B \cap C)}}(\mathbf{x}^{B \cap C} + \mathbf{z}^{B \cup C}) \quad (\text{B19})$$

$$= \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_A} \text{ad}_{\mathbf{z}^B}(\mathbf{x}^A + \mathbf{z}^{B \cup C}) \quad (\text{B20})$$

$$= \text{ad}_{\mathbf{x}_A} \text{ad}_{\mathbf{z}_A}(\mathbf{x}^A + \mathbf{z}^{C \setminus (B \cap C)}) \quad (\text{B21})$$

$$= \text{ad}_{\mathbf{x}_A}(\mathbf{x}^A + \mathbf{z}^{A \cup [C \setminus (B \cap C)]}) \quad (\text{B22})$$

$$= \mathbf{z}^{A \cup [C \setminus (B \cap C)]}. \quad (\text{B23})$$

For the case where $A \cap C \neq \emptyset$ and $|A| = 0 \bmod 2$, we have that

$$\text{ad}_{\mathbf{p}_1} \cdots \text{ad}_{\mathbf{p}_{l-1}}(\mathbf{p}_l) = \text{ad}_{\mathbf{x}_a} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{\bar{A}}} \text{ad}_{\mathbf{z}_{\bar{A}}} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{\bar{A} \cup (B \cap C)}} \text{ad}_{\mathbf{z}_{B \cap C}} \times (\mathbf{x}^{B \cap C} + \mathbf{z}^{(B \cup C) \setminus (B \cap C)}) \quad (\text{B24})$$

$$= \text{ad}_{\mathbf{x}_a} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{\bar{A}}} \text{ad}_{\mathbf{z}_{\bar{A}}} \text{ad}_{\mathbf{z}^B} \text{ad}_{\mathbf{x}_{\bar{A} \cup (B \cap C)}} \times (\mathbf{x}^{B \cap C} + \mathbf{z}^{B \cup C}) \quad (\text{B25})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} \text{ad}_{z_{\bar{A}}} \text{ad}_{z^B} (\tilde{x}^{\bar{A}} + z^{B \cup C}) \quad (\text{B26})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} \text{ad}_{z_{\bar{A}}} (\tilde{x}^{\bar{A}} + z^{C \setminus (B \cap C)}) \quad (\text{B27})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} (\tilde{x}^{\bar{A}} + z^{\bar{A} \cup [C \setminus (B \cap C)]}) \quad (\text{B28})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} (x_a + z^{\bar{A} \cup [C \setminus (B \cap C)]}) \quad (\text{B29})$$

$$= \text{ad}_{x_a} (x_a + z^{A \cup C}) \quad (\text{B30})$$

$$= z^{A \cup C} \quad (\text{B31})$$

$$\equiv z^{A \cup [C \setminus (B \cap C)]}. \quad (\text{B32})$$

For the case where $A \cap C = \emptyset$, $|A| = 0 \pmod 2$ and $A \neq \emptyset$, we have that

$$\text{ad}_{p_1} \cdots \text{ad}_{p_{l-1}} (p_l) = \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} \text{ad}_{z_{\bar{A}}} \text{ad}_{x_{\bar{A} \setminus (B \cap C)}} \text{ad}_{z^B} \text{ad}_{x_{\bar{A} \setminus (B \cap C)}} \quad (\text{B33})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} \text{ad}_{z_{\bar{A}}} \text{ad}_{z^B} \times (\tilde{x}^{\bar{A}} + z^{(B \cup C) \setminus (B \cap C)}) \quad (\text{B34})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} \text{ad}_{z_{\bar{A}}} (\tilde{x}^{\bar{A}} + z^C) \quad (\text{B35})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} \text{ad}_{x_{\bar{A}}} (\tilde{x}^{\bar{A}} + z^{\bar{A} \cup C}) \quad (\text{B36})$$

$$= \text{ad}_{x_a} \text{ad}_{z^B} (x_a + z^{\bar{A} \cup C}) \quad (\text{B37})$$

$$= \text{ad}_{x_a} (x_a + z^{A \cup [C \setminus (B \cap C)]}) \quad (\text{B38})$$

$$= z^{A \cup [C \setminus (B \cap C)]}. \quad (\text{B39})$$

Proof. Since it is trivial to produce a sequence that generates $\mathbf{0} \in \mathbb{F}_2^{2n}$, for example by considering the sequence x_1, x_1 , we focus our attention on generating $\mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}$. In fact, we know from Lemma B1 that it suffices to demonstrate that for each subset $T \subseteq \{1, \dots, n\}$, $z^T \in G \bigcup_{r=1}^{\infty} G^{\text{ad}^{(r)}}$. As the case where $T = \emptyset$ corresponds to $z^T = \mathbf{0}$, we only consider $T \neq \emptyset$ henceforth. For $1 \leq i \leq j \leq k$, let us define the following notation:

$$S_{i:j} := \bigcup_{l=i}^j S_l, \quad (\text{B40})$$

$$T_{i:j} := T \cap S_{i:j}. \quad (\text{B41})$$

If $i = j$, we simply write S_j and T_j .

The proof proceeds by demonstrating the following two points:

(i) There exist sequences $u_1, \dots, u_r$ and $w_1, \dots, w_t$ such that

$$\text{ad}_{u_1} \cdots \text{ad}_{u_{r-1}} (u_r) = z^{T_1}, \quad (\text{B42})$$

$$\text{ad}_{w_1} \cdots \text{ad}_{w_{t-1}} (w_t) = z^{T_1 \cup \{s_1\}}. \quad (\text{B43})$$

(ii) If $k \geq 2$ and there exist sequences $u'_1, \dots, u'_{r'}$ and $w'_1, \dots, w'_{t'}$ such that

$$\text{ad}_{u'_1} \cdots \text{ad}_{u'_{r'-1}} (u'_{r'}) = z^{T_{1:j}}, \quad (\text{B44})$$

$$\text{ad}_{w'_1} \cdots \text{ad}_{w'_{t'-1}} (w'_{t'}) = z^{T_{1:j} \cup \{s_j\}} \quad (\text{B45})$$

for a given $1 \leq j \leq k-1$, then there exists a sequence $u_1, \dots, u_r$ such that

$$\text{ad}_{u_1} \cdots \text{ad}_{u_{r-1}} (u_r) = z^{T_{1:j+1}}, \quad (\text{B46})$$

and, if in addition $j+1 < k$, there also exists a sequence $w_1, \dots, w_t$ such that

$$\text{ad}_{w_1} \cdots \text{ad}_{w_{t-1}} (w_t) = z^{T_{1:j+1} \cup \{s_{j+1}\}}. \quad (\text{B47})$$

Before proving these two points, let us comment on why they suffice for establishing the theorem. If $k = 1$, then $T_1 = T$ and we are finished using the first point alone. If $k \geq 2$, then from the first point we know we can produce z^{T_1} and $z^{T_1 \cup \{s_1\}}$, so by the second point we know we can also produce $z^{T_{1:2}}$ and $z^{T_{1:2} \cup \{s_2\}}$. By iteratively applying the second point, we know we can produce $z^{T_{1:j}}$ for all $1 \leq j \leq k$, including $j = k$ itself, for which $T_{1:k} = T$. That is, in the case where $k \geq 2$, the proof proceeds inductively, with the first point being the base case and the second point the inductive step.

Proof of (i). If $T_1 = \emptyset$, defining $u_1, \dots, u_r$ to be the sequence x_1, x_1 trivially gives the desired result. Otherwise, we define $u_1, \dots, u_r$ to be the sequence $p_1, \dots, p_l$ from Lemma B2 for $A = T_1$, $B = S_1$, and $C = \emptyset$. As a result of that lemma, we get that

$$\text{ad}_{u_1} \cdots \text{ad}_{u_{r-1}} (u_r) = z^{A \cup [C \setminus (B \cap C)]} = z^{T_1}, \quad (\text{B48})$$

as required.

For the sequence $w_1, \dots, w_t$, we again use Lemma B2, now with $A = T_1 \cup \{s_1\}$ instead (meaning that $A \neq \emptyset$ regardless of whether T_1 is empty). Defining $w_1, \dots, w_t$ to be the resultant sequence $p_1, \dots, p_l$, we get that

$$\text{ad}_{w_1} \cdots \text{ad}_{w_{t-1}} (w_t) = z^{A \cup [C \setminus (B \cap C)]} = z^{T_1 \cup \{s_1\}}. \quad (\text{B49})$$

Proof of (ii). Suppose that $k \geq 2$ and that there exist sequences $u'_1, \dots, u'_{r'}$ and $w'_1, \dots, w'_{t'}$ such that for a given $1 \leq j \leq k-1$,

$$\text{ad}_{u'_1} \cdots \text{ad}_{u'_{r'-1}} (u'_{r'}) = z^{T_{1:j}}, \quad (\text{B50})$$

$$\text{ad}_{w'_1} \cdots \text{ad}_{w'_{t'-1}} (w'_{t'}) = z^{T_{1:j} \cup \{s_j\}}. \quad (\text{B51})$$

If $T_{1:j} = \emptyset$, then we are in an analogous situation to point (i), and so the same reasoning applies. If T_{j+1} is also empty, then the required sequence is trivial same as above; otherwise, we can leverage Lemma B2 to obtain the desired sequences by defining $B := S_{j+1}$ and $A := T_{j+1}$ to obtain $u_1, \dots, u_r$, and, if in addition $j+1 < k$, by defining $B := S_{j+1}$ and $A := T_{j+1} \cup \{s_{j+1}\}$ to obtain $w_1, \dots, w_t$. Henceforth, let us assume that $T_{1:j} \neq \emptyset$.

To construct the new sequence $u_1, \dots, u_r$ and $w_1, \dots, w_t$ we can make further use of Lemma B2. For the former, we first note that if $T_{j+1} = \emptyset$, then $T_{1:j+1} = T_{1:j}$ so we merely take $u_1, \dots, u_r$ to be $u'_1, \dots, u'_{r'}$. Otherwise, taking $A = T_{j+1}$, $B = S_{j+1}$ and $C = T_{1:j} \cup \{s_j\}$, we get that $B \cap C = \{s_j\}$ and furthermore that the sequence $p_1, \dots, p_l$ from Eq. (B9) is such that

$$\text{ad}_{p_1} \cdots \text{ad}_{p_{l-1}} (p_l) = z^{A \cup [C \setminus (B \cap C)]} = z^{T_{j+1} \cup (T_{1:j} \setminus \{s_j\})} = z^{T_{1:j+1}}. \quad (\text{B52})$$

Noting that in each case in Eq. (B9), $p_l = z^{T_{1:j} \cup \{s_j\}}$, which is the element produced by the adjoint sequence $w'_1, \dots, w'_{t'}$

by assumption, defining $\mathbf{u}_1, \dots, \mathbf{u}_r$ to be the sequence $\mathbf{p}_1, \dots, \mathbf{p}_{l-1}, \mathbf{w}'_1, \dots, \mathbf{w}'_{l'}$ gives the desired result.

If $j+1 < k$, we can apply similar reasoning to produce the sequence $\mathbf{w}_1, \dots, \mathbf{w}_l$. Taking $A = T_{j+1} \cup \{s_{j+1}\}$, $B = S_{j+1}$, and $C = T_{1:j} \cup \{s_j\}$, Lemma B2 ensures that the sequence $\mathbf{p}_1, \dots, \mathbf{p}_l$ from Eq. (B9) satisfies

$$\begin{aligned} \text{ad}_{\mathbf{p}_1} \cdots \text{ad}_{\mathbf{p}_{l-1}}(\mathbf{p}_l) &= \mathbf{z}^{A \cup (C \setminus B \cap C)} = \mathbf{z}^{(T_{j+1} \cup \{s_{j+1}\}) \cup (T_{1:j} \setminus \{s_j\})} \\ &= \mathbf{z}^{T_{1:j+1} \cup \{s_{j+1}\}}. \end{aligned} \quad (\text{B53})$$

As above, taking $\mathbf{w}_1, \dots, \mathbf{w}_l$ to be the concatenated sequence $\mathbf{p}_1, \dots, \mathbf{p}_{l-1}, \mathbf{w}'_1, \dots, \mathbf{w}'_{l'}$ gives the desired result. ■

APPENDIX C: PROOF OF THEOREM 2

In this section we prove an equivalent form of Theorem 2 using the mapping and notation from Appendix A. In particular, we will make use of the quantity $\text{wt}(\cdot)$ which is with defined with respect to a given symplectic basis. We assume that such a basis, denoted $\{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_1, \dots, \mathbf{z}_n\}$, is given throughout this section. For the proof of the theorem, we make use of the following lemma:

Lemma C1. Let $\mathbf{u}, \mathbf{v} \in \mathbb{F}_2^{2n}$ be such that $\mathbf{v} \neq \mathbf{u}$, $\text{wt}(\mathbf{v}) = n$ and $\text{ad}_{\mathbf{v}}(\mathbf{u}) \neq \mathbf{0}$. Then $\text{wt}(\text{ad}_{\mathbf{v}}(\mathbf{u})) = n - \text{wt}(\mathbf{u}) + \sigma$, where $1 \leq \sigma \leq n$ is an odd integer.

Proof. Let us write

$$\mathbf{v} = \sum_{i=1}^n \alpha_i \mathbf{x}_i + \beta_i \mathbf{z}_i, \quad (\text{C1})$$

$$\mathbf{u} = \sum_{i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})} \gamma_i \mathbf{x}_i + \delta_i \mathbf{z}_i, \quad (\text{C2})$$

where $\alpha_i, \beta_i, \gamma_i, \delta_i \in \{0, 1\}$. Since $\text{wt}(\mathbf{v}) = n$, we have that for all $i \in [n]$, α_i and β_i are not both zero, and, similarly, for all $j \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})$, γ_j and δ_j are not both zero. Suppose that $\text{ad}_{\mathbf{v}}(\mathbf{u}) \neq \mathbf{0}$. In particular, this means that

$$\mathbf{v}^\top \Lambda \mathbf{u} = \sum_{i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})} \alpha_i \delta_i + \beta_i \gamma_i = 1 \pmod{2}. \quad (\text{C3})$$

For this to be the case, there must be an odd number of $i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})$ such that $\alpha_i \delta_i + \beta_i \gamma_i = 1$. This can only occur if either α_i and δ_i are both 1 and at most one of β_i and γ_i is 1, or β_i and γ_i are both 1 and at most one of α_i and δ_i is 1. Let us define $\sigma := |\{i : \alpha_i \delta_i + \beta_i \gamma_i = 1\}|$.

Next, note that $\text{ad}_{\mathbf{v}}(\mathbf{u}) \neq \mathbf{0}$ also means that

$$\begin{aligned} \text{ad}_{\mathbf{v}}(\mathbf{u}) = \mathbf{v} + \mathbf{u} &= \sum_{i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})} (\alpha_i + \gamma_i) \mathbf{x}_i + (\beta_i + \delta_i) \mathbf{z}_i \\ &+ \sum_{j \in [n] \setminus (\mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u}))} \alpha_j \mathbf{x}_j + \beta_j \mathbf{z}_j. \end{aligned} \quad (\text{C4})$$

It follows that

$$\begin{aligned} \text{wt}(\text{ad}_{\mathbf{v}}(\mathbf{u})) &= \text{wt} \left(\sum_{i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})} (\alpha_i + \gamma_i) \mathbf{x}_i + (\beta_i + \delta_i) \mathbf{z}_i \right) \\ &+ \text{wt} \left(\sum_{j \in [n] \setminus (\mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u}))} \alpha_j \mathbf{x}_j + \beta_j \mathbf{z}_j \right). \end{aligned} \quad (\text{C5})$$

Due to the requirements on the α_i and β_i , we know that

$$\begin{aligned} \text{wt} \left(\sum_{j \in [n] \setminus (\mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u}))} \alpha_j \mathbf{x}_j + \beta_j \mathbf{z}_j \right) \\ = |[n] \setminus (\mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u}))| = n - \text{wt}(\mathbf{u}). \end{aligned} \quad (\text{C6})$$

Finally, the implications of the fact that $\mathbf{v}^\top \Lambda \mathbf{w} = 1$ discussed above ensure that

$$\text{wt} \left(\sum_{i \in \mathcal{X}(\mathbf{u}) \cup \mathcal{Z}(\mathbf{u})} (\alpha_i + \gamma_i) \mathbf{x}_i + (\beta_i + \delta_i) \mathbf{z}_i \right) = \sigma, \quad (\text{C7})$$

which completes the proof. ■

The equivalent statement to Theorem 2 in the symplectic picture is the following theorem:

Theorem C1. Let $\mathcal{G}$ be a symplectic basis of $\mathbb{F}_2^{2n}$ as above. Then for any $\mathbf{v} \in \mathbb{F}_2^{2n}$, we have that

$$\langle \mathcal{G} \cup \{\mathbf{v}\} \rangle_{[\cdot, \cdot]} \subsetneq \mathbb{F}_2^{2n}. \quad (\text{C8})$$

Proof. The proof is established by demonstrating that, for any choice of $\mathbf{v} \in \mathbb{F}_2^{2n}$, there is always some element $\mathbf{w} \in \mathbb{F}_2^{2n}$ such that $\mathbf{w} \notin \langle \mathcal{G} \cup \{\mathbf{v}\} \rangle_{[\cdot, \cdot]}$. Since the trivial case $\mathbf{v} = \mathbf{0}$, results in $\langle \mathcal{G} \cup \{\mathbf{v}\} \rangle_{[\cdot, \cdot]} = \mathcal{G} \subsetneq \mathbb{F}_2^{2n}$, we assume from now on that $\mathbf{v} \neq \mathbf{0}$. We consider two cases:

- (i) $\text{wt}(\mathbf{v}) < n$,
- (ii) $\text{wt}(\mathbf{v}) = n$.

Before presenting the proofs of these cases, let us note that, along with the relations defining the symplectic set Eq. (A6), we also have that

$$\mathbf{x}_i^\top \Lambda \mathbf{v} \iff i \in \mathcal{Z}(\mathbf{v}), \quad (\text{C9})$$

$$\mathbf{z}_i^\top \Lambda \mathbf{v} \iff i \in \mathcal{X}(\mathbf{v}). \quad (\text{C10})$$

Proof of (i). Suppose that $\mathbf{v} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}$ is such that $\text{wt}(\mathbf{v}) < n$. Then there exists some $j \in [n]$ such that $j \notin \mathcal{X}(\mathbf{v}) \cup \mathcal{Z}(\mathbf{v})$. The proof proceeds by showing that

$$\mathbf{w} := \begin{cases} \mathbf{x}_i + \mathbf{x}_j, & \text{for some } i \in \mathcal{X}(\mathbf{v}) \text{ if } \mathcal{X}(\mathbf{v}) \neq \emptyset \\ \mathbf{z}_i + \mathbf{x}_j, & \text{for some } i \in \mathcal{Z}(\mathbf{v}) \text{ otherwise.} \end{cases} \quad (\text{C11})$$

Note the case where both $\mathcal{X}(\mathbf{v})$ and $\mathcal{Z}(\mathbf{v})$ are empty is disallowed by the assumption that $\mathbf{v} \neq \mathbf{0}$. In the following, we assume $\mathcal{X}(\mathbf{v})$ is nonempty, but the case where only $\mathcal{Z}(\mathbf{v})$ is nonempty proceeds analogously by replacing $\mathbf{x}_i$ by $\mathbf{z}_i$.

Suppose for a contradiction that $\mathbf{x}_i + \mathbf{x}_j \in \langle \mathcal{G} \cup \{\mathbf{v}\} \rangle_{[\cdot, \cdot]}$. This means there exists a sequence of elements $\mathbf{u}_1, \dots, \mathbf{u}_r \in \mathcal{G} \cup \{\mathbf{v}\}$ such that

$$\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) = \mathbf{x}_i + \mathbf{x}_j. \quad (\text{C12})$$

Since $\mathbf{x}_i + \mathbf{x}_j \notin \mathcal{G} \cup \{\mathbf{v}\}$, it must be that $r \geq 2$. Moreover, since $\mathbf{x}_i, \mathbf{x}_j$ are linearly independent, it must be that $\mathbf{x}_i + \mathbf{x}_j \neq \mathbf{0}$. For $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) \neq \mathbf{0}$ to hold, it must be that

$$\text{ad}_{\mathbf{u}_t} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) = \mathbf{u}_t + \cdots + \mathbf{u}_r \neq \mathbf{0} \quad (\text{C13})$$

for all $t = 1, \dots, r-1$. Furthermore, for $\mathbf{u}_1 + \cdots + \mathbf{u}_r$ to equal $\mathbf{x}_i + \mathbf{x}_j$, it must be the case that there is some $l \in \{1, \dots, r\}$ such that $\mathbf{u}_l = \mathbf{x}_j$. However, since for $\mathbf{u} \in \mathcal{G} \cup \{\mathbf{v}\}$, $\mathbf{x}_j^\top \Lambda \mathbf{u} = 1$ if and only if $\mathbf{u} = \mathbf{z}_j$, it follows that the only

possible sequences $\mathbf{u}_1, \dots, \mathbf{u}_r$ containing $\mathbf{x}_j$ and satisfying $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) \neq 0$ for all $t = 1, \dots, r-1$ are of the form

$$\mathbf{u}_1, \dots, \mathbf{u}_r = \begin{cases} \dots, \mathbf{x}_j, \mathbf{x}_j, \mathbf{z}_j, \mathbf{z}_j, \mathbf{x}_j, \mathbf{x}_j, \mathbf{z}_j, \mathbf{z}_j \\ \dots, \mathbf{z}_j, \mathbf{z}_j, \mathbf{x}_j, \mathbf{x}_j, \mathbf{z}_j, \mathbf{z}_j, \mathbf{x}_j, \mathbf{z}_j. \end{cases} \quad (\text{C14})$$

That is, the sequence is comprised only of $\mathbf{x}_j$ and $\mathbf{z}_j$, and only in a specific alternating form. In any case, the resultant adjoint sequence is

$$\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) = \begin{cases} \mathbf{x}_j \\ \mathbf{z}_j \\ \mathbf{x}_j + \mathbf{z}_j, \end{cases} \quad (\text{C15})$$

none of which equal $\mathbf{x}_i + \mathbf{x}_j$, providing the desired contradiction.

Proof of (ii). Now suppose that $\mathbf{v}$ is such that $\text{wt}(\mathbf{v}) = n$. The proof proceeds by demonstrating the claim that $\text{wt}(\mathbf{u}_1, \dots, \mathbf{u}_r)$ is odd for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_r \in \mathcal{G} \cup \{\mathbf{v}\}$ such that $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) \neq 0$. This proves case (ii) since any $\mathbf{w} \in \mathbb{F}_2^{2n}$ with $\text{wt}(\mathbf{w})$ even is such that $\mathbf{w} \notin (\mathcal{G} \cup \{\mathbf{v}\})_{[\cdot, \cdot]}$.

We prove the claim by induction on the length of the sequence, and by making use of Lemma C1. Since n is odd, the elements of $\mathcal{G} \cup \{\mathbf{v}\}$ all have odd weight, so the base case of sequences of length 1 holds true. Suppose that $\text{wt}(\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r))$ is odd for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_r \in \mathcal{G} \cup \{\mathbf{v}\}$ such that $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) \neq 0$ and let $\mathbf{u} \in \mathcal{G} \cup \{\mathbf{v}\}$. Since $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) \neq 0$, we have that $\text{ad}_{\mathbf{u}_1} \cdots \text{ad}_{\mathbf{u}_{r-1}}(\mathbf{u}_r) = \mathbf{u}_1 + \cdots + \mathbf{u}_r$, and hence also that $\text{wt}(\mathbf{u}_1 + \cdots + \mathbf{u}_r)$ is odd by the assumption. There are two cases to consider: (a) $\mathbf{u} = \mathbf{x}_i$ or $\mathbf{u} = \mathbf{z}_i$ for some $i \in [n]$ or (b) $\mathbf{u} = \mathbf{v}$. Since the claim is concerned with adjoint sequences that are nonzero, let us assume that $\mathbf{u}^\top \Lambda(\mathbf{u}_1 + \cdots + \mathbf{u}_r) = 1$.

For (a), if $\mathbf{u} = \mathbf{x}_i$, then $\mathbf{u}^\top \Lambda(\mathbf{u}_1 + \cdots + \mathbf{u}_r) = 1$ if and only if $\mathbf{z}_i$ appears in the sequence an odd number of times [including in the expansion of any $\mathbf{u}_l = \mathbf{v}$ using Eq. (A8)]. Similarly, if $\mathbf{u} = \mathbf{z}_i$, then $\mathbf{u}^\top \Lambda(\mathbf{u}_1 + \cdots + \mathbf{u}_r) = 1$ if and only if $\mathbf{x}_i$ appears in the sequence an odd number of times. In either case, if $\mathbf{u}^\top \Lambda(\mathbf{u}_1 + \cdots + \mathbf{u}_r) = 1$, then $\text{wt}(\mathbf{u} + \mathbf{u}_1 + \cdots + \mathbf{u}_r) = \text{wt}(\mathbf{u}_1 + \cdots + \mathbf{u}_r)$, and hence is odd.

For (b), since we are assuming that $\mathbf{u}^\top \Lambda(\mathbf{u}_1, \dots, \mathbf{u}_r) = 1$, we are in the scenario covered by Lemma C1. Accordingly, we know that

$$\text{wt}(\mathbf{u} + \mathbf{u}_1 + \cdots + \mathbf{u}_r) = n - \text{wt}(\mathbf{u}_1 + \cdots + \mathbf{u}_r) + \sigma \quad (\text{C16})$$

for some odd integer σ . Since n and $\text{wt}(\mathbf{u}_1 + \cdots + \mathbf{u}_r)$ are both odd, the right-hand side of the above expression is guaranteed to be odd. This completes the inductive proof of the claim and thus also of the theorem. ■

APPENDIX D: MEASUREMENT-BASED QUANTUM COMPUTATION

In Sec. IV D of the main text, we presented a family of graph states related to the generating sets presented in Theorem 1, and claimed that this family represents a universal resource for MBQC. Recall that a universal resource for MBQC is a family of states Ψ , such that for any state $|\gamma\rangle$ on n qubits there exists a state $|\varphi\rangle \in \Psi$ on $m \geq n$ qubits such that $|\gamma\rangle$ can be obtained deterministically from $|\varphi\rangle$ via local operations and classical communication (LOCC). For the family $\Psi = \{|G_{n,l}\rangle | n, l \in \mathbb{N}\}$, the first requirement of the

definition of universal resource, namely, that any state $|\gamma\rangle$ can be obtained from some $|G_{n,l}\rangle$, follows from Theorem 1 and the construction of $|G_{n,l}\rangle$ as discussed in the main text. Establishing the remaining requirement, that $|\gamma\rangle$ can be obtained *deterministically*, is the purpose of this appendix. This necessitates a brief review of some background on MBQC, with which we begin.

In MBQC, a computation is specified by the following: (i) a choice of graph describing the graph state for the computation, (ii) designated subsets of vertices of the graph whose qubits represent the input and output of the computation, (iii) a set of single-qubit measurements performed on the graph state, the positive projectors of which produce the desired logical unitary applied to the input. However, since quantum-mechanical measurements are inherently probabilistic in general, there is no guarantee that the positive outcome will be obtained for each measurement. Accordingly, there is a further requirement for performing the desired unitary operation with certainty, namely, (iv) that there is an ordering of the measurements and a method for conditionally adapting them if a negative measurement outcome occurs, so that the overall result is the same logical unitary operation. This method is based on the properties of the stabilizers of the graph state being considered, which we now briefly review.

Let $|G\rangle$ denote a graph state corresponding to the simple, connected graph G with vertex set V and edge set E . For each $v \in V$, the graph state $|G\rangle$ satisfies the equation

$$X_v \bigotimes_{v' \in N_v^G} Z_{v'} |G\rangle = |G\rangle, \quad (\text{D1})$$

$\underbrace{\qquad\qquad\qquad}_{=:K_v}$

where N_v^G denotes the neighborhood of v in G . As a consequence of this equation, it is possible to “correct” for the occurrence of a negative measurement outcome by the following reasoning: Suppose the qubit corresponding to vertex w is measured in the basis $\{|+\theta\rangle, |-\theta\rangle\}$, i.e., it is measured in the XY plane of the Bloch sphere. Let $v \in V$ be a neighbor of w in G . Then, since $|-\theta\rangle = Z|+\theta\rangle$, we have that

$$\langle -\theta |_w |G\rangle = \langle +\theta |_w Z_w |G\rangle = \langle +\theta |_w X_v \bigotimes_{v' \in N_v^G \setminus w} Z_{v'} |G\rangle. \quad (\text{D2})$$

In this case, we see that we have effectively obtained the desired measurement outcome on w at the cost of needing to apply the remaining operations of K_v . However, if the measurements on the qubits that would receive these operations are suitably chosen and yet to be performed, these operations can be effected by changing the measurement bases appropriately. For example, suppose that qubit v and each qubit $v' \in N_v^G \setminus w$ are to be measured in the XY plane with respect to the bases $\{|+\theta_v\rangle, |-\theta_v\rangle\}$ and $\{|+\theta_{v'}\rangle, |-\theta_{v'}\rangle\}$ for $v' \in N_v^G \setminus w$. Using that $X|\pm\theta\rangle = |\pm\theta\rangle$ and $Z|\pm\theta\rangle = |\pm\theta + \pi\rangle$, we see that

$$\begin{aligned} & \langle -\theta_w |_w \langle +\theta_v |_v \bigotimes_{v' \in N_v^G \setminus w} \langle +\theta_{v'} |_{v'} |G\rangle \\ &= \langle +\theta_w |_w \langle +(-\theta_v) |_{-v} \bigotimes_{v' \in N_v^G \setminus w} \langle +(\theta_{v'} + \pi) |_{v'} |G\rangle. \end{aligned} \quad (\text{D3})$$

This example has assumed measurements in the XY plane of the Bloch sphere, however measurements in the XZ- and YZ-plane are also possible, by that analogous reasoning (for measurements in the YZ plane, the positive and negative projectors differ by the application of an X , while those of measurements in the XZ plane differ by a Y ; in the latter case, this requires a product of the operators K_v). In each case, there are common features of the correcting process, namely, that (1) the negative outcome is “corrected” by “completing” a stabilizer K_v (or product thereof) on other qubits of the graph state, and (2) that the measurements on those qubits are yet to be performed.

So, to ensure that a computation can proceed with certainty on a given graph state, we need to find a sequence of measurements such that *every* measurement can be corrected by the above method. It turns out that whether this is possible is a property of the graph underlying the graph state in conjunction with the assignment of planes of the Bloch sphere (i.e., XY, XZ, and YZ) to each of the qubits, indicating which type of measurement is to be performed there. If it is possible to correct for every measurement, then the graph is said to have gflow [43], which is characterized in the following definition:

Definition 1. Let $G = (V, E)$ be a graph, I and O be input and output subsets of V respectively, and $\omega : V \setminus O \rightarrow \{XY, XZ, YZ\}$ be a map assigning measurement planes to qubits. The tuple (G, I, O, ω) has *gflow* if there exists a map $g : V \setminus O \rightarrow 2^{V \setminus I}$, where $2^{V \setminus I}$ denotes the powerset of $V \setminus I$, and a partial order over V such that the following hold for all $v \in V \setminus O$:

- (1) if $v' \in g(v)$ and $v' \neq v$, then $v < v'$;
- (2) if $v' \in \text{Odd}(g(v))$ and $v' \neq v$, then $v < v'$;
- (3) if $\omega(v) = XY$, then $v \notin g(v)$ and $v \in \text{Odd}(g(v))$;
- (4) if $\omega(v) = XZ$, then $v \in g(v)$ and $v \in \text{Odd}(g(v))$;
- (5) if $\omega(v) = YZ$, then $v \in g(v)$ and $v \notin \text{Odd}(g(v))$;

where $\text{Odd}(K) := \{\tilde{v} \in V : |N_{\tilde{v}}^G \cap K| = 1 \bmod 2\}$ for any $K \subseteq V$.

In effect, the map g specifies the stabilizer or product of stabilizers corresponding to the correction for each qubit, i.e., the correction occurs by completing $\prod_{v' \in g(v)} K_{v'}$ for each v .

It was shown in Ref. [43] that the presence of gflow is a necessary and sufficient condition for ensuring determinism in MBQC. Accordingly, to finish demonstrating that the family of graph states presented in the main text is a valid universal resource, it suffices to show that each member graph state has gflow. In general, there are many different choices for gflow for a given graph state, which has led to, e.g., the development of blind quantum computing protocols [44], albeit ones with imperfect security guarantees [45]. For our purposes, we need only specify one choice of gflow for each member graph state, which we turn to now.

Consider Fig. 5, which depicts the same graph states $|G_{n,l}\rangle$ as Fig. 4 in the main text but now with labels assigned to the vertices. Recall that each vertex colored red is to be measured in the YZ plane of the Bloch sphere, while every vertex colored blue is to be measured in the XY plane. We will consider the leftmost column of blue vertices as the input set and the rightmost column of blue vertices as the output set. We can define a gflow $|G_{n,l}\rangle$ with partial order given by the lexicographical order of the labels and with map g defined by sending each vertex v colored red to the set $\{v\}$ and every vertex w colored blue to the set $\{w'\}$ where w' is the nearest neighbor of w to the right. The remainder of this appendix defines this gflow formally.

We are considering the graph $G_{n,l}$ with vertex set V given by

$$V = \begin{cases} \{1, \dots, l(2n+1)\} & \text{if } n \equiv 0 \pmod{2} \\ \{1, \dots, l(2n+2)\} & \text{if } n \equiv 1 \pmod{2}. \end{cases} \quad (\text{D4})$$

We can specify the edge set E by stating the sets of neighbors for each vertex, which is more convenient for verifying the choice of partial order and mappings defined below are valid gflows. For n even, we have that

$$N_v^G = \begin{cases} \{v+1, \dots, v+n\}, & \text{if } v = j(2n+1)+1 \text{ for some } j \in \{0, \dots, l-1\} \\ \{1, v+n\} & \text{if } v = i \text{ for } i \in \{2, \dots, n+1\} \\ \{v-n-1, j(2n+1)+1, v+n\} & \text{if } v = j(2n+1)+i \text{ for } j \in \{1, \dots, l-1\}, i \in \{2, \dots, n+1\} \\ \{v-n\} & \text{if } v = (l-1)(2n+1)+i \text{ for } i \in \{n+2, \dots, 2n+1\} \\ \{v-n, v+n+1\}, & \text{if } v = j(2n+1)+i \text{ for } j \in \{0, \dots, l-2\}, i \in \{n+2, \dots, 2n+1\}. \end{cases} \quad (\text{D5})$$

For n odd, we have that

$$N_v^G = \begin{cases} \{v+2, \dots, v+1+\frac{n+1}{2}\}, & \text{if } v = j(2n+2)+1 \text{ for } j \in \{0, \dots, l-1\} \\ \{v+1+\frac{n+1}{2}, \dots, v+n\}, & \text{if } v = j(2n+2)+2 \text{ for } j \in \{0, \dots, l-1\} \\ \{1, v+n\}, & \text{if } v = i \text{ for } i \in \{3, \dots, 1+\frac{n+1}{2}\} \\ \{2, v+n\}, & \text{if } v = i \text{ for } i \in \{3+\frac{n+1}{2}, \dots, n+2\} \\ \{1, 2, v+n\}, & \text{if } v = 2+\frac{n+1}{2} \\ \{v-n-2, j(2n+2)+1, v+n\}, & \text{if } v = j(2n+2)+i \text{ for } j \in \{1, \dots, l-1\}, i \in \{3, \dots, 1+\frac{n+1}{2}\} \\ \{v-n-2, j(2n+2)+2, v+n\}, & \text{if } v = j(2n+2)+i \text{ for } j \in \{1, \dots, l-1\}, i \in \{3+\frac{n+1}{2}, \dots, n+2\} \\ \{v-n-2, j(2n+2)+1, j(2n+2)+2, v+n\}, & \text{if } v = j(2n+2)+2+\frac{n+1}{2} \text{ for } j \in \{1, \dots, l-1\} \\ \{v-n\}, & \text{if } v = (l-1)(2n+2)+i \text{ for } i \in \{n+3, \dots, 2n+2\} \\ \{v-n, v+n+2\}, & \text{if } v = j(2n+2)+i \text{ for } j \in \{0, \dots, l-2\}, i \in \{n+3, \dots, 2n+2\}. \end{cases} \quad (\text{D6})$$

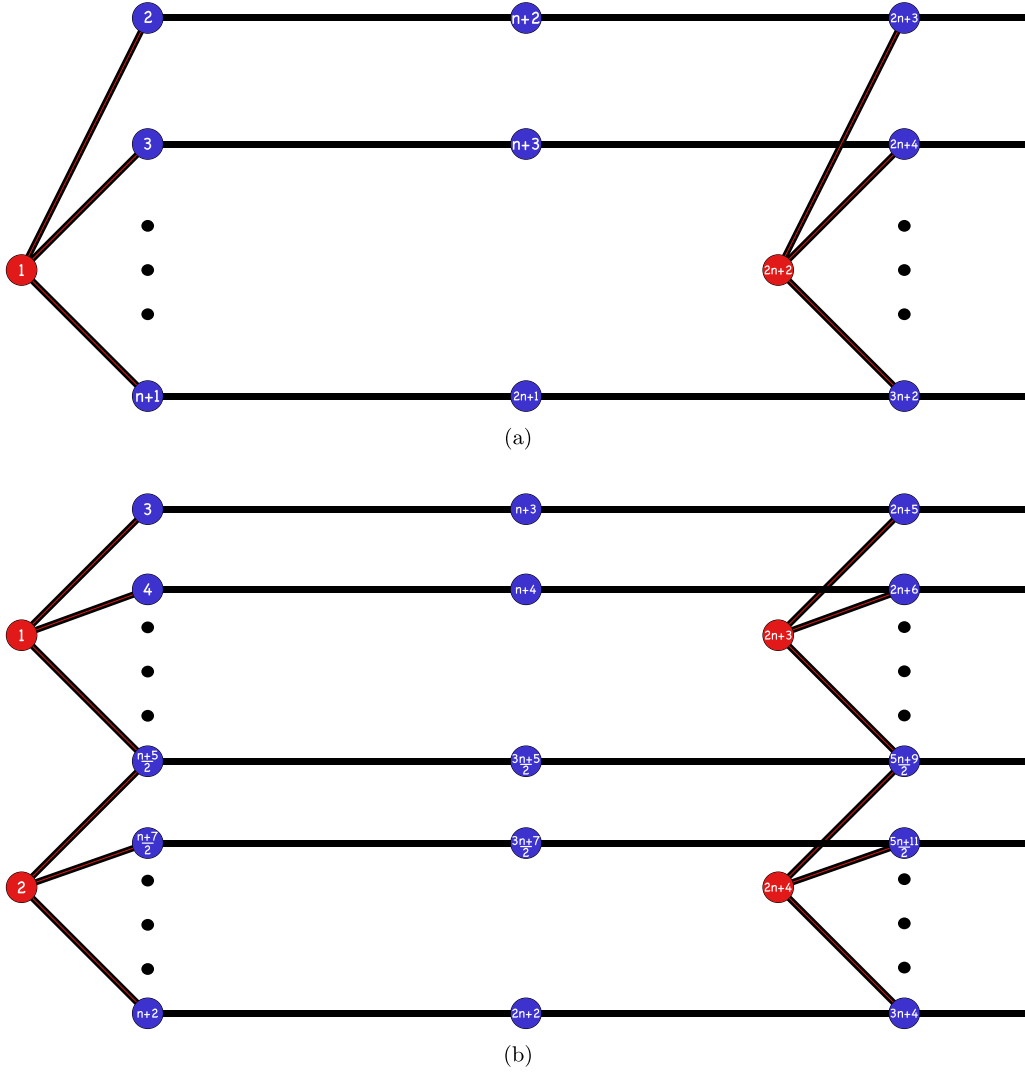


FIG. 5. This figure depicts fragments of the graph states $|G_{n,l}\rangle$ that form the universal resource introduced in the main text but with vertices labeled to aid the definition of gflow presented in this appendix. Panel (a) depicts $|G_{n,l}\rangle$ for n even and panel (b) depicts $|G_{n,l}\rangle$ for n odd.

The input and output sets are defined to be

$$I = \begin{cases} \{2, \dots, n+1\} & \text{if } n = 0 \bmod 2 \\ \{3, \dots, n+2\} & \text{if } n = 1 \bmod 2, \end{cases} \quad (\text{D7})$$

and

$$O = \begin{cases} \{(l-1)(2n+1) + n+2, \dots, l(2n+1)\}, & \text{if } n = 0 \bmod 2 \\ \{(l-1)(2n+2) + n+3, \dots, l(2n+2)\}, & \text{if } n = 1 \bmod 2. \end{cases} \quad (\text{D8})$$

In the case where n is even, the map ω that specifies the planes of measurement is defined by $\omega(v) = \text{YZ}$ if $v = j(2n+1) + 1$ for some $j \in \{0, \dots, l-1\}$, with all other vertices being assigned XY. Similarly, in the case where n is odd, ω is defined by $\omega(v) = \text{YZ}$ if $v = j(2n+2) + 1$ or $v = j(2n+2) + 2$ for some $j \in \{0, \dots, l-1\}$, and $\omega(v) = \text{XY}$ otherwise.

In both cases, the partial order on vertices can be taken to be the order of the corresponding labels, i.e., $1 < 2 < \dots < l(2n+1)$ in the even n case and $1 < 2 < \dots < l(2n+2)$ in the odd case. We define the map g_{even} via

$$g_{\text{even}}(v) = \begin{cases} \{v\}, & \text{if } v = j(2n+1) + 1 \text{ for some } j \in \{0, \dots, l-1\} \\ \{v+n\}, & \text{if } v = j(2n+1) + i \text{ for } j \in \{0, \dots, l-1\}, i \in \{2, \dots, n+1\} \\ \{v+n+1\}, & \text{if } v = j(2n+1) + i \text{ for } j \in \{0, \dots, l-1\}, i \in \{n+2, \dots, 2n+1\}. \end{cases} \quad (\text{D9})$$

Similarly, the map g_{odd} is defined via

$$g_{\text{even}}(v) = \begin{cases} \{v\}, & \text{if } v = j(2n+2) + i \text{ for } j \in \{0, \dots, l-1\}, i \in \{1, 2\} \\ \{v+n\}, & \text{if } v = j(2n+2) + i \text{ for } j \in \{0, \dots, l-1\}, i \in \{3, \dots, n+2\} \\ \{v+n+1\}, & \text{if } v = j(2n+2) + i \text{ for } j \in \{0, \dots, l-1\}, i \in \{n+3, \dots, 2n+2\}. \end{cases} \quad (\text{D10})$$

Since both g_{even} and g_{odd} assign singleton sets to each vertex, it follows that for each v , $\text{Odd}(g_{\text{even}}(v))$ (resp. $\text{Odd}(g_{\text{odd}}(v))$) is just the sets of neighbors of the vertex in $g_{\text{even}}(v)$ [$g_{\text{odd}}(v)$]. By observing Eqs. (D5) and (D6), it can be verified that both g_{even} and g_{odd} satisfy the requirements of Definition 1 for the choice of partial order outlined above. For example, for n even, we see that the elements of $g_{\text{even}}(v) \setminus v$ and $N_{g_{\text{even}}(v)}^G \setminus v$ all have labels greater than v and hence are later than v in

the choice of partial order, meaning that conditions 1 and 2 are satisfied. For $v = j(2n+1) + 1$ with $j \in \{0, \dots, l-1\}$, we have that $\omega(v) = YZ$ and moreover that $g_{\text{even}}(v) = \{v\}$ and that $\text{Odd}(g_{\text{even}}(v)) = \{v+1, \dots, v+n\}$, so condition 5 holds. For all other v , we have that $\omega(v) = XY$ and that $g_{\text{even}}(v)$ contains a neighbor of v meaning that $v \notin g_{\text{even}}(v)$ but $v \in \text{Odd}(g_{\text{even}}(v))$ so condition 3 is satisfied. The reasoning for g_{odd} proceeds analogously.

-
- [1] W. Lechner, P. Hauke, and P. Zoller, A quantum annealing architecture with all-to-all connectivity from local interactions, *Sci. Adv.* **1**, e1500838 (2015).
 - [2] M. Fellner, A. Messinger, K. Ender, and W. Lechner, Universal parity quantum computing, *Phys. Rev. Lett.* **129**, 180503 (2022).
 - [3] M. Lanthaler, C. Dłaska, K. Ender, and W. Lechner, Rydberg-blockade-based parity quantum optimization, *Phys. Rev. Lett.* **130**, 220601 (2023).
 - [4] K. Ender, R. ter Hoeven, B. E. Niehoff, M. Drieb-Schön, and W. Lechner, Parity quantum optimization: Compiler, *Quantum* **7**, 950 (2023).
 - [5] M. Drieb-Schön, K. Ender, Y. Javanmard, and W. Lechner, Parity quantum optimization: Encoding constraints, *Quantum* **7**, 951 (2023).
 - [6] M. Fellner, K. Ender, R. ter Hoeven, and W. Lechner, Parity quantum optimization: Benchmarks, *Quantum* **7**, 952 (2023).
 - [7] I. D. Smith, M. Cautrès, D. T. Stephen, and H. P. Nautrup, Optimally generating $\mathfrak{su}(2^N)$ using Pauli strings, *Phys. Rev. Lett.* **134**, 200601 (2025).
 - [8] B. Klaver, S. Rombouts, M. Fellner, A. Messinger, K. Ender, K. Ludwig, and W. Lechner, Swap-less implementation of quantum algorithms, [arXiv:2408.10907](https://arxiv.org/abs/2408.10907).
 - [9] IBM Quantum, <https://quantum.ibm.com/>, accessed: 2025-02-05.
 - [10] I. D. Smith, H. P. Nautrup, and H. J. Briegel, Parity quantum computing as yz-plane measurement-based quantum computing, *Phys. Rev. Lett.* **132**, 220602 (2024).
 - [11] R. Raussendorf and H. J. Briegel, A one-way quantum computer, *Phys. Rev. Lett.* **86**, 5188 (2001).
 - [12] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, Measurement-based quantum computation, *Nat. Phys.* **5**, 19 (2009).
 - [13] R. Raussendorf and H. Briegel, Computational model underlying the one-way quantum computer, *Quant. Inf. Comp.* **6**, 433 (2002).
 - [14] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, *Phys. Rev. A* **68**, 022312 (2003).
 - [15] M. Van den Nest, A. Miyake, W. Dür, and H. J. Briegel, Universal resources for measurement-based quantum computation, *Phys. Rev. Lett.* **97**, 150504 (2006).
 - [16] D. D'Alessandro, *Introduction to Quantum Control and Dynamics*, Chapman & Hall/CRC Applied Mathematics & Nonlinear Science (Taylor & Francis, New York, 2007).
 - [17] G. M. Huang, T. J. Tarn, and J. W. Clark, On the controllability of quantum-mechanical systems, *J. Math. Phys.* **24**, 2608 (1983).
 - [18] F. Albertini and D. D'Alessandro, Notions of controllability for quantum mechanical systems, in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)* (IEEE, New York, 2001), Vol. 2, pp. 1589–1594.
 - [19] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
 - [20] Brian C. Hall, *Lie groups, Lie Algebras, and Representations* (Springer, Switzerland, 2013).
 - [21] M. Fellner, A. Messinger, K. Ender, and W. Lechner, Applications of universal parity quantum computation, *Phys. Rev. A* **106**, 042442 (2022).
 - [22] A. Messinger, M. Fellner, and W. Lechner, Constant depth code deformations in the parity architecture, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, New York, 2023), Vol. 1, pp. 120–130.
 - [23] G. Aguilar, S. Cichy, J. Eisert, and L. Bittel, Full classification of Pauli Lie algebras, [arXiv:2408.00081](https://arxiv.org/abs/2408.00081).
 - [24] E. Bäumer and S. Woerner, Measurement-based long-range entangling gates in constant depth *Phys. Rev. Res.* **7**, 023120 (2025).
 - [25] Z. Chen, A. Megrant, J. Kelly, R. Barends, J. Bochmann, Y. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, J. Mutus *et al.*, Fabrication and characterization of aluminum airbridges for superconducting microwave circuits, *Appl. Phys. Lett.* **104**, 052602 (2014).
 - [26] A. Dunsworth, R. Barends, Y. Chen, Z. Chen, B. Chiaro, A. Fowler, B. Foxen, E. Jeffrey, J. Kelly, P. Klimov *et al.*, A method for building low loss multi-layer wiring for superconducting microwave devices, *Appl. Phys. Lett.* **112**, 063502 (2018).

- [27] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, *Phys. Rev. A* **93**, 060302(R) (2016).
- [28] H. Paik, A. Mezzacapo, M. Sandberg, D. T. McClure, B. Abdo, A. D. Córcoles, O. Dial, D. F. Bogorin, B. L. T. Plourde, M. Steffen, A. W. Cross, J. M. Gambetta, and J. M. Chow, Experimental demonstration of a resonator-induced phase gate in a multiqubit circuit-qed system, *Phys. Rev. Lett.* **117**, 250502 (2016).
- [29] D. E. Browne and H. J. Briegel, One-way quantum computation—a tutorial introduction, [arXiv:quant-ph/0603226](https://arxiv.org/abs/quant-ph/0603226).
- [30] A. Cowtan, S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah, Phase gadget synthesis for shallow circuits, *Electron. Proc. Theor. Comput. Sci.* **318**, 213 (2020).
- [31] R. R. Ferguson, L. Dellantonio, A. A. Balushi, K. Jansen, W. Dür, and C. A. Muschik, Measurement-based variational quantum eigensolver, *Phys. Rev. Lett.* **126**, 220501 (2021).
- [32] T. N. Kaldenbach and M. Heller, Mapping quantum circuits to shallow-depth measurement patterns based on graph states, *Quantum Sci. Technol.* **10**, 015010 (2025).
- [33] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [34] M. Proietti, F. Cerocchi, and M. Dispenza, Native measurement-based quantum approximate optimization algorithm applied to the max k -cut problem, *Phys. Rev. A* **106**, 022437 (2022).
- [35] W. Lechner, Quantum approximate optimization with parallelizable gates, *IEEE Trans. Quantum Eng.* **1**, 1 (2020).
- [36] A. Weidinger, G. B. Mbeng, M. Fellner, D. Khachatryan, and W. Lechner, Performance of parity QAOA for the signed max-cut problem, [arXiv:2409.14786](https://arxiv.org/abs/2409.14786).
- [37] W. Dür, H. Aschauer, and H.-J. Briegel, Multiparticle entanglement purification for graph states, *Phys. Rev. Lett.* **91**, 107903 (2003).
- [38] H. Aschauer, W. Dür, and H.-J. Briegel, Multiparticle entanglement purification for two-colorable graph states, *Phys. Rev. A* **71**, 012319 (2005).
- [39] H. Poulsen Nautrup and H. J. Briegel, Measurement-based quantum computation from Clifford quantum cellular automata, *Phys. Rev. A* **110**, 062617 (2024).
- [40] D. Gottesman, *Stabilizer Codes and Quantum Error Correction* (California Institute of Technology, 1997).
- [41] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Phys. Rev. A* **70**, 052328 (2004).
- [42] S. Anders and H. J. Briegel, Fast simulation of stabilizer circuits using a graph-state representation, *Phys. Rev. A* **73**, 022334 (2006).
- [43] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix, Generalized flow and determinism in measurement-based quantum computation, *New J. Phys.* **9**, 250 (2007).
- [44] A. Mantri, T. F. Demarie, N. C. Menicucci, and J. F. Fitzsimons, Flow ambiguity: A path towards classically driven blind quantum computation, *Phys. Rev. X* **7**, 031004 (2017).
- [45] I. D. Smith, M. Krumm, L. J. Fiderer, H. P. Nautrup, and H. J. Briegel, The min-entropy of classical-quantum combs for measurement-based applications, *Quantum* **7**, 1206 (2023).