

## Equivalent noise properties of scalable continuous-variable cluster states

Blayne W. Walshe <sup>1,\*</sup>, Rafael N. Alexander <sup>1</sup>, Takaya Matsuura <sup>1</sup>, Ben Q. Baragiola <sup>1,2</sup> and Nicolas C. Menicucci <sup>1</sup>

<sup>1</sup>Centre for Quantum Computation and Communication Technology, School of Science, RMIT University, Melbourne, Victoria 3000, Australia

<sup>2</sup>Yukawa Institute for Theoretical Physics, Kyoto University, Kitashirakawa Oiwakecho, Sakyo-ku, Kyoto 606-8502, Japan



(Received 20 May 2023; accepted 27 July 2023; published 6 October 2023)

Optical continuous-variable cluster states (CVCSs) in combination with Gottesman-Kitaev-Preskill (GKP) qubits enable fault-tolerant quantum computation so long as these resources are of high enough quality. Previous studies concluded that a particular CVCS, the quad rail lattice (QRL), exhibits a lower GKP gate-error rate than others do. We show in this work that many other experimentally accessible CVCSs also achieve this level of performance by identifying operational equivalences to the QRL. Under this equivalence, the GKP Clifford gate set for each CVCS maps straightforwardly from that of the QRL, inheriting its noise properties. Furthermore, each cluster state has at its heart a balanced four-splitter—the four-mode extension to a balanced beam splitter. We classify all four-splitters, show they form a single equivalence class under SWAP and parity operators, and give a construction of any four-splitter with linear optics, thus extending the toolbox for theoretical and experimental cluster-state design and analysis.

DOI: [10.1103/PhysRevA.108.042602](https://doi.org/10.1103/PhysRevA.108.042602)

### I. INTRODUCTION

Measurement-based quantum computing (QC) with continuous-variable (CV) systems relies on preparing large-scale entangled resource states, called CV cluster states [1,2] on which adaptive measurements are sufficient for universal QC. Optics has shown itself to be an excellent platform for demonstrating these states on a large scale using squeezed vacuum and passive linear optical components [3–9]. These cluster states are accessible with current technology and highly scalable. Indeed, experiments have demonstrated large cluster states [10–12] including two-dimensional (2D) CV cluster states whose connectivity makes them resources for universal quantum computing [13,14].

In measurement-based quantum computing (MBQC) protocols using CV cluster states, performing homodyne detection on local modes teleports quantum information throughout the cluster and applies deterministic Gaussian operations that depend on the measurement bases [15–17]. Hindering quantum computation is inherent noise from the fact that physical states can only be finitely squeezed [18,19]. As measurements are performed, this noise accumulates and eventually overwhelms the intended calculation [20]. As a countermeasure, digital quantum information can be encoded into a Gottesman-Kitaev-Preskill (GKP) error-correcting code [21], a class of bosonic code that interfaces seamlessly with CVCSs [22] because GKP codes admit all-Gaussian Clifford-gate operations and magic-state production [23]. Moreover, GKP codes are resistant to various types of CV errors, including errors from noisy teleportation [21,24,25]. Thus, combining CV cluster states with GKP codes provides a pathway to fault tolerance [26–30].

In order to use the GKP code with a CV cluster state, a specific set of Gaussian operations is required—those that realize GKP Clifford gates. Bespoke methods have been proposed for some CV cluster states [31,32], each involving a set of homodyne measurement angles tailored for each gate and for each cluster state. For some CV cluster states, this includes position-quadrature measurements to delete certain modes [7,31], effectively wasting them while injecting their noise into the remaining modes. Furthermore, some proposals require optical switches to inject GKP states when error correction is needed [13,31].

A detailed study [32] of a specific CV cluster state—the quad-rail lattice (QRL) [4,33]—showed that it serves as an efficient resource for computing with the square-lattice GKP code: the full set of GKP Cliffords can be performed in a single teleportation step without requiring deletions, thus minimizing inherent noise from the squeezed states in the cluster. Further, replacing the squeezed states with a particular type of GKP state called a *qunaught* state [32,34] (also called GKP sensor states [35]) obviates the need for optical switches and automatically performs GKP error correction during gate execution [34]. Together, these properties give the QRL the lowest logical GKP-Clifford gate noise of any known 2D CV cluster state [32]. Lower gate noise places lower demands on the quality of the GKP resources, making the QRL appear to stand out as the CV cluster state of choice in the quest for fault tolerance. But this is not the whole story.

In this work, we show that several other 2D CV cluster states can achieve the same performance as the QRL. Previous analysis indicated that the bilayer square lattice (BSL) [7], the double bilayer square lattice (DBSL) [14], and the modified bilayer square lattice (MBSL) [13] CV cluster states perform worse than the QRL [31]. We analyze these cluster states along with another one—a three-dimensional (3D) extension to the DBSL [28]—and show that each, when modified with

\*blayneyw@gmail.com

an additional beam splitter, is operationally equivalent to the QRL. Moreover, for the BSL and DBSL, the additional beam splitter is not even necessary—it can be added virtually by correlating certain specific homodyne measurement angles and postprocessing the outcomes. This equivalence provides two benefits. First, it gives a recipe to map the measurement angles for GKP Clifford gates from the QRL to the other cluster states. Second, every equivalent CV cluster state inherits the QRL’s low GKP-logical gate noise. Together, these make the BSL, DBSL, and MBSL in principle as useful as the QRL for achieving fault tolerance with the GKP code.

We further show that the CV cluster states above are members of a large class of scalable CV cluster states whose internal coupling uses balanced *four-splitters* [33]—a generalization of a balanced beam splitter to four modes. We characterize all balanced four-splitters and show that they fall into an equivalence class defined by a small set of operations. We provide a procedure to engineer any four-splitter using only linear-optical components—a network of four balanced beam splitters along with SWAP gates and parity operators.

Section II gives properties of beam-splitter networks and introduces quantum optical conventions used throughout this work. Section III shows the equivalence of the BSL, DBSL, MBSL, and extension of the DBSL to the QRL cluster state. Section IV analyzes this equivalence when the CV cluster states are used in conjunction with the GKP code. Finally, Sec. V classifies the equivalence class of four-splitters and shows how to construct them from linear optical components.

## II. BEAM-SPLITTER NETWORKS

A balanced beam splitter mixing modes  $j$  and  $k$  is described by the unitary operator

$$\hat{B}_{jk} := e^{\frac{\pi}{4}(\hat{a}_j \hat{a}_k^\dagger - \hat{a}_j^\dagger \hat{a}_k)} = e^{-i\frac{\pi}{4}(\hat{q}_j \hat{p}_k - \hat{p}_j \hat{q}_k)}, \quad (1)$$

where  $\hat{a}_j$  and  $\hat{a}_k^\dagger$  are mode annihilation and creation operators satisfying  $[\hat{a}_j, \hat{a}_k^\dagger] = \delta_{j,k}$ , and  $\hat{q}_j = \frac{1}{\sqrt{2}}(\hat{a}_j + \hat{a}_j^\dagger)$  and  $\hat{p}_j = \frac{-i}{\sqrt{2}}(\hat{a}_j - \hat{a}_j^\dagger)$  are the position and momentum quadrature operators. The beam splitter above is depicted in a right-to-left circuit diagram as

$$\hat{B}_{jk} = \begin{array}{c} \text{---} j \\ | \\ \text{---} k \end{array} \quad (2)$$

with the arrow pointing from mode  $j$  to mode  $k$ .

A beam splitter generates a linear transformation of the mode operators  $\hat{a}_j$  and  $\hat{a}_k$  that can be described by a unitary,  $2 \times 2$  matrix. For the beam splitter in Eq. (1), this matrix is real and thus orthogonal, and we denote this relationship as

$$\hat{B}_{jk} \rightarrow \mathbf{R}_{jk} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad (3)$$

where the arrow ( $\rightarrow$ ) is to be interpreted as

$$(\hat{U} \rightarrow \mathbf{U}) \implies (\hat{U}^\dagger \hat{a} \hat{U} = \mathbf{U} \hat{a}), \quad (4)$$

where  $\hat{U}$  is a unitary operator representing passive linear-optical elements,  $\hat{a} = (\hat{a}_1, \dots, \hat{a}_N)^\top$  is a column vector of  $N$  annihilation operators, and  $\mathbf{U} \in \mathbf{U}(N)$  is the unitary matrix acting to linearly combine the annihilation operators,

representing this Heisenberg action. (This relation is one-directional because the converse is only true up to an overall phase.) Note that  $\mathbf{R}_{jk}$  may be a submatrix within a larger matrix over more modes.

The complex conjugate of the beam-splitter unitary swaps the indices and results in a transpose of its associated matrix:  $(\hat{B}_{jk}^\dagger = \hat{B}_{kj}) \rightarrow (\mathbf{R}_{jk}^\top = \mathbf{R}_{kj})$ . In a circuit diagram, Eq. (2), the complex conjugate changes the direction of the arrow between the wires.

We refer to a series of beam splitters as a *beam-splitter network*. This term is general and allows any beam splitters and any number of modes. In this work, however, we will be concerned with four-mode beam-splitter networks consisting of identical, balanced beam splitters as defined in Eq. (1). For example, consider a beam-splitter network given by the unitary operator

$$\hat{B}_{\text{network}} = \hat{B}_{24} \hat{B}_{13} \hat{B}_{34} \hat{B}_{12}, \quad (5)$$

which is described by the circuit<sup>1</sup>

$$\hat{B}_{\text{network}} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \quad (6)$$

Note that we use right-to-left circuits in this work, which benefits from a direct mapping to operator notation; compare the circuit to Eq. (5). This beam-splitter network produces a unitary transformation of the mode operators  $\hat{a}_j$  described by an orthogonal matrix  $\mathbf{R}_{\text{network}}$ ,

$$\hat{B}_{\text{network}} \rightarrow \mathbf{R}_{\text{network}} = \mathbf{R}_{24} \mathbf{R}_{13} \mathbf{R}_{34} \mathbf{R}_{12}, \quad (7)$$

where each beam-splitter matrix  $\mathbf{R}_{jk}$  from Eq. (3) has been expanded into the appropriate  $4 \times 4$  matrix over all four modes. Note that nonidentical beam splitters that share a mode do not commute, e.g.,  $[\hat{B}_{jk}, \hat{B}_{k\ell}] \neq 0$ , which is reflected in the fact that their expanded rotation matrices also do not commute,  $[\mathbf{R}_{jk}, \mathbf{R}_{k\ell}] \neq 0$ .

### A. CV SWAP gates and parity operators

Modes can be exchanged using a CV SWAP gate [36]

$$\text{SWAP}_{jk} = \text{SWAP}_{kj} = \begin{array}{c} \text{---} j \\ \times \\ | \\ \times \\ \text{---} k \end{array} \quad (8)$$

These satisfy several useful gate identities; see Eq. (F4). The action of SWAP gates on a beam-splitter network can also be represented by matrix multiplication of network matrices  $\mathbf{R}_{\text{network}}$ ,

$$\text{SWAP}_{jk} \hat{B}_{\text{network}} \rightarrow \mathbf{P}_{jk} \mathbf{R}_{\text{network}}, \quad (9)$$

where  $\mathbf{P}_{jk}$  is a permutation matrix between indices  $j$  and  $k$ . Left multiplication by  $\mathbf{P}_{jk}$  swaps rows  $j$  and  $k$ . Similarly,

<sup>1</sup>The fact that the beam splitters in Eqs. (5) and (6) are ordered in the same way is a feature of our right-to-left circuit notation. When using left-to-right circuits, the ordering in the circuit description is opposite that in the equation. See Ref. [32] for further motivation for choosing this convention.

the right action of a SWAP gate is represented as matrix multiplication from the right by  $\mathbf{P}_{jk}$ , which swaps columns  $j$  and  $k$ .

The single-mode parity operator is defined as

$$\hat{F}^2 := e^{i\pi\hat{a}^\dagger\hat{a}}, \quad (10)$$

expressed here as two applications of the Fourier transform  $\hat{F}$ . Parity transformations flip the sign of a quadrature operator, i.e.,  $\hat{F}^2\hat{q}\hat{F}^2 = -\hat{q}$ . The left action of a parity operator on a beam splitter gives

$$\hat{F}_j^2\hat{\mathbf{B}}_{\text{network}} \rightarrow \mathbf{M}_j\mathbf{R}_{\text{network}}, \quad (11)$$

where  $\mathbf{M}_j$  is a diagonal matrix with elements equal to 1 except entry  $(j, j)$ , which is  $-1$ . This flips the sign of the  $j$ th row in  $\mathbf{R}_{\text{network}}$ . Similarly, the right action of a parity operator gives matrix multiplication from the right by  $\mathbf{M}_j$ , which flips the sign of the  $j$ th column.

Both SWAP gates and parity operators can reverse the direction of a beam splitter; see Eq. (F2).

### B. Quadrature measurements

In CV cluster-state quantum computing, modes are usually measured via homodyne detection. Adjusting the phase of the local oscillator allows measurements of any quadrature,

$$\hat{p}_\theta := \hat{R}^\dagger(\theta)\hat{p}\hat{R}(\theta) = \hat{p}\cos\theta + \hat{q}\sin\theta, \quad (12)$$

where

$$\hat{R}(\theta) := e^{i\theta\hat{a}^\dagger\hat{a}} \quad (13)$$

is the phase-delay operator (also called the rotation operator). The eigenstates  $|t\rangle_{p_\theta} = \hat{R}^\dagger(\theta)|t\rangle_p$  satisfy  $\hat{p}_\theta|t\rangle_{p_\theta} = t|t\rangle_{p_\theta}$ . In a right-to-left circuit, measurement of a quadrature  $\hat{p}_\theta$  with outcome  $m$  is indicated via a projection onto the bra  ${}_{p_\theta}\langle m| = {}_p\langle m|\hat{R}(\theta)$ , or in circuit form

$${}_{p_\theta}\langle m| \text{---} = {}_p\langle m| \text{---} \boxed{R(\theta)} \text{---}. \quad (14)$$

Momentum measurements are given by  $\theta = 0$ , and position measurements by  $\theta = \frac{\pi}{2}$ . Also, a parity operator  $\hat{F}^2 = \hat{R}(\pi)$  in Eq. (10) serves only to change the sign of the measurement outcome:

$${}_{p_\theta}\langle m|\hat{F}^2 = {}_{p_\theta+\pi}\langle m| = -{}_{p_\theta}\langle m| = {}_{p_\theta}\langle -m|, \quad (15)$$

where the subscript “ $-p_\theta$ ” represents a measurement of the operator  $-\hat{p}_\theta$ .

### C. Adding and removing beam splitters virtually by restricting homodyne angles

The CV cluster states in Sec. III have at their heart beam-splitter networks over four modes. However, only the QRL uses four beam splitters; the others use only three. The equivalences to the QRL involve inserting the missing fourth beam splitter into the other cluster state’s beam-splitter network.

We show here that in some cases, the missing beam splitter can be inserted virtually simply by correlating homodyne measurement angles. Moreover, by the same process, an existing beam splitter can be effectively deleted. The action of a beam splitter on two eigenstates of the same quadrature is

$$\hat{\mathbf{B}}_{21}|t\rangle_{p_\theta} \otimes |s\rangle_{p_\theta} = \left| \frac{1}{\sqrt{2}}(s+t) \right\rangle_{p_\theta} \otimes \left| \frac{1}{\sqrt{2}}(s-t) \right\rangle_{p_\theta}. \quad (16)$$

Taking the Hermitian conjugate of this expression (recall  $\hat{\mathbf{B}}_{21}^\dagger = \hat{\mathbf{B}}_{12}$ ) gives

$${}_{p_\theta}\langle t| \otimes {}_{p_\theta}\langle s| \hat{\mathbf{B}}_{12} = \left\langle \frac{1}{\sqrt{2}}(s+t) \right|_{p_\theta} \otimes \left\langle \frac{1}{\sqrt{2}}(s-t) \right|_{p_\theta}. \quad (17)$$

We can interpret this expression in terms of quadrature measurements, as described in Sec. II B, on both modes. This allows us to account for the effects of a beam splitter before measurements of the same quadrature by classically using the proper linear combinations of the measurement outcomes. Doing so effectively deletes the beam splitter from the circuit. Similarly, given measurements of two modes in the same quadrature with no beam splitter preceding them, a “virtual beam splitter” can be added to a circuit if the outcomes are processed similarly. Note that inserting or deleting beam splitters *does not* depend on the measurement outcomes, only on the fact that the measured quadratures are the same.

The circuit to remove a beam splitter before identical quadrature measurements is

$$\begin{array}{c} {}_{p_\theta}\langle m_1| \text{---} \\ \downarrow \\ {}_{p_\theta}\langle m_2| \text{---} \end{array} = \begin{array}{c} \left\langle \frac{1}{\sqrt{2}}(m_1+m_2) \right| \text{---} \\ \left\langle \frac{1}{\sqrt{2}}(m_2-m_1) \right| \text{---} \end{array}, \quad (18)$$

and to add a virtual beam splitter is

$$\begin{array}{c} {}_{p_\theta}\langle m_1| \text{---} \\ {}_{p_\theta}\langle m_2| \text{---} \end{array} = \begin{array}{c} \left\langle \frac{1}{\sqrt{2}}(m_1-m_2) \right| \text{---} \\ \left\langle \frac{1}{\sqrt{2}}(m_2+m_1) \right| \text{---} \end{array}. \quad (19)$$

In both cases above, the physical operation is on the left of the equal sign, and the circuit to which it is equivalent is on the right.

Note that the insertion and deletion of a beam splitter is accompanied by an orthogonal transformation on the measurement outcomes. Collecting the outcomes into a vector  $\mathbf{m} = (m_1, m_2)^\top$ , removing a beam splitter is accompanied by the transformation  $\mathbf{m} \mapsto \mathbf{R}^\top \mathbf{m}$  and inserting one by  $\mathbf{m} \mapsto \mathbf{R} \mathbf{m}$ .

### III. EQUIVALENCE OF BEAM-SPLITTER CONSTRUCTED CLUSTER STATES

The quad-rail lattice (QRL) cluster state, introduced in Ref. [4], is a resource for universal, fault-tolerant quantum computing. In this section we briefly describe the construction of several types of scalable CV cluster states and show that they are equivalent to the QRL given certain restrictions. We consider the bilayer square lattice (BSL) [7], the double bilayer square lattice (DBSL) [14], a 3D extension to the DBSL

[28], and the modified bilayer square lattice (MBSL) [13]. The work by Larsen *et al.* [31] provides further details for each of the cluster states discussed here. The equivalences we show below are the link that let us apply the detailed analysis of the QRL in previous work [32] to the other cluster states under consideration.

The construction the cluster states considered in this work begins with the preparation of two-mode squeezed states (TMSSs), generated by sending pairs of squeezed vacuum states through beam splitters. Differences arise from different ways of stitching the TMSSs together with additional beam splitters. Stitching involves four-mode entangling operations, in which one half of four TMSSs are coupled together with beam splitters. For each cluster state, equivalence to the QRL involves inserting an additional beam splitter in this process; doing so effectively gives a new, distinct cluster state with more connectivity between modes: we call these the *completed* version of the original cluster state, i.e., the completed BSL (cBSL). When used for CV quantum computing, all of the modes in the cluster state are measured via homodyne detection. We use this fact and show that the needed beam splitter can be inserted *virtually* by restricting specific measurement angles, which allows the unmodified cluster states to be used as if they were a different cluster state with more connectivity. Under this restriction, we refer to a cluster state as *virtually completed*, i.e., the virtually completed BSL (vcBSL). The exception in the cluster states considered here is the MSBL, for which the missing beam splitter cannot be inserted virtually.

### A. Representations of CV cluster states

Large-scale CV cluster states have a complicated entanglement structure spanning many modes, so representing them is a challenge in itself. The graphical calculus for Gaussian pure states [37] offers a visual representation of these states as graphs that faithfully encode their precise wave functions. A simplified version of this graphical calculus is developed in Ref. [4] that is applicable for a large class of scalable CV cluster states, which includes all of the ones examined in this work. It streamlines the original formalism while maintaining the faithful representation of the full state. It also introduces the notation of an arrow between nodes for a beam splitter between the corresponding modes, as well as the simultaneous use of both graphs and arrows to illustrate the steps in the construction of a cluster state.

The *macronode wire* [18] is the primitive unit we start with when representing each of these cluster states. It is the basic highway down which quantum information is channeled, and our work within this paper builds upon the details of teleportation that we established in Ref. [34] and extended to two modes for the QRL in Ref. [32]. Figure 1 shows several representations of the macronode wire: Figs. 1(a) and 1(b) show a series of modes coupled by beam splitters (black beam splitters apply before the red ones), Fig. 1(c) shows a hybrid representation where two-mode squeezed states are coupled by beam splitters, and Fig. 1(d) a simplified graph.

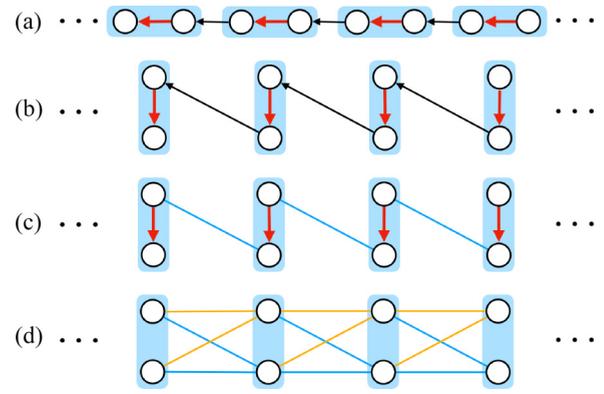


FIG. 1. Four representations of the same macronode wire. Black and red beam splitters, represented by arrows, couple modes prepared in squeezed states. The black beam splitters are applied first to generate TMSSs, then the red beam splitters couple one half of neighboring TMSSs into a macronode wire. (a) Representation showing the modes coupled by beam splitters. (b) Folding the macronode wire to emphasize two-mode macronode groupings, indicated in blue. (c) Black beam splitters, acting first, generate TMSSs between modes, each represented as two modes with a blue edge between them. Pairs of TMSSs are coupled by red beam splitters. (d) The action of both black and red beam splitters in graphical form, revealing the entanglement structure of the macronode wire [4].

For the mathematical details of this representation, the reader may consult Ref. [4], but many readers may be satisfied thinking of the graphs (in blue and yellow) as representing the entanglement structure of a multimode Gaussian state, upon which beam splitters act in the locations and order prescribed. If the node being acted upon is isolated (not a graph, just a node), one may think of this node as representing a squeezed state. Thus, the macronode wire above is a collection of squeezed states, upon which the black beam splitters act to create a collection of TMSSs, followed by another set of beam splitters (the red ones) that link these TMSSs into a macronode wire whose full graph is shown in d).

Figure 2 employs the three representations in Figs. 1(b)–1(d) to describe the cluster states of interest in this work. The figure caption provides additional information about the different representations and their uses. In each type of cluster state, adjacent macronode wires are coupled together by dashed black beam splitters, enabling two-mode gates.

With this perspective, we introduce a unifying description that treats the core of each cluster state as a four-mode beam-splitter network  $\hat{B}_{\text{network}}$ , with each one described in its own subsection below.

### B. The teleportation gadget

As the modes in the cluster states are measured via homodyne detection, states are teleported down the macronode wires that comprise them and across couplings. Each of the cluster-state constructions we consider has at its core a



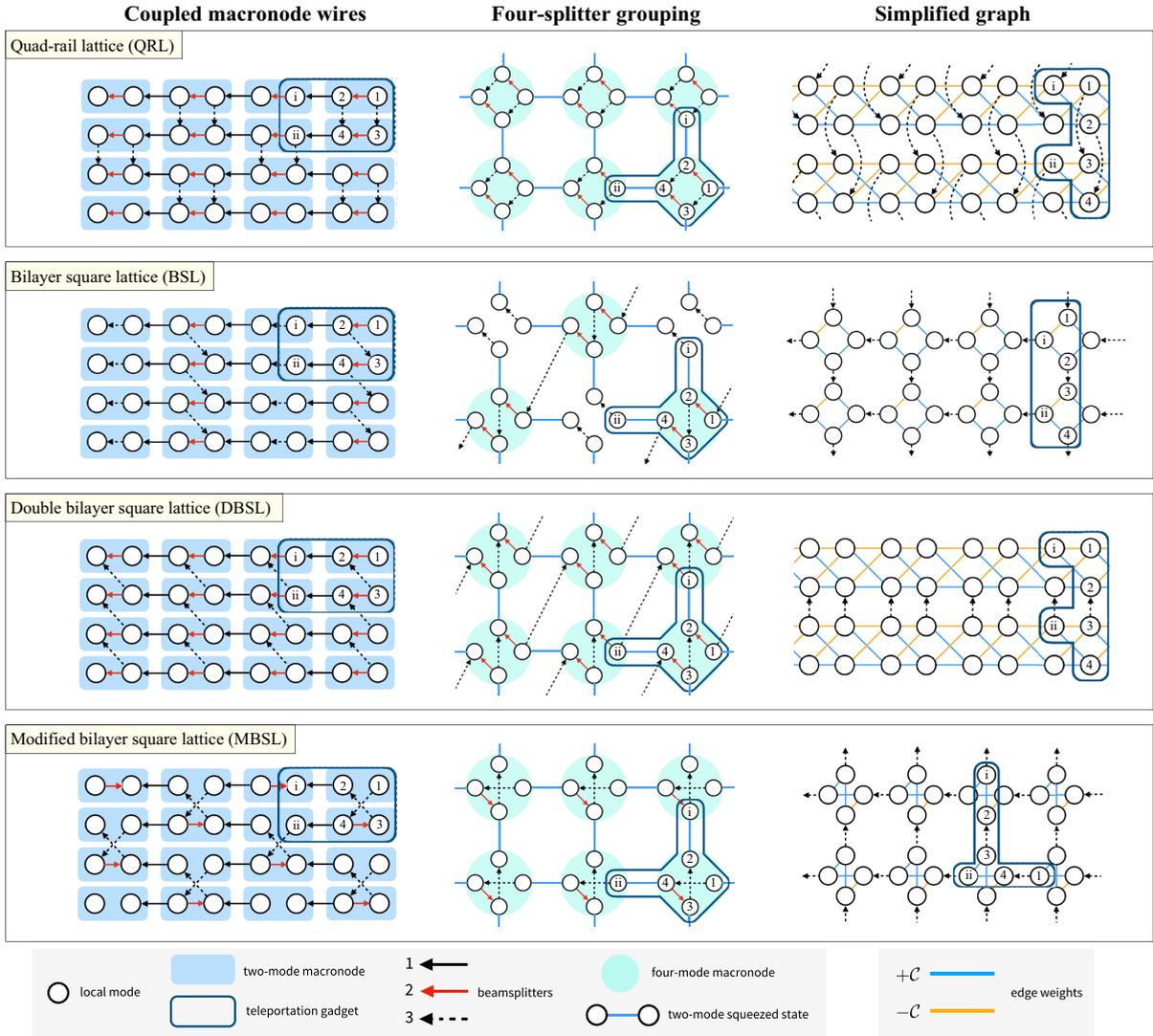


FIG. 2. Representations of the CV cluster states of interest in this work. Column 1 represents each cluster state as coupled macronode wires—a representation introduced in Ref. [32] for the QRL. Dashed black beam splitters provide the interwire coupling. The box highlights a six-mode teleportation gadget that provides the computing power for each cluster state. The QRL, BSL, and DBSL all comprise macronode wires that are coupled together in different ways. The macronode wires in the MBSL, however, are periodically missing beam splitters (missing horizontal red arrows compared to the others). Column 2 highlights the two-mode squeezed states that are generated initially (two nodes connected by a blue line) and how these states are coupled together at regular intervals by a four-splitter that is either complete (QRL) or incomplete (others); see Sec. VB. This representation shows groupings of four modes (different to the grouping in the first column), where each mode is half of an entangled pair (a TMSS). The QRL representation here differs slightly from that presented in Ref. [31]. We do not address this specifically here because the four-splitter structure (the internal structure of the macronodes) is the same in that work as it is here, so the argument we make here about the gate noise holds regardless. Column 3 shows a representation that employs the simplified graphical calculus [4] along with additional beam splitters (see main text). Similar representations for some of these states can be found in other works [4,7,31,33]. This column highlights the different types of construction: For the QRL and DBSL, macronode wires are constructed first and then connected to one another, and for the BSL and MBSL, nonlocal modes are coupled in groups of four, and then these groupings are connected. (Comparison with the other columns shows that the BSL can also be represented as coupled macronode wires; we choose here to follow the description in Ref [7].) Some works show the full graph-state structure [7,31,33] by implementing the graph rules of the final beam splitter [4,37], but we omit this description as it is graphically unwieldy and does not aid our attempts to compare them.

Appendix A,

where the position- and momentum-shear gates are

$$\hat{V}(\theta, \theta') = \hat{R}\left(\theta - \frac{\pi}{2}\right) \hat{P}[2 \cot(\theta - \theta')] \hat{R}\left(\theta - \frac{\pi}{2}\right) \quad (26)$$

$$= \hat{R}(\theta - \pi) \hat{P}_p[2 \cot(\theta - \theta')] \hat{R}(\theta), \quad (27)$$

$$\hat{P}(\sigma) := e^{i\frac{\sigma}{2}\hat{q}^2} \quad \text{and} \quad \hat{P}_p(\sigma) := e^{i\frac{\sigma}{2}\hat{p}^2}. \quad (28)$$

### D. Bilayer square lattice (BSL)

The bilayer square lattice (BSL) cluster state, introduced in Ref. [33] and studied elsewhere [7,31], is depicted in Fig. 2(b). The BSL begins with a collection of TMSSs. Groups of two modes, one from each TMSS, are chained together with beam splitters (red). These chains are periodically connected using a final set of beam splitters (dashed black). In some works [7,31,33], this procedure is represented in terms of the four-mode groupings created by the first two sets of beam splitters (black and red) [4],

$$(29)$$

The final set of beam splitters (dashed black) stitches together each group of four modes,

$$(30)$$

where we have included blue shading to identify the connection to the macronode wire, Fig. 1. With the macronode wire stretched out straight, we can identify the beam splitters that couple two macronode wires together; see column 1 of Fig. 2. In circuit form they are

$$\hat{B}_{\text{BSL}} = (31)$$

corresponding to the orthogonal matrix

$$\mathbf{R}_{\text{BSL}} = \mathbf{R}_{23}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 0 & 0 & \sqrt{2} & \sqrt{2} \end{bmatrix}. \quad (32)$$

Note that these are *not* the same four modes as in Eqs. (29) and (30), which is evident from the beam-splitter styles. Rather, this four-mode grouping is best seen in Fig. 2(b).

#### 1. Completing the BSL and connecting to the QRL

The BSL's beam-splitter network, Eq. (31), is *incomplete* because it contains three beam splitters, not four, as described in Sec. II C. The incompleteness is reflected in the fact that the BSL's matrix, Eq. (32), contains elements other than  $\pm\frac{1}{2}$ , indicating that it does not create equal superpositions of position and momentum operators across all modes.

The BSL can be completed by inserting a beam splitter  $\hat{B}_{14}$  between modes 1 and 4 to create new CV cluster state that we call the *completed* BSL (cBSL)<sup>3</sup> whose beam-splitter

<sup>3</sup>The choice  $\hat{B}_{41}$  also works. That choice would give a different orthogonal matrix, connection to the QRL, and two-mode gate.

network,  $\hat{B}_{\text{cBSL}} := \hat{B}_{14}\hat{B}_{\text{BSL}}$ , is given in circuit form by

$$\hat{B}_{\text{cBSL}} = (33)$$

with associated orthogonal matrix,

$$\mathbf{R}_{\text{cBSL}} = \mathbf{R}_{14}\mathbf{R}_{23}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}. \quad (34)$$

The cBSL's beam-splitter network can be written in terms of the QRL's under a single SWAP on the left and a parity operator, Eq. (10), on the right,

$$\hat{B}_{\text{cBSL}} = \text{SWAP}_{34}\hat{B}_{\text{QRL}}\hat{F}_4^2. \quad (35)$$

which has a circuit

$$\hat{B}_{\text{cBSL}} = (36)$$

This can be verified using the circuit identity in Eq. (F4). Now consider the cBSL beam-splitter network in the two-mode teleportation gadget, Circuit (20). Replacing  $\hat{B}_{\text{network}}$  with Eq. (35), pushing the SWAP into the measurements, and bouncing [34] the  $\hat{F}^2$  to the second output mode, we get the two-mode gate, Eq. (21), for the cBSL,

$$\hat{V}_{\text{cBSL}}^{(2)}(\boldsymbol{\theta}) = \hat{F}_2^2\hat{B}_{21}[\hat{V}_1(\theta_1, \theta_2) \otimes \hat{V}_2(\theta_4, \theta_3)]\hat{B}_{12}, \quad (37)$$

where the vector  $\boldsymbol{\theta}$  specifies the measurement angles in the initial gadget, Circuit (20). In circuit form, the two-mode gate is

$$(38)$$

Up to the parity operator, this gate is nearly the same as that for the QRL, Circuit (25), except that the measurement angles  $\theta_3$  and  $\theta_4$  have swapped positions. The equivalence has revealed that the cBSL can implement any teleported gate that the QRL can by properly choosing the measurement angles.

#### 2. Virtually completing the BSL

In the teleportation gadget, Eq. (20), a beam splitter network is followed by homodyne detection. This gives us the freedom to insert beam splitters into the measurement side of the circuit virtually by restricting measurement angles; see Sec. II C. Thus, we do not need to generate the cBSL physically—instead, we can complete the BSL virtually, allowing it to behave as if it were the cBSL. To do so, restrict

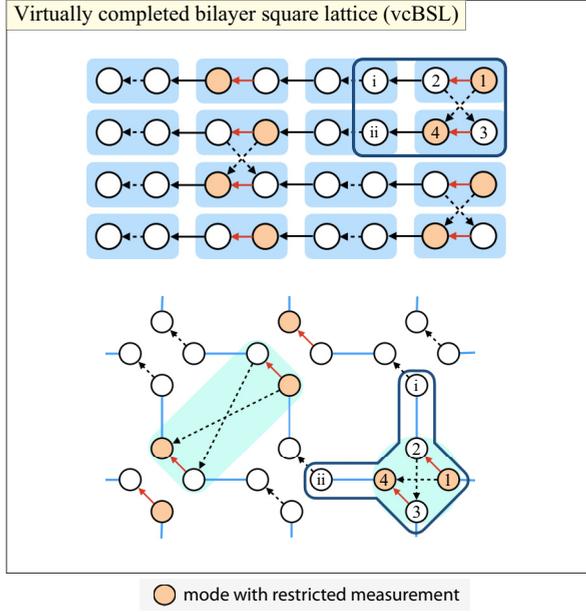


FIG. 3. The BSL with chosen restricted measurement angles, allowing it to behave as the cBSL. Local modes in orange are measured in the same basis (same measurement angle), allowing insertion of virtual beam splitters or virtual deletion of existing ones (see Sec. II C). The virtual beam splitters are those between orange modes. Compare with the unrestricted BSL in Fig. 2. The choice here provides isolated two-mode teleportation gadgets in a sparse brickwork structure.

the measurement bases on modes 1 and 4 to be the same (indicated by the orange-shaded modes in Fig. 3) to insert a virtual beam splitter between those modes,

$${}_{p_{\theta,1}}\langle m_1 | \otimes {}_{p_{\theta,4}}\langle m_4 | \hat{B}_{\text{BSL}} = \left\langle \frac{m_1 - m_4}{\sqrt{2}} \right\rangle_{p_{\theta,1}} \otimes \left\langle \frac{m_1 + m_4}{\sqrt{2}} \right\rangle_{p_{\theta,4}} \hat{B}_{\text{cBSL}}. \quad (39)$$

In the teleportation gadget, this gives the circuit for the *virtually completed BSL* (vcBSL),

$$\begin{array}{c} {}_{p_{\theta_1}}\langle \frac{m_1 - m_4}{\sqrt{2}} | \\ {}_{p_{\theta_2}}\langle m_2 | \\ {}_{p_{\theta_3}}\langle m_3 | \\ {}_{p_{\theta_4}=\theta_1}\langle \frac{m_1 + m_4}{\sqrt{2}} | \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}, \quad (40)$$

with the virtual beam splitter indicated as the dashed arrow.

As long as this measurement angle restriction is respected, the vcBSL can be used in the teleportation gadget as if it were the cBSL. (Feed-forward operations should also respect the linear transformations on the outcomes.) Thus, the vcBSL has the same two-mode gate as the cBSL under that restriction:

$$\hat{V}_{\text{vcBSL}}^{(2)}(\theta) = \hat{V}_{\text{cBSL}}^{(2)}(\theta)|_{\theta_1=\theta_4} \quad (41)$$

given in circuit form as

$$\begin{array}{c} \text{---} \\ \uparrow \\ \text{---} \end{array} \begin{array}{c} V(\theta_1, \theta_2) \\ \text{---} \\ \downarrow \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \\ \\ \\ \end{array} \quad (42)$$

Although any teleported gate that the QRL can implement is also available to the cMBSL via the relation in Eq. (68), this is not the case for the vcBSL—the measurement restrictions required to insert the missing beam splitter constrain the available gates.

In the larger structure of the cluster state, freedom to restrict measurement angles in order to virtually add or remove beam splitters allows different connections between modes and between two-mode teleportation gadgets. In practice, restricted measurement angles can be made on the fly to tailor the connectivity to the problem at hand. Figure 3 illustrates a choice that produces a brickwork-like structure, similar to that of the QRL, although here the brickwork structure is more sparse, and this choice allows for two-mode gates followed by single-mode gates.

### E. Double bilayer square lattice (DBSL)

The DBSL, introduced in Ref. [14], is a modification of the BSL with a construction that does not begin with the unit described in Eq. (29). Rather, constructing the DBSL begins by producing a macronode wire, Eq. (1), which is then wound up into a cylinder by introducing nonlocal beam-splitter connections [31]. Position measurements can then be used to cut out 2D cluster-state sheets with the structure of the DBSL. (Further position measurements can reduce this DBSL to a BSL [14]).

Larsen *et al.* [14] experimentally generated a DBSL cluster state and in a follow-up work [17] used the state to implement a variety of measurement-based Gaussian operations on one and two modes. To do so, they used an alternate perspective on the DBSL cluster state that identifies macronode wires running perpendicular to those in column 3 of Fig. 2(c), again using position measurements to pare the DBSL down to a simpler cluster state. That subfigure depicts a portion of one of these 2D sheets, revealing that the DBSL can be locally viewed as separate macronode wires connected periodically by beam splitters.

Column 1 of Fig. 2(c) depicts four of these coupled macronode wires stretched out into the rows of a two-mode cluster state. We will use this representation to propose a method of using the DBSL that doesn't require any such deletions.

The box indicates the teleportation gadget containing the four-mode beam-splitter network  $\hat{B}_{\text{DBSL}}$ ,

$$\hat{B}_{\text{DBSL}} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \downarrow \\ \uparrow \\ \downarrow \\ \downarrow \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}, \quad (43)$$

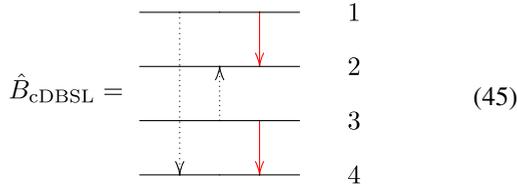
which has unitary matrix

$$\mathbf{R}_{\text{DBSL}} = \mathbf{R}_{32}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & -1 \\ 0 & 0 & \sqrt{2} & \sqrt{2} \end{bmatrix}. \quad (44)$$

While the beam-splitter networks for the BSL and DBSL differ only in the direction of a single beam splitter [compare Eqs. (31) to (43)], the cluster states at large have a more significant difference—the DBSL has double the connectivity between macronode wires; compare the DBSL and BSL in column 1 of Fig. 2. More connectivity provides more access to two-mode gates, as they can only be done between connected wires.

### 1. Completing the DBSL and connecting to the QRL

The DBSL's beam-splitter network, Eq. (43), is incomplete in the same way the BSL's is. In the same way, inserting a beam splitter between modes 1 and 4 gives the *completed DBSL* (cDBSL) with beam-splitter network  $\hat{B}_{\text{cDBSL}} := \hat{B}_{14}\hat{B}_{\text{DBSL}}$ . In circuit form,



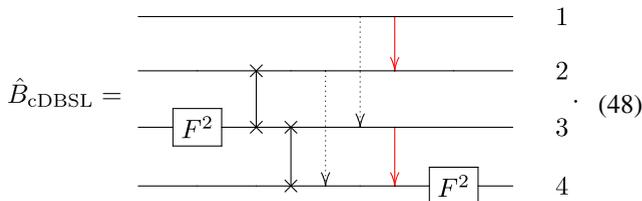
and with associated unitary matrix

$$\mathbf{R}_{\text{cDBSL}} = \mathbf{R}_{14}\mathbf{R}_{32}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}. \quad (46)$$

The completed DBSL (cDBSL) beam-splitter network can be written in terms of the QRL under a parity operation and two SWAPs on the left and a parity operation on the right:

$$\hat{B}_{\text{cDBSL}} = \hat{F}_3^2 \text{SWAP}_{23} \text{SWAP}_{34} \hat{B}_{\text{QRL}} \hat{F}_4^2, \quad (47)$$

which has a circuit

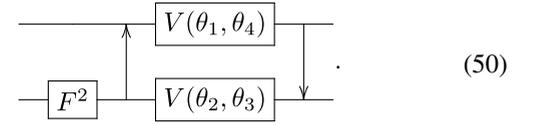


Using the cBSL beam-splitter network in the two-mode teleportation gadget, Circuit (20), gives a two-mode gate

$$\hat{V}_{\text{cDBSL}}^{(2)}(\boldsymbol{\theta}) = \hat{F}_2^2 \hat{B}_{21} [\hat{V}_1(\theta_1, \theta_4) \otimes \hat{V}_2(\theta_2, \theta_3)] \hat{B}_{12}, \quad (49)$$

where the vector  $\boldsymbol{\theta}$  specifies the measurement angles in the initial gadget, Circuit (20). In circuit form, the two-mode

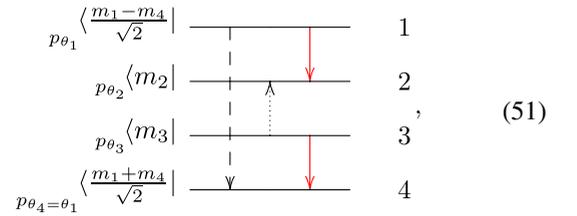
gate is



Up to the parity operator, this gate is nearly the same as that for the QRL, Circuit (25), except that some measurement angles have swapped positions.

### 2. Virtually completing the DBSL

Now consider the DBSL in the two-mode teleportation gadget, Circuit (20). We complete the DBSL's beam-splitter network in the same way as we did for the BSL's, Eq. (39): restricting the measurement bases on modes 1 and 4 and inserting a beam splitter between these modes. Doing so gives the circuit for the *virtually completed DBSL* (vcDBSL),

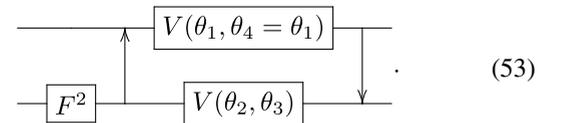


with the virtual beam splitter indicated as the dashed arrow.

The vcDBSL functions like the cDBSL in the teleportation gadget, with the same two-mode gate under the measurement restrictions above,

$$\hat{V}_{\text{vcDBSL}}^{(2)}(\boldsymbol{\theta}) = \hat{V}_{\text{cDBSL}}^{(2)}(\boldsymbol{\theta})|_{\theta_1 = \theta_4} \quad (52)$$

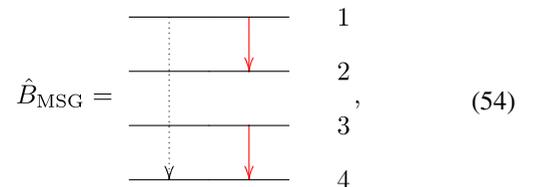
represented in circuit form as



In the larger structure of the cluster state, Fig. 4 illustrates a choice of restricted measurement angles that gives a highly connected brickwork structure just like that of the QRL (and more densely connected than the BSL; Fig. 3).

### F. 3D DBSL Mikkell-splitter gadget (MSG)

After their work with the DBSL, Larsen *et al.* [28] extended the DBSL into a 3D cluster state that utilizes variable beam splitters to choose the location of their two mode gate. In doing so, they identified a teleportation gadget of the same form as Circuit (20), which we dub the *Mikkell-splitter gadget* (MSG) after the given name of the first author. The MSG's beam-splitter network  $\hat{B}_{\text{MSG}}$  is



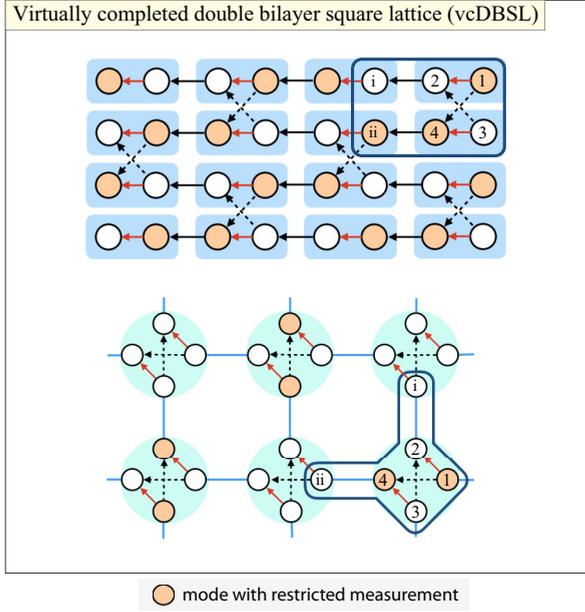


FIG. 4. The DBSL with restricted measurement angles, allowing it to function as the cDBSL. Local modes in orange are measured in the same basis to virtually insert or delete beam splitters; see Sec. II C. Compare with the unrestricted BSL in Fig. 2. The virtual beam splitters are those between orange modes. The choice here provides isolated teleportation gadgets in a dense brickwork structure like that of the QRL.

with the unitary matrix

$$\mathbf{R}_{\text{MSG}} = \mathbf{R}_{14}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \\ 1 & -1 & 1 & 1 \end{bmatrix}. \quad (55)$$

### 1. Completing the MSG and connecting to the QRL

The MSG's beam-splitter network can be completed in a similar way to that of the BSL and the DBSL, with the virtual beam splitter instead being inserted between modes 2 and 3. Restricting these measurement bases leads to the same completed beam-splitter network and associated orthogonal matrix as the BSL, i.e.,  $\hat{\mathbf{B}}_{\text{cMSG}} := \hat{\mathbf{B}}_{23}\hat{\mathbf{B}}_{\text{MSG}} = \hat{\mathbf{B}}_{\text{cBSL}}$ ,

$$\hat{\mathbf{B}}_{\text{cMSG}} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array}, \quad (56)$$

with associated orthogonal matrix,

$$\mathbf{R}_{\text{cMSG}} = \mathbf{R}_{23}\mathbf{R}_{14}\mathbf{R}_{34}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}. \quad (57)$$

Consequently, the connection to the QRL is also identical, Eq. (35), and the two-mode gate in the teleportation gadget

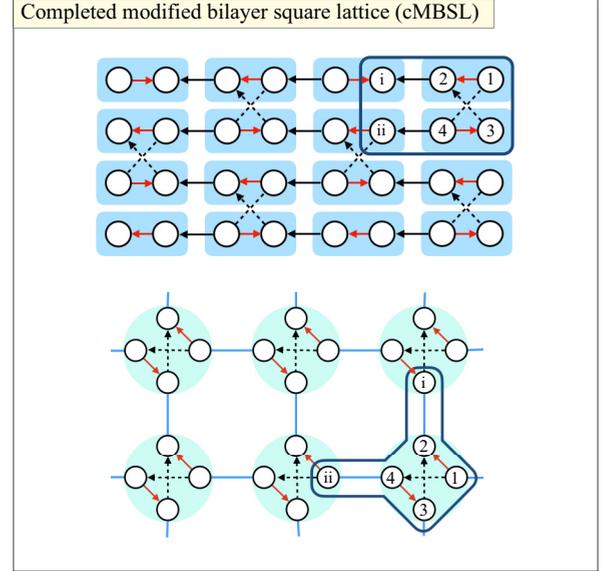


FIG. 5. Adding an additional beam splitter to the MBSL produces the completed MBSL (cMBSL). The MBSL cannot be completed virtually by restricting measurement angles. The teleportation gadgets form a dense brickwork structure like that of the QRL.

is the same as the BSL, (37),

$$\hat{\mathcal{V}}_{\text{cMSG}}^{(2)}(\boldsymbol{\theta}) = \hat{\mathcal{V}}_{\text{cBSL}}^{(2)}(\boldsymbol{\theta}), \quad (58)$$

with circuit form given in Circuit (38).

The 3D cluster state of Ref. [28] reduces down to a form of the DBSL when the second variable beam splitter therein is removed. This connection is clear from their identical completed four-splitter. However, this connection also identifies the useful fact that these four-splitters extend beyond 2D lattices and can be used to define two-mode gates wherever four modes meet together at a lattice site—regardless of dimension of the greater lattice. This will become more important as we generalize these four-splitters in Sec. V.

### 2. Virtually completing the MSG

Now consider the MSG in the two-mode teleportation gadget, Circuit (20). We complete the MSG's beam-splitter network by restricting the measurement bases on modes 2 and 3 and to insert a virtual beam splitter between these modes. Doing so gives the circuit for the *virtually completed MSG* (vcMSG),

$$\begin{array}{c} p_{\theta_1} \langle m_1 | \\ p_{\theta_2} \langle \frac{m_2 - m_3}{\sqrt{2}} | \\ p_{\theta_3 = \theta_2} \langle \frac{m_2 + m_3}{\sqrt{2}} | \\ p_{\theta_4} \langle m_4 | \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array}, \quad (59)$$

with the virtual beam splitter indicated as the dashed arrow.

In the teleportation gadget, the vcMSG functions like the cMSG with the same two-mode gate under the measurement-angle restrictions,

$$\hat{\mathcal{V}}_{\text{vcMSG}}^{(2)}(\boldsymbol{\theta}) = \hat{\mathcal{V}}_{\text{cMSG}}^{(2)}|_{\theta_2 = \theta_3} \quad (60)$$

in circuit form

$$(61)$$

### G. Modified bilayer square lattice (MBSL)

Asavanant *et al.* [13] introduced and experimentally created the modified bilayer-square-lattice (MBSL) cluster state. It is constructed by first coupling TMSSs into basic four-mode groupings

$$(62)$$

and then stitching these groupings together using additional beam splitters; this is shown in column 3 of Fig. 2. When viewed from the perspective of coupled macronode wires, the MBSL has a different structure than the others considered in this work. That is because, in addition to the beam-splitter network being incomplete—a trait shared by the other cluster states—the macronode wires in the MBSL themselves are also incomplete, which can be seen in column 1 of Fig. 2 as missing horizontal red beam splitters.

The beam-splitter network for the MBSL is

$$\hat{B}_{\text{MBSL}} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}, \quad (63)$$

with the corresponding orthogonal matrix

$$\mathbf{R}_{\text{MBSL}} = \frac{1}{2} \begin{bmatrix} \sqrt{2} & 0 & 1 & -1 \\ 0 & \sqrt{2} & 1 & 1 \\ 0 & -\sqrt{2} & 1 & 1 \\ \sqrt{2} & 0 & -1 & 1 \end{bmatrix}. \quad (64)$$

The larger MBSL cluster-state construction has a brickwork-like structure, but as mentioned, it is missing critical beam splitters that complete the macronode wires themselves.

#### Completing the MBSL and connecting to the QRL

Physically inserting the fourth beam splitter gives a new CV cluster state (depicted in Fig. 5)—the completed MBSL (cMBSL). The cMBSL's beam-splitter network is  $\hat{B}_{\text{cMBSL}} := \hat{B}_{\text{MBSL}}\hat{B}_{12}$ , given in circuit form as

$$\hat{B}_{\text{cMBSL}} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \quad (65)$$

with associated unitary matrix

$$\mathbf{R}_{\text{cMBSL}} = \mathbf{R}_{14}\mathbf{R}_{32}\mathbf{R}_{43}\mathbf{R}_{12} = \frac{1}{2} \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \quad (66)$$

The connection to the QRL is given by

$$\hat{B}_{\text{cMBSL}} = \hat{F}_3^2 \text{SWAP}_{14} \text{SWAP}_{23} \text{SWAP}_{34} \hat{B}_{\text{QRL}} \quad (67)$$

with the circuit

$$\hat{B}_{\text{cMBSL}} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \quad (68)$$

Since the fourth beam splitter was physically inserted into the circuit, the completed MBSL has no measurement restrictions in its teleported two-mode gate:

$$\hat{V}_{\text{cMBSL}}^{(2)}(\theta) = \hat{F}_2^2 \hat{B}_{21} [\hat{V}_1(\theta_4, \theta_3) \otimes \hat{V}_2(\theta_1, \theta_2)] \hat{B}_{12} \quad (69)$$

and can perform any gate that the QRL can via Eq. (68). The larger cluster state structure of the cMBSL is the same as that for the cDBSL shown in Fig. 4. This is the same brickwork structure as the QRL, albeit with a different beam-splitter network. Thus, the cMBSL is as versatile as the QRL is for cluster-state computing as the QRL is—it can implement the same gates and can do it as often as the QRL can.

Unlike the other CV cluster states above, the MBSL cannot be completed by restricting measurement bases, because its missing fourth beam splitter is not on the measurement side of the circuit. Rather, it is on the state side of the circuit, meaning that a beam splitter must be physically included in the circuit; see Sec. VB for more details.

## IV. CLUSTER-STATE COMPUTING WITH THE GKP CODE

Recent work [32] showed that the QRL is compatible with the Gottesman-Kitaev-Preskill (GKP) encoding of a qubit into a mode [21]. This compatibility rests on two key features [32]. First, the ability to perform GKP error correction can be built into the QRL cluster state itself by replacing the usual TMSSs with GKP Bell pairs made from GKP quanaught states [see Eq. (70) below] and beam splitters. As the modes of the cluster state are measured, the potentially noisy input states to each teleportation gadget are automatically projected into the GKP code space, which is followed by an outcome-dependent correction. Second, choosing appropriate measurement angles in the QRL teleportation gadget gives a complete generating set of GKP Clifford gates. These gates are executed in a single teleportation step and require only four modes in the beam-splitter network.

Using GKP Bell pairs in the cluster state allows GKP error correction to be performed at the same time as each Clifford gate. Walshe *et al.* [32] calculated the logical gate noise from simultaneous implementation of a GKP Clifford

and GKP error correction and found that, given GKP qunaught states of high enough quality, fault tolerance is achievable by concatenation with a larger qubit code.

The above features are not exclusive to the QRL. Using the equivalence of the other cluster states to the QRL, established in Sec. III, we show here that (a) each cluster state can implement a full generating set of GKP Cliffords while at the same time implementing GKP error correction, and (b) the associated GKP Clifford-gate noise is identical to that of the QRL, thus making any of the cluster states suitable for fault-tolerant quantum computation.

### A. Constructing cluster states with GKP Bell pairs

GKP Bell pairs are created by mixing two GKP qunaught states on a beam splitter [34]. We consider (unnormalized) energy-constrained qunaught states, also called *noisy qunaught states*, which are defined as

$$|\bar{\mathcal{D}}\rangle = \hat{N}(\beta) |\mathcal{D}\rangle = \hat{N}(\beta) \int ds \text{III}_{\sqrt{2\pi}}(s) |s\rangle_q, \quad (70)$$

with energy damping operator  $\hat{N}(\beta) := e^{-\beta \hat{a}^\dagger \hat{a}}$ , and  $\text{III}_T(x)$  is a Dirac comb with period  $T$ . The position (and momentum) noisy qunaught wave function is a comb of Gaussians with broad Gaussian envelope that damps peaks far from the origin. Two noisy GKP qunaughts on a beam splitter produce a noisy GKP Bell pair, which itself is equivalent to a noisy GKP projector acting on one half of an EPR pair [34],

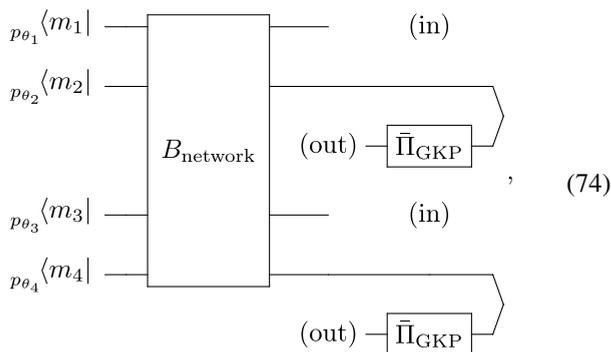
$$\begin{aligned} \hat{B}_{12} |\bar{\mathcal{D}}\rangle \otimes |\bar{\mathcal{D}}\rangle &= \hat{N}_1(\beta) \otimes \hat{N}_2(\beta) |\Phi_{\text{GKP}}\rangle \\ &= \hat{N}_2(\beta) \hat{\Pi}_{\text{GKP},2} \hat{N}_1(\beta) |\text{EPR}\rangle, \end{aligned} \quad (71)$$

where  $|\Phi\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$  is a two-qubit Bell state,  $|\text{EPR}\rangle := \int ds |s\rangle_q \otimes |s\rangle_q$  is a two-mode EPR state, and the noisy GKP projector is

$$\hat{\Pi}_{\text{GKP}} := \hat{N}(\beta) \hat{\Pi}_{\text{GKP}} \hat{N}(\beta), \quad (73)$$

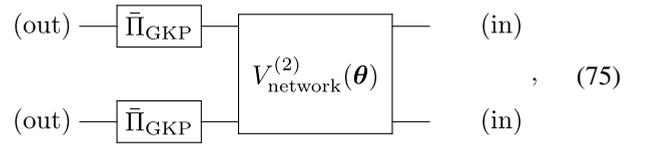
with  $\hat{\Pi}_{\text{GKP}}$  being the ideal GKP projector.

Replacing the ancilla states  $|\psi\rangle$  and  $|\phi\rangle$  in Circuit (20) with noisy qunaught states  $|\bar{\mathcal{D}}\rangle$  teleports the two inputs through two GKP Bell pairs during the implementation of Gaussian gate  $\hat{V}_{\text{network}}^{(2)}(\theta)$ . Using the identity in Eq. (72), the circuit becomes



with the noisy GKP projectors appearing only on the output modes. Applying the circuit transformation rules presented in

Ref. [34] gives the simplified circuit



showing that the implemented gate,  $\hat{V}_{\text{network}}^{(2)}(\theta)$ , depends only on the details of the beam-splitter network and measurement angles there; the noisy GKP projectors act afterwards. Thus, using GKP Bell pairs in the construction of any of the cluster states enables teleportation-based GKP error correction. An alternative method is to insert GKP Bell pairs into a CVCS after the fact; a method to do so is given in Appendix E.

### B. Implementing GKP Clifford gates

For the square-lattice GKP code, a set of CV unitaries that enables a complete generating set GKP Clifford gates is [32]

$$\underbrace{\{\hat{f}, \hat{F}, \hat{P}(\pm 1), \hat{C}_Z(\pm 1)\}}_{\text{CV unitaries}} \mapsto \underbrace{\{\hat{I}, \hat{H}, \hat{P}, \hat{C}_Z\}}_{\text{GKP Cliffords}}. \quad (76)$$

where  $\hat{F} = \hat{R}(\frac{\pi}{2})$ ,  $\hat{P}(\sigma)$  is given in Eq. (28), and  $\hat{C}_Z(g) := e^{ig\hat{q}\otimes\hat{q}}$ . The single-qubit GKP Cliffords are implemented in a single teleportation step of a macronode wire [32,34] and are thus accessible to any of the cluster states above since we have expressed them as coupled macronode wires. Whether a two-qubit gate required to complete the set of GKP Clifford gates is accessible depends on the beam-splitter network in the teleportation gadget. Walshe *et al.* [32] found that, in a single teleportation step, the QRL can implement  $\hat{C}_Z(1)$ , the SWAP gate, and a suite of simultaneous single-mode gates. The measurement angles that do so are given in the first column of Table I.

The completed cluster states—cBSL, cDBSL, cMSG, and cMBSL—each have access to all of the QRL's gates using the correspondences in the previous section. But what about the virtually completed versions, which are completed by restricting measurement angles? In Table I, we leverage the connections between the cluster states above and the QRL to find the measurement angles that implement the same gates on the virtually completed cluster states, up to parity operations (which do not affect GKP codes).<sup>4</sup> We find that the vcBSL, vcDBSL, and vcMSG can natively implement almost all of the Clifford gates that the QRL can—the SWAP is the only one for which we could not find a set of angles. In the last column, we give the measurement angles from Ref. [28] that generate  $\hat{C}_Z(1)$  (up to local Fourier transforms) for the MSG. This gate was found natively for that cluster state [28] and does not result from the QRL connection in this work—highlighting the fact that this list is not exhaustive. Additional measurement angles implementing the same logical-GKP gates may exist for any of the cluster states, including the noncompleted versions.

<sup>4</sup>Single-mode GKP codes on any lattice centered at the origin are even parity [38].

TABLE I. Dictionary of two-mode GKP Clifford gates. The mapping takes specific sets of measurement angles  $\{\theta_1^Q, \theta_2^Q, \theta_3^Q, \theta_4^Q\}$  that implement gates for the quad-rail lattice (QRL) given in Ref. [32] and shows the reordering of and restrictions on those angles that perform the same gates on the bilayer square lattice (BSL), the double bilayer square lattice (DBSL), and the Mikkel-splitter gadget (MSG) through their virtually completed counterparts vcBSL, vcDBSL, and vcMSG. The constant angle is defined  $\chi = \arctan 2$  [32]. Each mapping indicates a measurement-angle restriction required to complete the beam-splitter network; see Sec. III.

Architecture mapping	QRL $\{\theta_1^Q, \theta_2^Q, \theta_3^Q, \theta_4^Q\}$	vcBSL <sup>a</sup> $\{\theta_1^Q, \theta_2^Q, \theta_4^Q, \theta_3^Q\} _{\theta_1^Q=\theta_3^Q}$	vcDBSL <sup>a</sup> $\{\theta_1^Q, \theta_4^Q, \theta_2^Q, \theta_3^Q\} _{\theta_1^Q=\theta_3^Q}$	vcMSG <sup>a</sup> $\{\theta_1^Q, \theta_2^Q, \theta_4^Q, \theta_3^Q\} _{\theta_2^Q=\theta_4^Q}$
$\hat{C}_Z(1)$	$\{\frac{\pi}{2}, \frac{\pi}{2} \pm \chi, \frac{\pi}{2}, \frac{\pi}{2} \mp \chi\}$	$\{\frac{\pi}{2}, \frac{\pi}{2} \pm \chi, \frac{\pi}{2} \mp \chi, \frac{\pi}{2}\}$	$\{\frac{\pi}{2}, \frac{\pi}{2} \mp \chi, \frac{\pi}{2} \pm \chi, \frac{\pi}{2}\}$	$\{-\chi, 0, 0, \chi\}$ <sup>b</sup>
SWAP	$\{0, \frac{\pi}{2}, \frac{\pi}{2}, 0\}$	— <sup>c</sup>	— <sup>c</sup>	— <sup>c</sup>
$\hat{I} \otimes \hat{I}$	$\{\frac{\pi}{2}, 0, \frac{\pi}{2}, 0\}$	$\{\frac{\pi}{2}, 0, 0, \frac{\pi}{2}\}$	$\{\frac{\pi}{2}, 0, 0, \frac{\pi}{2}\}$	$\{\frac{\pi}{2}, 0, 0, \frac{\pi}{2}\}$
$\hat{F} \otimes \hat{F}$	$\{\frac{3\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{4}\}$	$\{\frac{3\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}\}$	$\{\frac{3\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}\}$	$\{\frac{3\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}\}$
$\hat{P}(1) \otimes \hat{P}(1)$	$\{\frac{\pi}{2}, \frac{\pi}{2} \mp \chi, \frac{\pi}{2}, \frac{\pi}{2} \mp \chi\}$	$\{\frac{\pi}{2}, \frac{\pi}{2} \mp \chi, \frac{\pi}{2} \mp \chi, \frac{\pi}{2}\}$	$\{\frac{\pi}{2}, \frac{\pi}{2} \mp \chi, \frac{\pi}{2} \mp \chi, \frac{\pi}{2}\}$	$\{\frac{\pi}{2}, \frac{\pi}{2} \mp \chi, \frac{\pi}{2} \mp \chi, \frac{\pi}{2}\}$

<sup>a</sup>Each gate is accompanied by a parity operator  $\hat{F}^2$  on the second mode. This has no effect of the noise properties of the gate operation.

<sup>b</sup>These measurement angles, given in Ref. [28], implement  $[\hat{F}^\dagger \otimes \hat{F}]\hat{C}_Z(1)$  and are not mapped from the QRL. Incidentally, our results show that this gate is available to the QRL by the inverse mapping.

<sup>c</sup>We found no mappings between SWAP gates for the QRL (isolated or combined with single-mode gates) compatible with the angle restrictions on the vcBSL, vcDBSL, and vcMSG.

### Logical gate noise

As a cluster state is measured, unavoidable noise accumulates due to the fact that the cluster states are built from finitely squeezed resources—TMSSs or GKP Bell pairs. The CV noise translates to logical qubit errors when the cluster state is combined with encoded GKP qubits. For high-quality GKP input states, Walshe *et al.* [32] calculated the logical noise accompanying each GKP Clifford gate in Eq. (76) and found that a 1% gate error rate is achievable with GKP resources and a CV cluster state both having squeezing of at least 11.9 dB. This minimum value is set by the two-mode  $\hat{C}_Z(1)$  gate, which is the poorest performing gate in the set.

The origin of this noise lies in the fact that all states used in the protocol have finite squeezing; none is ideal. This is quantified for the GKP Bell pairs using the damping operator  $\hat{N}(\beta)$ , Eq. (70). Inspecting Circuit (75) reveals that the damping operators are entirely contained in the noisy GKP projectors, Eq. (73), and are independent of the details of the beam-splitter network and measurements that enact the gate  $\hat{V}_{\text{network}}^{(2)}(\theta)$ . This means that if an identical  $\hat{V}_{\text{network}}^{(2)}(\theta)$  is found for two different cluster state constructions, the associated logical-gate noise will be the same.

All of the completed cluster states—the cBSL, cDBSL, cMSG, and cMBSL—can implement any gate that the QRL can, including the entangling GKP Clifford gate  $\hat{C}_Z(1)$  (up to single-mode Fourier transforms and parity operators). Table I shows that the vcBSL, vcDBSL, and vcMSG can also implement  $\hat{C}_Z(1)$  natively as well as any subset of the QRL's gates that respect the angle restrictions; SWAP is a notable exception. This means that the gate noise on  $\hat{C}_Z(1)$  is the same as the QRL's, and each cluster state inherits the logical-gate noise (as a function of squeezing) calculated in Ref. [32] for the gates native to the architecture—all gates for physically completed four-splitters and a subset of the gates for virtually completed ones.

Our conclusion appears to conflict with a prior study of gate noise for the BSL and DBSL [31], which concluded that higher-quality GKP resources were needed there than for

the QRL. In that study, deleting certain modes from the BSL and DBSL (using position measurements) was required before computation in order to isolate macronode wires. Each deleted mode introduces noise into the remaining ones, increasing the requirements on the GKP resources. In contrast, the equivalence of the BSL and DBSL with the QRL in the present work arises from utilizing each cluster state efficiently by selecting macronode wires—distinct from those in Ref. [17]—that require no deletions before use.

We note that although the GKP logical Clifford gate noise is identical across the different cluster states, they may have different uses due to inherently different connectivity. For example, the DBSL has a denser set of teleportation gadgets than the BSL, permitting two-mode gates more often; compare Figs. 3 and 4. The effects of these differences on explicit calculations of fault-tolerance thresholds are left to future work. Our purpose here is to show that many cluster states perform just as well as the QRL in terms of gate noise and thus inherit the favorable performance reported in Ref. [32].

### V. BALANCED FOUR-SPLITTERS

In the above section, we compared a small set of cluster state constructions that have been introduced in the literature. The QRL is unique among these constructions in that it stitches together one half of four TMSSs (or GKP Bell pairs [32]) using *four* beam splitters; all the other constructions use three. The key to connecting the cluster states above to the QRL is completing them by inserting a fourth beam splitter—either physically (in the case of the MBSL) or virtually using restricted measurements (all others). Here we show that these completed are members of a class of four-mode transformations generated by *balanced four-splitters*, a term coined in Ref. [33].<sup>5</sup> This extends the gate-noise results—and others

<sup>5</sup>In that reference, the term referred specifically to  $\hat{B}_{\text{QRL}}$ , Eq. (6). Here we expand this term to include a larger class of unitary gates, of which  $\hat{B}_{\text{QRL}}$  is one example.

that may follow from this work—to apply not only to the small set of cluster states discussed here but to every cluster state that might be conceived of that shares the four-splitter properties described herein. For generality, we extend this definition to  $n$  modes.

*Definition 1 (Balanced  $n$ -splitter).* Let  $\hat{U}$  be an  $n$ -mode linear-optical unitary operator with  $\hat{U} \rightarrow \mathbf{U} \in \mathbf{U}(n)$  [see Eq. (4) for notation].  $\hat{U}$  is called a *balanced  $n$ -splitter* if all elements of  $\mathbf{U}$  have the same magnitude. The matrix  $\mathbf{U}$  is called the  *$n$ -splitter matrix* corresponding to  $\hat{U}$ .

*Definition 2 (Real balanced  $n$ -splitter).* Let  $\hat{U}$  be a balanced  $n$ -splitter.  $\hat{U}$  is called a *real balanced  $n$ -splitter* if its  $n$ -splitter matrix  $\mathbf{U} \in \mathbf{O}(n)$  or, equivalently, if all elements of  $\mathbf{U}$  are real.

Four-splitters over  $n$  modes map every mode operator  $\hat{a}_j$  to an equal-magnitude (i.e., balanced) linear combination of all  $n$  mode operators, namely,  $\hat{U}^\dagger \hat{a}_j \hat{U} = \sum_{k=1}^n U_{jk} \hat{a}_k$  with  $|U_{jk}| = \frac{1}{\sqrt{n}}$ . Given a balanced  $n$ -splitter  $\hat{U} \rightarrow \mathbf{U}$ , then  $\mathbf{U}$  is  $n^{-1/2}$  times an order- $n$ , complex Hadamard matrix [39,40], by definition of the latter. When  $\hat{U}$  is a real balanced  $n$ -splitter, then the associated Hadamard matrix is real rather than complex.

Our work will largely focus on real balanced  $n$ -splitters. We need the following important definition:

*Definition 3 (Equivalence of real balanced  $n$ -splitters).* Two real balanced  $n$ -splitters  $\hat{U}_1 \rightarrow \mathbf{R}_1$  and  $\hat{U}_2 \rightarrow \mathbf{R}_2$  are called *equivalent* if  $\sqrt{n}\mathbf{R}_1$  and  $\sqrt{n}\mathbf{R}_2$  are equivalent as order- $n$  real Hadamard matrices [41], i.e., if one can be transformed into the other by a sequence of row permutations, column permutations, row negations, and column negations.

For real balanced  $n$ -splitters with  $n \in \{2, 4, 8, 12\}$ , all belong to a single equivalence class for the given  $n$  [41], but for larger  $n$ , multiple equivalence classes exist. We will focus on the cases of  $n = 2$  and  $n = 4$  below.

Immediately from the definition above, we see that the beam splitter  $\hat{B}_{jk} \rightarrow \mathbf{R}_{jk}$  defined in Eq. (3) is a real balanced two-splitter. But it is not the only one. For two modes, there are exactly eight possible real balanced two-splitters, corresponding to four cases of exactly one negative sign in  $\mathbf{R}_{jk}$  and four cases of exactly three negative signs in  $\mathbf{R}_{jk}$ . This is a property of the set of order-2 (real) Hadamard matrices [41].

One can straightforwardly verify that for this small set of Hadamard matrices, swapping the rows is equivalent to negating one column, and swapping the columns is equivalent to negating one row. Also, if negation of each row is allowed, then negation need only be allowed on one fixed column (say, the second column) to generate all possible real two-splitters from a given one. Thus, the full set of equivalence operations can be reduced to simply row negation and negation of the second column. This is an example of a smallest generating set, as well, by a simple counting argument: since there are  $8 = 2^3$  real balanced two-splitters, a minimum of three binary operations is necessary to generate all of them from a single one.

The set of real balanced four-splitters displays similar behavior. While the QRL four-splitter  $\hat{B}_{\text{QRL}}$  [Eq. (22)] defines one particular example of a real balanced four-splitter, many other examples exist. In fact, there are 768 real balanced four-splitters, one for each order-4 real Hadamard matrix [41]. We are interested in treating this set of matrices as an equivalence

class under a minimal set of matrix operations because each of these corresponds to a linear-optical unitary gate: permutations are SWAP gates, and negating a row or column is a  $\hat{F}^2$  gate before or after the four-splitter, respectively. We will use this below to demonstrate the functional equivalence of all real balanced four-splitters in the CV cluster-state setting.

Theorem 1, proven next, shows that the equivalence class of order-4 Hadamard matrices can be generated with a smaller set of matrix operations than is generally needed—row negations, row permutations, and a single column negation are sufficient. This is indeed the minimal set of operations.

*Theorem 1.* Let  $\hat{U}_1 \rightarrow \mathbf{R}_1$  and  $\hat{U}_2 \rightarrow \mathbf{R}_2$  be real balanced four-splitters. Then  $\mathbf{R}_1$  can be transformed into  $\mathbf{R}_2$  by some sequence of row permutations, row negations, and negation of any single column.

*Proof.* Since every  $\mathbf{R}_{1,2} = \frac{1}{2}\mathbf{H}_{1,2}$ , where  $\mathbf{H}_1$  and  $\mathbf{H}_2$  are order-4 real Hadamard matrices. Our strategy will be to show that the set of all order-4 real Hadamard matrices,  $H_4$ , is a single equivalence class according to the conditions of equivalence in the theorem. Since  $|H_4| = 768$  [41], we will use the available operations to construct 768 unique matrices, each of which is a member of  $H_4$ .

Let

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \in H_4. \quad (77)$$

Notice that the row parity (number of minus signs in a given row) of each row of  $\mathbf{H}_0$  is even. Orthogonality ensures that the  $4!$  possible row permutations and  $2^4$  possible row negations generate distinct matrices (since no row can equal any other row, nor can it equal any row's negation). This accounts for  $4! \times 2^4 = 384$  of the elements of  $H_4$  (including  $\mathbf{H}_0$ ). All of these have all-even row parity. To access the other half of  $H_4$ , we negate the last column of each of the matrices already generated. Each of these new matrices is distinct from any of the others already obtained because row permutations and row negations preserve row parity, but a single column negation flips the parity of all rows. Thus, we have produced another 384 members of  $H_4$  (with odd row parity) that are all distinct from each other and from the other half (with even row parity). This exhausts the 768 members of  $H_4$ . The above procedure can be repeated using any of the other columns in place of the last one, showing the choice of column to be arbitrary. The result equally applies after multiplying each of these matrices by  $\frac{1}{2}$ . This proves the theorem. ■

*Note:* This subsection has used the precise term *real balanced four-splitter* in order to distinguish it from (more general) balanced four-splitters. In the rest of this work, where the context is clear, the term *four-splitter* is used to mean a real balanced four-splitter, which is the focus of our work.

### A. Physical constructions of four-splitters

First, we prove the following theorem about beam-splitter networks:

*Theorem 2.* Let  $\hat{B}_{\text{network}}$  be a beam splitter network comprising four real balanced beam splitters [Eq. (2)]. Then

$\hat{B}_{\text{network}}$  is a real balanced four-splitter if and only if all of the following conditions are satisfied:

- (1) Each mode interacts with exactly two beam splitters.
- (2) All modes interact with a first beam splitter before any mode interacts with a second.
- (3) No pair of beam splitters interacts with the same two modes.

*Proof.* Let  $\hat{B}_{\text{network}} \rightarrow \mathbf{R}$ . We proved the result by exhaustive search. Each beam splitter can start at any of the four modes and end at any of the three other modes ( $4 \times 3$ ), and there are four of these choices to be made independently, giving a total of  $(4 \times 3)^4 = 20\,736$  possible versions of  $\mathbf{R}$ . We constructed each of these matrices and checked if it was a real balanced four-splitter matrix. Independently, we checked whether each beam-splitter network satisfied each of the three conditions individually and in combination. We found, by comparing these lists, that satisfaction of all three conditions is both necessary and sufficient for  $\mathbf{R}$  to be a real balanced four-splitter matrix. ■

A bit more intuition into this result is possible. Given conditions 2 and 3, without loss of generality, we can choose to partition the four modes, labeled  $(j, k, \ell, m)$ , as  $\{\{j, k\}, \{\ell, m\}\}$ , such that the first two beam splitters act on modes within the same partition. Condition 1 now guarantees that the last two beam splitters act across the partition  $\{\{j, \ell\}, \{k, m\}\}$ . (If this partitioning is not respected, just swap mode labels  $\ell$  and  $m$  in the first step.) This means that  $\mathbf{R}$  has the form

$$\mathbf{R} = \frac{1}{2} \begin{bmatrix} \pm & 0 & \pm & 0 \\ 0 & \pm & 0 & \pm \\ \pm & 0 & \pm & 0 \\ 0 & \pm & 0 & \pm \end{bmatrix} \begin{bmatrix} \pm & \pm & 0 & 0 \\ \pm & \pm & 0 & 0 \\ 0 & 0 & \pm & \pm \\ 0 & 0 & \pm & \pm \end{bmatrix}, \quad (78)$$

where each  $\pm$  independently represents either  $+1$  or  $-1$  (fixed by the beam splitters in question), and the row and column ordering are  $(j, k, \ell, m)$ . This block form guarantees that  $\mathbf{R}$  has entries with equal magnitude. The fact that it represents a product of real beam-splitter matrices guarantees that  $\mathbf{R} \in \text{O}(4)$ , which proves sufficiency of the conditions. We did not find an elegant way to rigorously prove necessity of the conditions, but the exhaustive search guarantees that

$$\mathbf{R} = \mathbf{R}_{km}^{(T)} \mathbf{R}_{j\ell}^{(T)} \mathbf{R}_{\ell m}^{(T)} \mathbf{R}_{jk}^{(T)} =: \mathbf{R}_{4\text{-split}} \quad (79)$$

is a real balanced four-splitter matrix, where  $(T)$  indicates that the matrix shown may be transposed or not. This means the form above allows each beam splitter to point in either direction between the modes in the subscript. Note that the first two matrices commute,  $[\mathbf{R}_{jk}^{(T)}, \mathbf{R}_{\ell m}^{(T)}] = 0$ , and so do the last two,  $[\mathbf{R}_{j\ell}^{(T)}, \mathbf{R}_{km}^{(T)}] = 0$ —the corresponding beam-splitter pairs exhibit the same commutations.

Next, we note that not all four-splitters can be decomposed into four real beam splitters alone, each of the form in Eq. (2). This can be seen with a simple counting argument. Consider four beam splitters over four modes arranged according to Eq. (79). There are  $4!$  ways to freely assign mode labels and  $2^4$  ways to choose to transpose or not each beam-splitter matrix. This gives  $4! \times 2^4 = 384$  beam-splitter networks. Noting the commutativity within the beam-splitter pairs, as described below Eq. (79), we can reduce this total by a factor of  $2^2$  to ac-

count for this possible reordering of the beam splitters within each pair. Accounting for this operational equivalence (since the beam splitters can act simultaneously) gives  $384/4 = 96$  physically distinct beam-splitter networks.

By explicit computation, we found that these 96 physical beam-splitter networks generate only 40 unique four-splitters out of the 768 total. This is due to collisions, where more than one beam-splitter network produces the same four-splitter matrix. Indeed, it was observed previously that pairs of beam splitters in the QRL commute [4,5,33],

$$\hat{B}_{\text{QRL}} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array}, \quad (80)$$

giving a two-to-one correspondence with a four-splitter matrix.<sup>6</sup> We find that 48 of the 96 beam-splitter networks exhibit the same behavior: one pair of beam splitters commutes with the other pair. Consequently, this set generates 24 four-splitters. The remaining 48 physical beam-splitter networks have a different property that gives three-to-one correspondences with four-splitters. Each beam-splitter network of this type is equivalent to two other beam-splitter networks. An example is the cBSL,

$$\hat{B}_{\text{cBSL}} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \quad (81)$$

The 48 physical beam-splitter networks of this type produce the remaining 16 four-splitters and complete the set of 40 that are possible using beam-splitter networks alone.

The above reveals that we must go beyond beam-splitter networks alone in order to construct the full set of 768 four-splitters. However, starting from any of these 40, Theorem 1 guarantees that we can generate the full equivalence class given three types of matrix operation: row swaps, row negation, and a column negation. These matrix operations are implemented physically with SWAP gates and parity operators, as described in Eqs. (9) and (11), respectively. Following Theorem 1, we include a single  $\hat{F}^2$  before the beam-splitter network, up to four  $\hat{F}^2$  after it (one for each mode), and final SWAP gates that may permute the modes in any way.

Starting from the 96 physical beam-splitter networks constructed above, two choices of parity operator beforehand,  $2^4$  afterwards, and  $4!$  mode swaps gives  $96 \times 2 \times 2^4 \times 4! = 73\,728$  total linear optical networks. In fact, this is exactly 96 times 768, the number of unique four-splitters. Theorem 1 guarantees that each four-splitter is represented. The question

<sup>6</sup>Unlike the commutation within beam-splitter pairs, this commutation relation still requires a definite ordering of the pairs to be chosen in the experiment. It just so happens that either ordering gives the same result.

is how many times each one appears. We numerically computed every network's orthogonal matrix and found that each unique four-splitter is realized by 96 different linear-optical networks.<sup>7</sup>

### B. Incomplete four-splitters and how to complete them

Most of the cluster states we analyzed in Sec. III have at their heart beam-splitter networks comprising three beam splitters (rather than four). In each case, we went through a procedure to *complete* them with a fourth beam splitter added either virtually by restricting measurement angles or, in the case of the MBSL, by inserting the fourth beam splitter physically. These illustrate the two types of incomplete beam-splitter network we consider: either the missing fourth beam splitter is (a) on the measurement side of the circuit or (b) on the state side of the circuit.

The two types of incomplete four-splitter correspond to respectively dropping (a)  $\mathbf{R}_{km}^{(T)}$  or (b)  $\mathbf{R}_{jk}^{(T)}$  in Eq. (79). For type (a), this gives orthogonal matrices where two of the rows have two 0 entries and two  $\pm\sqrt{2}$  entries, such as for the BSL in Eq. (32). For type (b), the orthogonal matrices have two columns with two 0 entries and two  $\pm\sqrt{2}$  entries, such as for the MBSL in Eq. (64). Importantly, the operations that define the equivalence class in Theorem 1 do not couple the two types of incomplete four-splitter; doing so would require a transpose-type operation, which is not available.

Restated in terms of physical operations: beam-splitter networks of type (b) cannot be transformed to type (a) using the left action of SWAPs and parity operators, i.e., on the measurement side. If this were possible, it would allow type (b) to be completed by restricting measurement angles in a teleportation gadget.

One last situation must be addressed. Is it possible to complete a type (b) beam-splitter network by restricting measurements and postprocessing the outcomes in a more general way? We show in Appendix D that the answer is “no,” which ultimately means that incomplete four-splitters of type (b) cannot be completed by restricting measurements—the fourth beam splitter must be included in another way.<sup>8</sup>

## VI. CONCLUSION

By demonstrating an equivalence to the well-studied QRL architecture, we have shown that the BSL, DBSL, MSG, and MBSL architectures are all compatible with the streamlined quantum computing techniques introduced in Ref. [32], which provide a complete Clifford gate set for the GKP code within a single teleportation gadget, minimizing gate noise arising from noise in the ancillae.

<sup>7</sup>Note: This is 96 linear-optical networks that construct a *single* unique balanced four-splitter, as opposed to the 96 physical beam-splitter networks, referred to earlier, which can generate 40 different balanced four-splitters between them.

<sup>8</sup>Of course, *any* real beam-splitter network can be generated virtually by choosing all measurement angles to be the same. This does nothing of value, however, since it also trivially undoes any other beam splitters on the associated modes.

To show this equivalence, we identify macronode wires that periodically couple to one another throughout each cluster state. This identification allows the cluster states to be used without introducing additional noise that arises from deleting modes.

The equivalence to the QRL requires *completing the four-splitter* within each type of cluster state. For the BSL, DBSL, and MSG, correlating specific measurement angles produces virtual beam splitters that complete the four-splitters, such that the couplings behave just like the ones in the QRL do, up to SWAP gates and single-mode parity operators. Once completed, each cluster state has access natively to a generating set of GKP Clifford gates (inherited from the QRL but limited to those that satisfy the angle restrictions), which have the same intrinsic gate noise as the QRL [32]. These are sufficient to implement any Clifford gate.<sup>9</sup>

Physically completing the four-splitter (required for the MBSL, optional for the others) eliminates these angle restrictions, allowing all QRL gates to be implemented natively on the new cluster state. This shows that all architectures based on complete four-splitters are entirely equivalent in terms of their available gates and noise properties due to finite squeezing.

The specific cluster states in Sec. III are a small subset of those available to experimentalists with passive optical components. The analysis in Sec. V of all balanced four-splitters that are equivalent under trivial passive operations (SWAP and parity gates) provides new pathways to designing scalable cluster states. In practice, the class of four-splitters (and the cluster states they comprise) that can be completed by correlating measurement angles may be more useful than the QRL-type cluster states due to fewer optical components (one fewer beam splitter). That is because fewer optical components can allow for simpler experimental set ups.

Further, the noise equivalence we show here applies only to gate noise introduced by imperfect GKP states throughout the cluster. Optical components themselves introduce losses and other noise due to imperfections and alignment errors; reducing their number could improve gate noise under more realistic models. On the other hand, the connectivity in the context of each cluster state at large may favor a true four-splitter in the teleportation gadget in order to leverage more flexibility for routing quantum information [33] and implementing simultaneous gates. For a given computation, there may be advantages to choosing a specific cluster state to reduce logical gate depth and consequently logical noise.

In this work we focus on Clifford gates for the square-lattice GKP code. It is not known whether cluster states can implement a full set of Clifford gates for GKP codes on other lattices in a single teleportation step, and, if so, *which* cluster states do. Our general analysis of four-splitters should be useful in this study: once a particular four-splitter and measurement angles are found that provide the full gate set,

<sup>9</sup>Some work remains to discern the effects of these measurement restrictions on any specific architecture, and we leave that to those doing the practical implementations. We have shown, nevertheless, that a generating set of Clifford gates is available natively for each architecture.



where  $\hat{V}(\theta_1, \theta_2) = \hat{R}(\theta_1 - \frac{\pi}{2})\hat{P}(2 \cot \theta_-)\hat{R}(\theta_1 - \frac{\pi}{2})$ , and the displacement amplitude is

$$\mu_{1,2} = -\frac{m_1 e^{i\theta_2} + m_2 e^{i\theta_1}}{\sin(\theta_1 - \theta_2)}. \quad (\text{A11})$$

For  $\theta_1 - \theta_2 = n\pi$ , with integer  $n$ , these expressions are not defined. This corresponds to the case where the two modes in Eq. (A4) are measured in the same basis (up to parity). This effectively removes the beam splitter, uncoupling the modes and precluding teleportation.

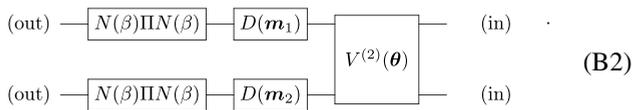
## APPENDIX B: DISPLACEMENTS FOR TWO-MODE GATES

As presented in the main text, the teleported gate in Eq. (75) lacks a two-mode displacement operator originating from the four measurement outcomes in the teleportation gadget. We give its form here since knowing the displacement is necessary in practice to perform corrections during the teleportation procedure.

Up to a local parity operator but including the single-mode outcome-dependent displacements, the two-mode gate realized by the completed teleportation gadgets in Sec. III take the form

$$\hat{V}^{(2)}(\boldsymbol{\theta}, \boldsymbol{m}) = \hat{B}_{1,2}\left(\pm \frac{\pi}{4}\right) \left[ \begin{array}{c} \hat{D}_1(\mu_{1,2})\hat{V}_1(\theta_1, \theta_2) \\ \otimes \\ \hat{D}_2(\mu_{3,4})\hat{V}_2(\theta_3, \theta_4) \end{array} \right] \hat{B}_{1,2}\left(\pm \frac{\pi}{4}\right), \quad (\text{B1})$$

where  $\boldsymbol{m}$  stands in for the four outcomes, and we use vertical tensor product notation to shorten the expression. The  $\pm$  and  $\mp$  are independent signs (indicating the different possible directions of each 50:50 beam splitter), and each single-mode displacement amplitude is given by Eq. (A11). Pushing the displacement operators to the left through the beam splitter gives the more general form of Eq. (75),



$$\begin{array}{l} (\text{out}) - \boxed{N(\beta)\Pi N(\beta)} - \boxed{D(m_1)} - \text{---} - \text{---} - (\text{in}) \\ (\text{out}) - \boxed{N(\beta)\Pi N(\beta)} - \boxed{D(m_2)} - \text{---} - \text{---} - (\text{in}) \end{array} \quad (\text{B2})$$

The amplitudes are mixed by the beam splitter to give

$$\boldsymbol{m}_1 = \frac{1}{\sqrt{2}}(\mu_{1,2} \pm \mu_{3,4}), \quad (\text{B3})$$

$$\boldsymbol{m}_2 = \frac{1}{\sqrt{2}}(\mu_{3,4} \mp \mu_{1,2}), \quad (\text{B4})$$

with the sign  $\pm$  determined by the sign in the leftmost beam splitter.

## APPENDIX C: CIRCUIT IDENTITY FOR BEAM SPLITTERS OVER THREE MODES

Consider beam splitters that contain a transmission parameter  $\theta$ ,

$$\hat{B}_{jk}(\theta) := e^{-\theta(\hat{a}_j \hat{a}_k^\dagger - \hat{a}_j^\dagger \hat{a}_k)} = \begin{array}{c} \text{---} j \\ | \\ \theta \\ | \\ \text{---} k \end{array} \quad (\text{C1})$$

with  $\theta \in \{-\pi, \pi\}$ , described by orthogonal matrices

$$\mathbf{R}_{jk}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (\text{C2})$$

which are identical in form to 2D rotation matrices.

When a third mode is included, labeled by  $\ell$ , the matrix in Eq. (C2) expands into the 3D matrix

$$\mathbf{R}_{jk}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{C3})$$

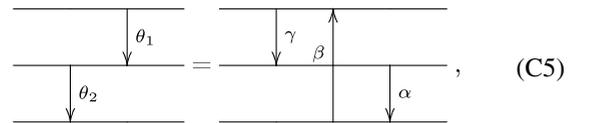
which can be interpreted as a rotation in three dimensions around principle  $z$  axis by angle  $\theta$ ,  $\mathbf{R}_{jk}(\theta) = \mathbf{R}_z(\theta)$ . Beam splitters between other pairs of modes describe rotations around the other principle axes:  $\mathbf{R}_{k\ell}(\theta) = \mathbf{R}_x(\theta)$  for  $x$  rotations, and  $\mathbf{R}_{\ell j}(\theta) = \mathbf{R}_y(\theta)$  for  $y$  rotations. Note that swapping the subscripts on any matrix takes  $\theta \rightarrow -\theta$ , which reverses the direction of the arrow in the circuit diagram on the right-hand side of Eq. (C1).

This association leads to several important consequences. Any beam-splitter network over three modes containing any number of real beam splitters can be described by a 3D rotation matrix. Decomposing this matrix using Euler angles or Tait-Bryan angles gives its description in terms of three principle-axis rotation matrices, each corresponding to a single beam splitter. Thus, any three-mode beam-splitter network (with real beam splitters) can be reduced to three beam splitters or fewer.

As an example, consider reordering a beam-splitter network of two beam splitters that share one mode,  $\hat{B}_{k\ell}(\theta_2)\hat{B}_{jk}(\theta_1)$ . Suppose we seek the specific Tait-Bryan decomposition into fundamental-axis rotations in the order  $z - y - x$  (right-to-left). The associated rotation matrices satisfy<sup>11</sup>

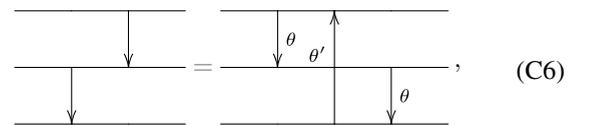
$$\mathbf{R}_x(\theta_2)\mathbf{R}_z(\theta_1) = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha) \quad (\text{C4})$$

corresponding to the circuit equivalence,



$$\begin{array}{c} \text{---} \\ | \\ \theta_1 \\ | \\ \text{---} \\ | \\ \theta_2 \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \gamma \\ | \\ \beta \\ | \\ \alpha \\ | \\ \text{---} \end{array}, \quad (\text{C5})$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are functions of  $\theta_1$  and  $\theta_2$ . For the case of two balanced beam splitters,



$$\begin{array}{c} \text{---} \\ | \\ \theta \\ | \\ \text{---} \\ | \\ \theta' \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \theta \\ | \\ \theta' \\ | \\ \theta \\ | \\ \text{---} \end{array}, \quad (\text{C6})$$

where  $\theta = \arctan \frac{1}{\sqrt{2}}$  and  $\theta' = \arctan -\frac{\sqrt{3}}{3}$ .

<sup>11</sup>Note that the corresponding rotation is already in an Euler form,  $\mathbf{R}_z(0)\mathbf{R}_x(\theta_2)\mathbf{R}_z(\theta_1)$ , and a different Tait-Bryan form,  $\mathbf{R}_y(0)\mathbf{R}_x(\theta_2)\mathbf{R}_z(\theta_1)$ .

**APPENDIX D: TYPE (b) INCOMPLETE  
FOUR-SPLITTERS CANNOT BE COMPLETED  
BY RESTRICTING MEASUREMENTS**

Here we show that type (b) incomplete four-splitters, which lack a fourth beam splitter on the state side of the beam-splitter network, cannot be completed by choosing specific measurement angles and postprocessing of the measurement outcomes. Consider a beam-splitter network of this type,  $\hat{B}^{(3)}$ , that, when completed, gives a four-splitter  $\hat{B}_{4\text{-split}}$ .  $\hat{B}^{(3)}$  is equivalent to the desired four-splitter  $\hat{B}_{4\text{-split}}$  with a residual operator on the measurement side,

$$\hat{B}^{(3)} = \hat{B}_{\text{res}} \hat{B}_{4\text{-split}} \quad (\text{D1})$$

where the residual operator  $\hat{B}_{\text{res}} := \hat{B}^{(3)} \hat{B}_{4\text{-split}}^\dagger$  is itself another network of real beam splitters.  $\hat{B}_{\text{res}} \rightarrow \mathbf{R}_{\text{res}}$  has an associated orthogonal matrix  $\mathbf{R}_{\text{res}}$  composed from the individual beam-splitter matrices in the network.

The question is whether there exist measurement angles  $\theta := \{\theta_1, \theta_2, \theta_3, \theta_4\}$  and some linear transformation of the measurement outcomes that, together, virtually delete the residual beam-splitter network  $\hat{B}_{\text{res}}$ , i.e.,

$${}_p \langle \mathbf{m} | \hat{B}_{\text{res}} = \sqrt{|\det \mathbf{L}^{-1}|} {}_{p'} \langle \mathbf{L}^{-1} \mathbf{m} |. \quad (\text{D2})$$

We consider a Heisenberg-picture linear transformation described by  $\mathbf{L}_{\text{res}}$  more general than that of the orthogonal matrix  $\mathbf{R}_{\text{res}}$  in order to allow for the possibility that postprocessing beyond orthogonal transformation may be required. For example, scaling the outcome on a single mode can insert a squeezing operator:  ${}_q \langle m | = \sqrt{\zeta} {}_q \langle \zeta m | \hat{S}(\zeta)$  [34]. Note that doing so introduces a change in measure, which is the origin of the determinant in Eq. (D2). Additionally,  $\mathbf{L}_{\text{res}}$  can only linearly transform outcomes in the chosen measurement bases; we further include the possibility that deletion also requires a new set of angles  $\theta'$ .

Theorem 3, proved below, shows that  $\hat{B}_{\text{res}}$  cannot be deleted unless all the modes are measured in the same basis,  $\theta_j = \theta_k$  for all  $j, k$ , which trivially disentangles them. For Theorem 3 to apply here, a further property of  $\mathbf{R}_{\text{res}}$  is required: it must be neither block diagonal nor row permutations away from block diagonal. This can be shown using its decomposition into constituent matrices, Eq. (79). Without loss of generality, we choose an incomplete four-splitter missing a beam splitter between modes  $j$  and  $k$  in Eq. (79), i.e.,  $\mathbf{R}^{(3)} = \mathbf{R}_{km} \mathbf{R}_{jl} \mathbf{R}_{lm}$ .<sup>12</sup> Then we have

$$\mathbf{R}_{\text{res}} = \mathbf{R}^{(3)} \mathbf{R}_{4\text{-split}}^\text{T} = \mathbf{R}_{km} \mathbf{R}_{jl} \mathbf{R}_{jk}^\text{T} \mathbf{R}_{jl}^\text{T} \mathbf{R}_{km}^\text{T}. \quad (\text{D3})$$

Every type (b) residual beam-splitter network has this form—a pair of commuting beam splitters on the left, a pair of commuting beam splitters on the right, and a fifth beam splitter that commutes with neither of the pairs.

We observe that the matrix in Eq. (D3) has no zero entries. To show this, let us consider the following permutation

$\{m, k, j, \ell\} \rightarrow \{1, 2, 3, 4\}$  and define the matrix  $\mathbf{Q}$  that realizes this permutation for any  $\{m, k, j, \ell\}$ . Then we have<sup>13</sup>

$$\begin{aligned} \mathbf{Q} \mathbf{R}_{\text{res}} \mathbf{Q}^{-1} &= \mathbf{R}_{12} \mathbf{R}_{34} \mathbf{R}_{23} \mathbf{R}_{43} \mathbf{R}_{21} \quad (\text{D4}) \\ &= \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 + \sqrt{2} & 1 - \sqrt{2} & -1 & -1 \\ 1 - \sqrt{2} & 1 + \sqrt{2} & -1 & -1 \\ 1 & 1 & 1 + \sqrt{2} & 1 - \sqrt{2} \\ 1 & 1 & 1 - \sqrt{2} & 1 + \sqrt{2} \end{pmatrix}, \end{aligned} \quad (\text{D5})$$

which has no zero entries.<sup>14</sup> Since permutation does not change the values of entries,  $\mathbf{R}_{\text{res}}$  itself has no zero entries. The next theorem essentially states that the beam-splitter network that corresponds to such a matrix cannot be canceled as in Eq. (D2) for any nontrivial choices of the measurement angles  $\theta$ . (That is, the only choice that works is all angles equal, which virtually deletes the entire beam-splitter network.)

*Theorem 3.* Let  $\mathbf{m}$  denote a vector of  $N$  homodyne measurement outcomes and  $\theta$  denote a vector of  $N$  choices of homodyne-measurement angles with  $\theta_j \in (-\frac{\pi}{2}, \frac{\pi}{2}]$  for all  $j \in \{1, \dots, N\}$ . Let  $\hat{B}^{(N)} \rightarrow \mathbf{R}$  be a real beam-splitter network, where  $\mathbf{R} \in \text{O}(N)$  is an  $N$ -dimensional real orthogonal matrix that is neither block diagonal nor row permutations away from block diagonal. Let  $\theta'$  be another vector of  $N$  choices of homodyne-measurement angles, with  $\theta'_i \in (-\frac{\pi}{2}, \frac{\pi}{2}]$  for all  $i \in \{1, \dots, N\}$ . Finally, let  $\mathbf{L} \in \mathbb{R}^{N \times N}$  be an  $N$ -dimensional real matrix such that there exists a corresponding Gaussian unitary  $\hat{U}_G$  that satisfies the following:<sup>15</sup>

$$\sqrt{|\det \mathbf{L}^{-1}|} {}_p \langle \mathbf{L}^{-1} \mathbf{m} | = {}_p \langle \mathbf{m} | \hat{U}_G. \quad (\text{D6})$$

Then, given  $\theta$  and  $\hat{B}^{(N)}$ , there does not exist a pair  $(\theta', \mathbf{L})$  that satisfies

$$\forall \mathbf{m} \in \mathbb{R}^N, \quad {}_{p'} \langle \mathbf{m} | \hat{B}^{(N)} = \sqrt{|\det \mathbf{L}^{-1}|} {}_{p'} \langle \mathbf{L}^{-1} \mathbf{m} |, \quad (\text{D7})$$

unless  $\theta = \theta \mathbf{1}_N$ , where  $\mathbf{1}_N = (1, 1, \dots, 1)^\text{T}$ .

*Proof.* Let  $\hat{R}^{(N)}(\theta)$  be the  $N$ -mode local phase rotation with the rotation angles given by  $\theta$ . Then the condition (D7) is equivalent to

$$\forall \mathbf{m} \in \mathbb{R}^N, \quad {}_p \langle \mathbf{m} | \hat{R}^{(N)}(\theta) \hat{B}^{(N)} = {}_p \langle \mathbf{m} | \hat{U}_G \hat{R}^{(N)}(\theta'). \quad (\text{D8})$$

Since this holds for all  $\mathbf{m} \in \mathbb{R}^N$ , we have

$$\hat{R}^{(N)}(\theta) \hat{B}^{(N)} = \hat{U}_G(\mathbf{L}) \hat{R}^{(N)}(\theta'). \quad (\text{D9})$$

Now the left-hand side, as well as  $\hat{R}^{(N)}(\theta')$ , is a linear-optical unitary and thus does not contain squeezing. Therefore, in order for the above to be equal,  $\hat{U}_G$  must also be a linear-optical unitary:  $\hat{U}_G \rightarrow \mathbf{U} \in \text{U}(N)$ . From Eq. (D6),  $\hat{U}_G$  does not mix  $\hat{p}$ s and  $\hat{q}$ s, and  $\mathbf{L}$  is the (real) matrix by which the  $\hat{p}$  evolve. This restricts  $\mathbf{U}$  such that  $\mathbf{U} = \mathbf{L} \in \text{O}(N)$ , i.e., it is real orthogonal.

<sup>13</sup>With the transposes restored in Eq. (79), the signs may change in Eq. (D5), but the matrix still has no zeros.

<sup>14</sup>Note that this is the residual matrix associated with the MBSL in Sec. III G.

<sup>15</sup>The determinant is required because of the following identity:  $\int d\mathbf{m} \hat{U}_G^\dagger(\mathbf{L}) |\mathbf{m}\rangle_{p'} \langle \mathbf{m} | \hat{U}_G(\mathbf{L}) = \int d\mathbf{m} |\det \mathbf{L}^{-1}| |\mathbf{L}^{-1} \mathbf{m}\rangle_{p'} \langle \mathbf{L}^{-1} \mathbf{m} |$ .

<sup>12</sup>For clarity of presentation, we assume no transposes on the matrices in Eq. (79). The result still holds with them restored, however; the notation just becomes cluttered.

This means that  $\hat{U}_G$  is a real beam-splitter network denoted by  $\hat{B}^{(N)} \rightarrow \mathbf{R}' \in O(N)$ , with  $\mathbf{R}' = \mathbf{L}$ .

Equation (D9) can now be rewritten as

$$\hat{R}^{(N)}(\boldsymbol{\theta})\hat{B}^{(N)} = \hat{B}'^{(N)}\hat{R}^{(N)}(\boldsymbol{\theta}'). \quad (\text{D10})$$

Since they are linear-optical unitaries, we consider their representations as unitary matrices acting on the vector of annihilation operators. It is known that for a beam-splitter network and a local phase rotation,  $\hat{B}^{(N)} \rightarrow \mathbf{R}$  and  $\hat{R}^{(N)}(\boldsymbol{\theta}) \rightarrow \text{diag}(e^{i\theta})$ , where  $\text{diag}(e^{i\theta})$  is a diagonal matrix of phases. With these in mind, one can see that Eq. (D10) is equivalent to

$$\text{diag}(e^{i\theta})\mathbf{R} = \mathbf{R}'\text{diag}(e^{i\theta}'). \quad (\text{D11})$$

Taking the transpose of the both sides leads to

$$\mathbf{R}^T \text{diag}(e^{i\theta}) = \text{diag}(e^{i\theta}')\mathbf{R}'^T. \quad (\text{D12})$$

Combining Eqs. (D12) and (D11) gives

$$\mathbf{R}^T \text{diag}(e^{2i\theta})\mathbf{R} = \text{diag}(e^{2i\theta}'). \quad (\text{D13})$$

The above implies that the left-hand side is the diagonalization of the right-hand side, which is itself diagonal. Since the spectrum of a normal matrix is unique up to a permutation of the eigenvalues, we obtain

$$(\mathbf{P}\mathbf{R})^T \text{diag}(e^{2i\theta}')(\mathbf{P}\mathbf{R}) = \text{diag}(e^{2i\theta}'), \quad (\text{D14})$$

where  $\mathbf{P}$  is a permutation matrix that satisfies  $\mathbf{P}\text{diag}(e^{2i\theta}')\mathbf{P}^T = \text{diag}(e^{2i\theta})$ . Equation (D14) means that  $\mathbf{P}\mathbf{R}$  and  $\text{diag}(e^{2i\theta})$  commute. However, it is known that any matrix that commutes with a diagonal matrix has a block-diagonal form that acts nontrivially only within the eigenspaces of degenerate eigenvalues. Given that  $2\theta'_j \in (-\pi, \pi]$  for all  $j \in \{1, \dots, N\}$ , Eq. (D14) contradicts the initial assumption that  $\mathbf{R}$  is neither block diagonal nor a row permutation away from block diagonal unless all the elements of  $\boldsymbol{\theta}'$  are the same. In this exceptional case, we are restricted to choosing<sup>16</sup>  $\boldsymbol{\theta} = \boldsymbol{\theta}' = \theta\mathbf{1}_N$  by Eq. (D14), which completes the proof. ■

Note that the restriction  $\theta_j \in [-\frac{\pi}{2}, \frac{\pi}{2})$  for any  $j \in \{1, \dots, N\}$  does not imply the loss of generality because other measurement angles can be simulated by putting negative signs on the measurement outcomes with appropriate measurement angles in this range. As we see in Eq. (D5) the matrix  $\mathbf{R}_{\text{res}}$  has no zero entries and thus is neither block diagonal nor row permutation away from block diagonal. This fact combined with the above theorem means that, putting aside the trivial case  $\boldsymbol{\theta} = \theta\mathbf{1}_N$ , it is impossible to absorb the residual beam splitter  $\hat{B}_{\text{res}}$  by the change of the measurement angles as well as the linear postprocessing of the measurement outcomes. Combining this with Eq. (D1) implies that this (equivalence class of) incomplete four-splitter  $\hat{B}^{(3)}$  [i.e., type (b)] cannot be made equivalent to a four-splitter (followed by homodyne measurement and reinterpretation of its outcomes) with nontrivial choices of measurement angles.

<sup>16</sup>Technically, they can differ by integer multiples of  $\pi$ , which is what the 2 in  $\text{diag}(e^{2i\theta})$  allows. Since these are physically identical (just negate the outcome of the measurement), this freedom of definition provides no benefit, and we do not dwell on it.

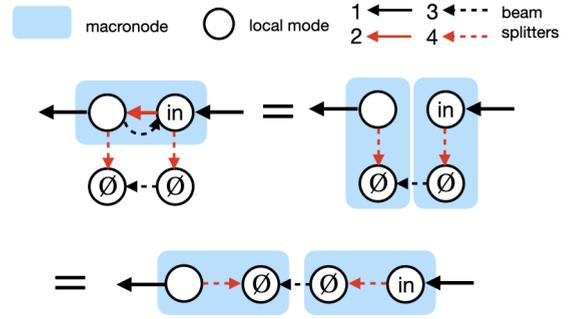


FIG. 6. GKP Bell pair insertion using a balanced four-splitter. Empty circles are local modes in unspecified states, circles filled with  $\emptyset$  are local modes prepared in pure GKP qunaught states (noisy versions also work). This process inserts two  $\emptyset$  states in-line in a single-mode macronode wire (either isolated or embedded in a two-dimensional cluster state). Described from left to right, two inputs,  $|\emptyset\rangle$ , are introduced alongside an existing macronode wire, and a canonical balanced four-splitter is used to interact the two states with a two-mode macronode. The two beam splitters contained within the macronode cancel out. We then regroup the modes into two new macronodes, each one containing a single  $\emptyset$  state. Each of these macronodes, when teleported through, performs error correction on either the position or momentum quadratures [34]. Together, full GKP error correction is possible between these two macronodes. Note that one is not required to use qunaught states. This circuit is, more generally, a way to insert a Kraus state  $\hat{B}_{12}|\psi\rangle \otimes |\phi\rangle$  into a macronode wire.

## APPENDIX E: GKP BELL PAIR INSERTION

The noise analysis in Ref. [32] for the QRL assumes that the CV cluster state is entirely constructed from GKP qunaught states. In practice, a deterministic source of GKP qunaughts may not be available, and the cluster state may have locations where squeezed momentum states were swapped in to fill in the gaps [26,29,30,32]. Here we give a procedure to insert a GKP Bell pair—or any other state made from two single-mode states coupled on a beam splitter—into a macronode wire (which may be a part of a larger cluster state) after it has been created.

By undoing one of the wire's original beam splitters, an additional pair of modes containing a GKP Bell pair can be inserted into a macronode wire; see Fig. 6. This procedure is implemented by the circuit<sup>17</sup>

$$\begin{array}{c}
 p_{\theta_1} \langle m_1 | \\
 p_{\theta_a} \langle m_a | \\
 p_{\theta_b} \langle m_b | \\
 p_{\theta_2} \langle m_2 |
 \end{array}
 \begin{array}{c}
 \text{---} \\
 \text{---} \\
 \text{---} \\
 \text{---}
 \end{array}
 \begin{array}{c}
 \uparrow \\
 \downarrow \\
 \downarrow \\
 \downarrow
 \end{array}
 \begin{array}{c}
 1 \\
 a \\
 b \\
 2
 \end{array}
 \quad (\text{E1})$$

<sup>17</sup>This circuit starts from the red beam splitters in Fig. 6. The black ones are omitted because they link the top and bottom wires to other wires not shown.

where modes 1 and 2, originally coupled by the red beam splitter on the right-hand side, are those from the macronode wire. Modes  $a$  and  $b$  are the new modes containing GKP Bell quanaught states. The dashed beam splitter from the mode 2 to 1 eliminates the red one, leaving

$$\begin{array}{c}
 p_{\theta_1} \langle m_1 | \text{-----} 1 \\
 \phantom{p_{\theta_1}} \phantom{\langle m_1 |} \phantom{\text{-----}} \phantom{1} \\
 p_{\theta_a} \langle m_a | \text{-----} |\emptyset\rangle \quad a \\
 \phantom{p_{\theta_a}} \phantom{\langle m_a |} \phantom{\text{-----}} \phantom{|\emptyset\rangle} \phantom{a} \\
 p_{\theta_b} \langle m_b | \text{-----} |\emptyset\rangle \quad b \\
 \phantom{p_{\theta_b}} \phantom{\langle m_b |} \phantom{\text{-----}} \phantom{|\emptyset\rangle} \phantom{b} \\
 p_{\theta_2} \langle m_2 | \text{-----} 2
 \end{array} \quad (E2)$$

This circuit is simplified with application of Eq. (A10) [34] to give

$$\begin{array}{c}
 D(\mu) \quad V(\theta_1, \theta_a) \\
 \Pi_{\text{GKP}} \\
 D(\mu') \quad V(\theta_b, \theta_2)
 \end{array} \quad (E3)$$

and  $\mu$  ( $\mu'$ ) is a function of  $m_1$  and  $m_a$  ( $m_b$  and  $m_2$ ). This adds two further measurement degrees of freedom, enabling an additional  $\hat{V}$  gate, but only offers GKP error correction after the first one. Inserting GKP Bell pairs throughout the macronode wires is effectively the same as including GKP states at the beginning of cluster state generation on half of the ancillas.

**APPENDIX F: CV CIRCUIT IDENTITIES**

Here we list a collection of useful decompositions and identities for CV circuit elements.

*Beam splitter decomposition:*

$$\theta = \begin{array}{c} S^\dagger(\sec \theta) \\ \tan \theta \quad -\tan \theta \\ S(\sec \theta) \end{array} \quad (F1a)$$

$$= -\tan \theta \begin{array}{c} S(\sec \theta) \\ \tan \theta \\ S^\dagger(\sec \theta) \end{array} \quad (F1b)$$

where  $\hat{C}_X^{jk}(g) := e^{-ig\hat{q}_j \otimes \hat{p}_k}$  describes the controlled-shift gates in the circuit, with the control being mode  $j$  and target mode  $k$ .

*Reversing a beam splitter:*

$$\begin{array}{c} \times \\ \times \\ \times \end{array} \theta = \begin{array}{c} F^2 \\ \theta \\ F^2 \end{array} = \begin{array}{c} \times \\ \times \\ \times \end{array} \theta \quad (F2)$$

*$\hat{C}_Z$  decomposition:*

$$g = \begin{array}{c} P(-g) \\ P(g) \end{array} \quad (F3)$$

where  $\hat{C}_Z^{jk}(g) := e^{ig\hat{q}_j \otimes \hat{q}_k}$  describes the controlled-shift gates on the left-hand side.

*SWAP decompositions:*

$$\begin{array}{c} \times \\ \times \end{array} = \begin{array}{c} F^2 \\ \oplus \quad -1 \\ \oplus \end{array} \quad (F4a)$$

$$= \begin{array}{c} F^2 \\ \downarrow \quad \downarrow \end{array}, \quad (F4b)$$

Interestingly, this implies that beam splitters as defined in Eq. (1) act on CV systems in analogy to the way  $\sqrt{\text{SWAP}}$  gates act on qubits or qudits.

*EPR state:*

$$|\text{EPR}\rangle := \int ds |s\rangle_q \otimes |s\rangle_q = \begin{array}{c} \text{---} \\ \text{---} \end{array} \quad (F5)$$

*The teleported gate:*

$$\begin{array}{c} |\psi\rangle \\ \downarrow \\ |\phi\rangle \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ \frac{1}{\sqrt{\pi}} A(\psi, \phi) \end{array} \quad (F6)$$

where  $\hat{A}(\psi, \phi) = \iint d^2\alpha \tilde{\psi}(\alpha_L)\phi(\alpha_R)\hat{D}(\alpha)$ .

*Basic bounce:*

$$\begin{array}{c} O \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ O^T \end{array} \quad (F7)$$

with the transpose defined in the position basis.

*Beam-splitter double bounce:*

$$\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (F8)$$

*$\hat{C}_X$  reordering:*

$$\begin{array}{c} \oplus \\ b \\ \oplus \\ a \end{array} = \begin{array}{c} \oplus \\ b \\ \oplus \\ a \end{array} \quad (F9)$$

- [1] N. C. Menicucci, P. van Loock, M. Gu, C. Weedbrook, T. C. Ralph, and M. A. Nielsen, Universal quantum computation with continuous-variable cluster states, *Phys. Rev. Lett.* **97**, 110501 (2006).
- [2] M. Gu, C. Weedbrook, N. C. Menicucci, T. C. Ralph, and P. van Loock, Quantum computing with continuous-variable clusters, *Phys. Rev. A* **79**, 062318 (2009).
- [3] S. T. Flammia, N. C. Menicucci, and O. Pfister, The optical frequency comb as a one-way quantum computer, *J. Phys. B: At., Mol. Opt. Phys.* **42**, 114009 (2009).
- [4] N. C. Menicucci, Temporal-mode continuous-variable cluster states using linear optics, *Phys. Rev. A* **83**, 062314 (2011).
- [5] P. Wang, M. Chen, N. C. Menicucci, and O. Pfister, Weaving quantum optical frequency combs into continuous-variable hypercubic cluster states, *Phys. Rev. A* **90**, 032325 (2014).
- [6] R. N. Alexander, P. Wang, N. Sridhar, M. Chen, O. Pfister, and N. C. Menicucci, One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator, *Phys. Rev. A* **94**, 032327 (2016).
- [7] R. N. Alexander, S. Yokoyama, A. Furusawa, and N. C. Menicucci, Universal quantum computation with temporal-mode bilayer square lattices, *Phys. Rev. A* **97**, 032302 (2018).
- [8] B.-H. Wu, R. N. Alexander, S. Liu, and Z. Zhang, Quantum computing with multidimensional continuous-variable cluster states in a scalable photonic platform, *Phys. Rev. Res.* **2**, 023138 (2020).
- [9] R. Yang, J. Zhang, I. Klich, C. González-Arciniegas, and O. Pfister, Spatiotemporal graph states from a single optical parametric oscillator, *Phys. Rev. A* **101**, 043832 (2020).
- [10] M. Chen, N. C. Menicucci, and O. Pfister, Experimental realization of multipartite entanglement of 60 modes of a quantum optical frequency comb, *Phys. Rev. Lett.* **112**, 120505 (2014).
- [11] S. Yokoyama, R. Ukai, S. C. Armstrong, C. Sornphiphatphong, T. Kaji, S. Suzuki, J.-i. Yoshikawa, H. Yonezawa, N. C. Menicucci, and A. Furusawa, Ultra-large-scale continuous-variable cluster states multiplexed in the time domain, *Nat. Photonics* **7**, 982 (2013).
- [12] J.-i. Yoshikawa, S. Yokoyama, T. Kaji, C. Sornphiphatphong, Y. Shiozawa, K. Makino, and A. Furusawa, Generation of one-million-mode continuous-variable cluster state by unlimited time-domain multiplexing, *APL Photonics* **1**, 060801 (2016).
- [13] W. Asavanant, Y. Shiozawa, S. Yokoyama, B. Charoensombutamon, H. Emura, R. N. Alexander, S. Takeda, J.-i. Yoshikawa, N. C. Menicucci, H. Yonezawa, *et al.*, Generation of time-domain-multiplexed two-dimensional cluster state, *Science* **366**, 373 (2019).
- [14] M. V. Larsen, X. Guo, C. R. Breum, J. S. Neergaard-Nielsen, and U. L. Andersen, Deterministic generation of a two-dimensional cluster state, *Science* **366**, 369 (2019).
- [15] P. van Loock, Examples of Gaussian cluster computation, *J. Opt. Soc. Am. B* **24**, 340 (2007).
- [16] W. Asavanant, B. Charoensombutamon, S. Yokoyama, T. Ebihara, T. Nakamura, R. N. Alexander, M. Endo, J.-i. Yoshikawa, N. C. Menicucci, H. Yonezawa, and A. Furusawa, Time-domain-multiplexed measurement-based quantum operations with 25-MHz clock frequency, *Phys. Rev. Appl.* **16**, 034005 (2021).
- [17] M. V. Larsen, X. Guo, C. R. Breum, J. S. Neergaard-Nielsen, and U. L. Andersen, Deterministic multi-mode gates on a scalable photonic quantum computing platform, *Nat. Phys.* **17**, 1018 (2021).
- [18] R. N. Alexander, S. C. Armstrong, R. Ukai, and N. C. Menicucci, Noise analysis of single-mode Gaussian operations using continuous-variable cluster states, *Phys. Rev. A* **90**, 062324 (2014).
- [19] B. W. Walshe, L. J. Mensen, B. Q. Baragiola, and N. C. Menicucci, Robust fault tolerance for continuous-variable cluster states with excess antisqueezing, *Phys. Rev. A* **100**, 010301(R) (2019).
- [20] M. Ohliger, K. Kieling, and J. Eisert, Limitations of quantum computing with Gaussian cluster states, *Phys. Rev. A* **82**, 042336 (2010).
- [21] D. Gottesman, A. Kitaev, and J. Preskill, Encoding a qubit in an oscillator, *Phys. Rev. A* **64**, 012310 (2001).
- [22] G. Pantaleoni, B. Q. Baragiola, and N. C. Menicucci, Hidden qubit cluster states, *Phys. Rev. A* **104**, 012431 (2021).
- [23] B. Q. Baragiola, G. Pantaleoni, R. N. Alexander, A. Karanjai, and N. C. Menicucci, All-Gaussian universality and fault tolerance with the Gottesman-Kitaev-Preskill code, *Phys. Rev. Lett.* **123**, 200502 (2019).
- [24] K. Noh, V. V. Albert, and L. Jiang, Quantum capacity bounds of Gaussian thermal loss channels and achievable rates with Gottesman-Kitaev-Preskill codes, *IEEE Trans. Inf. Theory* **65**, 2563 (2019).
- [25] J. Wu and Q. Zhuang, Continuous-variable error correction for general Gaussian noises, *Phys. Rev. Appl.* **15**, 034073 (2021).
- [26] N. C. Menicucci, Fault-tolerant measurement-based quantum computing with continuous-variable cluster states, *Phys. Rev. Lett.* **112**, 120504 (2014).
- [27] K. Fukui, A. Tomita, A. Okamoto, and K. Fujii, High-threshold fault-tolerant quantum computation with analog quantum error correction, *Phys. Rev. X* **8**, 021054 (2018).
- [28] M. V. Larsen, C. Chamberland, K. Noh, J. S. Neergaard-Nielsen, and U. L. Andersen, Fault-tolerant continuous-variable measurement-based quantum computation architecture, *PRX Quantum* **2**, 030325 (2021).
- [29] J. E. Bourassa, R. N. Alexander, M. Vasmer, A. Patil, I. Tzitrin, T. Matsuura, D. Su, B. Q. Baragiola, S. Guha, G. Dauphinais, K. K. Sabapathy, N. C. Menicucci, and I. Dhand, Blueprint for a scalable photonic fault-tolerant quantum computer, *Quantum* **5**, 392 (2021).
- [30] I. Tzitrin, T. Matsuura, R. N. Alexander, G. Dauphinais, J. E. Bourassa, K. K. Sabapathy, N. C. Menicucci, and I. Dhand, Fault-tolerant quantum computation with static linear optics, *PRX Quantum* **2**, 040353 (2021).
- [31] M. V. Larsen, J. S. Neergaard-Nielsen, and U. L. Andersen, Architecture and noise analysis of continuous-variable quantum gates using two-dimensional cluster states, *Phys. Rev. A* **102**, 042608 (2020).
- [32] B. W. Walshe, R. N. Alexander, N. C. Menicucci, and B. Q. Baragiola, Streamlined quantum computing with macronode cluster states, *Phys. Rev. A* **104**, 062427 (2021).
- [33] R. N. Alexander and N. C. Menicucci, Flexible quantum circuits using scalable continuous-variable cluster states, *Phys. Rev. A* **93**, 062326 (2016).

- [34] B. W. Walshe, B. Q. Baragiola, R. N. Alexander, and N. C. Menicucci, Continuous-variable gate teleportation and bosonic-code error correction, *Phys. Rev. A* **102**, 062411 (2020).
- [35] K. Duivenvoorden, B. M. Terhal, and D. Weigand, Single-mode displacement sensor, *Phys. Rev. A* **95**, 012305 (2017).
- [36] T. J. Volkoff and Y. Subaşı, Ancilla-free continuous-variable SWAP test, *Quantum* **6**, 800 (2022).
- [37] N. C. Menicucci, S. T. Flammia, and P. van Loock, Graphical calculus for Gaussian pure states, *Phys. Rev. A* **83**, 042335 (2011).
- [38] A. L. Grimsmo, J. Combes, and B. Q. Baragiola, Quantum computing with rotation-symmetric bosonic codes, *Phys. Rev. X* **10**, 011058 (2020).
- [39] C. Colbourne and J. Dinitz, *Handbook of Combinatorial Designs* (CRC Press, Boca Raton, FL, 2007)
- [40] K. J. Horadam, *Hadamard Matrices and Their Applications* (Princeton University Press, Princeton, 2007).
- [41] M. Hirasaka, K. Kim, and H. Yu, Isomorphism classes of association schemes induced by Hadamard matrices, *Eur. J. Combin.* **51**, 37 (2016).