Flexible quantum circuits using scalable continuous-variable cluster states

Rafael N. Alexander[*] and Nicolas C. Menicucci[†]

School of Physics, The University of Sydney, Sydney, New South Wales 2006, Australia
and School of Science, RMIT University, Melbourne, Victoria 3001, Australia

We show that measurement-based quantum computation on scalable continuous-variable (CV) cluster states admits more quantum-circuit flexibility and compactness than similar protocols for standard square-lattice CV cluster states. This advantage is a direct result of the macronode structure of these states—that is, a lattice structure in which each graph node actually consists of several physical modes. These extra modes provide additional measurement degrees of freedom at each graph location, which can be used to manipulate the flow and processing of quantum information more robustly and with additional flexibility that is not available on an ordinary lattice.

I. INTRODUCTION

Quantum information processing using measurement-based quantum computing (MBQC) [1] is divided into two steps: (1) preparation of a universal, highly entangled resource state (the standard choice is a *cluster state* with a square-lattice graph [2]), followed by (2) a sequence of single-site projective measurements with feedforward.

The last 15 years have seen the emergence of numerous extensions, improvements, and generalizations of this basic model. Important for this work is its generalization from cluster states made of qubits to those made of continuous-variable (CV) quantum systems [3]. Unlike their photonic-qubit counterparts [4,5], optical CV cluster states can be generated both deterministically and on a large scale with minimal experimental equipment. They need only offline squeezing and linear optics [6], all of which can be implemented using a single optical parametric oscillator (OPO) [7–10]. Extremely large cluster states of this type can be made with existing technology based on either frequency modes [11,12] or temporal modes [13,14].

Using CV cluster states for quantum computation comes with a price. Ideal states are infinitely squeezed [3,15]; thus noise is introduced into the computation due to the fact that only finite squeezing resources (and hence, finite energy) can be used in generating the state [15–17]. If left unchecked, this noise limits the length of computation possible using these states [18,19]. Nevertheless, it is still possible to achieve universal fault-tolerant quantum computation with CV cluster states [20] by employing known quantum-error-correction protocols [21], provided that the experimentally achievable squeezing levels are high enough. The current best-recorded squeezing level in an optical setup is 12.7 dB of squeezing [22], while the lowest theoretical upper bound on the required squeezing for fault-tolerant quantum computing is 20.5 dB [20].

Closing this squeezing gap in scalable CV cluster state implementations is of paramount importance for their use in large-scale, fault-tolerant quantum computation. A significant step in this direction is the development of resource-customized measurement-based protocols that capitalize on the available squeezing in order to minimize the noise per gate [10,17,23].

In the same vein, here we give a measurement protocol that is customized for a type of universal CV cluster state that is particularly scalable, known as the *quad-rail lattice* (QRL) [12,14]. The generation procedure of the QRL is particularly simple owing to the fact that its graph [12,14] is self-inverse and bipartite [8,9]. Indeed, it needs only two-mode squeezed states (TMSSs) and a single four-port linear-optics gate (known as a foursplitter) as building blocks [12,14]. This state's graph contains within it a square-lattice topology (making it universal) with respect to four-mode lattice sites known as *macronodes*. Our protocol leverages extra degrees of freedom present in each macronode, resulting in improved circuit compactness and flexibility. This work extends the macronode protocol presented in Ref. [17], which applies to the one-dimensional (1D) resource state known as the *CV dual-rail wire* [11–14,17].

The structure of this article is as follows: In Sec. II we review some basics of Gaussian pure states and the QRL [12,14]. In Sec. III we introduce the basic components of our measurement protocol, including encoding, unitary gates, and measurement readout. In Sec. IV we describe how these elements can be composed, allowing for flexible design of quantum circuits. In Sec. V we compare this protocol to previous work. We conclude in Sec. VI.

II. BACKGROUND

Throughout this article, we adopt the following conventions for all modes: $\hat{q} = \frac{1}{\sqrt{2}}(\hat{a} + \hat{a}^\dagger)$, $\hat{p} = \frac{1}{i\sqrt{2}}(\hat{a} - \hat{a}^\dagger)$. Using $[\hat{a},\hat{a}^\dagger] = 1$, this implies that $[\hat{q},\hat{p}] = i$ with $\hbar = 1$.

A. Symplectic formalism and gate definitions

The Heisenberg action of an N-mode Gaussian unitary $\hat{U}$ acting on the vector of Heisenberg-picture operators $\hat{\mathbf{x}} = \binom{\hat{\mathbf{q}}}{\hat{\mathbf{p}}}$ can be written as

$$\hat{U}^\dagger \hat{\mathbf{x}} \hat{U} = \mathbf{S}_{\hat{U}} \hat{\mathbf{x}}, \qquad (2.1)$$

_____
[*]rafael.alexander@sydney.edu.au
[†]ncmenicucci@gmail.com

where we have ignored displacements and

$$\mathbf{S}_{\hat{U}} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \quad (2.2)$$

is a $2N \times 2N$ real, symplectic matrix. Some useful examples are given below.

The phase-delay gate is defined to be

$$\hat{R}(\theta) := \exp(i\theta\hat{a}^\dagger\hat{a})$$
$$= \exp\left[\frac{i\theta}{2}(\hat{q}^2 + \hat{p}^2 - 1)\right]. \quad (2.3)$$

Its Heisenberg action on $\hat{\mathbf{x}} = (\hat{q}, \hat{p})^T$ is given by the symplectic matrix

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \quad (2.4)$$

Note that $\hat{R}(-\omega\,\delta t)$ implements *forward* time evolution for an oscillator with frequency ω over a small time interval $\delta t > 0$. Thus, for *positive* θ, the gate $\hat{R}(\theta)$ will *delay* the oscillator by a time interval θ/ω. This motivates our choice of terminology and sign convention for this gate.

In the Schrödinger picture, a phase delay by θ [i.e., $\hat{R}(\theta)$] rotates the state's Wigner function *counterclockwise* by an angle θ. Viewed instead from the Heisenberg picture, this operation rotates the vector $\hat{\mathbf{x}}$ of quadrature operators in the same fashion—i.e., counterclockwise by θ.

The single-mode squeezing gate we use has the following (nonstandard) definition:

$$\hat{S}(s) := \hat{R}(\operatorname{Im}\ln s)\exp\left[-\frac{1}{2}(\operatorname{Re}\ln s)(\hat{a}^2 - \hat{a}^{\dagger 2})\right]$$
$$= \hat{R}(\operatorname{Im}\ln s)\exp\left[-\frac{i}{2}(\operatorname{Re}\ln s)(\hat{q}\hat{p} + \hat{p}\hat{q})\right], \quad (2.5)$$

where $s \in \mathbb{R}\backslash\{0\}$ is called the *squeezing factor*. This is related to the more commonly used *squeezing parameter r* through

$$|s| = e^r. \quad (2.6)$$

This gate differs from the ordinary squeezing gate only by an additional π phase delay when $s < 0$. Its Heisenberg action on $\hat{\mathbf{x}}$ is given by the symplectic matrix

$$\mathbf{S}(s) = \begin{pmatrix} s & 0 \\ 0 & s^{-1} \end{pmatrix}. \quad (2.7)$$

In the Heisenberg picture, this evolution multiplies the $\hat{q}$ quadrature by s and the $\hat{p}$ quadrature by s^{-1}. We define this to be what is meant by "squeezing by a factor of s." (In addition to the π phase delay when $s < 0$, this operation antisqueezes $\hat{q}$ and squeezes $\hat{p}$ when $|s| > 1$, and vice versa if $|s| < 1$.)

The beamsplitter gate is defined to be

$$\hat{B}_{ij}(\theta) := \exp[-\theta(\hat{a}_i^\dagger\hat{a}_j - \hat{a}_j^\dagger\hat{a}_i)]$$
$$= \exp[-i\theta(\hat{q}_i\hat{p}_j - \hat{q}_j\hat{p}_i)], \quad (2.8)$$

where $\sin\theta$ is the reflectivity of the beamsplitter. Its Heisenberg action on $\hat{\mathbf{x}} = (\hat{q}_i, \hat{q}_j, \hat{p}_i, \hat{p}_j)^T$ is given by

$$\mathbf{B}_{ij}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{pmatrix}. \quad (2.9)$$

While this beamsplitter is often defined in the literature with additional phase delays incorporated (in order to match the physics more closely), the definition here matches that in Refs. [14,17,24] and is more suitable for analysis of CV quantum-computing applications.

A useful property of this gate is that up to displacements, it commutes with the action of the same single-mode Gaussian unitary gate on two modes, i.e.,

$$[\hat{B}_{ij}(\theta), \hat{U}_i\hat{U}_j] = 0, \quad (2.10)$$

where $\hat{U}$ is a single-mode Gaussian unitary gate without displacements.

Proof of Equation (2.10). It suffices to check that their symplectic matrix representations commute. Denote the symplectic matrix representation of $\hat{U}$ by $\mathbf{U}$. Then the symplectic matrix representation of $\hat{U}_i\hat{U}_j$ can be represented as $\mathbf{U} \otimes \mathbf{I}$, where $\otimes$ is a Kronecker product and $\mathbf{I}$ is the 2×2 identity matrix. Note similarly that $\mathbf{B}(\theta) = \mathbf{I} \otimes \mathbf{R}(\theta)$. Clearly, these matrices commute. ∎

The 50:50 beamsplitter gate is defined as

$$\hat{B}_{ij} := \hat{B}_{ij}\left(\frac{\pi}{4}\right), \quad (2.11)$$

i.e., it is a special case of the above-defined beamsplitter where $\theta = \frac{\pi}{4}$, and the dependence on the angle is dropped for notational convenience. Note that $\hat{B}_{ij}^\dagger = \hat{B}_{ji}$.

Finally, the foursplitter gate is defined to be

$$\hat{A}_{jklm} := \exp\left[\frac{\pi}{4}\left((\hat{a}_k^\dagger + \hat{a}_l^\dagger)(\hat{a}_j - \hat{a}_m) - \text{H.c.}\right)\right]$$
$$= \exp\left[-i\frac{\pi}{4}\left((\hat{q}_j - \hat{q}_m)(\hat{p}_k + \hat{p}_l)\right.\right.$$
$$\left.\left. + (\hat{q}_k + \hat{q}_l)(\hat{p}_m - \hat{p}_j)\right)\right], \quad (2.12)$$

where "H.c." abbreviates the Hermitian conjugate (†) of the first term in the exponent. Its Heisenberg action on $\hat{\mathbf{x}} = (\hat{q}_i, \hat{q}_j, \hat{q}_k, \hat{q}_l, \hat{p}_i, \hat{p}_j, \hat{p}_k, \hat{p}_l)^T$ is given by

$$\mathbf{A}_{ijkl} := \begin{pmatrix} \tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}} \end{pmatrix}, \quad (2.13)$$

where $\mathbf{0}$ denotes the 4×4 matrix of zeros and

$$\tilde{\mathbf{A}} = \frac{1}{2}\begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (2.14)$$

This gate admits the following convenient decompositions into four 50:50 beamsplitters [12,14]:

$$\hat{A}_{ijkl} = \hat{B}_{ij}\hat{B}_{kl}\hat{B}_{ik}\hat{B}_{jl} = \hat{B}_{ik}\hat{B}_{jl}\hat{B}_{ij}\hat{B}_{kl}. \quad (2.15)$$

B. Graphical calculus for Gaussian pure states

In this article, we will be describing the properties of a Gaussian pure state (the QRL). For convenience, we will represent this state by its graph [24], which is defined using the graphical calculus for Gaussian pure states, summarized below.

1. Graphs

Given an undirected, complex-weighted graph on N nodes with adjacency matrix $\mathbf{Z} (= \mathbf{Z}^{\mathrm{T}})$ and $\mathrm{Im}\,\mathbf{Z} > 0$ [24], $\mathbf{Z}$ uniquely defines the position-space wave function

$$\psi_{\mathbf{Z}}(\mathbf{q}) := \frac{(\det \mathrm{Im}\,\mathbf{Z})^{1/4}}{\pi^{N/4}} \exp\left[\frac{i}{2}\mathbf{q}^{\mathrm{T}}\mathbf{Z}\mathbf{q}\right] \qquad (2.16)$$

of the N-mode Gaussian pure state $|\psi_{\mathbf{Z}}\rangle$, where $\mathbf{q}$ is a column vector of c-numbers. It also gives a compact description of the nullifiers of $|\psi_{\mathbf{Z}}\rangle$:

$$(\hat{\mathbf{p}} - \mathbf{Z}\hat{\mathbf{q}})|\psi_{\mathbf{Z}}\rangle = \mathbf{0}, \qquad (2.17)$$

where $\hat{\mathbf{q}} = (\hat{q}_1, \dots \hat{q}_N)^{\mathrm{T}}$ and $\hat{\mathbf{p}} = (\hat{p}_1, \dots \hat{p}_N)^{\mathrm{T}}$ are column vectors of operators. Every Gaussian pure state uniquely defines (up to phase-space displacements and overall phase) an associated graph $\mathbf{Z}$ [24].

2. Graph update rule

In the language of the graphical calculus, Schrödinger-picture evolution of a Gaussian unitary $\hat{U}$ can be represented up to displacements and overall phase by a graph update rule

$$\hat{U}|\psi_{\mathbf{Z}}\rangle = |\psi_{\mathbf{Z}'}\rangle \qquad (2.18)$$

with

$$\mathbf{Z}' = (\mathbf{C} + \mathbf{D}\mathbf{Z})(\mathbf{A} + \mathbf{B}\mathbf{Z})^{-1}, \qquad (2.19)$$

where the submatrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are defined via the Heisenberg action of $\hat{U}$, as in Eq. (2.2).

3. Simplified graphs

The Gaussian pure states that we consider in this article are specified by few graphical parameters, i.e., edge and self-loop weights in $\mathbf{Z}$. When representing such states by their corresponding graph, it is convenient to use a simplified set of rules known as the *simplified graphical calculus* [14]. It makes use of the following conventions: no self-loops are drawn, and the color of an edge indicates the sign of its edge weight [see Figs. 1(a) and 1(b)]. In addition to these (standard) conventions, we will use differently colored nodes—green instead of black—to denote the inclusion of an input state localized to a single graph node, as shown in Fig. 1(c). The self-loops (not shown) on all noninput graph nodes have weight

$$i\varepsilon := i \,\mathrm{sech}\, 2r, \qquad (2.20)$$

where the squeezing parameter r gives the amount of vacuum squeezing used in preparing the state [12,14]. All edge weights between different nodes are

$$\pm\mathcal{C}t := \pm\mathcal{C} \tanh 2r, \qquad (2.21)$$

which is the product of the *edge-weight coefficient* $\mathcal{C}$ (specified on each figure) and a squeezing-dependent factor t, along with
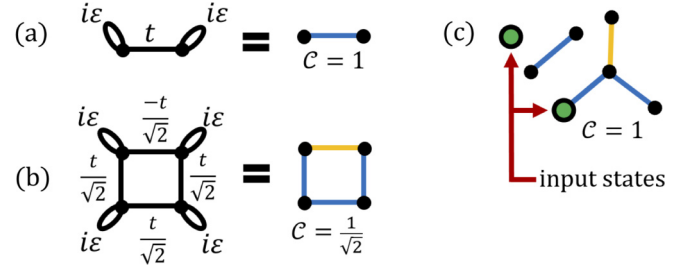

FIG. 1. (a) Two-mode continuous-variable cluster state represented using the full graphical calculus [24] (left) and the simplified graphical calculus [14] (right). Edge weights ε and t are defined in Eq. (2.20) and Eq. (2.21), respectively. (b) We similarly represent a four-mode square CV cluster state. (c) Seven-mode state containing two inputs (green nodes)—one is disconnected (tensor product with the rest of the state), and the other is attached to a three-mode Gaussian pure state.

a sign $\pm$ denoted by blue or yellow, respectively. Note that $\varepsilon \to 0$ and $t \to 1$ as the squeezing parameter $r \to \infty$, which corresponds to the high-squeezing limit.

Although our use of the graphical calculus strictly applies only when the input states (green nodes) are themselves Gaussian pure states, this choice is purely for representational convenience. All results presented here hold for general input states, including non-Gaussian and/or mixed states.

C. The quad-rail lattice

The QRL can be generated from a collection of two-mode cluster states [defined in Fig. 1(a)][1] arranged along edges of a square lattice by applying a foursplitter gate [Eq. (2.12)] to each four-mode lattice site, a.k.a. a *macronode* [12,14]. The resulting QRL is defined by its four-layered square-lattice graph, as shown in Fig. 2(b). Further details about the generation of this state can be found in Refs. [14] and [12].

The QRL is universal for MBQC. To see this, consider measuring the top three layers of modes in $\hat{q}$ basis. Note that such measurements can be implemented experimentally via homodyne detection [12,14]. Graphically, this action is represented by node deletion [24], resulting in a square-lattice CV cluster state as shown in Fig. 2(c). Up to displacements, this is the canonical resource state for universal MBQC with CVs [3,15]. Unwanted displacements (due to $\hat{q}$ measurements on the top three layers) can be straightforwardly taken into account in the measurement protocol by feedforward.

Achieving universal quantum computation this way is not optimal, however, because projecting down to a canonical CV cluster state results in an ordinary lattice with $\mathcal{C} = \frac{1}{4}$ (instead of $\mathcal{C} = 1$), which introduces excessive noise when used in a computation [17]. We use the remainder of this article to introduce a different—and much more favorable—MBQC protocol that runs directly on the full QRL [Fig. 2(b)].

---

[1]Equivalently, two-mode squeezed states can be used by incorporating a $\frac{\pi}{4}$ phase delay into the measurement of all nodes [25].
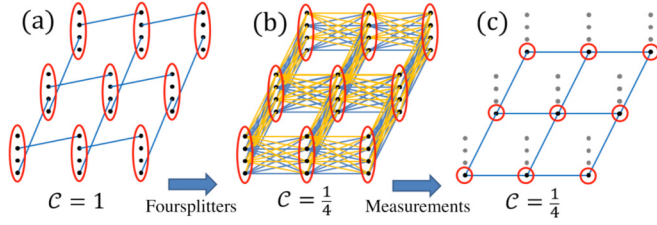
FIG. 2. Simplified graphical-calculus representation [14] of the construction of the quad-rail lattice and conversion to an ordinary continuous-variable cluster state. (a) A collection of two-mode continuous-variable cluster states. These pairs are "stitched" together by a foursplitter gate [Eq. (2.12)] at each macronode (indicated by the red ovals) in order to construct the quad-rail lattice. (b) This graph defines the quad-rail lattice state. (c) Measuring the top three layers (faded) of the quad-rail lattice in the $\hat{q}$ basis produces a square-lattice continuous-variable cluster state as shown. Note that each site (red circle) contains only one mode. In the original proposal [14], universal quantum computation proceeded via the standard measurement-based protocol [15]. Removing the extra nodes and links from (b), however, wastes squeezing resources [17]. Instead, our proposal directly employs the state shown in (b), making more efficient use of the available resources (the advantages are discussed in Sec. V).

III. USING MACRONODES FOR MBQC

The basic idea for our protocol is that quantum computation can proceed via measurements on the QRL directly (rather than first reducing to the square-lattice cluster state). We break this section into five parts: encoding (Sec. III A), measurements (Sec. III B), single-mode gates (Sec. III C), two-mode gates (Sec. III D), and measurement readout (Sec. III E).

A. Encoding

In MBQC, once the resource state is prepared, the only allowable operations are local measurements. In our protocol, local measurements implement logic gates on *macrolocally* encoded input states. This means that input states are localized with respect to a particular macronode, but they are distributed *nonlocally* between the four physical modes that make it up. (The reason for this will become evident once we present our protocol.)

Each macronode admits two natural tensor-product decompositions. The first is the usual one defined in terms of the *physical modes* (P). The second—which is more useful for our purposes—is to define four *distributed modes* (D) as balanced linear combinations of the physical modes. Specifically, in the Heisenberg picture,

$$\hat{\mathbf{a}}_{\mathrm{D}} := \mathbf{A}^{-1} \hat{\mathbf{a}}_{\mathrm{P}}, \qquad (3.1)$$

where $\hat{\mathbf{a}}_{\mathrm{P}} := (\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4)^{\mathrm{T}}$ and $\hat{\mathbf{a}}_{\mathrm{D}} := (\hat{a}_a, \hat{a}_b, \hat{a}_c, \hat{a}_d)^{\mathrm{T}}$. Note that numerical (alphabetical) subscripts are used for the physical (distributed) modes.

The mapping in Eq. (3.1) is exactly the inverse of a foursplitter gate [Eq. (2.12)]. Figure 3 displays the QRL with respect to the physical modes (a) and with respect to the distributed modes (b). Notice that the former has fencelike connections between adjacent macronodes, while the latter consists merely of disjoint pairs. Also notice that the graphs
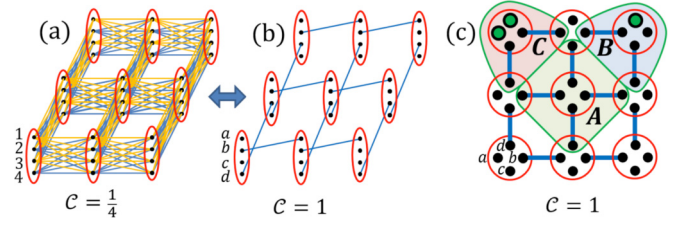

FIG. 3. (a, b) Two equivalent ways to represent the quad-rail lattice using the simplified graphical calculus [14]. The left graph represents the state using the physical-mode decomposition of each macronode, while the right graph represents the exact same state using the distributed-mode decomposition, with Eq. (3.1) connecting the two. Red ellipses indicate the macronodes (four-mode subsystems) that are left invariant by the change of mode decomposition. (c) Birds-eye view of the quad-rail lattice with respect to distributed modes with mode label conventions shown in the bottom left macronode. We include three input states and highlight three examples of input configurations within a macronode. In A, we have a "blank" macronode that contains no input states. In B and C, respectively, one and two of the two-mode cluster states have been replaced with an input state.

in Figs. 2(a) and 3(b) are visually identical. Nevertheless, they represent different physical states because they are defined with respect to different mode decompositions (physical and distributed, respectively).

For the rest of this article, we will use distributed modes exclusively because this allows for the simplest description of information propagation through the QRL. We allow input states to occupy any of the four possible distributed modes (a,b,c,d) within a macronode. Unless otherwise specified, we assume that a maximum of two of the distributed modes within a given macronode are occupied by an input state. This guarantees that there is at least one two-mode cluster state per input that connects to an adjacent macronode. This condition is required in order to implement unitary gates (otherwise the output has no place to go). Three examples of input-state configurations are given in Fig. 3(c).

B. Macronode measurements

Our protocol implements Gaussian unitary gates on encoded input states by locally measuring the physical modes that make up each macronode in a rotated quadrature basis $\hat{p}(\theta) := \hat{p}\cos\theta - \hat{q}\sin\theta$. We vectorize the measurement bases for a given macronode measurement using

$$\hat{\mathbf{p}}_{\mathrm{P}}(\boldsymbol{\theta}) := \left(\hat{p}_1(\theta_1), \hat{p}_2(\theta_2), \hat{p}_3(\theta_3), \hat{p}_4(\theta_4) \right)^{\mathrm{T}}, \qquad (3.2)$$

where $\boldsymbol{\theta} := (\theta_1, \theta_2, \theta_3, \theta_4)$. Note that local measurements with respect to the physical modes will generally correspond to nonlocal (four-body) measurements with respect to the distributed modes (and the inputs).

To characterize the effective logic gate implemented by macronode measurement, we consider the two-input case [as in C in Fig. 3(c)]. This case is the most general, as the no- and single-input cases are special cases with both or one of the inputs replaced by half of a two-mode CV cluster state.

There are $\binom{4}{2} = 6$ different two-input macronode configurations (as shown in Fig. 4) and thus 12 total input configurations
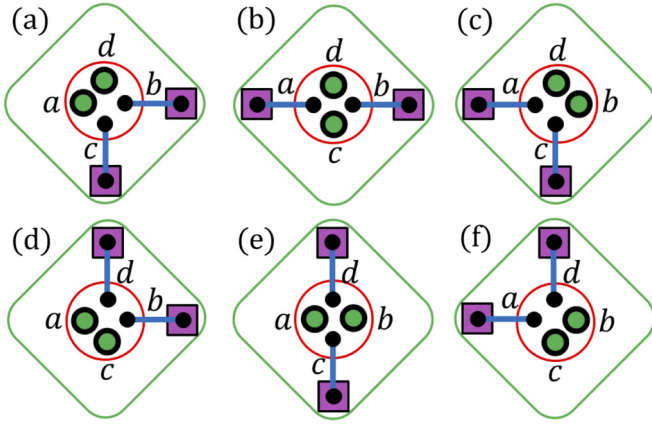
FIG. 4. Birds-eye view of six macronode configurations with two input states and two adjacent output modes (inside the purple boxes). We assign distributed-mode labels $\{a,b,c,d\}$ consistent with Fig. 3. Each input can be assigned to one of the two green modes, and each can also be mapped to either output, resulting in a total of 24 distinct processes. We omit macronode labels and labels on the output modes.

with distinct input states. In addition, each input must be paired with a two-mode cluster state that contains the corresponding output mode. There are two possibilities, resulting in 24 distinct input-to-output mode configurations. It suffices to characterize the single case shown in Fig. 5 because all other configurations are related to this by applying a permutation on the distributed modes prior to measurement, and this can be taken into account by a simple change to the homodyne angles.

To see this, define a generic permutation gate via its four-mode symplectic matrix representation,

$$\boldsymbol{\sigma} = \begin{pmatrix} \tilde{\boldsymbol{\sigma}} & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{\sigma}} \end{pmatrix}, \tag{3.3}$$

where $\tilde{\boldsymbol{\sigma}}$ is some 4×4 permutation matrix (a single 1 entry in each row and column and all other entries 0). It is sufficient to check the commutation properties of the foursplitter gate
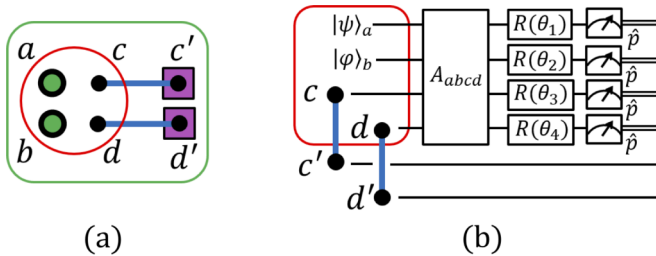

(a)   (b)

FIG. 5. (a) A specific case of a single macronode with two input states and two adjacent output modes equivalent to Fig. 4(e). The output mode for a is c' and the output mode for b is d'. (b) Quantum circuit for macronode measurement of a general two-input macronode for arbitrary inputs $|\psi\rangle$ and $|\phi\rangle$ encoded within distributed modes a and b, respectively. Locally measuring the physical modes is exactly equivalent to first applying a foursplitter gate on the distributed modes and then doing the desired measurements. Up to measurement-dependent displacements and finite-squeezing effects, the output state is given in Eq. (3.5).

with each element of any generating set of all four-mode permutation gates. Let $\boldsymbol{\sigma}_{jk}$ denote the permutation gate that swaps modes j and k. Then we have that

$$\mathbf{A}^{-1}\boldsymbol{\sigma}_{1,2}\mathbf{A} = \boldsymbol{\sigma}_{2,4},$$
$$\mathbf{A}^{-1}\boldsymbol{\sigma}_{1,3}\mathbf{A} = \boldsymbol{\sigma}_{3,4}, \tag{3.4}$$
$$\mathbf{A}^{-1}\boldsymbol{\sigma}_{1,4}\mathbf{A} = \boldsymbol{\sigma}_{2,3}\mathbf{R}_2(\pi)\mathbf{R}_3(\pi),$$

where $\mathbf{R}(\pi)$ is defined in Eq. (2.3). Thus, by commuting through the foursplitter gate, each four-mode permutation gate $\boldsymbol{\sigma}$ is mapped to a combination of a new permutation gates and some single-mode π phase delays. These gates can be incorporated directly into the macronode measurements by permuting the choice of measurement angles (e.g., $\theta_i \leftrightarrow \theta_j$) and adding π phase delays (e.g., $\theta_i \mapsto \theta_i + \pi$).

For the case shown in Fig. 5 and neglecting measurement-dependent displacements and finite-squeezing effects (which are discussed in the proof below), the most general Gaussian unitary that can be applied on the two encoded input modes $|\psi\rangle$ and $|\varphi\rangle$ by measuring in $\hat{\mathbf{p}}_{\mathrm{P}}(\boldsymbol{\theta})$ is

$$|\psi\rangle_a \, |\varphi\rangle_b \mapsto \hat{G}_{c'd'}(\boldsymbol{\theta}) \, |\psi\rangle_{c'} \, |\varphi\rangle_{d'}, \tag{3.5}$$

where $\theta_1 \neq \theta_3$, $\theta_2 \neq \theta_4$, and

$$\hat{G}_{jk}(\boldsymbol{\theta}) := \hat{B}_{jk}^{\dagger} \hat{V}_j(\theta_1, \theta_3) \hat{V}_k(\theta_2, \theta_4) \hat{B}_{jk}. \tag{3.6}$$

Sandwiched between the pair of 50:50 beamsplitters is the single-mode unitary gate

$$\hat{V}_j(x,y) := \hat{R}_j\left(\frac{x+y}{2}\right)\hat{S}_j\left(\tan\left[\frac{x-y}{2}\right]\right)\hat{R}_j\left(\frac{x+y}{2}\right). \tag{3.7}$$

Notice that the output states automatically emerge at distributed modes (c', d') of adjacent macronodes.

Proof of Equation (3.5). We start with Fig. 6, which shows a macronode measurement circuit where the foursplitter is decomposed into four beamsplitters [using Eq. (2.15)]. To go from Figs. 6(b) to 6(c) we used an interferometric symmetry of the pair of two-mode cluster states on modes (c, c') and (d, d') derived in Appendix C of Ref. [10]: acting with $\hat{B}_{cd}$ on this state is equivalent to acting with $\hat{B}_{d'c'}$ instead [25].

Figure 6(c) shows that macronode measurement is equivalent to two copies of a gate teleportation circuit [10,17] conjugated by beamsplitters ($\hat{B}_{ab}$ and $\hat{B}_{d'c'}$). The gate teleportation circuits each implement

$$\hat{V}(r, m_j, m_k, \theta_j, \theta_k) := \hat{N}(r)\hat{D}(m_j, m_k, \theta_j, \theta_k)\hat{V}(\theta_j, \theta_k), \tag{3.8}$$

where j and k are 1 and 3 (2 and 4) for the top (bottom) subcircuit in Fig. 6(c), $\hat{V}$ is defined in Eq. (3.7), and

$$\hat{D}(m_j, m_k, \theta_j, \theta_k) = \hat{D}\left[\frac{-ie^{i\theta_k}m_j - ie^{i\theta_j}m_k}{\sin(\theta_j - \theta_k)}\right] \tag{3.9}$$

is a phase-space displacement [$\hat{D}(\alpha) = e^{\alpha\hat{a}^{\dagger} - \alpha^*\hat{a}}$] that depends on the homodyne angles and measurement outcomes m_j and m_k associated with measuring modes j and k. Finally,

$$\hat{N}(r) = e^{-\varepsilon\hat{q}^2/2}e^{-\varepsilon\hat{p}^2/2t^2}\hat{S}(t^{-1}) \tag{3.10}$$
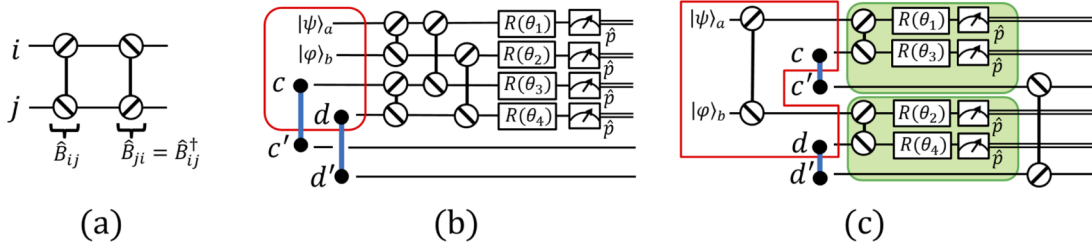
FIG. 6. (a) Two-mode circuit representation of 50:50 beamsplitter $\hat{B}_{ij}$ and $\hat{B}_{ji}$. The overall effect of this circuit on modes i and j is to implement the identity gate. (b) Macronode measurement as in Fig. 5 but with the foursplitter gate $\hat{A}_{abcd}$ replaced with four 50:50 beamsplitters, as in Eq. (2.15). (c) Restructured macronode measurement circuit equivalent to (b) (see text for details). The vertical ordering of the modes has been changed. Modes enclosed within the red box belong to the measured macronode. Note that the two subcircuits within the green regions are identical up to the choice of measurement angles. Each of these subcircuits can be individually interpreted as a CV teleportation protocol with generalized homodyne measurements [26]. Equivalently, they are each a single macronode measurement on the *CV dual-rail quantum wire*, discussed in Ref. [17].

is a nonunitary operator that applies the noise from finite squeezing to the state (after which the state must be renormalized) [17].

The macronode measurement maps

$$|\psi\rangle_a |\varphi\rangle_b \mapsto \hat{G}_{c'd'}(r,\mathbf{m},\boldsymbol{\theta}) |\psi\rangle_{c'} |\varphi\rangle_{d'}, \qquad (3.11)$$

where $\mathbf{m} = (m_1,m_2,m_3,m_4)$, $\boldsymbol{\theta} = (\theta_1,\theta_2,\theta_3,\theta_4)$, and

$$\hat{G}_{jk}(r,\mathbf{m},\boldsymbol{\theta})$$
$$:= \hat{B}_{kj} \hat{V}_j(r,m_1,m_3,\theta_1,\theta_3) \hat{V}_k(r,m_2,m_4,\theta_2,\theta_4) \hat{B}_{jk}. \quad (3.12)$$

In the limit of large squeezing and when all measurement outcomes are zero, we have

$$\hat{V}(\theta_j,\theta_k) = \lim_{r\to\infty} \hat{V}(r,0,0,\theta_j,\theta_k), \qquad (3.13)$$

and so

$$\hat{G}(\boldsymbol{\theta}) = \lim_{r\to\infty} \hat{G}(r,\mathbf{0},\boldsymbol{\theta}). \qquad (3.14)$$

In the more general case, the displacements can either be actively corrected at each step or merely accounted for using feedforward [15]. From this, Eq. (3.5) can be seen as the large squeezing limit of Eq. (3.11). In the rest of this article, we ignore displacements and finite-squeezing effects for simplicity of presentation. ∎

Note that for $\theta_1 = \theta_3$ or $\theta_2 = \theta_4$, Eq. (3.7) diverges in the squeezing factor and thus cannot represent a physical unitary operation. Nevertheless, the case where all four angles are equal ($\theta_1 = \theta_2 = \theta_3 = \theta_4$) will later be shown to correspond to measurement readout; see Sec. III E. Next we consider some examples of single- and two-mode Gaussian gates that are special cases of Eq. (3.6).

C. Single-mode Gaussian unitary gates

The first examples we consider are single-mode Gaussian unitary gates. Consider restricting the homodyne angles so that

$$\theta_2 = \theta_1 \quad \text{and} \quad \theta_4 = \theta_3. \qquad (3.15)$$

In this case, the single-mode gates sandwiched between the beamsplitters above in Eq. (3.6) are identical. Using Eq. (2.10), the beamsplitters cancel, resulting in

$$\hat{G}_{jk}(\boldsymbol{\theta})\Big|_{\substack{\theta_2=\theta_1\\\theta_4=\theta_3}} = \hat{V}_j(\theta_1,\theta_3)\hat{V}_k(\theta_1,\theta_3), \qquad (3.16)$$

which implements a pair of single-mode gates on the input states. As the same gate gets implemented on both inputs, a single macronode measurement does not allow for the two input states to evolve independently.

Independent single-mode gates can still be applied in the single-input case by ignoring the effect on the unused distributed mode. A single-mode $\hat{V}$ gate is sufficient to generate arbitrary single-mode Gaussian unitary gates up to displacements (and only two applications are required for all of them) [17].

Applying further restrictions so that $\theta_3 = \pm\theta_1$ implements a pair of phase delays and squeezers, respectively:

$$\hat{G}_{jk}(\boldsymbol{\theta})\Big|_{\theta_4=\theta_3=\theta_2=\theta_1} = \hat{R}_j(2\theta_1)\hat{R}_k(2\theta_1), \qquad (3.17)$$

and

$$\hat{G}_{jk}(\boldsymbol{\theta})\Big|_{\theta_4=\theta_3=-\theta_2=-\theta_1} = \hat{S}_j(\tan\theta_1)\hat{S}_k(\tan\theta_1). \qquad (3.18)$$

D. Two-mode Gaussian unitary gates

Here we provide different restrictions on the homodyne measurement angles $\boldsymbol{\theta}$ that yield interesting examples of two-mode gates from Eq. (3.6). Setting

$$\theta_3 = -\theta_1 \quad \text{and} \quad \theta_4 = -\theta_2 \qquad (3.19)$$

implements the two-mode-squeezing operation

$$\hat{G}_{jk}(\boldsymbol{\theta})\Big|_{\substack{\theta_3=-\theta_1\\\theta_4=-\theta_2}} = \hat{B}_{jk}^\dagger \hat{S}_j(\tan\theta_1)\hat{S}_k(\tan\theta_2)\hat{B}_{jk}. \qquad (3.20)$$

We can also implement a linear-optics gate by setting

$$\theta_3 = \theta_1 - \frac{\pi}{2} \quad \text{and} \quad \theta_4 = \theta_2 - \frac{\pi}{2}. \qquad (3.21)$$

This implements

$$\hat{G}_{jk}(\boldsymbol{\theta})\Big|_{\substack{\theta_3=\theta_1-\frac{\pi}{2}\\\theta_4=\theta_2-\frac{\pi}{2}}}$$

$$= \hat{B}_{jk}^\dagger \hat{R}_j\left(2\theta_1 + \frac{\pi}{2}\right)\hat{R}_k\left(2\theta_2 + \frac{\pi}{2}\right)\hat{B}_{jk}$$

$$= \hat{R}_j(\theta_+)\hat{R}_k(\theta_+)\left[\hat{R}_j\left(\frac{\pi}{2}\right)\hat{B}_{jk}(\theta_-)\hat{R}_k\left(\frac{\pi}{2}\right)\right], \quad (3.22)$$

where $\theta_\pm = \theta_1 \pm \theta_2$. Thus, up to some additional phase delays, the above gate implements a variable beamsplitter.
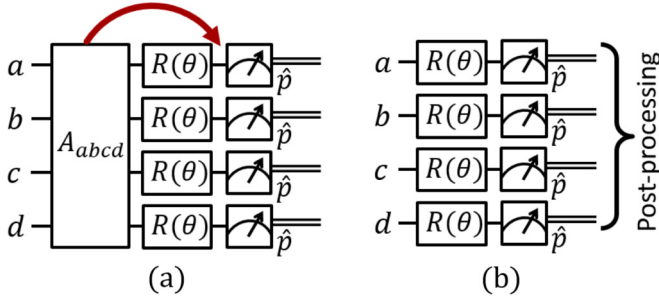
FIG. 7. (a) Here we show a macronode measurement circuit with respect to the distributed modes. In the special case of restricting measurement angles such as in Eq. (3.23), we can commute the phase delays past the foursplitter gate using Eq. (3.24), indicated by the red arrow. (b) Acting with a foursplitter gate immediately before a collection of $\hat{p}$ measurements is equivalent to only measuring in $\hat{p}$ and then classically taking linear combinations of the outcomes (postprocessing).

E. Measurement readout

In addition to implementing unitary gates, we must also be able to perform projective measurements on the encoded states. This can be implemented directly on the QRL, and we allow up to (all) four of the distributed modes to be filled with inputs. Each input that shares a macronode during measurement readout will be measured in the same homodyne basis. (This means that modes to be measured in different bases must be located within different macronodes.)

To measure each distributed mode within a single macronode in the homodyne basis $\hat{p}(\theta)$, one simply has to apply the following restriction on the measurement angles:

$$\theta = \theta_1 = \theta_2 = \theta_3 = \theta_4, \qquad (3.23)$$

as we now show. By decomposing the foursplitter gate $\hat{A}$ using Eq. (2.15) and applying the beamsplitter commutation relations in Eq. (2.10), it is straightforward to verify that

$$[\hat{A}_{ijkl}, \hat{R}_i(\theta)\hat{R}_j(\theta)\hat{R}_k(\theta)\hat{R}_l(\theta)] = 0. \qquad (3.24)$$

Thus, with these restricted measurements, the foursplitter in the standard macronode measurement circuit as shown in Fig. 7(a) can be commuted through the phase delays as shown.

Measuring $\hat{p}$ on all physical modes after the gate $\hat{A}$ is equivalent to just measuring the modes in $\hat{p}$ and taking linear combinations (given by $\tilde{\mathbf{A}}$) of the measurement outcomes:

$$\hat{A}^{\dagger}_{1,2,3,4} \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{pmatrix} \hat{A}_{1,2,3,4} = \tilde{\mathbf{A}} \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{pmatrix}. \qquad (3.25)$$

The physical foursplitter that is applied can be undone by classical postprocessing (applying $\tilde{\mathbf{A}}^{-1}$) on the actual measurement outcomes. Thus this macronode measurement can be implemented by measuring all of the distributed modes locally in the basis $\hat{p}(\theta)$, as shown in Fig. 7(b).

IV. CONSTRUCTING FLEXIBLE QUANTUM CIRCUITS

In the previous sections, we saw how input states can be encoded macrolocally [defined by Eq. (3.1)] and how
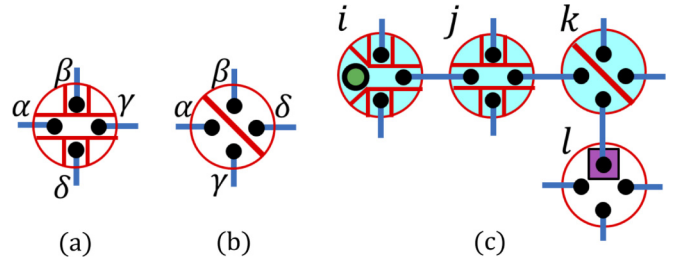

FIG. 8. (a, b) Rather than labeling the modes within each macronode to indicate how inputs are mapped onto outputs, we introduce additional red lines that partition each macronode such that the pairs of labels $\{\alpha, \gamma\}$ and $\{\beta, \delta\}$ share a partition. (a) and (b) above show two examples of this. (c) A connected sequence of macronodes $\{i, j, k, l\}$ on the quad-rail lattice. Embedding of a quantum wire within the quad-rail lattice. We use light-blue macronode coloring to indicate the use of the restricted measurements [as in Eq. (3.15)].

homodyne measurements on macronodes are sufficient to implement a variety of Gaussian unitary gates [of the form of Eq. (3.6)], as well as measurement readout. Now we briefly describe how connected regions of macronodes can be measured in order to implement quantum circuits.

We will start with how to construct quantum wires. In Sec. III C we showed that for a specific configuration of input and output modes $(a \mapsto c', b \mapsto d')$, restricting the measurement angles so that $\theta_1 = \theta_2$ and $\theta_3 = \theta_4$ ensures that the input states that share a macronode do not interact, i.e., only single-mode gates are applied. This result can be generalized for arbitrary input and output mode configurations by employing the permutation freedom discussed in Sec. III B. By appropriately modifying the homodyne angles, we can apply the same single-mode gates and teleport inputs at sites α and β to γ' and δ', respectively, for any valid assignment of $\{\alpha, \beta, \gamma, \delta\} \mapsto \{a, b, c, d\}$. We represent this graphically as shown in Figs. 8(a) and 8(b).

By restricting measurements like this on a connected sequence $(i, j, k, \dots, l)$ of macronodes on the QRL, (up to displacements and finite-squeezing effects) we can implement a single-mode Gaussian unitary $\hat{V}_l \cdots \hat{V}_k \hat{V}_j \hat{V}_i$ (omitting dependence on homodyne angles) on an input initially encoded within macronode i and have it propagate through the sequence of macronodes $(i, j, k, \dots, l)$, outputting into macronode l. We illustrate this by way of example in Fig. 8(c). These sequences thus act as embedded quantum wires, equivalent to the CV dual-rail wires described in Ref. [17].

Multiple wires can be embedded within the QRL provided that no two wires overlap on a lattice edge. Because we allow for up to two input states to share any macronode at a given time, these wires are free to intersect and cross one another. Note that when two wires meet at a macronode, the same single-mode Gaussian unitary gate gets applied to both inputs at that macronode.

Alternatively, the macronodes that act as junctions between two wires can be used to implement a two-mode Gaussian unitary, as discussed in Sec. III D. Therefore, wires and intersection sites can be used to implement single- and two-mode Gaussian unitary gates, respectively, and these components are sufficient to generate arbitrary multimode Gaussian unitaries. Measurement readout (homodyne detection) can
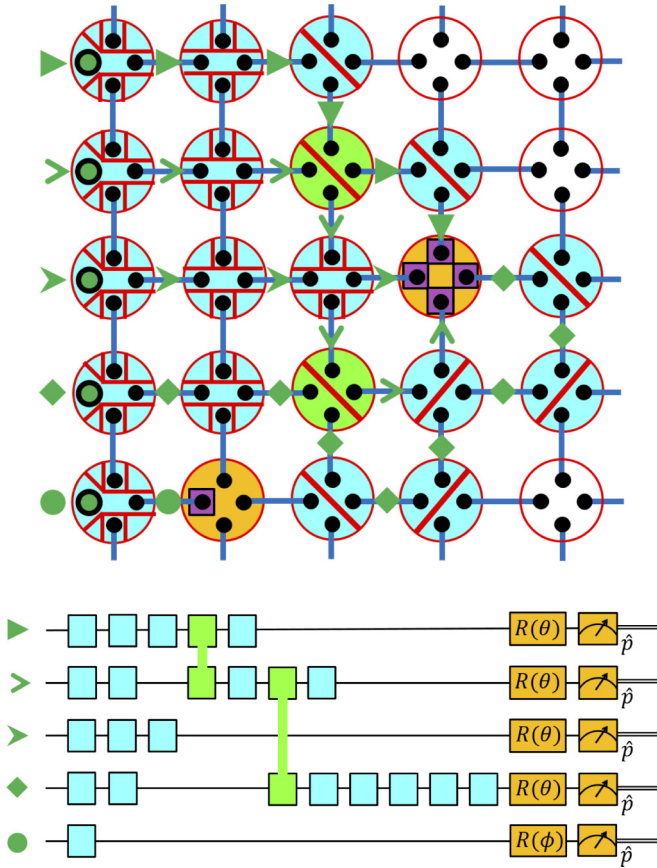
FIG. 9. (Above) An example measurement scheme on the quad-rail lattice. There are five encoded input states on the left-hand side, which we label by different arrow symbols. We label lattice edges to indicate how these inputs propagate along the lattice. Light-blue macronode coloring indicates application of single-mode Gaussian unitaries only [of the form Eq. (3.16)]. Green macronode coloring indicates the application of a two-mode Gaussian unitary, such as those described in Sec. III D. Orange macronode coloring is used to describe a measurement readout step on the lattice, as in Sec. III E. (Below) A quantum-circuit description of the overall Gaussian unitary implemented above. Light-blue small boxes are single-mode Gaussian unitary gates and connected green boxes are two-mode Gaussian unitary gates. Measurement operations are colored orange.

be implemented by connecting up to four wires to a given macronode and measuring it with restrictions as in Eq. (3.23).

By combining these results, we have a highly flexible means for implementing quantum circuits on the QRL. See Fig. 9 for an example. This is analogous to a field-programmable gate array (FPGA), since the QRL is a versatile resource that can be configured by the user at the "software level" into many different gate networks by the choice of measurement bases. With access to vacuum input states and arbitrary displacements, these operations are sufficient to implement arbitrary Gaussian computations.

Non-Gaussian resource

Gaussian operations alone are known not to be universal for quantum computing [27]. Full universality can be achieved, however, by diverting a subset of the QRL nodes to photon counters instead of homodyne detectors [14,15]. Depending on the particular practical implementation—which could even include encoded qubits and error correction [20]—it might be more favorable to periodically inject non-Gaussian resources [20,28] instead of counting photons. We leave further discussion of such elements to future work.

V. COMPARISON WITH PREVIOUS WORK

How does our scheme compare with other previously established CV cluster-state protocols? Below, we compare it with three alternatives, focusing on the following four features: (1) *circuit flexibility*, which is the maneuverability of the quantum wires; (2) *compactness*, which is the minimum number of sites that must be measured in order to implement a desired class of gates; (3) *noise per gate due to finite squeezing*; and (4) *scalability*.

Canonical CV cluster state. The original CV-measurement-based protocol introduced in Refs. [3] and [15] uses a single-rail $\mathcal{C} = 1$ square-lattice CV cluster state. Circuit flexibility is limited because the wires are generally constrained to run horizontally along the lattice, and two-mode gates can only be applied between nearest-neighbor wires. In general, single-mode Gaussian gates will require four steps along the lattice [17,29], thus limiting compactness as well. The natural two-mode gate is limited to the $\hat{C}_Z$ gate. Noise due to finite squeezing is known to depend on the edge weight ($\mathcal{C} = 1$) [17]. As such, the amount of noise per single-mode Gaussian unitary gate is roughly similar between this protocol and the QRL protocol introduced here. This resource state is theoretically convenient to analyze, which is why it is often used for initial studies [3,15,20], but it is less amenable to scalable design than macronode-based approaches (see Ref. [10] and references therein).

Projected quad-rail lattice. The original CV-measurement-based protocol can be modified to run on a $\mathcal{C} = \frac{1}{4}$ square-lattice cluster state [14,17]. This resource state has the advantage that it can be generated scalably (by the process shown in Fig. 2). This protocol has the same features as in the $\mathcal{C} = 1$ case except with poorer noise properties [17]. Specifically, the lower edge weight $\mathcal{C} = \frac{1}{4}$ means that using the QRL in this projected fashion will introduce significantly more noise (due to finite squeezing) than will applying the full QRL protocol introduced here.

Bilayer square lattice. We also consider the highly scalable bilayer-square-lattice (BSL) resource state recently introduced in Ref. [10] (on which we are authors). Like the QRL, this state affords a similar macronode-based protocol, which we refer to here as the *BSL protocol*. Like with the above two cases, circuit flexibility is limited because quantum wires are restricted to run horizontally, and the natural two-mode gates (which includes, but is not limited to, the $\hat{C}_Z$ gate) can only be applied between nearest-neighbor wires. In terms of compactness, the BSL protocol is similar to the QRL protocol since the individual wires themselves are actually CV dual-rail wires [17]. For technical reasons, however, these wires require twice as many steps to implement each single-mode gate (four, as compared to the usual two). This results in poorer noise performance than the QRL protocol.

Thus our protocol shares the strengths of the others. It has relatively good noise performance (similar to the canonical CV cluster state), compactness (similar to the BSL), and scalability (similar to projected QRL and the BSL). In addition, it is the only protocol that allows highly flexible quantum-circuit design: the extra degrees of freedom per site allow for the quantum wires to be more flexibly directed and even to crisscross and intersect one another, thus simplifying two-mode interactions between initially distant wires. In addition, the broad class of two-mode gates that can be implemented with a single macronode measurement includes two-mode squeezing and a variable beamsplitter. Thus the QRL protocol is especially well suited to quantum-optics applications.

VI. CONCLUSION

In this article we generalized CV measurement-based protocols to a scalable cluster state known as the quad-rail lattice. This came with several advantages. In particular, we found that quantum wires can be threaded through the lattice sites, allowing for greater flexibility in implementing quantum circuits on the cluster. Unlike single-rail CV cluster-state wires [15], these wires are embedded versions of the *CV dual-rail wire* (discussed in Ref. [17]), and thus they are more compact and do not introduce excessive levels of noise due to finite squeezing [17]. Our protocol is also well suited to implementing a variety of two-mode gates at the intersection points of these wires—such as two-mode-squeezing and beamsplitter gates. Thus we have generalized the one-

dimensional macronode protocols introduced in Ref. [17] to the two-dimensional case.

Several features that our protocol exhibits—including non-local input states and the ability to reroute wires—are similar to those found in generalizations of measurement-based quantum computing based on tensor networks [30,31]. These similarities likely stem from their shared use of entangled pairs as basic building blocks. It is curious that these extra features are naturally exhibited in experimentally favorable schemes for implementing CV cluster-state computations. It is worth considering the possibility that macronode-based qubit resource states might show similar advantages.

This work highlights the importance of focusing on macronode-based construction methods of CV resource states for quantum computing [10–14], which also have the advantage of being the most scalable methods available to date. Adapting the measurement protocol to the quad-rail lattice—rather than converting it to the standard square-lattice resource—yields a richer, more dynamic mode of computation and opens further research avenues towards closing the gap between theoretical models and experimental implementations.

[1] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, Phys. Rev. Lett. **86**, 5188 (2001).

[2] H. J. Briegel and R. Raussendorf, Persistent Entanglement in Arrays of Interacting Particles, Phys. Rev. Lett. **86**, 910 (2001).

[3] 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).

[4] M. A. Nielsen, Optical Quantum Computation Using Cluster States, Phys. Rev. Lett. **93**, 040503 (2004).

[5] D. E. Browne and T. Rudolph, Resource-Efficient Linear Optical Quantum Computation, Phys. Rev. Lett. **95**, 010501 (2005).

[6] P. van Loock, C. Weedbrook, and M. Gu, Building Gaussian cluster states by linear optics, Phys. Rev. A **76**, 032321 (2007).

[7] N. C. Menicucci, S. T. Flammia, H. Zaidi, and O. Pfister, Ultracompact generation of continuous-variable cluster states, Phys. Rev. A **76**, 010302(R) (2007).

[8] N. C. Menicucci, S. T. Flammia, and O. Pfister, One-Way Quantum Computing in the Optical Frequency Comb, Phys. Rev. Lett. **101**, 130501 (2008).

[9] S. T. Flammia, N. C. Menicucci, and O. Pfister, The optical frequency comb as a one-way quantum computer, J. Phys. B **42**, 114009 (2009).

[10] 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, arXiv:1509.00484.

[11] M. Chen, N. C. Menicucci, and O. Pfister, Experimental Realization of Multipartite Entanglement of 60 Modes of the Quantum Optical Frequency Comb, Phys. Rev. Lett. **112**, 120505 (2014).

[12] 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).

[13] 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. Photon. **7**, 982 (2013).

[14] N. C. Menicucci, Temporal-mode continuous-variable cluster states using linear optics, Phys. Rev. A **83**, 062314 (2011).

[15] M. L. 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).

[16] H. Cable and D. E. Browne, Bipartite entanglement in continuous variable cluster states, New J. Phys. **12**, 113046 (2010).

[17] 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).

[18] M. Ohliger, K. Kieling, and J. Eisert, Limitations of quantum computing with Gaussian cluster states, Phys. Rev. A **82**, 042336 (2010).

[19] M. Ohliger and J. Eisert, Efficient measurement-based quantum computing with continuous-variable systems, Phys. Rev. A **85**, 062318 (2012).

[20] N. C. Menicucci, Fault-Tolerant Measurement-Based Quantum Computing with Continuous-Variable Cluster States, Phys. Rev. Lett. **112**, 120504 (2014).

[21] D. Gottesman, A. Kitaev, and J. Preskill, Encoding a qubit in an oscillator, Phys. Rev. A **64**, 012310 (2001).

[22] T. Eberle, S. Steinlechner, J. Bauchrowitz, V. Handchen, H. Vahlbruch, M. Mehmet, H. Muller-Ebhardt, and R. Schnabel, Quantum Enhancement of the Zero-Area Sagnac Interferometer Topology for Gravitational Wave Detection, Phys. Rev. Lett. **104**, 251102 (2010).

[23] G. Ferrini, J. Roslund, F. Arzani, Y. Cai, C. Fabre, and N. Treps, Optimization of networks for measurement-based quantum computation, Phys. Rev. A **91**, 032314 (2015).

[24] N. C. Menicucci, S. T. Flammia, and P. van Loock, Graphical calculus for Gaussian pure states, Phys. Rev. A **83**, 042335 (2011).

[25] N. Gabay and N. C. Menicucci, Passive interferometric symmetries of multimode Gaussian pure states, Phys. Rev. A **93**, 052326 (2016).

[26] S. L. Braunstein and H. J. Kimble, Teleportation of Continuous Quantum Variables, Phys. Rev. Lett. **80**, 869 (1998).

[27] S. D. Bartlett and B. C. Sanders, Efficient Classical Simulation of Optical Quantum Information Circuits, Phys. Rev. Lett. **89**, 207903 (2002).

[28] J. Wenger, R. Tualle-Brouri, and P. Grangier, Non-Gaussian Statistics from Individual Pulses of Squeezed Light, Phys. Rev. Lett. **92**, 153601 (2004).

[29] R. Ukai, J. I. Yoshikawa, N. Iwata, P. van Loock, and A. Furusawa, Universal linear Bogoliubov transformations through one-way quantum computation, Phys. Rev. A **81**, 032315 (2010).

[30] D. Gross, J. Eisert, N. Schuch, and D. Perez-Garcia, Measurement-based quantum computation beyond the one-way model, Phys. Rev. A **76**, 052315 (2007).

[31] D. Gross and J. Eisert, Novel Schemes for Measurement-Based Quantum Computation, Phys. Rev. Lett. **98**, 220503 (2007).