Measurement-based quantum computation with qubit and continuous-variable systems

by

Rafael N. Alexander

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the
School of Physics
Faculty of Science

January 2017
Quantum computers offer impressive computational speed-ups over their present-day (classical) counterparts. In the measurement-based model, quantum computation is driven by single-site measurements on a large entangled quantum state known as a cluster state. This thesis explores extensions of the measurement-based model for quantum computation in qubit and continuous-variable systems.

Within the qubit setting, we consider the task of characterizing how well a small-scale measurement-based quantum device can perform logic gates. We adapt a pre-existing scheme known as randomized benchmarking into the setting of measurement-based quantum computation on a one-dimensional cluster state. A key feature of randomized benchmarking is that it uses random sequences of gates. We show how the intrinsic randomness of measurement-based quantum computation can be harnessed when implementing them.

Within the continuous-variable setting, we consider optical cluster states that can be generated with current technology. We propose a compact method for generating universal cluster states based on optical-parametric-oscillator technology. We consider how finite-squeezing effects manifest in computation and show that pre-existing measurement-based protocols are suboptimal. We propose new measurement-based protocols that have better noise properties, compactness, and circuit flexibility. As an application, we introduce a measurement-based method for implementing interferometry. In this model, the finite-squeezing noise can be dealt with as a photon-loss process. Building further on this work, we investigate the resource requirements of a measurement-based BosonSampling device, proving simultaneous efficiency in time, space, and squeezing (energy) resources.

These results offer new insights into how to build, use, and characterize a measurement-based quantum computer.
Statement of contribution

The research contributions presented in chapters 2-6 of this thesis have been published within five refereed (or currently under review) research journal articles, as described below. Note that these chapters are not presented in the order in which they were published. On all five papers, I am the primary (lead) author.

Chapter 2: Randomized benchmarking in measurement-based quantum computation

This chapter contains the published paper Ref. [1]. This project was suggested by Stephen Bartlett. I undertook the technical analysis under the supervision of Stephen Bartlett and with input from Peter Turner. I wrote the paper with edits from Stephen Bartlett and Peter Turner.

Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

This chapter contains the published paper Ref. [2]. This project continued from work completed during my Honours year under the supervision of Nicolas Menicucci. During Honours, I specified how to implement gates using the single-rail continuous-variable quantum wire (CVW), macronode, and dictionary protocols, and had preliminary noise results. This constituted approximately 50% of the results in Secs. 3.3 and 3.4. The remainder of the results in these sections as well as those in Sec. 3.5 were completed during my PhD. I undertook the technical analysis under supervision of Nicolas Menicucci. The entirety of the paper was written by me during my PhD with edits from Nicolas Menicucci and Seiji Armstrong and input from Ryuji Ukai. The results in Sec. 3.4.1 were independently derived by Ryuji Ukai and Seiji Armstrong.

Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

This chapter contains the accepted paper Ref. [3]. This project was based on an initial experimental design suggested by Pei Wang, Niranjan Sridhar, Moran Chen, and Olivier Pfister. Further development of the experimental design was undertaken by all authors, lead by Pei Wang under the supervision of Olivier Pfister. This part of the project is included as Secs. 4.3 and 4.4 of the paper, which were written by Pei Wang and Olivier Pfister with edits by all authors. I undertook design of the measurement protocol
and theoretical analysis under the supervision of Nicolas Menicucci. These theoretical aspects make up the majority of the paper. I wrote the other sections of the paper with edits by Pei Wang, Olivier Pfister, Nicolas Menicucci, and input from all authors.

Chapter 5: Flexible quantum circuits using scalable continuous-variable cluster states

This chapter contains the published paper Ref. [4]. I suggested this project. I undertook the technical analysis with input from Nicolas Menicucci. I wrote the paper with edits from Nicolas Menicucci.

Chapter 6: Measurement-based linear optics

This chapter contains the preprint article Ref. [5], which has been submitted for publication. This work is based on initial calculations I performed with Natasha Gabay with input from Nicolas Menicucci. I undertook the technical analysis supervised by Nicolas Menicucci with input from Natasha Gabay and Peter Rohde. I wrote the paper with edits from all authors.
Acknowledgements

I am deeply grateful to my supervisors Nick Menicucci and Stephen Bartlett. Nick has been an excellent mentor throughout my PhD and during my Honours year. He has supplied me with a wealth of knowledge, high research standards, and motivation. Working with him has been an extremely worthwhile and enjoyable experience. Darmok and Jalad on the ocean. Over the years, Stephen has played an important role in my academic development, starting with an introduction into the field of quantum information. Ever since then, he has been reliable source of encouragement, expertise, and guidance. I am also grateful to Andrew Doherty and Steve Flammia for their advice, teachings, and anecdotes.

My PhD experience was thoroughly enriched by my discussions and experiences with colleagues, friends, and family. I would like to thank the members of the quantum information science group at the University of Sydney for maintaining a stimulating and supportive atmosphere. I am particularly grateful to Harrison Ball, Courtney Brell, Jacob Bridgeman, Simon Burton, Chris Chubb, Andrew Darmawan, Natasha Gabay, Chris Granade, Robin Harper, Angela Karanjai, Richard Küng, Hakop Pashayan, Sam Roberts, Alan Robertson, and Dominic Williamson. I would also like to thank members of the RMIT School of Science for providing me with a friendly and welcoming environment during my stay in Melbourne. Special thanks goes to Jamie Booth, Jared Cole, Andrew Greentree, Lachlan Hennessy, Salvy Russo, Jackson Smith, Scott Todd, Jesse Vaitkus, Nicolas Vogt, and Sam Wilkinson. For all their help, insights, and contributions, I thank my collaborators: Seiji Armstrong, Moran Chen, Natasha Gabay, Oliver Pfister, Peter Rohde, Niranjan Sridhar, Peter Turner, Ryuji Ukai, and Pei Wang. I would also like to thank members of the broader research community for their contributions and discussions. In particular, I thank Francesco Arzani, Ben Baragiola, Gavin Brennen, Carl Caves, Josh Combes, Tommaso Demarie, Tom Douce, Giulia Ferrini, Kosuke Fukai, Akira Furusawa, Mirko Lobino, Nana Liu, Damian Markham, Kavan Modi, Gerardo Paz-Silva, Felix Pollock, David Poulin, Robert Raussendorf, Peter van Loock, and Shota Yokoyama.

Some of the most valuable discussions and learning experiences I had during my PhD were during visits to overseas research groups. For giving me this opportunity, I am indebted to my hosts Peter van Loock, Giulia Ferrini, Thomas Coudreau, Pérola Milman, Robert Raussendorf, David Poulin, Andrew Darmawan, Olivier Pfister, and Tommaso Demarie.

For all their kind help, I thank Geraldine Arriesgado, Lorraine Dimassi, Alexis George, Arwen Pagon, Leanne Price, Eve Teran, and Wicky West.
Finally, I thank my family and friends for their constant love and support, especially Veronica Quinteros and Carmen Núñez-Durán. Special thanks go to Michael Alexander, James Beattie, Caitlin Carter, Jenny Wu, and Whisky for keeping me sane (at least, more so than usual) in the final year.
# Contents

<table>
<thead>
<tr>
<th>Abstract</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement of contribution</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
</tbody>
</table>

1 Introduction 1
   1.1 Notation and definitions for qubit computation ................... 2
   1.2 Measurement-based quantum computation with qubits ............... 4
      1.2.1 Measurement-based gates .................................. 4
      1.2.2 Fault tolerance ........................................... 8
   1.3 Quantum computation with continuous variables ................... 8
      1.3.1 Notation and definitions .................................. 9
   1.4 Measurement-based quantum computation with
      continuous-variables cluster states .......................... 12
      1.4.1 Measurement-based gate implementation ................... 13
      1.4.2 Fault tolerance with continuous-variable systems .......... 16
      1.4.3 A brief history of optical continuous-variable cluster states . 18
   1.5 Thesis structure ............................................. 19
   1.6 List of acronyms used in this thesis .......................... 21

2 Randomized benchmarking in measurement-based quantum computing 23
   2.1 Abstract ..................................................... 23
   2.2 Introduction ............................................... 23
   2.3 Background .................................................. 25
      2.3.1 Preliminaries ............................................ 25
      2.3.2 Randomized benchmarking ................................. 26
      2.3.3 Measurement-based quantum computation ............... 28
   2.4 Randomized benchmarking in MBQC ............................. 29
      2.4.1 RB within the measurement-based model ............... 30
      2.4.1.1 Noise in MBQC logic gates ........................... 30
      2.4.1.2 SPAM errors ........................................ 31
      2.4.2 RB using the Clifford group ........................... 31
2.4.2.1 The role of randomness .......................... 32
2.4.3 RB using derandomized 2-designs .................. 34
2.5 Conclusion ............................................. 36
2.6 Appendix: Average gate fidelity derivation .......... 37
2.7 Appendix: Clifford angles .......................... 39
2.8 Appendix: 2-design elements ........................ 40

3 Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states 41
3.1 Abstract ................................................. 41
3.2 Introduction .......................................... 42
3.3 CVW protocol .......................................... 45
3.3.1 MBQC on uniformly-weighted CVWs .......... 46
3.3.2 Noise dependence on \( g \) ......................... 50
3.3.3 Wigner function formalism ....................... 53
3.4 Macronode protocol ..................................... 59
3.4.1 Implementing gates .................................. 64
3.4.2 Scalar variance for the macronode protocol .... 65
3.4.3 Protocol comparison: noise per Gaussian unitary 67
3.4.4 Dictionary protocol .................................. 68
3.5 Application: Number of measurements per gate .... 72
3.5.1 CVW protocol ........................................ 73
3.5.2 Dictionary protocol .................................. 74
3.6 Conclusions and Discussion .......................... 77
3.7 Appendix: Comparative noise bounds between the measurement protocols 78
3.7.0.1 Comparing the CVW and macronode protocols 78
3.7.0.2 Comparing the dictionary and general macronode protocols 80
3.7.0.3 Comparing the dictionary protocol to the CVW protocol 81
3.8 Appendix: Proof of scalar variance divergences for gates with high squeezing 82

4 One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator 85
4.1 Abstract ................................................. 85
4.2 Introduction .......................................... 86
4.3 State generation ....................................... 88
4.4 Experimental Details ................................... 89
4.5 Basics of quantum computing on the bilayer square lattice 90
4.5.1 Computing with macronodes ...................... 92
4.5.2 Keeping square graphs disconnected .............. 94
4.6 Universal gate set ..................................... 95
4.6.1 Single-qumode gates ............................. 95
4.6.2 Projective measurement ......................... 98
4.6.3 Two-qumode gates ................................ 98
4.6.4 Alternative representation of two-qumode gate implementation 100
4.6.5 Achieving Universal Quantum Computation ...... 102
4.7 Displacements and finite-squeezing effects ......... 102
I dedicate this thesis to my brothers Fernando, Oscar, Samuel, Francisco, and Felipe.
Chapter 1

Introduction

Computers are machines that process information, thereby assisting us in problem solving. They are physical devices, and therefore the mathematical models we use to describe them have physically motivated axioms built in. How information can be stored, processed, and readout fundamentally depends on the available means by which we can encode, controllably manipulate, and measure the information within the degrees of freedom of a physical device. Modern computers are governed by the laws of classical physics, and hence, they are referred to as classical computers.

Recent progress in the field of quantum information science has improved the feasibility of building a quantum computer, which is a device that harnesses the rules of quantum mechanics for computation. Part of what makes quantum computation interesting is that it is believed that classical computers cannot efficiently simulate quantum mechanical systems of many particles. For example, BosonSampling [6] is a task that involves sampling from the output distribution of a random multi-port interferometer whose ports are each fed either one or zero photon(s). There is strong evidence suggesting that classical computers cannot efficiently complete this task. Running computation through a controlled quantum system allows us to efficiently solve problems that are not known to admit an efficient classical algorithm, and the potential applications extend beyond just simulating quantum systems that are of interest to physicists. The classic example of this is the integer factorization problem, which admits an efficient solution via Shor’s quantum algorithm [7].

There are many equivalent models of quantum computation, including the circuit model [8], measurement-based model [9], adiabatic model [10], and topological model [11]. Each of these can be realized in a variety of physical architectures, including superconductor circuits [12, 13], trapped ions [14, 15], quantum dots [16], and photonic [17, 18] or continuous-variable [19, 20] implementations with optical modes. The focus of this
thesis is on the measurement-based model with applications to optical implementations. We propose various extensions of the measurement-based model, including the characterizing of noisy quantum gates, proposing extremely compact resource generation, new measurement-based protocols tailored to scalable resource states, and a new method for implementing scalable interferometry.

In this chapter, we review the basics of qubit and continuous-variable (CV) measurement-based quantum computation (MBQC). We will only give a broad-brush summary, focusing on some key concepts, definitions, and notation. Where relevant, we provide references for further details.

1.1 Notation and definitions for qubit computation

The study of computation involves accounting for the physical resources (e.g., time, space, energy) required to solve problems, and in particular, how the required amount scales with the input size of the problem in the asymptotic limit. If the required resources scales at most polynomially with this input size (in the asymptotic limit), then we say that the computation is efficient.

Quantum computation is most commonly described in terms of qubits, i.e., two-level quantum mechanical systems that are used to store quantum information [8, 21, 22]. The Hilbert space of a single qubit is spanned by the computational basis \{\ket{0}, \ket{1}\}, which is given the matrix representation

\[
\ket{0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \ket{1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.
\]

(1.1)

Single qubit state space contains superpositions of these basis states, generally written as \(\alpha \ket{0} + \beta \ket{1}\) for some \(\alpha, \beta \in \mathbb{C}\) with \(|\alpha|^2 + |\beta|^2 = 1\). Below, we will make use of the alternative single-qubit basis given by \(\ket{\pm} := (\ket{0} \pm \ket{1})/\sqrt{2}\). The \(n\)-qubit state space is spanned by the computational basis \(\{\ket{b} = \bigotimes_{i=1}^{n} \ket{b_i}\}\) for all length-\(n\) bit strings \(b = b_1b_2\ldots b_n\). Included with this space are entangled states, such as \((\ket{000} + \ket{111})/\sqrt{2}\) for \(n = 3\) [21].

Quantum computations can be represented by quantum circuits consisting of a standard input state (most commonly, all qubits in the \(\ket{0}\) state), a network of gates from a finite gate set (representing time evolution of the input), and single-qubit measurements in the computational basis. This is known as the quantum circuit model [8]. A set of gates is said to be universal for quantum computation if any unitary operation can be approximated to arbitrary accuracy using only these gates.
Definitions and circuit representations of some standard single- and two-qubit gates are shown in Fig. 1.1. Quantum measurement devices corresponding to some observable $E$ project individual qubits into an eigenstate $|e_i\rangle$ conditioned on getting outcome $e_i$. These are represented by a “measurement box”, as shown in Fig 1.1 (note the two equivalent different styles). Double thick lines are used to represent the classical outcomes of such measurements. These are sometimes connected to unitary gates to indicate that those gates are to be applied only conditioned on getting a certain measurement outcome.

We now define some useful gate sets. The $n$ qubit Pauli group $\mathcal{P}_n$ is defined to be the group generated by the $n$-fold tensor products of all Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$  \hspace{1cm} (1.2)

The $n$-qubit Clifford gates $\mathcal{C}_n$ is the set of gates (which forms a group) that is the normalizer of the Pauli group:

$$\mathcal{C}_n := \{ U \in U(2^n) \mid U^\dagger PU \in \mathcal{P}_n, \forall P \in \mathcal{P}_n \}.$$  \hspace{1cm} (1.3)

This group is generated by the gate set $\{ Z_{\pi/2}, H, C_Z \}$ on $n$ qubits. $\mathcal{C}_n$ is not universal for quantum computation, and circuits consisting of only Clifford gates, input $|0\rangle$ states, and measurements in the Pauli $X$, $Y$, and $Z$ bases can efficiently simulated by a classical computer. This result is known as the Gottesman-Knill theorem [23].

We can construct a universal gate set by combining $\mathcal{C}_n$ with a non-Clifford gate. The most commonly used gate for this is the $Z_{\pi/4}$ gate.
Next, we will use these definitions in order to introduce the measurement-based model for quantum computation. For a more general introduction to quantum computation, we recommend Refs. [8, 22].

1.2 Measurement-based quantum computation with qubits

The measurement-based model [9] has offered many experimental and theoretical insights into quantum computation. It restructures the task of building a quantum computer into just two essential steps: 1) construct a sufficiently entangled resource state, and 2) perform single-site measurements specified by the desired computation.

Note that there is no need for an on-demand entangling operation as all of the required entanglement is generated during state preparation. This fact can be used to reduce the overhead in some schemes for quantum computation. The classic example is the proposal by Nielsen [18], which applies to linear optics quantum computing [17] wherein on-demand entangling gates are challenging. The measurement-based model has also lead to significant theoretical results in neighbouring fields of quantum information. These include new insights into quantum foundational questions [24, 25], cryptographic protocols for quantum computing [26, 27], and many-body physics [28, 29].

Over the years, the original protocol [9] has been extended into a rich framework [30–35] that is applicable to a variety of quantum states [36–39]. Our focus will be closely aligned to the original model [9] where quantum circuits are translated into a sequence of measurements applied to a multi-qubit resource state known as the cluster state [40]. This state can be generated on \( n \) qubits by a constant-depth circuit of commuting gates defined by some \( n \)-node graph \( G \):

\[
\prod_{\langle i, j \rangle_G} C_{Z_{i,j}} \bigotimes_{k=1}^{n} |+\rangle_k
\]

(1.4)

where \( \langle i, j \rangle_G \) restricts pairings to being nearest neighbours on the graph \( G \). Below we describe the basics of implementing gates via single-site measurements on cluster states.

1.2.1 Measurement-based gates

We will begin by considering single-qubit gates, and in particular, gates of the form \( M_{\theta,m} := X^m H Z_\theta \). The usefulness of \( M_{\theta,0} \) for single-qubit quantum computation depends on the available choices of the parameter \( \theta \). For instance, for the set \( \{ \theta = \frac{n\pi}{2} \text{ for } \theta \in \mathbb{N} \} \)
Chapter 1: Introduction

Figure 1.2: (a) Single-step cluster state measurement for implementing the $M_{\theta,m}$. (b) A graphical representation of the cluster state. Qubit labels match (a). The input is indicated in green. The measurement is indicated by the $\theta$ symbol. (c) Logical circuit corresponding to this process. We separate the measurement outcome dependence of this gate, defining $M_{\theta} := M_{\theta,0}$.

generates the single-qubit Clifford group, whereas, the set $\{\theta = \frac{n\pi}{4} \text{ for } \theta \in \mathbb{N}\}$ is universal for single-qubit quantum computation [8].

The gate $M_{\theta,0}$ can be applied to an arbitrary single-qubit input state $|\psi\rangle$ by performing a single-site measurement on the two-qubit state $C_{Z_1,2} |\psi\rangle_1 |+\rangle_2$, as shown in Fig. 1.2. Qubit 1 is measured in the basis

$$X(\theta) := Z_{\theta}^1XZ_{\theta} = c_{\theta}X - s_{\theta}Y,$$

(1.5)

after which, qubit 2 is in the state $M_{\theta,m} |\psi\rangle$, where each outcome $m \in \{0, 1\}$ is equally likely. In the case where $m = 1$, $M_{\theta,0}$ can be implemented by applying an $X$ gate afterwards. Note that it is the set of possible measurement bases $X(\theta)$ that specifies which gates that can be implemented.

This approach can be extended to implement a sequence of gates $U = M_{\theta_n,0} \ldots M_{\theta_1,0}$ on a single qubit. We require the preparation of a linear cluster state and the ability to perform measurements of the form $X(\theta)$, as shown in Fig. 1.3 (a, b).

By concatenating the single-step protocol described above, we see that the general case with random measurement outcomes results in one of $2^n$ possible unitary gates

$$M_{\theta_n,m_n} \ldots M_{\theta_1,m_1} = X^{m_n}M_{\theta_n,0} \ldots X^{m_1}M_{\theta_1,0},$$

(1.6)

which are generated by the probabilistic $X$ gates that interleave the desired $M_{\theta,0}$ gates. Note that

$$M_{\theta_k,0}X^{m_i} = Z^{m_i}M_{(-1)^{m_i}\theta_k,0}$$

(1.7)

and therefore, an unwanted $X$ gate acting before the gate $M_{\theta,0}$ is equivalent to a $Z$ gate acting after $M_{-\theta,0}$. If measurement at site $k$ is delayed until after result $m_i$ is known, then adapting the measurement angle $\theta_k \mapsto (-1)^{m_i}\theta_k$ allows us to “pull through” the
unwanted $X$ gate leftwards (as in Eq. 1.7), thus implementing the $M_{\theta,0}$ followed by a known Pauli. Doing this will sometimes result in having some unwanted $Z$ gates interleaving the $M_{\theta,0}$ gates as well. These can be pulled through as follows

$$M_{\theta,0}Z^{m_1} = X^{m_1}M_{\theta,0}. \quad (1.8)$$

Therefore, by adapting the measurement bases conditioned on past measurement outcomes (known as *feedforward*) the overall evolution deterministically implements the desired $U$ up to a known final Pauli gate (as shown in Fig. 1.3 (c)), which can be corrected afterwards.

In order to implement universal quantum computation, we also require a method for implementing entangling gates. To do so on input states $|\psi\rangle$ and $|\phi\rangle$, the required state preparation and measurement procedure is as shown in Fig. 1.3 (d). We will refer to this type of cluster state as a “horseshoe network” due to the shape of its graph, shown in Fig. 1.3 (e). Measuring qubit 1 (2) in basis $X(\theta_1)$ ($X(\theta_2)$) teleports the encoded state onto qubit 3 (4) and implements the two-qubit gate

$$(-1)^{m_1m_2}[(X^{m_1}Z^{m_2}) \otimes (X^{m_2}Z^{m_1})]C_Z [M_{\theta_1,0} \otimes M_{\theta_2,0}]. \quad (1.9)$$

The $m_1 = m_2 = 0$ case is equivalent to a $C_Z$ gate preceded by $M_{\theta,0}$ gates, as shown in Fig. 1.3 (f). This gate forms a universal gate set when combined with a gate set

---

**Figure 1.3:** All conventions are as in Fig. 1.2. For a clearer presentation, we omit subscript labels in subfigures (b, c, e, f). (a) Implementing measurements on a linear cluster state. (b) Graph of a linear cluster state. (c) Logic circuit corresponding to this process. (d) generating and measuring a horseshoe network. (e) Graph of the horseshoe network cluster state. (f) Logic gate corresponding to this process.
Chapter 1: Introduction

Figure 1.4: (a) The square lattice cluster state is a universal resource for measurement-based quantum computation. With appropriately encoded inputs (for example, those shown in green), single-site measurements can reduce the square lattice into a network made up of wires and horseshoes. Performing $Z$ measurements “deletes” qubits, and setting $\theta = 0$ corresponds to a measurement in the $X$ basis, implementing entanglement swapping. Measuring the middle two rows as shown results in a state that is equivalent to the cluster state in (b) up to local Pauli gates that depend on the measurement-outcomes. These additional Pauli-gates can be dealt with using feedforward when implementing measurement-based gates. (b) Network composed of wires and horseshoes. (c) Corresponding logic circuit.

for universal single-qubit computation. The other cases only differ by a final two-qubit Pauli gate.

Unwanted Pauli gates preceding this gate can be pulled through the $M_{\theta,0}$ gates by measurement adaptivity following the procedure described above, and then through the $C_Z$ gate, which merely maps the Pauli gate to some other element of $P_2$ because $C_Z$ is a Clifford gate.

Therefore, the resource states shown in Figs. 1.3 (b) and (e) enable the measurement-based implementation of a single and two-qubit gate (up to a known Pauli gate), respectively.

A cluster state (or some other entangled resource state) is universal if arbitrary quantum circuits can be implemented on it via single-site measurements. A cluster state with a square lattice graph is universal because the above wire and horseshoe networks can be realized as subnetworks within it by using measurements in the $X$ and $Z$ basis. In Fig. 1.4 we show a minimal example on this resource that is used to implement both single- and two-qubit gates.

Thus universal gates can be implemented via single-site measurements on an appropriate cluster state. In order to perform measurement readout of the information in the cluster state, cluster sites containing logical qubits can simply be measured in the $\hat{Z}$ basis, remembering to account for the final Pauli correction.
1.2.2 Fault tolerance

How to compute whilst mitigating against the noise that is inherent to quantum systems is one of the most active areas of research in quantum information theory. Arbitrarily long quantum computations are possible with modest overhead using quantum error correction \[41\] provided that the noise acts locally (with respect to the physical qubits) and that the noise level is below some threshold value. This celebrated result is known as the threshold theorem for fault-tolerant quantum computation \[42, 43\].

The standard approach to implementing a fault-tolerant quantum computation involves encoding the components of the desired quantum logic circuit—the qubits, gates, and measurement readout—into some quantum error correcting code (or combinations of various codes) \[44\]. There are many different types of quantum error correcting codes and these vary in terms of resource overhead, noise tolerance, and which fault-tolerant gates can be implemented conveniently \[45–48\]. For a review of this subject see Refs. \[8, 41, 49, 50\]. Note that there also exists schemes that apply specifically to measurement-based model for quantum computation \[51–53\].

For quantum computation with continuous-variable systems (discussed in the next section), all known protocols for fault-tolerant quantum computation make use of encoded discrete-variable systems (such as qubits) on which standard quantum error correction can be applied. We will review some of the developments in this field later in Sec. 1.4.2.

The remainder of this thesis will only deal with the issue of fault tolerance indirectly. Nevertheless, the results introduced in later chapters may be useful in characterizing the noise processes on the underlying quantum hardware of continuous-variable cluster state computations, thereby informing quantum error correction strategies.

1.3 Quantum computation with continuous variables

Not all quantum systems can be simply described in terms of qubits, and yet some of these systems are viable candidates for implementing quantum computation. For instance, the complex quadrature amplitudes that describe modes of the quantized electromagnetic field are fundamentally continuous-variable degrees of freedom. Exploring quantum computation over such systems has yielded several important results, such as protocols that allow for deterministic and large-scale generation of universal resource states for measurement-based quantum computation by using compact laser systems \[3, 54–58\].
1.3.1 Notation and definitions

For consistency with later chapters and to avoid notational ambiguity with the qubit case, all CV operators will have “^” hats. An individual CV system will be referred to as a *qumode* or a *mode* (interchangeably). In an *n*-mode system, we label the operators corresponding to the continuous degrees of freedom of interest by $\hat{q} := (\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_n)^T$, and their conjugate variables by $\hat{p} := (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n)^T$. These satisfy the canonical commutation relations (with $\hbar = 1$)

$$[\hat{q}_j, \hat{p}_k] = i\delta_{j,k}, \quad (1.10)$$

where $\delta_{j,k}$ here is the Kronecker delta function. We will refer to each $\hat{q}_i$ ($\hat{p}_i$) as a position (momentum) variable, even when describing systems other than a particle moving through space. In particular, we will be interested in the case where $\hat{q}$ and $\hat{p}$ represent the quadratures of the electromagnetic field, which are the real and imaginary part of the complex field amplitude respectively.

Both kinds of operators have continuous spectra, with eigenstates

$$\hat{q} |r\rangle_q = r |r\rangle_q \quad (1.11)$$

$$\hat{p} |t\rangle_p = t |t\rangle_p \quad (1.12)$$

$\forall r, t \in \mathbb{R}$. These each form an orthonormal and complete basis

$$\int_{-\infty}^{\infty} |r\rangle_{q,p} \langle r|_{q,p} dr = \mathbb{I} \quad (1.13)$$

$$\langle r|_{q,p} |t\rangle_{q,p} = \delta(r - t) \quad (1.14)$$

where here $\delta$ is the Dirac delta function. These bases are related by the unitary Fourier transform operation

$$|t\rangle_p := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\pi r} |r\rangle_q \, dr =: \hat{F} |t\rangle_q. \quad (1.15)$$

Intuitively, the $\hat{q}$ and $\hat{p}$ bases can be thought of as generalizations of the qubit computational and $\{+\rangle, -\rangle\}$ bases, with the Fourier transform generalising the Hadamard gate. A general single-mode pure state can be represented by a normalized wave function

$$|\psi\rangle = \int_{-\infty}^{\infty} \psi(r) |r\rangle_q \, dr. \quad (1.16)$$
Gaussian pure states are an important class of states that have a particularly simple structure. They can be represented up to a global phase by an $n$-mode wavefunction

$$\psi_{Z,c,d}(q) = \pi^{-n/4}(\operatorname{det} \operatorname{Im} Z)^{1/4}e^{\frac{1}{2}(q-c)^T Z (q-c)} e^{id^T q}$$

(1.17)

for some $c, d \in \mathbb{R}^n$ and $n \times n$ complex symmetric matrix $Z$ with positive definite imaginary part [59]. $Z$ is an adjacency matrix that specifies the graph of the Gaussian pure state, and is usually written as $Z = V + iU$ with real-valued $n \times n$ symmetric matrices $V$ and $U$, such that $U > 0$ [59].

Note that the position and momentum basis states themselves are not normalizable and do not represent physical states. Their wave functions correspond to delta functions

$$|r\rangle_{q,p} = \int_{-\infty}^{\infty} \delta(t-r) |t\rangle_{q,p} dt.$$  

(1.18)

We will consider them as displaced versions of the extremal points in a family of Gaussian pure states parametrized by the squeezing factor $s > 0$

$$\psi_{s^{-2},0,0}(q) = (s\sqrt{\pi})^{-1/2} e^{-\frac{q^2}{2s^2}}.$$  

(1.19)

The $s \to 0$ and $s \to \infty$ limits both correspond to infinite squeezing, resulting in a position and momentum eigenstate respectively:

$$\lim_{s \to 0} \int_{-\infty}^{\infty} \psi_{s^{-2},r,0}(t) |t\rangle_q dt = |r\rangle_q$$  

(1.20)

$$\lim_{s \to \infty} \int_{-\infty}^{\infty} \psi_{s^{-2},0,r}(t) |t\rangle_q dt = |r\rangle_p.$$  

(1.21)

The state $\psi_{s^{-2},0,0}(q)$ is referred to as the squeezed vacuum state since the unsqueezed $s = 1$ case corresponds to the ground state wavefunction of a quantum harmonic oscillator. In the next section, we will describe the CV measurement-based protocol in terms of position and momentum eigenstates. However it should be understood that implementing it will necessarily involve substituting in a physical approximate state (such as a squeezed Gaussian state).

Now we introduce the basics of the CV computational model, which was originally formulated by Lloyd and Braunstein [19]. A gate set is considered universal for CV computation if it can be used to approximate arbitrarily well the evolution under any Hamiltonian that is polynomial in $\hat{q}_i$ and $\hat{p}_i$, $\forall i$ [19]. We will consider gate sets composed of single- and two-mode gates. Computations can be described by quantum circuits made up of a network of gates from a universal gate set applied to modes in a standard initial
Chapter 1: Introduction

In measurement-devices, when $\hat{E}$ is some linear combination of $\hat{q}$ and $\hat{p}$, the possible outcomes are some $m \in \mathbb{R}$. Setting $\hat{E} = (\hat{q}^2 + \hat{p}^2 - 1)/2 = \hat{n}$ has $m \in \mathbb{N}$, corresponding to a photon counting measurement.

state (most commonly, $\psi_{i,0,0}(q)$), terminating in measurement-readout steps in the $\hat{q}$ basis.

In addition to the usual time and space resources considered in the qubit circuit model (measured by the number of modes and depth of the quantum circuit), we must also consider the importance of squeezing as a resource [60–63]. In the optical setting, high squeezing corresponds to a high photon number (energy) as $E \propto s^2 + s^{-2}$. In this thesis (particularly, in Ch. 6), we will require that the energy in an optical CV computation scales at most as a polynomial in the instance size of the problem in order for it to be considered efficient.

Some commonly used CV operations and circuit identities are defined in Fig. 1.5. The $n$-mode Weyl-Heisenberg group $\mathcal{W}_n$ is generated by the $n$-fold tensor product of shift operators $\hat{X}(s)$ and $\hat{Z}(t)$. This group is analogous to $\mathcal{P}_n$ for qubits.

The Gaussian unitary group $\mathcal{G}_n$ is a Lie group whose corresponding Lie algebra consists of all quadratic polynomials in $\hat{q}_i$ and $\hat{p}_i$, $\forall i$ (with the commutator as the Lie bracket). When acting by conjugation, it implements linear transformations on the set of position and momentum operators, and therefore, it normalizes $\mathcal{W}_n$. Thus, $\mathcal{G}_n$ can be seen as the CV analog of the Clifford group. When applied to states, a Gaussian unitary maps Gaussian pure states to Gaussian pure states [64]. $\mathcal{G}_n$ is generated by the gates $\{\hat{X}(s), \hat{Z}(t), \hat{F}, \hat{P}(\sigma), \hat{C}_Z(1) \mid \forall s, t, \sigma \in \mathbb{R}\}$ [65].

Circuits composed of Gaussian unitaries, Gaussian pure states, and measurements of observables that are linear combinations of the position and momentum operators are known to admit efficient classical simulation [66]. This is the CV analogue of the Gottesman-Knill theorem for qubits.
Chapter 1: Introduction

We can arrive at a universal gate set by extending the Gaussian unitary group by a non-Gaussian element, such as a the cubic phase gate $\hat{K}(\chi)$ [19, 67]. Note that we only require the ability to implement this gate for some particular $\chi \in \mathbb{R}$ [19, 65]. Another way to achieve universality is to allow for measurements of degree-2 (or higher) polynomials in the position and momentum operators, such as photon number resolving measurements $(\hat{q}^2 + \hat{p}^2 - 1)/2 = \hat{n}$ [68].

Next we describe how the CV quantum computation can be implemented by measurement-based quantum computation.

1.4 Measurement-based quantum computation with continuous-variables cluster states

Recall that measurement-based quantum computation proceeds via single-site measurements on a premade entanglement resource state. Here we will review the basics of the original proposal using CVs, introduced in Refs. [20, 65].

We consider entangled states known as continuous-variable cluster states (CVCSs). An ideal $n$-mode CVCS is defined via a real-valued $n \times n$ symmetric adjacency matrix $A$:

$$\hat{C}_Z(A) = \bigotimes_{k=1}^n |0\rangle_{p,k},$$

where $\hat{C}_Z(A) = \exp(i \frac{1}{2} \hat{q}^T A \hat{q})$, which applies a $\hat{C}_Z$ gate of weight $A_{i,j}$ between modes $i$ and $j$. Recall that qubit cluster states are unambiguously specified by graphs with no edge weights (see equation (1.4)). This is possible because the qubit $C_Z$ gate is unique and self inverse. In the CV case, there is a continuous family of $\hat{C}_Z$ gates that can be applied between any two modes, therefore ideal CV cluster state must be specified by graphs with real-valued edge weights.

Ideal CVCSs are not physical states because they are defined using momentum eigenstates. We will approximate them using Gaussian states. For example, one can replace the momentum eigenstates in equation (1.22) with highly squeezed vacuum states (the large $s$ case). More generally, we will consider any Gaussian pure belonging to a family $\psi_{Z(s),0,0}(\hat{q})$, which is parametrized by some global squeezing factor $s \geq 0$, to be an approximation to the ideal state defined by $A$ if

$$\lim_{s \to \infty} Z(s) = A. \quad (1.23)$$

These families of states are referred to as approximate CVCSs [59].
Chapter 1: Introduction

Next we will describe how single-site measurements can be used to implement a universal gate set on an ideal CV cluster state. We consider the standard choice: a square lattice graph with all edge weights equal to one. In this chapter we will forego a description of MBQC on approximate CVCSs, though details can be found within Ref. [65]. The reason is that such effects are already treated more generally in Ch. 3, and we generalize these results to CVCS with other graphs in later chapters.

1.4.1 Measurement-based gate implementation

We will start by defining the single-mode gate \( \hat{L}(b,c) := \hat{F}\hat{K}(b\chi)\hat{P}(c) \), where \( \chi \in \mathbb{R}\setminus0 \) is a fixed parameter, and \( b \in \{0,1\} \), \( c \in \mathbb{R} \) are degrees of freedom. Intuitively, this set of operations constitutes an “on-off” degree of freedom on some fixed \( \hat{K}(\chi) \) gate, and complete continuous freedom of the Gaussian shear gate \( \hat{P}(c) \). This gate set is universal for single-mode CV quantum computation [19, 65]. In order to implement these gates, we will require the ability to measure

\[
\hat{p}(b,c) := \hat{P}^\dagger(c)\hat{K}^\dagger(b\chi)\hat{p}\hat{K}(b\chi)\hat{P}(c) = \hat{p} + b\chi q^2 + c\tilde{q}.
\]

on single CVCS sites.

A single \( \hat{L}(b,c) \) gate can be implemented on some input state \( |\psi\rangle \) by a single-site measurement of \( \hat{p}(b,c) \) on a two-mode CV cluster state \( C_Z(1,2)|\psi\rangle_1|0\rangle_p,2 \), as shown in Figs. 1.6(a, b). This implements \( \hat{X}(m)\hat{L}(b,c) \), conditioned on getting some measurement outcome \( m \in \mathbb{R} \). Correcting with \( \hat{X}(-m) \) allows for deterministic implementation of \( \hat{L}(b,c) \).

Measurement-based implementation of a sequence of \( \hat{L}(b,c) \) gates with different \( b \in \{0,1\} \) and \( c \in \mathbb{R} \) requires a series of single-site measurements on a linear cluster state, as shown in Figs. 1.7(a, b). Each measurement implements a random gate, generated by

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1_6}
\caption{(a) Measurement-based circuit that implements the gate \( \hat{X}(m)\hat{L}(b,c) \). The entangling operation shown is a \( C_Z(1) \) gate, where we have omitted the gate parameter label. (b) Two mode CV cluster state including an arbitrary input state in the green coloured node. The parameters \( b \) and \( c \) indicate measurement in some basis \( \hat{p}(b,c) \). The edge weight of the graph is equal to one. (c) Single mode logic circuit implemented by one-step continuous-variable measurement-based quantum computation.}
\end{figure}
Chapter 1: Introduction

Figure 1.7: Measurement-based gate implementations. Conventions are as in Fig. 1.6. We omit subscripts in subfigures (b, c, e, f). (a) Preparation and measurement of a linear continuous-variable cluster state. (b) Graphical representation of a linear cluster state. All links are weight one. (c) Logical circuit corresponding to this circuit. Arbitrary single-mode unitaries can be approximated with circuits of this form. Note the Weyl-Heisenberg shift operators that must be corrected at the end. (d) Construction and measurement of the continuous-variable “horseshoe network” cluster state. This is used to implement an entangling gate between inputs $|\psi\rangle$ and $|\phi\rangle$. (e) Graph of the continuous-variable horse-shoe network state, all links are weight one. (f) Logic circuit implemented by measuring through the horseshoe network. The entangling gate here is $\hat{C}_Z(1)$.

the position displacement degree of freedom corresponding to the random measurement outcomes, $m_i \in \mathbb{R}$ $\forall i$. These interleave the desired gates:

$$X(m_n)\hat{L}(b_n, c_n)X(m_{n-1})\hat{L}(b_{n-1}, c_{n-1}) \ldots X(m_2)\hat{L}(b_2, c_2)X(m_1)\hat{L}(b_1, c_1). \quad (1.25)$$

As with the qubit case, the unwanted displacements can be “pulled through” to the end, simply resulting in a single overall Weyl-Heisenberg displacement. This follows from the following two relations:

$$\hat{L}(b_k, c_k)\hat{X}(m_i) = e^{\frac{i}{2} \chi b_k m_i^2} e^{\frac{i}{2} c_k m_i^2} \hat{Z}(m_i)\hat{X}(-c_k m_i - \chi b_k m_i^2) \hat{L}(b_k, c_k + 2\chi b_k m_i) \quad (1.26)$$

$$\hat{L}(b_k, c_k)\hat{Z}(m_i) = \hat{X}(-m_i)\hat{L}(b_k, c_k) \quad (1.27)$$

Though the first of these expressions seems a little complicated, it is merely an element of $\mathcal{W}_1$ preceded by a different $\hat{L}$ gate. Thus, we can use measurement-adaptivity just like in the qubit case (in this case, we adapt $c_k \mapsto c_k - 2\chi b_k m_i$) in order to implement the desired gate deterministically, up to a final known element of $\mathcal{W}_1$. Therefore, we have shown that a linear CVCS is a universal resource for single-mode CV quantum computation, implementing circuits of the form shown in Fig. 1.7(c).
Chapter 1: Introduction

To extend this to universal quantum computation, we require a way to implement entangling gates in a measurement-based fashion. Just like in the qubit case, we can implement two-mode gates by constructing a horseshoe network cluster state as shown in Figs. 1.7(d, e). The horizontal link of the graph implements a “teleported” $\hat{C}_Z(1)$ gate after a set of single-mode $\hat{L}$ gates:

$$e^{im_1m_2} \left[ (\hat{X}(m_1)\hat{Z}(m_2)) \otimes (\hat{X}(m_2)\hat{Z}(m_1)) \right] \hat{C}_Z(1) \left[ \hat{L}(b_1, c_1) \otimes \hat{L}(b_2, c_2) \right]$$

(1.28)

Notice again that the result depends on the measurement outcomes $m_1, m_2 \in \mathbb{R}$. Up to a Weyl-Heisenberg shift, this is an entangling gate that extends the single-mode operations described above to a universal gate set. The only remaining issue is how any unwanted Weyl-Heisenberg shifts that occur before this gate can be dealt with. First, they must be pulled through the single mode $\hat{L}$ gates. This can be achieved by adapting the measurement basis, as described above. Next, the $C_Z(1)$ gate is an element of the Gaussian unitary group, and therefore any Weyl-Heisenberg displacements immediately after the $\hat{L}$ gates can be commuted to the end, merely resulting in some final (potentially different) Weyl-Heisenberg shift.

These measurement-based gate circuits can be realised as subnetworks on a larger two-dimensional square lattice CV cluster state in an analogous way to the qubit case, as shown in Fig. 1.8. Universality of the square lattice CV cluster state with unit edge weights follows immediately from this fact.

Finally, measurement-readout of encoded modes within the cluster can be performed by single-site measurement in the $\hat{q}$ basis.

Thus, we have shown how to implement universal quantum computation using CVCS. One difference of between the CV and qubit cases is that we used a specific gate set in the CV case (we chose single-mode evolution generated specifically by cubic Hamiltonians in $\hat{q}$). Though different types of operations could have been chosen, basing the discussion on measurements of the form $\hat{p}(b, c)$ (allowing for gates of the form $\hat{L}(b, c)$) has the flavour of practicality: in the case of optical CVCSs, the $b = 0$ case can be implemented using just homodyne detection, and the $b = 1$ case can be implemented by first preparing $\hat{K}(\chi)|0\rangle_p$ offline (which can be achieved using cluster state machinery and photon counting measurements [65]), then replacing $|0\rangle_p$ at the relevant step to implement the gate via homodyne detection.
Chapter 1: Introduction

Figure 1.8: (a) Graphical representation of a square lattice continuous-variable cluster state. All edge weights are equal to one. Input states are encoded within the green nodes on the left edge. This state is universal because it enables universal single- and two-mode gates to be implemented via single-site measurements. We show this for a two-mode example circuit involving both single- and two-mode gates. The middle two rows of modes are measured in either the \( \hat{q} \) basis (the red nodes) or the \( \hat{p} \) basis. Up to local Weyl-Heisenberg gates, the cluster state after measurement is as shown in (b). (b) Reduced continuous-variable cluster state. The graph contains both wire and horseshoe graph substructures. Measuring the remaining modes implements the circuit shown in (c). (c) Basic two-mode circuit that contains both single- and two-mode gates. Circuits of this type can be implemented deterministically on a square lattice cluster state up to a known random Weyl-Heisenberg shift on each mode.

1.4.2 Fault tolerance with continuous-variable systems

Unlike in the qubit case, general continuous-variable quantum computations are not known to be fault tolerant. Known error correction techniques cannot even mitigate against small errors, which tend to build up at a faster rate than they can be corrected for during computations. This difference between qubit and CV computation has been made apparent in the context of stabilizer codes [23]. In the qubit case such codes are ubiquitous for quantum error correction [8, 41, 49, 50], whereas the CV generalizations of these codes [69, 70] (called Gaussian error-correcting codes) cannot enable fault-tolerant computations [71].

Though general CV computations are hindered by noise, it is still possible run fault-tolerant qubit-level computations on a CV quantum computer [17, 67, 72]. This involves embedding qubits within the larger Hilbert space belonging to some number of modes. The optimal choice of qubit encoding depends on many factors, including the experimental difficulty of generating the chosen qubit states, implementing gates, and performing measurement readout. We will now review a selection of qubit encodings.

Photonic encoding schemes take advantage of the natural discrete level structure provided by the Fock basis of each mode. For example, the dual-rail encoding used in the original scheme for efficient linear-optical quantum computation [17] identifies the qubit computational basis with the two-mode single-photon states \( \{ |0 \rangle \otimes |1 \rangle , |1 \rangle \otimes |0 \rangle \} \),
where $|i\rangle$ represents the $i$ photon Fock state. With this encoding, universal quantum computation can be achieved efficiently with linear-optics gates and photon-counting measurements [17]. By applying qubit quantum error correction directly, this scheme has been proven to be fault-tolerant with respect to standard sources of noise, such as photon loss [18, 52, 73–76]. In addition, photonic qubit encodings that make use of higher photon number Fock states can be employed to deal with multiple photon-loss events per mode [77] and more general noise processes such as photon addition and dephasing [78].

More generally, qubits can be defined with respect to any choice of orthogonal (or almost orthogonal) CV state wavefunctions. For instance, one can consider superpositions of quasi-orthogonal coherent states $|\alpha\rangle \propto \hat{X}(\sqrt{2}\Re \alpha)\hat{Z}(\sqrt{2}\Im \alpha)|0\rangle$. This type of encoding is called a cat code or a coherent state code [79]. The basic cat code uses $\{|\alpha\rangle, |\!-\!\alpha\rangle\}$ with $\alpha \in \mathbb{R}$ for computation basis states (which are orthogonal as $|\alpha| \to \infty$) and additionally only requires linear optics, homodyne detectors, and conditional photon counting measurements for universal quantum computation [80]. This basic encoding has been studied in combination with standard qubit quantum error correction [81–83]. Increasing the number of superpositions and varying the amplitude of the coherent states can improve the performance of this code with respect to typical noise models (such as photon loss) [84, 85].

Another important encoding is the Gottesman-Kitaev-Preskill (GKP) encoding for a single qubit in a single mode, which is designed to correct for small phase-space shift errors (such as $\hat{X}(\delta s), \hat{Z}(\delta t)$) [67, 86]. The qubit computational basis is identified with orthogonal superpositions of position eigenstates that are equally spaced in position space: $\{\sum_{j=-\infty}^{\infty} |\sqrt{\pi} j (2 + n)\rangle_q | n \in \{0, 1\}\} [67]$. Note that the qubit states have infinite energy (they are an infinite sum of states that are each infinitely squeezed in position), therefore they must be substituted with finite-energy approximate GKP states in practice [67]. This encoding has the appealing feature that Clifford gates require only Gaussian operations, which simplifies implementations of qubit-level quantum error correction [87].

Most of the qubit encodings mentioned above have only been analyzed in the context of the CV circuit model for quantum computation. An important difference in the CV measurement-based model arises due to the finitely squeezed nature of CVCSs. Even in the absence of experimental sources of noise, CV measurement-based quantum computations are limited in length by intrinsic noise due to finite squeezing effects (unless some form of active error correction is used) [60–62]. These finite squeezing effects can be modelled as a Gaussian Bosonic noise channel [88] whose strength depends on the amount of squeezing used to generate the CVCS [2, 65]. Despite this complication, fault-tolerant quantum computation with CVCSs is known to be possible with the GKP encoding if
the overall squeezing levels exceed 20.5 dB [87]. Though other qubit (or qudit) encodings have not yet been applied to CV measurement-based quantum computation, further investigation involving encodings suited dealing with photon loss noise is motivated by results presented later in Ch. 6, which show that in some CVCS computations the finite squeezing noise can be modelled as a photon loss channel.

In the next section, we briefly review the history of the developments in optical CVCS generation schemes.

1.4.3 A brief history of optical continuous-variable cluster states

The first proposed protocol for making physical CVCSs in optics is known as the canonical method. To generate CVCSs, the procedure is to start with squeezed vacuum states and then entangle them with $\hat{C}_Z(1)$ gates (which all commute) [65]. Optical implementation of the $\hat{C}_Z(1)$ gate can be achieved through linear optics (beamsplitters and phase shifts) and (online) squeezing elements [89]. However, the requirement of online squeezing is still experimentally challenging [90].

This difficulty was circumvented in the linear-optics method [91] by noting that the resource demands of the canonical optical circuit for CVCS generation can be simplified down to offline squeezing (preparing squeezed vacuum states), beamsplitters, and phase shifts [91]. However, the beamsplitters are not one-for-one with the $\hat{C}_Z(1)$ gates (and do not commute). Furthermore, changing the cluster state by even a single mode generally requires an entirely different beamsplitter network by this method.

For Gaussian measurement-based quantum computation, the linear optics method has been substantially improved by the possibility of implementing mode-transformations equivalent to the linear optics network via post-processing data from multi-pixel homodyne detectors [92], or by using local oscillator mode-reshaping [93]. The key benefits of these schemes is the compactness and the reconfigurability of the cluster state generation circuit, allowing for gate-tailored noise optimization strategies [94].

The linear optics cluster state generation procedure was greatly simplified by frequency-mode OPO techniques [54–56, 58, 95, 96]. All of the required squeezing and linear optics can take place inside an optical-parametric-oscillator (OPO), which consists of a laser-pumped nonlinear crystal within an optical cavity. The modes are defined using different frequencies of the cavity’s optical frequency comb. Removing the beamsplitter array sets demands on the nonlinear crystal and the frequency content of the pump beam [54–56]. Once these prerequisites can be achieved within the laboratory [96], this method offers excellent scalability: the pump beam complexity scales as a constant with
Chapter 1: Introduction

the number of modes. This scheme has been used to generate linear cluster states of the type considered in Ch. 3 consisting of 60 entangled frequency modes [96]. By using multiple OPOs, this method can be used to generate high-dimensional graph structures [58].

The single-QND-gate method [97] improves on the canonical method by using temporally encoded modes. Noting that an entire cluster state need not be generated all at once, a basic set of optical machinery can be used repeatedly to extend the state as required. This is the so-called “Wallace and Gromit” approach [97].

The temporal-mode linear-optics method [57] amalgamates both the single-QND-gate and frequency-mode OPO methods. It uses temporal modes and OPO machinery in order to generate large-scale cluster states with very compact experimental setups. The most recent demonstration generated CVCSs (of the kind considered in Ch. 3) composed of one-million entangled temporal modes [98].

Compact CVCS generation can also be achieved using Raman quantum memories [99]. This was the first proposal to make use of both time and frequency bins (known as a hybrid approach) within a single spatial mode, with the number of logical inputs scaling linearly with the number of frequencies.

Our results in Ch. 4 build on all of the above. It is a hybrid temporal/frequency modes approach that requires only a single-OPO in order to generate a universal resource state for CV measurement-based quantum computation.

1.5 Thesis structure

In this thesis we extend previous work on measurement-based quantum computation for qubits and CVs by introducing a collection of new protocols. Chapters 2-6 can each be read as a self-contained article. As a result, some of the introductory material has been repeated, and some of the definitions vary from chapter to chapter. Here we give a brief summary.

Chapter 2

Here we provide a review of randomized benchmarking and qubit measurement-based quantum computation. Then we provide two protocols based on the Clifford group and measurement-based 2-designs [100] for implementing measurement-based randomized benchmarking for a single-qubit.
Chapter 3

In this chapter we review continuous-variable measurement-based quantum computation on a linear cluster state known as the dual-rail quantum wire. We consider three modes of computation and explore the relation between graphical parameters of the state and noise due to finite squeezing.

Chapter 4

Here we introduce a new universal resource state for measurement-based quantum computation: the bilayered square lattice CV cluster state. We show that this method can be generated using a highly compact hybrid temporal/frequency mode setup involving only a single-OPO and a simple constant depth linear optics circuit. We also introduce a macronode-based measurement protocol, generalizing the scheme in chapter 3 to a two-dimensional resource.

Chapter 5

Here we consider a universal continuous-variable cluster state known as the quad-rail lattice. We show that this state exhibits a novel mode of measurement-based quantum computation, enabling greater circuit flexibility and compactness.

Chapter 6

Using the scheme introduced in chapter 5, we give a proposal for measurement-based linear optics. We show that the finite squeezing noise channel can be modified into an effective photon loss channel. We use this scheme to introduce a measurement-based BOSONSAMPLING device, and consider the squeezing resource requirements for CV cluster states.

Chapter 7

In this concluding chapter we summarize our results and suggest possible extensions for future work.
1.6 List of acronyms used in this thesis

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKLT state</td>
<td>Affleck-Kennedy-Lieb-Tasaki state</td>
</tr>
<tr>
<td>AWG</td>
<td>Arrayed waveguide grating</td>
</tr>
<tr>
<td>BS</td>
<td>Beamsplitter</td>
</tr>
<tr>
<td>BSL</td>
<td>Bilayer square lattice</td>
</tr>
<tr>
<td>CPTP map</td>
<td>Completely-positive trace-preserving map</td>
</tr>
<tr>
<td>CV</td>
<td>Continuous variable</td>
</tr>
<tr>
<td>CVCS</td>
<td>Continuous-variable cluster state</td>
</tr>
<tr>
<td>CVW</td>
<td>Continuous-variable quantum wire</td>
</tr>
<tr>
<td>CW</td>
<td>Continuous wave</td>
</tr>
<tr>
<td>DRW</td>
<td>Dual-rail quantum wire</td>
</tr>
<tr>
<td>EOM</td>
<td>Electro-optic modulator</td>
</tr>
<tr>
<td>EPR state</td>
<td>Einstein-Podolsky-Rosen state</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-programmable gate array</td>
</tr>
<tr>
<td>HWP@θ</td>
<td>Half wave place at angle θ</td>
</tr>
<tr>
<td>LO</td>
<td>Local oscillator</td>
</tr>
<tr>
<td>LOQC</td>
<td>Linear optical quantum computing</td>
</tr>
<tr>
<td>MBLO</td>
<td>Measurement-based linear optics</td>
</tr>
<tr>
<td>MBMZ</td>
<td>Measurement-based Mach-Zehnder</td>
</tr>
<tr>
<td>MBQC</td>
<td>Measurement-based quantum computation</td>
</tr>
<tr>
<td>MZI</td>
<td>Mach-Zehnder interferometer</td>
</tr>
<tr>
<td>OPO</td>
<td>Optical parametric oscillator</td>
</tr>
<tr>
<td>PBS</td>
<td>Polarized beamsplitter</td>
</tr>
<tr>
<td>QND gate</td>
<td>Quantum non-demolition gate</td>
</tr>
<tr>
<td>QRL</td>
<td>Quad-rail lattice</td>
</tr>
<tr>
<td>RB</td>
<td>Randomized benchmarking</td>
</tr>
</tbody>
</table>


Chapter 1: Introduction

**SPAM errors** State preparation and measurement errors

**TMS state** two-mode squeezed state

**VIPA** Virtually-imaged phase arrays
Chapter 2

Randomized benchmarking in measurement-based quantum computing

2.1 Abstract

Randomized benchmarking is routinely used as an efficient method for characterizing the performance of sets of elementary logic gates in small quantum devices. In the measurement-based model of quantum computation, logic gates are implemented via single-site measurements on a fixed universal resource state. Here we adapt the randomized benchmarking protocol for a single qubit to a linear cluster state computation, which provides partial, yet efficient characterization of the noise associated with the target gate set. Applying randomized benchmarking to measurement-based quantum computation exhibits an interesting interplay between the inherent randomness associated with logic gates in the measurement-based model and the random gate sequences used in benchmarking. We consider two different approaches: the first makes use of the standard single-qubit Clifford group, while the second uses recently introduced (non-Clifford) measurement-based 2-designs, which harness inherent randomness to implement gate sequences.

2.2 Introduction

In the measurement-based model [9], quantum computation proceeds via adaptive single-site measurements on an entangled resource state of many qubits such as the cluster...
Chapter 2: Randomized benchmarking in measurement-based quantum computing

state [40]. The computational power of this model is equivalent to standard approaches to universal fault-tolerant quantum computation, assuming that all operations can be implemented with sufficiently small error [53, 101, 102]. Because this model does not require an on-demand entangling gate, it is appealing for candidate physical architectures where such gates cannot be performed deterministically. The leading example is linear optical quantum computing (LOQC) [17, 18, 103], where the basic building blocks are single-photon sources, linear optics, and photon-number resolving detectors with feedforward.

As quantum devices with progressively smaller error rates are developed, there is a growing need for techniques to efficiently characterize the noise associated with elementary components such as logic gates. Although it may sound desirable, a complete description of the error processes of a quantum device is prohibitively expensive due to the exponentially bad scaling in size [73, 104]. An additional concern is how to observe gate errors in the presence of noise from state preparation and measurement (SPAM), which often dominate. The randomized benchmarking (RB) protocol [105–107] is a technique that allows for efficient, partial characterization of a target gate set whilst being insensitive to noise from SPAM [108].

Randomized benchmarking performs well with realistic noise using only small data sets [108–110]. The basic protocol has been extended to include tests for time-dependence, non-Markovianity [108, 109, 111, 112], robustness to leakage errors [113], reconstruction of the unital part of general completely-positive trace-preserving (CPTP) maps [114], and extracting tomographic data from quantum gates [115].

Here we adapt the original RB protocol to the setting of measurement-based quantum computation (MBQC). By combining ideas from RB and MBQC on linear cluster states, we provide two protocols for estimating the average gate fidelity for two different single-qubit gate sets. The first gate set is the single-qubit Clifford group, and the second is the recently proposed measurement-based exact 2-design [100], which leverages the intrinsic randomness of MBQC to implement random sequences of gates. Our schemes fully inherit the advantages of the RB protocol, namely that the average gate fidelity can be computed efficiently and with low sensitivity to errors in preparation of the (logical) input and final measurement readout [107].

We review the RB protocol and MBQC in Sec. 2.3. We discuss our protocols for implementing RB on a linear cluster state with the Clifford group and with the measurement-based 2-designs in Sec. 2.4.
2.3 Background

Here we review the standard RB protocol [107], and fix our notation.

2.3.1 Preliminaries

Consider a \( d = 2^n \) dimensional Hilbert space \( (\mathbb{C}^2)^\otimes n \) corresponding to an \( n \)-qubit system. A unitary operation (gate) is denoted by \( U \); the corresponding superoperator that acts on density matrices \( \rho \) is denoted by \( U(\rho) = U\rho U^\dagger \). We denote \( U^\dagger(\rho) = U^\dagger \rho U \) and \( U^m(\rho) = U^m \rho U^m\dagger \). General (non-unitary) superoperators are denoted \( D, E \), etc., and in addition we use \( \tilde{U} \) to denote a noisy approximation to the ideal unitary gate \( U \). Common unitary gates we will see include the \( X, Y, \) and \( Z \) Pauli matrices, the Hadamard gate \( H \), the \( CZ \) gate \( (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z) \) and single-qubit \( Z \) rotations by \( \theta \), \( Z_\theta = e^{-i\theta Z/2} \).

We will make use of the Clifford phase gate \( P := Z_{\pi/2} \). Here we use “\( \circ \)” to denote channel composition and “\( \circlearrowright \)” for right-to-left sequential composition of channels, i.e., \( \bigcirc_{i=1}^n E_i(\rho) := E_n \circ \cdots \circ E_1(\rho) \).

Quantum states \( \rho_1, \rho_2 \) are commonly compared by their fidelity \( F \), given by

\[
F(\rho_1, \rho_2) = \left( \text{tr} \sqrt{\sqrt{\rho_1} \rho_2 \sqrt{\rho_1}} \right)^2. \tag{2.1}
\]

This definition also allows for comparisons between two quantum gates \( E_1, E_2 \). The gate fidelity between these two gates is defined to be

\[
F(E_1, E_2) = \int d\psi F(E_1(\psi), E_2(\psi)), \tag{2.2}
\]

where the integral is over the set of all pure states with respect to the uniform measure \( d\psi \).

For a noisy implementation \( \tilde{U} \) of an ideal unitary gate \( U \), the gate fidelity \( F(\tilde{U}, U) \) gives a measure of (one minus) the average case error rate of the gate. While the gate fidelity is a measure of the average case error, in many applications—such as computing thresholds for fault tolerance—the worst case error is the relevant figure of merit [116] (quantified, for example, by the diamond norm distance between the ideal and noisy gates). The gate fidelity can be used to bound the worst case error rate [117–119].

Let \( G = \{U_r, r = 1, 2, \ldots, |G|\} \) be a set of ideal (unitary) gates. For each \( U_r \in G \), let \( \mathcal{U}_r \) be the ideal unitary gate as a superoperator and \( \tilde{\mathcal{U}}_r \) be a noisy approximation to this
gate. The average gate fidelity for the gate set \( G \), denoted \( F_G \), is defined to be

\[
F_G = \frac{1}{|G|} \sum_{r=1}^{|G|} F(\tilde{U}_r, U_r).
\]

The RB protocol allows us to characterize the experimental implementation of a gate set \( G \) by estimating the value of \( F_G \), provided that \( G \) forms a unitary 2-design:

**Definition (2-design):** A set of unitary gates \( G = \{U_r\} \) is a unitary 2-design if for any quantum channel \( \mathcal{E} \), the action of the twirl of \( \mathcal{E} \) over \( G \) on an arbitrary state \( \rho \) is equivalent to that of the twirl over the entire \( n \)-qubit unitary group \([120, 121]\)

\[
\int dU U^\dagger \circ \mathcal{E} \circ U(\rho) = \frac{1}{|G|} \sum_{r=1}^{|G|} U_r^\dagger \circ \mathcal{E} \circ U_r(\rho),
\]

where \( dU \) is the uniform (Haar) measure.

For \( n \) qubits, a commonly used 2-design is the \( n \)-qubit Clifford group \([122–124]\).

### 2.3.2 Randomized benchmarking

We now briefly review RB together with a derivation (originally due to Magesan *et al.* [107]) of how RB yields an estimate of the average gate fidelity. In our review of this derivation, we relax the condition that the 2-design have a group structure. This relaxation will be important when we consider RB in the MBQC case, which will make use of non-Clifford 2-designs.

The standard RB protocol proceeds as follows. Choose a set of unitary gates \( G \) that forms a unitary 2-design, and for which the inverse element of any sequence of gates can be efficiently computed. Choose a sequence length \( s \), and a number \( K_s \) of gate sequences for that length. Draw \( K_s \) many sequences of \( s \) gates from \( G \) uniformly at random. For the \( i \)-th sequence, \( 1 \leq i \leq K_s \), denote the \( j \)-th element of the sequence by \( U_j^{(i)} \), where \( 1 \leq j \leq s \). Note that each \( U_j^{(i)} \) is an element \( U_r \in G \) from the gate set. For each sequence, compute \( U_{s+1}^{(i)} = (U_s^{(i)} U_{s-1}^{(i)} \cdots U_1^{(i)})^\dagger \). Note that when \( G \) does not form a group, \( U_{s+1}^{(i)} \notin G \) in general.

Although the sequences are ideally described by noiseless unitary gates \( U_r \) sampled from \( G \), in practice these gates will be noisy. The noisy gates \( \tilde{U}_j^{(i)} \) can be decomposed into a composition of the ideal unitary gate \( U_j^{(i)} \) followed by an arbitrary CPTP map \( \mathcal{D}_j^{(i)} \),
i.e., the noisy gate is described by
\[ \tilde{U}_j^{(i)}(\rho) = D_j^{(i)} \circ \mathcal{U}_j^{(i)}(\rho). \] (2.5)

Let \( \tilde{\psi} \) denote the mixed state describing the noisy preparation of the ideal state \( \psi := |\psi\rangle \langle \psi| \). The total noisy evolution of this state under the \( i \)-th sequence is then
\[ \tilde{U}^{(i)}(\tilde{\psi}) := \bigotimes_{j=1}^{s+1} \tilde{U}_j^{(i)}(\tilde{\psi}) = \bigotimes_{j=1}^{s+1} [D_j^{(i)} \circ \mathcal{U}_j^{(i)}](\tilde{\psi}). \] (2.6)

At the conclusion of the sequence, a measurement described by the effects \{\( \tilde{E}_\psi, \mathbb{I} - \tilde{E}_\psi \)\} is performed, which is the noisy implementation of the ideal projective measurement \{\( |\psi\rangle \langle \psi|, \mathbb{I} - |\psi\rangle \langle \psi| \)\}. This measurement gives what is known as the survival probability for the sequence \( i \),
\[ \text{tr} \left[ \tilde{E}_\psi \tilde{U}^{(i)}(\tilde{\psi}) \right]. \] (2.7)

Its average over all \( K_s \) random sequences \( \mathcal{U}^{(i)} \) results in the sequence fidelity
\[ F_G(s, K_s) := \frac{1}{K_s} \sum_{i=1}^{K_s} \text{tr} \left[ \tilde{E}_\psi \tilde{U}^{(i)}(\tilde{\psi}) \right]. \] (2.8)

This can be viewed as an estimate of the average defined by the set of all sequences of length \( s \). As the number of sequences \( K_s \) increases, the sequence fidelity converges to the uniform average over all sequences
\[ F_G(s) = \frac{1}{|G|^s} \sum_{i=1}^{|G|^s} \text{tr} \left[ \tilde{E}_\psi \tilde{U}^{(i)}(\tilde{\psi}) \right], \] (2.9)
where there are a total of \( |G|^s \) sequences, and each sequence \( i \) is taken with equal weight in order to satisfy the 2-design condition. A key feature of RB is that it scales well in both the number of qubits and the sequence length \( s \). This is due to the fact that \( F_G(s, K_s) \) converges quickly to \( F_G(s) \) in the number of sequences measured \( K_s \) [107, 108].

Estimating \( F_G(s) \) for various sequence lengths \( s \) can be used to produce an approximation to the average gate fidelity \( \bar{F}_G \). The original derivation [107] is reviewed in Appendix 2.6, but presented without the assumption that \( G \) is a group. This derivation yields an exponential decay of the sequence fidelity as a function of \( s \), of the form
\[ F_G(s) \approx A_0 (2\bar{F}_G - 1)^s + B_0 \] (2.10)
where \( A_0 \) and \( B_0 \) are nuisance parameters that contain information about the noise in state preparation and measurement; see Appendix 2.6. Equation (2.10) is known as the
Chapter 2: Randomized benchmarking in measurement-based quantum computing

0th order expansion of $F_G(s)$. By performing the RB protocol above for various $s$, we can fit the 0th order model to the measurement data to find $\bar{F}_G \ [110]$.

Key assumptions in this derivation were that the noise per gate when decomposed as in Eq. (2.5) is Markovian and that it has low dependence on which gate was being applied, as well as on time, i.e., $D^{(i)}_j \simeq D$ independent of $i, j$. It was shown in Ref. [107] that in this regime, the effect of including gate dependent perturbations to the noise $D$ can be neglected for the purposes of calculating the average gate fidelity. Note that these assumptions are sufficient but not necessary – we will impose them later in Sec. 2.4.1.1 to establish a regime under which the 0th order model of RB is guaranteed to be valid.

In the case when $G$ is not a group, then the final sequence inverse $U^{-1}_{s+1}$ may not be an element of $G$ (and, perhaps, instead performed by changing the measurement basis). To directly extend the proof by Magesan et al. [107] to such cases, we further assume that the noise superoperator $D^{(i)}_{s+1}$ corresponding to the sequence inverse (or final measurement) is independent of the choice of sequence.

2.3.3 Measurement-based quantum computation

We now briefly review the measurement-based model for quantum computation, with a focus on the aspects that will be used in designing an RB protocol within this model.

In the measurement-based model, the task of building a quantum computer is broken into two steps: 1) prepare a cluster state [40] with a suitable graph structure (e.g., a linear chain for single-qubit gates, or a square lattice for universal quantum computation); 2) Perform single-qubit measurements on this resource, allowing for future measurement bases to be adaptively changed conditioned on past measurement outcomes [125].

We focus our attention on linear cluster states, which allow for sequences of single-qubit gates in the MBQC model. A linear cluster state is defined on $n$-qubits with a single-qubit input $\psi$ as:

$$\bigotimes_{i=1}^{n-1} C_{Z,i,i+1}(\psi \otimes |+\rangle \langle +| \otimes |^{n-1})$$

(2.11)

where $C_Z$ gates are applied to nearest neighbors with respect the linear graph shown in Fig. 2.1(a).

Quantum computation proceeds via single-qubit measurements in the $XY$-plane of the qubit as shown in Fig. 2.1(b). Measuring the first cluster site in this way and obtaining the outcome $m \in \{0, 1\}$ implements the logic gate

$$\mathcal{M}_{\theta,m}(\psi) := X^m \circ \mathcal{H} \circ Z_\theta(\psi),$$

(2.12)
Figure 2.1: (a) Cluster state with wire graph (shown left) with an input state on the leftmost and green node. Measurement angles are labeled in the center of each node. (b) Measuring the input state in some basis in the $XY$-plane (i.e., a measurement in the eigenbasis of $X_{\theta_1} = X \cos \theta_1 - Y \sin \theta_1$) yields the output shown in the circuit on the right.

where we recall that $X$ is the superoperator describing the unitary Pauli $X$ gate, $Z_{\theta}$ describes a rotation by $\theta$ about the $z$-axis, and $H$ is the Hadamard gate $H$.

In the absence of noise, both measurement outcomes are equally probable. Furthermore, the outputs only differ up to a Pauli $X$ correction, i.e., $M_{\theta,1} = X \circ M_{\theta,0}$.

Though the gate that gets implemented after each measurement step is probabilistic (either $M_{\theta,0}$ or $M_{\theta,1}$), the overall unitary evolution due to several sequential measurements can still be made deterministic up to a known Pauli gate by using measurement feedforward—i.e., introducing a time ordering to the measurements and allowing the choice of the future measurement bases to depend on the outcomes of prior qubit measurements [125, 126].

An important exception is when $\theta_i$ is an integer multiple of $\pi/2 \ \forall i$. In this case the gates are Clifford and changing the measurement angle is equivalent to flipping the measurement outcome in post-processing, i.e., $M_{n \frac{\pi}{2},0} = M_{-n \frac{\pi}{2},1}$ for some integer $n$. The measurement angles do not need to be chosen adaptively based on previous measurement outcomes (as is typically required for non-Clifford gates in MBQC), and so all such measurements can be performed simultaneously. The final Pauli gate can be absorbed into the final measurement process.

### 2.4 Randomized benchmarking in MBQC

In this section we first give the basics of implementing RB on a linear cluster state and then we outline two schemes that use different 2-designs.
2.4.1 RB within the measurement-based model

For some 2-design $G$, each sequence of gates $U_r \in G$ will be implemented by measurements on a linear cluster state. We will analyze the use of specific gate sets in Secs. 2.4.2 and 2.4.3, but first we present an analysis of how RB schemes are generally performed in MBQC, focusing on how the expected gate noise matches the noise assumptions imposed in the RB proof.

Throughout, we assume that the same fixed number of measurements $q$ are used for all gates in $G$. (In general, each $U_r \in G$ may require a different number of measurements to be implemented.) For example, any single-qubit gate can be implemented by MBQC using $q = 3$ measurements on a linear cluster state [9]. As described by Eq. (2.12), the randomness of the measurement outcomes means that the logic gates performed in this way will not be deterministic and will depend on the measurement outcome. The required total length of the linear cluster state is $(s + 1)q + 1$. (If instead the sequence inverse is incorporated into the final measurement then only a $(sq + 1)$-long cluster state is required per run.)

2.4.1.1 Noise in MBQC logic gates

Noisy cluster state preparation, storage, and measurement will translate into an effective noise channel per gate as the measurement-based computation proceeds. The RB noise assumptions require that the errors on the cluster state be local so that gate noise from measuring different cluster qubits is uncorrelated. When the noise is modeled as in the circuit shown in Fig. 2.2, Markovian noise in state preparations, entangling gates and measurements results in an effective Markovian noise channel per gate.

Now consider decomposing the noisy logical gate $\tilde{U}_j^{(i)}$ as a sequence of $q$ measurements followed by a noise map, as

$$\tilde{U}_j^{(i)}(\theta, m) = D_{\text{seq}(q)} \circ \bigodot_{k=1}^{q} M_{\theta_k, m_k}$$

(2.13)

where $D_{\text{seq}(q)}$ is some total noise channel after an ideal gate $\bigodot_{k=1}^{q} M_{\theta_k, m_k} = \tilde{U}_j^{(i)}(\theta, m)$, and we include a dependence on $\theta = (\theta_1, \ldots, \theta_q)$ and $m = (m_1, \ldots, m_q)$. The noise assumptions also require that $D_{\text{seq}(q)}$ be independent of time and the gate being implemented. The validity of these assumptions will depend on the relevant noise sources for cluster state preparation and measurement.
Chapter 2: Randomized benchmarking in measurement-based quantum computing

2.4.1.2 SPAM errors

As mentioned above, errors that occur in the preparation of the cluster state can lead to logical errors in the MBQC logic gates. In addition to these gate errors, MBQC will also have logical state preparation and measurement (SPAM) errors. While the logic gates in MBQC can be robustly protected from many forms of errors by symmetry arguments [29, 129–131], this is not generally true for SPAM errors and so these can be expected to dominate in MBQC as they do in many other implementations of quantum computing. Nonetheless, for the purposes of RB, a natural choice of input state is \( \psi = |+\rangle |+\rangle \), which is automatically “encoded” on the edge of the linear cluster state when prepared as in Fig. 2.1. After the inverse operation \( U^{(s+1)} \) is applied, the final measurement is in the \( X \) basis.

2.4.2 RB using the Clifford group

Here we discuss the first of our protocols for measurement-based RB, referred to as Clifford RB. The distinguishing feature of this scheme is that it uses the single-qubit Clifford group \( \mathbb{C}_1 \) as the set \( \mathcal{G} \) of logic gates. The Clifford group forms a unitary 2-design.

We set the number of measurements per logic gate to be \( q = 3 \), as this is the maximum number of measurements required to implement all arbitrary single-qubit Clifford gates. Note that this protocol can be straightforwardly extended to any \( q \geq 3 \) by using more measurements per gate. The basic building block of our scheme is the three node cluster wire shown in Fig. 2.3.

Using the Clifford group simplifies the experimental setup as all measurement devices need only to be programmed to measure in either the Pauli \( X \) or \( Y \) basis since the measurement angles are all integer multiples of \( \frac{\pi}{2} \) (see Appendix 2.7 for gate-to-measurement
conclusion table). Furthermore, we do not need to make use of measurement feedforward as the gate implemented can only differ from the desired case (e.g., \(m_i = 0, \forall i\)) by a known Pauli gate, as

\[
U_j^{(i)}(\theta, \mathbf{m}) = \bigotimes_{k=1}^3 M_{\frac{\pi}{2} n_k, m_k} = X^{b_1} \circ Z^{b_2} \circ \left( \bigotimes_{k=1}^3 M_{\frac{\pi}{2} n_k, 0} \right), \tag{2.14}
\]

where

\[
b_1 = m_3 + m_2 n_3 + m_1 (n_2 n_3 + 1) \tag{2.15}
\]

\[
b_2 = m_2 + m_1 n_2. \tag{2.16}
\]

As a consequence, each sequence can be measured simultaneously in a single time step on a linear cluster state.

The protocol begins by generating a sequence \(U^{(i)}\) of length \(s\) from \(\mathcal{C}_1\) uniformly at random. The inverse is computed in the case where all measurement outcomes are assumed to be zero. The corresponding measurements are made on a length \(3s + 4\) linear cluster state, with the final qubit measured in the Pauli \(X\) basis. Repeating this process \(K_s\) times and over different sequence lengths \(s\) yields an estimate for the survival probability, from which \(\bar{F}_{\mathcal{C}_1}\) can be extracted for gates implemented via 3 measurements.

From Eq. (2.14), each \(U_j^{(i)}(\theta, \mathbf{m})\) is only implemented up to a random Pauli gate, i.e., the actual gate implemented with outcomes \(\mathbf{m} = \{m_1, \ldots, m_q\}\) is

\[
U_j^{(i)}(\theta, \mathbf{m}) \in \{IU_j^{(i)}(\theta, 0), XU_j^{(i)}(\theta, 0), YU_j^{(i)}(\theta, 0), ZU_j^{(i)}(\theta, 0)\}. \tag{2.17}
\]

So long as the angles are chosen uniformly at random from the Clifford table in Appendix 2.7, the gate implemented will also be uniformly random, irrespective of the measurement outcomes. Thus, the indeterminism of the logic gates does not interfere with this measurement-based RB protocol.

Next we consider the case when the probability of getting a 0 or 1 for each measurement is equally likely. In particular, we show how this results in a simplification of the original Clifford RB protocol.

### 2.4.2.1 The role of randomness

In the above, the protocol required random sequences of Clifford gates. However, as a result of the indeterminacy of the measurement outcomes, each chosen sequence can
result in one of $4^3$ possible sequences occurring. Thus, much of the randomness required by the above protocol is redundant.

When each cluster state measurement yields outcome 0 or 1 with equal probability, the scheme can be simplified. Note that $C_1$ can be factored into right cosets of its Pauli subgroup (ignoring phases) $P_1 = \{I, X, Y, Z\}$, i.e., $C_1 = \cup_{g \in T_1} P_1 g$, where $T_1 := \{I, P, H, PH, HP, PHP\}$. As a result, a random sequence of $C_1$ elements can be implemented by initializing the above protocol with only a random sequence of elements of $T_1$. The larger Clifford group ($|C_1| = 24$) is generated uniformly from $T_1$ by the additional random Pauli gate provided that the measurement outcomes are themselves distributed uniformly.

In general, noise on the cluster state will mean that measurement outcomes may not occur with equal probability. In such cases, the full Clifford RB protocol (selecting from $C_1$ rather than $T_1$ at random) can be used.

Alternatively, we can restore uniformity into the measurement outcome distributions to deal with problematic noise channels. (The alternative measurement-based RB protocol presented in the next section requires uniformly distributed measurement outcomes.)

The basic idea is to inject additional randomness into MBQC in order to restore uniformity in outcomes. At each measurement step $k$, we introduce a uniformly random binary variable $c_k$. When $c_k = 1$, the measurement outcome is flipped, i.e., $m_k \mapsto m_k + 1 \mod 2$, and otherwise it is left alone. We have effectively defined new measurement outcome variables $m'_k := c_k + m_k \mod 2$. This is equivalent to applying a perfect Pauli Z on the cluster qubit $k$ prior to measurement, or alternatively, locally swapping the definitions of the positive and negative $X$ axes. The effective measurement variable $m'_k$ is a uniformly random binary variable. In order to use this for MBQC, the feedforward...
procedure must be adjusted accordingly. We also note that this trick will modify the effective noise channel.

Basing this scheme on the Clifford group should allow for generalization to the multi-qubit setting while preserving the advantages discussed above. For instance, with a universal cluster state (say, on a 2D square lattice), then Clifford circuits can still be implemented in a single time step as there is no need for active feedforward. When the measurement outcomes are uniformly distributed, random elements of the \( n \)-qubit Clifford group \( C_n \) can be generated by implementing a random element of \( T_n \)—a set containing one element from each coset of the \( n \)-qubit Pauli group \( P_n \) in \( C_n \). As in the single-qubit case, each element of \( T_n \) will be implemented along with a random Pauli, generating the full Clifford group if the measurement outcomes are uniformly random. Also, the inverse element of an \( n \)-qubit RB scheme can always be efficiently computed as a consequence of the Gottesman-Knill theorem [132].

### 2.4.3 RB using derandomized 2-designs

As we saw in the previous protocol, the intrinsic randomness of MBQC can be leveraged to simplify the implementation. We now consider an alternative to the single-qubit Clifford group that extends this idea further: by using recently proposed measurement-based unitary 2-designs from Ref. [100], RB can be performed using a single, fixed set of measurements and relying entirely on the measurement randomness to implement random gate sequences. We refer to this protocol as derandomized RB, and it allows for the characterization of more general non-Clifford logic gates in the MBQC model.

As with Clifford RB, this scheme does not use any feedforward. When a linear cluster state is measured with a repeating pattern of \( q \) fixed measurement bases, each set of \( q \) measurements can generate up to \( 2^q \) distinct unitary evolutions.

A necessary ingredient of this scheme is that some of the cluster qubits be measured in bases other than integer multiples of \( \pi/2 \). Otherwise, the implemented gates will only differ by a known Pauli gate (as in Eq. (2.14)). As \( P_1 \) is only a unitary 1-design, so too is the entire gate set, and it is insufficient for RB.

As shown in Ref. [100], a family of 2-designs can be generated using cluster states of various lengths. Here we consider the simplest case: a \( q = 5 \) sequence with measurement bases corresponding to angles

\[
(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = \left( \phi_1, \frac{\pi}{4}, \cos^{-1}\left(\frac{1}{\sqrt{3}}\right), \frac{\pi}{4}, \phi_2 \right).
\]  

(2.18)
Each element of the unitary 2-design is implemented by making five fixed measurements on the cluster wire. The noise operator per 2-design element $D_{\text{seq}(5)}$ described the noise added after 5 measurements.

The resulting gate set $G$ implements a unitary 2-design provided that the measurement outcomes are all equally probable. Note that $\phi_1$ and $\phi_2$ are free parameters, which we set equal to zero for simplicity.

In the absence of noise, the gate applied is

$$U_j^{(i)}(m) = \bigotimes_{k=1}^{5} M_{\theta_k, m_k}$$

$$= \bigotimes_{k=1}^{5} X^{m_k} \circ H \circ Z_{\theta_k}.$$  \hspace{1cm} (2.19)

Define the gate applied when all measurement outcomes are zero as

$$Q := \bigotimes_{k=1}^{5} M_{\theta_k, 0}.$$ \hspace{1cm} (2.20)

Commuting each factor of $H \circ Z_{\theta_k}$ to the right in Eq. (2.19) we get

$$U_j^{(i)}(m) = A_5^{m_5} \circ A_4^{m_4} \circ A_3^{m_3} \circ A_2^{m_2} \circ A_1^{m_1} \circ Q$$ \hspace{1cm} (2.21)

where

$$A_i = \bigotimes_{k=1}^{6-i} \left( H \circ Z_{\theta_k} \right) \circ Z \circ \left( H \circ Z_{\theta_k} \right)^\dagger.$$ \hspace{1cm} (2.22)

Note that each $A_i$ is a $\pi$ rotation about some axis. These are expressed as $2 \times 2$ matrices in Appendix 2.8. Therefore, the structure of each $G$ element is a fixed unitary $Q$, followed by a sequence of $\pi$ flips, which (by construction [100]) must each be applied with probability 1/2. If noise in the state preparation or measurement results in a non-uniform probability distribution of measurement outcomes, then a strategy such as the one detailed in Sec. 2.4.2.1 should be used to restore uniformity.

To use this unitary 2-design to implement a sequence of $s$ elements for RB, the sequence of measurements in Eq. (2.18) is repeated $s$ times on a length-$(5s + 1)$ linear cluster.
Chapter 2: Randomized benchmarking in measurement-based quantum computing

state. The basic idea of this RB scheme is shown in Fig. 2.4. We will assume that the inverse is applied via a rotated qubit measurement on the last cluster qubit.

In this scheme, the sequence of random gates is generated by the indeterminacy of the measurement outcomes. As a result, the inverse element is not known a priori. To determine the sequence inverse, the input state’s evolution must be dynamically tracked. It is well known that the evolution of the state of a single qubit can be efficiently simulated classically [126].

A key advantage of this scheme’s use for RB is that it uses a fixed repeating pattern of measurement angles. This could simplify experimental implementations, as the setup would not have to be substantially changed between different sequences. This could also reduce noise introduced by the control in cases where sequences of gates are actively randomized. We note that some randomness may still need to be injected to restore uniformity in the measurement outcomes as a result of noise, as discussed in Sec. 2.4.2.1.

2.5 Conclusion

As we have shown, the basic machinery of randomized benchmarking can be translated into measurement-based quantum computation. Rather than interfering with the randomized benchmarking protocol, the intrinsic randomness of measurement-based quantum computation can be used to simplify it by partially (as in Clifford RB) or completely (as in derandomized RB) eliminating the need for drawing a random sequence of gates before each run. This work aims to establish a connection between advances in large scale cluster state generation [128, 133–135] and theoretical results for characterizing low-noise quantum devices.

For the benchmarking of gates beyond single-qubit operations, Clifford RB generalizes naturally to the 2D square-lattice cluster state, on which the entire multi-qubit Clifford group can be implemented on arbitrary inputs via single-site measurements with angles that are integer multiples of \( \pi/2 \). Feedforward could still be performed entirely in post processing, and so a sequence of gates can still be implemented by performing all measurements on the cluster state simultaneously. It is known that derandomized measurement patterns can produce approximate \( t \)-designs in the multi-qubit case [100], however the existence of exact multi-qubit measurement based designs is an open question. This work further motivates the search for such ensembles.

An important generalization of this work would be to characterize the validity of RB under more realistic noise sources. Such an extension could potentially make use of higher
order expansions of the derivation by Magesan et al. [107] in order to deal with gate-dependent noise sources. An extension highly relevant to linear-optical implementations would be to find a way to deal with photon loss [18], which is a non-Markovian (leakage) process. Dealing with this kind of noise is beyond the scope of our protocol, although recent theoretical developments have shown that the RB protocol can be adapted to such cases [109, 113].

Another possible extension of this work could be to consider alternative gatesets $G$ that are suitable for RB and can be conveniently implemented via MBQC. For instance, the dihedral RB protocol in Ref. [136] requires rotations about the $Z$ axis and bit flips ($X$). Within the measurement-based model on a linear cluster state, this can be straightforwardly implemented using two measurement steps per gate, where the gate specifies the angle on odd qubits and all even qubits are measured in the $X$ basis. We leave a more detailed analysis to future work.

Our work has also focused exclusively on cluster states as the resource for MBQC. Another generalization would be to develop RB schemes for alternative resource states such as the Affleck-Kennedy-Lieb-Tasaki (AKLT) state [137–139].

2.6 Appendix: Average gate fidelity derivation

Here we show how estimating $F_G(s)$ is related to $\bar{F}_G$. In contrast to the original derivation in Ref. [107], we will do so without assuming $G$ is a group.

By definition,

$$F_G(s) = \frac{1}{|G|^s} \sum_{i=1}^{|G|^s} \text{tr} \left[ \tilde{E}_\psi \circ \left( \bigcirc_{j=1}^{s+1} D_j^{(i)} \circ U_j^{(i)} \right) (\tilde{\psi}) \right]$$

(2.23)

First, we will assume we are working in a regime where the noise has little gate and time dependence. The $0^{th}$ order approximation in RB makes the assumption that $D_j^{(i)} \approx D, \forall i, j$ with $j \leq s$, which is a good approximation in the limit of low gate and time dependence on the noise [107]. It also requires that $D^{(i)}_{s+1}$ is independent of the choice of gate sequence $U^{(i)}$, i.e., $D^{(i)}_{s+1} \approx D^{\text{inv}}, \forall i$. Note that this is automatically satisfied when $G$ is a group (and therefore, closed under inverses) by extending the previous assumption to include $j = s + 1$. The sum in Eq. (2.23) is over all length $s$ sequences of gates from $G$, therefore it can be broken down into $s$ independent sums as follows

$$F_G(s) = \frac{1}{|G|^s} \sum_{r_s=1}^{|G|} \cdots \sum_{r_1=1}^{|G|} \text{tr} \left[ \tilde{E}_\psi D^{\text{inv}} \circ \left( \bigcirc_{i=1}^{s} U_{r_i} \right) \circ \left( \bigcirc_{j=1}^{s} D \circ U_{r_j} \right) (\tilde{\psi}) \right].$$

(2.24)
Next, we need to make use of the following: if we twirl a channel $\mathcal{D}$ with the unitary 2-design $G$, then we get a depolarizing channel $\mathcal{D}_D(\rho) = p\rho + (1-p)\frac{1}{2}I$ \cite{114, 120}. That is:

$$
\frac{1}{|G|} \sum_{r=1}^{|G|} \mathcal{U}_r \circ \mathcal{D} \circ \mathcal{U}_r^\dagger(\rho) = \mathcal{D}_D(\rho), \forall \rho. \tag{2.25}
$$

Crucially, this depolarizing channel has the same average fidelity as the original channel \cite{107}, i.e., $F(\mathcal{D}_D, I) = F(\mathcal{D}, I)$. Note also that $\mathcal{U} \circ \mathcal{D}_D(\rho) = \mathcal{D}_D \circ \mathcal{U}(\rho)$ for all unitary channels $\mathcal{U}$. Then the sums implement independent twirls over the first $s$ noise channels $\mathcal{D}$:

$$
\frac{1}{|G|^s} \sum_{r_s=1}^{|G|} \cdots \sum_{r_1=1}^{|G|} \mathcal{D}_\text{inv} \circ \left( \bigotimes_{i=1}^s \mathcal{U}_{r_i} \right)^\dagger \circ \bigotimes_{j=1}^s \left[ \mathcal{D} \circ \mathcal{U}_{r_j} \right] (\tilde{\psi}) \tag{2.26}
$$

$$
= \frac{1}{|G|^{s-1}} \sum_{r_{s-1}=1}^{|G|} \cdots \sum_{r_1=1}^{|G|} \mathcal{D}_\text{inv} \circ \mathcal{D} \circ \left( \bigotimes_{i=1}^{s-1} \mathcal{U}_{r_i} \right)^\dagger \circ \bigotimes_{j=1}^{s-1} \left[ \mathcal{D} \circ \mathcal{U}_{r_j} \right] (\tilde{\psi}) \tag{2.27}
$$

$$
\cdots = \mathcal{D}_\text{inv} \circ \left( \bigotimes_{i=1}^s \mathcal{D} \right) (\tilde{\psi}) \tag{2.28}
$$

where 2.29 results from repeatedly twirling a $\mathcal{D}$ operator and commuting the resulting $\mathcal{D}_D$ leftwards as in 2.26-2.28. Then

$$
F_G(s) = \text{tr} \left[ \tilde{E}_\psi \mathcal{D}_\text{inv} \circ \left( \bigotimes_{i=1}^s \mathcal{D} \right) (\tilde{\psi}) \right] \tag{2.30}
$$

$$
= \text{tr} \left[ \tilde{E}_\psi \mathcal{D}_\text{inv}(\tilde{\psi}) \right] p^s + (1-p^s) \text{tr} \left[ \tilde{E}_\psi \mathcal{D}_\text{inv}(I/2) \right] \tag{2.31}
$$

where we get $s$ copies of $\mathcal{D}_D$ in the first line. Setting $A_0 := \text{tr} \left[ \tilde{E}_\psi \mathcal{D}_\text{inv}(\tilde{\psi} - I/2) \right]$ and $B_0 := \text{tr} \left[ \tilde{E}_\psi \mathcal{D}_\text{inv}(I/2) \right]$, we get

$$
F_G(s) \approx A_0 p^s + B_0 \tag{2.32}
$$

This is known as the 0th order expansion of $F_G(s)$ \cite{107}. The terms $A_0$ and $B_0$ are nuisance parameters that contain information about the noise in state preparation and measurement. By performing the RB protocol above for various $s$, we can fit the 0th order model to the measurement data to find $p$ \cite{110}. Then, the average fidelity of the depolarizing channel, and hence, $\mathcal{D}$ is simply given by $\frac{1}{2}(1 + p)$ \cite{107}.

Therefore,

$$
F_G(s) \approx A_0 (2F_G(s) - 1)^s + B_0. \tag{2.33}
$$
Chapter 2: Randomized benchmarking in measurement-based quantum computing

2.7 Appendix: Clifford angles

Here we provide a list of measurement angles that implement elements of the single-qubit Clifford group, assuming that all measurement outcomes are zero. Note that each element can be written as a product of generators $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ and $H = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

To implement the full list of elements, the required measurement angles are:

<table>
<thead>
<tr>
<th>Gate</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{\pi}{2}$</td>
</tr>
<tr>
<td>$P$</td>
<td>0</td>
<td>$\frac{3\pi}{2}$</td>
<td>$\frac{3\pi}{2}$</td>
</tr>
<tr>
<td>$P^2$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{3\pi}{2}$</td>
<td>$\frac{3\pi}{2}$</td>
</tr>
<tr>
<td>$P^3$</td>
<td>0</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{\pi}{2}$</td>
</tr>
<tr>
<td>$H$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PH$</td>
<td>0</td>
<td>$\frac{\pi}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$P^2H$</td>
<td>0</td>
<td>$\pi$</td>
<td>0</td>
</tr>
<tr>
<td>$P^3H$</td>
<td>0</td>
<td>$\frac{3\pi}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$HP$</td>
<td>0</td>
<td>0</td>
<td>$\pi$</td>
</tr>
<tr>
<td>$PHP$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{\pi}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$P^2HP$</td>
<td>0</td>
<td>$\pi$</td>
<td>$\frac{3\pi}{2}$</td>
</tr>
<tr>
<td>$P^3HP$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{3\pi}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$PHP$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{\pi}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$P^2HP$</td>
<td>0</td>
<td>$\pi$</td>
<td>$\frac{3\pi}{2}$</td>
</tr>
<tr>
<td>$P^3HP$</td>
<td>$\frac{\pi}{2}$</td>
<td>$\frac{3\pi}{2}$</td>
<td>0</td>
</tr>
</tbody>
</table>
2.8 Appendix: 2-design elements

Here we give the $A$ matrices from section Sec. 2.4.3. These offer a compact description of one of the unitary 2-designs discussed in Ref. [100].

\[
A_1 = \begin{pmatrix}
\frac{1}{\sqrt{3}} & -\frac{1}{6}(1 + i)(\sqrt{3} + 3i) \\
\frac{1}{6}(1 + i)(3 + i\sqrt{3}) & -\frac{1}{\sqrt{3}}
\end{pmatrix}
\quad (2.34)
\]

\[
A_2 = \begin{pmatrix}
\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}}(1 + i) \\
\frac{1}{\sqrt{3}}(1 - i) & -\frac{1}{\sqrt{3}}
\end{pmatrix}
\quad (2.35)
\]

\[
A_3 = \begin{pmatrix}
0 & e^{-i\pi/4} \\
e^{i\pi/4} & 0
\end{pmatrix}
\quad (2.36)
\]

\[
A_4 = \begin{pmatrix}
1 & 0 \\
0 & -1
\end{pmatrix} = Z
\quad (2.37)
\]

\[
A_5 = \begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix} = X
\quad (2.38)
\]

\[
Q = Z_{\pi/4} \circ H \circ Z_{\cos^{-1}(\frac{1}{\sqrt{3}})} \circ H \circ Z_{\pi/4} \circ H
\quad (2.39)
\]

Note that $A_3$ is an element of the Clifford group.
Chapter 3

Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

3.1 Abstract

We consider measurement-based quantum computation that uses scalable continuous-variable cluster states with a one-dimensional topology. The physical resource, known here as the dual-rail quantum wire, can be generated using temporally multiplexed offline squeezing and linear optics or by using a single optical parametric oscillator. We focus on an important class of quantum gates, specifically Gaussian unitaries that act on single modes, which gives universal quantum computation when supplemented with multi-mode operations and photon-counting measurements. The dual-rail wire supports two routes for applying single-qumode Gaussian unitaries: the first is to use traditional one-dimensional quantum-wire cluster-state measurement protocols. The second takes advantage of the dual-rail quantum wire in order to apply unitaries by measuring pairs of qumodes called macronodes. We analyze and compare these methods in terms of the suitability for implementing single-qumode Gaussian measurement-based quantum computation.
3.2 Introduction

The introduction of measurement-based quantum computation (MBQC) over a decade ago [9] showed that adaptive local projective measurements alone are sufficient for quantum computation if a particular type of entangled resource called a cluster state [40] is available. In the optical regime, the continuous-variable (CV) Gaussian analogue [20, 64, 140] of qubit cluster states can be generated deterministically [65, 91] and in a highly scalable fashion [54–58, 96, 97, 141]. The generation of these states represents a big step towards achieving quantum computation using CVs [20, 65].

The canonical method for the construction of continuous-variable cluster states (CVCSs) uses momentum eigenstates [20, 65]. In the optical setting, finitely squeezed states are typically used instead, as momentum eigenstates have infinite energy. Using squeezed states results in the construction of approximate CVCSs, which are intrinsically noisy. There is no way to eliminate this noise entirely [60–62], but recent work has shown [87] that fault-tolerant MBQC is possible using finitely squeezed CVCSs as long as qubit-based quantum information is appropriately encoded in the qumodes [67] and the level of squeezing in the cluster state (and encoded qubits) is above a fixed, finite value called the squeezing threshold.

Related to this issue is the development of methods for MBQC that attempt to use available experimental squeezing resources more efficiently, in the sense that they introduce less noise from finite squeezing. Improving these methods will help to reduce the experimental demands set by an error-corrective approach for dealing with finite squeezing. One approach is to optimize the CVCS generation process to produce better-quality approximations of ideal CVCSs from the available resources [91].

An additional concern for top-down approaches is the trade-off between the quality of the approximation and the scalability of the construction process. Methods for generating CVCSs that employ optical parametric oscillators (OPOs) have shown excellent scalability. Two examples are the single-OPO method [54–56] and the temporal-mode linear optics method [57].

The single-OPO method generates entangled states in a single-shot, using an OPO. All of the squeezing and linear optics takes place inside the OPO, and the OPO cavity eigenmodes serve as the carriers of quantum information, referred to here as qumodes [54–56, 95, 142]. This method sets demands on the OPO (specifically the nonlinear crystal contained within it) and the frequency content of the pump beam. Once these prerequisites can be achieved within the laboratory [96], CVCSs can be generated with

\(^1\)From here on we will assume that the term ‘CVCS’ refers to the broader class of approximate physical states that approach the idealized case in the infinite squeezing limit.
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

a pump beam complexity that scales as a constant with the number of qumodes. By using multiple OPOs, this method can be used to generate higher-dimensional graph structures [58].

The temporal-mode linear optics method works by generating a small section of the cluster state and then repeatedly extending it (as required) using a basic set of optical machinery [97]. It uses temporally encoded qumodes, offline squeezing, and linear optics to generate the cluster [57]. In a recent result, this was achieved on the scale of over 10,000 entangled qumodes [141]. Both of these methods can generate states with 1D and 2D topologies [57]. We call these the dual-rail quantum wire (DRW)\(^2\) and the quad-rail lattice, respectively.

This work provides a basic framework for characterizing quantum computation on the DRW, showing how the noise introduced to the computation by finite squeezing depends on the measurement protocol used to implement gates. We will focus on the set of unitaries that can be implemented using just homodyne detection on the DRW. This set is an important subgroup of all single-qumode unitaries: single-qumode Gaussian unitaries [64, 65]. Adding the ability to count photons enables universal single-qumode MBQC on the DRW.

Extending this to universal quantum computation requires supplementing the above resources with a multi-qumode gate. Some results in this direction involve introducing additional linear cluster-state resources and Bell measurements in order to apply entangling gates between pairs of qumodes [143].

Alternatively, one can use a CVCS with higher-dimensional graph structure to perform a two-qumode gate using measurements alone. On a CVCS with 2D topology, such as the quad-rail lattice, homodyne detection alone implements all multi-qumode Gaussian unitaries, and the addition of photon counting enables fully universal MBQC [65]. Our analysis will be limited to the DRW, with generalization to the quad-rail lattice left to future work.

In particular, we consider two measurement protocols. The first applies traditional continuous-variable quantum wire (CVW) cluster measurements [65, 144, 145] to the state, using the fact that the DRW can be converted to a CVW. We will refer to this protocol as the CVW protocol. The other type treats the DRW as a double-thick quantum wire with pairs of nodes called macronodes at each wire site. We refer to this protocol as the macronode protocol. It bears some resemblance to sequential CV teleportation [141]. These approaches are illustrated in Fig. 3.1.

\(^2\)Not to be confused with dual-rail photonic qubits, as in Ref. [103].
Our analysis of the CVW protocol involves consideration of a class of CVWs containing states generated by methods of interest discussed above. This class is characterized by just two parameters, which are weights that label the edges of the CVW graph [59]. We relate these graphical parameters to noise introduced by finite squeezing during single-qumode Gaussian quantum computation, showing that, despite scalability inherited from the DRW generation process, the CVW protocol is a suboptimal strategy because the values of the graphical variables for the CVW introduce excessive noise to MBQC.

The key feature of the macronode protocol is that it does not involve conversion of the DRW into a CVW and thereby makes full use of the available squeezing. We show that this type of protocol can be used to implement arbitrary single-qumode Gaussian unitary gates using fewer qumodes than the CVW protocol. We also discuss an interesting

Figure 3.1: (Color online) Two ways of implementing measurement-based quantum computation on the dual-rail quantum wire (DRW). (a) Simplified graph [57] of the DRW. (b) The continuous-variable quantum-wire (CVW) protocol involves converting the DRW to a CVW by measurement of the position-quadrature (\(\hat{q}\)) basis on the top qumodes [57], followed by single-qumode homodyne measurements in some quadrature bases \(\hat{b}_i\) to evolve and propagate the state to the right along the wire [65, 144, 145]. (c) The macronode protocol involves encoding the input state within the leftmost macronode (pair of qumodes). Each macronode is measured by homodyne detection of its constituent qumodes (\(\hat{b}_{ia}, \hat{b}_{ib}\)). Graph weights [59] have been omitted for convenience. Color represents the sign/phase of each link. Blue/orange represents \(\pm\) sign respectively, and red is a complex sign of \(i\). The magnitude of each red self-loop is \(\varepsilon_D = \text{sech}(2\alpha)\) where \(\alpha > 0\) is the overall squeezing parameter [57]. The adjoining edges have magnitude \(g_D = \frac{1}{2} \tanh(2\alpha)\). The black edge and self-loops contained in the green ovals label the modes containing the encoded input state.
special case of the macronode protocol that allows us to reproduce a CVW-like mode of computation, which we call here the dictionary protocol. This protocol allows us to port measurement procedures (and hence, algorithms) that apply to the CVW directly to the DRW. We show that the noise properties of the general macronode protocol are more favorable than both the dictionary and the CVW protocols, which both perform comparably in terms of noise.

In Sec. 3.3 we review single-qumode Gaussian quantum computation on a class of CVWs known as uniformly weighted wires. Next we quantify the noise introduced in a computation due to finite squeezing for the CVW protocol. In Sec. 3.4 we introduce the macronode and dictionary protocols. We compare them to each other and to the CVW protocol, with respect to noise per single-qumode Gaussian unitary, showing that the macronode protocol always outperforms the others. As a quantitative application of our results, in Sec. 3.5 we analyze the noise of implementing rotation gates using three or four measurements in the different protocols. We show that the extra degree of freedom in the four-measurement case can lead to a dramatic reduction in the noise for particular gates, while three measurements remain favorable for others. Section 3.6 concludes with some discussion.

3.3 CVW protocol

Traditionally, the connection between CVCSs and graphs is as follows [65]. Nodes/vertices represent momentum eigenstates, and weighted links/edges between them represent the application of a controlled-Z gate (defined below) between two qumodes, with the gate interaction strength being equal to the edge weight (usually the weight of each graph edge is 1). As a graphical description of CVCSs, this is unphysical because the corresponding states cannot be normalized. However, it can be taken as the infinite-squeezing limit of approximate CVCSs, which are Gaussian pure states [59]. Given a particular ideal CVCS with adjacency matrix $A$, the corresponding family of approximate CVCSs is defined by those states whose complex graph $Z$ [59] approaches $A$ in the infinite-squeezing limit. Since $A$ has real entries, the imaginary part of $Z$ must vanish in this limit.

We will consider a restricted class of approximate CVCSs that are CVWs with uniformly weighted graphs of the following form [59]:

$$Z = gA_{BL} + i\varepsilon I.$$  \hspace{1cm} (3.1)
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Here $A_{BL}$ is a binary (B) adjacency matrix corresponding to a linear (L) graph. The parameter $g$ is allowed to take any real value (it comes from the controlled-Z gate; see Eq. (3.5)) and is assigned to all the links between neighbouring nodes on the graph. The second term describes the self-loop edges, which all have weight $\varepsilon$, and we require that $\varepsilon \to 0$ in the infinite-squeezing limit. For CVWs produced from the DRW, these weights are denoted as $g_D$ and $\varepsilon_D$ and have a specific form that depends on the overall squeezing parameter, $\alpha > 0$ [57] (see Fig. 3.1). They are defined as

$$
g_D := \frac{1}{2} \tanh(2\alpha) \tag{3.2}
$$

and

$$
\varepsilon_D := \text{sech}(2\alpha). \tag{3.3}
$$

We will assume we are working in the general uniformly-weighted wire case, except for when drawing conclusions specifically for CVWs produced from the DRW. We use the notation $\{g, \varepsilon\}$ and $\{g_D, \varepsilon_D\}$, respectively, in order to distinguish these cases.

3.3.1 MBQC on uniformly-weighted CVWs

Once a suitable CVW resource state has been generated, single-qumode Gaussian computation proceeds by measuring linear combinations of the canonical position and momentum quadrature operators—$\hat{q}$ and $\hat{p}$, respectively—on nodes on the wire. We employ the conventions that $[\hat{q}, \hat{p}] = i$ and $\hbar = 1$, which means that the variance of a qumode in its vacuum state is always $\langle \hat{q}^2 \rangle_{\text{vac}} = \langle \hat{p}^2 \rangle_{\text{vac}} = \frac{1}{2}$. The particular measurements in question will be

$$
\hat{b}_i := \alpha_i \left[ (\cos \theta_i) \hat{p} + (\sin \theta_i) \hat{q} \right] = \hat{p} + \sigma_i \hat{q}, \tag{3.4}
$$

where $\sigma_i = \tan \theta_i$ and $\alpha_i = \sec \theta_i$. Performing the logical measurement $\hat{b}_i$ is equivalent to physically measuring the rotated quadrature operator $(\cos \theta_i)\hat{p} + (\sin \theta_i)\hat{q}$ and then multiplying the measurement outcome by $\alpha_i$; such measurements can be achieved experimentally through homodyne detection [20, 145]. Below, we describe the effect of these measurements on an input state, but first we define some standard single- and two-qumode Gaussian operations [65].

The controlled-Z gate is

$$
\hat{C}_Z(g) := \exp(ig\hat{q} \otimes \hat{q}), \tag{3.5}
$$

where $g$ is the interaction strength [59]. It is a two-qumode entangling gate. The single-qumode squeezing gate, which squeezes the $\hat{q}$ quadrature by a factor of $s > 0$ (called the
squeezing factor), is
\[
\hat{S}(s) := \exp \left[ -i \left( \frac{\ln s}{2} \right) (\hat{q}\hat{\rho} + \hat{\rho}\hat{q}) \right],
\]  
(3.6)

where \( \ln s \) is called the **squeezing parameter**. We represent its Heisenberg action on the vector of single-qumode quadrature operators \( \hat{\mathbf{x}} = (\hat{q}, \hat{p})^T \) by the symplectic matrix \( \mathbf{S}(s) \):

\[
\hat{S}^\dagger(s) \hat{\mathbf{x}} \hat{S}(s) = \mathbf{S}(s) \hat{\mathbf{x}} = \begin{pmatrix} s & 0 \\ 0 & s^{-1} \end{pmatrix} \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix}.
\]  
(3.7)

In the Heisenberg picture, this operator has the action of rescaling the position and momentum quadratures by \( s \) and \( s^{-1} \) respectively.

The shearing gate is defined as
\[
\hat{P}(\sigma) := \exp \left( \frac{i\sigma \hat{q}^2}{2} \right),
\]  
(3.8)

with \( \sigma \) called the **shearing parameter**. We represent its Heisenberg action on \( \hat{\mathbf{x}} \) by the symplectic matrix \( \mathbf{P}(\sigma) \):

\[
\hat{P}^\dagger(\sigma) \hat{\mathbf{x}} \hat{P}(\sigma) = \mathbf{P}(\sigma) \hat{\mathbf{x}} = \begin{pmatrix} 1 & 0 \\ \sigma & 1 \end{pmatrix} \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix}.
\]  
(3.9)

In the Heisenberg picture, this operator acts as a shear in phase space parallel to the momentum axis by a gradient \( \sigma \).

The Fourier transform is
\[
\hat{F} := \exp \left[ \frac{i\pi}{4} (\hat{q}^2 + \hat{p}^2) \right]
\]  
(3.10)

We represent its Heisenberg action on \( \hat{\mathbf{x}} \) by the symplectic matrix \( \mathbf{F} \):

\[
\hat{F}^\dagger \hat{\mathbf{x}} \hat{F} = \mathbf{F} \hat{\mathbf{x}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix}.
\]  
(3.11)

In the Heisenberg picture, this operator acts as a \( \frac{\pi}{2} \) clockwise rotation of the quadratures.

This is a special case of a more general rotation,
\[
\hat{R}(\theta) := \exp \left[ \frac{i\theta}{2} (\hat{q}^2 + \hat{p}^2) \right],
\]  
(3.12)

whose Heisenberg action on \( \hat{\mathbf{x}} \) is given by the symplectic matrix \( \mathbf{R}(\theta) \):

\[
\hat{R}^\dagger(\theta) \hat{\mathbf{x}} \hat{R}(\theta) = \mathbf{R}(\theta) \hat{\mathbf{x}} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \hat{q} \\ \hat{p} \end{pmatrix}.
\]  
(3.13)
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

In the Heisenberg picture, this operator rotates the quadrature operators clockwise by an angle $\theta$. These gates will be useful throughout the rest of this Article.

Now let us return to characterizing CVW measurements. It is sufficient to consider the measurement of a small portion of the CVW, as illustrated in Fig. 3.2 for one measurement with a single-qumode input state $|\psi\rangle$. The input state is encoded on the left-most wire node, which we label as the $i^{th}$ node. In the Schrödinger picture, the measurement of the $i^{th}$ node translates the input state $|\psi\rangle_i$ one node to the right and applies the following operation on the encoded input state:

$$|\psi\rangle_i \mapsto \hat{N}(\varepsilon) \hat{X} \left( \frac{m_i}{g} \right) \hat{U}_i |\psi\rangle_{i+1},$$

(3.14)

where we refer to $\hat{U}_i$ as the logic gate (after one measurement), $\hat{X} \left( \frac{m_i}{g} \right)$ as the displacement, and $\hat{N}(\varepsilon)$ as the noise operator. These are discussed below.

The logic gate $\hat{U}_i$ can be decomposed as

$$\hat{U}_i = \hat{F} \hat{S}(g) \hat{P}(\sigma_i),$$

(3.15)

which are all defined above. Any single-qumode Gaussian unitary can be decomposed into a finite sequence of $\hat{U}_i$'s (up to displacements) [145]. While such gates are parameterized by three degrees of freedom in general, at least four CVW measurements are required in order to implement them [145]. In principle one might naively think that with three measurement degrees of freedom, it is at least possible to get arbitrarily close to all single-qumode Gaussian unitaries. We shall see later in Sec. 3.5.1 that this is ruled out in practice because the noise from finite squeezing diverges around the unachievable gates (independent of the amount of squeezing). For this reason, we assume four measurements are used, as this avoids such divergences while still being sufficient for implementing arbitrary single-qumode Gaussian unitaries [144, 145].

Next we define the phase-space displacements,

$$\hat{X}(u) := \exp (-iu\hat{p}), \quad \hat{Z}(v) := \exp (iv\hat{q}).$$

(3.16)

In the Heisenberg picture

$$\hat{q} \xrightarrow{\hat{X}(u)} \hat{q} + u, \quad \hat{p} \xrightarrow{\hat{Z}(v)} \hat{p} + v.$$  

(3.17)

If this is the beginning of a computation, then we can set $|\psi_{\text{in}}\rangle = \hat{S}(s)|0\rangle$ (with $s \gg 1$). Otherwise we can assume $|\psi_{\text{in}}\rangle$ is just the output of some previous computation step on the cluster or a state injected onto the cluster by an entangling operation with the left-most node.
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

\[ |\psi\rangle \quad |0\rangle \quad S(e^{-1/2}) \quad \hat{N}(\varepsilon) \hat{X}(\frac{m_g}{g}) \hat{U}_1 |\psi\rangle \]

Figure 3.2: (Color online) (a) Circuit diagram for an element of Gaussian measurement-based quantum computation. The input \(|\psi\rangle\) is entangled with a momentum-squeezed vacuum state by a \(C_Z(g)\) interaction (see Eq. (3.5)). The dotted line encapsulates CVCS construction. After measuring the top qumode with outcome \(m_1\), the following operations are applied to the output \(|\psi\rangle\): logic gate \(\hat{U}_1\) (see Eq. (3.15)), a displacement, \(\hat{X}(\frac{m_g}{g})\) (see Eq. (3.16)), and noise \(\hat{N}(\varepsilon)\) (see Eq. (3.19)). (b) Graphical representation of a CVW with an input state on the leftmost-qumode. When using the graphical representation, we must assume that the input state is Gaussian [59]. In Sec. 3.3.3 we use Wigner functions to generalize the description to mixed input states. The blue solid part of the wire is represented in the circuit diagram above.

In either case, the other quadrature is left alone. In general, the position-quadrature \((\hat{q})\) displacement operator \(\hat{X}(\frac{m}{g})\) which acts on the the output state in Eq. (3.14) has to be corrected for, either by applying the inverse operation on the output, which we call the correction

\[ \hat{C} := \hat{X}\left(-\frac{m}{g}\right), \quad (3.18) \]

or by adaptive measurement protocols, with future measurements depending on prior measurement outcomes. One caveat is that for measurement-based implementations of Gaussian unitaries, the adaptive measurement protocol is trivial: future measurements do not need to be adjusted based on prior measurement outcomes. Thus, only a final phase-space displacement correction is required for these protocols [65].

The noise operator from Eq. (3.14) is

\[ \hat{N}(\varepsilon) \propto \exp\left(-\varepsilon \hat{q}^2\right), \quad (3.19) \]

which is not unitary and requires the output state to be normalized afterward (hence the \(\propto\) symbol). It applies noise from finite squeezing to the state. For large squeezing, \(\varepsilon\) is small. In the Schrödinger picture, this operator multiplies the state’s position-space wavefunction by a 0-mean Gaussian with variance \(\varepsilon^{-1}\), called a Gaussian envelope (usually large). Equivalently, it convolves the state’s momentum-space wavefunction by a 0-mean Gaussian with variance \(\varepsilon\) (usually small). In terms of wavefunctions, these two actions are equivalent, and only one or the other is ever needed to describe the action.
of this operator. An intuitive explanation is that in the position-space representation, the part of the wavefunction corresponding to large displacements in position (from the origin) is suppressed by the envelope. In the momentum-space representation, the wavefunction is “blurred out” by the convolution [20].

As will be shown in Sec. 3.3.3, in the Wigner-function representation this operation has two simultaneous actions: (1) multiplying the state’s Wigner function by a 0-mean Gaussian envelope in position with variance $\frac{1}{2}\varepsilon^{-1}$ and also (2) convolving the state’s Wigner function in momentum by a 0-mean Gaussian with variance $\frac{1}{2}\varepsilon$.

Note that in terms of wavefunctions, just one action (either envelope or convolution) is needed to represent the complete action of this operator, while two actions are involved in the Wigner representation. Also note that the variance of the envelope and that of the convolution are both reduced by a factor of 2 when moving from the wavefunction representation to the Wigner representation. This can be understood as accounting for the fact that the wavefunction is an amplitude, while the Wigner function is a (quasi-)probability [146].

Finally, notice that as $\varepsilon \to 0$, $\hat{N}(\varepsilon) \to \hat{I}$, where $\hat{I}$ is the identity. Thus, wires with small $\varepsilon$ introduce less noise, which accords with our intuition about large squeezing corresponding to a better-quality CVCS [20, 65]. In the next section, we will investigate how the noise depends on the wire weight $g$ by considering gates implemented by multiple measurements.

### 3.3.2 Noise dependence on $g$

First, consider a wire with uniform weight $g = 1$. Although the $\hat{N}(\varepsilon)$ operator applied after one measurement introduces noise into the $\hat{p}$ quadrature, the Fourier transform that also gets applied will cause the noise from subsequent measurements to be distributed across both quadratures in a manner that depends also on the shearing parameters. In the case when all the shearing parameters are set to zero, noise will be added equally between the quadratures and the gate applied after $n$ measurements will be $\hat{F}^n$ (note that $\hat{F}^4 = \hat{I}$).

For $g \neq 1$, there is an additional squeezing operation that rescales the relative weight of some of the noise (see Eq. (3.15)) assuming that the ideal logical gate ($\hat{U}_n \cdots \hat{U}_2 \hat{U}_1$) is fixed. This is easy to see by considering an even number of measurements and ignoring the displacement terms, resulting in the total operation shown below:

$$
\hat{N}(\varepsilon) \hat{F} \hat{S}(g) \hat{P}(\sigma_n) \hat{N}(\varepsilon) \hat{F} \hat{S}(g) \hat{P}(\sigma_{n-1}) \cdots \hat{N}(\varepsilon) \hat{F} \hat{S}(g) \hat{P}(\sigma_2) \hat{N}(\varepsilon) \hat{F} \hat{S}(g) \hat{P}(\sigma_1). \tag{3.20}
$$
Commuting every odd squeezing operation to the left, past the following Fourier transform, noise, and shearing operations, we get

\[ \hat{N}(\varepsilon)\hat{F}\hat{P}(\sigma'_n)\hat{N}(\varepsilon g^{-2})\hat{F}\hat{P}(\sigma_{n-1})\cdots\hat{N}(\varepsilon)\hat{F}\hat{P}(\sigma'_2)\hat{N}(\varepsilon g^{-2})\hat{F}\hat{P}(\sigma_1). \]  

(3.21)

Where \( \sigma'_i = \sigma_i g^{-2} \) for \( i \) even. The form of the above expression (Eq. (3.21)) has the equivalent interpretation of the operation that is applied when a \( g = 1 \) wire is measured, only with every second noise parameter being rescaled. We can realise this graphically if we interpret this rescaling instead as the application of a local squeezing operation \( \hat{S}(g) \) on every second wire node. Then, assuming a Gaussian input state labeled by the complex self-loop weight \( z_\psi \), we have have effectively remodeled the weight-\( g \) wire

\[ z_\psi \quad i\varepsilon \quad i\varepsilon \quad i\varepsilon \quad i\varepsilon \quad \cdots \quad g \quad g \quad g \quad g \quad \cdots \]  

(3.22)

into a weight-1 wire with non-uniform self-loop weights:

\[ z_\psi \quad i\varepsilon g^{-2} \quad i\varepsilon \quad i\varepsilon g^{-2} \quad i\varepsilon \quad \cdots \quad 1 \quad 1 \quad 1 \quad 1 \quad \cdots \]  

(3.23)

Note that we have assumed that the input state is Gaussian in order to represent the CVW state using the graphical calculus [59]. This provides an intuitive pictorial representation of the remodeling procedure. The result is fully general, however, and applies to arbitrary inputs, including mixed states.\(^4\)

The above shows that it is possible to encorporate the change in wire weight \( (1 \mapsto g) \) into a rescaling by \( g^{-2} \) of the measurement basis and noise parameter, on half the nodes. Consequently, the lower the value of \( g \), the higher the overall noise introduced. In fact, the parameter \( g \) sets the noise bias between the quadratures since the noise alternates between the quadratures due to the Fourier transform, and only the \( \hat{q} \)-quadrature noise gets rescaled. In terms of overall noise, lower-weight wires amplify the noise from finite squeezing and are therefore suboptimal.

\(^4\)This can be verified straightforwardly with Wigner functions using the methods employed in Sec. 3.3.3. We leave an explicit proof as an exercise for the reader.
Another way to take the weight $g$ into account is to rescale the measurement outcomes of every node by $\sqrt{g}$, like applying a $\hat{S}(\sqrt{g})$ on every node as shown on the wire below.

\[
\begin{array}{ccccccc}
\hat{z} & i\hat{e} & i\hat{e} & i\hat{e} & i\hat{e} \\
1 & 1 & 1 & 1 & 1
\end{array}
\]

This results in a CVW weighted uniformly along links and self-loops, with noise seemingly introduced in equal quantities to each quadrature (the self-loop weights of every odd node is the same as every even node). In doing so, the input state $z_\psi$ has effectively been squeezed on the wire ($z_\psi \rightarrow z_\psi g^{-1}$). This can be thought of as an “encoding” onto the effective “$g = 1$” wire with uniform self-loops ($\varepsilon' = \varepsilon g^{-1}$). The measurement protocol must then be changed, $\hat{p} + \sigma_1 \hat{q} \rightarrow \hat{p} + \sigma_1 g^{-1} \hat{q}$, in order to effect an equivalent logic gate. The advantage is that the noise properties are described by a single parameter $\varepsilon'$ up to the encoding. While it appears as though the noise is added to the quadratures in equal amounts, this is only after applying the encoding, which effectively rescales the quadratures relative to one another (and biases the un-biased noise structure). However, if the computational protocol is assumed to start with a blank ancilla squeezed by $\sqrt{\varepsilon g^{-1}}$ (rather than $\sqrt{\varepsilon}$, see footnote 3), then we can treat our wire as having weight $g = 1$ with rescaled self-loops.

Let us consider the types of CVWs generated from the DRW. By performing $\hat{q}$ measurements on all the top nodes, the DRW can be converted to the CVW with uniform edge weight $g_D$ and uniform self-loop weight $\varepsilon_D$ (see Eqs. (3.2) and (3.3)). The edge weight $g_D$ is upper bounded by the value $\frac{1}{2}$. By applying the parameter rescaling corresponding to Eq. (3.24) and defining $\varepsilon'_D := \varepsilon_D g_D^{-1}$, we see that this protocol amplifies the effect of the self-loop noise that gets added:

\[
\hat{N}(\varepsilon'_D) = \hat{N}(2 \text{csch} 2\alpha) \rightarrow \hat{N}(2\varepsilon_D)
\]

where the limit is that of large squeezing ($\alpha \rightarrow \infty$). The arguments above highlight the importance of the CVW edge weight $g$ on the computation, demonstrating why using the CVW protocol is suboptimal: $g_D$ is small relative to $g = 1$ for the standard CVW [20, 65, 144], and this results in more noise from finite squeezing.

The purpose of the next section will be to derive a more quantitative description of the noise using the Wigner-function formalism. This will address the following issues: First, we wish to compare the CVW and macronode protocols in terms of how much noise is applied per gate, yet the noise will depend on the measurement bases, which will be completely different and could even involve a different number of measurements. Second, when applying the measurement-dependent correction, the Gaussian envelope
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

that is the manifestation of the noise from finite squeezing acting on the position-space wavefunction will be displaced relative to the origin. Thus, the noise depends upon the measurement outcome [65]. The Wigner-function formalism allows us to describe mixed states, particularly the output states of MBQC that are averaged over possible measurement outcomes. This will help us to define a quantity that captures the average noise introduced by \( n \) measurements and hence compare the measurement protocols in a noise-per-gate manner.

3.3.3 Wigner function formalism

Wigner functions are quasi-probability distributions that provide a phase-space picture for arbitrary mixed states. Given an arbitrary input state \( \hat{\rho} \), its Wigner function is given by [146]

\[
W_{\hat{\rho}}(q,p) := \frac{1}{2\pi} \int dq \langle q - \frac{x}{2} | \hat{\rho} | q + \frac{x}{2} \rangle e^{ixp},
\]  

(3.26)

where the subscript \( q \) labels position basis states, and \( \mathbf{x} = (q_1, q_2, \ldots, p_1, p_2, \ldots)^T \) is a vector of \( c \)-numbers. We let \( \mathbf{x}_i = (q_i, p_i)^T \) denote the \( i \)th qumode register.

We now define the action of unitary operations on quantum states in the Wigner-function formalism. Given a quantum state \( \hat{\rho} \) with Wigner function \( W_{\hat{\rho}}(q,p) \) as above, the state evolves under a unitary operator \( \hat{B} \) as \( \hat{\rho} \rightarrow \hat{B} \hat{\rho} \hat{B}^\dagger \), whose Wigner function is \( W_{\hat{B} \hat{\rho} \hat{B}^\dagger}(q,p) \).

We label the Wigner representation of an arbitrary unitary operator \( \hat{B} \) by the same symbol but with calligraphic font:

\[
\mathcal{B}[W_{\hat{\rho}}(q,p)] := W_{\hat{B} \hat{\rho} \hat{B}^\dagger}(q,p),
\]  

(3.27)

for all unitaries \( \hat{B} \).

Under Gaussian unitary evolution \( \hat{E} \) with Heisenberg-picture symplectic representation \( \mathbf{E} \) defined by

\[
\mathbf{x} \xrightarrow{\hat{E}} \mathbf{E}^\dagger \mathbf{x} \hat{E} =: \mathbf{E} \mathbf{x} + \mathbf{c},
\]  

(3.28)

where \( \mathbf{x} = (\hat{q}_1, \hat{q}_2, \ldots, \hat{p}_1, \hat{p}_2, \ldots)^T \), the Wigner-function arguments (which are \( c \)-numbers) update in the reverse way to the Heisenberg evolution of operators:

\[
\mathcal{E}[W_{\hat{\rho}}(\mathbf{x})] = W_{\mathcal{E}_{\hat{\rho}} \hat{E} \dagger}(\mathbf{x}) \\
= W_{\hat{\rho}}(\mathbf{E}^{-1}(\mathbf{x} - \mathbf{c})).
\]  

(3.29)

Note that under the action of \( \mathcal{E} \), the argument of the Wigner function updates using \( \mathbf{E}^{-1} \) after a displacement by \(-\mathbf{c}\), while the Heisenberg evolution of quadrature operators due to \( \hat{E} \) uses \( \mathbf{E} \) before a displacement by \(+\mathbf{c}\) (Eq. 3.28).
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Now we recast the evolution shown in Fig. 3.2 in the Wigner formalism, extending the results in Ref. [65] to weight-$\epsilon$ wires and single-qumode Gaussian unitaries. Define the Gaussian function $G_y(x)$ to be

$$ G_y(x) = \frac{1}{\sqrt{\pi y}} \exp\left( -\frac{x^2}{y} \right), $$

(3.30)

which is a normalized Gaussian with mean 0 and variance $y/2$. Incidentally, we can write the noise operator from Eq. (3.19) as $\hat{N}(\epsilon) \propto G_2/\epsilon(\hat{q})$.

Let the (possibly mixed) input state $\rho_{in}$ be represented by the Wigner function $W_{in}(q, p)$. A blank cluster qumode—i.e., a momentum-squeezed state with (large) squeezing factor $\epsilon^{-1/2}$—is represented by the Wigner function $G_{1/\epsilon}(q)G_\epsilon(p)$. The initial two-qumode state in Fig. 3.2, which consists of the input state attached via $\hat{C}_Z(g)$ to a blank cluster qumode, is represented by the following Wigner function:

$$ W_{in}(q_1, p_1 - gq_2)G_{1/\epsilon}(q_2)G_\epsilon(p_2 - gq_1). $$

(3.31)

Define the symplectic-matrix representation of $\hat{U}_i$ (see Eq. (3.15)) to be

$$ U_i = \begin{pmatrix} -\sigma_i g^{-1} & -g^{-1} \\ g & 0 \end{pmatrix}. $$

(3.32)

Then the Wigner function $W_{out}(x_2)$ for the output state after a single CVW node measurement (in the basis $\hat{p} + \sigma_1 \hat{q}$) and after applying the correction operator $\hat{C}$ (see Eq. (3.18)) is given by

$$ P(m_1)W_{out}(x_2) = \int d\tau_2 W_{in}(U_1^{-1}(x_2 + g\tau))G_\epsilon(g\tau_2)G_{1/\epsilon}(q_2 + m_1 \frac{1}{g}), $$

(3.33)

where $P(m_1)$ is the probability of measuring outcome $m_1$, and $\tau = (0, \tau_2)^T$. Eq. (3.33) shows that the noise from finite squeezing manifests as both a phase-space Gaussian convolution in momentum and a Gaussian envelope in position with measurement-outcome-dependent mean. As the measurement outcomes will be different each time, we consider the measurement-averaged distortion on the output state:

$$ W_{avg}(x_2) = \int dm_1 P(m_1)W_{out}(x_2) $$

$$ = \int d\tau'_2 W_{in}(U_1^{-1}(x_2 + \tau'))G_\epsilon(\tau'_2), $$

(3.34)

where $\tau' = g\tau = (0, \tau'_2)^T = (0, g\tau_2)^T$. Thus, the average effect of noise from finite squeezing is a Gaussian convolution, similar to the $g = 1$ case treated in Ref. [65].
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Iterating the above expression yields the average Wigner function \( W^{\text{avg}}_{\text{avg}}(x_n) \) after \( n \) homodyne measurements \( \hat{b}_i = \hat{p} + \sigma_i \hat{q} \) on an \( n \)-node CVW. Define \( W^{(0)}_{\text{avg}}(x) := W_{\text{in}}(x) \).

\[
W^{(n)}_{\text{avg}}(x_n) = \int d\tau_2^{(n)} W^{(n-1)}_{\text{avg}} \left( U^{-1}_n(x_n + \tau^{(n)}) \right) 
\times G_{\varepsilon} \left( \tau_2^{(n)} \right), \tag{3.35}
\]

where \( \tau^{(n)} = (0, \tau_2^{(n)})^T \), and \( U_n \) is the symplectic matrix representation of the Heisenberg action of \( \hat{U}_n \). Just as in Eq. (3.19), each \( i \)th measurement convolves \( W^{(i-1)}_{\text{avg}} \) in the momentum quadrature. Now we have expressed the average output state using the Wigner formalism.

We are interested in characterizing MBQC on the CVW in terms of how much noise is added from finite squeezing. There are a couple of different ways to “unpack” Eq. (3.35) into a description involving the desired computation and the added noise. If the noise and displacements are ignored, then measuring the first \( n \) nodes on a CVW applies the operation

\[
\hat{U}_\sigma = \hat{U}_n \cdots \hat{U}_2 \hat{U}_1, \tag{3.36}
\]

where the overall unitary applied depends on the shearing parameters \( \sigma := (\sigma_1, \ldots, \sigma_n) \). This can be thought of as the ideal operation applied in the absence of noise and after the displacements are corrected for. Define

\[
\tilde{U}_i := U_i \cdots U_1, \tag{3.37}
\]

and

\[
\check{U}_i := U_n \cdots U_{i+1}, \tag{3.38}
\]

where \( n \) is the total number of homodyne measurements made on the CVW. Note that \( \check{U}_0 = \tilde{U}_i \check{U}_i = \tilde{U}_n \), which is the symplectic representation of the Heisenberg action of the full \( \hat{U}_\sigma \). The Wigner function for the (ideal) output state without noise and displacements (i.e., ignoring \( \hat{N}(\varepsilon) \hat{X}(\frac{m_i}{2}) \) in Eq. (3.14)) is

\[
W_{\text{ideal}}(x) := W_{\text{in}} \left( \tilde{U}_n^{-1} x \right). \tag{3.39}
\]

Using this, Eq. (3.35) can be expanded out as

\[
W^{(n)}_{\text{avg}}(x_n) = \int d\tau_2^{(n)} \cdots d\tau_2^{(1)} W_{\text{ideal}} \left( x_n + \sum_{i=1}^{n} \tilde{U}_i \tau^{(i)} \right) 
\times G_{\varepsilon} \left( \tau_2^{(n)} \right) \cdots G_{\varepsilon} \left( \tau_2^{(1)} \right). \tag{3.40}
\]
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

This expression shows us that the average effect of \( n \) homodyne measurements relative to the ideal unitary evolution (which maps \( W_{\text{in}} \rightarrow W_{\text{ideal}} \)) is \( n \) gate-dependent Gaussian convolutions. Let \( \mathcal{N}_\sigma \) denote the application of these Gaussian convolutions to an arbitrary Wigner function \( W(x) \):

\[
\mathcal{N}_\sigma[W(x)] := \int d\tau_2^{(n)} \cdots d\tau_2^{(1)} W\left(x + \sum_{i=1}^{n} \tilde{U}_i \tau^{(i)}\right) \\
\times G_\varepsilon\left(\tau_2^{(n)}\right) \cdots G_\varepsilon\left(\tau_2^{(1)}\right). \tag{3.41}
\]

The Wigner function \( W_{\text{avg}}^{(n)} \) can then be expressed as

\[
W_{\text{avg}}^{(n)}(x_n) = \mathcal{N}_\sigma \circ \mathcal{U}_\sigma[S_{\text{in}}(x_n)] \\
= \mathcal{N}_\sigma[S_{\text{ideal}}(x_n)]. \tag{3.42}
\]

Thus, \( \mathcal{N}_\sigma \) can be thought of as the average noise added by the \( n \)-measurement channel after a perfect logic gate \( \tilde{U}_\sigma \) is applied.

To arrive at an expression for the effect of the noise from finite squeezing before the perfect logic gate is applied, consider undoing the ideal unitary—i.e., applying \( \tilde{U}_\sigma^\dagger \)—to see how the noise causes the output Wigner function to differ from that of the input. By using Eq. (3.29), we can define a new Wigner function for this case:

\[
W_{\text{undo}}^{(n)}(x_n) := \mathcal{U}_\sigma^{-1} \left[W_{\text{avg}}^{(n)}(x_n)\right] \\
= W_{\text{avg}}^{(n)}(\tilde{U}_n x_n). \tag{3.43}
\]

This is the average Wigner function after performing \( n \) measurements, applying the correction operators, and undoing the ideal logic gate \( \tilde{U}_\sigma \). Expanding this gives

\[
W_{\text{undo}}^{(n)}(x_n) = \int d\tau_2^{(n)} \cdots d\tau_2^{(1)} W_{\text{in}}\left(x_n + \sum_{i=1}^{n} \tilde{U}_i \tau^{(i)}\right) G_\varepsilon\left(\tau_2^{(n)}\right) \cdots G_\varepsilon\left(\tau_2^{(1)}\right). \tag{3.44}
\]

Similar to the above, we define \( \mathcal{N}_\sigma \) to be the map that applies these Gaussian convolutions to an arbitrary Wigner function \( W(x) \):

\[
\mathcal{N}_\sigma[W(x)] := \int d\tau_2^{(n)} \cdots d\tau_2^{(1)} W\left(x + \sum_{i=1}^{n} \tilde{U}_i \tau^{(i)}\right) G_\varepsilon\left(\tau_2^{(n)}\right) \cdots G_\varepsilon\left(\tau_2^{(1)}\right). \tag{3.45}
\]
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Since \( W^{(n)}_{\text{undo}}(x_n) = \mathcal{N}_\sigma[W_{\text{in}}(x_n)] \), we can expand the average output state as

\[
W^{(n)}_{\text{avg}}(x_n) = U_\sigma \left[ W^{(n)}_{\text{undo}}(x_n) \right] = U_\sigma \circ \mathcal{N}_\sigma[W_{\text{in}}(x_n)].
\]  (3.46)

Thus, \( \mathcal{N}_\sigma \) can be thought of as the average noise added by the \( n \)-measurement channel before a perfect logic gate \( \hat{U}_\sigma \) is applied.

We have two equivalent descriptions for the output of the \( n \)-homodyne CVW channel, shown by Eqs. (3.40) and (3.44), which are quite similar in form. These expressions can be simplified somewhat. In what follows, we will restrict our attention to the noise-before-gate decomposition (Eq. (3.44)). The other case can be derived analogously.

For \( n \geq 2 \) measurements, we can replace the \( n \) convolutions—each along a single phase-space direction—with a single bivariate phase-space convolution that will depend on all \( n \) measurements, as shown below. This is guaranteed to have Gaussian form, as the convolution of \( n \) Gaussians is itself Gaussian. Thus,

\[
W^{(n)}_{\text{undo}}(x_n) = \int \! d\kappa_1 d\kappa_2 W_{\text{in}}(x_n + \kappa) B_{\Sigma_n}(\kappa),
\]  (3.47)

where \( \kappa = (\kappa_1, \kappa_2)^T \) contains the new dummy variables for the bivariate convolution, replacing \( \tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)} \), and

\[
B_K(\kappa) = \left( \pi \sqrt{\det(K)} \right)^{-1} \exp(-\kappa^T K^{-1} \kappa) \]  (3.48)

is a normalized bivariate Gaussian distribution with covariance matrix \( \frac{1}{2} K \). Then,

\[
\Sigma_n = \sum_{i=1}^{n} \tilde{U}_i^{-1} \Sigma_\star \tilde{U}_i^{-T},
\]  (3.49)

where

\[
\Sigma_\star := \begin{pmatrix} 0 & 0 \\ 0 & \epsilon \end{pmatrix}.
\]  (3.50)

We can interpret \( \Sigma_\star \) as the covariance matrix for a single Gaussian convolution that represents the action of the \( \hat{N}(\epsilon) \) operator from Eq. (3.19) averaged over measurement outcomes. For future use in the Appendices, we also define

\[
\Sigma^* := \begin{pmatrix} \epsilon & 0 \\ 0 & 0 \end{pmatrix}.
\]  (3.51)

57
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Recall we are using the noise-before-gate decomposition (Eq. (3.46)). The analogous expression to Eq. (3.47) for the noise-after-gate decomposition (Eq. (3.40)) is

$$ W_{\text{avg}}^{(n)}(x_n) = \int d\kappa_1 d\kappa_2 W_{\text{ideal}}(x_n + \kappa) B_{\Sigma_n'}(\kappa), $$

(3.52)

where

$$ \Sigma_n' = \sum_{i=1}^n \tilde{U}_i^{-1} \Sigma_i \tilde{U}_i^{-T}. $$

(3.53)

For the rest of this Article, we will focus on the noise-before-gate description of CVW computation (as in Eq. (3.46)).

The covariance matrix $\Sigma_n$ is the only part of equation Eq. (3.47) that depends on the measurements in any way. It characterises the bivariate Gaussian convolution in terms of the number of measurements, which observable is measured, and ultimately the gate that was performed.

Consider the trace of $\frac{1}{2}\Sigma_n$, which is invariant under phase-space rotations. It is the sum of the variances along any two orthogonal phase-space directions, such as $q$ and $p$. Alternatively, by changing from cartesian coordinates $(q, p)$ to polar coordinates $(r, \theta)$, with $r = \sqrt{q^2 + p^2}$, this quantity can be interpreted as the average radial variance of the Wigner function (averaging uniformly over $\theta$) since $\Delta r^2 = \Delta q^2 + \Delta p^2$. Convolving $W_{\text{in}}$ with $B_{\Sigma}$ adds the variance of the bivariate Gaussian distribution to that of the input state. Thus, we can quantify the noise added to the state by defining the scalar variance

$$ SV(n) := \frac{1}{2} \text{tr}[\Sigma_n], $$

(3.54)

which quantifies the noise of the average output Wigner function, which in turn depends on the number of measurements, the choice of each measurement basis, and parameters $\epsilon$ and $g$.

As we shall see later, there are many ways of applying gates through single-qumode measurements on the CVW. In terms of the scalar variance (or how much noise is added), these methods will not be equivalent, and it will be useful to define the minimized scalar variance

$$ SV(n) := \min_{\{\sigma | \tilde{U}_n = E\}} SV(n), $$

(3.55)

where the minimization is over the measurement degrees of freedom (i.e., shearing parameters/homodyne angles) with the constraint that the total unitary applied is equal to the desired gate unitary $\tilde{E}$—i.e., $\tilde{U}_n = E$. Throughout this Article, we will use calligraphic font to denote the minimized version of the scalar variance with respect to any free measurement degrees of freedom and some gate. This will allow for a fair comparison between the CVW and macronode protocols.
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

The $g$ dependence of $SV(n)$ can be expressed simply for an even number of measurements $n$. Assuming that the logic gate in the noiseless limit ($\hat{U}_g$) is fixed, applying a rescaling of the shearing parameters as in Eq. (3.23) reduces the dependence of the scalar variance in terms of $g$ to be

$$SV(n) = \sum_{k=1}^{n/2} \left( f_{2k-1} + g^{-2} f_{2k} \right),$$

(3.56)

where each $f_i$ is a positive multivariate polynomial (defined below) in the shearing parameters $\sigma_1, \sigma_2', \ldots, \sigma_i^{(r)}$, where the parameters with even indices have been rescaled, namely $\sigma_j' = \sigma_j g^{-2}$, and all the shearing parameters are fixed by requiring that the measurement procedure effect the gate $\hat{U}_n \cdots \hat{U}_2 \hat{U}_1$ and by the condition that $SV(n)$ is minimized. Then,

$$f_i = \frac{1}{2} \text{tr} \left( \tilde{T}_i^{-1} \Sigma \tilde{T}_i^{-T} \right),$$

(3.57)

where $\tilde{T}_i = T_i T_{i-1} \cdots T_1$, with $T_j = FP(\sigma_j^{(r)})$, where $\sigma_j$ is primed for even $j$’s only. $T_j$ is just $U_j$ after remodelling to a $g = 1$ CVW (by rescaling the shearing parameters). Thus, we have that $T_{j+1} T_j = U_{j+1} U_j$, $\forall j \in \mathbb{N}$. Defining the $f_i$ in terms of rescaled shearing parameters and $T_i$’s suppresses their explicit dependence on $g$.

Eq. (3.56) shows that $SV(n) \to \infty$ as $g \to 0$ when the ideal gate (i.e., in the noiseless limit) is fixed. The $g \to 0$ limit can be understood as the “unconnected cluster limit,” where we expect no information to propagate along the wire. In the large-$g$ limit, only the even $f_i$ terms disappear, in analogy to Eq. (3.21). Also note that in the infinite-squeezing limit ($\varepsilon \to 0$), $SV(n) \to 0$. Thus, $B_K(\kappa) \to \delta(\kappa)$ and hence,

$$\lim_{\varepsilon \to 0} W_{\text{undo}}^{(n)}(x_n) = W^{(0)}(x_n),$$

(3.58)

as required. This analysis confirms what was discussed above, that CVWs with small $g$ weights amplify noise from finite squeezing. Motivated by this, we consider an alternative approach in the next section.

### 3.4 Macronode protocol

Macronode-based computation does not involve converting the DRW into a CVW, and consequently, some features of the computation will differ due to use of the additional DRW structure. We can describe the each macronode by the vector of quadrature
operators corresponding to its constituent physical qumodes
\[ \hat{x}_i = (\hat{q}_{ia}, \hat{q}_{ib}, \hat{p}_{ia}, \hat{p}_{ib})^T, \]  
(3.59)

where \( a \) and \( b \) label distinct physical qumodes comprising the macronode \( i \). To treat the DRW as a double-thick quantum wire, we must define the single-qumode logical subspace within each macronode.

To this end, we define the quadrature operators
\[ \hat{q}_{i\pm} := \frac{1}{\sqrt{2}} (\hat{q}_{ia} \pm \hat{q}_{ib}), \quad \hat{p}_{i\pm} := \frac{1}{\sqrt{2}} (\hat{p}_{ia} \pm \hat{p}_{ib}), \]  
(3.60)

which correspond to the distributed modes labeled + and −. Note that the physical modes and distributed modes represent alternative tensor-product decompositions of the same two-qumode Hilbert space of macronode \( i \). This means that the entanglement structure of a given state will appear different depending on which tensor-product decomposition is used [147]. We label the physical modes as such because they correspond to the particular temporal modes [57, 141] or frequency modes [55, 58, 96] on which the DRW is defined. The distributed modes are so called because they are distributed over the physical ones, either symmetrically (+) or anti-symmetrically (−). The mathematical transformation between the two types of modes is equivalent to a 50/50 beamsplitter interaction.

The logical qumode is defined as the + distributed mode (a.k.a. the + macronode subspace), with quadrature operators \((\hat{q}_{i+}, \hat{p}_{i+})\). This is the natural choice because it allows for simple encoding of input states via 50/50 beamsplitter interaction [141, 145]. As we shall see, macronode computation on such qumodes bears strong resemblance to CV teleportation, as previously pointed out in Ref. [141].

To drive computation on the “+” encoded qumodes, we only need local homodyne measurements on the composite pairs of physical qumodes in the macronode. We refer to such measurements in the following way:
\[ \hat{b}_{i(a,b)} = (c_{\theta_a} \hat{p}_{ia} + s_{\theta_a} \hat{q}_{ia}, \quad c_{\theta_b} \hat{p}_{ib} + s_{\theta_b} \hat{q}_{ib}), \]  
(3.61)

where we use the shorthand \( s_{\theta} := \sin \theta \) and \( c_{\theta} := \cos \theta \). These measurements are written in terms of local homodyne angles \( \theta_a \) and \( \theta_b \), which will be used in what follows. We can describe the effect of these measurements on the input graphically by using both the physical and the distributed modes, as in Fig. 3.3. We see that using distributed modes reduces the description of this process to one that requires only 3 graph nodes.
Figure 3.3: (Color online) Macronode-based computation applies a logical gate to an input encoded in a macronode by measuring the leftmost macronode. Here we assume that the input state is Gaussian (and represent it by the self-loop weight \(z_\psi\)) so that we can describe macronode measurement using the graphical calculus [59]. This is for illustrative purposes only, and the same statements can be made for general input states. The top part of the diagram shows a section of the DRW graph corresponding to the \((a,b)\) macronode decomposition—that is, each node represents a physical qumode. The pair of local homodyne measurements \(\hat{b}_1 a, \hat{b}_1 b\) apply the state transformation \(z_\psi \rightarrow z_\psi'\) (this transformation is described in more detail in Fig. 3.4). On the bottom is the equivalent description using distributed modes (see Eq. (3.60)). Notice that while the measurements in this decomposition are non-local (they are effectively Bell measurements), the input state \(z_\psi\) and output state \(z_\psi'\) are localized on some graph node. Following either set of arrows from the bottom left to the bottom right then shows the graphical evolution of the input state. Also note that on the bottom pair of figures, the bottom right node is merely a spectator. Thus, we can describe this process using distributed modes by a single-qumode input, a two-qumode cluster state (the diagonally joined pair of nodes on the bottom left graph), and Bell measurements (which correspond to local measurements on the physical qumodes). This highlights the similarity between macronode computation and CV teleportation [141].

We then represent the logical effect of a macronode measurement as a quantum circuit in Fig. 3.4.

After a macronode measurement \(\hat{b}_{i(a,b)}\) of the input macronode \(i\) with measurement outcomes \(m_{ia}\) and \(m_{ib}\), the total operation applied to the \(+\) encoded input state, with the result left in macronode \(i+1\) is

\[
|\psi\rangle_{i+} \rightarrow \hat{N}_m \hat{C}_m \hat{V}_i |\psi\rangle_{(i+1),+},
\]

(3.62)

where each of the suboperations—the noise operator \(\hat{N}_m\), the displacement \(\hat{C}_m\), and desired unitary gate \(\hat{V}_i\)—are described below.

The noise operator \(\hat{N}_m\) involves two applications of \(\hat{N}\) (Eq. 3.19) separated by a Fourier transform, which means the noise gets added to both the \(\hat{q}\) and \(\hat{p}\) quadratures. We have that

\[
\hat{N}_m = \hat{N}(\varepsilon_D) \hat{N}_p \left( \frac{\varepsilon_D}{\ell^2} \right),
\]

(3.63)
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

\[|\psi\rangle_{1+}, S(\varepsilon - \frac{1}{2} D) \quad \text{and} \quad |0\rangle_{2+}, S(\varepsilon - \frac{1}{2} D)\]

\[|\psi\rangle_{1+} \quad \text{and} \quad |0\rangle_{2+}, S(\varepsilon - \frac{1}{2} D)\]

**Figure 3.4:** (Color online) Circuit diagram showing a basic element of macronode-based quantum computation. The state immediately after the dotted line is equivalent to a small section of DRW using distributed modes, as shown in Fig. 3.3. In that Figure, \(|\psi\rangle\) is represented by \(z_\psi\) (assumed Gaussian), and the pair of other modes is equivalent to the leftmost diagonally connected pair of nodes. The box labeled \(B^*\) does not represent a physical gate. Instead, it is a change from distributed modes (blue solid circuit wires) to physical modes (red dashed circuit wires)—see Eq. (3.60). Computation proceeds via local physical homodyne measurements \(\hat{b}_1(a,b)\). The bottom qumode remains in the distributed-mode basis, which is why the circuit line remains blue. Alternatively, we can interpret this diagram in a different way: If \(B^*\) is taken to be a physical 50/50 beamsplitter gate, and if the colors and subscript labels are ignored, this diagram shows how to construct a small section of the DRW, inject an input state, and use local macronode measurements to drive MBQC. This reveals that the transformation from physical to distributed modes, Eq. (3.60), is the same as that of the beamsplitter gate used in DRW construction [57]. The squeezing factor acting on the pair of vacuum modes is given by \(\varepsilon^{-1/2}\), and the effective \(\hat{C}_Z\) interaction strength parameter is \(t = \tanh(2\alpha)\), where \(\alpha\) is the overall squeezing parameter [57]. The total operation applied on the logical state is \(\hat{N}_m \hat{C}_m \hat{V}_1\) (see Eqs. (3.63), (3.65), and (3.69)).

where \(t = \tanh(2\alpha)\), and

\[\hat{N}_p(\varepsilon) := \hat{F}^\dagger \hat{N}(\varepsilon) \hat{F} \propto \exp\left(-\frac{-\varepsilon p^2}{2}\right), \quad (3.64)\]

which is just \(\hat{N}(\varepsilon)\) with its behavior exchanged with respect to position and momentum.

In general, \(\hat{N}_m\) adds noise asymmetrically to the quadratures, but in the large-squeezing limit \((t^2 \to 1)\), it is almost symmetric. This is clearly different to the CVW case, where each step introduces noise to one of the quadratures in an alternating fashion (see Eq. (3.19)).

The correction operator

\[\hat{C}_m := \hat{X}(m_q) \hat{Z}(m_p), \quad (3.65)\]
is a phase-space displacement in momentum by $m_p$ followed by one in position by $m_q$, where

\begin{align}
    m_q &:= \sqrt{2}(m_{ib}s_{\theta_{ia}} + m_{ia}s_{\theta_{ib}}) s_{2\phi_i}, \\
    m_p &:= -t\sqrt{2}(m_{ib}c_{\theta_{ia}} + m_{ia}c_{\theta_{ib}}) s_{2\phi_i},
\end{align}

(3.66) (3.67)

which are written in terms of the sum and difference of the local homodyne angles:

$$\theta_{i\pm} := \frac{\theta_{ia} \pm \theta_{ib}}{2}.$$  

(3.68)

Both shifts depend on the actual macronode measurement outcomes $m_{ia}$ and $m_{ib}$, as well as on the choice of observable $\hat{b}_{(a,b)}$. Notice that this is different from the CVW case, in which the correction is solely a position shift and depends only on the measurement outcome and not on the choice of observable (see Eq. (3.18)).

Each macronode measurement implements a gate $\hat{V}_i$ dependent on the two parameters $\theta_{i\pm}$:

$$\hat{V}_i := \hat{S}\left(\frac{1}{t}\right) \hat{R}(\theta_{i+}) \hat{S}(\tan \theta_{i-}) \hat{R}(\theta_{i+}).$$

(3.69)

Recall that $\hat{S}$ is a squeezing operator (Eq. (5.5)), and $\hat{R}$ is a rotation operator (Eq. (5.3)). Since each macronode measurement offers twice as many measurement degrees of freedom per site as the CVW protocol, arbitrary single-qumode Gaussian operations can be completed with just two macronode measurements [141] instead of four individual node measurements, using a section of DRW half as long as the CVW required by the CVW protocol. A proof of this is given in the following subsection.

As a computational unit, a single macronode measurement bears a resemblance to sequential measurement of a pair of CVW qumodes (two CVW protocol measurements). Both procedures offer two measurement degrees of freedom that can be used for gate implementation. Also, they both apply a pair of noise operators and displacements. The noise and displacement operators are separated by a Fourier transform in both cases, which means that noise gets added to both quadratures, and the input states are shifted in both phase-space directions. In order to draw a more quantitative comparison between these protocols, we shall see in the next section how this method compares to the CVW protocol in terms of how much noise is introduced per Gaussian unitary gate. The quantity we compare is the scalar variance, which we derive for the macronode protocol below.
3.4.1 Implementing gates

Here show that only two macronode measurements are required in order to apply an arbitrary Gaussian unitary using the macronode protocol.\(^5\) This was shown in Ref. [141] for \(t = 1\), and we generalize the proof to arbitrary \(t\) (even though it will be later restricted to \(t = \tanh 2\alpha\)).

Define \(R(\phi)\) and \(S(s)\) to be the symplectic matrix representations of \(\hat{R}(\phi)\) (Eq. (5.3)) and \(\hat{S}(s)\) (Eq. (5.5)) respectively. Let

\[
M_i := R(\theta_1^+) S(\tan \theta_1^-) R(\theta_1^+),
\]

where \(\theta_1^\pm\) are defined in Eq. (3.68). Notice the similarity of the definition of the symplectic matrix \(M_i\) to that of the unitary \(\hat{V}_i\) from Eq. (3.69). Since \(\hat{V}_i\) is the symplectic representation of the latter, we can write it in terms of \(M_i\):

\[
\hat{V}_i = S(\frac{1}{t}) M_i.
\]

Furthermore, \(V_i S(\frac{1}{t}) = S(\frac{1}{t}) M_i S(\frac{1}{t})\), and we can incorporate the left- and right-multiplication of \(M_i\) by a squeezing operation into a change of rotation and squeezing parameters:

\[
V_i S(\frac{1}{t}) = S(\frac{1}{t}) M_i S(\frac{1}{t})
= R(\theta_1') S(\tan \theta_1') R(\theta_1'),
\]

where

\[
\theta_1' \pm = \frac{1}{2} \left[ \tan^{-1} \left( \frac{\sigma_{1a}}{t^2} \right) \pm \tan^{-1} \left( \frac{\sigma_{1b}}{t^2} \right) \right],
\]

and \(\sigma_{1(a,b)} = \tan \theta_{1(a,b)}\).

Arbitrary single-qumode Gaussian unitaries can be decomposed into the form \(\hat{R}(\theta) \hat{S}(\eta) \hat{R}(\varphi)\) [89, 145], whose Heisenberg-picture symplectic representation is just \(R(\theta) S(\eta) R(\varphi)\). By using two iterations of \(V_i\) and setting \(\theta_{1-} = \frac{\pi}{4}\), we have

\[
\left. V_2 V_1 \right|_{\theta_{1-} = \frac{\pi}{4}} = V_2 S \left( \frac{1}{t} \right) R(\theta_1+) S(1) R(\theta_1+)
= V_2 S \left( \frac{1}{t} \right) R(2\theta_1+),
\]

\(^5\)For \(t = 1\) this has the same form as the operation introduced as \(M_{eq}\) in Ref. [145]. For this reason we have labeled the matrix in Eq. (3.70) by the letter \(M\).
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

where we have used that $S(1) = I$ and $R(a)R(b) = R(a + b)$. Now, using the definition of $V_2$ in conjunction with Eq. (3.72),

$$V_2 V_1 |_{\theta_1 = \frac{\pi}{4}} = R(\theta_2^+ \eta \tan \theta_2^-) R(\theta_2^+ + 2\theta_1^+) = R(\theta_2^+ \eta \tan \theta_2^- + 2\theta_1^+),$$

(3.75)

Setting $\theta = \theta_2^+, \eta = \tan \theta_2^-$ and $\phi = \theta_2^+ + 2\theta_1^+$, we recover the decomposition of an arbitrary single-qumode Gaussian unitary, as required,

$$V_2 V_1 |_{\theta_1 = \frac{\pi}{4}} = R(\theta)S(\eta)R(\phi).$$

(3.76)

It follows that two macronode measurements are sufficient to implement an arbitrary single-qumode Gaussian unitary.

While the above restriction of setting $\theta_1 = \frac{\pi}{4}$ yields a unique and sufficient decomposition for all single-qumode Gaussian unitaries, it is not the optimal choice of homodyne-measurement angles for the purposes of minimizing noise from finite squeezing over all such unitaries. Nevertheless, we use this decomposition for convenience in our proofs for the relative bounds between the noise for each protocol because it provides an upper bound on the true noise for the optimal decomposition of the macronode protocol.

3.4.2 Scalar variance for the macronode protocol

Now we wish to compare the macronode protocol to the CVW protocol in terms of how much noise is introduced per single-qumode Gaussian unitary gate. We will use the Wigner formalism to define the scalar variance for this protocol, which offers a compact description of the noise and its dependence on which measurements are made. Furthermore, it allows the noise analysis and protocol comparison to apply to arbitrary input states (including mixed inputs). Then, in Sec. 3.3.3, we derive relative bounds between the scalar variances for each protocol, establishing a quantitative comparison.

Consider a small section of the DRW as in Fig. 3.4. The initial state of the DRW in the logical basis has a Wigner function of the form

$$W_m(q_{1+}, p_{1+})W_{CVCS}(q_{1-}, q_{2+}, p_{1-}, p_{2+}),$$

(3.77)
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

where $W_{\text{in}}(x)$ is the Wigner function for the single-qumode input state, and

$$W_{\text{CVCS}}(q_{1-}, q_{2+}, p_{1-}, p_{2+}) := G_{1/2\varepsilon}(q_{1-})G_{2/2\varepsilon}(q_{2+})G_{1\varepsilon}(p_{1-} - t q_{2+})G_{2\varepsilon}(p_{2+} - t q_{1-})$$

(3.78)

is the Wigner function for the two-qumode CVCS.

After measuring $\hat{b}_1(a, b)$ in accordance with Fig. 3.4, applying a displacement $\hat{C}_m^\dagger$ to cancel the measurement-dependent displacement (Eq. (3.65)), and then averaging the output-state Wigner function over measurement outcomes, we get an expression for our output $W_{\text{avg}}(x_{2+})$, which is analogous to Eq. (3.34) for the CVW. Define $V_1$ to be the symplectic matrix representation of the Heisenberg action of $\hat{V}_1$ (Eq. (3.69)). Then,

$$W_{\text{avg}}(x_{2+}) = \int d\eta_1 d\eta_2 W_{\text{in}}(V_1^{-1} x_{2+} + \eta) B \Sigma_m(\eta),$$

(3.79)

where $\eta = V_1^{-1} \tau$,

$$\Sigma_1^m = V_1^{-1} \Sigma_1^* V_1^{-T},$$

(3.80)

and

$$\Sigma_1^* = \begin{pmatrix} \frac{\tau}{\tau} & 0 \\ 0 & \varepsilon \end{pmatrix}.$$  

(3.81)

The matrix $\frac{1}{2} \Sigma_1^*$ is interpreted as the covariance matrix corresponding to a pair of Gaussian convolutions, analogous to Eq. (3.63).

Now we repeat the steps taken in Sec. 3.3.3. By iteration, we can get the general form for the average Wigner function after $n$ measurements ($W_{\text{avg}}^{(n)}(x)$). The $2n$ convolutions can be simplified down to a single bivariate Gaussian convolution with dummy variables $\kappa_1$ and $\kappa_2$. We will also apply $V_1^{-1} \cdots \hat{V}_n^{-1}$, the inverse of the total operation applied in the infinite squeezing limit after $n$ measurements. This results in $W_{\text{undo}}^{(n)}(x_n) = W_{\text{avg}}^{(n)}(\tilde{V}_n x_n)$, where $\tilde{V}_n = V_n V_{n-1} \cdots V_1$. Then,

$$W_{\text{undo}}^{(n)}(x_n) = \int d\kappa_1 d\kappa_2 W_{\text{in}}(x_{(n+1), +} + \kappa) B \Sigma_m^{(n)}(\kappa),$$

(3.82)

where $\kappa = (\kappa_1, \kappa_2)^T$, and

$$\Sigma_n^m = \sum_{i=1}^n \tilde{V}_i^{-1} \Sigma_i^* \tilde{V}_i^{-T}.$$  

(3.83)

Analogous to $\Sigma_n$ in the CVW calculation, $\Sigma_n^m$ reveals how the noise is affected by the number and type of measurement made. Indeed, it is almost exactly the same form as Eq. (3.53), differing by replacing $\Sigma_i$ with $\Sigma_i^*$, which has an additional non-zero diagonal entry corresponding to the second Gaussian convolution in the position quadrature (see
Eq. (3.63)). By defining

$$SV_m(n) := \frac{1}{2} \text{tr}(\Sigma_m^n),$$

(3.84)

we can describe the average effect of noise from finite squeezing from using the macronode protocol after \(n\) measurements and compare it to the CVW protocol scalar variance \(SV(n)\) that we found earlier in Eq. (3.54).

### 3.4.3 Protocol comparison: noise per Gaussian unitary

Having derived expressions for the scalar variance of the CVW and macronode protocols after \(n\) measurements, we can compare each quantity for the number of measurements required to perform an arbitrary Gaussian unitary. For the CVW protocol this is four node measurements. For the macronode protocol this is two macronode measurements.

In each case, we have four degrees of freedom in the measurements to implement a Gaussian unitary, which is described by three degrees of freedom. Thus, we have one degree of freedom \(\theta_{\text{free}}\) left to optimize such that the scalar variance is minimized. For each of the protocols, let \(\hat{E}\) be the desired Gaussian unitary gate, and define the minimum added noise per Gaussian unitary gate by

$$SV(n) := \min_{\theta_{\text{free}}} SV(n),$$

(3.85)
in the case of the CVW protocol, and

$$SV_m(n) := \min_{\theta_{\text{free}}} SV_m(n),$$

(3.86)
in the case of the macronode protocol, where calligraphic font distinguishes this minimized quantity on the left-hand side from the one on the right, which is for a particular gate. It can be evaluated by:

1. Solving the constraint equation for three of the four free homodyne angles.

2. Minimizing the corresponding scalar variance function \(SV_{(m)}(n)\) over the remaining free homodyne angle.

For the CVW, we have \(n = 4\), the constraint equation is \(\mathbf{E} = \tilde{\mathbf{U}}_4\), where \(\mathbf{E}\) is the symplectic representation of Heisenberg action of \(\hat{E}\) (Eq. (3.28)), and the corresponding homodyne angles are \(\theta_1, \theta_2, \theta_3,\) and \(\theta_4\). For the macronode protocol, we have \(n = 2\), the constraint equation is \(\mathbf{E} = \tilde{\mathbf{V}}_2\), and the corresponding homodyne angles are \(\theta_{1a}, \theta_{1b}, \theta_{2a}, \theta_{2b}\).
These quantities represent the minimum noise introduced by finite squeezing per Gaussian unitary gate. Similar work was presented in Ref. [148] for specific examples of gates implemented with four measurements on CVWs. When applied to the CVW protocol, the procedure outlined above generalizes those results to include arbitrary single-qumode Gaussian unitaries implemented using \( n \) CVW node measurements.

We now present the following bound between the minimum scalar variances for the CVW and macronode protocols. This bound is derived in Appendix 3.7. For any \( \hat{E} \), we have that

\[
SV(4) \geq SV_m(2) + \frac{3\varepsilon}{t^2},
\]

where \( SV(4) \) and \( SV_m(2) \) are the minimum scalar variances for implementing the gate \( \hat{E} \) using four CVW measurements and two macronode measurements, respectively (Eq. (3.86)). Hence, the macronode protocol introduces less noise than the CVW protocol per Gaussian unitary gate.

In the next section we will discuss an application of the macronode protocol. It is possible, through a restriction of the single-qumode measurements, to retrieve a CVW-like protocol from macronode computation. We call this restriction the dictionary protocol. If given CVW-protocol measurements that correspond to a desired gate, the dictionary protocol offers a simple translation to macronode measurements, allowing one to apply the same logic gate using the macronode protocol.

### 3.4.4 Dictionary protocol

We showed above how the macronode protocol can be used to implement arbitrary Gaussian unitaries, just like standard MBQC using the CVW. By restricting the allowed local homodyne measurements we can deepen this similarity to the level of how each individual macronode measurement transforms the input state. In other words, we provide a measurement dictionary that applies the same gate as the CVW site for site.

This property provides a direct recipe for adapting CVW measurement protocols for any Gaussian gate or algorithm to a macronode measurement protocol.

There is only one choice for \( \hat{b}_{(a,b)} \) that reduces Eq. (3.69) to a CVW form as in Eq. (3.15). It is the following restriction: set \( \theta_{1a} = \pi/2 \) in Eq. (3.61), or equivalently, at each macronode, measure along the basis \( \hat{b}_d = (\hat{q}_a, \hat{p}_b + \sigma \hat{q}_b) \), where \( \sigma = \tan \theta \). Note that at each macronode, we are restricted to only half the degrees of freedom as in the general macronode protocol. This restriction means that the single-qumode measurements bear a close resemblance to the CVW protocol, where measuring \( \hat{q} \) deletes the top part of the DRW (see Fig. 3.1 (b)). The key difference between the CVW and dictionary protocols
is the encoding of the input state prior to measurement. In the CVW protocol, the input state is encoded on a single CVW node, whereas for the dictionary protocol, it is encoded in the + macronode subspace. Therefore, while the physical qumode measurements have the same form, they cannot be said to have the same effect on the input states.

Under this restriction, Eq. (3.69) reduces to

\[
\hat{V}_i \bigg|_{\theta_a = \frac{\pi}{2}} = \hat{F} \hat{S}(t) \hat{P}(2\sigma_{i-}) =: \hat{W}_i,
\]

where \( t = \tanh 2\alpha \). This is exactly the form as in Eq. (3.15) for a uniform \( g = t \) wire, up to a factor of two in the shearing parameter. However, the noise and correction sub-operations still vary from the CVW case.

Here, the noise operator is the same as for the general macronode protocol, as defined in Eq. (3.63). Note that this does not mean that they introduce the same amounts of noise per Gaussian unitary. That will depend on the measurement bases and the number of measurements made, both of which will vary with the gate implemented.

The measurement-dependent displacement \( \hat{C}_d \) is given by restricting Eq. (3.65) to the case where \( \theta_a = \frac{\pi}{2} \),

\[
\hat{C}_d = \hat{Z} \left(-t\sqrt{2}m_a\right) \hat{X} \left(\frac{\sqrt{2}m_b + \sqrt{2}s_{\theta}m_a}{tc_{\theta}}\right). \tag{3.89}
\]

Contrast the above expression with the CVW case, where the displacement is simply \( \hat{X} \left(\frac{m}{g}\right)\). The displacement for the dictionary protocol involves a displacement in both the position and momentum quadratures, and furthermore, there is a dependence on the measurement basis (since \( \hat{C}_d \) depends on \( \theta \)). For the case where \( \theta = 0 \), the correction operator is similar to the measurement-dependent displacement that occurs in CV quantum teleportation [149].

Because of the equivalence between \( \hat{U}_i \) for the CVW and \( \hat{W}_i \) for the dictionary protocol, we are free to use previous results that apply to the CVW showing that four such measurements are sufficient for all Gaussian unitaries [145].

Given a particular Gaussian unitary and corresponding measurements on the CVW, the dictionary translation rule is

\[
\hat{p} + \sigma_{i-} \hat{q} \mapsto \hat{b}^d_{(a,b)} = \left(\hat{q}_a, \hat{p}_b + \frac{\sigma_{i-} - \hat{q}_b}{2}\right), \tag{3.90}
\]

where the left hand side corresponds to the CVW and the right hand side corresponds to the dictionary protocol. While this applies the same gate to the input state, the
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

noise and correction operators will not translate so simply. A set of measurement bases that minimize the noise from finite squeezing for a particular Gaussian unitary in the CVW case will not necessarily be the optimal choice for the dictionary protocol (and vice versa).

We can also attempt to “remodel” away the effective $g = t$ wire to a $g = 1$ wire in an analogous way to the CVW protocol. Though limited, we show that such a comparison can still be made.

Consider the total operation applied to an arbitrary input state $|\psi\rangle$ by taking $n$ dictionary-protocol macronode measurements. If displacements are ignored, it can be arranged into the following form:

$$\prod_{i=1}^{n} \left[ \hat{N}(\varepsilon_D) \hat{F} \hat{S}(t) \hat{P}(2\sigma_i) \hat{N}(\varepsilon_D) \right] |\psi\rangle,$$  \hspace{1cm} (3.91)

where the ordering in the product is decreasing from $n \rightarrow 1$, left to right. Note that the two noise operators $\hat{N}(\varepsilon_D)$ are separated by a Fourier transform and a squeezing operation. Thus, the noise will be added unequally to the quadratures, as in Eq. (3.63). By commuting squeezing terms to the front and back of each unit (so that they cancel with neighboring terms), we get:

$$\prod_{i=1}^{n} \left[ \hat{S}\left(\frac{1}{\sqrt{t}}\right) \hat{N}\left(\frac{\varepsilon_D}{t}\right) \hat{F} \hat{P}\left(\frac{2\sigma_i}{t}\right) \hat{N}\left(\frac{\varepsilon_D}{t}\right) \hat{S}\left(\sqrt{t}\right) \right] |\psi\rangle.$$  \hspace{1cm} (3.92)

The squeezers in the “bulk” cancel, leaving only squeezing terms from the first and last term:

$$\hat{S}\left(\frac{1}{\sqrt{t}}\right) \hat{N}^{-1}\left(\frac{\varepsilon_D}{t}\right) \left( \prod_{i=1}^{n} \left[ \hat{N}\left(\frac{2\varepsilon_D}{t}\right) \hat{F} \hat{P}\left(\frac{2\sigma_i}{t}\right) \hat{N}\left(\frac{2\varepsilon_D}{t}\right) \hat{S}\left(\sqrt{t}\right) \right] \right) \hat{N}\left(\frac{\varepsilon_D}{t}\right) \hat{S}\left(\sqrt{t}\right) |\psi\rangle,$$  \hspace{1cm} (3.93)

where

$$\hat{N}^{-1}\left(\frac{\varepsilon_D}{t}\right) := \hat{N}\left(-\frac{\varepsilon_D}{t}\right)$$  \hspace{1cm} (3.94)

(up to renormalization of the final state) is the inverse operation to $\hat{N}\left(\frac{\varepsilon_D}{t}\right)$, defined only formally in order to reduce the leftmost $\hat{N}\left(\frac{2\varepsilon_D}{t}\right)$ in the following way: $\hat{N}^{-1}\left(\frac{\varepsilon_D}{t}\right) \hat{N}\left(\frac{2\varepsilon_D}{t}\right) = \hat{N}\left(\frac{\varepsilon_D}{t}\right)$.

We can identify the encoding operation $\hat{E}_d := \hat{N}\left(\frac{\varepsilon_D}{t}\right) \hat{S}\left(\sqrt{t}\right)$. By including the noise operator it makes the structure of each unit (in the square brackets below) be of the

\[^6\text{Note that } \hat{N}^{-1}(\varepsilon)|\psi\rangle \text{ is not in general a normalizable wavefunction for an arbitrary input state } |\psi\rangle. \text{ It is normalizable, however, in the case where } |\psi\rangle \propto \hat{N}(\delta)|\phi\rangle \text{ for some normalizable state } |\phi\rangle \text{ and } \delta > \varepsilon.\]
same form as for the CVW protocol:

$$\hat{E}_{d}^{-1}\left(\prod_{i=1}^{n}\left[\hat{N}\left(\frac{2\varepsilon D}{t}\right)\hat{F}\hat{P}\left(\frac{2\sigma_{i}}{t}\right)\right]\right)\hat{E}_{d}|\psi\rangle,$$

(3.95)

where $\hat{E}_{d}^{-1} = \hat{S}\left(\frac{1}{\sqrt{t}}\right)\hat{N}^{-1}\left(\frac{\varepsilon}{t}\right)$. Hence, up to encoding, this is equivalent to a $g = 1$ wire with $\frac{2\varepsilon}{t}$ self-loop weights. Recall that the $\frac{t}{2}$ weight wire could be remodeled into a weight $g = 1$ wire with self-loop weights $\frac{2\varepsilon D}{t}$ as well (see Eq. (3.25)). Then, up to the encoding and decoding relations, the dictionary protocol and the weight-$\frac{t}{2}$ CVW introduce similar amounts of noise.

Even though the dictionary and CVW protocols appear similar based on the above reasoning, it might be possible to bound the scalar variance of one by the other over all Gaussian unitaries. If this were possible, then the protocol that introduced less noise would be the better choice for implementing Gaussian unitary gates. In Appendix 3.7 we show by counterexample that it is not possible to derive relative bounds on the scalar variances between these protocols over all Gaussian unitary gates. Thus, they can be said to be roughly equivalent, and neither can be said to be optimal in terms of noise per gate.

Finally, defining

$$SV_{d}(n) := SV_{m}(n)\bigg|_{\theta_{1a}=\pi/2,\ldots,\theta_{na}=\pi/2},$$

(3.96)

we can derive a bound between the minimum noise introduced per Gaussian unitary by the dictionary protocol ($SV_{d}(4)$, defined analogously to Eq. (3.86)) and the general macronode protocol:

$$SV_{d}(4) \geq SV_{m}(2) + \frac{\varepsilon_{D}(1 + 2\sqrt{2t})}{t^{2}}.$$ 

(3.97)

The proof of this inequality is given in Appendix 3.7. This inequality shows that in the best case, the noise introduced when applying any Gaussian unitary by the dictionary protocol will always be greater than for the general macronode protocol case. Despite this, the dictionary protocol is still useful because, as its name indicates, it provides a direct, dictionary-like translation from CVW measurement protocols to protocols that can be used on the DRW.

In this section we discussed the key features of the macronode protocol. We demonstrated that it can introduce less noise per gate than the CVW protocol over all Gaussian unitaries. Minimizing the noise from finite squeezing is an important feature of any measurement-based scheme for quantum computing using CVCSs [145]. Thus, this result shows the importance of considering how measurements are used to implement
Chapter 3: Noise analysis of single-qudmode Gaussian operations using continuous-variable cluster states

unitary gates, and it highlights the benefits of a macronode-based approach. Furthermore, we also showed that the macronode protocol saves on resource overhead by only requiring half as much DRW length as the CVW protocol to implement arbitrary single-qudmode Gaussian unitaries. We also introduced the dictionary protocol, which acts as a translation rule for running CVW algorithms using macronode-based measurements on the DRW. While it introduced more noise per gate than the macronode protocol, it was found to be roughly equivalent to the CVW protocol while maintaining a deep similarity to the CVW in the structure of the measurements used to implement a given gate. In the next section, we use our analysis of the CVW, macronode, and dictionary protocols to compare their performance as the number of measurement degrees of freedom used to implement each Gaussian unitary is varied.

3.5 Application: Number of measurements per gate

In the discussion of measurement protocols above, we have assumed that four measurement degrees of freedom are available per Gaussian unitary gate. This might seem surprising given that an arbitrary single-qudmode Gaussian unitary is specified (up to displacements) by three parameters [145]. Nevertheless, there exists a small set (of measure zero) of single-qudmode gates that cannot be achieved by three CVW measurements [145]. Furthermore, it is claimed without proof in Ref. [145] that one cannot even get close to these forbidden gates without the noise due to finite squeezing becoming arbitrarily large. In this section we will explore this notion in a systematic way, applying the noise analysis framework from the previous sections to the three- and four-measurement CVW and dictionary protocols. Our analysis shows why protocols using fewer than four measurements are inadequate for implementing arbitrary single-qudmode gates, even in some approximate sense.

We must be careful when choosing gates to analyze since large noise can also result from trying to implement a gate with more squeezing than is available in the original CV cluster state, a fact made rigorous in Appendix 3.8. In order to isolate the effect we wish to show (noise from too few measurements) from high-squeezing noise (which will occur regardless of the number of measurements), we restrict our analysis to phase space rotations $\hat{R}(\theta)$, which do not contain any squeezing—i.e., when decomposed as $\hat{R}(\theta) = \hat{R}(\phi)S(\eta)\hat{R}(\varphi)$, the squeezing parameter $\ln \eta = 0$.

Generically, we expect that using a larger number of cluster measurements per gate will introduce more noise from finite squeezing [60–62]. We show that this holds true for some gates. However, we find that for a large class of gates, this intuition breaks: less noise is introduced when implementing with four measurements than with three. This
has been pointed out for a few specific gates on a particular cluster state of experimental interest [148]. Here we show this to be true for a large class of rotation gates implemented by the CVW and macronode protocols.

### 3.5.1 CVW protocol

On the CVW, the scalar variance after the three-measurement implementation of a rotation gate \( \hat{R}(\theta) \)—called \( SV_{\hat{R}(\theta)}(3) \) and defined below—is unique since the constraint \( \hat{R}(\theta) = \hat{U}_3 \hat{U}_2 \hat{U}_1 \) uniquely specifies all the measurement degrees of freedom. As there are no degrees of freedom to minimize over, \( SV(3) \) can be used interchangeably with \( SV(3) \) (see Eq. (3.55)). Then, using the results of Sec. 3.3.3, we have

\[
SV_{\hat{R}(\theta)}(3) = \frac{1}{2} \sum_{i=1}^{3} \text{tr} \left[ \left( \tilde{U}_i^{3R}(\theta) \right)^{-1} \Sigma_* \left( \tilde{U}_i^{3R}(\theta) \right)^{-T} \right], \tag{3.98}
\]

where \( U_i^{3R}(\theta) \) is just \( U_i \) constrained by \( \tilde{U}_3 = R(\theta) \).

For the four-measurement implementation, the analogous quantity to Eq. (3.98) has one free measurement degree of freedom. We denote the minimum scalar variance for rotation gates by calligraphic font, \( SV_{\hat{R}(\theta)}(4) \), where the minimization is over the one free measurement angle \( \theta_{\text{free}} \), as in Eq. (3.86). Then,

\[
SV_{\hat{R}(\theta)}(4) = \min_{\theta_{\text{free}}} \frac{1}{2} \sum_{i=1}^{4} \text{tr} \left[ \left( \tilde{U}_i^{4R}(\theta) \right)^{-1} \Sigma_* \left( \tilde{U}_i^{4R}(\theta) \right)^{-T} \right], \tag{3.99}
\]

where \( U_i^{4R}(\theta) \) is just \( U_i \) constrained by \( \tilde{U}_4 = R(\theta) \) and such that the scalar variance is minimized.

In Fig. 3.5 we plot these scalar variances for arbitrary rotations by \( \theta \) using three (dashed blue) and four (red) measurements on the CVW. Notice that the noise diverges as a function of angle for the three-measurement case but not for four. This behaviour is generic for all levels of squeezing. In fact, the divergences exactly correspond to those rotation gates that cannot be implemented by the three-measurement CVW protocol [145]. Note that there are some values of \( \theta \) (such as \( \theta = \pi \)) for which \( SV_{\hat{R}(\theta)}(3) < SV_{\hat{R}(\theta)}(4) \). Thus, there exist instances where applying the three macronode protocol is more efficient than with four measurements—and therefore it could be leveraged to minimize noise further under certain conditions—even though it is clear from Fig. 3.5 that this cannot be the general rule.
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Figure 3.5: (Color online) Minimum scalar variance per rotation gate for three and four measurements on the CVW, $\hat{S}V_{R(\theta)}(3)$ (dashed blue) and $\hat{S}V_{R(\theta)}(4)$ (solid red). Technically, only $\hat{S}V_{R(\theta)}(4)$ has been minimized (represented by the caligraphic font) as $\hat{S}V_{R(\theta)}(3)$ is unique (it has no free measurement degree of freedom). Units on the vertical axis are such that the vacuum variance is 1/2. Although the three-measurement protocol introduces the least noise at some particular $\theta$, for $\theta$ in the vicinity of $\pi/2$ or $3\pi/2$, which correspond to gates that cannot be implemented by three CVW measurements [145], the noise becomes arbitrarily large. On the other hand, the noise for the four-measurement protocol remains bounded for all $\theta$. In this plot, the squeezing parameter is $\alpha = 0.5756$, corresponding to 5 dB of squeezing ($\# \text{ dB} = 10 \log_{10} e^{2\alpha}$), approximately the levels achieved in Ref. [141].

3.5.2 Dictionary protocol

We can also consider implementing rotations through three and four macronode measurements using the dictionary protocol. Analogous to the CVW protocol case above, we shall denote the scalar variance for a three-measurement implementation of a rotation gate as $\hat{S}V_{d,R(\theta)}(3)$ by using Eq. (3.96). As with the CVW protocol, this scalar variance is uniquely determined by the rotation angle $\theta$, and therefore, we can use $\hat{S}V_{d}(3)$ and its minimized counterpart $\hat{S}V_{d}(3)$ interchangeably (in the three-measurement case they represent the same quantity). Thus,

$$\hat{S}V_{d,R(\theta)}(3) = \frac{1}{2} \sum_{i=1}^{3} \text{tr} \left[ \left( \tilde{W}_{i}^{3R}(\theta) \right)^{-1} \Sigma_{i}^{*} \left( \tilde{W}_{i}^{3R}(\theta) \right)^{-T} \right], \quad (3.100)$$

where $\tilde{W}_{i}^{3R}(\theta)$ is just $\tilde{W}_{i}$ constrained by $\tilde{W}_{3} = R(\theta)$, and where $W_{i}$ is the symplectic matrix representation of the Heisenberg action of $W_{i}$. Just as in the four-measurement CVW protocol case, the scalar variance for the four-macronode measurement implementation ($\hat{S}V_{d,R(\theta)}(4)$) has a free measurement degree of freedom. Denote the minimum
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

scalar variance for rotation gates by using calligraphic font, $\mathcal{SV}_{d,R(\theta)}$, where the minimization is over the free measurement degree of freedom ($\theta_{\text{free}}$). Then,

$$
\mathcal{SV}_{d,R(\theta)}(4) = \min_{\theta_{\text{free}}} \frac{1}{2} \sum_{i=1}^{4} \text{tr} \left[ \left( \tilde{W}_i R(\theta) \right)^{-1} \Sigma_i^* \left( \tilde{W}_i^T R(\theta) \right)^{-T} \right],
$$

where the calligraphic font denotes minimization over the free measurement degree of freedom, and $\tilde{W}_i$ constrained by $\tilde{W}_4 = R(\theta)$ such that the scalar variance is minimized.

Shown in Fig. 3.6 are the scalar variances for rotation gates as a function of angle for three macronode measurements (the dashed blue line) and four macronode measurements (the solid red line) using the dictionary protocol. There is a striking similarity between the three- and four-measurement scalar variance rotation plots in Figs. 3.5 and 3.6. Like in the CVW case, the three-measurement dictionary protocol diverges for certain values of $\theta$. In fact, these are the same values of $\theta$ as the ones that have diverging noise in the CVW case. This connection is expected because, as its name implies, the dictionary protocol is a node-for-node mapping of the CVW protocol to macronodes. As such, gates that cannot be applied with three measurements in the CVW case [145] should similarly fail in the dictionary case, and the noise of both protocols should diverge as one tries to implement gates that are arbitrarily close to them. This is exactly what we see.

Note that there exist other values of $\theta$ for which $\mathcal{SV}_{d,R(\theta)}(3) < \mathcal{SV}_{d,R(\theta)}(4)$. Therefore, the three-measurement protocol could be applied in certain cases to minimize the noise per gate further than what is possible with four measurements, even though the existence of divergences for certain angles rules out its use for general single-qumode Gaussian unitaries.

Also shown in Fig. 3.6 (and defined below) is the scalar variance of a two-macronode-measurement implementation of a rotation gate as a function of angle and using the suboptimal shearing parameters from Eq. (3.74) (dot-dashed black line). Denote this as

$$
\mathcal{SV}_{m,R(\theta)}^{\text{so}}(2) = \frac{1}{2} \sum_{i=1}^{2} \text{tr} \left[ \left( \tilde{V}_i^{2R(\theta)} \right)^{-1} \Sigma_i^* \left( \tilde{V}_i^{2R(\theta)} \right)^{-T} \right].
$$

By the cyclic property of the trace, for all rotations

$$
\mathcal{SV}_{m,R(\theta)}^{\text{so}}(2) = \text{tr}[\Sigma_i^*].
$$

By comparing Figs. 3.5 and 3.6, we observe that the general macronode protocol introduces less than half as much noise for rotation gates as either the CVW or the dictionary
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Figure 3.6: (Color online) Minimum scalar variance per rotation gate using the dictionary protocol for three \( (SV_{d,R(\theta)}(3), \text{dashed blue}) \) and four measurements \( (SV_{d,R(\theta)}(4), \text{solid red}) \), as well as the general macronode protocol \( (SV_{m,R(\theta)}^m(2), \text{dot-dashed black}) \) for two measurements. These are plotted as a function of \( \theta \), which specifies the rotation applied to the state. Technically, only \( SV_{d,R(\theta)}(4) \) has been minimized (represented by caligraphic font) as \( SV_{d,R(\theta)}(3) \) is unique (it has no free measurement degrees of freedom), and \( SV_{m,R(\theta)}^m(2) \) is an upper bound on the minimum for the macronode protocol. Units on the vertical axis are such that the vacuum variance is 1/2. The variance added by the general macronode protocol is upper-bounded by the black line (corresponding to the suboptimal solution from Eq. (3.74)) that does not vary with angle. Although the three-macronode-measurement protocol introduces the least noise at some particular values of \( \theta \), for \( \theta \) in the vicinity of \( \frac{\pi}{2} \) or \( \frac{3\pi}{2} \), the noise diverges. These angles correspond to gates that cannot be implemented by three CVW measurements [145] (also expected here because the dictionary protocol is a node-for-node adaptation of the CVW protocol). The noise in the vicinity of these gates becomes arbitrarily large. On the other hand, the noise for the four-measurement protocol remains bounded for all \( \theta \). In this plot, the squeezing parameter is \( \alpha = 0.5756 \), corresponding to 5 dB of squeezing \( (\# \text{ dB} = 10 \log_{10} e^{2\alpha}) \), approximately the levels achieved in Ref. [141].

protocol using four measurements. It also outperforms the three-measurement versions of both protocols, albeit by a lesser margin for some angles.

We did not consider the possibility of a three-measurement case for the general macronode protocol here because the measurement degrees of freedom are grouped pairwise per macronode measurement. Therefore, the notion of a three-measurement protocol does not translate clearly.

In summary, it was claimed without proof in Ref. [145]—and shown for specific cases in Ref. [148]—that using four measurements instead of three in a CV cluster-state protocol reduces the overall noise due to finite squeezing because there is an additional measurement degree of freedom that can be adjusted in order to minimize the noise. We have shown this to be true for most rotation gates using the CVW and dictionary protocols. Nevertheless, we have also shown that there are cases where the three-measurement
protocol performs better than its four-measurement counterpart, as shown in Figs. 3.5 and 3.6. Therefore, three-measurement protocols could be suitable for implementing gates with less noise in certain cases. Finally, we found that implementing rotations with just two measurements using the macronode protocol performs better than either of the other two protocols for both three and four measurements.

3.6 Conclusions and Discussion

We have considered two different approaches to implementing single-qumode Gaussian computation using the dual-rail quantum wire (DRW) resource. We characterized the noise properties of a class of approximate continuous-variable quantum wires (CVWs) in terms of graphical parameters $g$ and $\varepsilon$, which are edge and self-loop weights on the CVW graph respectively [59]. We discussed how the value of $g$ affected the logical unitary and noise applied at each measurement and how to modify the measurement protocol on CVWs in order to treat a CVW with uniform weights $\{g, \varepsilon\}$ as if it were one with uniform weights $\{1, \varepsilon g^{-1}\}$ instead. This allows us to parameterize the class of uniform CVWs by a single parameter $\varepsilon g^{-1}$.

We introduced the macronode protocol for the DRW and proved that it introduces less noise from finite squeezing per single-qumode Gaussian unitary than the CVW protocol. Furthermore, it uses wires of half the length to do so. However, by itself it is by no means a cure for this noise, and it must be combined with other techniques such as active error correction to achieve fault-tolerant quantum computation [60–62, 87].

Our noise analysis was able to compare a variety of different measurement protocols in terms of the newly defined quantity, the scalar variance $SV(n) = \frac{1}{2} \text{tr}[\Sigma]$. This allowed us to quantify the advantage of using the macronode protocol over the CVW protocol for arbitrary Gaussian unitaries. While we applied this formalism specifically for the choice of parameters given by the temporal-mode linear-optics method [57, 141] and the single-OPO method [55, 58], these results are general and can be applied directly to arbitrary weight-$g$ wires over an arbitrary number of linear quadrature measurements.

These results should be extendable to a broader class of continuous-variable cluster states, such as states with a 2D square-lattice graph structure [56, 58] or even higher-dimensional structures such as the hypercubic lattice [58].

The dictionary protocol provides a theoretical link between the CVW and the DRW, which could potentially be extended to map across other cluster-state features, such as conversion to toric code states [150, 151]. In this Article, we have only considered single-qumode Gaussian unitaries on the DRW, but we anticipate that a similar analysis could be performed for the quad-rail resource state discussed in Ref. [57] since homodyne
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

detection on that resource enables multi-qumode Gaussian unitaries. This extension is left to future work.

3.7 Appendix: Comparative noise bounds between the measurement protocols

We did not optimize the macronode protocol in deriving these bounds. Instead we used the suboptimal parameters chosen in Eq. (3.74) to simplify the calculation. This only provides an upper bound on the minimum macronode-protocol scalar variance.

The first bound we derive compares the CVW protocol and the general macronode protocol.

3.7.0.1 Comparing the CVW and macronode protocols

Let the desired Gaussian unitary be denoted by $\hat{E}$. Then define $E$ to be the symplectic matrix representation of its Heisenberg action.

Using Eq. (3.54) and recalling $U_i$ is the symplectic matrix representation of $\hat{U}_i$ (Eq. (3.32)), we expand the scalar variance after four measurements for the CVW protocol:

$$SV(4) = \frac{1}{2} \text{tr} \left[ \hat{U}_4^{-1} \Sigma_{\hat{U}_4} \hat{U}_4^{-T} \right] + \frac{1}{2} \text{tr} \left[ \hat{U}_3^{-1} \Sigma_{\hat{U}_3} \hat{U}_3^{-T} \right] + \frac{1}{2} \text{tr} \left[ \hat{U}_2^{-1} \Sigma_{\hat{U}_2} \hat{U}_2^{-T} \right] + \frac{1}{2} \text{tr} \left[ \hat{U}_1^{-1} \Sigma_{\hat{U}_1} \hat{U}_1^{-T} \right],$$

(3.104)

where $\hat{U}_i = U_i U_{i-1} \cdots U_1$. Recall that $\Sigma^* = \begin{pmatrix} \Sigma & 0 \\ 0 & \epsilon \end{pmatrix}$ represents the covariance matrix of the bivariate Gaussian convolution that arises in the Wigner-function description of a single macronode measurement. Recall the definition of $\Sigma^*$ from Eq. (3.51), and define

$$T(E) := \frac{1}{2} \text{tr} [E^{-T} \Sigma^* E^{-1}],$$

(3.105)

we can expand the first term in Eq. (3.104) by applying the constraint equation $E = \hat{U}_4$ and observing that $\Sigma^* = \Sigma + \frac{1}{2} \Sigma^*$. Then,

$$\frac{1}{2} \text{tr} \left[ \hat{U}_4^{-1} \Sigma_{\hat{U}_4} \hat{U}_4^{-T} \right] = T(E) - \frac{1}{2\epsilon^2} \text{tr} \left[ \hat{U}_4^{-1} \Sigma_{\hat{U}_4} \hat{U}_4^{-T} \right]$$

$$= T(E) - \frac{1}{8} \text{tr} \left[ \hat{U}_3^{-1} \Sigma_{\hat{U}_3} \hat{U}_3^{-T} \right],$$

(3.106)
where we have used the following in order to get the second equality:

\[
\text{tr}\left[\tilde{U}_4^{-1}\Sigma^*\tilde{U}_4^{-T}\right] = \text{tr}\left[\tilde{U}_3^{-1}U_4^{-1}\Sigma^*U_4^{-T}\tilde{U}_3^{-T}\right], \tag{3.107}
\]

and

\[
U_4^{-1}\Sigma^*U_4^{-T} = P(-\sigma_4) \mathbf{S}\left(\frac{2}{t}\right) F^{-1}\Sigma^*F \mathbf{S}\left(\frac{2}{t}\right) P(-\sigma_4)^T = \frac{t^2}{4} P(-\sigma_4) \Sigma_s P(-\sigma_4)^T = \frac{t^2}{4} \Sigma_s. \tag{3.108}
\]

Then, by substituting Eq. (3.106) into Eq. (3.104),

\[
SV(4) = T(E) + \frac{3}{8} \text{tr}\left[U_3^{-1}\Sigma_3 U_3^{-T}\right] + \frac{1}{2} \text{tr}\left[U_2^{-1}\Sigma_2 U_2^{-T}\right] + \frac{1}{2} \text{tr}\left[U_1^{-1}\Sigma_1 U_1^{-T}\right]. \tag{3.109}
\]

Next we minimize the last two terms with respect to the shearing parameters. The minimum occurs when \(\sigma_1 = \sigma_2 = 0\). Then,

\[
\min_{\sigma_1,\sigma_2} \left(\text{tr}\left[U_2^{-1}\Sigma_2 U_2^{-T}\right] + \text{tr}\left[U_1^{-1}\Sigma_1 U_1^{-T}\right]\right) = \text{tr}\left[\left(\mathbf{S}\left(\frac{2}{t}\right) F^{-1}\right)^2 \Sigma_s \left(\mathbf{F} \mathbf{S}\left(\frac{2}{t}\right)\right)^2\right] + \text{tr}\left[\mathbf{S}\left(\frac{2}{t}\right) F^{-1}\Sigma_s F \mathbf{S}\left(\frac{2}{t}\right)\right]. \tag{3.110}
\]

Then, noting that \(\left(\mathbf{S}\left(\frac{2}{t}\right) F^{-1}\right)^2 = -\mathbf{I}\) and

\[
\text{tr}\left[\mathbf{S}\left(\frac{2}{t}\right) F^{-1}\Sigma_s F \mathbf{S}\left(\frac{2}{t}\right)\right] = \frac{4}{t^2} \text{tr}[\Sigma^*], \tag{3.111}
\]

we arrive at

\[
\text{tr}\left[\tilde{U}_2^{-1}\Sigma_2 \tilde{U}_2^{-T}\right] + \text{tr}\left[\tilde{U}_1^{-1}\Sigma_1 \tilde{U}_1^{-T}\right] \geq \text{tr}[\Sigma_s] + \frac{4}{t^2} \text{tr}[\Sigma^*] = \text{tr}[\Sigma_s^*] + \frac{3}{t^2} \text{tr}[\Sigma^*]. \tag{3.112}
\]

Now consider the macronode protocol, measuring out 2 macronodes. Applying the constraint

\[
E = \tilde{V}_2, \tag{3.113}
\]
where $\tilde{V}_2 = V_2 V_1$, the scalar variance $SV_m(2)$ can be written out explicitly as

$$SV_m(2) = T(E) + \frac{1}{2} \text{tr} \left[ \tilde{V}_1^{-1} \Sigma^* \tilde{V}_1^{-\top} \right]. \tag{3.114}$$

The minimum scalar variance $SV_m(2)$ is bounded from above by the scalar variance for the suboptimal solution $SV_{m}^{\text{so}}(2)$ that uses the choice of homodyne angles in Eq. (3.74). Now,

$$SV_{m}^{\text{so}}(2) = \frac{1}{2} \text{tr} \left[ E^{-\top} \Sigma^* E^{-1} \right] + \frac{1}{2} \text{tr} \left[ V_1^{-\top} \Sigma^* V_1^{-1} \right] = T(E) + \frac{1}{2} \text{tr} [\Sigma^*], \tag{3.115}$$

where we have used the cyclic property of the trace and the fact that $R^T(\theta_1) = R^{-1}(\theta_1)$ in the second equality, as well as $\frac{1}{2} \text{tr} \left[ S \left( \frac{1}{2} \right) \Sigma^* S \left( \frac{1}{2} \right) \right] = \frac{1}{2} \text{tr} [\Sigma^*]$ and Eq. (3.105). Applying Eqs. (3.115) and (3.112) to Eq. (3.109), we have

$$SV(4) \geq SV_m^{\text{so}}(2) + \frac{3}{2t^2} \text{tr}[\Sigma^*] + \frac{3}{8} \text{tr} \left[ \tilde{U}_3^{-1} \Sigma \tilde{U}_3^{-\top} \right]. \tag{3.116}$$

As the only constraint that has been placed on the shearing parameters (or equivalently the homodyne angles) is $\tilde{U}_4 = E = \tilde{V}_2$, we are free to replace $SV(4)$ with the scalar variance that has been minimized over the one free measurement angle, $SV(4)$. The last term is minimized when $\sigma_1 = \sigma_2 = \sigma_3 = 0$. Making a substitution for these parameters yields

$$SV(4) \geq SV_m^{\text{so}}(2) + \frac{3\epsilon_D}{t^2}, \tag{3.117}$$

from which it follows that

$$SV(4) \geq SV_m(2) + \frac{3\epsilon_D}{t^2}. \tag{3.118}$$

Hence the macronode protocol introduces less noise than the CVW protocol.

### 3.7.0.2 Comparing the dictionary and general macronode protocols

Again, denote the desired Gaussian unitary by $\hat{E}$. Measuring the $i$th macronode on the DRW by the dictionary protocol (in the $\hat{q}_{ia}$ and $\hat{p}_{ib} + \sigma_i \hat{q}_{ib}$ bases) applies a Gaussian unitary with symplectic matrix

$$W_i = \begin{bmatrix} -\frac{2\sigma_i}{t} & -\frac{1}{t} \\ \frac{1}{t} & 0 \end{bmatrix}. \tag{3.119}$$

We will consider the scalar variance after four measurements $SV_d(4)$ with the requirement that

$$\tilde{W}_4 = E, \tag{3.120}$$

where $\tilde{W}_4$ is a symplectic matrix that represents the noise introduced by the macronodes. The scalar variance $SV_d(4)$ can be written out explicitly as

$$SV_d(4) = T(E) + \frac{1}{2} \text{tr} \left[ \tilde{V}_1^{-1} \Sigma^* \tilde{V}_1^{-\top} \right]. \tag{3.121}$$

The minimum scalar variance $SV_d(4)$ is bounded from above by the scalar variance for the suboptimal solution $SV_{d}^{\text{so}}(4)$ that uses the choice of homodyne angles in Eq. (3.88). Now,

$$SV_{d}^{\text{so}}(4) = \frac{1}{2} \text{tr} \left[ E^{-\top} \Sigma^* E^{-1} \right] + \frac{1}{2} \text{tr} \left[ V_1^{-\top} \Sigma^* V_1^{-1} \right] = T(E) + \frac{1}{2} \text{tr} [\Sigma^*], \tag{3.122}$$

where we have used the cyclic property of the trace and the fact that $R^T(\theta_1) = R^{-1}(\theta_1)$ in the second equality, as well as $\frac{1}{2} \text{tr} \left[ S \left( \frac{1}{2} \right) \Sigma^* S \left( \frac{1}{2} \right) \right] = \frac{1}{2} \text{tr} [\Sigma^*]$ and Eq. (3.105). Applying Eqs. (3.122) and (3.120) to Eq. (3.119), we have

$$SV_d(4) \geq SV_{d}^{\text{so}}(4) + \frac{3}{2t^2} \text{tr}[\Sigma^*] + \frac{3}{8} \text{tr} \left[ \tilde{U}_3^{-1} \Sigma \tilde{U}_3^{-\top} \right]. \tag{3.123}$$

As the only constraint that has been placed on the shearing parameters (or equivalently the homodyne angles) is $\tilde{U}_4 = E = \tilde{V}_2$, we are free to replace $SV_d(4)$ with the scalar variance that has been minimized over the one free measurement angle, $SV_d(4)$. The last term is minimized when $\sigma_1 = \sigma_2 = \sigma_3 = 0$. Making a substitution for these parameters yields

$$SV_d(4) \geq SV_{d}^{\text{so}}(4) + \frac{3\epsilon_D}{t^2}, \tag{3.124}$$

from which it follows that

$$SV_d(4) \geq SV_d(2) + \frac{3\epsilon_D}{t^2}. \tag{3.125}$$

Hence the macronode protocol introduces less noise than the CVW protocol.
where $\tilde{W}_i = \tilde{W}_i \tilde{W}_{i-1} \cdots \tilde{W}_1$, and $E$ is the symplectic matrix representation of the Heisenberg action of $\tilde{E}$. The scalar variance can be written out explicitly as

$$SV_d(4) = T(E) + \frac{1}{2} \text{tr} \left[ \tilde{W}_3^{-1} \Sigma^*_3 \tilde{W}_3^{-T} \right] + \frac{1}{2} \text{tr} \left[ \tilde{W}_2^{-1} \Sigma^*_2 \tilde{W}_2^{-T} \right] + \frac{1}{2} \text{tr} \left[ \tilde{W}_1^{-1} \Sigma^*_1 \tilde{W}_1^{-T} \right]. \quad (3.121)$$

Note that

$$\min_{\sigma_1 \in \mathbb{R}} \text{tr} \left[ \tilde{W}_1^{-1} \Sigma^*_1 \tilde{W}_1^{-T} \right] = \text{tr} [\Sigma^*], \quad (3.122)$$

which is just saying that the minimum noise introduced after one dictionary macronode measurement corresponds to measuring $\hat{q}_{1a}$ and $\hat{p}_{1b}$.

Then, combining the above with Eq. (3.115), and using the suboptimal scalar variance $SV^\text{so}_m(2)$ (using suboptimal homodyne angles as in Eq. (3.74)) as an upper bound on $SV_m(2)$, we have the following relation:

$$SV_d(4) \geq SV^\text{so}_m(2) + \frac{1}{2} \text{tr} \left[ \tilde{W}_3^{-1} \Sigma^*_3 \tilde{W}_3^{-T} \right] + \frac{1}{2} \text{tr} \left[ \tilde{W}_2^{-1} \Sigma^*_2 \tilde{W}_2^{-T} \right]. \quad (3.123)$$

Now, what is the minimum value that the last two terms can take? By minimizing those terms over shearing parameters $\sigma_1$, $\sigma_2$, and $\sigma_3$, irrespective of the level of squeezing,

$$\frac{1}{2} \text{tr} \left[ \tilde{W}_3^{-1} \Sigma^*_3 \tilde{W}_3^{-T} \right] + \frac{1}{2} \text{tr} \left[ \tilde{W}_2^{-1} \Sigma^*_2 \tilde{W}_2^{-T} \right] \geq \frac{\varepsilon D (1 + 2\sqrt{2t})}{t^2}. \quad (3.124)$$

Then, using $SV^\text{so}_m(2) \geq SV_m(2)$,

$$SV_d(4) \geq SV_m(2) + \frac{\varepsilon D (1 + 2\sqrt{2t})}{t^2}. \quad (3.125)$$

which is our bound. This bound holds for all dictionary-protocol shearing parameters such that Eqs. (3.120) and (3.113) are satisfied. In particular it holds for the shearing parameters that correspond the the minimum scalar variance, $SV_d(4)$. Then,

$$SV_d(4) \geq SV_m(2) + \frac{\varepsilon D (1 + 2\sqrt{2t})}{t^2}. \quad (3.126)$$

Thus, we have proved that the dictionary protocol introduces more noise than the two-measurement macronode protocol over all single-qumode Gaussian unitaries.

### 3.7.0.3 Comparing the dictionary protocol to the CVW protocol

Unfortunately, there is no comparative bound on the noise between the CVW and the dictionary protocol over all Gaussian unitaries. A simple argument for this is to consider a pair of examples. In one case, the dictionary protocol introduces the least noise, and
in the other, the CVW protocol does. Consider the identity operation $E = I$. The dictionary protocol introduces $\frac{2\varepsilon_D (1+t^2)}{t}$ units of noise compared to $\frac{\varepsilon_D (4+t^2)}{t^2}$ for the CVW protocol. In the large-squeezing limit ($t \to 1$), the dictionary protocol introduces less noise:

\[
(\text{dictionary}) \ 4\varepsilon_D \leq 5\varepsilon_D \ (\text{CVW}).
\] (3.127)

For the operation $E = R(\pi)S(2)$, the dictionary protocol introduces $\frac{55\varepsilon_D}{8}$ units of noise versus $\frac{\varepsilon_D (5+12t+8t^2)}{4t^2}$ for the CVW protocol. In the large-squeezing limit,

\[
(\text{dictionary}) \ \frac{55\varepsilon_D}{8} \geq \frac{25\varepsilon_D}{4} \ (\text{CVW}),
\] (3.128)

and thus, there exist gates for which one protocol is more efficient than the other, and vice versa. This shows that the CVW protocol and the dictionary protocol are roughly equivalent.

### 3.8 Appendix: Proof of scalar variance divergences for gates with high squeezing

Here we prove that, for each of the measurement protocols, any gate that requires high levels of squeezing (relative to that present in the initial CVCS) will introduce a large amount of noise, regardless of how many measurements are used to implement the gate.

Let $\hat{E}$ be an arbitrary single-qumode Gaussian unitary, and let $E$ denote the symplectic matrix representation of its Heisenberg action on the vector of quadrature operators. We can decompose $E$ as a squeezing matrix $S(\eta)$ sandwiched between a pair of rotation matrices $R(\theta)$ and $R(\varphi)$ [89, 145]:

\[
E = R(\theta)S(\eta)R(\varphi).
\] (3.129)

Consider the scalar variances for the CVW protocol and the macronode protocol (proof for the dictionary protocol is analogous) with respect to an $n$-measurement implementation of the gate $\hat{E}$. For the CVW protocol, assume that $n \geq 4$, which is the minimum number of measurements required to implement any single-qumode Gaussian unitary [145]. Then,

\[
SV(n) = \frac{1}{2} \sum_{i=1}^{n} \text{tr} \left[ \hat{U}_i^{-1} \Sigma_i \hat{U}_i^{-\dagger} \right].
\] (3.130)
Chapter 3: Noise analysis of single-qumode Gaussian operations using continuous-variable cluster states

Note that $\text{tr} \left[ \tilde{\mathbf{U}}_i^{-1} \Sigma_i \tilde{\mathbf{U}}_i^{-T} \right] > 0$ for all $i$. Therefore,

$$SV(n) > \frac{1}{2} \text{tr} \left[ \tilde{\mathbf{U}}_{n-1}^{-1} \Sigma_{n-1} \tilde{\mathbf{U}}_{n-1}^{-T} \right] + \frac{1}{2} \text{tr} \left[ \tilde{\mathbf{U}}_n^{-1} \Sigma_n \tilde{\mathbf{U}}_n^{-T} \right].$$  \hspace{1cm} (3.131)

Now, the requirement that the gate $\hat{E}$ is applied by $n$ CVW measurements boils down to demanding that $\tilde{\mathbf{U}}_n = \mathbf{E}$. This implies that $\tilde{\mathbf{U}}_{n-1} = \mathbf{U}_{n-1}^{-1} \mathbf{E}$. Then,

$$SV(n) > \frac{1}{2} \text{tr} \left[ \mathbf{E}^{-1} \mathbf{U}_n \Sigma_n \mathbf{U}_n^T \mathbf{E}^{-T} \right] + \frac{1}{2} \text{tr} \left[ \mathbf{E}^{-1} \Sigma_n \mathbf{E}^{-T} \right].$$  \hspace{1cm} (3.132)

From Eq. (3.32) and Eq. (3.51), we have $\mathbf{U}_j \Sigma_j \mathbf{U}_j^T = g \Sigma^*$, which depends only on the uniform self-loop and edge weights, $\varepsilon$ and $g$, respectively. By substitution in the first term for $j = n$, the above inequality becomes

$$SV(n) > \frac{1}{2} \text{tr} \left[ \mathbf{E}^{-1} (\Sigma_n + g \Sigma^*) \mathbf{E}^{-T} \right].$$  \hspace{1cm} (3.133)

We now turn our attention to the macronode protocol. Assume that $n \geq 2$, the minimum number of macronode measurements required to implement any single-qumode Gaussian unitary (see Sec. 3.4). The scalar variance for the macronode protocol is

$$SV_m(n) = \frac{1}{2} \sum_{i=1}^{n} \text{tr} \left[ \tilde{\mathbf{V}}_i^{-1} \Sigma_i \tilde{\mathbf{V}}_i^{-T} \right].$$  \hspace{1cm} (3.134)

For each $i$, $\text{tr} \left[ \tilde{\mathbf{V}}_i^{-1} \Sigma_i \tilde{\mathbf{V}}_i^{-T} \right] > 0$, so

$$SV_m(n) > \frac{1}{2} \text{tr} \left[ \tilde{\mathbf{V}}_n^{-1} \Sigma_n \tilde{\mathbf{V}}_n^{-T} \right].$$  \hspace{1cm} (3.135)

Recall that $\Sigma_i^* = \Sigma_i + t^{-2} \Sigma^*$. We also require that $\tilde{\mathbf{V}}_n = \mathbf{E}$. Thus,

$$SV_m(n) > \frac{1}{2} \text{tr} \left[ \mathbf{E}^{-1} (\Sigma_n + t^{-2} \Sigma^*) \mathbf{E}^{-T} \right].$$  \hspace{1cm} (3.136)

We have found a lower bound for $SV_m(n)$ that is of the same form as the lower bound for $SV(n)$ in Eq. (3.133) (with $t$ in place of $g$). We omit the proof for the dictionary protocol since it is identical to the macronode case up to the requirement that $n \geq 4$ and setting $\tilde{\hat{V}}_i = \tilde{\hat{W}}_i$ by the appropriate restriction.
Now consider this lower bound for general $g$. Denote this quantity as $R(g)$. Using the cyclic property of the trace, we see that

$$
R(g) := \frac{1}{2} \text{tr} \left[ E^{-1}(\Sigma s + g^{-2}\Sigma^*)E^{-T} \right] = \frac{1}{2} \text{tr} \left[ (EE^T)^{-1}(\Sigma s + g^{-2}\Sigma^*) \right].
$$

(3.137)

Using the decomposition $E = R(\theta)S(\eta)R(\phi)$, note that $EE^T = R(\theta)S(\eta^2)R(-\theta)$. Therefore, $R(g)$ has no dependence on $\phi$. Evaluating Eq. (3.137) explicitly yields

$$
R(g) = \eta^2 \left( s^2 \varepsilon + \frac{c^2 \varepsilon}{g^2} \right) + \eta^{-2} \left( c^2 \varepsilon + \frac{s^2 \varepsilon}{g^2} \right) \geq \varepsilon (\eta^2 + \eta^{-2}) \min \{ 1, g^{-2} \}.
$$

(3.138)

Thus, for the CVW protocol,

$$
SV(n) > \begin{cases} 
    \varepsilon (\eta^2 + \eta^{-2}) & \text{if } |g| \leq 1, \\
    g^{-2} \varepsilon (\eta^2 + \eta^{-2}) & \text{if } |g| > 1.
\end{cases}
$$

(3.139)

Since $t = \tanh 2\alpha < 1$, for the macronode and dictionary protocols,

$$
SV_{(m,d)}(n) > \varepsilon (\eta^2 + \eta^{-2}).
$$

(3.140)

Therefore, for fixed $\varepsilon$ and $g$, in the large or small limit of $\eta$, the scalar variance for each protocol diverges. For the CVW case with $g \leq 1$ and for the macronode and dictionary protocols, high fidelity in a gate containing squeezing is only possible when $\varepsilon \ll \min \{ \eta^2, \eta^{-2} \}$. When $g > 1$ in the CVW case, however, a more lenient condition emerges: $g^{-2} \varepsilon \ll \min \{ \eta^2, \eta^{-2} \}$. Thus, increasing the edge weight $g$ in a CVW to be above 1 may allow for gates with higher squeezing to be implemented with the same $\varepsilon$. This makes sense in terms of the remodeling protocol of Sec. 3.3.2, which showed that increasing $g$ is, in a certain sense, like decreasing $\varepsilon$. This can be understood by recalling that a higher value of $g$ represents a stronger $\hat{C}_Z$ gate, which itself requires higher squeezing to implement [89]. Instead of doing this, one could just redirect that extra squeezing into an effort to decrease $\varepsilon$ even further.
Chapter 4

One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

4.1 Abstract

One-way quantum computing is experimentally appealing because it requires only local measurements on an entangled resource called a cluster state. Record-size, but non-universal, continuous-variable cluster states were recently demonstrated separately in the time and frequency domains. We propose to combine these approaches into a scalable architecture in which a single optical parametric oscillator and simple interferometer entangle up to \((3 \times 10^3 \text{ frequencies}) \times (\text{unlimited number of temporal modes})\) into a new and computationally universal continuous-variable cluster state. We introduce a generalized measurement protocol to enable improved computational performance on the new entanglement resource.
4.2 Introduction

One-way quantum computing [9] is a form of measurement-based quantum computing (MBQC) [152, 153] and an appealing alternative to the circuit model [8], which is being more widely pursued [154]. In one-way quantum computing, the primitives of the universal gate set are pre-encoded in a “quantum substrate” that is a generic, yet precise, entangled cluster state described by a graph specifying the entanglement structure of the qubits [40] or qumodes [140]. Quantum computing proceeds solely from single-node measurements on the cluster graph and feedforward of the measurement results [9, 20].

Quantum error correction and fault tolerance in one-way quantum computing have been theoretically proven feasible for qubit cluster states [53], with thresholds comparable to those for concatenated codes ($10^{-3}$ to $10^{-6}$), and then later improved using topological methods to thresholds slightly above the percent level [51]. Fault tolerance has recently been proven for continuous-variable (CV) cluster states in terms of required levels of squeezing, the squeezing threshold being no more than 20.5 dB for a $10^{-6}$ error rate [87]. Since the techniques used in Ref. 87 mirror those in Ref. 53, this threshold value is conservative and can most likely be improved.

A fully fledged, scalable experimental demonstration of one-way quantum computing has yet to be achieved, as none of the proof-of-principle implementations using four photonic qubits [155, 156] or four optical qumodes [148] employed a scalable architecture.

Recently, one-dimensional cluster-state entanglement was demonstrated, at record sizes, over the continuous variables represented by the quantum amplitudes of the electromagnetic field, a.k.a. qumodes. This was achieved in the frequency domain [96], with 60 simultaneously addressable entangled qumodes, and in the time domain [57, 141], with $10^4$ sequentially addressable entangled qumodes. Solely technical issues reduced these numbers from their potential higher values of $3 \times 10^3$ qumodes in the frequency architecture [157] and unlimited qumodes in the temporal architecture [141]. Besides this scalability breakthrough, optical implementations of quantum information offer other advantages such as room temperature operation, naturally low decoherence, and significant potential for device integration [158, 159].

We show that one can create computationally universal CV cluster states by entangling, both in time and in frequency, the quantum frequency comb of EPR pairs emitted from a single optical parametric oscillator (OPO). Based on previous results [96, 141], the lattice for this state could potentially be up to $3 \times 10^3$ nodes in one dimension (frequency) and unlimited in the other (time bins). We then show that this state enables universal quantum computing.
This work combines the best of all previous proposals for scalable CV cluster states: It employs Gaussian states with bipartite, self-inverse graphs—which are known to be highly scalable [55, 56]—and reduces the experimental requirements by simultaneously utilizing both frequency multiplexing [58, 96] and temporal multiplexing [57, 97, 141]. In addition, these architectures are known to admit more compact computation [141] with more favorable noise properties [2] when compared to approaches based on CV cluster states generated by the canonical method [20, 65]. Those so-called canonical CV cluster states [59]—which also admit a temporal [97] and a time-frequency implementation [99]—are not so easily scalable in optics due to frequent use of the CV controlled-Z gate.

Our proposal, in contrast, employs macronode-based cluster states [56] entangled into a bilayer square lattice (BSL), which has 2 qumodes per macronode (hence ‘bilayer’), instead of 4 as in previous proposals [55, 56, 58, 59]. The BSL CV cluster state admits
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

a more versatile elementary gate set than do canonical CV cluster states [20, 65], generalizing an analogous result for single-qumode operations on the CV dual-rail quantum wire discussed in Ch. 3.

The structure of this Article proceeds as follows: In Sec. 4.3 we describe the BSL resource state and give an explicit experimental procedure for how to generate it. In Sec. 4.5 we describe our measurement protocol, describing how single-site measurements can be used for universal quantum computation on the BSL. In Sec. 4.7 we discuss noise due to finite squeezing and we conclude in Sec. 4.8.

4.3 State generation

Construction of the BSL CV cluster state is illustrated in Fig. 4.1 and described in more detail here. A type-II OPO is pumped at two frequencies $2\nu_0 \pm \Delta \nu$, one of each polarization ($Y$ and $Z$). Each pump produces a number of two-mode squeezed (TMS) states [160] over the frequency comb of the OPO eigenmodes, as shown in Fig. 4.1(a). These states are each a Gaussian approximation to an Einstein-Podolski-Rosen (EPR) state [161] between two frequencies that add to the corresponding pump frequency. Now, even if the pump beams are continuous wave (CW), we still can, and will, logically assign pieces of the output beam to sequential time bins [141].

The OPO modes have linewidth $\delta \nu$ and are spaced by the free spectral range $\Delta \nu$. Each output frequency $\nu_n = \nu_0 + n\Delta \nu$ has a corresponding frequency index $n$ and associated macronode index $m$ [58]

$$m := (-1)^n n, \quad (4.1)$$

which we will call the node index for short and is used to label qumodes sequentially (rather than by frequency) in Fig. 4.1(a). Indeed, phasematching two frequencies $\nu_n$ and $\nu_{n'}$ requires $n + n' = \pm 1$, and all TMS states are generated between adjacent node indices (i.e., $m - m' = \pm 1$ [58]) in Fig. 4.1(a).

A $\pi/4$ polarization rotation (by a halfwave plate at $\pi/8$ rad from the horizontal principal axis of the OPO crystals), equivalent to a balanced beamsplitter for polarization qumodes, entangles these TMS states into a temporal sequence of frequency-encoded dual-rail quantum wires [58, 96], as shown in Fig. 4.1(b). A Mach-Zehnder interferometer (MZI) of path difference $c(2\Delta \nu)^{-1}$ [162, 163] separates frequencies of even and odd frequency index (and node index) into separate beams. For all odd qumodes, the $Z$ polarization is then time-delayed with respect to the $Y$ polarization by the interval $\delta t$ between two

88
consecutive time bins. The result is shown in Fig. 4.1(c). A final $\frac{\pi}{4}$ polarization rotation on the odd qumodes (another “balanced beamsplitter”) yields the BSL graph of Fig. 4.1(d).

A final phase delay by $\frac{\pi}{2}$ (not shown) on either all odd or all even frequencies converts this into a finitely squeezed CV cluster state with the same ideal graph as in panel (d). *It is this state that we call the BSL CV cluster state.* The fact that the BSL is a bipartite, self-inverse graph makes this possible and ensures the scalability of the scheme [55–59]. (See the general discussion of bipartite, self-inverse graphs in Ref. 59.)

### 4.4 Experimental Details

We now outline the basic experimental requirements for generating the BSL CV cluster state, verifying its entanglement structure, and using it for quantum information processing.

*Generating* the BSL CV cluster state requires a “musical score” condition—i.e., the measurement times must be compatible with resolving all qumode frequencies: $\delta t \gg \Delta \nu^{-1}$, an easily fulfilled condition. In addition, the measurement times must allow one to achieve maximum squeezing—that is, they must be at least as long as the OPO cavity storage time [164]. This translates into $\delta t \gg \delta \nu^{-1} \gg \Delta \nu^{-1}$, since $\delta \nu$ is also half the squeezing bandwidth [165]. This condition can also be easily fulfilled in practice [141] and ensures that the time bins contain maximally squeezed qumodes, to the extent permitted by the experiment’s squeezing limit (mainly determined by the intracavity losses).

Moreover, it is important to remember that as long as the undepleted pump approximation remains valid, the number of modes to be entangled has no bearing on the required pump power. To see that the undepleted-pump approximation holds for our scheme, note that a typical 100-mW pump power (i.e., $2.5 \times 10^{17}$ photons/s for green light) and a typical OPO cavity lifetime of 20 ns together yield $5 \times 10^{9}$ pump photons available for downconversion. Squeezing of 20 dB corresponds to 24.5 OPO photons per output mode (since $\langle \hat{a}^\dagger \hat{a} \rangle = \sinh^2 r$, with $\# \mathrm{dB} = 10 \log_{10} e^{2r} \approx 8.69 r$). With each pump photon downconverting into two daughter photons, even with one million output modes the total number of pump photons required is only $\frac{1}{2} \times 24.5 \times 10^{9} = 1.2 \times 10^{7}$, which is just 0.25% of the total number available. Therefore, pump depletion is indeed negligible.

To *verify* that the BSL CV cluster state has been generated successfully, we use a balanced homodyne measurement with a two-tone local oscillator (LO), as demonstrated
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

in our two previous works [95, 96]. For entanglement characterization alone, the qumodes do not need to be separated in frequency.

When using the BSL CV cluster state for quantum information processing, complete qumode separation is required. The qumode separation is straightforward in the time and polarization domains. Experimental techniques that have been honed on classical optical frequency combs [166] can be used for the frequency domain qumode separation. Such techniques include virtually-imaged phase arrays (VIPAs), arrayed waveguide gratings (AWGs), as well as diffraction gratings and combinations thereof. After separation, the individual beams will be directed to homodyne detectors or photon counters as required by the particular algorithm [65]. In the case of homodyne detection, the local oscillators will likely need to be derived from a stable classical comb, be it a femtosecond laser or a cavity-enhanced EOM, whose beam can be overlapped with the OPO’s and subjected to the same frequency separation method. The use of integrated optics may assist in implementing this scheme to full scale.

4.5 Basics of quantum computing on the bilayer square lattice

The BSL CV cluster state is easily shown to be universal for MBQC. Simply measure $\hat{q}$ on all qumodes of one (e.g., $Y$) polarization, resulting in a CV cluster state with an ordinary square-lattice graph, which can be used with standard CV MBQC protocols [20, 65]. This is shown in Fig. 4.2(a).

Using so-called deletion measurements (as above) to simplify the graph structure of a CV cluster state is a standard way to prove universality of a given graph [56, 57], but it is a wasteful procedure to follow in practice since half of the graph nodes and their connectivity are lost. More precisely, this method inefficiently uses available squeezing (and therefore entanglement [167]), which leads to extra noise when using these resources for quantum computing [2]. Furthermore, lattice edges are at $45^\circ$ to the direction of increasing temporal index, meaning that either the information must flow in a zig-zag pattern or the lattice width will have to scale linearly with the length of the computation, hindering the scheme’s scalability.

Fortunately, there exists a more favorable MBQC protocol that eschews all these complications and makes better use of the structure of the BSL CV cluster state, while still using just single-site measurements. The idea is to use both layers of the graph simultaneously and in a way analogous to the conventional (single-layer) square lattice protocol, as shown in Fig. 4.2(b). Each lattice site, which we call a macronode [56], is
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Figure 4.2: Two MBQC protocols applied on the bilayer square-lattice cluster state \( \mathcal{C} = 2^{-3/2} \). In both cases, information is encoded on the left (in purple nodes) and flows from left to right along (green) wire segments. Wires are separated by lines of sacrificial qumodes (shown in the red segments). These are referred to as control macronodes \( c(i, i+1) \) because measurements on them control whether one- or two-qumode gates are applied on the adjacent wire macronodes \( w(i) \) and \( w(i+1) \). Two-qumode gates are represented by a connecting green segment between two adjacent wires. (a) Standard one-way protocol [20, 65] applied to the BSL graph after measuring \( \hat{q} \) on all \( Y \)-polarized qumodes (shown faded). Time-ordering of the nodes has been preserved, resulting in an atypical nodal arrangement of an ordinary square-lattice graph. Information propagates at 45° to the direction of increasing temporal index. Control nodes are measured in the \( \hat{q} \)-basis to delete them or in a different basis to implement a two-qumode gate. (b) New MBQC protocol taking advantage of the BSL structure. Both layers of the lattice are used simultaneously, and quantum information propagates in the direction of increased temporal index, i.e., horizontally on the figure. Control and wire macronodes are now at a constant frequency, as shown. See text for further details.

composed of two qumodes (one of each polarization). Qumodes with even node index carry the quantum information to be processed and are therefore called wire macronodes (for ‘quantum wires’). Those with odd node index control the connectivity between the wires and are therefore called control macronodes. Input states are localized with respect to the macronode structure and are encoded within the symmetric subspace of each macronode (defined in the section below). One- and two-qumode Gaussian gates are selected by the choice of homodyne measurement angles.

To simplify the presentation, we will introduce our protocol within the context of an infinitely-squeezed BSL resource state. Any physical CV cluster state can only be finitely squeezed [20, 59], and this leads to introducing finite squeezing effects into the computation [2], which we discuss in Sec. 4.7.
4.5.1 Computing with macronodes

For a given macronode with node index $m$, comprised of individual qumodes labeled $Y$ and $Z$, we define the symmetric (+) and anti-symmetric (−) qumodes via

$$
\hat{a}_{m\pm} := \frac{1}{\sqrt{2}}(\hat{a}_m Z \pm \hat{a}_m Y),
$$

which is mathematically equivalent to a $\frac{\pi}{4}$ polarization rotation into the diagonal and anti-diagonal qumode decomposition (equivalently, a 50:50 beamsplitter between the two qumodes). Input states at a particular time step will either be the output state from the previous time step or new states directly injected into the BSL via an optical switch [141]. They are localized to macronodes but distributed (symmetrically) between the two physical qumodes. We further define quadrature operators $\hat{q}$ (position) and $\hat{p}$ (momentum) for each qumode through $\hat{a} = \frac{1}{\sqrt{2}}(\hat{q} + i\hat{p})$, which satisfy $[\hat{q}, \hat{p}] = i$ with $\hbar = 1$.

Before describing our measurement protocol, we also provide some definitions for useful CV logic gates. These include an optical phase delay by $\theta$, $\hat{R}(\theta) := \exp(i\theta \hat{a}^\dagger \hat{a}) = \exp\left[\frac{i\theta}{2}(\hat{q}^2 + \hat{p}^2 - 1)\right]$, and a 50:50 beamsplitter between qumodes $i$ and $j$, $\hat{B}_{ij} := \exp\left[-\frac{\pi}{4}(\hat{a}_i^\dagger \hat{a}_j - \hat{a}_j^\dagger \hat{a}_i)\right]$

$$= \exp\left[-i\frac{\pi}{4}(\hat{q}_i \hat{p}_j - \hat{q}_j \hat{p}_i)\right].$$

We also define a (nonstandard) single-qumode squeezing operation:

$$\hat{S}(s) := \hat{R}(\text{Im} \ln s) \exp\left[-\frac{1}{2}(\text{Re} \ln s)(\hat{a}^2 - \hat{a}^{\dagger 2})\right]$$

$$= \hat{R}(\text{Im} \ln s) \exp\left[-\frac{i}{2}(\text{Re} \ln s)(\hat{q}^2 + \hat{p}^2)\right],$$

where $s$ is known as the squeezing factor. This gate is just an ordinary squeezing gate with squeezing parameter $r = \ln |s|$, followed by a $\pi$ phase delay if and only if $s < 0$. We chose this form of the gate so that for all real $s \neq 0$, its Heisenberg action on the quadratures is $\hat{S}(s)^\dagger \hat{q} \hat{S}(s) = s\hat{q}$ and $\hat{S}(s)^\dagger \hat{p} \hat{S}(s) = s^{-1}\hat{p}$.

As is standard in MBQC, once the entangled resource state is prepared, quantum computation proceeds solely through adaptive local measurements. Here we restrict the measurements to linear combinations of the quadrature operators, which will be shown
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Figure 4.3: (a) Simplified graphical-calculus representation [57] of the bilayer square-lattice (BSL) CV cluster state. Here, qumodes are ordered according to temporal index. Input states are encoded within macronodes on the left, shown in purple. Here and also in (b), $C = (2\sqrt{2})^{-1}$. (b) As in (a) but time ordering has been partially sacrificed in order to make the square-lattice graph structure more apparent. (c) Each macronode is now represented in terms of the logical-mode tensor-product structure [see Eq. 4.2]. We use the same time-ordered node arrangement as in (b). Unlike in the previous subfigures, here the graph has a lower connectivity [it is a disjoint collection of square graphs] and all input states are localized. In this subfigure, $C = 2^{-1/2}$. We indicate internal qumode labeling on the top right macronode of each lattice.

to be sufficient to implement arbitrary multi-qumode Gaussian unitaries. Experimentally, this can be performed through homodyne detection. For any given qumode, we define the rotated quadrature operators

$$\mathbf{x}(\theta) = \left( \begin{array}{c} \hat{q}(\theta) \\ \hat{p}(\theta) \end{array} \right) := \hat{R}^\dagger(\theta) \left( \begin{array}{c} \hat{q} \\ \hat{p} \end{array} \right) \hat{R}(\theta)$$

$$= \left( \begin{array}{cc} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{array} \right) \left( \begin{array}{c} \hat{q} \\ \hat{p} \end{array} \right), \quad (4.6)$$

where the second line shows the symplectic-matrix representation of the Heisenberg action of a phase delay by $\theta$ [2].

In Fig. 4.3 we show alternative (and equivalent) graphical representations of the BSL CV cluster state. Recall that within each macronode the map from the physical and logical mode labels is given by Eq. 4.2. We can apply this map to every macronode, giving us a graph where each node now represents the symmetric or anti-symmetric mode of the enclosing macronode, as shown in Fig. 4.3(c). This graph reveals a simpler underlying logical structure that will provide us with a convenient framework for describing how homodyne measurements on the physical modes can implement useful gates.

Generically, due to the non-local nature of the map from physical $(Z, Y)$ to logical $(+, -)$
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

mode labels, local measurements on the physical modes will effectively “stitch together” the disjoint square graphs present in Fig. 4.3(c). For a macronode $m$, measurement of $\hat{p}_{mZ}(\theta_{mZ})$ and $\hat{p}_{mY}(\theta_{mY})$ can be represented by the following quantum circuit:

$$\rho_{m,+/-} \left\{ \begin{array}{c} \begin{array}{c} B \\ R(\theta_{mZ}) \\ R(\theta_{mY}) \\ p \\ p \end{array} \end{array} \right. \right.$$  \tag{4.7}

where $\rho_{m,+/-}$ denotes the input state with respect to the logical $(+,-)$ mode tensor product structure. Above, the 50:50 beamsplitter takes us from logical to physical mode labels. Rotated quadratures are measured as in Eq. 4.6. Such measurements on the wire macronodes connect square subgraphs with their neighbors in the horizontal direction, enabling “wire-like” transmission along the BSL. The measurements on the control macronodes connect these neighboring wires vertically.

4.5.2 Keeping square graphs disconnected

For a fixed macronode, there exists a one-parameter class of homodyne angles that do not connect the adjacent square graphs. Specifically, when $\theta_{mZ} = \theta_{mY} = \theta$, the above circuit (4.7) is equivalent to

$$\rho_{m,+/-} \left\{ \begin{array}{c} \begin{array}{c} \text{R(\theta)} \\ B \\ p \\ p \end{array} \end{array} \right. \right.$$  \tag{4.8}

where the single-qumode rotation gates commute with the 50:50 beamsplitter because the rotation angles are the same [168]. This in turn is equivalent to

$$\rho_{m,+/-} \left\{ \begin{array}{c} \begin{array}{c} \text{R(\theta)} \\ p \\ +,- \\ p \end{array} \end{array} \right. \right.$$  \tag{4.9}

where all circuit elements are now local, and we take the sum and difference of the measurement outcomes. Therefore, choosing $\theta_{mZ} = \theta_{mY}$ for a particular wire or control macronode $m$ will disconnect the neighboring regions of the BSL graph in the horizontal or vertical direction, respectively. By restricting all control macronode measurements in this way and including the required post processing (i.e., sum and difference of outcomes), the disjoint square graphs of Fig. 4.3(c) remain uncoupled by homodyne measurements with respect to the physical modes.
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Figure 4.4: (Color online) Implementing single- and two-qumode gates on the bi-layer square lattice. Node indices [Eq. (4.1)] for all macronodes are provided on the left. A red ellipse indicates a restriction on the measurements of that macronode—specifically, $\theta_{cZ} = \theta_{cY}$ for control macronode $c$. (a) We begin with the configuration as in Fig. 4.3(c), with each macronode decomposed into logical $(+, -)$ modes. Note that measurements on macronodes 1 and 5 are restricted (red coloring). This decouples the two fully displayed square graphs from their partially displayed neighbors above and below. (b) Same as (a), except we visually separate the internal nodes of the control macronodes. (c) Starting with (b), we decompose the wire macronodes (within the green regions) with into physical $(Z, Y)$ modes in order to reveal a pair of CV dual-rail wires [2, 96, 141]. Restricting the measurements (red ellipse) of control macronode 3 allows one to implement single-qumode gates [2] on each wire independently [Sec. 4.6.1]. (d) Alternatively, if we set $\theta_{3Z} \neq \theta_{3Y}$, then control macronode 3 will mediate an entangling gate between the two neighboring wires [Sec. 4.6.3].

4.6 Universal gate set

The methods above allow us to apply single-qumode gates on adjacent wires without them interacting. Alternatively, relaxing the restriction on a particular control macronode implements a two-qumode gate between the adjacent wires at that location. In this section, we elaborate on this and construct a universal gate set for quantum computation on the BSL.

4.6.1 Single-qumode gates

Fig. 4.4(a)–(c) shows a new way to represent the BSL such that all measurements are local, but with respect to a mixture of physical $(Z, Y)$ and logical $(+, -)$ mode labels. As information propagates along the lattice in the direction of increasing time index, information will flow strictly in the horizontal direction, and there will be no interactions between neighboring wire macronodes.
The structure shown in Fig. 4.4(c) is identical to a collection of CV dual-rail quantum wires \[2, 57\], which are resources for universal single-qumode quantum computation. Therefore, we can implement single-qumode gates on the BSL by directly implementing the macronode protocol for the CV dual-rail wire from Ch. 3. We briefly review it here. If the qumodes at the left-most wire site are measured in the bases $\hat{p}_{mZ}(\theta_{mZ})$ and $\hat{p}_{mY}(\theta_{mY})$, as depicted here,

\[
\hat{p}(\theta_{mZ}) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Figure 4.5: (Color online) Sign convention for measurements on the control macronodes. Both physical modes of each ± control macronode (alternating top to bottom, as shown above) are measured in the basis specified by homodyne angle \( \theta = \pm \frac{\pi}{4} \), respectively.

Fig. 4.4(c)]. Note that these constraints jointly affect nodes of separate neighboring wires, which share a control macronode.

Although these constraints do not completely specify the set of possible measurements on the control macronodes, some care has to be taken in assigning the measurement angles. For one thing, constraining all control macronode measurements to the same angle would projectively measure the encoded information (as discussed below in Sec. 4.6.2), thereby ending the computation at that point. On the other hand, attempting to use control macronode degrees of freedom to locally implement some desired gate on a particular wire would necessarily implement a nontrivial gate on both neighboring wires. For this reason, we fix all measurements on the control macronodes and only use the measurements on the wire macronodes to implement gates.

A particularly convenient choice is to set the homodyne angles to be \( \theta_{cZ} = \theta_{cY} = \pm \frac{\pi}{4} \) (for control macronode \( c \)), where the sign alternates vertically with each row of control macronodes, as shown in Fig. 4.5. For one physical time step (i.e., measuring one wire macronode \( w \) and its neighboring control macronodes) on a wire above a row of ± control macronodes, this implements

\[
\hat{V} \left( \mp \frac{\pi}{4}, \pm \frac{\pi}{4} \right) \hat{V}(\theta_{wZ}, \theta_{wY}), \quad (4.13)
\]

where the first gate \( \hat{V}(\theta_{wZ}, \theta_{wY}) \) results from measurement of the wire macronode \( w \), and the second gate \( \hat{V}(\mp \frac{\pi}{4}, \pm \frac{\pi}{4}) \) results from the measurements of the two control macronodes above and below, as in Fig. 4.3(a). Plugging into Eq. (4.11), we get

\[
\hat{V} \left( \mp \frac{\pi}{4}, \pm \frac{\pi}{4} \right) = \hat{S}(\mp 1) \quad (4.14)
\]
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Noting that $\hat{S}(-1)\hat{V}(\theta_wZ, \theta_wY) = \hat{V}(\theta_wY, \theta_wZ)$, a single measurement step on the BSL implements $\hat{V}(\theta_wY, \theta_wZ)$ or $\hat{V}(\theta_wZ, \theta_wY)$, depending on whether the control macronodes below the wire are $+$ or $-$, respectively. Two applications of these gates generate all single-qumode Gaussian unitaries (up to displacements) \[2, 141\].

As mentioned above, we have neglected a ubiquitous phase-space displacement (dependent on the measurement outcomes) and the effects of finite squeezing in our discussion above. We did this in order to present clearly the basic logic of the protocol. The details of the additional displacements and squeezing effects can be found in Sec. 4.7.

4.6.2 Projective measurement

Notice that when $\theta_wZ = \theta_wY = \theta$, the squeezing term in Eq. (4.11) diverges. A gate is not applied in this case. Instead, this projectively measures both logical modes in $\hat{p}(\theta)$, as can be seen from the symmetry discussed in Sec. 4.5.2.

4.6.3 Two-qumode gates

In the above, we found that (by appropriately restricting the control macronode measurements) we could treat the BSL as a collection of independent non-interacting quantum wires. This protocol can be extended to also include a two-qumode entangling gate by lifting the measurement restrictions on control macronodes that lie between neighboring wires. This corresponds to the case shown in Fig. 4.4(d). We parameterize the choice of measurements by the vector of homodyne angles $\theta = (\theta_1Z, \theta_1Y, \theta_2Z, \ldots, \theta_5Y)$.

We would like our two-qumode gate protocol to be compatible with single-qumode gates applied on adjacent regions of the BSL. With respect to Fig. 4.4(d), we allow $\theta_1Z$ and $\theta_3Y$ to be free parameters, while $\theta_1Y = \theta_5Z = \theta_5Y = \pm \frac{\pi}{4}$, corresponding to macronode 3 being a $\mp$ control macronode, respectively. Correspondingly, selecting homodyne angles $\theta = \left(\pm \frac{\pi}{4}, \pm \frac{\pi}{4}, \theta_2Z, \theta_2Y, \theta_3Z, \theta_3Y, \theta_4Z, \theta_4Y, \pm \frac{\pi}{4}, \pm \frac{\pi}{4}\right)$ (4.15) implements a two-qumode gate whose form we will now derive.

Our strategy for the derivation will be to use symmetries of CV cluster states and “beamsplitter gymnastics” to reduce the evolution to a form that can be interpreted as a combination of two steps of evolution on the CV dual-rail wire \[2\] interleaved with two additional beamsplitters. To this end, we call attention to Fig. 4.6, which shows that measurements on the original resource shown in (a) are equivalent to the same
Figure 4.6: (Color online) Beamsplitter gymnastics. All graphs are drawn in terms of logical (+, −) modes. A 50:50 beamsplitter $\hat{B}_{ij}$ between two qumodes $i$ and $j$ is indicated by a red arrow from $i$ to $j$. Where applicable, dashed-arrow beamsplitters always act before solid-arrow beamsplitters. (a) We start from Fig. 4.4(a). Measuring control macronode 3 in the physical $(Z,Y)$ modes is equivalent to performing a beamsplitter as shown and then measuring in the logical modes. (b) Since all qumodes of control macronodes 1 and 5 are measured in the same basis, we are free to insert an additional beamsplitter between them as shown [see Sec. 4.5.2]. This is the key observation. (c) The squares in (b) (with $\mathcal{C} = 2^{-1/2}$) can be replaced by pairs of two-qumode CV cluster states (with $\mathcal{C} = 1$) followed by two additional beamsplitters as shown (dashed) [57]. These occur before the other two (solid). (d) By direct calculation using their symplectic representation [59], $\hat{B}_{il}\hat{B}_{kj}(\hat{B}_{kl}\hat{B}_{ij}) = (\hat{B}_{kl}\hat{B}_{ij})\hat{B}_{lj}\hat{B}_{ik}$. (e) The symmetries of a pair of two-qumode CV cluster states (see Appendix 4.11) allow for the beamsplitter to be moved to the other two qumodes as shown.

measurements on the resource shown in (e). As such, we can read off the evolution from the last subfigure, using knowledge of evolution on the CV dual-rail wire from Ch. 3.

We summarise this procedure here, referring to Fig. 4.6(e). First, the leftmost wire macronodes (2 and 4) are measured, applying $\hat{V}(\theta_2Z, \theta_2Y) \otimes \hat{V}(\theta_4Z, \theta_4Y)$ to the input and teleorting the output into qumodes $1^-$ and $3^-$, respectively. Then, the 50:50 beamsplitter between those two qumodes (dotted arrow) is applied. Next, the solid-arrow beamsplitters and measurements of the control macronodes implement the gate $\hat{V}(\pm \frac{\pi}{4}, \theta_3Z) \otimes \hat{V}(\theta_3Y), \pm \frac{\pi}{4})$, teleporting the output to qumodes $2^+$ and $4^+$ at the following timestep. Finally, the last dotted-arrow beamsplitter acts on this output, concluding the evolution.

Thus, up to displacements and neglecting finite-squeezing-induced noise (see Sec. 4.7), the total gate applied is the combination of all of these individual gates:

$$\hat{B}_{2+,4+} \left[ \hat{V} \left( \pm \frac{\pi}{4}, \theta_3Z \right) \otimes \hat{V} \left( \theta_3Y, \pm \frac{\pi}{4} \right) \right] \hat{B}_{2+,4+} \left[ \hat{V} (\theta_2Z, \theta_2Y) \otimes \hat{V} (\theta_4Z, \theta_4Y) \right], \quad (4.16)$$

where the tensor product is $\mathcal{H}_{(2+)} \otimes \mathcal{H}_{(4+)}$. This captures the most general type of two-qumode Gaussian unitary gate compatible with our framework.

Though we have the general form, it is useful to give particular measurement parameters that reduce the two-qumode gates into a simple form. It is also desirable to choose a
form that is commonly included in universal gate sets, such as the CV controlled-Z ($\hat{C}_Z$) gate [65], defined as $\hat{C}_Z(g) = \exp[i g \hat{q} \otimes \hat{q}]$.

While there is no valid choice of measurement parameters in $\theta$ that yields an exact $\hat{C}_Z$ gate, it is possible to implement one followed by phase delays that can be corrected in the next step by applying the single-qumode measurement protocol immediately after this gate. Again assuming macronode 3 is a $\mp$ control macronode, choosing

$$\theta = \pm \left( \frac{\pi}{4}, \frac{\pi}{8}, \frac{3\pi}{8}, \frac{\pi}{4}, \pm \phi, \frac{\pi}{4}, \mp \phi, -\frac{\pi}{8}, \frac{3\pi}{8}, \frac{\pi}{4} \right)$$

reduces Eq. (4.16) to

$$\left[ \hat{R} \left( \mp \frac{3\pi}{4} \right) \otimes \hat{R} \left( \pm \frac{\pi}{4} \right) \right] \hat{C}_Z(2 \cot \phi),$$

which is a tunable-strength $\hat{C}_Z$ gate followed by (known, fixed) phase delays that can be undone at the next time step. Appendix 4.12 contains the detailed derivation. Once again, we postpone discussing finite-squeezing effects and outcome-dependent displacements until Sec. 4.7.

### 4.6.4 Alternative representation of two-qumode gate implementation

In the previous two subsections, we showed how measurements on the control macronodes selected between applying either a pair of single-qumode gates or a two-qumode gate on neighboring wires. Here we provide an alternative description of this mechanism that employs more fully the graphical calculus for Gaussian pure states [59].

Rather than finding a graphical description of the BSL that uses a mixture of physical ($Z, Y$) and logical ($+, -$) mode labels as in Fig. 4.4, we can instead consider the graphical representation of the premeasurement of the control macronodes (in analogy to “wire shortening” in cluster state terminology [65]), as shown in Fig. 4.7. Note that for the measurement-based implementation of Gaussian gates, cluster nodes can be measured in any order since the result is equivalent up to a final phase-space displacement [65].

The edge weights that are changed by the measurements are functions of the homodyne angles on the control macronodes and are given below in the large squeezing limit. We get these from the graph transformation rules [59] corresponding to homodyne measurements on the physical modes of the BSL and then taking the limit $r \to \infty$ (we choose this limit
FIGURE 4.7: Graphical-calculus representation [59] of measurements on a subregion of the bilayer square lattice with inputs in purple. Here we show how measuring the control macronodes (1, 3, 5) in two different ways leads to different connectivities of the wires. (a) The lattice prior to measurement. We assume that $\theta_1^Z = \theta_1^Y$ and $\theta_5^Z = \theta_5^Y$. Having $\theta_3^Z \neq \theta_3^Y$ or $\theta_3^Z = \theta_3^Y$ will result in a graph as shown in subfigures (b) and (c), respectively. (b) After measurement, the resulting graph has connecting edges between the wire macronodes. This is consistent with the application of a two-qumode gate between the encoded inputs as was shown in Sec. 4.6.3. Relevant graphical weights are defined in Eqs. (4.19), (4.20), and (4.21). (c) After measurement, there are no graph edges connecting the input macronodes. Therefore, performing measurements on the input macronodes results in the application of single-qumode gates only. Thus, these entangled pairs can be thought of as separate quantum wires. Note that the four remaining edge weights share a dependence on $\theta_3^Z$. In other words, the weights of adjacent quantum wires—and hence the single-qumode gate applied on them—are logically dependent in general. This is consistent with what was shown in Sec. 4.6.1. Unlabeled edges all have $C = 2^{-1/2}$.

For clarity of presentation only. The edge weights in Fig. 4.7 are

$$f_i = \frac{1}{4}(\cot \theta_i^Z - \cot \theta_i^Y),$$

$$h_{ij} = \frac{1}{4}(\cot \theta_i^Z - \cot \theta_i^Y - \cot \theta_j^Z - \cot \theta_j^Y),$$

$$g_{ij} = \frac{1}{4}(\cot \theta_i^Z + \cot \theta_i^Y - \cot \theta_j^Z - \cot \theta_j^Y).$$

After the control macronodes are measured and when $\theta_3^Z \neq \theta_3^Y$, wire macronodes 2 and 4 are clearly connected by horizontal and diagonal links [see Fig. 4.7(b)]. Attempting to “teleport” the input states through this highly connected resource state will entangle the input states. Contrast this with the case when $\theta_3^Z = \theta_3^Y$ [see Fig. 4.7(c)], where the resource state is simply a pair of unconnected entangled pairs. The latter is useful for propagating input states horizontally across the lattice without entangling the inputs [2].
4.6.5 Achieving Universal Quantum Computation

A CV controlled-$Z$ gate $\hat{C}_Z$ can be applied between any two adjacent wires at any point on the BSL by locally substituting the macronode protocol with the entangling-gate protocol described in Sec. 4.6.3. This can be done repeatedly so long as each wire is involved in at most one $\hat{C}_Z$ gate at a time. Together with vacuum input states and Weyl-Heisenberg displacement operations, these gates are universal for multimode Gaussian computation [65].

In order to achieve universal quantum computation, we also need to include non-Gaussian resources [65]. (Sub-universal algorithms, such as state verification [96, 141], only require homodyne detection.) In principle, it does not matter which type of resource is used [19]. Typical examples include photon-counting measurements supplemented with Gaussian resources to implement a cubic phase gate [67] or preparation and injection of non-Gaussian magic states such as photon subtracted states [169]. We leave the detailed implementation to future work.

4.7 Displacements and finite-squeezing effects

Thus far, we have neglected both the measurement-outcome-dependent displacements and finite squeezing effects that always arise in CV MBQC [2, 65]. We account for them here.

Since all evolution on the BSL can be reduced to evolution on the CV dual-rail wire (plus additional beamsplitters in the case of the two-qumode gate), all we need to do to take into account the effects of the measurement outcomes and finite squeezing is to use Eq. 3.62 from Ch. 3, which amounts to replacing $\hat{V}(\theta_j, \theta_k)$ [Eq. (4.11)] with

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

(4.22)

where

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

(4.23)

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

$$\hat{N}(r) = e^{-\epsilon \hat{q}^2/2}e^{-\epsilon \hat{p}^2/2a^2} \hat{S}(t^{-1})$$

(4.24)
is a nonunitary operator that captures the effects of finite squeezing. We recover Eq. (4.11) in the limit of large squeezing and when all measurement outcomes are zero:

$$\hat{V}(\theta_j, \theta_k) = \lim_{r \to \infty} \hat{V}(r, 0, 0, \theta_j, \theta_k). \quad (4.25)$$

More generally, the displacements can either be actively corrected at each step or merely accounted for using feedforward [65].

Noise from finite squeezing is ubiquitous in all MQBC protocols using CV cluster states, but fault tolerance is still possible using quantum error correction [67] provided that the overall squeezing levels—which set the amount of noise introduced per gate [2]—are high enough [87]. The only known threshold result [87] states that no more than 20.5 dB of squeezing will be required. Squeezing levels in temporal-mode [141] and frequency-mode [96] cluster-state experiments (5 dB and 3.2 dB, respectively) fall short of this, but state-of-the-art experiments in optics [170] are within an order of magnitude (12.7 dB). The existence of a compact and scalable protocol such as the one presented here is likely to further spur on experimental and theoretical work to close this gap.

**Technical note.**—The astute reader will note that this presentation differs from that of Ch. 3 in two ways. First, the $r$-dependent squeezing term $\hat{S}(t^{-1})$ appears after the displacements in Eq. (4.22), while it appears before them in Eq. (3.62) of Ch. 3. We have modified our displacement operator (4.23) accordingly (Cf. Eqs. (3.66) and (3.67) in Ch. 3), which allows us to group all finite-squeezing effects to the end and allows our displacement to depend only on the measurement angles and outcomes (and not on $r$). Second, we have written the displacement in terms of the standard quantum-optics displacement operator, which relates to the Weyl-Heisenberg displacements as $\hat{X}(s)\hat{Z}(t) = \text{(phase)}\hat{D}[(s + it)/\sqrt{2}]$, and we ignore the overall phase.

### 4.8 Conclusion

We have proposed an extremely compact and scalable method for producing—from a single OPO and simple interferometer—a continuous-variable (CV) cluster state of unprecedented size $[(3 \times 10^3) \times \infty]$ that is universal for quantum computation. The proposal has all the advantages of record-breaking temporal- and frequency-multiplexed schemes [96, 141] while vastly increasing the size of the lattice by utilising both types of multiplexing at once. This is the most compact and scalable proposal for CV cluster states to date, and it is implementable today using demonstrated quantum-optical technology. In addition, we have generalized the one-way model for quantum computing to utilize the generated resource for quantum computation. The result translates familiar
notions of CV measurement-based quantum computing (MBQC) to the particular state proposed here, generalizing prior work based on one-dimensional, macronode-based CV cluster states [2, 141].

The vast majority of the existing literature on CV cluster states to date has treated canonical CV cluster states (i.e., those described in Refs. [20, 65, 140]) as the appropriate target for an MBQC resource state. The work presented here—as well as the entire research direction upon which it is based—shows that we should shift the focus onto CV cluster states with a macronode structure [2, 55–58, 96, 141]. These schemes, which are all based on bipartite, self-inverse graphs [56], have been demonstrated to have unprecedented scalability [96, 141] and to admit novel, flexible [4], and more efficient [2] quantum-computing schemes within the MBQC paradigm.

The work presented here further underscores this point, emphasizing the importance of bipartite, self-inverse graphs and of focussing on scalable designs from the ground up when working with CV cluster states. One might hope that the optimized protocols available for these states [2, 4] could be used to improve the fault-tolerance threshold for MBQC using CV cluster states [87]. We leave this question to future work.

### 4.9 Appendix: Graphical calculus for Gaussian pure states

Any \( N \)-qumode Gaussian pure state \( |\psi_Z\rangle \) can be represented uniquely (up to phase-space displacement and overall phase) by an \( N \)-node, complex-weighted, undirected graph [59]. This graph \( Z \) can be represented pictorially or, equivalently, by a corresponding \( N \times N \) complex-valued adjacency matrix

\[
Z := V + iU,
\]

where \( V \) and \( U \) are \( N \times N \) symmetric real-valued matrices, and \( U > 0 \). This object is related to the wavefunction in the following way:

\[
\psi_Z(q) = \left(\frac{\text{det} U}{\pi^{N/4}}\right)^{1/4} \exp \left[ i \frac{q^T Z q}{2} \right],
\]

A covariance matrix for this state can be expressed in terms of the matrices in Eq. (4.26). First, denote the vector of \( 2n \) position and momentum quadrature operators as

\[
\hat{x} := (\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_N, \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_N)^T.
\]
Then \[59\],
\[
\Sigma := \frac{1}{2} \langle \hat{x}, \hat{x}^T \rangle = \frac{1}{2} \begin{pmatrix} U^{-1} & U^{-1}V \\ VU^{-1} & U + VU^{-1}V \end{pmatrix},
\]
(4.29)
which in turn allows us to give an expression for the Wigner function:
\[
W(x) = (2\pi)^{-N} (\det \Sigma)^{-1/2} \exp \left[-\frac{1}{2} x^T \Sigma^{-1} x \right].
\]
(4.30)
For some Gaussian unitary \(\hat{U}\), we can define \(|\psi_Z'\rangle\) to be
\[
|\psi_Z'\rangle := \hat{U} |\psi_Z\rangle.
\]
(4.31)
\(|\psi_Z'\rangle\) is also a Gaussian pure state (by the definition of a Gaussian unitary). How is the graph \(Z'\) (corresponding to state \(|\psi_Z'\rangle\)) related to the original graph \(Z\) by the Gaussian unitary? The Heisenberg action of \(\hat{U}\) on \(\hat{x}\) is linear, which means it can be represented as \[59\]
\[
\hat{U}^\dagger \hat{x} \hat{U} =: S_{\hat{U}} \hat{x}
\]
(4.32)
where \(S_{\hat{U}}\) is a \(2N \times 2N\) symplectic matrix. If we represent \(S_{\hat{U}}\) as
\[
S_{\hat{U}} = \begin{pmatrix} A & B \\ C & D \end{pmatrix},
\]
(4.33)
then the corresponding graph update rule is \[59\]
\[
Z \rightarrow Z' = (C + DZ)(A + BZ)^{-1}.
\]
(4.34)

4.10 Appendix: Simplified graphical calculus

In general, representing all the features of \(Z\) requires an appropriately connected graph with all edges (including self-loops) labeled by complex-valued weights \[59\]. When representing Gaussian pure states with uniformly weighted graphs, it is convenient to employ a simplified set of rules. In the main text and wherever possible in the supplementary material, we represent Gaussian pure states using simplified graphs, as introduced in Ref. \[57\]. This allows us to represent graph edge weights by color and omit self-loops from the illustrations.

With the exception of the (omitted) self-loop weights, the edge weights are implicitly defined as \(\pm C t\), where \(C\) is called the edge-weight coefficient and can be thought of as
Figure 4.8: Going from using the simplified graphical calculus description of the bilayer square lattice with $C = 2^{-1/2}$ (left)—with edge weights defined implicitly by coloring—to the full graphical calculus description (right) [59]. Edge weights $t$ and $\varepsilon$ are defined in text [Eqs. (4.35) and (5.20), respectively].

The edge weight magnitude in the infinite squeezing limit, while

$$t := \tanh 2r$$

(4.35)

can be thought of as a rescaling factor that depends on an overall squeezing parameter $r$ for the state. In the graphs, signs of + and − are represented by blue and orange coloring, respectively, and $C$ is indicated within relevant figure captions. Note that in the infinite squeezing limit ($r \to \infty$), edge weight $\pm Ct \to \pm C$. For all graphs, all black nodes have self-loop edges with weight $i\varepsilon$, where

$$\varepsilon := \text{sech} 2r.$$ 

(4.36)

Technically, the simplified graphical calculus representations used in the majority of the figures of this Article are valid for both infinite- and finite-squeezing cases [57]. To include finite squeezing explicitly, the full graphical calculus [59] must be used. To do this, simply replace the simplified disjoint square graphs in Fig. 4.3(c) by more detailed versions with self-loops and edge weights as in Fig. 4.8.

We note that there is a subtlety in Fig. 4.1 of the main text. The state that exists at various stages (a)–(d) of the optical circuit diagram is technically not a CV cluster state, but is in fact an $\mathcal{H}$-graph state [59] that would have an edge weight of $-i \sinh 2r$. However, at every stage of the diagram this state can be converted into a CV cluster state with edge weights as quoted and with the same simplified graphical representation [57] by simply applying an optical phase delay of $\pi/2$ (a.k.a. a Fourier transform [65]) on half the qumodes (specifically, all qumodes with either even or odd frequencies). In practice, this difference is unimportant because this phase delay can be incorporated directly into the homodyne measurements acting on the final state, and in fact, the simplified
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

graphical calculus [57] is defined in Ref. [57] to represent both types of states (i.e., with or without these final Fourier transforms).

4.11 Appendix: Beamsplitter symmetries of a pair of two-qumode CV cluster states

We use Ref. [168] to derive the symmetries of a pair of two-qumode CV cluster states. This result is used to equate panels (d) and (e) of Fig. 4.6.

Each individual CV cluster state shown in Fig. 4.6(d) has a graph given by

\[
Z_1 = \begin{pmatrix} i \sech 2r & \tanh 2r \\ \tanh 2r & i \sech 2r \end{pmatrix} = i(\sech 2r) I + (\tanh 2r) \sigma_x \tag{4.37}
\]

and an alternative graph representation of [59, 168]

\[
K_1 := (I + iZ)(I - iZ)^{-1} = \begin{pmatrix} 0 & i \tanh r \\ i \tanh r & 0 \end{pmatrix} = i(\tanh r) \sigma_x, \tag{4.38}
\]

A pair of such states (one between qumodes \(i\) and \(j\) and a separate one between qumodes \(k\) and \(l\)) has the alternative graph

\[
K_2 = \begin{pmatrix} K_1 & 0 \\ 0 & K_1 \end{pmatrix} = i(\tanh r) I \otimes \sigma_x, \tag{4.39}
\]

with rows and columns ordered \((i, j, k, l)\). Note that \(\otimes\) here merely indicates a matrix Kronecker product and has nothing to do with a tensor product of Hilbert spaces.

An interferometric Hamiltonian \(\frac{1}{2} \hat{a}^\dagger \hat{M} \hat{a} \hat{M}^\dagger\), with \(\hat{M} = \hat{M}^\dagger\), generates a symmetry of the Gaussian pure state defined by \(K\) if and only if \(\hat{M} K = -(K \hat{M})^T [168]\). One choice (among many) for \(\hat{M}\) that works for \(K_2\) is \(\hat{M} = \sigma_y \otimes I\). This generates a one-parameter class of symmetry operations [168], one example of which is

\[
\exp \left( -i \frac{\pi}{4} \hat{a}^\dagger \hat{M} \hat{a} \right) = \exp \left[ -\frac{\pi}{4} (a^\dagger_k a_k + a^\dagger_l a_l - \text{H.c}) \right] = \hat{B}_{ik} \hat{B}_{jl}. \tag{4.40}
\]
Since this pair of beamsplitters is a symmetry of the pair of CV cluster states, acting with \( \hat{B}_{ik} \) alone is equivalent to acting with \( \hat{B}_{jl}^\dagger = \hat{B}_{lj} \) alone, which is exactly the symmetry employed in Fig. 4.6(e).

### 4.12 Appendix: Derivation of the two-qumode gate

Here we derive the two-qumode gate [Eq. (4.18)] implemented by using the measurement settings from Eq. (4.17) in Eq. (4.16). As usual, we neglect outcome-dependent displacements and finite-squeezing effects, with discussion of these effects relegated to Sec. 4.7. Before we start, we define the following abbreviations of phase shifts and squeezing on two qumodes at a time:

\[
\hat{R}(\theta_j, \theta_k) := \hat{R}(\theta_j) \otimes \hat{R}(\theta_k), \tag{4.41}
\]
\[
\hat{S}(s_j, s_k) := \hat{S}(s_j) \otimes \hat{S}(s_k). \tag{4.42}
\]

For the chosen measurement settings [Eq. (4.17)], the bottom line of Eq. (4.16) (measurements of wire macronodes) gives

\[
\hat{B} \left[ \hat{V} \left( \pm \frac{\pi}{8}, \pm \frac{3\pi}{8} \right) \otimes \hat{V} \left( \pm \frac{\pi}{8}, \pm \frac{3\pi}{8} \right) \right] = \hat{B} \left\{ \hat{R} \left( -\frac{3\pi}{4}, -\frac{3\pi}{4} \right) \right\}, \tag{4.43}
\]

where the two cases on the right correspond to the top and bottom signs, respectively, and we omit subscripts on \( \hat{B} \) for clarity. Next, we evaluate the top line of Eq. (4.16) (measurements of control macronodes), which gives

\[
\hat{B} \left[ \hat{V} \left( \pm \frac{\pi}{4}, \pm \frac{\pi}{4} + \phi \right) \otimes \hat{V} \left( \pm \frac{\pi}{4} - \phi, \pm \frac{\pi}{4} \right) \right] = \hat{B} \left\{ \hat{R} \left( -\frac{3\pi}{4}, -\frac{3\pi}{4} \right) \right\}, \tag{4.44}
\]

The total gate is therefore the following product of the two lines:

\[
\hat{B} \left[ \hat{R} \left( \frac{\phi}{2}, -\frac{\phi}{2} \right) \hat{S} \left( \tan \frac{\phi}{2}, \tan \frac{\phi}{2} \right) \hat{R} \left( -\frac{3\pi}{4}, -\frac{3\pi}{4} \right) \right] \times \hat{B} \left( \hat{R} \left( -\frac{3\pi}{4}, -\frac{3\pi}{4} \right) \right). \tag{4.45}
\]
Chapter 4: One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator

Noting that $\hat{B}\hat{R}(\theta, \theta) = \hat{R}(\theta, \theta)\hat{B}$, the full gate becomes

$$\hat{R} \left( \pm \frac{\pi}{4}, \pm \frac{\pi}{4} \right) \hat{B} \hat{R} \left( \frac{\phi}{2}, -\frac{\phi}{2} \right) S \left( \tan \frac{\phi}{2}, \tan \frac{\phi}{2} \right) \hat{R} \left( \frac{\phi}{2}, -\frac{\phi}{2} \right) \hat{B} \hat{R} \left( \frac{\pi}{2}, \frac{\pi}{2} \right), \quad (4.46)$$

where we have used the Bloch-Messiah decomposition [167] of the $\hat{C}_Z$ gate. This reduces the gate to its final form:

$$\hat{R} \left( \mp \frac{3\pi}{4}, \pm \frac{\pi}{4} \right) \hat{C}_Z(2\cot\phi). \quad (4.47)$$
Chapter 5

Flexible quantum circuits using scalable continuous-variable cluster states

5.1 Abstract

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.

5.2 Introduction

Quantum information processing using measurement-based quantum computing (MBQC) [9] 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 [40]), 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 [20]. Unlike their photonic-qubit counterparts [18, 133], 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 [91], all of which can be implemented using a single optical parametric oscillator (OPO) [3, 54–56]. Extremely large cluster states of this type can be made with existing technology based on either frequency modes [58, 96] or temporal modes [57, 141].

Using CV cluster states for quantum computation comes with a price. Ideal states are infinitely squeezed [20, 65]; 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 [2, 61, 65]. If left unchecked, this noise limits the length of computation possible using these states [60, 62]. Nevertheless, it is still possible to achieve universal fault-tolerant quantum computation with CV cluster states [87] by employing known quantum-error-correction protocols [67], 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 [170], whilst the lowest theoretical upper bound on the required squeezing for fault tolerant quantum computing is 20.5 dB [87].

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 [2, 3, 94].

In the same vein, here we give a new measurement protocol that is customized for a type of universal CV cluster state that is particularly scalable, known as the quad-rail lattice (QRL) [57, 58]. The generation procedure of the QRL is particularly simple owing to the fact that its graph [57, 58] is self-inverse and bipartite [55, 56]. Indeed, it needs only two-mode squeezed states (TMSSs) and a single 4-port linear optics gate (known as a foursplitter) as building blocks [57, 58]. 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 Ch. 3, which applies to the 1D resource state known as the CV dual-rail wire [2, 57, 58, 96, 141].

The structure of this Article is as follows: In Sec. 5.3 we review some basics of Gaussian pure states and the QRL [57, 58]. In Sec. 5.4 we introduce the basic components of our measurement protocol, including encoding, unitary gates, and measurement readout. In Sec. 5.5 we describe how these elements can be composed, allowing for flexible design of
quantum circuits. In Sec. 5.6 we compare this protocol to previous work. We conclude in Sec. 5.7.

5.3 Background

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

5.3.1 Symplectic formalism and gate definitions

The Heisenberg action of an \( N \)-mode Gaussian unitary \( \hat{U} \) acting on the vector of Heisenberg-picture operators \( \hat{x} = (\hat{q}, \hat{p})^T \) can be written as

\[
\hat{U}^\dagger \hat{x} \hat{U} = S_{\hat{U}} \hat{x},
\]

where we have ignored displacements and

\[
S_{\hat{U}} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}
\]

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

The phase-delay gate is defined to be

\[
\hat{R}(\theta) := \exp(i\theta \hat{a}^\dagger \hat{a})
\]

\[
= \exp \left[ \frac{i\theta}{2} (\hat{q}^2 + \hat{p}^2 - 1) \right].
\]

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

\[
R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.
\]

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

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

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

$$
\hat{S}(s) := \hat{R}(\text{Im} \ln s) \exp \left[ -\frac{1}{2}(\text{Re} \ln s)(\hat{a}^2 - \hat{a}^\dagger 2) \right],
$$

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

$$
|s| = e^r.
$$

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

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

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 $\pi$ phase delay when $s < 0$, this operation anti-squeezes $\hat{q}$ and squeezes $\hat{p}$ when $|s| > 1$, and vice versa if $|s| < 1$.)

The beamsplitter gate is defined to be

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

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

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

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. [2, 57, 59] 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.,

\[
\left[ \hat{B}_{ij}(\theta), \hat{U}_i \hat{U}_j \right] = 0,
\]

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

**Proof of Equation (5.10).** 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 kronnecker 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. \( \square \)

**The 50:50 beamsplitter gate** is defined as

\[
\hat{B}_{ij} := \hat{B}_{ij} \left( \frac{\pi}{4} \right),
\]

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 \( \hat{B}^\dagger_{ij} = \hat{B}_{ji} \).

**Figure 5.1:** (a) Two-mode continuous-variable cluster state represented using the full graphical calculus [59] (left) and the simplified graphical calculus [57] (right). Edge weights \( \varepsilon \) and \( t \) are defined in Eq. (5.20) and Eq. (5.21), 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.
Finally, the foursplitter gate is defined to be

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

(5.12)

where “h.c.” abbreviates the hermitian conjugate ($\dagger$) of the first term in the exponent. Its Heisenberg action on $\hat{x} = (\hat{q}_i, \hat{q}_j, \hat{q}_k, \hat{q}_l, \hat{p}_i, \hat{p}_j, \hat{p}_k, \hat{p}_l)^T$ is given by

$$
A_{ijkl} := \begin{pmatrix}
\hat{A} & 0 \\
0 & \hat{A}
\end{pmatrix},
$$

(5.13)

where $0$ denotes the $4 \times 4$ matrix of zeroes and

$$
\hat{A} = \frac{1}{2} \begin{pmatrix}
1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1
\end{pmatrix}.
$$

(5.14)

This gate admits the following convenient decompositions into four 50:50 beamsplitters [57, 58]:

$$
\hat{A}_{ijkl} = \hat{B}_{ij} \hat{B}_{kl} \hat{B}_{ik} \hat{B}_{jl} = \hat{B}_{ik} \hat{B}_{jl} \hat{B}_{ij} \hat{B}_{kl}.
$$

(5.15)

5.3.2 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 [59], which is defined using the graphical calculus for Gaussian pure states, summarized below.

Graphs.—Given an undirected, complex-weighted graph on $N$ nodes with adjacency matrix $Z (= Z^\dagger)$ and $\text{Im} \ Z > 0$ [59], $Z$ uniquely defines the position-space wavefunction

$$
\psi_Z(q) := \frac{(\text{det} \text{Im} \ Z)^{1/4}}{\pi^{N/4}} \exp \left[ \frac{i}{2} q^\dagger Z q \right]
$$

(5.16)

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

$$
(\hat{p} - Z\hat{q}) |\psi_Z\rangle = 0,
$$

(5.17)
where \( \hat{\mathbf{q}} = (\hat{q}_1, \ldots, \hat{q}_N)^T \) and \( \hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_N)^T \) are column vectors of operators. Every Gaussian pure state uniquely defines (up to phase-space displacements and overall phase) an associated graph \( \mathbf{Z} \) [59].

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

\[
\hat{U} \left| \psi_{\mathbf{Z}} \right\rangle = \left| \psi_{\mathbf{Z'}} \right\rangle \tag{5.18}
\]

with

\[
\mathbf{Z'} = (\mathbf{C} + \mathbf{DZ})(\mathbf{A} + \mathbf{BZ})^{-1} \tag{5.19}
\]

where the submatrices \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) and \( \mathbf{D} \) are defined via the Heisenberg action of \( \hat{U} \), as in Eq. (5.2).

Simplified graphs.—The Gaussian pure states that we consider in this chapter 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 [57]. 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 Fig. 5.1(a) and (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 localised to a single graph node, as shown in Fig. 5.1(c). The self-loops (not shown) on all non-input graph nodes have weight

\[
i\varepsilon := i \tanh 2r, \tag{5.20}
\]

where the squeezing parameter \( r \) gives the amount of vacuum squeezing used in preparing the state [57, 58]. All edge weights between different nodes are

\[
\pm C t := \pm C \tanh 2r, \tag{5.21}
\]

which is the product of the edge-weight coefficient \( C \) (specified on each figure) and a squeezing-dependent factor \( t \), along with a sign \( \pm \) denoted by blue/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
Chapter 5: Flexible quantum circuits using scalable continuous-variable cluster states

non-Gaussian and/or mixed states.

5.3.3 The quad-rail lattice

The QRL can be generated from a collection of two-mode cluster states [defined in Fig. 5.1(a)]\(^1\) arranged along edges of a square lattice by applying a foursplitter gate [Eq. (5.12)] to each four-mode lattice site, a.k.a. a macronode [57, 58]. The resulting QRL is defined by its four-layered square-lattice graph, as shown in Fig. 5.2(b). Further details about the generation of this state can be found in Refs. [57, 58].

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 [57, 58]. Graphically, this action is represented by node deletion [59], resulting in square-lattice CV cluster state as shown in Fig. 5.2(c). Up to displacements, this is the canonical resource state for universal MBQC with CVs [20, 65]. 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 \(C = \frac{1}{4}\) (instead of \(C = 1\)), which introduces excessive noise when used in a computation [2]. Here we introduce a different—and much more favorable—MBQC protocol that runs directly on the full QRL, Fig. 5.2(b).

5.4 Using macronodes for MBQC

The basic idea for our new 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. 5.4.1), measurements (Sec. 5.4.2), single-mode gates (Sec. 5.4.3), two-mode gates (Sec. 5.4.4), and measurement readout (Sec. 5.4.5).

5.4.1 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

---

\(^1\)Equivalently, two-mode squeezed states can be used by incorporating a \(\frac{\pi}{4}\) phase delay into the measurement of all nodes [168].
Figure 5.2: Simplified graphical-calculus representation [57] 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. (5.12)) 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) only contains one mode. In the original proposal [57], universal quantum computation proceeded via the standard measurement-based protocol [65]. Removing the extra nodes and links from (b), however, wastes squeezing resources [2]. Instead, our proposal directly employs the state shown in (b), making more efficient use of the available resources (the advantages are discussed in Sec. 5.6).

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,

$$\hat{a}_D := A^{-1}\hat{a}_P,$$

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

The mapping in Eq. (5.22) is exactly the inverse of a foursplitter gate [Eq. (5.12)]. Figure 5.3 displays the QRL with respect to the physical modes (a) and with respect to the distributed modes (b). Notice that the former has fence-like connections between adjacent macronodes, while the latter consists merely of disjoint pairs. Also notice that the graphs in Fig. 5.2(a) and Fig. 5.3(b) are visually identical. Nevertheless, they represent different physical states because they are defined with respect to different mode decompositions (physical and distributed, respectively).
Chapter 5: Flexible quantum circuits using scalable continuous-variable cluster states

Figure 5.3: (a), (b) Two equivalent ways to represent the quad-rail lattice using the simplified graphical calculus [57]. 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. (5.22) connecting the two. Red ellipses indicate the macronodes (4-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.

For the rest of this Chapter, 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. 5.3(c).

5.4.2 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

$$\hat{p}_\nu(\theta) := \left(\hat{p}_1(\theta_1), \hat{p}_2(\theta_2), \hat{p}_3(\theta_3), \hat{p}_4(\theta_4)\right)^T,$$

(5.23)

where $\theta := (\theta_1, \theta_2, \theta_3, \theta_4)$. 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. 5.3(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. 5.4) and thus 12 total input configurations 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. 5.5 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:

\[
\sigma = \begin{pmatrix} \hat{\sigma} & 0 \\ 0 & \hat{\sigma} \end{pmatrix}
\]

(5.24)

where \( \hat{\sigma} \) is some \( 4 \times 4 \) 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 four-splitter gate with each element of any generating set of all four-mode permutation gates.
Let $\sigma_{jk}$ denote the permutation gate that swaps modes $j$ and $k$. Then we have that

\begin{align}
A^{-1}\sigma_{1,2}A &= \sigma_{2,4}, \\
A^{-1}\sigma_{1,3}A &= \sigma_{3,4}, \\
A^{-1}\sigma_{1,4}A &= \sigma_{2,3}R_2(\pi)R_3(\pi),
\end{align}

where $R(\pi)$ is defined in Eq. (5.3). Thus, by commuting through the four-splitter gate, each 4-mode permutation gate $\sigma$ is mapped to a combination of a new permutation gates and some single-mode $\pi$ 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 $\pi$ phase delays (e.g., $\theta_i \rightarrow \theta_i + \pi$).

For the case shown in Fig. 5.5 and neglecting measurement-dependent displacements and finite-squeezing effects, the output state is given in Eq. (5.26).

\begin{equation}
|\psi\rangle_a |\phi\rangle_b \mapsto \hat{G}_{c'd'}(\theta) |\psi\rangle_{c'} |\phi\rangle_{d'},
\end{equation}

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

\begin{equation}
\hat{G}_{jk}(\theta) := \hat{B}_{jk}^\dagger \hat{V}_j(\theta_1, \theta_3)\hat{V}_k(\theta_2, \theta_4)\hat{B}_{jk}.
\end{equation}
Sandwiched between the pair of 50:50 beamsplitters is the single-mode unitary gate

$$\hat{V}_j(x, y) := \hat{R}_j \left( \frac{x + y}{2} \right) \hat{S}_j \left( \tan \left( \frac{x - y}{2} \right) \right) \hat{R}_j \left( \frac{x + y}{2} \right).$$ (5.28)

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

**Proof of Equation (5.26).** We start with Fig. 5.6, which shows a macronode measurement circuit where the four-splitter is decomposed into four beamsplitters [using Eq. (5.15)]. To go from Fig. 5.6(b) to Fig. 5.6(c) we used an interferometric symmetry of the pair of two-mode cluster states on modes ($c, c'$) and ($d, d'$) derived in Appendix 4.11 of Ch. 4: acting with $\hat{B}_{cd}$ on this state is equivalent to acting with $\hat{B}_{d'c'}$ instead [168].
Chapter 5: Flexible quantum circuits using scalable continuous-variable cluster states

Fig. 5.6(c) shows that macronode measurement is equivalent to two copies of a gate teleportation circuit [2, 3] conjugated by beamsplitters ($\hat{B}_{ab}$ and $\hat{B}_{d,c'}$). The gate teleportation circuits each implement

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

(5.29)

where $j$ and $k$ are 1 and 3 (2 and 4) for the top (bottom) subcircuit in Fig. 5.6(c), $\hat{V}$ is defined in Eq. (5.28), and

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

(5.30)

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

$$\hat{N}(r) = e^{-\varepsilon q^2 / 2} e^{-\varepsilon p^2 / 2} \hat{S}(t^{-1})$$

(5.31)

is a non-unitary operator that applies the noise from finite squeezing to the state (after which the state must be renormalized) [2].

The macronode measurement maps

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

(5.32)

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

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

(5.33)

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

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

(5.34)

and so

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

(5.35)

In the more general case, the displacements can either be actively corrected at each step or merely accounted for using feedforward [65]. From this, Eq. (5.26) can be seen as the large squeezing limit of Eq. (5.32). In the rest of this Chapter, we ignore displacements and finite-squeezing effects for simplicity of presentation.
Note that for $\theta_1 = \theta_3$ or $\theta_2 = \theta_4$, Eq. (5.28) 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. 5.4.5. Next we consider some examples of single- and two-mode Gaussian gates that are special cases of Eq. (5.27).

### 5.4.3 Single-mode Gaussian unitary gates

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

$$\theta_2 = \theta_1 \quad \text{and} \quad \theta_4 = \theta_3. \quad (5.36)$$

In this case, the single-mode gates sandwiched between the beamsplitters above in Eq. (5.27) are identical. Using Eq. (5.10), the beamsplitters cancel resulting in

$$\hat{G}_{jk}(\theta)|_{\theta_2=\theta_1} = \hat{V}_j(\theta_1, \theta_3)\hat{V}_k(\theta_1, \theta_3), \quad (5.37)$$

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) [2].

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

$$\hat{G}_{jk}(\theta)|_{\theta_4=\theta_3=\theta_2=\theta_1} = \hat{R}_j(2\theta_1)\hat{R}_k(2\theta_1), \quad (5.38)$$

and

$$\hat{G}_{jk}(\theta)|_{\theta_4=\theta_3=-\theta_2=-\theta_1} = \hat{S}_j(\tan \theta_1)\hat{S}_k(\tan \theta_1). \quad (5.39)$$
Figure 5.7: (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. (5.44), we can commute the phase delays past the foursplitter gate using Eq. (5.45), 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 (post-processing).

### 5.4.4 Two-mode Gaussian unitary gates

Here we provide different restrictions on the homodyne measurement angles $\theta$ that yield interesting examples of two-mode gates from Eq. (5.27). Setting

$$
\theta_3 = -\theta_1 \quad \text{and} \quad \theta_4 = -\theta_2
$$

implies the two-mode-squeezing operation

$$
\hat{G}_{jk}(\theta)\big|_{\theta_3=-\theta_1, \theta_4=-\theta_2} = \hat{B}_{jk}^\dagger \hat{S}_j(tan \theta_1) \hat{S}_k(tan \theta_2) \hat{B}_{jk}.
$$

We can also implement a linear-optics gate by setting

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

This implements

$$
\hat{G}_{jk}(\theta)\big|_{\theta_3=\theta_1-\frac{\pi}{2}, \theta_4=\theta_2-\frac{\pi}{2}} = \hat{B}_{jk}^\dagger \hat{R}_j \left(2\theta_1 + \frac{\pi}{2}\right) \hat{R}_k \left(2\theta_2 + \frac{\pi}{2}\right) \hat{B}_{jk}
$$

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

where $\theta_+ = \theta_1 + \theta_2$. Thus, up to some additional phase delays, the above gate implements a variable beamsplitter.
5.4.5 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:

\[
\theta = \theta_1 = \theta_2 = \theta_3 = \theta_4, \tag{5.44}
\]

as we now show. By decomposing the foursplitter gate \( \hat{A} \) using Eq. (5.15) and applying the beamsplitter commutation relations in Eq. (5.10), it is straightforward to verify that

\[
\left[ \hat{A}^{ijkl}, \hat{R}_i(\theta)\hat{R}_j(\theta)\hat{R}_k(\theta)\hat{R}_l(\theta) \right] = 0. \tag{5.45}
\]

Thus, with these restricted measurements, the foursplitter in the standard macronode measurement circuit as shown in Fig. 5.7(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 \( \tilde{A} \)) of the measurement outcomes:

\[
\hat{A}_{1,2,3,4}^{\dagger} \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{pmatrix} \hat{A}_{1,2,3,4} = \tilde{A} \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{pmatrix}. \tag{5.46}
\]

The physical four-splitter that is applied can be undone by classical post-processing (applying \( \tilde{A}^{-1} \)) 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. 5.7(b).

5.5 Constructing flexible quantum circuits

In the previous sections, we saw how input states can be encoded macrolocally [defined by Eq. (5.22)] and how homodyne measurements on macronodes are sufficient to implement
Circuit flexibility is limited because the wires are generated using a single-rail measurement-based protocol introduced in Refs [15, 20], but it is less amenable to scalable design than macronode-based approaches (see Section 3.16). Green macronode coloring indicates the application of a two-mode Gaussian unitary, such as those described in Eqs. (3.16). Orange macronode coloring is used to describe a different gate network. Depending on the homodyne angles, we can apply the same single-mode gates and teleport inputs at sites \( \alpha \) and \( \beta \) to \( \gamma \) and \( \delta \) respectively, for any valid assignment of \( \{\alpha, \beta, \gamma, \delta\} \mapsto \{a, b, c, d\} \). We represent this graphically as shown in Fig. 5.8(a) and (b).

By restricting measurements like this on a connected sequence \((i, j, k, \ldots, l)\) of macronodes on the QRL, (up to displacements and finite-squeezing effects) we can implement a single-mode Gaussian unitary \( \hat{V}_i \cdots \hat{V}_k \hat{V}_j \hat{V}_i \) (omitting dependence on homodyne angles) on an input initially encoded within macronode \( i \) and have it propagate through the sequence of macronodes \((i, j, k, \ldots, l)\), outputing into macronode \( l \). We illustrate this by way of example in Fig. 5.8(c). These sequences thus act as embedded quantum wires, equivalent to the CV dual-rail wires described in Ch. 3.

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...
Circuit flexibility is limited because the wires are gen-
us a single-rail measurement-based protocol introduced in Refs 
Due to finite squeezing per gate in order to implement a desired class of gates; (3) is the minimum number of sites that must be measured verability of the quantum wires; (2) four features: (1)
established CV cluster-state protocols? Below, we com-
ting photons. We leave further discussion of such elements resources known as magic states —Gaussian operations alone might more favorable to periodically inject non-Gaussian even include encoded qubits and error correction —The original CV

FIG. 8. (a), (b) Rather than labelling the modes within each macronode to indicate how inputs are mapped onto the lattice. 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. (5.37)]. Green macronode coloring indicates the application of a two-mode Gaussian unitary, such as those described in Sec. 5.4.4. Orange macronode coloring is used to describe a measurement readout step on the lattice, as in Sec. 5.4.5. (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.
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. 5.4.4. 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 multi-mode Gaussian unitaries. Measurement readout (homodyne detection) can be implemented by connecting up to four wires to a given macronode and measuring it with restrictions as in Eq. (5.44).

By combining these results, we have a highly flexible means for implementing quantum circuits on the QRL. See Fig. 5.9 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 [66]. Full universality can be achieved, however, by diverting a subset of the QRL nodes to photon counters instead of homodyne detectors [57, 65]. Depending on the particular practical implementation—which could even include encoded qubits and error correction [87]—it might be more favorable to periodically inject non-Gaussian resources [87, 169] instead of counting photons. We leave further discussion of such elements to future work.

5.6 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 [20, 65] uses a single-rail $C = 1$ 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 \[2, 145\], 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 \((\mathcal{C} = 1)\) \[2\]. 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 \[20, 65, 87\], but it is less amenable to scalable design than macronode-based approaches (see Ch. 4 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 \[2, 57\]. This resource state has the advantage that it can be generated scalably (by the process shown in Fig. 5.2). This protocol has the same features as in the \(\mathcal{C} = 1\) case except with poorer noise properties \[2\]. 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 Ch. 4. 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 \[2\]. 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 criss-cross 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 include two-mode squeezing and a variable beamsplitter. Thus, the QRL protocol is especially well suited to quantum-optics applications.
Chapter 5: Flexible quantum circuits using scalable continuous-variable cluster states

5.7 Conclusion

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 [65], these wires are embedded versions of the CV dual-rail wire (discussed in Ch. 3), and thus, they are more compact and do not introduce excessive levels of noise due to finite squeezing [2]. 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 Ch. 3 to the two-dimensional case.

Several novel features that our protocol exhibits—including nonlocal input states and the ability to re-route wires—are similar to those found in generalizations of measurement-based quantum computing based on tensor networks [31, 32]. 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 [3, 57, 58, 96, 141], 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.
Chapter 6

Measurement-Based Linear Optics

6.1 Abstract

A major challenge in optical quantum processing is implementing large, stable interferometers. Here we propose a virtual, measurement-based interferometer that is programmed on the fly solely by the choice of homodyne measurement angles. The effects of finite squeezing are captured as uniform amplitude damping. We compare our proposal to existing (physical) interferometers and consider its performance for BosonSampling, which could demonstrate post-classical computational power in the near future. We prove its efficiency in time and squeezing (energy) in this setting.

6.2 Introduction

Large-scale stable interferometers form the backbone of any optical architecture for processing photonic quantum information. This includes schemes for universal quantum computation, including linear-optics quantum computing [17], continuous-variable (CV) hybrid quantum computing [171, 172], and atomic-optical hybrid schemes [173], as well as other applications such as quantum metrology [174], quantum walks [175], and homomorphic encryption [176]. Linear-optical interferometers have attained new importance with the advent of BosonSampling [6], a subuniversal algorithm that shows great promise in demonstrating a clear quantum advantage over classical computers in the near future [6, 177].
In conventional experiments, these interferometric networks are typically built up out of bulk linear-optical elements (i.e., beamsplitters and phase delays). Though relatively simple to implement, these networks are limited by the scale and complexity afforded by the laboratory optics bench and are therefore unsuitable for large-scale applications [76].

One approach to solving this problem is to leverage integrated optics technology. Miniaturized optical elements can be lithographically printed on chip, which enables far greater scalability [178, 179]. Though this approach has shown great potential, such experimental architectures still fall short of the required scale for useful applications by several orders of magnitude. Combining many of these devices into a single interferometer to achieve the requisite scale is further hampered by mode mismatch, coupling losses, and nonuniformity in device properties.

Here we take a different path to large-scale and compact linear optics: measurement-based linear optics (MBLO). Rather than passing physical modes through optical elements in real space, MBLO implements very large virtual interferometers using highly compact cluster-state machinery [57, 58, 96, 98, 141].

The key feature of our proposal is the use of a scalable CV cluster state called the quad-rail lattice (QRL) [4, 57, 58]. Though any universal CV cluster state can in principle implement linear optics by measurement-based quantum computation (MBQC) [65], the QRL is uniquely suited to MBLO for three reasons. First, it can be generated on an unprecedented scale from compact experimental setups [96, 98, 141] using either temporal [57] or frequency modes [58]. Second, single- and two-mode linear-optics gates are naturally implemented using single-site measurements on the QRL, minimizing noise due to finite squeezing [2, 4]. Finally, this noise—which is ubiquitous in CV MBQC and usually appears as additive Gaussian noise [2, 65, 87]—can be coaxed into appearing as pure photon loss with efficiency $\gamma = \tanh^2 r$ for each simulated optical element, where $r$ is the overall squeezing parameter of the QRL.

As an application, we discuss efficient BosonSampling [6] using MBLO. Demonstrating post-classical computing with BosonSampling lends itself naturally to MBLO because of the size and variability of the interferometer required. We prove that BosonSampling using MBLO is simultaneously efficient in time and squeezing (as measured by average energy [63]). Such efficiency is necessary to show post-classical processing power.
Chapter 6: Measurement-Based Linear Optics

Figure 6.1: Generating and measuring the temporal-mode quad-rail lattice [57]. Squeezed vacuum states (a) pass through six balanced beamsplitters (b) and two delay loops (c) with delays $\Delta t$ and $M\Delta t$, with $M$ an odd integer. Cluster modes are generated and measured with homodyne detection (d) at the same rate. After $ML$ time steps of duration $\Delta t$, the QRL is an $(M \times L)$-macronode lattice. Input (output) states can be inserted (removed) using a switching device (e) [141] on any of the four rails (one example shown).

6.3 Linear optics with cluster states

For concreteness, we choose to illustrate MBLO using the temporal-mode implementation of the QRL [57], although analogous results hold for the frequency-mode version [58]. The full experiment to generate and use the QRL using temporal modes is shown in Fig. 6.1. Its compactness is evident. Homodyne detection alone programs and implements the desired linear optics.

The QRL is a macronode-based CV cluster state [56] (with 4 modes per macronode) that has a square-lattice topology [4, 57]. It is convenient to describe the QRL in terms of distributed modes, which are balanced linear combinations of the four physical modes that make up each macronode [4]. Up to two of the four distributed modes within a given macronode contain the quantum information encoded at that site in the lattice.

Suppose that quantum information is encoded within $f(a)$ and $f(b)$, where $f$ is an arbitrary automorphism on the mode labels reflecting the permutation symmetry inherent to computing on the QRL [4]. Then, mode $f(c)$ and its partner $f(c)'$ (indicated by the prime) within an adjacent macronode are in a two-mode squeezed state $^1$, and similarly for $f(d)$ and $f(d)'$.

$^1$ Ch. 5 uses two-mode CV cluster states instead of two-mode squeezed states, but Footnote 1 in that reference explains that the two conventions are related simply by rotating all homodyne angles by $\frac{\pi}{4}$. We use the former convention because it connects this work more naturally to CV teleportation [149].
Define the two-mode linear-optics gate

\[
\hat{V}_{ij}(\theta, \phi) := \hat{R}_i(\theta) \hat{R}_j(\theta) \left[ \hat{R}_i \left( \frac{\pi}{2} \right) \hat{B}_{ij}(\phi) \hat{R}_j \left( \frac{\pi}{2} \right) \right]
\]

\[
= \hat{B}_{ij}^\dagger \left( \frac{\pi}{4} \right) \hat{R}_i \left( 2\xi_+ \right) \hat{R}_j \left( 2\xi_- \right) \hat{B}_{ij} \left( \frac{\pi}{4} \right)
\]

(6.1)

where \( \xi_{\pm} := \frac{1}{2}(\theta \pm \phi - \frac{\pi}{2}) \), \( \hat{R}_j(\theta) := e^{i\theta \hat{a}_j^\dagger \hat{a}_j} \) is a phase delay by \( \theta \) on mode \( j \), and \( \hat{B}_{ij}(\phi) := e^{-\phi(\hat{a}_i^\dagger \hat{a}_j - \hat{a}_j^\dagger \hat{a}_i)} \) is a variable beamsplitter on modes \( i \) and \( j \). The second line of Eq. (6.1) is a Mach-Zehnder-type decomposition of \( \hat{V}_{ij} \). For any given macronode, there exists a choice of homodyne measurement angles that will implement \( \hat{V}_{ij} \) on the encoded information and teleport it from \( f(a) \rightarrow f(c)' \) and from \( f(b) \rightarrow f(d)' \) [4], i.e.,

\[
|\psi\rangle_{f(a)} |\varphi\rangle_{f(b)} \mapsto \hat{V}_{ij}(\theta, \phi) |\psi\rangle_{f(c)'} |\varphi\rangle_{f(d)'}.
\]

(6.2)

This operation, which we refer to as a measurement-based Mach-Zehnder (MBMZ), is the primitive for MBLO. Importantly, it is merely the choice of homodyne angles that determines which gate is applied. The ubiquitous measurement-dependent displacements and finite-squeezing effects of CV MBQC [65] are discussed in the following section.

Large networks of beamsplitters and phase delays can be implemented by composing
MBMZs on the QRL. We consider a (generalizable) six-input example in Fig. 6.2. The flow of inputs (i.e., $f$) through each macronode is indicated by the orange and purple ribbons. The color of each macronode indicates which gate is applied: white applies $\hat{V}(\pi, 0) = \hat{I}$, blue applies $\hat{V}(\theta - \pi, 0) = \hat{R}(\theta) \otimes \hat{R}(\theta)$, and green applies $\hat{V}(\theta, \phi)$. As the first two cases are single-mode gates, the inputs teleport through the macronode without interacting (like non-interacting quantum wires) [4].

Each of the six blue macronodes in the first column of Fig. 6.2 contributes one phase degree of freedom. Together, these are sufficient to implement arbitrary phase delays on the inputs. For columns in the bulk, green macronodes implement variable beamsplitters, and both green and blue macronodes contribute one phase degree of freedom each, altogether allowing arbitrary independent phase delays to act on each mode after the beamsplitters. Therefore, the total logical action of the teleportation network in Fig. 6.2 is equivalent to the linear optics shown in Fig. 6.3. Arbitrary $m$-mode interferometers require an $(m + 1) \times (k + 2)$-macronode QRL, where $k$ is the depth of the network. A general $m$-mode interferometer can be decomposed into a depth $k = m$ network of this type [180], although for some applications, a smaller network may be sufficient ($k < m$). Note that each path through the QRL crosses $2(k+1)$ macronodes (excluding the output macronodes).

6.4 Finite squeezing as uniform loss

We now analyze the role of displacements and finite squeezing in MBLO. Each macronode measurement, illustrated in Fig. 6.4(a), displaces the input states in phase space by an amount dependent on the (random) measurement outcomes $m_{f(i)}$ [4]. We pretend that these displacements are undone after each macronode measurement (using $\hat{D}_1$
and $\hat{D}_2$ in the figure), but in practice all displacements will be corrected in one shot at the very end [65].

This circuit can be restructured into a pair of CV teleportation circuits [149] sandwiched between linear optics, as shown in Fig. 6.4(b). To see this, note that equal phase delays commute past the preceding 50:50 beamsplitter [4]. Next, phase delays acting on modes $f(c)$ and $f(d)$ can be transferred to modes $f(c)'$ and $f(d)'$, respectively, using the symmetries of a two-mode squeezed state [168, 181]. Finally, the displacements $\hat{D}_1$ and $\hat{D}_2$ are commuted past the Gaussian unitaries $\hat{B}(\frac{\pi}{4})$ and $\hat{R}(\xi \pm)$, resulting in a new set of displacements, $\hat{D}_3 = \hat{D}(g\alpha_3)$ and $\hat{D}_4 = \hat{D}(g\alpha_4)$, where $\hat{D}(\alpha) := e^{a\hat{a}^\dagger - \alpha^* \hat{a}}$, $\alpha_3 := im_{f(c)} + m_{f(a)}$, and $\alpha_4 := im_{f(d)} + m_{f(b)}$. The gain parameter $g > 0$ allows us to tailor the noise associated with the teleportation [182], as shown next.

The case $g = 1$ corresponds to the original CV teleportation protocol [149, 183]—i.e., an identity gate with additive Gaussian noise introduced into the evolution [2, 4, 149, 183]. Since this type of noise involves photon creation, it is undesirable for linear optics. Gain tuning may also enable tailoring the noise for specific applications [182] in more general CV cluster-state computations.

By setting $g = \tanh r$, the noise model becomes pure amplitude damping (photon loss) [182, 184, 185], as shown in Fig. 6.4(c), with efficiency $\gamma = g^2 = \tanh^2 r$. Direct calculation using the symplectic representation (with loss modeled as a beamsplitter [186]) shows that the same loss channel $\mathcal{L}$ applied to each of $m$ modes commutes with arbitrary (lossless) linear optics on those modes. This shows the equivalence of (c) and (d) in Fig. 6.4.

Since the green macronodes in Fig. 6.2 are uniformly spaced, we can further commute all loss to the beginning of the entire network. Then, finite squeezing for a depth-$k$ MBLO circuit results in an effective loss channel with efficiency $\gamma_{\text{eff}} := \gamma^{2(k+1)} = (\tanh r)^{4(k+1)}$ applied to each input state before (or, equivalently, after) the implemented linear optics, which are now considered lossless.

This conversion of squeezing into loss enables direct comparison with other quantum computing architectures. In particular, we can use $\gamma_{\text{eff}}$ to compare the squeezing demands of MBLO to actual losses in a physical interferometer. A recent experiment [179] implemented a general 6-mode interferometer with 42% average insertion loss, corresponding to $\gamma_{\text{eff}} = 0.58$. Achieving the same performance in MBLO (in an otherwise lossless implementation with depth $k = 6$) would require $r \approx 2.32$, corresponding to 20.1 dB of squeezing. (Note: $\#\text{dB} = 10 \log_{10} e^{2r} \approx 8.69 r$). While this is experimentally demanding, it is within reach of near-term technology given that the state of the art is now 15 dB [187].
Chapter 6: Measurement-Based Linear Optics

Figure 6.4: (a) Homodyne detection on one macronode in the quad-rail lattice (with respect to the distributed modes) can be represented by the above circuit [4]. We have selected homodyne angles such that the transformation in Eq. (6.2) is implemented. (b) The macronode circuit is decomposed as a pair of CV teleportation circuits [149] (shown in the green boxes) sandwiched between linear optics. (c) Macronode measurement with finite-squeezing effects and gain-tuned displacements [182] is equivalent to a Mach-Zehnder interferometer with equal loss $L$ on each arm. (d) Because the loss $L$ is the same on both arms, it can be commuted to the beginning (see text).

6.5 BosonSampling with MBLO

With frugal experimental requirements, BosonSAMPLING efficiently samples from a distribution that is strongly believed to be computationally hard to simulate [6], making it of great interest as a potential candidate for demonstrating the first post-classical quantum algorithm.

In MBLO-based BosonSAMPLING, we inject (Fig. 6.1) $n$ single photons and $m - n$ vacuum states into an $(m + 1) \times (k + 2)$-macronode QRL. We choose $m = n^2$ in order to
ensure collision-free output configurations [6] and $k = m$ so we can implement an arbitrary unitary with nearest-neighbor interactions (Fig. 6.3) [180]. Alternatively, one could implement Gaussian (“scattershot”) BosonSampling [188, 189] by attenuating the squeezing in some of the two-mode squeezed states at the beginning of the protocol and photodetecting half of each one. This would nondeterministically project input states into either vacuum or single-photon states. After MBLO, the output is switched out of the QRL (see Fig. 6.1), appropriately displaced, and measured via photodetection, thereby sampling from a statistical distribution of photon-number configurations [6].

With amplitude damping inherent to MBLO (due to finite squeezing), sometimes $< n$ photons in total will be measured at the output. Such an instance is a failure, and we repeat the protocol until success. MBLO has efficiency $\gamma_{\text{eff}}$ for each mode, so the success probability of the device is $\gamma_n^m$ (the $m - n$ vacuum states are unchanged by loss), and it takes $T := \gamma_n^{-m}$ trials on average to yield a single successful measurement event.

For an efficient implementation, it is necessary that $T = T_p(n)$ is some fixed polynomial in $n$. Clearly, if $r$ is held constant, then $T = (\coth r)^4n(k+1)$ grows exponentially in $n$. We can reduce this scaling by allowing the squeezing parameter $r$ to grow with $n$, but for efficiency in the squeezing [63], this must scale at most logarithmically with $n$ in order to ensure the average energy $E \propto \sinh^2 r = O(e^{2r})$ is polynomial in $n$.

Let $\ell := 4n(k + 1) = 4n(n^2 + 1)$, using $k = m = n^2$. Then, noting that the function $x \mapsto \frac{1}{2} \log \coth x$ is self-inverse and that $1 < \coth x < 1 + x^{-1}$ for $x > 0$, the relation $T = (\coth r)^\ell = T_p(n)$ implies

$$r = \frac{1}{2} \log \coth \left( \frac{\log T_p}{2\ell} \right) < \frac{1}{2} \log \left( 1 + \frac{2\ell}{\log T_p} \right), \quad (6.3)$$

ensuring that simultaneous efficiency in both time and squeezing is possible.

Having proven efficiency at the theoretical level, we now address practicality. We assume the temporal-mode implementation (Fig. 6.1) with wave-packet duration $\Delta t \approx 150\text{ns}$ [141], which means a single experiment requires $\tau := (m + 1)(k + 2)\Delta t = (n^2 + 1)(n^2 + 2)\Delta t$ time to complete. Once $n$ photons have been successfully injected as inputs, a successful experiment will take time $\tau T$ on average. In Fig. 6.5, using Eq. (6.3), we plot the squeezing required for $\tau T = 1 \text{ day}$ for various $n$. The narrowness of the blue region (1 min $< \tau T < 1 \text{ year}$) demonstrates that the blue line is effectively a hard boundary because of the exponential scaling with constant $r$. In the lower (purple) region, greater than astronomical timescales would be required, while experiments in the upper (white) region are split second.
Notice that squeezing levels \( \sim 17 \text{ dB} \) (only 2 dB higher than current levels [187]) would enable sampling 5 photons from 25 modes, which would outperform the largest experimental demonstration to date (4 photons from 8 modes) [190]. Finally, the recently reported 6-mode interferometer [179] has efficiency corresponding to 20.1 dB of squeezing (as noted above), which would enable MBLO sampling of 6 photons from 36 modes—a much larger interferometer.

Our results are optimistic in that we have neglected additional noise sources (discussed below). On the other hand, they are also conservative in that we postselect on no lost photons for BOSON-SAMPLING. Tolerating some loss through approximate BOSON-SAMPLING [191] will likely allow for even lower squeezing while retaining computational hardness [192].

### 6.6 Conclusion

Measurement-based linear optics offers a novel approach to large-scale interferometry. With MBLO, we get a large virtual interferometer from a compact physical setup. This
compactness will be an advantage when experimentally confronting the usual sources of noise—e.g., mode mismatch [191], coupling losses [141], and phase locking [98].

One might question the wisdom of employing squeezing (a nonlinear operation) for linear optics. For small experiments, this would be a valid objection. The value of MBLO lies in its simplicity and flexibility when implementing large-scale interferometers.

Spurred on by the recent detection of gravitational waves [193], there is significant ongoing experimental drive to improve squeezing technology [170, 194, 195] for next-generation gravitational-wave astronomy [196]. Experimental squeezing levels have recently been elevated to 15 dB [187] with high homodyne efficiency (99.5%), high phase sensitivity (1.7 mrad), and low total optical losses (2.5%). Progress in improving squeezing—a physical and information-theoretic resource [63, 89]—will directly translate into payoffs for other squeezing-based applications [87], including MBLO.
Chapter 7

Conclusion

Ever since its original conception, the measurement-based model for quantum computation has sparked new experimental and theoretical insights into the nature of quantum computation. In this thesis we have presented several measurement-based protocols that exhibit various novel and practical features. In every case, these features emerge from a careful examination of how basic gates are implemented by single-site measurements.

In Ch. 2 we introduced two measurement-based schemes for single-qubit randomized benchmarking. Rather than complicating the protocol, we saw that the intrinsic randomness of measurement-based quantum computation could be harnessed to simplify it. An important next step is the generalization to higher dimensional cluster states. As discussed in Ch. 2, many of the features of the Clifford protocol should carry over to the universal two-dimensional cluster state setting. This work opens up the possibility of developing methods for characterizing more general forms of measurement-based computations, such as non-network models [30].

For scalable continuous-variable cluster states, the benefits of a resource tailored measurement protocol were made clear in Chs. 3, 4, and 5. We showed that the one-dimensional macronode protocol had better noise properties and were more compact than the standard measurement protocol. Furthermore, these advantages were also shown to carry over to two different measurement protocols over universal resource states.

These results motivate a paradigm shift whereby canonical continuous-variable cluster states are replaced by variants with bipartite self-inverse graphs [54, 59]. We highlight two key advantages of this approach. The first is that such states can be generated using a very simple constant-depth linear optics circuit and a single OPO, as shown in Ch. 4.
Chapter 7: Conclusion

The second is that measurement-based protocols over these states can exhibit novel and practical features, such as enhanced circuit flexibility discussed in Ch 5.

Spurred on by recent success in the one-dimensional case [96, 98, 141], there is an immediate call for experimental demonstrations of large-scale universal resource state generation and logic gate implementation. In parallel to these developments, there is a pressing need for detailed theoretical analysis that addresses physical sources of noise (such as losses, state impurities, etc). Intimately linked to such concerns is the question of how to best to incorporate quantum error correction machinery, with the ultimate goal being a large-scale, universal, and fault-tolerant quantum computer.

One possible approach is to attempt a two-way extension of the fault-tolerance result in Ref. [87]: it would likely be adaptable to the setting of scalable continuous-variable cluster states and generalized to include physical sources of noise (as well as finite squeezing). Alternatively, one could try harness measurement-based linear optics for the purpose of implementing fault-tolerant linear optics quantum computation [17, 18, 74].

The strength of continuous-variable cluster state computing lies in the scalability and versatility of the resource states. The results in Ch. 6 attest to this by showing how a finitely-squeezed cluster state affords an elegant reinterpretation as a lossy interferometer. This connection allows for new comparisons to be drawn between starkly different optical quantum processing architectures, and opens new pathways towards developing a scalable quantum processor.
Bibliography


145


[70] T. Aoki et al., “Quantum error correction beyond qubits”, 2009. 16


Bibliography


154


155


