Classical Simulations of Gaussian Boson Sampling

by

Rachel Sarah Chadwick

University of Bristol

School of Physics

October 2021

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Doctor of Philosophy in the Faculty of Science.

Word count: $\sim 43,000$

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE:.....

Abstract

There is a lot of effort currently being poured into the development of quantum technologies in the hope that they will speed up certain tasks significantly. Boson sampling is a task that can be run on a near-term device and so was proposed as a promising candidate for a demonstration of quantum advantage. As experiments are performed of a variant, Gaussian boson sampling, claiming to have demonstrated this, it is increasingly important to know where that boundary lies and to improve classical algorithms for simulating Gaussian boson sampling. Furthermore, there has been a wealth of applications of Gaussian boson sampling suggested beyond just a method for reaching quantum advantage. These applications raise the question of whether a quantum-inspired classical algorithm could be used to improve current methods of solving these problems. In this thesis we explore approximate Monte Carlo and chain-rule algorithms for simulating Gaussian boson sampling to minimise the complexity of sampling under perfect conditions with the aim that they can be applied to quantum-inspired algorithms. They could also be adapted to include experimental imperfections for simulating experiments.

Acknowledgements

During my five years in Bristol, I am fortunate to have worked with so many great people in QETLabs. At times, being surrounded by so many intelligent people can be demoralising, but one of the most important things I learned during these years is just how privileged I am to have smart friends who inspire and motivate me to be better. With that in mind, this is an opportunity to thank some of the people who have made this thesis possible, but there are many deserving people who I will have missed for which I apologise.

First I must thank the QECDT. Thank you to the management for giving me this opportunity and guiding me through the first year making me into a quantum physicist. Thank you to my cohort, Brian, David, George, Giorgos, João, Joe, Jorge, Konstantina, Max, Ross, and Will with whom I have shared many great experiences and without whom I would not have enjoyed my first year nearly as much.

Secondly, to the 'simulations' group (the Laing gang), I'm honoured to be a part of such a fantastic group of people who have the perfect combination of being both extremely helpful and intelligent. To those in the office, I am always grateful for the many conversations that are occasionally useful but always entertaining and my time in Bristol would have been much less interesting without you. Alex M, thanks for all the tech/coding support (even when I didn't ask for it) and the insightful conversations about sleep amongst other things. Patrick, thanks for asking every single day if I had "sampled any bosons", pretending to understand my explanations about hafnians, and entertaining me with magic tricks (I still haven't figured them out). Nicola, thanks for listening to me moan about my work not working and making useful suggestions. Jake, thanks for your shared love of boson sampling, involving me in many projects, and scheduling meetings during my 'working hours'. I would definitely not have progressed nearly as much during my PhD without your help. A special mention to Konstantina, my desk neighbour, hoop partner and general partner-in-crime. It's fair to say that most of the highlights of my time in Bristol have been with you, particularly in hoop training (I hope I did not cause you any permanent damage!) and our adventures abroad. I hope we will always be so in sync. Is a PhD *necessary*? Probably not. Is it *possible*? Yes, Dr Koteva.

I am lucky to have worked with several post-docs over the years. Alex N, thank you for bearing with me when I had no idea what I was doing, and introducing me to boson sampling and classical sampling algorithms. Levon, I am grateful for your patience, help and understanding during the many times I was adamant I was right despite in fact being wrong. Alex J, thanks for calming me

down and being the voice of reason during my panicking moments.

Thank you Anthony for your supervision during my PhD, and for your general life advice even if I have not always taken it. I appreciate your commitment to getting me across the finish line. I would also like to acknowledge Jake and Anthony for valuable feedback on this thesis.

I am grateful to my co-authors for making it possible for me to become an author. I'm proud to have my name alongside each of yours. I am especially thankful to Nico for going above and beyond collaborator expectations.

Finally, I'd like to thank my family. I'm sorry that I still can't explain what it is I do, but I appreciate your interest despite this. To my parents, I would not exist without you. But your support (both emotionally and financially) has made my path through life to this point not only possible, but easy too. I hope that with the submission of this thesis, you will be a little less stressed on my behalf.

"Everything is possible. The impossible just takes longer." - Dan Brown

List of Publications

Work explored in this thesis has been presented in the following manuscripts:

- N. Quesada, **R. S. Chadwick**, B. A. Bell, J. M. Arrazola, T. Vincent, H. Qi, and R. García-Patrón. *Quadratic Speed-Up for Simulating Gaussian Boson Sampling*, PRX Quantum 3, no. 1 (2022): 010306.

- J. F. F. Bulmer, B. A. Bell, **R. S. Chadwick**, A. E. Jones, D. Moise, A. Rigazzi, J. Thorbecke, U.-U. Haus, T. V. Vaerenbergh, R. B. Patel, I. A. Walmsley, and A. Laing. *The boundary for quantum advantage in Gaussian boson sampling*, Science Advances 8, eabl9236 (2022).

The algorithm introduced in"Quadratic Speed-Up for Simulating Gaussian Boson Sampling" is detailed in section (5.2.2). The techniques explored in "The boundary for quantum advantage in Gaussian boson sampling" for analysing the convergence and independence in MIS are described in section (4.3). In the manuscript they are applied to lossy GBS for one proposal distribution, which is given in section (4.2.5), whereas in this thesis they are applied to the lossless case for a variety of distributions.

Contents

1 Introduction **5**

1.1 Outline of thesis 6

2 Discrete and Continuous Variable Photonics **9**

2.1 Quantum states 9

2.2 Phase-space formalism 13

2.2.1 Describing states 13

2.2.2 Gaussian transformations 16

2.2.3 Gaussian states 20

2.2.3.1 Coherent states 20

2.2.3.2 Squeezed vacuum states 22

2.2.3.3 Squeezed coherent states 24

2.3 Permanents, hafnians and loop hafnians 29

2.3.1 Permanent 29

2.3.2 Hafnian 31

2.3.3 Loop hafnian 32

2.3.4 Connection between permanents, hafnians and loop hafnians 32

2.4 Boson sampling 33

2.4.1 Standard boson sampling 34

2.4.2 Scattershot boson sampling 37

2.4.3 Gaussian boson sampling 38

2.4.3.1 General Gaussian boson sampling 38

2.4.3.2 Reduction to scattershot boson sampling 44

2.4.3.3 Retrodictive intuition behind Gaussian boson sampling 46

2.4.4 Sampling from a probability distribution 49

3 Simulating Gaussian Boson Sampling with Rejection Sampling **51**

3.1 Rejection sampling 53

3.1.1 Exact rejection sampling 53

3.1.2 Approximate rejection sampling 57

3.2 Rejection sampling applied to Gaussian boson sampling 59

3.2.1 Sampling the number of photons . . . 61
3.2.2 Proposal distributions . . . 62
3.2.2.1 The stepped uniform distribution . . . 62
3.2.2.2 Peaked - rejection sampling . . . 64
3.2.2.3 Adaptive rejection sampling . . . 65
3.2.3 Hill climbing optimisation algorithms . . . 66
3.2.3.1 Steepest ascent hill climbing . . . 67
3.2.3.2 Stochastic hill climbing . . . 68
3.3 Numerical analysis of the accuracy and efficiency of Gaussian boson sampling with rejection sampling . . . 68
3.3.1 Comparison of hill climbing algorithms . . . 69
3.3.2 Efficiency of hybrid hill climbing . . . 74
3.3.3 Post-sampling analysis of the accuracy of rejection sampling . . . 76
3.3.3.1 Upper bound for the TVD between target and sampled distributions 78
3.3.3.2 An estimation of the exact TVD between target and approximate distributions . . . 79
3.3.3.3 Comparison of the upper bound method and exact method . . . 80
3.3.4 Complexity of Gaussian boson sampling with rejection sampling . . . 83
3.4 Discussion . . . 85

4 Simulating Gaussian Boson Sampling with Metropolised Independence Sampling 87
4.1 Markov chain Monte Carlo . . . 87
4.1.1 Markov chains . . . 87
4.1.2 Monte Carlo . . . 89
4.1.3 Metropolis-Hastings algorithm . . . 89
4.1.4 Metropolised independence sampling . . . 90
4.2 Metropolised independence sampling applied to Gaussian boson sampling . . . 92
4.2.1 Stepped uniform proposal . . . 92
4.2.2 Peaked proposal . . . 93
4.2.3 Coherent states . . . 94
4.2.4 Distinguishable squeezed states . . . 95
4.2.5 Independent pairs and singles . . . 102
4.3 Numerical analysis of the efficiency and accuracy of Metropolised independence sampling . . . 105
4.3.1 Acceptance rate test . . . 107
4.3.2 Testing the burn-in time . . . 111
4.3.3 Testing the thinning interval . . . 112
4.4 Discussion . . . 113

5 Chain-Rule Methods in Boson Sampling **117**
5.1 Sampling photon by photon in boson sampling 117
5.1.1 Chain rule of probability 117
5.1.2 Standard boson sampling 118
5.1.3 Gaussian boson sampling 121
5.2 Sampling mode by mode in Gaussian boson sampling 122
5.2.1 Sampling with mixed probabilities 122
5.2.2 Sampling with pure probabilities 124
5.3 Sampling mode by mode in standard boson sampling 129
5.4 Discussion 135

6 Conclusion **137**

List of Figures

2.1 Coherent and squeezed states in phase space . . . 23
2.2 Displaced squeezed states and squeezed displaced states equivalence . . . 25
2.3 General squeezed displaced state . . . 27
2.4 Properties of graphs . . . 30
2.5 The adjacency matrix of a graph . . . 31
2.6 The mapping between permanents, hafnians and loop hafnians . . . 33
2.7 Scattershot and GBS equivalence . . . 46
2.8 Retrodictive description of GBS . . . 48

3.1 Inverse transform sampling . . . 54
3.2 Inverse transform sampling and rejection sampling . . . 55
3.3 The error in approximate rejection sampling . . . 57
3.4 Proposal distributions for rejection sampling for GBS . . . 59
3.5 TVD for stepped uniform distribution . . . 69
3.6 TVD for peaked distribution . . . 70
3.7 Success probability for stepped uniform proposal . . . 71
3.8 Success probability for peaked proposal . . . 72
3.9 The efficiency of hill climbing . . . 75
3.10 Testing the estimation of the TVD . . . 81
3.11 The scaling with photon number of the number of probabilities in rejection sampling 84
3.12 The scaling with number of modes of the number of probabilities in rejection sampling 84

4.1 Proposal distributions for MIS . . . 93
4.2 State dependent proposal distributions for MIS . . . 106
4.3 The probability of accepting in MIS. . . . 108
4.4 Comparison of the probability of accepting . . . 109
4.5 The probability of accepting over the length of the chain . . . 110
4.6 Determining the required burn-in . . . 110
4.7 The burn-in time as a function of the number of photons . . . 111
4.8 Thinning intervals for the MIS proposal distributions . . . 114
4.9 Thinning intervals as a function of the number of photons . . . 114

5.1 TVD when brute force sampling and using the pure state chain-rule algorithm . . . 130
5.2 Comparison of the GBS chain-rule sampling algorithms 131

List of Tables

2.1 A summary of the mapping between quadrature and ladder descriptions 16

6.1 A comparison of the complexity of boson sampling variants 138

Chapter 1

Introduction

"Many of life's failures are people who did not realize how close they were to success when they gave up." - Thomas A. Edison

Quantum mechanics is not only useful to describe the world around us but can have technological applications with a rising interest in the development of universal quantum computers. There are many proposed platforms to achieve this, notably superconducting [1], trapped ions [2] and photonics [3]. However, the experimental challenges this involves in any platform are huge and we are still far off a realisation of such a device.

A first milestone in the assurance of the potential of quantum technologies is to experimentally demonstrate a task on a quantum device that vastly outperforms a classical computer. There are two terms commonly used in literature to describe this milestone: 'quantum advantage' and 'quantum supremacy'. Sometimes these terms are used interchangeably, but often quantum supremacy is reserved for more significant progress. In general, an improvement of many orders of magnitude will be required for a convincing demonstration. A few near-term devices have been proposed as promising candidates for a demonstration of quantum advantage where the hardware is designed for a specific task reducing the experimental requirements. One example is random circuit sampling [4], a near-term task for superconducting qubits with the only known application being for a quantum advantage demonstration which was claimed by the Google team [5]. In the field of quantum photonics, *boson sampling* is a task that samples from a probability distribution determined by matrix functions that are classically intractable to evaluate [6]. The original proposal of boson sampling involves sending single photons through a linear interferometer and measuring the resulting entangled state in the number basis. Since its conception, various alternatives have been proposed with Gaussian states (appropriately called Gaussian boson sampling) [7], Gaussian measurements [8], and heralded input photons [9], motivated by reduced hardware constraints. In order to determine whether an experiment is sufficiently difficult to pass the threshold of quantum advantage, it is important to find the best possible classical algorithms with which to compare.

In recent years, there have been two claims of quantum advantage by Gaussian boson sampling (GBS) [10, 11] and so it is important to be able to simulate the experimental conditions for a fair

comparison. The main imperfections that occur are photon loss and distinguishability. These have been researched extensively in the context of the original boson sampling proposal [12, 13, 14] but as GBS is a more recent area, the equivalent in GBS is somewhat behind albeit progressing [15, 16].

Although boson sampling was originally proposed for the sole purpose of demonstrating quantum advantage, many potentially useful applications have been suggested, particularly for Gaussian boson sampling, such as for simulating vibronic spectra [17, 18], molecular docking [19], graph matching [20], graph similarity [21, 22, 23], and point processes [24]. This raises the question of whether the quantum technology can motivate quantum-inspired algorithms to improve upon existing classical algorithms for these applications. Therefore it is an interesting problem to optimise the classical simulations of quantum devices. The first problem when optimising sampling algorithms is to find a method of sampling that does not involve the calculation of the whole sample space and is limited only by the complexity of a single or polynomial number of probabilities. This is applicable to all uses of classical simulations of GBS whether to push the limit of quantum advantage or for a quantum-inspired algorithm and is the focus of this thesis.

1.1 Outline of thesis

In this thesis we look at several ways to simulate GBS, using both Monte Carlo approximate methods and tricks to sample directly from the probability distribution. The aim is to find the classical complexity of GBS in the same way that it was found for standard boson sampling [25, 26]. With the applications of Gaussian boson sampling in mind, we choose to simulate the exact case without experimental imperfections but allowing for the inclusion of displacement in the input states. These methods can be extended to include experimental imperfections, but we do not cover that in this thesis. The structure is as follows:

In chapter 2, we introduce the terminology and theoretical concepts used throughout this thesis. We cover the important foundations of quantum optics, focussing on the description of Gaussian states and transformations, as well as matrix functions that underpin the boson sampling probability distributions. Finally, a detailed review of many variants of boson sampling is provided.

In chapter 3, we study the first Monte Carlo method, rejection sampling, as a tool to sample from the GBS distribution. We consider two proposal distributions and analyse the efficiency and accuracy of using these. A method to estimate the accuracy of the sampling in retrospect is developed with numerical tests to confirm its use. In chapter 4, we apply a second Monte Carlo method, a Metropolis-Hastings algorithm, to simulate GBS. Here we suggest more proposal distributions motivated by the states in the quantum device to try to find a closer match to the GBS distribution. We numerically test the requirements on the parameters to ensure good accuracy and observe the efficiency for the problem sizes we are able to fully calculate.

In chapter 5, we move away from approximate algorithms instead exploring the chain rule of probability to sample more efficiently. We introduce a new algorithm that reduces the time complexity from the previous best of $O(N^3 2^N)$ for N photons to $O(N^3 2^{N/2})$. Furthermore, we apply an algorithm that samples each mode at a time in GBS to standard boson sampling to find

an algorithm of $O(N2^{2N})$ for N photons. Finally, in chapter 6, we summarise our findings and provide an outlook of open problems.

Chapter 2

Discrete and Continuous Variable Photonics

"Only a Sith deals in absolutes." - Obi-Wan Kenobi

In this chapter we introduce the foundations that the rest of this thesis builds on. We begin with an overview of the description of photons both with discrete and continuous variables, with a particular focus on Gaussian states. Then we discuss matrix functions that arise in boson sampling and their role in graph theory, culminating in a thorough description of boson sampling, the focus of this thesis. Although it is largely a literature review, we hope that there is significant personal insight and potentially novel interpretations.

2.1 Quantum states

A photon is an excitation of a normal mode of the electromagnetic field and so can be described by a quantum harmonic oscillator. The Hamiltonian of a quantum harmonic oscillator for a single mode is given by [27, 28]

$$\hat{H} = \frac{-\hbar^2}{2m}\frac{\partial^2}{\partial^2 \hat{Q}} + \frac{1}{2}m\omega^2\hat{Q}^2 = \frac{\hat{P}^2}{2m} + \frac{1}{2}m\omega^2\hat{Q}^2, \tag{2.1}$$

where m is the mass of the particle, ω is the angular frequency of the oscillator, $\hbar$ is the reduced Planck's constant and $\hat{P}$ is defined as $-i\hbar\frac{\partial}{\partial\hat{Q}}$. The Hermitian operators $\hat{Q}$ and $\hat{P}$ are the quadrature operators corresponding to the position and momentum of the particle respectively. We define the quadrature eigenstates such that

$$\hat{Q}\left|q\right\rangle = q\left|q\right\rangle, \qquad \hat{P}\left|p\right\rangle = p\left|p\right\rangle, \tag{2.2}$$

where q and p are real continuous variables describing the observables position and momentum of the particle respectively. The operators obey the canonical commutation relation $[\hat{Q}, \hat{P}] = i\hbar$.

When using these Hermitian quadrature operators, we say we are working in the 'quadrature basis' or 'real picture'.

These quadrature operators fully determine the Hamiltonian describing the particle (up to some constants), but there are infinitely many pairs of conjugate operators that can be used instead. For example, the non-Hermitian ladder operators are defined as

$$\hat{a} = \sqrt{\frac{m\omega}{2\hbar}}\left(\hat{Q} + \frac{i}{m\omega}\hat{P}\right), \qquad \hat{a}^\dagger = \sqrt{\frac{m\omega}{2\hbar}}\left(\hat{Q} - \frac{i}{m\omega}\hat{P}\right), \tag{2.3}$$

where $\hat{a}$ is the annihilation operator and $\hat{a}^\dagger$ the creation operator. These operators are more useful when considering their action on Fock states, which are states that label the total number of photons, where $|n\rangle$ denotes n photons in the state. The annihilation and creation operators act on Fock states as

$$\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle\,, \qquad \hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle\,. \tag{2.4}$$

This means the creation operator increases the number of photons by one (creates a photon) and the annihilation operator reduces the photon number by one (destroys or annihilates a photon). A convenient consequence of this is that any Fock state can be written in terms of creation operators acting on the vacuum state (the state with no photons) as

$$|n\rangle = \frac{1}{\sqrt{n!}}(\hat{a}^\dagger)^n\,|0\rangle\,. \tag{2.5}$$

Furthermore, we can introduce a Hermitian operator, the number operator, such that

$$\hat{n}\,|n\rangle := \hat{a}^\dagger\hat{a}\,|n\rangle = n\,|n\rangle\,, \tag{2.6}$$

where the Fock state is its eigenstate and the observable n is the number of photons in the state, a discrete quantity. Unlike the quadrature eigenstates, the number eigenstates form a complete orthonormal basis so that $\langle n|n'\rangle = \delta_{n,n'}$. It follows from the definitions of $\hat{a}$ and $\hat{a}^\dagger$ and that $[\hat{Q},\hat{P}] = i\hbar$ that these operators obey the commutation relation $[\hat{a},\hat{a}^\dagger] = 1$. When using these non-Hermitian ladder operators, we say we are working in the 'ladder basis' or the 'complex picture'. In this basis, the Hamiltonian is given more compactly by

$$\hat{H} = \hbar\omega(\hat{a}^\dagger\hat{a} + \tfrac{1}{2}) = \hbar\omega(\hat{n} + \tfrac{1}{2}). \tag{2.7}$$

To reduce the explicit number of variables in the expressions for the creation and annihilation operators, we can nondimensionalise the quadrature operators by introducing the coordinates $\hat{q} = \sqrt{\frac{m\omega}{\hbar}}\hat{Q}$ and $\hat{p} = \sqrt{\frac{1}{\hbar m\omega}}\hat{P}$. So the ladder operators can be written simply as

$$\hat{a} = \tfrac{1}{\sqrt{2}}(\hat{q} + i\hat{p}), \qquad \hat{a}^\dagger = \tfrac{1}{\sqrt{2}}(\hat{q} - i\hat{p}), \tag{2.8}$$

and the Hamiltonian in terms of the annihilation and creation operators remains unchanged. The Hamiltonian in terms of these new coordinates is given by $\hat{H} = \frac{\hbar\omega}{2}(\hat{p}^2 + \hat{q}^2)$. The commutation rela-

tions of $\hat{q}$ and $\hat{p}$ also simplify to $[\hat{q}, \hat{p}] = i$. We note that there are various ways to nondimensionalise the quadrature operators by moving constants around. There are two other ways commonly used in literature that involve multiplying or dividing each by $\sqrt{2}$. These result in commutation relations $[\hat{q}, \hat{p}] = 2i$ and $[\hat{q}, \hat{p}] = \frac{i}{2}$. By comparing to the $\hat{Q}, \hat{P}$ commutation relation, these three choices would be consistent given $\hbar = 1, 2, \frac{1}{2}$ respectively. Similarly the three Hamiltonians would also be consistent with each other. Therefore, to distinguish which choice has been used it is convention to say $\hbar = 1$, 2 or $\frac{1}{2}$, where we use the nondimensionalised operators with $\hbar = 1$ throughout this thesis. Note that this choice of $\hbar$ does not affect the commutation relations or Hamiltonian in the complex picture.

The above description is for a single mode state. This can be generalised to a state with M modes which is described by a quantum harmonic oscillator with Hamiltonian $\hat{H} = \sum \hat{H}_i$ with $\hat{H}_i$ acting on mode i. The above description can be modified such that all operators and states take a subscript to denote the mode, eg. $\hat{a}_i^\dagger$ is the creation operator on mode i adding a photon in that mode. Operators in different modes commute so the commutation relations can be extended to $[\hat{q}_i, \hat{p}_j] = i\delta_{i,j}$ and $[\hat{a}_i, \hat{a}_j^\dagger] = \delta_{i,j}$ over multiple modes.

A photon is a bosonic quantum particle whose state is described by $|\psi\rangle$, a vector in the hilbert space $\mathcal{H}$. This state could be, for example, wavelength, polarisation, position or momentum. For many-particle systems with N photons, the multi-particle state is denoted with a tensor product between the states of the individual photons $|\psi\rangle = |\psi_1\rangle \otimes \ldots \otimes |\psi_N\rangle$. For convenience, we use the notation $|\psi_1\rangle \otimes \ldots \otimes |\psi_N\rangle \equiv |\psi_1\rangle \ldots |\psi_N\rangle \equiv |\psi_1, \ldots, \psi_N\rangle$. This vector is now from a tensor product of Hilbert spaces. In this representation of states, there is an implicit matching between the ith photon and the ith element in the ket $|\psi_i\rangle$. If the photons are distinguishable in another degree of freedom it is possible to order each photon from the 'first photon' to the 'Nth photon'. However if the photons are indistinguishable in all degrees of freedom (apart from what ψ describes) it is impossible to label them separately. To account for this, we need to symmetrise the state under the exchange of bosons (ie. if we change the ordering of two indistinguishable photons the state should remain unchanged). In order to do this, the state must be considered as a superposition of all the possible orderings of the photons consistent with the overall state: $|\psi\rangle = \mathcal{N} \sum_{\sigma \in S_N} |\psi_{\sigma(1)}, \psi_{\sigma(2)}, ..., \psi_{\sigma(N)}\rangle$, where S_N is the symmetric group which contains all permutations of elements 1 to N, and $\mathcal{N}$ is some normalisation constant. This representation of quantum states involving a tensor product of states for each particle is called the first quantisation formalism.

However, there is another representation that naturally takes into account the permutations of the indistinguishable photons. Given the labelling of photons to a particular position is not important, we only need to keep a count of how many photons are in each state. These are the Fock states already introduced, but we can expand beyond one mode. Instead of defining a basis with M vectors to describe the state of one photon, it is possible to write the state as a single vector with M elements. For example, if we are considering the state to be polarisation, the basis states could be $|\uparrow\rangle$ and $|\rightarrow\rangle$ which can be equivalently written as $|\uparrow\rangle \equiv |1, 0\rangle$ and $|\rightarrow\rangle \equiv |0, 1\rangle$. Here the first element in the ket corresponds to vertical polarisation and the second element to horizontal polarisation, and the number indicates the number of photons in that state. For the

case of only one photon the representations are completely equivalent. However when there is more than one photon, the basis with M elements must be symmetrised as explained above whereas the Fock state with N photons is still given by a vector with M elements where the sum of the elements is N. Again looking at the polarisation as an example, a state with one photon in each polarisation can be represented as $|\psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow,\rightarrow\rangle + |\rightarrow,\uparrow\rangle)$ in the first quantisation or more succinctly as the Fock state $|\psi\rangle = |1,1\rangle$. More generally for M basis states the Fock state is given by $|\psi\rangle = |n_1,\ldots,n_M\rangle$, where n_i is the number of photons in mode i. The term 'mode' can refer to any physical encoding of the basis states, for example, for an arbitrary number of modes, spatial modes (eg. optical fibres or waveguides on an integrated chip), or time-bins (photons are delayed with respect to each other), or for just two modes, polarisation. In the case where the total number of photons is not fixed, the space that the vectors exist in is expanded to a direct sum of Hilbert spaces, $\mathcal{H}^{(0)},\mathcal{H}^{(1)},\mathcal{H}^{(2)},\ldots$, where $\mathcal{H}^{(i)}$ is the Hilbert space for states with i photons. This is the Fock space and this representation is known as the second quantisation formalism.

So far we have used state vectors to describe a quantum system. These can be used to describe a pure basis state such as a photon being in the first mode $|1,0,0\rangle$, and also a pure state in a superposition described by a normalised linear combination of vectors for example $\frac{1}{\sqrt{2}}(|1,0,0\rangle + |0,1,0\rangle)$. However, it is possible to create a mixed state - a state prepared with some probability of being in each of a selection of pure states. These cannot be described by vectors alone and so we introduce density operators [29] as a more general way to describe a state. This density operator contains all the information about a state. For a pure state the density operator is given by $\hat{\rho} = |\psi\rangle\langle\psi|$, whereas for a mixed state which is prepared in state ψ_i with probability p_i the density operator is given by $\hat{\rho} = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. The purity of a state is defined as $\mathcal{P} := \mathrm{Tr}(\hat{\rho}^2) = \sum_i p_i^2$. For a pure state, there is only one term with $p = 1$ so the purity is equal to 1, whereas mixed states have more than one state and so all $p_i < 1$ and hence the purity $\mathcal{P} < 1$.

When measuring a state $\hat{\rho}$, the type of measurement performed corresponds to a basis of projection operators $\hat{\Pi}_j$ such that $\sum_j \hat{\Pi}_j = \hat{I}$, the identity operator. Each operator is associated with projecting the state onto state $|\pi_j\rangle$ and measuring the value j. The measurement projects the state onto the basis state $|\pi_j\rangle$ with probability $\mathrm{Tr}(\hat{\rho}\hat{\Pi}_j)$. For pure states where $\hat{\rho} = |\psi\rangle\langle\psi|$ measuring with the projective operator with basis $\hat{\Pi}_j = |\pi_j\rangle\langle\pi_j|$, the probability can be written as

$$\Pr(j|\psi_{\mathrm{pure}}) = \mathrm{Tr}(|\psi\rangle\langle\psi|\pi_j\rangle\langle\pi_j|) = \langle\psi|\pi_j\rangle\langle\pi_j|\psi\rangle = |\langle\pi_j|\psi\rangle|^2. \tag{2.9}$$

For mixed states,

$$\Pr(j|\psi_{\mathrm{mixed}}) = \mathrm{Tr}\left(\sum_i p_i |\psi_i\rangle\langle\psi_i|\pi_j\rangle\langle\pi_j|\right) = \sum_i p_i \langle\psi_i|\pi_j\rangle\langle\pi_j|\psi_i\rangle = \sum_i p_i |\langle\pi_j|\psi_i\rangle|^2, \tag{2.10}$$

which is the weighted sum of the individual probabilities of each pure state. As an example that will be used throughout this thesis, photon number resolving detectors measure the number of photons in a mode and project the state onto a Fock basis state $|n\rangle$. So the projection operator basis here is $|n\rangle\langle n|$ for all n which project onto state $|n\rangle$ measuring n photons with probability

$\mathrm{Tr}(\hat{\rho}\,|n\rangle\,\langle n|) = |\,\langle n|\psi\rangle\,|^2$ in the case of a pure state.

In the case with multiple degrees of freedom, it is sometimes useful to know properties of one of the subsystems without knowledge of the whole system. The reduced density operator of subsystem A describes the state of the subsystem A. This is found by taking the partial trace of the density operator of the whole system over the other subsystem B:

$$\hat{\rho}_A = \mathrm{Tr}_B(\hat{\rho}_{A,B}). \tag{2.11}$$

Two subsystems can be entangled or separable. A system is separable if its state can be written as a product of states in each subsystem: $|\psi_{A,B}\rangle = |\psi_A\rangle\,|\psi_B\rangle$. For example the state

$$|\psi\rangle = \frac{1}{2}(|0_A, 0_B\rangle + |2_A, 0_B\rangle + |0_A, 2_B\rangle + |2_A, 2_B\rangle) = \frac{1}{\sqrt{2}}(|0_A\rangle + |2_A\rangle) \otimes \frac{1}{\sqrt{2}}(|0_B\rangle + |2_B\rangle) \tag{2.12}$$

which is written in the Fock basis for two modes with two photons is separable. Taking the partial trace over subsystem B leaves subsystem A with the pure state density operator $\hat{\rho}_A = \mathrm{Tr}_B(\hat{\rho}) = \frac{1}{2}(|0_A\rangle + |2_A\rangle)(\langle 0_A| + \langle 2_A|)$ and therefore the state in subsystem A is independent of the subsystem B. If subsystem B is measured to have 2 photons, subsystem A is still in the state $\frac{1}{\sqrt{2}}(|0_A\rangle + |2_A\rangle)$, the same state as if B had measured no photons. In contrast,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|2_A, 0_B\rangle + |0_A, 2_B\rangle) \tag{2.13}$$

is an entangled state as the subsystems are not independent. This can be seen from the state vector because if system B is measured to have 2 photons, system A must have no photons and vice versa. More formally, taking the partial trace over subsystem B leaves subsystem A with the mixed state density operator $\hat{\rho}_A = \mathrm{Tr}_B(\hat{\rho}) = \frac{1}{2}(|0_A\rangle\,\langle 0_A| + |2_A\rangle\,\langle 2_A|)$. The condition for a separable state is that the reduced density operator when tracing over one subsystem will be pure whereas an entangled state will result in a mixed reduced density operator.

2.2 Phase-space formalism

In the previous section we introduced multimode states represented by vectors with M elements and quadrature and ladder operators that act on each mode separately. Working in terms of the quadrature operators is known as the phase-space formalism [30] and we explore here how to represent states and transformations in this formalism as well as the mapping to the equivalent ladder operators picture.

2.2.1 Describing states

When considering an M-mode continuous variable system, it is convenient to write the quadrature operators in a vector with $2M$ elements [31]. The $\hat{q}$ and $\hat{p}$ operators for all M modes can be ordered in any way, and there are two common conventions in literature. The first is $\hat{\boldsymbol{x}}^{(A)} =$

$(\hat{q}_1, \hat{p}_1, \hat{q}_2, \hat{p}_2, ..., \hat{q}_M, \hat{p}_M)^\top$ and the second is $\hat{\boldsymbol{x}}^{(B)} = (\hat{q}_1, \hat{q}_2, ..., \hat{q}_M, \hat{p}_1, \hat{p}_2, ..., \hat{p}_M)^\top = (\begin{smallmatrix}\hat{\boldsymbol{q}} \\ \hat{\boldsymbol{p}}\end{smallmatrix})$, where the labels A and B are used to distinguish between the conventions. In the first convention, we can succinctly write the commutation relations as $[\hat{x}_i^{(A)}, \hat{x}_j^{(A)}] = i\Omega_{i,j}^{(A)}$, where

$$\boldsymbol{\Omega}^{(A)} = \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{\omega}} \end{pmatrix}, \qquad \text{where} \quad \tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{2.14}$$

Similarly, in the second convention $[\hat{x}_i^{(B)}, \hat{x}_j^{(B)}] = i\Omega_{i,j}^{(B)}$, where now

$$\boldsymbol{\Omega}^{(B)} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{pmatrix}. \tag{2.15}$$

We can equivalently work in the complex basis defining a vector of creation and annihilation operators $\hat{\boldsymbol{\zeta}} = (\hat{a}_1, \ldots, \hat{a}_M, \hat{a}_1^\dagger, \ldots, \hat{a}_M^\dagger)^\top = (\hat{\boldsymbol{a}}^\top, \hat{\boldsymbol{a}}^\dagger)^\top$ where we introduce the *row* vector $\hat{\boldsymbol{a}}^\dagger$ for the creation operators but a *column* vector $\hat{\boldsymbol{a}}$ for the annihilation operators so that they are the Hermitian conjugate of each other. Again the operators could be in any order but this is the usual convention. From eq. (2.8) we can convert between the quadrature and complex bases by the transformation $\hat{\boldsymbol{\zeta}} = \boldsymbol{F}^{(A)}\hat{\boldsymbol{x}}^{(A)}$ or $\hat{\boldsymbol{\zeta}} = \boldsymbol{F}^{(B)}\hat{\boldsymbol{x}}^{(B)}$, where

$$\boldsymbol{F}^{(A)} = \begin{pmatrix} \boldsymbol{f} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{f} \end{pmatrix}, \qquad \text{where} \quad \boldsymbol{f} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \tag{2.16}$$

and

$$\boldsymbol{F}^{(B)} = \frac{1}{\sqrt{2}} \begin{pmatrix} \boldsymbol{I} & i\boldsymbol{I} \\ \boldsymbol{I} & -i\boldsymbol{I} \end{pmatrix}. \tag{2.17}$$

We find the second convention to be more convenient and use that one throughout the rest of this thesis, dropping the B labelling.

The commutation relations of the creation and annihilation operators can be written in a similar manner to the quadrature operators where the commutation matrix $\boldsymbol{\Omega}^{(c)}$ is now given in the ladder basis. The commutation relations are given by

$$\left[\hat{\zeta}_i, \hat{\zeta}_j^\dagger\right] = \left[\sum_k F_{i,k}\hat{x}_k, \sum_l \hat{x}_l F_{l,j}^\dagger\right] = \sum_k \sum_l F_{i,k}F_{l,j}^\dagger \left[\hat{x}_k, \hat{x}_l\right] = \sum_k \sum_l F_{i,k}F_{l,j}^\dagger i\Omega_{k,l} = i(F\Omega F^\dagger)_{i,j}. \tag{2.18}$$

Hence, we can define the commutation matrix in the number basis as

$$\boldsymbol{\Omega}^{(c)} := \boldsymbol{F}\boldsymbol{\Omega}\boldsymbol{F}^\dagger = -i \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{pmatrix}, \tag{2.19}$$

which gives commutation relations $\left[\hat{\zeta}_i, \hat{\zeta}_j^\dagger\right] = i\Omega_{i,j}^{(c)}$, and the superscript c specifies that it is for the

complex basis.

Now we have a convenient matrix and vector notation for multimode states, we consider how to fully describe these states. As mentioned previously, a pure state is described by a state vector but a mixed state needs a density operator to fully describe it. There are equivalent ways to fully characterise a state, eg. the Wigner function [32], the Q-function [33] and the P-function [34, 35]. Any state can be written in terms of basis states where the probability of measuring a state in one of these basis states with projection operator $\hat{P}$ is given by $\mathrm{Tr}(\hat{\rho}\hat{P})$. One such basis is represented by the Weyl operator $\hat{D}(\boldsymbol{x}) = \exp(i\hat{\boldsymbol{x}}^\top \boldsymbol{\Omega}\boldsymbol{x})$. An S-parametrised characteristic function is given by [30]

$$\chi(\boldsymbol{x}, S) = \mathrm{Tr}(\hat{\rho}\hat{D}(\boldsymbol{x})) \exp\left(\frac{S\boldsymbol{x}^\top \boldsymbol{x}}{4}\right). \tag{2.20}$$

The values of $S = 0, -1, +1$ correspond to the Wigner, Husimi Q and Glauber-Sudarshan P characterisation functions respectively. They are more often expressed in their complex forms as

$$\chi(\boldsymbol{\zeta}, S) = \mathrm{Tr}(\hat{\rho}\hat{D}(\boldsymbol{\zeta})) \exp\left(\frac{S\boldsymbol{\zeta}^\top \boldsymbol{\zeta}}{4}\right). \tag{2.21}$$

Equivalently a state can be described by its Wigner, Q or P functions, which are the Fourier transforms of the corresponding *characteristic* functions

$$F(\boldsymbol{x}, S) = \int_{\mathcal{R}^{2N}} \frac{d^{2N}\boldsymbol{y}}{(2\pi)^{2N}} \exp(-i\boldsymbol{x}^\top \boldsymbol{\Omega}\boldsymbol{y})\chi(\boldsymbol{y}, S), \tag{2.22}$$

and similarly in the complex form, where $W(\boldsymbol{x}) = F(\boldsymbol{x}, 0)$ is the Wigner function, $P(\boldsymbol{x}) = F(\boldsymbol{x}, 1)$ is the P function and $Q(\boldsymbol{x}) = F(\boldsymbol{x}, -1)$ is the Q function. The Wigner, Q and P functions are popular choices because they correspond to operators that are symmetrised over creation/annihilation operators, in antinormal order or ordered normally respectively. They can be used to calculate the expectation values of these types of operators

$$\langle \hat{O}^{(\mathrm{n})} \rangle = \int P(\boldsymbol{\zeta})\, \hat{O}^{(\mathrm{n})}\, d^{2N}\boldsymbol{\zeta}, \qquad \langle \hat{O}^{(\mathrm{a})} \rangle = \int Q(\boldsymbol{\zeta})\, \hat{O}^{(\mathrm{a})}\, d^{2N}\boldsymbol{\zeta}, \qquad \langle \hat{O}^{(\mathrm{s})} \rangle = \int W(\boldsymbol{\zeta})\, \hat{O}^{(\mathrm{s})}\, d^{2N}\boldsymbol{\zeta}, \tag{2.23}$$

where the superscripts (a), (n) and (s) denote antinormal, normal and symmetric ordered respectively. These functions are characterised by the statistical moments of the quantum state. For the Wigner function which is symmetrically ordered, the first moment (the vector of means), and the second moment (the covariance matrix) [36] are given by

$$\bar{\boldsymbol{x}} = \langle \hat{\boldsymbol{x}} \rangle = \mathrm{Tr}(\hat{\boldsymbol{x}}\hat{\rho}), \qquad V_{i,j} = \tfrac{1}{2}\langle \{\hat{x}_i - \bar{x}_i, \hat{x}_j - \bar{x}_j\} \rangle. \tag{2.24}$$

Due to the commutation relations which can be used to change the order of the annihilation and creation operators, the covariance matrix corresponding to the Q and P functions are given by [37]

$$\boldsymbol{V}_Q = \boldsymbol{V} + \tfrac{1}{2}\boldsymbol{I}, \qquad \boldsymbol{V}_P = \boldsymbol{V} - \tfrac{1}{2}\boldsymbol{I}, \tag{2.25}$$

Object	Real	Complex	Mapping
operators	$\hat{\boldsymbol{x}} = \begin{pmatrix} \hat{\boldsymbol{q}} \\ \hat{\boldsymbol{p}} \end{pmatrix}$	$\hat{\boldsymbol{\zeta}} = \begin{pmatrix} \hat{\boldsymbol{a}} \\ (\hat{\boldsymbol{a}}^\dagger)^\top \end{pmatrix}^\top$	$\hat{\boldsymbol{\zeta}} = \boldsymbol{F}\hat{\boldsymbol{x}}$
displacement	$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{pmatrix}$	$\boldsymbol{\zeta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{pmatrix}$	$\boldsymbol{\zeta} = \boldsymbol{F}\boldsymbol{x}$
vector of means	$\bar{\boldsymbol{x}} = \langle \hat{\boldsymbol{x}} \rangle = \begin{pmatrix} \bar{\boldsymbol{q}} \\ \bar{\boldsymbol{p}} \end{pmatrix}$	$\bar{\boldsymbol{\zeta}} = \langle \hat{\boldsymbol{\zeta}} \rangle = \begin{pmatrix} \bar{\boldsymbol{\alpha}} \\ \bar{\boldsymbol{\alpha}}^* \end{pmatrix}$	$\bar{\boldsymbol{\zeta}} = \boldsymbol{F}\bar{\boldsymbol{x}}$
covariance matrix	$\boldsymbol{V}$	$\boldsymbol{\Sigma}$	$\boldsymbol{\Sigma} = \boldsymbol{F}\boldsymbol{V}\boldsymbol{F}^\dagger$
commutation relations	$[\hat{x}_i, \hat{x}_j] = \Omega_{i,j}$	$[\hat{\zeta}_i, \hat{\zeta}_j] = \Omega^{(c)}_{i,j}$	$\boldsymbol{\Omega}^{(c)} = \boldsymbol{F}\boldsymbol{\Omega}\boldsymbol{F}^\dagger$
eigenstates	$\hat{x}_i \lvert x_i \rangle = x_i \lvert x_i \rangle$	$\hat{a}_i \lvert \alpha_i \rangle = \alpha_i \lvert \alpha_i \rangle$	N/A
transformation	$\boldsymbol{V}_{\text{vac}} \to \boldsymbol{S}\boldsymbol{V}_{\text{vac}}\boldsymbol{S}^\top$	$\boldsymbol{\Sigma}_{\text{vac}} \to \boldsymbol{M}\boldsymbol{\Sigma}_{\text{vac}}\boldsymbol{M}^\dagger$	$\boldsymbol{M} = \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^\dagger$

Table 2.1: A summary of the mapping between quadrature and ladder descriptions and the notation used throughout this thesis. The matrix $\boldsymbol{F}$ defined in eq. (2.17) can be used to convert between the real and complex descriptions of multimode states in the phase-space formalism. The mapping of the operators, first and second moments and commutation relations are shown with the notation used in this thesis.

while the first moment, which consists only of one creation or annihilation operator and therefore is normally, antinormally and symmetrically ordered, remains unchanged.

The moments above are written in the real quadrature basis, but can also be written in the complex number basis. To swap between the two descriptions, the conversion matrix $\boldsymbol{F}$ is used as follows:

$$\bar{\boldsymbol{\zeta}} = \begin{pmatrix} \bar{\boldsymbol{\alpha}} \\ \bar{\boldsymbol{\alpha}}^* \end{pmatrix} = \boldsymbol{F}\bar{\boldsymbol{x}}, \qquad \boldsymbol{\Sigma} = \boldsymbol{F}\boldsymbol{V}\boldsymbol{F}^\dagger, \tag{2.26}$$

where $\bar{\alpha}_i = (\bar{x}_i + i\bar{p}_i)/\sqrt{2}$. There is a special class of states which are fully defined by only these first two moments, ie. $\hat{\rho} = \hat{\rho}(\bar{\boldsymbol{x}}, \boldsymbol{V})$. These are the Gaussian states [38, 39] which must have Gaussian Wigner characteristic and Wigner functions:

$$\chi(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\boldsymbol{x}^\top(\boldsymbol{\Omega}\boldsymbol{V}\boldsymbol{\Omega}^\top)\boldsymbol{x} - i\bar{\boldsymbol{x}}^\top\boldsymbol{\Omega}^\top\boldsymbol{x}\right), \qquad W(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^\top\boldsymbol{V}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}}))}{\sqrt{\det(2\pi\boldsymbol{V})}}. \tag{2.27}$$

2.2.2 Gaussian transformations

A quantum channel is a transformation on a quantum state that is completely positive and trace-preserving of the density operator. In this thesis we restrict to the case of reversible transformations, given by unitary operators $\hat{\mathcal{U}}$ which act on the density operator as $\hat{\rho} \to \hat{\mathcal{U}}\hat{\rho}\hat{\mathcal{U}}^\dagger$. Gaussian transformations are those that take Gaussian states to Gaussian states. A general Gaussian unitary, $\hat{\mathcal{U}} = \exp\left(-i\hat{H}\right)$, is generated by a Hamiltonian which is at most second-order in the

creation/annihilation operators. Explicitly this Hamiltonian is given by

$$\hat{H} = i(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} + \hat{\boldsymbol{a}}^\dagger \boldsymbol{G} \hat{\boldsymbol{a}} + \hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top) + \text{h.c.} \tag{2.28}$$

where as before $\hat{\boldsymbol{a}}^\dagger = (\hat{a}_1^\dagger, \hat{a}_2^\dagger, ..., \hat{a}_M^\dagger)$ and $\hat{\boldsymbol{a}} = (\hat{a}_1, \hat{a}_2, .., \hat{a}_M)^\top$, h.c. stands for Hermitian conjugate, $\boldsymbol{G}$ and $\boldsymbol{H}$ are complex matrices and $\boldsymbol{\alpha}$ is a complex column vector.

A state can be described by its statistical moments such as the vector of means $\bar{\boldsymbol{x}}$ or $\bar{\boldsymbol{\zeta}}$ and covariance matrix $\boldsymbol{V}$ or $\boldsymbol{\Sigma}$ as defined previously. Therefore a transformation can be equivalently defined by its effect on these moments. Transforming these moments can be done by instead considering its action on the quadrature or ladder operators:

$$\begin{aligned}
\bar{x}_i &= \langle \hat{x}_i \rangle = \text{Tr}(\hat{\rho} \hat{x}_i) \to \text{Tr}(\hat{\mathcal{U}} \hat{\rho} \hat{\mathcal{U}}^\dagger \hat{x}_i) = \text{Tr}(\hat{\rho} \hat{\mathcal{U}}^\dagger \hat{x}_i \hat{\mathcal{U}}) \\
V_{i,j} &= \langle \hat{x}_i \hat{x}_j \rangle = \text{Tr}\left(\hat{\rho} \tfrac{1}{2} (\hat{x}_i \hat{x}_j + \hat{x}_j \hat{x}_i - 2\hat{x}_i \bar{x}_j - 2\hat{x}_j \bar{x}_i + 2\bar{x}_i \bar{x}_j)\right) \\
&\to \text{Tr}\left(\hat{\mathcal{U}} \hat{\rho} \hat{\mathcal{U}}^\dagger \tfrac{1}{2} (\hat{x}_i \hat{x}_j + \hat{x}_j \hat{x}_i - 2\hat{x}_i \bar{x}_j - 2\hat{x}_j \bar{x}_i + 2\bar{x}_i \bar{x}_j)\right) \\
&= \text{Tr}\left(\hat{\rho} \hat{\mathcal{U}}^\dagger \tfrac{1}{2} (\hat{x}_i \hat{x}_j + \hat{x}_j \hat{x}_i - 2\hat{x}_i \bar{x}_j - 2\hat{x}_j \bar{x}_i + 2\bar{x}_i \bar{x}_j) \hat{\mathcal{U}}\right).
\end{aligned} \tag{2.29}$$

The vector of means has an explicit dependence on $\hat{\mathcal{U}}^\dagger \hat{x}_i \hat{\mathcal{U}}$ and with inserting the identity $\hat{\boldsymbol{I}} = \hat{\mathcal{U}} \hat{\mathcal{U}}^\dagger$ between the quadrature operators in the covariance matrix, it can be seen that the effect on the covariance matrix can also be accounted for completely by the action on the quadrature operators. When considering the unitary transformations acting on the operators, either the ladder or the quadrature operators, we are working in the Heisenberg picture.

The transformation on the annihilation and creation operators is given by the Bogoliubov transformation [40]

$$\hat{\boldsymbol{a}} \to \boldsymbol{C} \hat{\boldsymbol{a}} + \boldsymbol{D} (\hat{\boldsymbol{a}}^\dagger)^\top + \boldsymbol{\alpha}, \qquad (\hat{\boldsymbol{a}}^\dagger)^\top \to \boldsymbol{C}^* (\hat{\boldsymbol{a}}^\dagger)^\top + \boldsymbol{D}^* \hat{\boldsymbol{a}} + \boldsymbol{\alpha}^*. \tag{2.30}$$

It is convenient to write this in matrix form as:

$$\begin{pmatrix} \hat{\boldsymbol{a}} \\ (\hat{\boldsymbol{a}}^\dagger)^\top \end{pmatrix} \to \begin{pmatrix} \hat{\mathcal{U}}^\dagger \hat{\boldsymbol{a}} \hat{\mathcal{U}} \\ \hat{\mathcal{U}}^\dagger (\hat{\boldsymbol{a}}^\dagger)^\top \hat{\mathcal{U}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{D}^* & \boldsymbol{C}^* \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{a}} \\ (\hat{\boldsymbol{a}}^\dagger)^\top \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{pmatrix}. \tag{2.31}$$

In more succinct notation, this is written as $\hat{\boldsymbol{\zeta}} \to \boldsymbol{M} \hat{\boldsymbol{\zeta}} + \boldsymbol{\zeta}$. The Bogoliubov transformation is canonical and so must preserve the canonical commutation relations $[\hat{a}_i, \hat{a}_j^\dagger] = \delta_{i,j}$ and $[\hat{a}_i, \hat{a}_j] = 0$. By fixing $[\hat{\mathcal{U}}^\dagger \hat{a}_i \hat{\mathcal{U}}, \hat{\mathcal{U}}^\dagger \hat{a}_j^\dagger \hat{\mathcal{U}}] = \delta_{i,j}$ and $[\hat{\mathcal{U}}^\dagger \hat{a}_i \hat{\mathcal{U}}, \hat{\mathcal{U}}^\dagger \hat{a}_j \hat{\mathcal{U}}] = 0$ and substituting eq. (2.31) into the commutation relations for $\hat{\boldsymbol{a}}^\dagger$ and $\hat{\boldsymbol{a}}$, the following conditions must be placed on the matrix corresponding to the unitary operator:

$$\boldsymbol{C}\boldsymbol{C}^\dagger - \boldsymbol{D}\boldsymbol{D}^\dagger = \boldsymbol{I}, \qquad \boldsymbol{C}\boldsymbol{D}^\top - \boldsymbol{D}\boldsymbol{C}^\top = \boldsymbol{0}. \tag{2.32}$$

We can convert these matrix conditions to the real quadrature coordinates:

$$\hat{\boldsymbol{x}} = \boldsymbol{F}^\dagger \hat{\boldsymbol{\zeta}} \to \boldsymbol{F}^\dagger \boldsymbol{M} \boldsymbol{F} \hat{\boldsymbol{x}} + \boldsymbol{F}^\dagger \boldsymbol{\zeta} = \boldsymbol{S}\hat{\boldsymbol{x}} + \boldsymbol{d}, \tag{2.33}$$

where $\boldsymbol{S} = \boldsymbol{F}^\dagger \boldsymbol{M} \boldsymbol{F}$ and $\boldsymbol{d} = \boldsymbol{F}^\dagger \boldsymbol{\zeta}$. This matrix $\boldsymbol{S}$ is symplectic and therefore by definition satisfies $\boldsymbol{S}\boldsymbol{\Omega}\boldsymbol{S}^\top = \boldsymbol{\Omega}$ [41]. Note that the conditions for matrix $\boldsymbol{M}$ in the complex coordinates are similar to the conditions for symplectic matrices and in fact the matrix $\boldsymbol{M}$ is isomorphic to symplectic matrices. This matrix is part of the Lie group $\mathrm{U}(n,n) = \{\boldsymbol{M} | \boldsymbol{M}\boldsymbol{\Omega}^{(c)}\boldsymbol{M}^\dagger = \boldsymbol{\Omega}^{(c)}\}$ [42].

The Hamiltonian in eq. (2.28) can be considered to be composed of three separate transformations applied in order. First we consider the term $i(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})$. We consider the action of this operator $\hat{D}(\boldsymbol{\alpha}) = \exp(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})$ on the creation/annihilation operators using the Baker-Hausdorff lemma [27]:

$$\hat{a}_i \to \hat{D}^\dagger \hat{a}_i \hat{D} = \mathrm{e}^{-(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})}\, \hat{a}_i\, \mathrm{e}^{(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})} = \hat{a}_i + \alpha_i \tag{2.34}$$

$$\hat{a}_i^\dagger \to \hat{D}^\dagger \hat{a}_i^\dagger \hat{D} = \mathrm{e}^{-(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})}\, \hat{a}_i^\dagger\, \mathrm{e}^{(\hat{\boldsymbol{a}}^\dagger \boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger \hat{\boldsymbol{a}})} = \hat{a}_i^\dagger + \alpha_i^*. \tag{2.35}$$

This does not mix the creation or annihilation operators and simply performs single-mode operations, displacing mode i by α_i in the phase space. The operator $\hat{D}(\boldsymbol{\alpha})$ is the called the **displacement operator** and is just the complex version of the Weyl operator introduced earlier. From the transformations, it is clear that the transformation matrix $\boldsymbol{M}_d$ and vector $\boldsymbol{\zeta}_d$ in the complex picture are given by

$$\boldsymbol{M}_d = \boldsymbol{I}, \qquad \boldsymbol{\zeta}_d = \boldsymbol{\zeta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{pmatrix}, \tag{2.36}$$

where $\boldsymbol{\zeta}$ is the displacement vector of the state and hence why the notation $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ in the operator was chosen. Similarly in the real picture, they are

$$\boldsymbol{S}_d = \boldsymbol{I}, \qquad \boldsymbol{d}_d = \boldsymbol{x} = \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{pmatrix}, \tag{2.37}$$

where $\boldsymbol{q}$ and $\boldsymbol{p}$ are the position and momentum eigenvalues respectively.

Next we look at the term $i(\hat{\boldsymbol{a}}^\dagger \boldsymbol{G} \hat{\boldsymbol{a}} - \hat{\boldsymbol{a}}^\dagger \boldsymbol{G}^\dagger \hat{\boldsymbol{a}}) = \hat{\boldsymbol{a}}^\dagger i\boldsymbol{G} \hat{\boldsymbol{a}} + \hat{\boldsymbol{a}}^\dagger (i\boldsymbol{G})^\dagger \hat{\boldsymbol{a}} = \hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}}$, where $\boldsymbol{E} = i\boldsymbol{G} + (i\boldsymbol{G})^\dagger$ is a Hermitian matrix. The operator $\hat{\mathcal{U}} = \exp\bigl(-i\hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}}\bigr)$ has the following action on the creation and annihilation operators:

$$\hat{a}_i^\dagger \to \hat{\mathcal{U}}^\dagger \hat{a}_i^\dagger \hat{\mathcal{U}} = \mathrm{e}^{(i\hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}})}\, \hat{a}_i^\dagger\, \mathrm{e}^{(-i\hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}})} = (\hat{\boldsymbol{a}}^\dagger \mathrm{e}^{(i\boldsymbol{E})})_i = (\hat{\boldsymbol{a}}^\dagger \boldsymbol{U}^\dagger)_i \tag{2.38}$$

$$\hat{a}_i \to \hat{\mathcal{U}}^\dagger \hat{a}_i \hat{\mathcal{U}} = \mathrm{e}^{(i\hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}})}\, \hat{a}_i\, \mathrm{e}^{(-i\hat{\boldsymbol{a}}^\dagger \boldsymbol{E} \hat{\boldsymbol{a}})} = (\mathrm{e}^{(-i\boldsymbol{E})} \hat{\boldsymbol{a}})_i = (\boldsymbol{U}\hat{\boldsymbol{a}})_i \tag{2.39}$$

where we define $\boldsymbol{U} = \mathrm{e}^{-iE}$. This operator mixes within the creation operators and the annihilation operators but not between them, corresponding to a multimode unitary transformation $\boldsymbol{U}$ in linear optics. It is useful to find the transformation of the column vectors $\hat{\boldsymbol{a}} \to \boldsymbol{U}\hat{\boldsymbol{a}}$ and $(\hat{\boldsymbol{a}}^\dagger)^\top \to \boldsymbol{U}^*(\hat{\boldsymbol{a}}^\dagger)^\top$.

From this and eq. (2.31), it is simple to find the transformation matrix and vector as

$$\boldsymbol{M}_u = \begin{pmatrix} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}^* \end{pmatrix} \qquad \boldsymbol{\zeta}_u = \boldsymbol{0}, \tag{2.40}$$

and in the real picture, using the mapping $\boldsymbol{S} = \boldsymbol{F}^\dagger \boldsymbol{M} \boldsymbol{F}$, the transformation is given by

$$\boldsymbol{S}_u = \begin{pmatrix} \mathrm{Re}(\boldsymbol{U}) & -\mathrm{Im}(\boldsymbol{U}) \\ \mathrm{Im}(\boldsymbol{U}) & \mathrm{Re}(\boldsymbol{U}) \end{pmatrix} \qquad \boldsymbol{d}_u = \boldsymbol{0}, \tag{2.41}$$

where $\mathrm{Re}(\boldsymbol{U})$ and $\mathrm{Im}(\boldsymbol{U})$ denote the real and imaginary parts of the matrix $\boldsymbol{U}$.

And finally we look at the last term $i(\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top - \hat{\boldsymbol{a}}^\top \boldsymbol{H}^\dagger \hat{\boldsymbol{a}})$. This is just a number and so its transpose must leave it unchanged. From this we can determine that $\boldsymbol{H}$ is symmetric. So the action of this operator $\hat{\mathcal{U}}_s = \exp\big(\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top - \hat{\boldsymbol{a}}^\top \boldsymbol{H}^* \hat{\boldsymbol{a}}\big)$ on the annihilation and creation operators is

$$\begin{aligned} \hat{a}_i^\dagger \to \hat{\mathcal{U}}_s^\dagger \hat{a}_i^\dagger \hat{\mathcal{U}}_s &= \mathrm{e}^{(-\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top + \hat{\boldsymbol{a}}^\top \boldsymbol{H}^* \hat{\boldsymbol{a}})}\, \hat{a}_i^\dagger\, \mathrm{e}^{(\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top - \hat{\boldsymbol{a}}^\top \boldsymbol{H}^* \hat{\boldsymbol{a}})} \\ &= \left[\hat{\boldsymbol{a}}^\dagger \cosh\left(2\sqrt{\boldsymbol{H}^* \boldsymbol{H}}\right) + \sinh\left(2\sqrt{\boldsymbol{H}^* \boldsymbol{H}}\right) \frac{\boldsymbol{H}^*}{\sqrt{\boldsymbol{H}^* \boldsymbol{H}}} \hat{\boldsymbol{a}} \right]_i \end{aligned} \tag{2.42}$$

$$\begin{aligned} \hat{a}_i \to \hat{\mathcal{U}}_s^\dagger \hat{a}_i \hat{\mathcal{U}}_s &= \mathrm{e}^{(-\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top + \hat{\boldsymbol{a}}^\top \boldsymbol{H}^* \hat{\boldsymbol{a}})}\, \hat{a}_i\, \mathrm{e}^{(\hat{\boldsymbol{a}}^\dagger \boldsymbol{H} (\hat{\boldsymbol{a}}^\dagger)^\top - \hat{\boldsymbol{a}}^\top \boldsymbol{H}^* \hat{\boldsymbol{a}})} \\ &= \left[\cosh\left(2\sqrt{\boldsymbol{H}^* \boldsymbol{H}}\right) \hat{\boldsymbol{a}} + \hat{\boldsymbol{a}}^\dagger \frac{\boldsymbol{H}}{\sqrt{\boldsymbol{H}^* \boldsymbol{H}}} \sinh\left(2\sqrt{\boldsymbol{H}^* \boldsymbol{H}}\right) \right]_i . \end{aligned} \tag{2.43}$$

If we restrict to the case where $\boldsymbol{H}$ is diagonal such that $2H_{i,i} = -\xi_i$, this is the case of single mode squeezing with the operator given by $\exp\left(\frac{1}{2}(-\xi_i \hat{a}_i^\dagger \hat{a}_i^\dagger + \xi_i^* \hat{a}_i \hat{a}_i)\right)$. In this case we can simplify the transformations to

$$\hat{a}_i^\dagger \to \hat{S}^\dagger \hat{a}_i^\dagger \hat{S} = \mathrm{e}^{(\hat{a}_i^\dagger \xi_i \hat{a}_i^\dagger - \hat{a}_i \xi_i^* \hat{a}_i)/2}\, \hat{a}_i^\dagger\, \mathrm{e}^{(-\hat{a}_i^\dagger \xi_i \hat{a}_i^\dagger + \hat{a}_i \xi_i^* \hat{a}_i)/2} = \hat{a}_i^\dagger \cosh(|\xi_i|) - \hat{a}_i \frac{\xi_i^*}{|\xi_i|} \sinh(|\xi_i|) \tag{2.44}$$

$$\hat{a}_i \to \hat{S}^\dagger \hat{a}_i \hat{S} = \mathrm{e}^{(\hat{a}_i^\dagger \xi_i \hat{a}_i^\dagger - \hat{a}_i \xi_i^* \hat{a}_i)/2}\, \hat{a}_i\, \mathrm{e}^{(-\hat{a}_i^\dagger \xi_i \hat{a}_i^\dagger + \hat{a}_i \xi_i^* \hat{a}_i)/2} = \hat{a}_i \cosh(|\xi_i|) - \hat{a}_i^\dagger \frac{\xi_i}{|\xi_i|} \sinh(|\xi_i|). \tag{2.45}$$

By using the notation $\xi = r\mathrm{e}^{i\phi}$, the transformation matrices and vectors for this single mode squeezing (on all modes) operation can be written as

$$\boldsymbol{M}_s = \begin{pmatrix} \oplus_i \cosh r_i & \oplus - \mathrm{e}^{i\phi_i} \sinh r_i \\ \oplus_i - \mathrm{e}^{-i\phi_i} \sinh r_i & \oplus_i \cosh r_i \end{pmatrix}, \qquad \boldsymbol{\zeta}_s = \boldsymbol{0} \tag{2.46}$$

$$\boldsymbol{S}_s = \begin{pmatrix} \oplus_i \cosh r_i - \cos\phi_i \sinh r_i & \oplus_i - \sin\phi_i \sinh r_i \\ \oplus_i - \sin\phi_i \sinh r_i & \oplus_i \cosh r_i + \cos\phi_i \sinh r_i \end{pmatrix}, \qquad \boldsymbol{d}_s = \boldsymbol{0}, \tag{2.47}$$

where each quadrant is a diagonal matrix comprised of the direct sum for each mode in the state with squeezing $r_i \mathrm{e}^{i\phi_i}$.

2.2.3 Gaussian states

Gaussian states are those that have Gaussian Wigner functions and can be fully described by the vector of means (the first moment) and the covariance matrix (the second moment). The simplest example is the vacuum state. This is the state with no photons, with a zero vector of means and half times the identity as the covariance matrix:

$$\bar{\boldsymbol{x}}_{\text{vac}} = \bar{\boldsymbol{\zeta}}_{\text{vac}} = \boldsymbol{0}, \qquad \boldsymbol{V}_{\text{vac}} = \boldsymbol{\Sigma}_{\text{vac}} = \tfrac{1}{2}\boldsymbol{I}. \tag{2.48}$$

A single-mode state can be represented on a phase-space diagram with axes q and p. For Gaussian states, the mean values of the quadratures (ie. the first moment) provide the coordinates of the centre of the state and uncertainties give the width of the state in the q and p directions. For states with diagonal covariance matrices, these uncertainties are given by the square root of the variances of the quadratures (ie. the second moment). Therefore, a single-mode vacuum state is represented in the phase space as a circle centred on the origin, with a diameter of $1/\sqrt{2}$.

Any pure Gaussian state can be made by a Gaussian operator (or combinations of Gaussian operators) acting on the vacuum state. In order to describe the state after the Gaussian transformation, we need to know how the vector of means and covariance matrix transform. Taking the transform of the vector of means $\bar{\boldsymbol{x}} \to \text{Tr}(\hat{\rho}\hat{\mathcal{U}}^\dagger \hat{\boldsymbol{x}}\hat{\mathcal{U}})$, and with the assumption that $\hat{\boldsymbol{x}} \to \boldsymbol{S}\hat{\boldsymbol{x}} + \boldsymbol{d}$ (as shown in the previous section), the vector of means can be written as

$$\langle \hat{x}_i \rangle \to \text{Tr}(\hat{\rho}(\boldsymbol{S}\hat{\boldsymbol{x}} + \boldsymbol{d})_i) = \text{Tr}\left(\hat{\rho}\sum_j S_{i,j}\hat{x}_j + \hat{\rho}d_i\right) = \sum_j S_{i,j}\text{Tr}(\hat{\rho}\hat{x}_j) + \text{Tr}(\hat{\rho}d_i) = \sum_j S_{i,j}\bar{x}_j + d_i, \tag{2.49}$$

where we use that $\text{Tr}(\hat{\rho}) = 1$ in the final step. Hence, we have shown it is possible to determine the vector of means by the transformation matrix and vector acting on the vacuum vector of means. Similar arguments can be applied to the covariance matrix and so we find

$$\begin{aligned} \bar{\boldsymbol{x}} &\to \boldsymbol{S}\bar{\boldsymbol{x}} + \boldsymbol{d}, & \boldsymbol{V} &\to \boldsymbol{S}\boldsymbol{V}\boldsymbol{S}^\top \\ \bar{\boldsymbol{\zeta}} &\to \boldsymbol{M}\bar{\boldsymbol{\zeta}} + \boldsymbol{\zeta}, & \boldsymbol{\Sigma} &\to \boldsymbol{M}\boldsymbol{\Sigma}\boldsymbol{M}^\dagger. \end{aligned} \tag{2.50}$$

Hence a Gaussian state can be completely described by its transformation matrix and vector in either the real or complex picture (ie. $\boldsymbol{S}$ and $\boldsymbol{d}$, or $\boldsymbol{M}$ and $\boldsymbol{\zeta}$).

Any combination of the three Gaussian transformations introduced previously can be applied to the vacuum state in any order resulting in a Gaussian state. We now look at common Gaussian states and describe them in the Gaussian formalism by their covariance matrices and vectors of means and also look at their expansions in the Fock basis.

2.2.3.1 Coherent states

The first Gaussian operator introduced was the displacement operator $\hat{D}(\boldsymbol{\alpha}) = \exp\left(\hat{\boldsymbol{a}}^\dagger\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}^*\hat{\boldsymbol{a}}\right)$. As shown previously, the transformation matrix is the identity matrix in both real and complex

pictures and the complex vector is $\boldsymbol{\zeta} = \binom{\boldsymbol{\alpha}}{\boldsymbol{\alpha}^*}$ and the real vector is $\boldsymbol{d}_d = \boldsymbol{x} = \binom{\boldsymbol{q}}{\boldsymbol{p}}$. Applying these transformations to the moments of the vacuum state, we find the vectors of means and covariances matrices for the state formed by applying the displacement operator with parameter $\boldsymbol{\alpha}$ or equivalently $(\boldsymbol{q}, \boldsymbol{p})$ are

$$\boldsymbol{V}_d = \tfrac{1}{2}\boldsymbol{I}, \qquad \bar{\boldsymbol{x}}_d = \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{pmatrix}, \qquad \boldsymbol{\Sigma}_d = \tfrac{1}{2}\boldsymbol{I}, \qquad \bar{\boldsymbol{\zeta}}_d = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{pmatrix}. \tag{2.51}$$

The covariance remains unchanged and so in the phase space this operator simply is a translation in a direction determined by the parameter $\alpha = |\alpha|\mathrm{e}^{i\theta}$ without changing the shape of the state. This geometric interpretation of the action of the displacement operator in phase space is shown in fig. (2.1a). The vector of means is determined solely by the displacement operators acting on a vacuum state and so the terms 'displacement' and 'vector of means' are used interchangeably. The state formed by applying a displacement operator on the vacuum state is denoted by $|\boldsymbol{\alpha}\rangle = \hat{D}(\boldsymbol{\alpha})\,|\boldsymbol{0}\rangle$ and called simply a displaced (vacuum) state or a coherent state.

Although it is neat to write the coherent state in its Gaussian formalism, it is convenient to write the state decomposed in the Fock basis, for example in order to measure the number of photons in a state. The displacement operator can act on many modes, but each mode is independent so without loss of generality, we consider the single mode version and apply separately to each mode. We define the displacement parameter $\alpha = |\alpha|\mathrm{e}^{i\theta}$, where θ is the angle from the q quadrature and $|\alpha|$ is the distance of the translation. The coherent state expanded in terms of the Fock basis is given by [28]

$$|\alpha\rangle = \exp\!\big(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\big)\,|0\rangle = \exp\!\big(-\tfrac{1}{2}|\alpha|^2\big)\sum_{n=0}^{\infty}\frac{|\alpha|^n\mathrm{e}^{in\theta}}{\sqrt{n!}}\,|n\rangle\,. \tag{2.52}$$

From this, and using $\langle N|n\rangle = \delta_{N,n}$, it is easy to find the probability amplitude of detecting N photons from a coherent state $|\alpha\rangle$:

$$\langle N|\alpha\rangle = \exp\!\big(-\tfrac{1}{2}|\alpha|^2\big)\sum_{n=0}^{\infty}\frac{|\alpha|^n\mathrm{e}^{in\theta}}{\sqrt{n!}}\,\langle N|n\rangle = \exp\!\big(-\tfrac{1}{2}|\alpha|^2\big)\frac{|\alpha|^N\mathrm{e}^{iN\theta}}{\sqrt{N!}}. \tag{2.53}$$

And therefore the probability of measuring N photons is

$$\Pr(N) = |\,\langle N|\alpha\rangle\,|^2 = \mathrm{e}^{-|\alpha|^2}\,\frac{|\alpha|^{2N}}{N!}. \tag{2.54}$$

We note that this is a Poisson distribution with parameter $|\alpha|^2$, so the number of photons detected from a coherent state is distributed as $N \sim \mathrm{Pois}(|\alpha|^2)$. This means that each photon from a coherent state is independent and suggests they behave similarly to distinguishable photons. A consequence of this is that finding the probability of a coherent state after mixing the modes together is still easy to do as we show next.

We can observe the effect of a linear optical transformation on a multimode coherent state by analysing its action on the creation and annihilation operators. First we assume that the state

formed by a unitary evolution acting on a multimode coherent state is another coherent state of a different displacement parameter:

$$\hat{D}(\boldsymbol{\beta})\left|\mathbf{0}\right\rangle = \hat{\mathcal{U}}\hat{D}(\boldsymbol{\alpha})\left|\mathbf{0}\right\rangle. \tag{2.55}$$

We note that the unitary (and its conjugate) leaves the vacuum state unchanged when acting on it. This can be seen because the transformation matrix $\boldsymbol{M}_u$ is Hermitian and therefore leaves the identity as the identity and the translation vector $\boldsymbol{\zeta}_u$ is the zero vector. Therefore the above equivalence can be rewritten as

$$\hat{D}(\boldsymbol{\beta})\left|\mathbf{0}\right\rangle = \hat{\mathcal{U}}\hat{D}(\boldsymbol{\alpha})\hat{\mathcal{U}}^\dagger\left|\mathbf{0}\right\rangle. \tag{2.56}$$

With the expansion of the exponential in the displacement operator the relation between $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ can be found by simply analysing the action of a unitary on the creation and annihilation operators:

$$\hat{D}(\boldsymbol{\beta}) = \exp\left(\hat{\boldsymbol{a}}^\dagger\boldsymbol{\beta} - \boldsymbol{\beta}^\dagger\hat{\boldsymbol{a}}\right) = \sum_{n=0}^{\infty}\frac{1}{n!}(\hat{\boldsymbol{a}}^\dagger\boldsymbol{\beta} - \boldsymbol{\beta}^\dagger\hat{\boldsymbol{a}})^n \tag{2.57}$$

$$\hat{\mathcal{U}}\hat{D}(\boldsymbol{\alpha})\hat{\mathcal{U}}^\dagger = \sum_{n=0}^{\infty}\frac{1}{n!}\left[\hat{\mathcal{U}}(\hat{\boldsymbol{a}}^\dagger\boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger\hat{\boldsymbol{a}})\hat{\mathcal{U}}^\dagger\right]^n = \sum_{n=0}^{\infty}\frac{1}{n!}\left[\hat{\mathcal{U}}\hat{\boldsymbol{a}}^\dagger\hat{\mathcal{U}}^\dagger\boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger\hat{\mathcal{U}}\hat{\boldsymbol{a}}\hat{\mathcal{U}}^\dagger\right]^n, \tag{2.58}$$

where we inserted the identity $\hat{I} = \hat{\mathcal{U}}^\dagger\hat{\mathcal{U}}$ in the second line such that in the power the $\hat{\mathcal{U}}^\dagger$ at the end cancels with the $\hat{\mathcal{U}}$ at the beginning of the next term when multiplying out the power. It is important to notice that for the action on the covariance matrix and vector of means the transformation was $\mathcal{U}^\dagger\hat{a}^\dagger\mathcal{U}$ whereas here we have $\mathcal{U}\hat{a}^\dagger\mathcal{U}^\dagger$. In eqs. (2.38) and (2.39), this simply sends $i \to -i$ and in turn the unitary matrix becomes the Hermitian conjugate and vice versa. Therefore the action here is

$$\hat{a}_i^\dagger \to \mathcal{U}\hat{a}_i^\dagger\mathcal{U}^\dagger = (\hat{\boldsymbol{a}}^\dagger\boldsymbol{U})_i \qquad \hat{a}_i \to \mathcal{U}\hat{a}_i\mathcal{U}^\dagger = (\boldsymbol{U}^\dagger\hat{\boldsymbol{a}})_i \tag{2.59}$$

and in vector form

$$(\hat{\boldsymbol{a}}^\dagger)^\top \to \mathcal{U}\hat{\boldsymbol{a}}^\dagger\mathcal{U}^\dagger = \boldsymbol{U}^\top(\hat{\boldsymbol{a}}^\dagger)^\top \qquad \hat{\boldsymbol{a}} \to \mathcal{U}\hat{\boldsymbol{a}}\mathcal{U}^\dagger = \boldsymbol{U}^\dagger\hat{\boldsymbol{a}}. \tag{2.60}$$

Inserting these transformations back into eqs. (2.56) and (2.58), we find

$$\hat{D}(\boldsymbol{\beta})\left|\mathbf{0}\right\rangle = \hat{\mathcal{U}}\hat{D}(\boldsymbol{\alpha})\hat{\mathcal{U}}^\dagger\left|\mathbf{0}\right\rangle = \sum_{n=0}^{\infty}\frac{1}{n!}\left[\hat{\boldsymbol{a}}^\dagger\boldsymbol{U}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\dagger\boldsymbol{U}^\dagger\hat{\boldsymbol{a}}\right]^n\left|\mathbf{0}\right\rangle = \hat{D}(\boldsymbol{U}\boldsymbol{\alpha})\left|\mathbf{0}\right\rangle. \tag{2.61}$$

Hence, a unitary operator acting on a coherent state takes it to a new coherent state with the vector displacement parameter multiplied by the matrix $\boldsymbol{U}$.

2.2.3.2 Squeezed vacuum states

The final Gaussian transformation is the squeezing operator. We define the single mode squeezing operator as $\hat{S}(\xi) = \exp\left(\frac{1}{2}(\xi^*\hat{a}^2 - \xi\hat{a}^{\dagger 2})\right)$, where $\xi = r\mathrm{e}^{i\phi}$ is the complex valued squeezing parameter with ϕ being twice the angle from the $\hat{q}$ quadrature. The creation and annihilation operators here

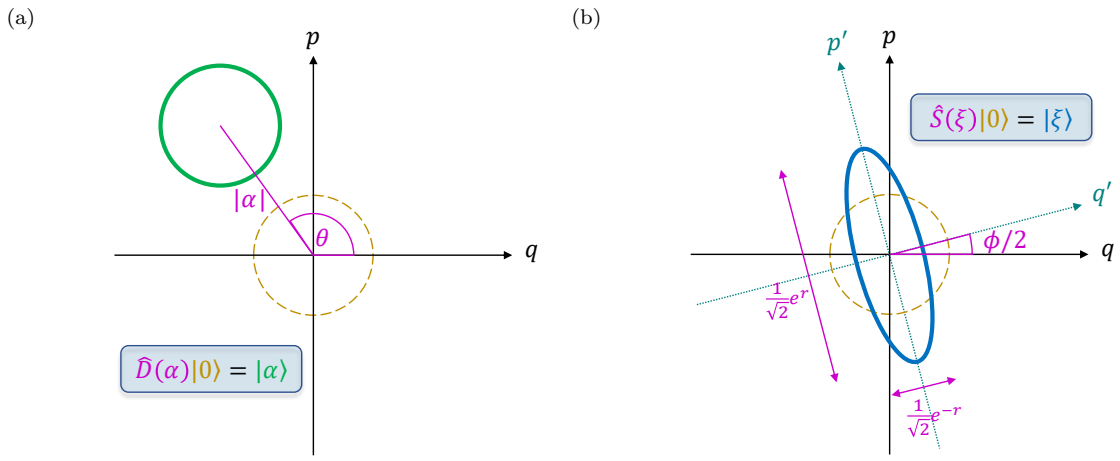

Figure 2.1: Coherent and squeezed states in phase space. The action of displacing (a) and squeezing (b) on the vacuum state (gold) in the phase space. (a) The displacement operator is a translation moving the centre of the vacuum state but maintaining the shape. The result (green) is a state with vector of means given by the displacement parameter α and the variance the same as for the vacuum (1/2). (b) The squeezing operator acts on the vacuum state positively scaling in one direction (stretching) and negatively scaling in the orthogonal direction (shrinking). The angle at which the scaling is done depends on the angle ϕ of the squeezing parameter. This angle simply rotates the axes on which the squeezing occurs. For an angle of zero, the q quadrature is reduced with the uncertainty given by $\mathrm{e}^{-r}/\sqrt{2}$ and the p quadrature is stretched to $\mathrm{e}^{r}/\sqrt{2}$. For any non-zero angle the operation can be considered as a rotation to new coordinates (turquoise) followed by a real valued squeezing and then reversing the rotation.

are quadratic and so photons are produced and destroyed in pairs. When acting the symplectic and complex transformation matrices (eqs. (2.47) and (2.46)) on the vacuum state, this corresponds to covariance matrices

$$\begin{aligned} \boldsymbol{V}_s &= \frac{1}{2}\begin{pmatrix} \cosh(2r) - \cos\phi\sinh(2r) & -\sin\phi\sinh(2r) \\ -\sin\phi\sinh(2r) & \cosh(2r) + \cos\phi\sinh(2r) \end{pmatrix}, \\ \boldsymbol{\Sigma}_s &= \frac{1}{2}\begin{pmatrix} \cosh(2r) & -\mathrm{e}^{i\phi}\sinh(2r) \\ -\mathrm{e}^{-i\phi}\sinh(2r) & \cosh(2r) \end{pmatrix}, \end{aligned} \tag{2.62}$$

with the vector of means unchanged. From this it is not very clear what this transformation does geometrically to the vacuum state in the phase space. However, setting the angle of the squeezing $\phi = 0$ so that the squeezing parameter is real, the covariance matrices are much simpler:

$$\boldsymbol{V} = \frac{1}{2}\begin{pmatrix} \mathrm{e}^{-2r} & 0 \\ 0 & \mathrm{e}^{2r} \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \frac{1}{2}\begin{pmatrix} \cosh(2r) & -\sinh(2r) \\ -\sinh(2r) & \cosh(2r) \end{pmatrix}. \tag{2.63}$$

In the quadrature picture, the q quadrature variance is reduced to $\mathrm{e}^{-2r}/2$ while the p quadrature variance is increased to $\mathrm{e}^{2r}/2$, which corresponds to squeezing in the q quadrature and anti-squeezing (expanding) in the p quadrature. In the case that the angle of the squeezing is not zero, we can consider rotating the coordinate system to find one that has real squeezing only. This is found by

rotating the coordinate system q and p by $\phi/2$ anticlockwise. This geometric interpretation of the squeezing operator acting on the vacuum state is given in fig. (2.1).

Now we can treat a squeezed state in the non-Gaussian formalism and expand it in the number basis as [28]

$$|\xi\rangle = \frac{1}{\sqrt{\cosh r}} \sum_{n=0}^{\infty} (-1)^n \frac{\sqrt{(2n)!}}{2^n n!} \mathrm{e}^{in\phi} \tanh^n r \, |2n\rangle \, . \tag{2.64}$$

From this series, it is convenient to find the probability amplitude of measuring $2N$ photons from a squeezed vacuum state:

$$\begin{aligned} \langle 2N|\xi\rangle &= \frac{1}{\sqrt{\cosh r}} \sum_{n=0}^{\infty} (-1)^n \frac{\sqrt{(2n)!}}{2^n n!} \mathrm{e}^{in\phi} \tanh^n r \, \langle 2N|2n\rangle \\ &= \frac{1}{\sqrt{\cosh r}} (-1)^N \frac{\sqrt{(2N)!}}{2^N N!} \mathrm{e}^{iN\phi} \tanh^N r \end{aligned} \tag{2.65}$$

Therefore the probability of measuring $2N$ photons from a squeezed vacuum state is

$$|\, \langle 2N|\xi\rangle \,|^2 = \frac{1}{\cosh r} \frac{(2N)!}{2^{2N} N!^2} \tanh^{2N} r. \tag{2.66}$$

It is easy to see that the probability of measuring an odd number of photons $|\, \langle 2N+1|\xi\rangle \,|^2$ will include the term $\langle 2N+1|2n\rangle = \delta_{2N+1,2n}$ which will always be zero because an odd number can never equal an even number. Therefore, as expected from the quadratic nature of the annihilation and creation operators, photons are only ever measured in pairs from a squeezed state.

2.2.3.3 Squeezed coherent states

A combination of squeezing and displacement operators will give a more general squeezed coherent state $|\alpha, \xi\rangle$. As the displacement and squeezing operators do not commute, it is important to specify which order they are applied. In general in this thesis, unless otherwise stated, we assume that the squeezing is applied first followed by the displacement as we find it to be more convenient. The probability distribution of the number of photons only depends on the position of the centre and shape of the state in phase space. Applying squeezing to the vacuum state changes its shape only and applying displacement to any state changes only the position of the centre. So we can treat the two processes separately if applying them in this order. On the other hand, if the squeezing is applied to the displaced state, that operation changes both the shape and central position so we cannot consider the operators separately. However, similarly to how swapping the order of a unitary and displacement operator is equivalent for different displacement parameters, changing the order of the squeezing and displacement can also be accounted for by changing the displacement parameter. Specifically a state with squeezing parameter $\xi = r\mathrm{e}^{i\phi}$ and displacement parameter α can equivalently be written as [43]

$$|\alpha, \xi\rangle = D(\alpha) S(\xi) \, |0\rangle = S(\xi) D(\beta) \, |0\rangle \tag{2.67}$$

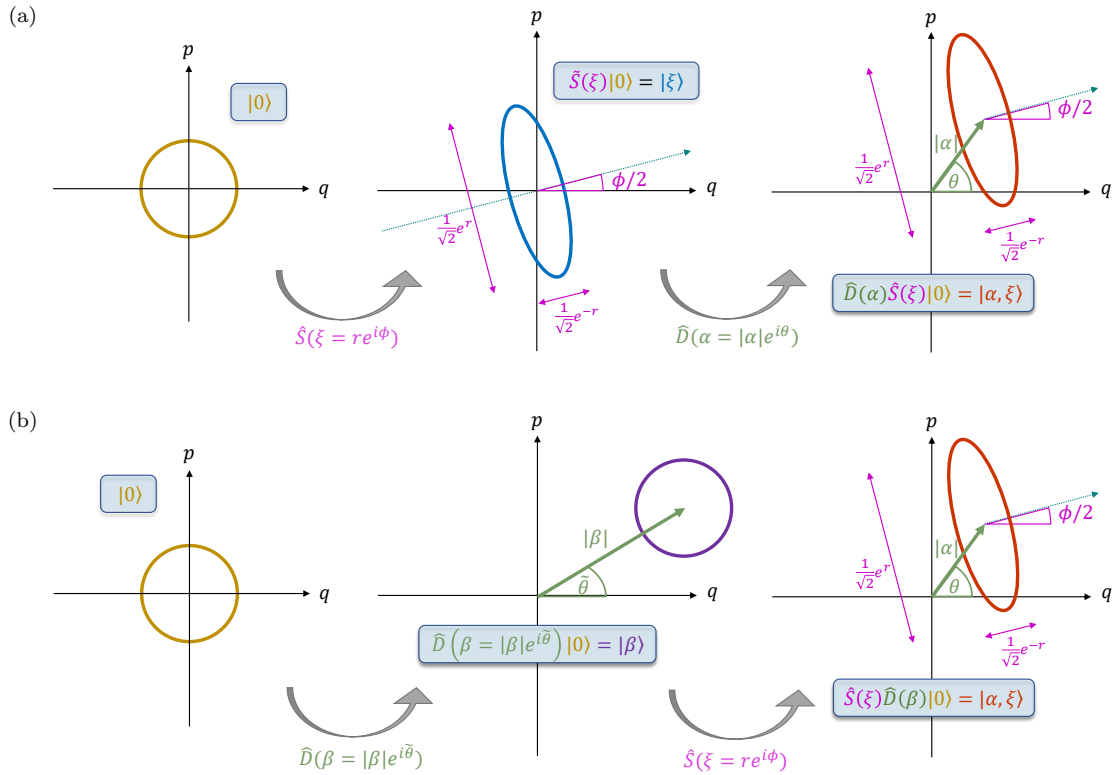

Figure 2.2: Displaced squeezed states and squeezed displaced states equivalence. (a) A vacuum state (gold) is first squeezed with squeezing parameter $\xi = re^{i\phi}$ and then displaced by displacement parameter $\alpha = |\alpha|e^{i\theta}$. (b) The same state is formed as in (a) but by first displacing by a different parameter $\beta = |\beta|e^{i\tilde{\theta}}$ and then squeezing by the same parameter ξ. The order of applying displacement and squeezing matters because the squeezing scales the whole phase space moving the centre of the state unless it is at the origin. The centre of the state determines the overall displacement.

where $\beta = \alpha\cosh r + \alpha^* e^{i\phi}\sinh r$, or to convert in the opposite direction $\alpha = \beta\cosh r - \beta^* e^{i\phi}\sinh r$.

We can derive this equivalence geometrically on a phase-space diagram. First we restrict to squeezing parallel to the q quadrature, corresponding to $\phi = 0$. Displacement is a translation and squeezing is a compression and enlargement in the direction of the q (x-axis) and p (y-axis) quadratures respectively. Therefore squeezing a displaced state moves the centre of the state towards the y-axis and away from the x-axis. In the case of squeezing first and then displacing with a displacement parameter $\alpha = |\alpha|e^{i\theta}$, the location of the centre depends only on the displacement and so is at the point $(|\alpha|\cos\theta, |\alpha|\sin\theta)$. In the case where we displace first with displacement parameter $\beta = |\beta|e^{i\theta}$, the centre of the state is at $(|\beta|\cos\theta, |\beta|\sin\theta)$. When we squeeze it, the q coordinates (ie. real part) undergo an enlargement by the factor e^{-r} and the p coordinates (ie. imaginary part) undergo an enlargement by the factor e^{r}. So now the centre of the state is at $(|\beta|\cos\theta\, e^{-r}, |\beta|\sin\theta\, e^{r})$. This is the real and imaginary parts of the displacement parameter which

can be found as

$$\begin{aligned}\alpha &= |\beta|\cos\theta\,\mathrm{e}^{-r} + i|\beta|\sin\theta\,\mathrm{e}^{r} \\ &= |\beta|\cos\theta(\cosh r - \sinh r) + i|\beta|\sin\theta(\cosh r + \sinh r) \\ &= |\beta|\cosh r(\cos\theta + i\sin\theta) - |\beta|\sinh r(\cos\theta - i\sin\theta) \\ &= |\beta|\cosh r\,\mathrm{e}^{i\theta} - |\beta|\sinh r\,\mathrm{e}^{-i\theta} \\ &= \beta\cosh r - \beta^{*}\sinh r,\end{aligned} \tag{2.68}$$

which agrees with the conversion for $\phi = 0$. A squeezed displaced state formed by applying squeezing and displacement in either order is depicted in the phase space in fig. (2.2).

Now we consider the general case where squeezing is along a line at an angle of $\phi/2$ to the x-axis (q quadrature). It is convenient to define the state in a new coordinate system such that the squeezing is along the new x-axis. This is simply done by a rotation of an angle $\phi/2$ anticlockwise which results in a state with displacement $\beta\mathrm{e}^{-i\phi/2}$. The squeezing is now applied parallel to the x-axis so the previous equation can be applied to this displaced state:

$$\alpha_{rotated} = \beta\mathrm{e}^{-i\phi/2}\cosh r - \beta^{*}\mathrm{e}^{i\phi/2}\sinh r. \tag{2.69}$$

This is the displacement parameter in the rotated coordinate system, so we need to rotate back to the correct coordinates. The displacement in the original coordinates is

$$\begin{aligned}\alpha &= (\beta\mathrm{e}^{-i\phi/2}\cosh r - \beta^{*}\mathrm{e}^{i\phi/2}\sinh r)\mathrm{e}^{i\phi/2} \\ &= \beta\cosh r - \beta^{*}\mathrm{e}^{i\phi}\sinh r\end{aligned} \tag{2.70}$$

as expected. This process of rotating to new coordinates for squeezing at an arbitrary angle is shown in fig. (2.3).

A squeezed coherent state $|\alpha, r\mathrm{e}^{i\phi}\rangle$ can also be expressed in the Fock basis as [28]

$$|\alpha, r\mathrm{e}^{i\phi}\rangle = \frac{\exp\left(-\frac{1}{2}(|\alpha|^2 + \alpha^{*2}\mathrm{e}^{i\phi}\tanh r)\right)}{\sqrt{\cosh r}}\sum_{n=0}^{\infty}\frac{1}{\sqrt{n!}}\left(\frac{e^{i\phi}\tanh r}{2}\right)^{n/2}\mathrm{H}_n\left(\frac{\alpha + \alpha^{*}\mathrm{e}^{i\phi}\tanh r}{\sqrt{2\mathrm{e}^{i\phi}\tanh r}}\right) \tag{2.71}$$

from which the probability for N photons can be found to be

$$\Pr(N) = \frac{\exp\left(-|\alpha|^2 - \frac{1}{2}(\alpha^{*2}\mathrm{e}^{i\phi} + \alpha^{2}\mathrm{e}^{-i\phi})\tanh r\right)}{N!\cosh r}\left(\frac{\tanh r}{2}\right)^{N}\left|\mathrm{H}_N\left(\frac{\alpha + \alpha^{*}\mathrm{e}^{i\phi}\tanh r}{\sqrt{2\mathrm{e}^{i\phi}\tanh r}}\right)\right|^2, \tag{2.72}$$

where H_N is the Nth Hermite polynomial. This is derived rigorously finding the decomposition in the Fock basis using the action of creation and annihilation operators and recursion relations. However, here we motivate the probability from the combination of squeezing and displacement. In quantum mechanics, the probability of a starting state ending in a final state is found by summing the probability amplitudes of all the ways that this could happen and then taking the modulus square of that total. So here we consider that a photon measured from a squeezed coherent state could have come from the squeezing part or the displaced part. Hence the probability amplitude

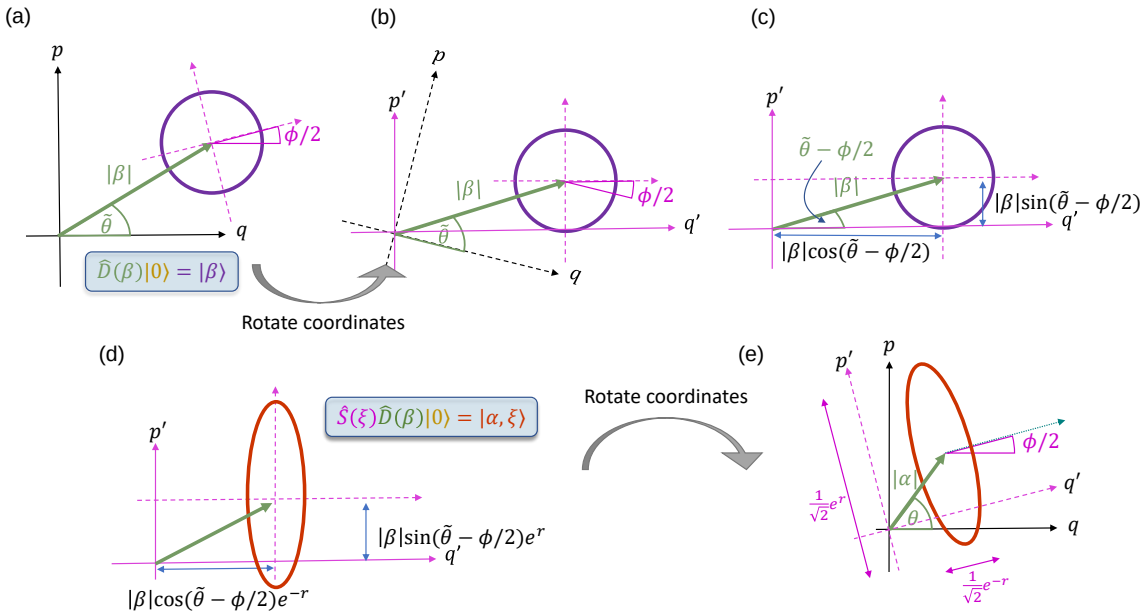

Figure 2.3: General squeezed displaced state. When squeezing is applied to a displaced state at an arbitrary angle it is convenient to rotate the coordinate system to align the angle of squeezing with the x-axis and then rotate back. (a) A displaced state at an angle of $\tilde{\theta}$. (b) Rotate to new coordinates p' and q' such that the squeezing is now parallel to the x-axis. (c) This gives a displaced state at an angle of $\tilde{\theta}-\phi/2$, where $\phi/2$ is the angle of the squeezing in the original frame. (d) Apply the squeezing now in the x direction. This multiplies all x coordinates by e^{-r} and all y coordinates by e^{r}. (e) Rotate back to the original coordinate system.

of a squeezed coherent state measuring N photons can be found by summing the amplitudes of all possible ways of getting N photons from a combination of 'displacement photons' and 'squeezing photons'. For example, if there are N photons, there could be $N-q$ squeezing photons and q displacement photons in $N!/(q!(N-q)!)$ ways, for any q.

First we consider the case where we have an even number of photons $N=2n$. Here we must have an even number of squeezing photons and therefore an even number of displacement photons. The probability amplitude is given by

$$\begin{aligned}
\langle 2n|\alpha, r\mathrm{e}^{i\phi}\rangle &= \mathcal{N}\sum_{q=0}^{n}\sqrt{\frac{(2n)!}{(2q)!(2n-2q)!}}\,\langle 2q|\gamma\rangle\,\langle 2n-2q|r\mathrm{e}^{i\phi}\rangle \\
&= \mathcal{N}\sum_{q=0}^{n}\sqrt{\frac{(2n)!}{(2q)!(2n-2q)!}}\mathrm{e}^{-|\gamma|^2/2}\frac{\gamma^{2q}}{\sqrt{(2q)!}}\frac{(-1)^{n-q}}{\sqrt{\cosh r}}\frac{\sqrt{(2n-2q)!}}{2^{n-q}(n-q)!}\mathrm{e}^{i\phi(n-q)}\tanh^{n-q} r \\
&= \mathcal{N}\frac{\mathrm{e}^{-|\gamma|^2/2}}{\sqrt{\cosh r}}\left(\frac{\mathrm{e}^{i\phi}\tanh r}{2}\right)^{n}\sum_{q=0}^{n}(-1)^{n-q}\frac{\sqrt{(2n)!}}{(2q)!(n-q)!}\left(\frac{2\gamma\mathrm{e}^{-i\phi/2}}{\sqrt{2\tanh r}}\right)^{2q} \\
&= \mathcal{N}\frac{\mathrm{e}^{-|\gamma|^2/2}}{\sqrt{(2n)!\,\cosh r}}\left(-\frac{\mathrm{e}^{i\phi}\tanh r}{2}\right)^{n}\mathrm{H}_{2n}\left(\frac{\gamma\mathrm{e}^{-i\phi/2}}{\sqrt{2\tanh r}}\right),
\end{aligned} \tag{2.73}$$

where we have used the following polynomial definition of the even numbered Hermite polynomials:

$$H_{2k}(x) = (-1)^k (2k)! \sum_{j=0}^{k} (-1)^j \frac{1}{(2j)!(k-j)!} (2x)^{2j}. \tag{2.74}$$

As we are considering the squeezing and displacement separately, we must change the displacement such that it can be considered as acting independently to the squeezing. Thus we use the displacement given by γ

$$\gamma = \alpha + \alpha^* \mathrm{e}^{i\phi} \tanh r. \tag{2.75}$$

This conversion is explored further in section (2.4.3.3). Putting this into the equation for the probability amplitude and taking the modulus square we find the expected probability as long as we include a normalisation constant $\mathcal{N}$ to ensure the probabilities sum to one.

Now we consider the case where we have an odd number of photons $N = 2n + 1$. Here we must have an even number of squeezing photons but an odd number of displacement photons. The probability amplitude is given by

$$\begin{aligned}
\langle 2n+1|\alpha, r\mathrm{e}^{i\phi}\rangle &= \mathcal{N} \sum_{q=0}^{n} \sqrt{\frac{(2n+1)!}{(2q+1)!(2n-2q)!}} \langle 2q+1|\gamma\rangle \langle 2n-2q|r\mathrm{e}^{i\phi}\rangle \\
&= \mathcal{N} \sum_{q=0}^{n} \sqrt{\frac{(2n+1)!}{(2q+1)!(2n-2q)!}} \mathrm{e}^{-|\gamma|^2/2} \frac{\gamma^{2q+1}}{\sqrt{(2q+1)!}} \times \\
&\qquad \frac{1}{\sqrt{\cosh r}} (-1)^{n-q} \frac{\sqrt{(2n-2q)!}}{2^{n-q}(n-q)!} \mathrm{e}^{i\phi(n-q)} \tanh^{n-q} r \\
&= \mathcal{N} \frac{\mathrm{e}^{-|\gamma|^2/2}}{\sqrt{\cosh r}} \left(\frac{\mathrm{e}^{i\phi} \tanh r}{2}\right)^{n+1/2} \sum_{q=0}^{n} (-1)^{n-q} \frac{\sqrt{(2n+1)!}}{(2q+1)!(n-q)!} \left(\frac{2\gamma \mathrm{e}^{-i\phi/2}}{\sqrt{2\tanh r}}\right)^{2q+1} \\
&= \mathcal{N} \frac{\mathrm{e}^{-|\gamma|^2/2} (-1)^n}{\sqrt{(2n+1)!\, \cosh r}} \left(\frac{\mathrm{e}^{i\phi} \tanh r}{2}\right)^{n+1/2} \mathrm{H}_{2n+1}\left(\frac{\gamma \mathrm{e}^{-i\phi/2}}{\sqrt{2\tanh r}}\right),
\end{aligned} \tag{2.76}$$

where we now use the polynomial expansion definition of odd numbered Hermite polynomials

$$H_{2k+1}(x) = (-1)^k (2k+1)! \sum_{j=0}^{k} (-1)^j \frac{1}{(2j+1)!(k-j)!} (2x)^{2j+1}. \tag{2.77}$$

Again we need to substitute the displacement parameter to be independent of the squeezing using eq. (2.75) and renormalise to find the correct probability amplitude.

It is clear by taking the modulus square of these probability amplitudes the expected expressions for the probabilities of measuring N photons from a squeezed displaced state [28] have been recovered by summing over all the possible ways of getting q photons from the displacement and $N - q$ photons from the squeezing. This may provide an intuitive description of the photon number probability from squeezed coherent states.

2.3 Permanents, hafnians and loop hafnians

In anticipation of describing boson sampling in the next section, we introduce the matrix functions that are used in the classical simulation of boson sampling. For the variants that we cover in this thesis, these are permanents, hafnians and loop hafnians. Although they are different functions, they are closely linked to each other and can all be connected to graph theory. We introduce some terminology for graphs below.

A **graph** is a collection of vertices (or nodes) with connecting edges which tell you something about the relationship between the two vertices they join. We denote an edge by the beginning and end vertex it connects, (V_i, V_j). In this thesis, we define graphs to be allowed 'loops' where an edge starts and ends at the same node, and refer to graphs that do not allow loops as '**simple graphs**' to distinguish the two cases. A graph may be **weighted** or not. If it is not weighted, any two vertices are either connected by an edge or they are not. Strictly speaking the existence of an edge implies a weighting of one and the lack of an edge means a weighting of zero. If it is a weighted graph, the edges can have any value associated to them. A graph is **undirected** if the weighting on the edges is the same going from vertex V_i to V_j as in the opposite direction from V_j to V_i. There is a special instance called a **bipartite graph** where the vertices can be split into two groups with no edges within each group. A **matching** in a graph is a collection of edges such that no vertex is included more than once. A **perfect matching** is a matching such that all vertices are included precisely once. These terms are all depicted in fig. (2.4). Any graph can be equivalently written in the form of its **adjacency matrix**, $\boldsymbol{A}$, where the element $A_{i,j}$ is the weighting between vertex V_i and V_j. In an unweighted graph, all these elements are either 0 (no edge) or 1 (an edge). In an undirected graph, as the weighting is the same in both directions, $(V_i, V_j) = (V_j, V_i) \implies A_{i,j} = A_{j,i}$, and so the adjacency matrix is symmetric. The mapping between graphs and their adjacency matrices is shown in fig. (2.5).

2.3.1 Permanent

The permanent is defined for any square $N \times N$ matrix, $\boldsymbol{A}$, as

$$\mathrm{Perm}(\boldsymbol{A}) = \sum_{\sigma \in S_N} \prod_{i=1}^{N} A_{i,\sigma(i)}, \tag{2.78}$$

where the sum is over the symmetric group, S_N, ie. all the permutations of the elements from 1 to N. For example, for $N = 3$, $S_3 = \{(1,2,3), (2,3,1), (3,1,2), (3,2,1), (2,1,3), (1,3,2)\}$. $A_{i,\sigma(i)}$ is the element of matrix $\boldsymbol{A}$ in row i and the column given by the ith element of ordering σ. The product inside the sum matches all integers from 1 to N with an element from a permutation of those integers. In terms of a graph, this is the product of the edge weightings in a perfect matching in a bipartite graph where the first subscript in $A_{i,\sigma(i)}$ corresponds to a vertex on the left-side and is matched to a vertex $\sigma(i)$ from the group of vertices on the right-hand side. The sum is over all possible permutations which covers all possible perfect matchings. So the permanent of a matrix $\boldsymbol{A}$ is equivalent to the total weight of all possible perfect matchings on a graph with adjacency matrix

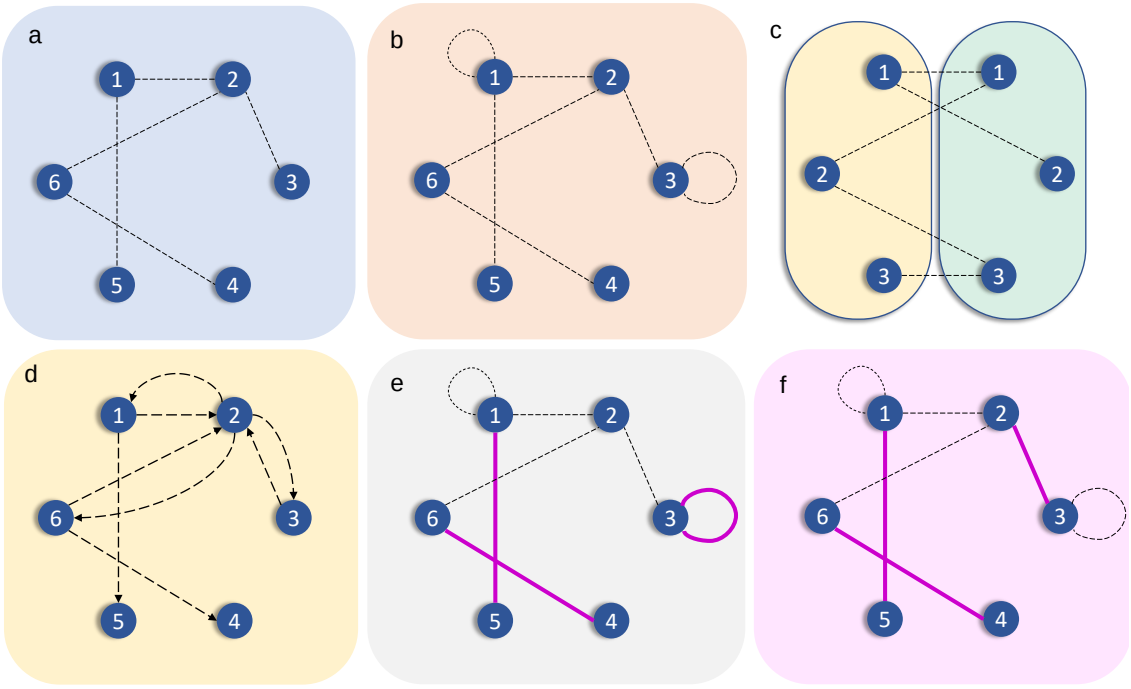

Figure 2.4: Properties of graphs. The vertices are represented by the blue circles and labelled with integers inside. The edges are shown by dashed black lines between the vertices. (a) A simple graph with no loops. Missing edges correspond to edges with a weighting of zero. (b) A graph with loops. (c) A bipartite graph. The vertices are split into two groups, left and right with no edges within each group. The vertices are relabelled as the halves can be treated separately. (d) A directed graph. All the edges have a direction associated with them with potentially different weights on edges between the same two vertices but in opposite directions. (e) A matching. The purple lines represent the matching {(1,5)(3,3)(4,6)} including vertices 1,3,4,5 and 6 once and not including vertex 2. (f) A perfect matching. The purple lines show a matching which includes all vertices {(1,5)(2,3)(4,6)}.

A. In an unweighted bipartite graph, the permanent is equal to the number of perfect matchings. This is because the permanent sums over all possible perfect matchings and adds weight 1 if all edges in that matching exist and 0 otherwise.

The naive way to calculate this quantity requires summing over all the permutations of N elements. However, there are $N!$ ways of permuting N elements and a factorial scaling means this quickly becomes unmanageable as the size of the matrix increases. There have been several algorithms to calculate the permanent with a better scaling. The first was introduced by Ryser [44] and has time complexity $O(N^2 2^N)$. This algorithm was improved on to reduce the complexity to $O(N2^N)$ by Nijenhuis and Wilf [45] and later another approach was developed by Glynn with the same complexity [46]. Although these algorithms are much faster than the naive approach, it was shown that calculating the permanent lies in the #P complexity class and therefore cannot be done in polynomial time [47].

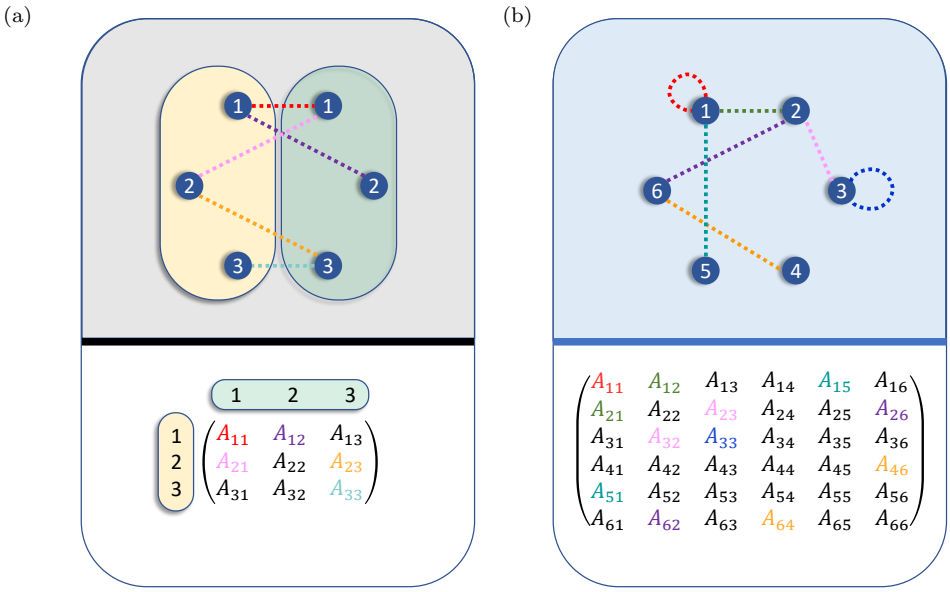

Figure 2.5: The adjacency matrix of a graph. The weighting of the edges are given by elements in the adjacency matrix associated with a graph. The elements for each edge are colour coordinated. In the bipartite graph (a), the nodes on the left are labelled separately to the ones on the right and this is reflected in the matrix by linking the left group with the row index and the right group with the column index. This allows the size of the matrix to be reduced for the same number of nodes compared to a general graph. For an undirected general graph (b) the weighting of the edges is the same in each direction and so the weighting is given in $A_{i,j}$ and $A_{j,i}$.

2.3.2 Hafnian

The hafnian is defined for a symmetric even-sized square $2N \times 2N$ matrix $\boldsymbol{A}$ as

$$\mathrm{Haf}(\boldsymbol{A}) = \frac{1}{N!2^N} \sum_{\sigma \in S_{2N}} \prod_{i=1}^{N} A_{\sigma(2i-1),\sigma(2i)}, \tag{2.79}$$

where S_{2N} is the symmetric group over $2N$. Here the product inside the sum is a perfect matching in a simple graph with adjacency matrix $\boldsymbol{A}$. However, the sum includes all permutations including permuting the ordering of the vertices for an edge (eg. includes (V_1, V_2) and (V_2, V_1)) which means accounting each perfect matching twice for every pair (ie. the same perfect matching is included 2^N times due to this). Also the ordering of the edges does not matter and summing over the symmetric group includes all the permutations of the edges (eg. $(V_1, V_2)(V_3, V_4)$ and $(V_3, V_4)(V_1, V_2)$) and this contributes another multiplicative factor for the number of ways of ordering the N edges (ie. $N!$). However the factor before the sum takes this over-inclusion into account and the hafnian is the sum of the weights of all perfect matchings over a simple graph [48]. In an unweighted graph, the hafnian is the number of perfect matchings.

The number of possible perfect matchings in a simple graph with $2N$ vertices is $(2N-1)!!$ which does not scale well as the size of the matrix increases and so a naive implementation of this formula

quickly becomes unmanageable. Again there are several algorithms that run in exponential time $\mathrm{O}(\mathrm{poly}(N)2^N)$ [49, 50, 51].

2.3.3 Loop hafnian

The loop hafnian is defined for a symmetric square $N \times N$ matrix as

$$\mathrm{Lhaf}(\boldsymbol{A}) = \sum_{\sigma \in \mathrm{SPM}} \prod_{i=1}^{N} A_{\sigma(2i-1),\sigma(2i)}, \tag{2.80}$$

where SPM is the set of perfect matchings of a graph with loops. This is a natural extension of the hafnian and can be computed with the same time complexity as for a hafnian [49]. The loop hafnian of an unweighted graph finds the number of perfect matchings in a graph with loops which is given by the telephone numbers.

2.3.4 Connection between permanents, hafnians and loop hafnians

There is an obvious connection between permanents, hafnians and loop hafnians when considering the graph theory representations; namely that they all sum over the weights of all the perfect matchings of different types of graphs. The most general is the loop hafnian which is the perfect matchings of an arbitrary graph with loops. The hafnian is a specific loop hafnian where the loops are not allowed which translates into a zero diagonal in the matrix:

$$\mathrm{Haf}(\boldsymbol{A}) = \mathrm{Lhaf}(\boldsymbol{0}, \boldsymbol{A}), \tag{2.81}$$

where the notation $\mathrm{Lhaf}(\boldsymbol{x}, \boldsymbol{Y})$ means to take the loop hafnian of the matrix $\boldsymbol{Y}$ but where the diagonal elements have been replaced with the vector $\boldsymbol{x}$.

A perfect matching in a *bipartite* graph can be considered as a perfect matching in a *general* graph where the vertices can be split into two groups such that the weightings on all edges between two vertices in the same group are zero. The adjacency matrix of such a general graph will have a block form when the vertices are ordered such that each group is separate. The leading diagonal blocks will be the all-zeros matrix corresponding to the zero weightings within each group. From this argument, it follows that the hafnian can be reduced to a permanent if the matrix is of the following form:

$$\mathrm{Perm}\,(\boldsymbol{A}) = \mathrm{Haf}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{A}^\top \\ \boldsymbol{A} & \boldsymbol{0} \end{pmatrix}. \tag{2.82}$$

The mapping of these matrix functions to graphs and their adjacency matrices is shown in fig. (2.6) to highlight the equivalence between them under certain conditions of the adjacency matrix.

A final useful identity is when the matrix is block diagonal. This corresponds to splitting the vertices into two groups as for the permanent, but now there are *only* edges within a group and none between them. As there is no linking between the two groups, they can be treated completely

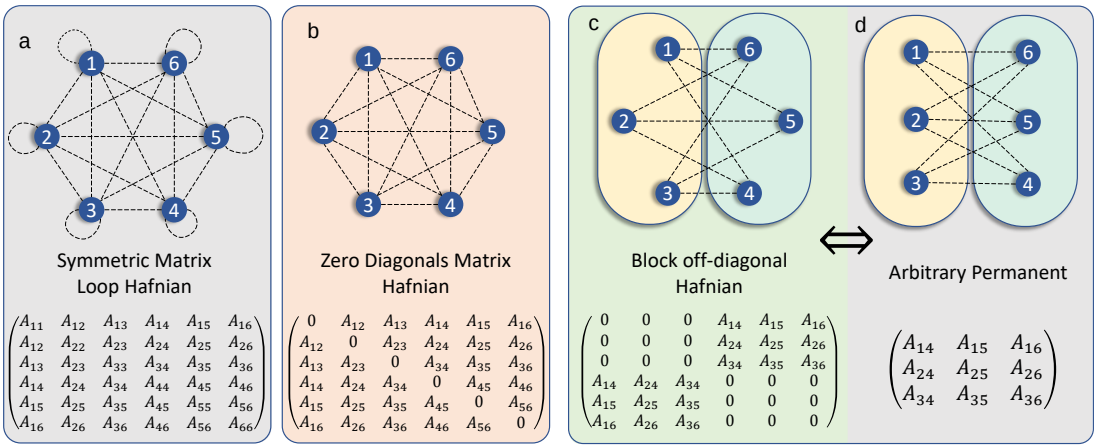

Figure 2.6: The mapping between permanents, hafnians and loop hafnians. (a) The loop hafnian is of a symmetric matrix which includes the weights of the loops on the diagonals. (b) The hafnian does not include the loops and therefore the hafnian of any symmetric adjacency matrix and the loop hafnian of this adjacency matrix with zeros on the diagonal are the same. (c) For a bipartite graph, the elements in the same group are not joined by edges which results in zeros in the block diagonals in the adjacency matrix. (d) The same graph but now we consider the labelling in one group as different to the other. In this way we can find an equivalence between the permanent of a matrix and a hafnian (or loop hafnian) of a symmetric block off-diagonal matrix.

independently and the hafnian becomes the product of hafnians of each subgraph:

$$\mathrm{Haf}\begin{pmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{pmatrix} = \mathrm{Haf}(\boldsymbol{C})\mathrm{Haf}(\boldsymbol{D}). \tag{2.83}$$

The calculation of two $N \times N$ hafnians is much faster than one $2N \times 2N$ matrix due to the exponential scaling of the complexity of hafnians and so this property can be used to provide a quadratic speed-up whenever a matrix is block diagonal. The above result does not make any assumptions about the size of either block and can be extended to loop hafnians too.

2.4 Boson sampling

Boson sampling is the task of sampling from a particular probability distribution. This probability distribution is naturally given by measuring an entangled state formed by passing a quantum state through an $M \times M$ interferometer. In the original proposal by Aaronson and Arkhipov, the input state is a Fock state with single photons in, typically the top, N modes [6]. There have since been many variants on this recipe which take advantage of less restrictive experimental requirements, particularly the need for deterministic single photon sources. In this section, we review the most common types of boson sampling.

2.4.1 Standard boson sampling

The original proposal introduced as BosonSampling and commonly referred to in literature by various names including Fock boson sampling, vanilla boson sampling and Aaronson and Arkhipov boson sampling, we call standard boson sampling throughout this thesis. We use the phrase 'boson sampling' to mean any variant, including but not limited to standard boson sampling. The experimental task is as follows: An input Fock state with N photons $|1,\dots,1,0,\dots,0\rangle$ is passed through an interferometer described by a unitary matrix, $\boldsymbol{U}$, resulting in an entangled state. This state is then measured with photon number resolving detectors to project onto a pure number state. The measured number of photons in each mode $(t_1,\dots,t_M)$ is referred to as the 'photon number pattern' or the 'output photon pattern' throughout this thesis. This task is equivalent to sampling from a probability distribution which is determined by the entangled state at the output of the interferometer. The quantum device naturally samples from this distribution, but if we want to sample from it classically, we need to be able to calculate probabilities of the output photon patterns. Here we derive the probability expression.

The input state is typically given by $|1,\dots,1,0,\dots,0\rangle = \hat{a}_1^\dagger\cdots\hat{a}_N^\dagger|0,\dots,0\rangle$. However, we generalise this to any Fock state with N photons $|s_1,\dots,s_M\rangle$, where $\sum_i s_i = N$. We use the notation $\boldsymbol{s}$ for the pattern in second quantisation and $\tilde{\boldsymbol{s}}$ in the first quantisation, $(\tilde{s}_1,\dots,\tilde{s}_N)$, where $\tilde{s}_i \in \{1,\dots,M\}$. As previously shown, the creation operators are transformed under a unitary transformation $\hat{\mathcal{U}}$ as $\hat{a}_i^\dagger \to \sum_j U_{ji}\hat{a}_j^\dagger$. Hence, the state given at the output of the interferometer $|\rho_{\text{out}}\rangle$ is given by

$$\begin{aligned}
|\rho_{\text{out}}\rangle &= \hat{\mathcal{U}}|\rho_{\text{in}}\rangle = \hat{\mathcal{U}}|s_1,\dots,s_M\rangle = \hat{\mathcal{U}}\frac{1}{\prod_i\sqrt{s_i!}}\hat{a}_{\tilde{s}_1}^\dagger\cdots\hat{a}_{\tilde{s}_N}^\dagger|0,\dots,0\rangle \\
&= \frac{1}{\prod_i\sqrt{s_i!}}\sum_{j_1=1}^{M} U_{j_1,\tilde{s}_1}\hat{a}_{j_1}^\dagger\cdots\sum_{j_N=1}^{M} U_{j_N,\tilde{s}_N},\hat{a}_{j_N}^\dagger|0,\dots,0\rangle \\
&= \frac{1}{\prod_i\sqrt{s_i!}}\sum_{j_1=1}^{M}\cdots\sum_{j_N=1}^{M} U_{j_1,\tilde{s}_1}\cdots U_{j_N,\tilde{s}_N}\hat{a}_{j_1}^\dagger\cdots\hat{a}_{j_N}^\dagger|0,\dots,0\rangle\,.
\end{aligned} \tag{2.84}$$

The probability of measuring a particular photon number pattern, $\boldsymbol{t} = (t_1,\dots,t_M)$, is determined by the probability amplitude, $\langle\boldsymbol{t}|\rho_{\text{out}}\rangle$, according to $\Pr(\boldsymbol{t}) = |\langle\boldsymbol{t}|\rho_{\text{out}}\rangle|^2$. We find the probability amplitude to be

$$\langle\boldsymbol{t}|\rho_{\text{out}}\rangle = \frac{1}{\prod_i\sqrt{t_i!s_i!}}\sum_{j_1=1}^{M}\cdots\sum_{j_N=1}^{M} U_{j_1,\tilde{s}_1}\cdots U_{j_N,\tilde{s}_N}\langle 0,\dots,0|\,\hat{a}_{\tilde{t}_1}\cdots\hat{a}_{\tilde{t}_N}\hat{a}_{j_1}^\dagger\cdots\hat{a}_{j_N}^\dagger|0,\dots,0\rangle\,. \tag{2.85}$$

The braket can be simplified to reduce the sum:

$$\langle 0,\dots,0|\,\hat{a}_{\tilde{t}_1}\cdots\hat{a}_{\tilde{t}_N}\hat{a}_{j_1}^\dagger\cdots\hat{a}_{j_N}^\dagger|0,\dots,0\rangle = \begin{cases}\prod_i j_i! & \text{if } \tilde{t}_1,\dots,\tilde{t}_N = j_1,\dots,j_N \text{ in any order}\\ 0 & \text{otherwise}\end{cases} \tag{2.86}$$

The product of factorials comes from the action of creation operators on Fock states: $(\hat{a}^\dagger)^k |0\rangle = \sqrt{k!}\,|k\rangle$, and similarly for annihilation operators: $\langle 0|\,(\hat{a})^k = \langle k|\sqrt{k!}$. All vectors $\boldsymbol{j} = (j_1, \ldots, j_N)$ that satisfy the condition $\tilde{t}_1, \ldots, \tilde{t}_N = j_1, \ldots, j_N$ in any order is the set of all permutations of integers $\tilde{t}_1, \ldots, \tilde{t}_N$ which is the definition of the symmetric group $S_{\tilde{\boldsymbol{t}}}$. In the case of repeated values in the output $\tilde{\boldsymbol{t}}$, the symmetric group includes the permutations of all elements including the repeated values (which gives the same pattern), whereas the sum over all $\boldsymbol{j}$ above only includes this vector/pattern once. So the value must be divided by the number of permutations that leave the vector unchanged when the sum is changed to be over the symmetric group. Hence, the sum in eq. (2.85) can be simplified to include only terms where $\boldsymbol{j} \in S_{\tilde{\boldsymbol{t}}}$:

$$\begin{aligned}\langle \boldsymbol{t}|\rho_{\text{out}}\rangle &= \frac{1}{\prod_i \sqrt{t_i! s_i!}} \sum_{\boldsymbol{j}\in S_{\tilde{\boldsymbol{t}}}} \frac{U_{j_1,\tilde{s}_1}\cdots U_{j_N,\tilde{s}_N}}{\prod_i j_i!} \prod_i j_i! = \frac{1}{\prod_i \sqrt{t! s_i!}} \sum_{\boldsymbol{j}\in S_{\tilde{\boldsymbol{t}}}} \prod_{k=1}^{N} U_{j_k,\tilde{s}_k} \\ &= \frac{1}{\prod_i \sqrt{t! s_i!}} \text{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}}),\end{aligned} \tag{2.87}$$

where Perm is the permanent matrix function and $\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}}$ is the submatrix of $\boldsymbol{U}$ including rows in $\tilde{\boldsymbol{t}}$ (the output photon pattern) and columns in $\tilde{\boldsymbol{s}}$ (the input photon pattern). In the first line, the product of factorials in the numerator comes from evaluating the braket and the same term in the denominator is to account for the inclusion of permutations over repeated values in the sum over the symmetric group. For example, the term corresponding to $j_1 = j_2 = 1$ should only appear once whereas it appears twice in $S_{\tilde{\boldsymbol{t}}}$. The last line simply comes from the definition of the permanent. So the probability of measuring output photon pattern $\boldsymbol{t}$ given a unitary operator $\hat{\mathcal{U}}$ acting on an input photon pattern $\boldsymbol{s}$ is given by

$$\Pr(\boldsymbol{t}|\boldsymbol{s}) = |\,\langle \boldsymbol{t}|\hat{\mathcal{U}}|\boldsymbol{s}\rangle\,|^2 = \frac{|\text{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}})|^2}{\prod_i t_i! s_i!}. \tag{2.88}$$

As an example, we take a simple well-known case of standard boson sampling: Hong-Ou-Mandel interference [52]. In this case there are only two modes with a photon injected into both input modes and interferred on a 50/50 beamsplitter. The input state is given by $|\psi_{\text{in}}\rangle = |1,1\rangle = \hat{a}_1^\dagger \hat{a}_2^\dagger |0,0\rangle$. So $N = M = 2$. The beamsplitter operation is given by the following matrix:

$$\boldsymbol{U}_{\text{BS}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \tag{2.89}$$

and so $\hat{a}_1^\dagger \to \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger - \hat{a}_2^\dagger)$ and $\hat{a}_2^\dagger \to \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger + \hat{a}_2^\dagger)/\sqrt{2}$. The state after the beamsplitter is then given by

$$\begin{aligned}|\psi_{\text{out}}\rangle =& \hat{\mathcal{U}}_{\text{BS}} |\psi_{\text{in}}\rangle = \hat{\mathcal{U}}_{\text{BS}}\, \hat{a}_1^\dagger \hat{a}_2^\dagger |0,0\rangle = \frac{1}{2}(\hat{a}_1^\dagger - \hat{a}_2^\dagger)(\hat{a}_1^\dagger + \hat{a}_2^\dagger)\,|0,0\rangle \\ =& \frac{1}{2}(\hat{a}_1^\dagger \hat{a}_1^\dagger + \hat{a}_1^\dagger \hat{a}_2^\dagger - \hat{a}_2^\dagger \hat{a}_1^\dagger - \hat{a}_2^\dagger \hat{a}_2^\dagger)\,|0,0\rangle = \frac{1}{2}(\hat{a}_1^\dagger \hat{a}_1^\dagger - \hat{a}_2^\dagger \hat{a}_2^\dagger\,|0,0\rangle = \frac{1}{\sqrt{2}}(|2,0\rangle - |0,2\rangle),\end{aligned} \tag{2.90}$$

where in the second line we use the commutativity of $\hat{a}_i^\dagger$ and $\hat{a}_j^\dagger$ if $i \neq j$, and $\hat{a}^\dagger\hat{a}^\dagger |0\rangle = \sqrt{2}\,|2\rangle$. So the state after the interferometer (single beamsplitter) is a superposition of both photons in the first mode and both photons in the second mode. The cancelling of the amplitudes for the coincidence of one photon in each mode is famously evidence for quantum interference as this cancellation wouldn't happen classically when we sum probabilities not amplitudes. For two photons in two modes, there are just three possible output photon patterns: $|2,0\rangle$, $|1,1\rangle$ and $|0,2\rangle$. Due to the orthogonality of Fock states, the probabilities can be found by taking the modulus square of the coefficients, eg. $\Pr(\boldsymbol{t} = (2,0)) = 1/2$.

Below we perform the same calculation except in terms of general $\boldsymbol{U}$ for a 2×2 interferometer to show more clearly that it is an example of the general equation (2.88):

$$\begin{aligned}
|\psi_{\text{out}}\rangle =& \hat{\mathcal{U}}_{\text{BS}}\,|\psi_{\text{in}}\rangle = \hat{\mathcal{U}}_{\text{BS}}\,\hat{a}_1^\dagger\hat{a}_2^\dagger\,|0,0\rangle = \sum_{j_1=1}^{2}\sum_{j_2=1}^{2} U_{j_1,1}U_{j_2,2}\hat{a}_{j_1}^\dagger\,\hat{a}_{j_2}^\dagger\,|0,0\rangle \\
=&(U_{11}U_{12}\hat{a}_1^\dagger\hat{a}_1^\dagger + U_{11}U_{22}\hat{a}_1^\dagger\hat{a}_2^\dagger + U_{12}U_{21}\hat{a}_2^\dagger\hat{a}_1^\dagger + U_{21}U_{22}\hat{a}_2^\dagger\hat{a}_2^\dagger)\,|0,0\rangle \\
=&\sqrt{2}U_{11}U_{12}\,|2,0\rangle + (U_{11}U_{22} + U_{12}U_{21})\,|1,1\rangle + \sqrt{2}U_{21}U_{22}\,|0,2\rangle \\
=&\frac{1}{\sqrt{2}}\text{Perm}(\boldsymbol{U}^{\{(1,1),(1,2)\}})\,|2,0\rangle + \text{Perm}(\boldsymbol{U}^{\{(1,2),(1,2)\}})\,|1,1\rangle + \frac{1}{\sqrt{2}}\text{Perm}(\boldsymbol{U}^{\{(2,2),(1,2)\}})\,|0,2\rangle\,,
\end{aligned} \tag{2.91}$$

where $\boldsymbol{U}^{\{(1,1),(1,2)\}}$ is the submatrix of $\boldsymbol{U}$ with columns 1 and 2, and row 1 repeated twice. Evaluating the permanents of the beamsplitter unitary will give the previous equation as expected. For example, the permanent of the 2×2 matrix with the row U_{11} and U_{12} repeated twice evaluates to 1. In the above form, it is clearer that the probabilities are given by eq. (2.88) as derived.

There is perhaps a more intuitive explanation for the derivation of the permanent in the probability, using comparisons to graph theory. We begin by remembering that in quantum interference we must sum the probability *amplitudes* of all the possible ways of transitioning from the input state to the output state. The probability is then given by taking the modulus square of this. So to find the probability of the measurement outcome $\boldsymbol{t} = (t_1, \ldots, t_M)$ from a Fock state $|\boldsymbol{s}\rangle = |s_1, \ldots, s_M\rangle$ input in an interferometer we consider all the possible ways of this happening. We can map the input photons and output photons to an undirected bipartite graph, with the input photons given by one group of vertices and the output photons by the other. The weight of the edge (V_i, V_j) is the probability amplitude of a photon that begins in input mode V_i ending in output mode V_j. As shown previously, the probability amplitude of beginning in mode i and ending in mode j is simply the element of the unitary matrix U_{ji}. Therefore the adjacency matrix of this graph representing the photons and their transition probabilities is just the unitary matrix representing the unitary evolution applied to the input Fock state, but keeping the columns of the input photons and rows of the output. If the input state $|\boldsymbol{s}\rangle$ is measured in the output state $|\boldsymbol{t}\rangle$, each photon in the input must be matched to one and only one of the photons at the output. This simply gives the path that each photon took, but because we are only considering the probability amplitudes this does not remove the quantum interference. In graph terminology, the matching of all vertices in one group to all vertices in the other is a perfect matching. The aim is to sum over the probability

amplitudes of all the possible matchings, where the probability amplitude of a particular path from input to output is given by the product of the individual weightings between each vertex pairing. So the probability amplitude of the input state being measured in the output state is the sum of the weights of all the perfect matchings, which as shown previously is precisely the definition of the permanent.

2.4.2 Scattershot boson sampling

The experimental challenge to produce single photons on-demand is a potential impediment in the task to realise a demonstration of standard boson sampling with many photons. One option is to repeat the experiment until enough photons are detected. If the probability of a photon being produced in one mode is p, then all N photons are produced with probability p^N. As the probability of all photon sources producing a photon deteriorates exponentially with the number of desired photons, this is not a feasible solution when the number of photons is high. Therefore, the first variant to be suggested was scattershot boson sampling [9]. In this scenario, the input states are two-mode squeezed states, where one mode of each state is directed to detectors to herald the presence of a photon pair and the other mode is fed into an interferometer. The setup is shown in fig. (2.7). Ideal two-mode squeezed states will always produce an even number of photons with n photons in one mode implying there are n photons in the other mode: $|\text{TMSS}\rangle = \sum p_n |n, n\rangle$. In scattershot boson sampling, the input states have low enough squeezing that the probability to produce more than two photons from each two-mode squeezer is negligible and we assume either no photons or one photon in each mode. This means that we do not fix the input state, but because we squeeze all M modes there is a much higher probability of producing N photons, now $\binom{M}{N} p^N (1-p)^{M-N}$.

The input state is not fixed and so when determining the probability of an output pattern $\boldsymbol{t}$ from a scattershot experiment we need to sum over the possible input states:

$$\Pr(\boldsymbol{t}) = \sum_{\boldsymbol{s}} \Pr(\boldsymbol{s}) \Pr(\boldsymbol{t}|\boldsymbol{s}), \tag{2.92}$$

where $\Pr(\boldsymbol{s})$ is the probability of heralding pattern $\boldsymbol{s}$ and $\Pr(\boldsymbol{t}|\boldsymbol{s})$ is the probability of measuring output pattern $\boldsymbol{t}$ conditional on heralding pattern $\boldsymbol{s}$. By heralding on half the modes, the input state collapses to the Fock state with the number of photons in each input mode determined by the number of photons detected in each heralding mode. Therefore the input state is precisely given by the heralded pattern and the probability of the output pattern conditional on the input state is the same as in standard boson sampling with the heralded pattern as the input, eq. (2.88). The probability of measuring a particular heralded pattern is simply the probability of measuring a two-mode squeezed state in the number basis. Therefore the probability of heralding $\boldsymbol{s}$ is the product of the probabilities of s_i photons in each mode of the two-mode squeezed state in modes $2i$ and $2i-1$. The probability of detecting n photons in each mode of a two-mode squeezed state

$\Pr(n)$ is given by

$$\Pr(n) = \frac{\tanh^{2n} r}{\cosh^2 r}, \tag{2.93}$$

for squeezing parameter r. In the case of equal squeezing on all modes, which is typically used, the probabilities of all heralding patterns with the same total number of photons are equal.

2.4.3 Gaussian boson sampling

Building on the use of squeezed states in boson sampling, we now describe a type of boson sampling that uses sampling from Gaussian states at the output of the interferometer, appropriately named Gaussian boson sampling (GBS) [7, 9, 53, 54]. This is contrary to standard boson sampling which inputs a non-Gaussian state, and scattershot in which a Gaussian state is input but the non-Gaussian photon measurement leaves a non-Gaussian state at the output. In GBS, we must input a Gaussian state - a squeezed, coherent, or squeezed coherent state - and use non-Gaussian measurements. For boson sampling to be a hard problem, we need at least one non-Gaussian element: the input state, the evolution or the measurement. Another proposal for GBS is the reverse where we input non-Gaussian states but have Gaussian measurements - namely homodyne or heterodyne measurements [8]. Under time reversal this is equivalent to the case of Gaussian states and non-Gaussian measurements and is much less common in literature. The focus of this thesis is the first proposal.

2.4.3.1 General Gaussian boson sampling

We need to know the probability of an output photon pattern given an input Gaussian state. Here we expand on the derivation outlined in [53]. The general probability of a state with density operator $\hat{\rho}$ being measured in state $|\boldsymbol{t}\rangle$ is given by

$$\Pr(\boldsymbol{t}) = \mathrm{Tr}(\hat{\rho}\Pi) = \mathrm{Tr}\left(\hat{\rho}\,|t_1,\ldots,t_M\rangle\,\langle t_1,\ldots,t_M|\right) = \mathrm{Tr}(\hat{\rho}\,|\boldsymbol{t}\rangle\,\langle\boldsymbol{t}|). \tag{2.94}$$

Any state can be represented by the Glauber-Sudarshan P representation which is the decomposition in terms of the coherent states [34, 35],

$$\hat{\rho} = \int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha} P_\rho(\boldsymbol{\alpha})\,|\boldsymbol{\alpha}\rangle\,\langle\boldsymbol{\alpha}|\,, \tag{2.95}$$

where P_ρ is the P-function, introduced in section (2.2.1). We can use the P-function representation of the number state projector to write the probability of an output photon pattern in terms of the phase-space representation [55]:

$$\begin{aligned}\Pr(\boldsymbol{t}) &= \mathrm{Tr}(\hat{\rho}\,|\boldsymbol{t}\rangle\,\langle\boldsymbol{t}|) = \mathrm{Tr}\left(\hat{\rho}\int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha} P_{\boldsymbol{t},\boldsymbol{t}}(\boldsymbol{\alpha})\,|\boldsymbol{\alpha}\rangle\,\langle\boldsymbol{\alpha}|\right) = \int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha} P_{\boldsymbol{t},\boldsymbol{t}}(\boldsymbol{\alpha})\,\langle\boldsymbol{\alpha}|\hat{\rho}|\boldsymbol{\alpha}\rangle \\ &= \pi^M \int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha} P_{\boldsymbol{t},\boldsymbol{t}}(\boldsymbol{\alpha})\frac{\langle\boldsymbol{\alpha}|\hat{\rho}|\boldsymbol{\alpha}\rangle}{\pi^M} = \pi^M \int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha} P_{\boldsymbol{t},\boldsymbol{t}}(\boldsymbol{\alpha}) Q_\rho,\end{aligned} \tag{2.96}$$

where Q_ρ is the Q-function of a state, introduced in section (2.2.1). The P-function for the number state projection operator is given by [37]

$$P_{n,n}(\alpha) = \frac{\mathrm{e}^{|\alpha|^2}}{n!}\partial_\alpha^n \partial_{\alpha^*}^n \delta(\mathrm{Re}(\alpha))\delta(\mathrm{Im}(\alpha)). \tag{2.97}$$

We confirm that this does give the expected operator $|n\rangle\langle n|$ here. First we expand the terms inside the integral:

$$\begin{aligned} P_{n,n}(\alpha)|\alpha\rangle\langle\alpha| &= \frac{\mathrm{e}^{|\alpha|^2}}{n!}\partial_\alpha^n \partial_{\alpha^*}^n \delta(\mathrm{Re}(\alpha))\delta(\mathrm{Im}(\alpha)) \left(\mathrm{e}^{-\frac{|\alpha|^2}{2}} \sum_{n_1=0}^{\infty} \frac{\alpha^{n_1}}{\sqrt{n_1!}}|n_1\rangle\right)\left(\mathrm{e}^{-\frac{|\alpha|^2}{2}} \sum_{n_2=0}^{\infty} \frac{(\alpha^*)^{n_2}}{\sqrt{n_2!}}\langle n_2|\right) \\ &= \delta(\mathrm{Re}(\alpha))\delta(\mathrm{Im}(\alpha))\frac{1}{n!}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\partial_\alpha^n\partial_{\alpha^*}^n \frac{\alpha^{n_1}}{\sqrt{n_1!}}\frac{(\alpha^*)^{n_2}}{\sqrt{n_2!}}|n_1\rangle\langle n_2| \\ &= \delta(\mathrm{Re}(\alpha))\delta(\mathrm{Im}(\alpha))\frac{1}{n!}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\frac{n_1!\alpha^{n_1-n}}{(n_1-n)!\sqrt{n_1!}}\frac{n_2!(\alpha^*)^{n_2-n}}{(n_2-n)!\sqrt{n_2!}}|n_1\rangle\langle n_2|. \end{aligned} \tag{2.98}$$

When integrating over all α, due to the delta function only the terms where $\alpha = 0$ are non-zero. It is clear to see that the only terms inside the sum that are non-zero for $\alpha = 0$ are when $n_1 - n = 0$ and $n_2 - n = 0$. Therefore after integrating we are left with

$$\int P_{n,n}(\alpha)|\alpha\rangle\langle\alpha| = \frac{1}{n!}\frac{n!}{\sqrt{n!}}\frac{n!}{\sqrt{n!}}|n\rangle\langle n| = |n\rangle\langle n|, \tag{2.99}$$

as expected. This P-function can be generalised in the case that the state is multimode by taking the product of each mode:

$$\hat{\rho} = \int_{\boldsymbol{\alpha}} P_\rho(\boldsymbol{\alpha})|\boldsymbol{\alpha}\rangle\langle\boldsymbol{\alpha}|, \tag{2.100}$$

where

$$P_\rho(\boldsymbol{\alpha}) = \prod_{i=1}^{M}\frac{\mathrm{e}^{|\alpha_i|^2}}{n_i!}\partial_{\alpha_i}^{n_i}\partial_{\alpha_i^*}^{n_i}\delta(\mathrm{Re}(\alpha_i))\delta(\mathrm{Im}(\alpha_i)) = \mathrm{e}^{\frac{1}{2}|\boldsymbol{\zeta}|^2}\prod_{i=1}^{M}\frac{1}{n_i!}\partial_{\alpha_i}^{n_i}\partial_{\alpha_i^*}^{n_i}\delta(\mathrm{Re}(\alpha_i))\delta(\mathrm{Im}(\alpha_i)). \tag{2.101}$$

The above equivalence is true due to the symmetry of $\boldsymbol{\zeta} = \begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\alpha}^*\end{pmatrix}$ giving $|\boldsymbol{\zeta}|^2 = \boldsymbol{\zeta}^\dagger\boldsymbol{\zeta} = \sum_{i=1}^{2M}|\boldsymbol{\zeta}_i|^2 = \sum_{i=1}^{M}2|\boldsymbol{\zeta}_i|^2 = \sum_{i=1}^{M}2|\boldsymbol{\alpha}_i|^2$. By changing eq. (2.27) into the ladder operator basis and swapping $\boldsymbol{V}$ with $\boldsymbol{V}_Q$, we can see the Q-function of a Gaussian state is given by

$$Q_\rho = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{\zeta}-\bar{\boldsymbol{\zeta}})^\dagger\boldsymbol{\Sigma}_Q^{-1}(\boldsymbol{\zeta}-\bar{\boldsymbol{\zeta}})\right)}{\sqrt{\det(\pi\boldsymbol{\Sigma}_Q)}}. \tag{2.102}$$

We substitute eq. (2.101) and eq. (2.102) into eq. (2.96):

$$\begin{aligned}\Pr(\boldsymbol{t}) &= \pi^M \int_{\boldsymbol{\alpha}} d^2\boldsymbol{\alpha}\, \mathrm{e}^{\frac{1}{2}|\boldsymbol{\zeta}|^2} \prod_{i=1}^{M} \left[\frac{1}{t_i!}\partial_{\alpha_i}^{t_i}\partial_{\alpha_i^*}^{t_i}\delta(\alpha)\right] \frac{\exp\left(-\frac{1}{2}(\boldsymbol{\zeta}-\bar{\boldsymbol{\zeta}})^\dagger \boldsymbol{\Sigma}_Q^{-1}(\boldsymbol{\zeta}-\bar{\boldsymbol{\zeta}})\right)}{\sqrt{\det(\pi\boldsymbol{\Sigma}_Q)}} \\ &= \prod_{i=1}^{M} \frac{1}{t_i!}\partial_{\alpha_i}^{t_i}\partial_{\alpha_i^*}^{t_i} \frac{\exp\left(-\frac{1}{2}\boldsymbol{\zeta}^\dagger(\boldsymbol{\Sigma}_Q^{-1}-\boldsymbol{I})\boldsymbol{\zeta}\right)\exp\left(\frac{1}{2}(\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}+\bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\boldsymbol{\zeta})\right)\exp\left(-\frac{1}{2}\bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}}\Bigg|_{\boldsymbol{\zeta}=0} \\ &= \frac{\exp\left(-\frac{1}{2}\bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \prod_{i=1}^{M} \frac{1}{t_i!}\partial_{\alpha_i}^{t_i}\partial_{\alpha_i^*}^{t_i} \exp\left(-\frac{1}{2}\boldsymbol{\zeta}^\dagger(\boldsymbol{\Sigma}_Q^{-1}-\boldsymbol{I})\boldsymbol{\zeta}\right)\exp\left(\frac{1}{2}(\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}+\bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\boldsymbol{\zeta})\right)\Bigg|_{\boldsymbol{\zeta}=0}.\end{aligned} \tag{2.103}$$

We introduce a permutation matrix

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{pmatrix} \tag{2.104}$$

which when applied to block matrices/vectors from the left swaps the top and bottom blocks and when applied from the right swaps the left and right blocks. It is clear from the block structure of $\boldsymbol{\zeta} = \binom{\alpha}{\alpha^*}$ that applying $\boldsymbol{X}$ to the left of this column vector or the right of the row vector (its transpose) is the same as taking its complex conjugate: $\boldsymbol{X}\boldsymbol{\zeta} = \boldsymbol{\zeta}^*$. Therefore in the first exponential term in the equation above the $\boldsymbol{\zeta}^\dagger$ can be replaced by $\boldsymbol{\zeta}^\top\boldsymbol{X}$ which simplifies the expression. The final exponential term can also be simplified. The term $\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}$ is a scalar and so must be equal to its transpose: $\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}} = \bar{\boldsymbol{\zeta}}^\top(\boldsymbol{\Sigma}_Q^{-1})^\top\boldsymbol{\zeta}^*$. Again using the permutation matrix, we convert to using $\boldsymbol{\zeta}^\dagger$ and $\boldsymbol{\zeta}$: $\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}} = \bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{X}(\boldsymbol{\Sigma}_Q^{-1})^\top\boldsymbol{X}\boldsymbol{\zeta}$. So both terms in the exponential function are equal if $\boldsymbol{\Sigma}_Q^{-1} = \boldsymbol{X}(\boldsymbol{\Sigma}_Q^{-1})^\top\boldsymbol{X}$. We note that the permutation matrix is symmetric, Hermitian and its own inverse. By simply using these properties and changing the order of the inverse and transposition $\boldsymbol{X}(\boldsymbol{\Sigma}_Q^{-1})^\top\boldsymbol{X} = (\boldsymbol{X}\boldsymbol{\Sigma}_Q^\top\boldsymbol{X})^{-1} = ((\boldsymbol{X}\boldsymbol{\Sigma}_Q\boldsymbol{X})^\top)^{-1}$. We can map to the real covariance matrix $\boldsymbol{V}$ to find $\boldsymbol{X}\boldsymbol{\Sigma}_Q\boldsymbol{X} = \boldsymbol{X}\boldsymbol{F}(\boldsymbol{V}+\frac{1}{2}\boldsymbol{I})\boldsymbol{F}^\dagger\boldsymbol{X} = \boldsymbol{F}^*(\boldsymbol{V}+\frac{1}{2}\boldsymbol{I})\boldsymbol{F}^\top = (\boldsymbol{F}(\boldsymbol{V}+\frac{1}{2}\boldsymbol{I})\boldsymbol{F}^\dagger)^\top = \boldsymbol{\Sigma}_Q^\top$, where we used the property that the real covariance matrix is symmetric. So we have shown that $\boldsymbol{\zeta}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}} = \bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\boldsymbol{\zeta}$. Therefore the probability can be simplified to

$$\Pr(\boldsymbol{t}) = \frac{\exp\left(-\frac{1}{2}\bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \prod_{i=1}^{M} \frac{1}{t_i!}\partial_{\alpha_i}^{t_i}\partial_{\alpha_i^*}^{t_i} \exp\left(\frac{1}{2}\boldsymbol{\zeta}^\top\boldsymbol{A}\boldsymbol{\zeta}\right)\exp(\boldsymbol{\gamma}\boldsymbol{\zeta})\Bigg|_{\boldsymbol{\zeta}=0} \tag{2.105}$$

where we introduce $\boldsymbol{A} = \boldsymbol{X}(\boldsymbol{I}-\boldsymbol{\Sigma}_Q^{-1})$ and $\boldsymbol{\gamma} = \bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}$.

We use Fáa di Bruno's formula [56] to evaluate the multidimensional partial derivatives. From this higher-order chain rule, the derivative of a Gaussian function is generally given by [57]

$$\prod_{i=1}^{n}\partial_{x_i}\exp(y) = \exp(y)\sum_{\pi}\prod_{B\in\pi}\left(\prod_{j\in B}\partial_{x_j}\right)y, \tag{2.106}$$

where π is all the partitions of the set $\{x_1,\ldots,x_n\}$ and B is all the subsets of the partition. For example, if $n=5$, one such partition in π is $\{x_1\},\{x_2,x_4\},\{x_3,x_5\}$ with B being the three subsets

$B_1 = \{x_1\}$, $B_2 = \{x_2, x_4\}$ and $B_3 = \{x_3, x_5\}$. This can be applied to the previous equation where we have $y = \frac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta}$, and the product is over $\boldsymbol{\zeta}_i$. One thing to be cautious of is that there are powers of derivatives according to the output photon pattern. This can be circumvented by defining a new vector $\tilde{\boldsymbol{\zeta}} = \boldsymbol{\zeta}^{\{\boldsymbol{t}\}}$ which is the vector with each element ζ_i and ζ_{i+M} added t_i times. To be more precise we can multiply the vector of displacements by a matrix $\tilde{\boldsymbol{P}}$ which is similar to a permutation matrix but does not necessarily conserve the number of occurrences of each element: $\tilde{\boldsymbol{\zeta}} = \tilde{\boldsymbol{P}}\boldsymbol{\zeta}$. It is convenient to introduce the $2M$-length vector $\boldsymbol{t}' = (\boldsymbol{t}, \boldsymbol{t})$ so that the output pattern in each mode is associated with both α and α^* in $\boldsymbol{\zeta}$. This way the product of partial derivatives becomes $\prod_{i=1}^{M} \partial_{\alpha_i}^{t_i} \partial_{\alpha_i^*}^{t_i} = \prod_{i=1}^{2M} \partial_{\zeta_i}^{t'_i} = \prod_{n=1}^{2N} \partial_{\tilde{\boldsymbol{\zeta}}_n}$. For example, if the output pattern of three photons in four modes is $\boldsymbol{t} = (2, 1, 0, 0)$, the 'permuted' vector would be given by $\tilde{\boldsymbol{\zeta}} = (\zeta_1, \zeta_1, \zeta_2, \zeta_{1+M}, \zeta_{1+M}, \zeta_{2+M})^\top = \boldsymbol{P}\boldsymbol{\zeta} = (\tilde{\boldsymbol{P}}_s \oplus \tilde{\boldsymbol{P}}_s)\boldsymbol{\zeta}$ where

$$\tilde{\boldsymbol{P}}_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{2.107}$$

and the product of partial derivatives would be $\partial_{\zeta_1}^2 \partial_{\zeta_2} \partial_{\zeta_{1+M}}^2 \partial_{\zeta_{2+M}} = \partial_{\zeta_1} \partial_{\zeta_1} \partial_{\zeta_2} \partial_{\zeta_{1+M}} \partial_{\zeta_{1+M}} \partial_{\zeta_{2+M}} = \partial_{\tilde{\zeta}_1} \partial_{\tilde{\zeta}_2} \partial_{\tilde{\zeta}_3} \partial_{\tilde{\zeta}_4} \partial_{\tilde{\zeta}_5} \partial_{\tilde{\zeta}_6}$. So the quantity of interest can be written as

$$\begin{aligned} \prod_{i=1}^{2M} \partial_{\zeta_i}^{t'_i} \exp\left(\tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta}\right) &= \prod_{n=1}^{2N} \partial_{\tilde{\zeta}_n} \exp\left(\tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta}\right) \\ &= \exp\left(\tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta}\right) \sum_{\pi} \prod_{B \in \pi} \left(\prod_{j \in B} \partial_{\tilde{\zeta}_j} \right) \tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta}. \end{aligned} \tag{2.108}$$

We note that the quadratic term $\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} = \sum_{i,j} \zeta_i A_{i,j} \zeta_j$ consists of a sum of terms where the elements in ζ_i appear in pairs only. Similarly, the linear term $\boldsymbol{\gamma}\boldsymbol{\zeta} = \sum \gamma_i \zeta_i$ is a sum of terms where the order of ζ_i is 1. Therefore, when performing the partial derivatives and setting $\boldsymbol{\zeta} = \boldsymbol{0}$, only those with a second partial derivative or first partial derivative of elements in $\tilde{\boldsymbol{\zeta}}$ are non-zero for the quadratic and linear terms respectively. This means that the sum is reduced to only be over partitions of the elements in $\tilde{\boldsymbol{\zeta}}$ with one or two elements in each subset. Performing the derivatives results in

$$\left(\prod_{j \in B} \partial_{\tilde{\zeta}_j} \right) \tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{A}\boldsymbol{\zeta} + \boldsymbol{\gamma}\boldsymbol{\zeta} \Big|_{\boldsymbol{\zeta}=\boldsymbol{0}} = \begin{cases} A_{j_1,j_2} & \text{if } |B| = 2 \\ \gamma_j & \text{if } |B| = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.109}$$

where $|B|$ is the number of elements in B.

First, consider the case when $\boldsymbol{\gamma} = \boldsymbol{0}$. This corresponds to a state with a zero vector of means (no displacement). The non-zero partitions are when each subset is of size 2, which is equivalent to all the ways of pairing the elements in the group. As shown earlier, all the perfect matchings between vertices is given by the hafnian of the adjacency matrix. In this case, where B is a partition of $2N$ elements, the vertices are given by the modes where photons were detected, as according to $\tilde{\zeta}_j$.

Defining the vector $\tilde{\boldsymbol{t}}' = (\tilde{\boldsymbol{t}}, \tilde{\boldsymbol{t}} + M)$, where $\tilde{\boldsymbol{t}} + M$ means adding M to each element in the vector $\tilde{\boldsymbol{t}}$, the weighting between vertices i and j is given by $A_{\tilde{t}'_i, \tilde{t}'_j}$. Therefore the adjacency matrix is the matrix $\boldsymbol{A}$ with each row and column appearing as many times as they appear in $\tilde{\boldsymbol{t}}'$, which we denote as $\boldsymbol{A}^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}}$. More precisely this matrix is found as $\boldsymbol{A}^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}} = \tilde{\boldsymbol{P}} \boldsymbol{A} \tilde{\boldsymbol{P}}^\top$. Hence, for a state with no displacement, the probability of measuring a pattern $\boldsymbol{t}$ at the output is given by

$$\Pr(\boldsymbol{t}) = \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \frac{\mathrm{Haf}(\boldsymbol{A}^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}})}{\prod t_i!}, \tag{2.110}$$

where Haf denotes the hafnian.

Now consider the more general case where the vector of means is non-zero. Here $\boldsymbol{\gamma}$ is not zero and so the partitions that contribute to the sum after taking the partial derivatives and setting $\boldsymbol{\zeta} = \boldsymbol{0}$ include those with subsets of size 1 as well as 2. As we saw previously, in graphs where loops are allowed, an edge can exist between two different vertices or between only one vertex (a loop). Therefore a perfect matching in a graph that allows loops is precisely the sum over all partitions into groups of size 1 or 2. From eq. (2.109), when the size of a group is 2, the weighting is given by A_{j_1, j_2} which are the off-diagonal terms, whereas when the size of the group is 1, the weighting is given by γ_j which are assigned to the loops corresponding to the diagonal terms in the adjacency matrix. Therefore for a general state with non-zero displacement, the probability of measuring a pattern $\boldsymbol{t}$ at the output is given by

$$\Pr(\boldsymbol{t}) = \frac{\exp\left(\bar{\boldsymbol{\zeta}}^\dagger \boldsymbol{\Sigma}_Q^{-1} \bar{\boldsymbol{\zeta}}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \frac{\mathrm{Lhaf}(\boldsymbol{\gamma}^{\{\tilde{\boldsymbol{t}}'\}}, \boldsymbol{A}^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}})}{\prod t_i!}, \tag{2.111}$$

where $(\boldsymbol{\gamma}^{\{\tilde{\boldsymbol{t}}'\}}, \boldsymbol{A}^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}})$ is the matrix $A^{\{\tilde{\boldsymbol{t}}', \tilde{\boldsymbol{t}}'\}}$ with the diagonal elements replaced by $\boldsymbol{\gamma}^{\{\tilde{\boldsymbol{t}}'\}}$ and Lhaf denotes the loop hafnian.

The above is the derivation of the formula for general states as shown in [53], but in the particular case when the state is pure the formula can be simplified. The $2N \times 2N$ (loop) hafnian can be simplified as the product of two $N \times N$ (loop) hafnians. Note that the matrix $\boldsymbol{A}$ only depends on the covariance matrix which does not depend on the displacement. We analyse the form of matrix $\boldsymbol{A}$ for an M-mode squeezed state with squeezing $r_m \mathrm{e}^{i\phi_m}$ on mode m followed by a unitary evolution $\boldsymbol{U}$. The covariance matrix for this state is given by

$$\boldsymbol{\Sigma} = \boldsymbol{M}_U \boldsymbol{M}_S \boldsymbol{\Sigma}_{\mathrm{vac}} \boldsymbol{M}_S^\dagger \boldsymbol{M}_U^\dagger, \tag{2.112}$$

where

$$\boldsymbol{M}_U = \begin{pmatrix} \boldsymbol{U} & 0 \\ 0 & \boldsymbol{U}^* \end{pmatrix}, \qquad \boldsymbol{M}_S = \begin{pmatrix} \oplus_m \cosh r_m & \oplus_m - \mathrm{e}^{i\phi_m} \sinh r_m \\ \oplus_m - \mathrm{e}^{-i\phi_m} \sinh r_m & \oplus_m \cosh r_m \end{pmatrix}. \tag{2.113}$$

Multiplying out these matrices, the covariance matrix is written as

$$\boldsymbol{\Sigma} = \frac{1}{2}\begin{pmatrix} \boldsymbol{U} \oplus_m \cosh(2r_m)\boldsymbol{U}^\dagger & -\boldsymbol{U} \oplus_m \mathrm{e}^{i\phi_m}\sinh(2r_m)\boldsymbol{U}^\top \\ -\boldsymbol{U}^* \oplus_m \mathrm{e}^{-i\phi_m}\sinh(2r_m)\boldsymbol{U}^\dagger & \boldsymbol{U}^* \oplus_m \cosh(2r_m)\boldsymbol{U}^\top \end{pmatrix}, \tag{2.114}$$

where $\oplus_m$ is the direct sum resulting in an $M \times M$ diagonal matrix. We need to find the inverse of $\boldsymbol{\Sigma}_Q = \boldsymbol{\Sigma} + \frac{1}{2}\boldsymbol{I}$, so we use the general formula for the inverse of block matrices. Given a general block matrix, its inverse is given by

$$\begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{pmatrix}^{-1} = \begin{pmatrix} (\boldsymbol{a}-\boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c})^{-1} & -\boldsymbol{a}^{-1}\boldsymbol{b}(\boldsymbol{d}-\boldsymbol{c}\boldsymbol{a}^{-1}\boldsymbol{b})^{-1} \\ -\boldsymbol{d}^{-1}\boldsymbol{c}(\boldsymbol{a}-\boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c})^{-1} & (\boldsymbol{d}-\boldsymbol{c}\boldsymbol{a}^{-1}\boldsymbol{b})^{-1} \end{pmatrix}. \tag{2.115}$$

In our case $\boldsymbol{a} = \frac{1}{2}\boldsymbol{U} \oplus_m (\cosh(2r_m)+1)\boldsymbol{U}^\dagger$ which has inverse $2\boldsymbol{U} \oplus_m 1/(\cosh(2r_m)+1)\boldsymbol{U}^\dagger$, and similarly $\boldsymbol{d} = \frac{1}{2}\boldsymbol{U}^* \oplus_m (\cosh(2r_m)+1)\boldsymbol{U}^\top$ has inverse $2\boldsymbol{U}^* \oplus_m 1/(\cosh(2r_m)+1)\boldsymbol{U}^\top$. Substituting these into the general equation for the inverse gives

$$\boldsymbol{\Sigma}_Q^{-1} = \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top \\ -\boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger & \boldsymbol{I} \end{pmatrix}, \tag{2.116}$$

where for clarity we have defined the diagonal matrix $\boldsymbol{R} = \oplus_m(-\mathrm{e}^{i\phi_m}\tanh r_m)$. From here it is easy to see the form of the matrix $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{X}(\boldsymbol{I} - \boldsymbol{\Sigma}_Q^{-1}) = \begin{pmatrix} \boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top \end{pmatrix}, \tag{2.117}$$

which is a block diagonal matrix with the blocks being the complex conjugate of each other and so we can write this as $\boldsymbol{A} = \boldsymbol{B}^* \oplus \boldsymbol{B}$, where $\boldsymbol{B} = \boldsymbol{U} \oplus_m (-\mathrm{e}^{i\phi_m}\tanh r_m)\boldsymbol{U}^\top$ and use the property of hafnians (eq. 2.83) to find $\mathrm{Haf}(\boldsymbol{A}) = |\mathrm{Haf}(\boldsymbol{B})|^2$.

In the case of non-zero displacement, we also need to replace the diagonal with $\boldsymbol{\gamma} = \bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1}$. Note that $\bar{\boldsymbol{\zeta}}$ is a function of the vector of means *after* the interferometer $\boldsymbol{\beta} = \boldsymbol{U}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the displacement in the input states. We observe that this vector has a block structure such that $\boldsymbol{\gamma} = (\tilde{\gamma}, \tilde{\gamma}^*)$:

$$\begin{aligned} \boldsymbol{\gamma} =& \bar{\boldsymbol{\zeta}}^\dagger\boldsymbol{\Sigma}_Q^{-1} = (\boldsymbol{\beta}^\dagger, \boldsymbol{\beta}^\top)\begin{pmatrix} \boldsymbol{I} & -\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top \\ -\boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger & \boldsymbol{I} \end{pmatrix} \\ &= \left(\boldsymbol{\beta}^\dagger - \boldsymbol{\beta}^\top\boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger, \boldsymbol{\beta}^\top - \boldsymbol{\beta}^\dagger\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top\right) \\ &= (\boldsymbol{\alpha}^\dagger\boldsymbol{U}^\dagger - \boldsymbol{\alpha}^\top\boldsymbol{R}^*\boldsymbol{U}^\dagger, \boldsymbol{\alpha}^\top\boldsymbol{U}^\top - \boldsymbol{\alpha}^\dagger\boldsymbol{R}\boldsymbol{U}^\top) \\ &= \begin{pmatrix} \boldsymbol{U}^*[\boldsymbol{\alpha}^* - \boldsymbol{R}^\dagger\boldsymbol{\alpha}] \\ \boldsymbol{U}[\boldsymbol{\alpha} - \boldsymbol{R}^\top\boldsymbol{\alpha}^*] \end{pmatrix}^\top. \end{aligned} \tag{2.118}$$

Therefore, the matrix in the loop hafnian, $(\boldsymbol{\gamma}, \boldsymbol{A})$ is also a block diagonal matrix and the loop hafnian can be simplified.

We can also simplify the prefactors in the probability expression for pure states. First we find

$\det(\boldsymbol{\Sigma}_Q)$. Again we use the general formula for block matrices:

$$\det\begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{pmatrix} = \det(\boldsymbol{d})\det(\boldsymbol{a} - \boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c}). \tag{2.119}$$

From eq. (2.114) it can be seen that for the covariance matrix $\boldsymbol{\Sigma}_Q = \boldsymbol{\Sigma} + \frac{1}{2}\boldsymbol{I}$, the blocks satisfy $\boldsymbol{a} - \boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c} = \boldsymbol{I}$ which has determinant 1. So we only need to find the determinant of $\boldsymbol{d} = \frac{1}{2}\boldsymbol{U}^* \oplus_m [\cosh(2r_m) + 1]\boldsymbol{U}^\top$. We use the multiplicative property of determinants to find $\det(\frac{1}{2}\boldsymbol{U}^* \oplus_m [\cosh(2r_m)+1]\boldsymbol{U}^\top) = \det(\frac{1}{2} \oplus_m [\cosh(2r_m)+1]\boldsymbol{U}^\top\boldsymbol{U}^*) = \frac{1}{2^M}\det(\oplus_m[\cosh(2r_m)+1])$. This is a diagonal matrix which has a determinant given by the product of the entries of the matrix. Hence $\det(\boldsymbol{\Sigma}_Q) = \frac{1}{2^M}\prod \cosh(2r_m) + 1$.

Finally, we need to determine the prefactor for the non-zero displacement case: $\exp(\bar{\boldsymbol{\zeta}}^\dagger \boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}})$. From above we have

$$\begin{aligned} \bar{\boldsymbol{\zeta}}^\dagger \boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}} &= (\boldsymbol{\beta}^\dagger, \boldsymbol{\beta}^\top)\begin{pmatrix} \boldsymbol{I} & -\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top \\ -\boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger & \boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\beta}^* \end{pmatrix} \\ &= 2|\boldsymbol{\beta}|^2 - \boldsymbol{\beta}^\top\boldsymbol{U}^*\boldsymbol{R}^*\boldsymbol{U}^\dagger\boldsymbol{\beta} - \boldsymbol{\beta}^\dagger\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top\boldsymbol{\beta}^* \\ &= 2(|\boldsymbol{\alpha}|^2 - \mathrm{Re}(\boldsymbol{\beta}^\dagger\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top\boldsymbol{\beta}^*)) \\ &= 2(|\boldsymbol{\alpha}|^2 - \mathrm{Re}(\boldsymbol{\alpha}^\dagger\boldsymbol{R}\boldsymbol{\alpha}^*)). \end{aligned} \tag{2.120}$$

So for the case of input squeezed displaced states with squeezing $r_m e^{i\phi_m}$ and displacement α_m in mode m passing through a unitary $\boldsymbol{U}$ measured with photon number resolving detectors, the probability of measuring pattern $\boldsymbol{t}$ can be written as

$$\Pr(\boldsymbol{t}) = \frac{\exp(|\boldsymbol{\alpha}|^2 - \mathrm{Re}(\boldsymbol{\alpha}^\dagger\boldsymbol{R}\boldsymbol{\alpha}^*))}{\prod_m \cosh^2 r_m}\frac{|\mathrm{Lhaf}((\boldsymbol{U}[\boldsymbol{\alpha} - \boldsymbol{R}^\top\boldsymbol{\alpha}^*])^{\{\tilde{\boldsymbol{t}}\}}, (\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}})|^2}{\prod_m t_m!}. \tag{2.121}$$

In the case of no displacement, the above equation reduces to

$$\Pr(\boldsymbol{t}) = \frac{1}{\prod_m \cosh^2 r_m}\frac{|\mathrm{Haf}((\boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}})|^2}{\prod_m t_m!}. \tag{2.122}$$

2.4.3.2 Reduction to scattershot boson sampling

Scattershot boson sampling is a particular case of GBS and as such it is possible to retrieve the expression for the probabilities in scattershot boson sampling from the general equations for GBS. In scattershot boson sampling, the input states are two-mode squeezed states but where one mode in each is used for heralding and typically not included as individual modes. However, to map it to a GBS set-up these heralding modes must be taken into account and so the number of modes doubles with half of the modes passing through the identity transformation. This mapping is shown in figs. (2.7a) and (2.7b). The layout here also involves swap gates to move all the heralding modes together such that it is easier to show that half of the modes undergo only the identity transformation and the other half pass through the desired scattershot unitary. To fully map to general GBS, the

input states should be single-mode squeezed states. A simple unitary transformation can be applied to two single-mode squeezed states to output a two-mode squeezed state: $|\text{TMSS}\rangle = \hat{U}_{\text{BS}}\hat{U}_{\text{PS}}\,|\xi,\xi\rangle$, where the corresponding matrices

$$\boldsymbol{U}_{\text{BS}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \qquad \boldsymbol{U}_{\text{PS}} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{2.123}$$

are the unitaries for a beamsplitter and phase shifter respectively. This is the application of a phase shifter on the lower mode followed by a 50/50 beamsplitter. Applying this transformation on each of the $2M$ modes and then applying the swap gates is depicted in fig. (2.7c) and that transformation is given by

$$\boldsymbol{U}_{\text{TMS}} = \frac{1}{\sqrt{2}}\left(\begin{array}{cccccccccc} 1 & i & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & i & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & i & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & & \vdots & \ddots & 1 & i \\ \hline -1 & i & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & i & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & i & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & & \vdots & \ddots & -1 & i \end{array}\right). \tag{2.124}$$

Therefore the whole unitary for the GBS description is given by $\boldsymbol{U}_{\text{GBS}} = (\boldsymbol{I} \oplus \boldsymbol{U})\boldsymbol{U}_{\text{TMS}}$, which can now be substituted into the general equation for input states with no displacement (eq. 2.122).

In scattershot boson sampling, the input states all have equal squeezing $\xi = r\mathrm{e}^{i\phi}$ and so the matrix $\boldsymbol{R}$ is a multiple of the identity ($\boldsymbol{R} = -\mathrm{e}^{i\phi}\tanh r\boldsymbol{I}$) and can be taken out of the hafnian

$$\Pr(\boldsymbol{t}) = \frac{\tanh^N r}{\cosh^{2M} r}\frac{|\text{Haf}((\boldsymbol{U}_{\text{GBS}}\boldsymbol{U}_{\text{GBS}}^{\top})^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}})|^2}{\prod_m t_m!}. \tag{2.125}$$

Using the fact that $\boldsymbol{U}_{\text{TMS}}\boldsymbol{U}_{\text{TMS}}^{\top} = -\boldsymbol{X}$, we find

$$\boldsymbol{U}_{\text{GBS}}\boldsymbol{U}_{\text{GBS}}^{\top} = -\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U} \end{pmatrix}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}^{\top} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{0} & \boldsymbol{U}^{\top} \\ \boldsymbol{U} & \boldsymbol{0} \end{pmatrix}. \tag{2.126}$$

If we split the output photon pattern into the heralding pattern and the desired pattern $\boldsymbol{t} = (\boldsymbol{t}_h, \boldsymbol{t}_d)$, the submatrix for which the hafnian is calculated is given by

$$(\boldsymbol{U}_{\text{GBS}}\boldsymbol{U}_{\text{GBS}}^{\top})^{\{(\tilde{\boldsymbol{t}}_h,\tilde{\boldsymbol{t}}_d),(\tilde{\boldsymbol{t}}_h,\tilde{\boldsymbol{t}}_d)\}} = -\begin{pmatrix} \boldsymbol{0}^{\{\tilde{\boldsymbol{t}}_h,\tilde{\boldsymbol{t}}_h\}} & \boldsymbol{U}^{\{\tilde{\boldsymbol{t}}_d,\tilde{\boldsymbol{t}}_h\}^{\top}} \\ \boldsymbol{U}^{\{\tilde{\boldsymbol{t}}_d,\tilde{\boldsymbol{t}}_h\}} & \boldsymbol{0}^{\{\tilde{\boldsymbol{t}}_d,\tilde{\boldsymbol{t}}_d\}} \end{pmatrix}, \tag{2.127}$$

where the superscript $\{\tilde{\boldsymbol{t}}_d, \tilde{\boldsymbol{t}}_h\}$ means to take rows in $\tilde{\boldsymbol{t}}_d$ and columns in $\tilde{\boldsymbol{t}}_h$. Using the property of

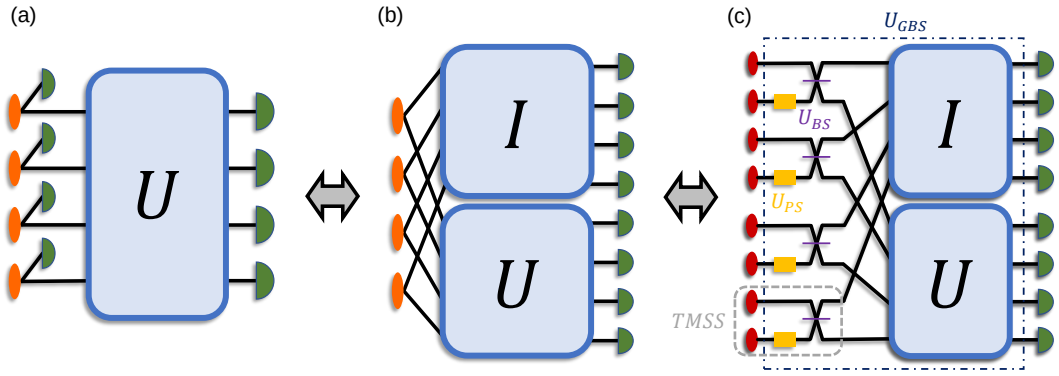

Figure 2.7: Scattershot and GBS equivalence. In (a) a usual set-up for scattershot boson sampling is depicted with input two-mode squeezed states where one mode is heralded and the other feeds into an interferometer. The mathematical equivalence of this is shown in (b), where the heralded photons can be considered to pass through the identity and detected exactly as the other photons are. To separate the modes such that the heralded modes are together and the others are grouped, the modes need to pass through swap gates. The two-mode squeezed states can also be described by two single-mode squeezed states passing through a phase shifter and beamsplitter. This is indicated by the TMSS box in (c). Putting together the phase shifters, beamsplitters, swap gates and identity interferometer, the unitary required to describe this set-up in the GBS picture is indicated by U_{GBS}.

hafnians, this reduces to a permanent and as expected the probability is given by

$$\Pr(\boldsymbol{t}) = \frac{\tanh^N r}{\prod_m \cosh^2 r_m} \frac{|\text{Perm}((\boldsymbol{U})^{\{\tilde{\boldsymbol{t}}_d, \tilde{\boldsymbol{t}}_h\}})|^2}{\prod_m t_m!}, \tag{2.128}$$

with the columns chosen according to the heralding pattern (which is equivalent to the input state in standard boson sampling) and the rows by the output pattern.

2.4.3.3 Retrodictive intuition behind Gaussian boson sampling

We provide an intuitive explanation behind the matrices that are found in the derivation of the probabilities for GBS. To do this we consider the probabilities in the retrodictive picture [58]. In the predictive picture, the probability of an outcome from a measurement on a state is known. The retrodictive probability is the reverse where given a measurement outcome, we wish to describe the state that was measured. This approach was applied when simulating boson sampling with an arbitrary input state with a Gaussian measurement performed [8, 59]. In this way it is possible to consider the input states as equivalent to the measurement basis and the detection events as the source. For example, in standard boson sampling, a Fock state $|\boldsymbol{s}\rangle$ is input into the modes and a detection pattern $\boldsymbol{t}$ is measured. However, the probability of this happening is the same as if the measurement pattern had been a Fock state at the input $|\boldsymbol{t}\rangle$ and the input state corresponds to the detection pattern $\boldsymbol{s}$ if we use time reversal of the states through the interferometer. Applying time reversal to a unitary simply results in its transpose matrix. This is supported by the probability amplitude of a particular output pattern being given by the permanent of a submatrix of $\boldsymbol{U}$ where

the columns are determined by the input states and the rows by the output detection pattern. By swapping the input and output modes, this corresponds to instead taking the columns from the output pattern and the rows from the input state. Because we know the probability amplitude of this event occurring should be the same due to time reversal, the unitary matrix must be transposed so that the same elements are included in the permanent.

In Gaussian boson sampling the input states are squeezed vacuum or squeezed displaced states. In the arbitrary GBS paper [59], a similar set-up to scattershot boson sampling is explored where the identity is now generalised to any unitary. The two-mode squeezed states are treated as under partial time reversal where one mode acts forwards in time and the other is reversed. In this description the output pattern in the top modes can be considered as the input where the photons go backwards in time to the squeezed states and then go forwards in the bottom modes where the output pattern is given. In this manner the GBS set-up is essentially equivalent to standard boson sampling with input Fock states measured in the number basis, but with an extra complication of a weighting in which mode they pass through in the middle because of the squeezing. This weighting corresponds to the probability that the pair of photons was produced from that squeezed state. So in the case where the squeezing is equal each mode is equally likely and the probabilities are scaled to reflect the probability of those photons being produced by any squeezed state. We expand on this idea for GBS with input squeezed states measured in the number basis by considering each state produces photons in pairs and so in each pair we can force them into two modes but then combine the modes before detection. Adding a mode and removing it before the measurement is a theoretical concept and does not change the probabilities. Furthermore, we can consider the pattern before recombining the modes as long as we only consider the probability *amplitudes*.

The mapping between a GBS set-up is shown in fig. (2.8). Each pair of photons can be considered as a single photon under partial time reversal. This single photon now starts in a mode in the top half, travels backwards in time through a unitary $\boldsymbol{U}$, which as explained above corresponds to the transpose of the unitary. Then it undergoes some probability of being created in the mode after the unitary and travels forwards through the lower unitary to be measured in a mode in the lower half. This is where the expression of the matrix $\boldsymbol{URU}^\top$ arises. For a particular output pattern the probability is given by summing over the probability amplitudes of all the paths to match the input to the output pattern. However, for a particular output pattern, the measurement does not distinguish whether the photon was in the theoretical top or bottom modes and so it is necessary to sum over all the ways of splitting the detection pattern in half. The ways of splitting a group in half and matching every element in one half to an element in the other is all the ways of pairing the elements. This is precisely the hafnian.

If we now include displacement in the input squeezed states, we can extend the summation over all paths to include similar tricks to those used in section (2.2.3.3) to treat the displacement independently of squeezing. So in this case we need to consider all scenarios where a subset of the photons came from the displacement and the rest from the squeezing. As 'displacement' photons are not paired whereas 'squeezing' photons are, we must sum all the ways of pairing $2s$ photons and not matching (or matching to themselves) the $N-2s$ remaining photons for all s. This corresponds

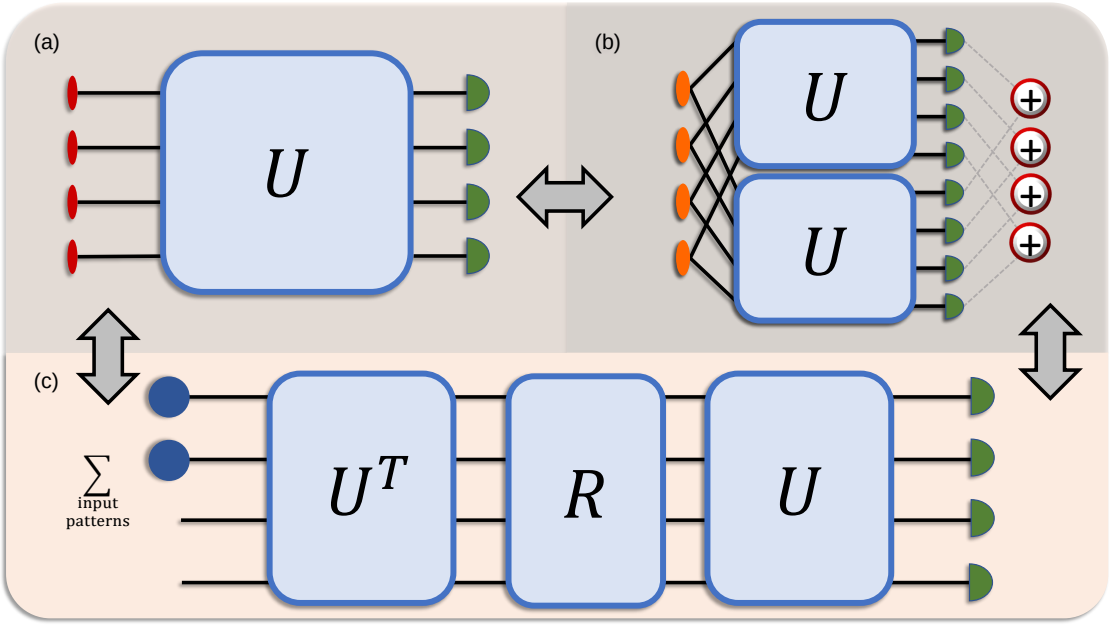

Figure 2.8: Retrodictive description of GBS. The set-up for GBS is depicted in (a), with single mode squeezing and number resolving detectors. Single mode squeezing always produces photons in pairs (in the ideal case) and so it's possible to consider forcing the two photons into separate modes (like two-mode squeezing) as long as they are recombined before measurement. This is shown in (b) where one photon from each pair passes through the top unitary and the other the bottom. If the states are not measured when they are in separate modes but instead recombined before measuring, these two pictures are equivalent. In a theoretical framework, it is possible to reverse this combining step and split the photon pattern such that half of the photons were detected in the upper modes. The probability can be found by summing the probability amplitudes of all these possible splits. In (c) we show the set-up that the probability amplitudes need to be calculated for. The sum over input photons is all the ways of choosing half of the photons from the output photon pattern. For each possible split, the probability amplitude of the input state being detected at the output is now true for photons passing through the interferometer shown here. The matrix $\boldsymbol{R}$ is not unitary and applies scaling to reflect the probabilities of originating from each squeezed state. From this set-up the emergence of the Haf($\boldsymbol{URU}^\top$) can be seen.

to a perfect matching with loops, where the loops represent the displacement photons and the pairs represent the squeezing photons. This is given by the loop hafnian of a matrix with diagonal and off-diagonal elements being the probability amplitudes of the photons coming from displacement and squeezing respectively. The probability amplitudes for the squeezing part are the same as for GBS with no displacement, i.e. $\boldsymbol{URU}^\top$. The amplitudes for the displacement photons need to be adjusted so it can be considered as independent of the squeezing. As can be seen from eq. (2.118), the required amplitude is proportional to $\boldsymbol{U}[\boldsymbol{\alpha} - \boldsymbol{R}\boldsymbol{\alpha}^*]$, where we look at the lower half of $\boldsymbol{\gamma}$ as the $\boldsymbol{URU}^\top$ matrix is in the bottom right block in eq. (2.117). Note that if the unitary is the identity and we consider one mode, we recover the relation for squeezed displaced states in eq. (2.75). More generally the displacement is given by $(\boldsymbol{\Sigma}_Q^{-1}\bar{\boldsymbol{\zeta}})^*$, where the complex conjugate is due to $\boldsymbol{A} = \boldsymbol{B}^* \oplus \boldsymbol{B}$, rather than the more natural $\boldsymbol{A} = \boldsymbol{B} \oplus \boldsymbol{B}^*$. The multiplication by the inverse of the Husimi Q matrix is perhaps not surprising as the Q matrix is the covariance matrix of the Gaussian state in the displacement basis. Specifically sampling from a Gaussian distribution with mean given by the displacement (β, β^*) and covariance $\boldsymbol{\Sigma}_Q^{-1}$ projects the state onto a coherent state. Therefore multiplying by its inverse can account for the implicit displacement in the state.

Hence, we have explained the appearance of the hafnian and loop hafnian in the probability expressions using the summation over all possible paths from input to output. Also we motivated the elements in the corresponding matrices using the retrodictive picture for squeezing.

2.4.4 Sampling from a probability distribution

In the previous section, we provided detailed analysis for the probability of an output pattern in the different variants of boson sampling. However, we stress that the task of boson sampling is indeed to *sample* from the distribution. This is potentially a very different problem, where sometimes sampling from a distribution can be easier than the calculation of a single probability if the structure of the distribution allows tricks to be implemented, but often it is harder to sample from a distribution. Generally any distribution can be sampled if the full distribution is known. This approach which we refer to as 'brute force sampling' can be extremely slow if the number of possible outcomes is large as the number of probabilities that need to be calculated affect the speed of the sampling. In boson sampling problems the scaling of the size of the sample space as the number of photons or modes increases is very problematic and the number of probabilities to calculate is the first bottleneck when using brute force sampling. This is the problem that is addressed in this thesis, where we suggest multiple methods to sample which do not require the knowledge of the full distribution.

Chapter 3

Simulating Gaussian Boson Sampling with Rejection Sampling

"I'm always trying to calculate the mathematical probability of certain outcomes."
- Paul Allen

The main aim of this thesis is to classically simulate Gaussian boson sampling, and as such we apply existing techniques for sampling from probability distributions to accomplish this. The task of sampling from a probability distribution can be simply expressed as follows: given a random variable X is distributed according to some probability distribution $f(x)$, an outcome or sequence of outcomes $\{x_i\}$ is output each with probability $f(x_i)$. We stress that this is not the same as finding the distribution $f(x)$ for all x. We begin by introducing the formal terminology associated with sampling [60].

In probability, an **experiment** is a method that generates an **observation** or **outcome** - a value of a variable. All the different values this variable can take form the **sample space**, Ω, where each of the possible values is called a **sample point**. If an experiment samples the random variable X, we denote a sample point as x_i, for any $i = 1, \ldots, |\Omega|$, and the sample space is the set $\{x_i | \, i = 1, \ldots, |\Omega|\}$, where $|\Omega|$ is the size of the sample space. Formally, a sample space must satisfy the following requirements:

1. The outcomes must be mutually exclusive - the outcome from an experiment can never be more than one sample point,

2. The outcomes must be collectively exhaustive - the sample space must contain all possible outcomes.

This means in an experiment, an observation x_i will be generated, which is one and only one sample point. An **event** is a subset of the sample space that is of interest. The **sample size**, N_s, is the number of observations/outcomes that are generated from the experiment to form a **sample**, which is a collection of observations. Experiments always generate outcomes according to a probability

distribution. If the variable is discrete, the probability that an outcome will take a certain value is given by its **probability mass function** (pmf), whereas if it is continuous, the probability is given by its **probability density function** (pdf). In both cases, we denote this probability as $f(x)$, where x may be discrete or continuous, and refer to it as the probability distribution. A **cumulative distribution function** (cdf) is the sum or integral of the probability distribution until that point, or equivalently the probability that the outcome will have at most that value, $F(x_i) = \Pr(X \leq x_i)$. For discrete quantitative variables, this involves summing over all the smaller values, whereas for continuous variables, instead integration over all the smaller values is required. For discrete qualitative variables, we can consider a cdf if we choose an arbitrary order of the sample space. We denote a random variable X that is distributed with a probability $f(x)$ as $X \sim f(x)$, where $f(x)$ is shorthand for $\Pr(X = x)$, the probability that the random variable has value x. We use upper case letters for random variables and lower case for particular realisations of them.

There are several ways to perform sampling from a distribution, and the best method is dependent on the distribution to be sampled. First consider the example of sampling from the uniform distribution between 0 and 1: $X \sim U(0, 1)$. We assume a computer has access to uniformly random integers, t, between 0 and $t_{\max}$. In reality a classical computer cannot generate completely random numbers and instead uses pseudo-random number generation. Clearly the value $x = t/t_{\max}$ will be a sample from the uniform distribution between 0 and 1, in steps of $1/t_{\max}$. This is an approximation to sampling from the uniform distribution where the precision is limited by the size of the step. However, by increasing $t_{\max}$, an arbitrary precision can be found. We note that any continuous probability density function will always be approximated by discretisation and treating it as a probability mass function. Therefore, we will generally refer to probability distributions in the context of probability mass functions but, unless otherwise stated, everything can be applied in the continuous case too.

For any general pmf $f(x)$, it is possible to sample from it by only sampling from the uniform distribution $U(0, 1)$ using the **inverse transform sampling** method [61]. In this method, the cumulative distribution function $F(x)$ is built from $f(x)$. First, a value, u, is sampled from the uniform distribution $U \sim U(0, 1)$, and this value is associated to the cumulative value: $u = F(x)$. If $U \sim U(0, 1)$, this means $x = F^{-1}(u)$ is an observation of a variable distributed by $X \sim F^{-1}(U(0, 1))$. Here the existence of the inverse cdf would imply that it is bijective. This is generally not true for cdfs. They are not injective because cdfs are weakly monotonically increasing meaning that several values of x may have the same cumulative value $F(x)$, which is the case when some outcomes have zero probability of occurring. Secondly, for variables with a finite sample space, only a finite number of values are possible for the cumulative probability (at most one for each sample point). Therefore the cdf is not surjective on the codomain $[0, 1]$. Hence the inverse cdf used here is defined as

$$F^{-1}(u) = \min\{x | F(x) \geq u\}. \tag{3.1}$$

This function maps any u that satisfies $F(x_{i-1}) < u \leq F(x_i)$ on to x_i. Therefore if u is sampled from $U(0, 1)$, the probability that u is within this range is equal to the difference between the upper

and lower limits which by the definition of the cdf is precisely the pmf $f(x_i)$. Hence $x_i = F^{-1}(u)$ is sampled with probability $f(x_i)$ as desired.

An example for a discrete variable is shown in Fig (3.1). The pmf is shown on the left and the height of the bars represent the probability of that value. It is intuitive that sampling any point in that graph uniformly will produce points in each colour in the correct proportions (ie. according to the probability distribution) as the higher the bar the more space there is to sample a point in it. Hence, sampling from a distribution is equivalent to uniformly sampling from the area of the pmf. However, it is not easy in practice to sample uniformly from this shape. Therefore the cdf is built as shown in the middle plot. There is a one-to-one correspondence between the cdf and the outcomes x_i and therefore sampling the cdf is equivalent to sampling an outcome. The plot on the right-hand side shows more clearly the mapping between the cdf and the outcomes, assigning the same area as in the pmf plot to each colour but now in a straight line. It is easy to sample uniformly a number between 0 and 1, and from this the colour at that height is the outcome sampled. When sampling using the inverse transform method, it is necessary to know the probabilities of at least all $x \leq x_i$ when outputting x_i. This is because we need to sum over the probabilities of all x in order until we find x_i such that $F(x_{i-1}) < u_i \leq F(x_i)$. Given it is possible to sample the final sample point (ie. highest value of the variable), we must assume that we might need to calculate the probabilities of all sample points. So the inverse transform sampling method is only useful when the entire distribution can be found.

It is also possible to sample multiple variables from a multidimensional probability distribution. In this case it is possible to consider the variables together and sample all of them from the distribution at once. However, it may be more convenient to sample one variable at a time, conditional on the outcomes of the previous variables, or independently if the variables are independent. For example, consider the distribution $\{X, Y, Z\} : X, Y, Z \sim U(0, 1)$. X, Y and Z are independent variables and so it is simplest to sample each variable independently to get the overall outcome $\{x, y, z\}$. This can be generalised to more complicated functions by using the inverse transform sampling method as above.

In many cases, the difficulty in sampling arises from the huge number of possible outcomes to consider. Then it becomes impossible to calculate the cumulative distribution for the whole sample space, which means inverse transform sampling is not possible. For the application we are interested in, GBS, this is true and so we now look at the first method to avoid this problem: rejection sampling.

3.1 Rejection sampling

3.1.1 Exact rejection sampling

The task is to sample a random variable $X \sim f(x)$, where $f(x)$ is known as the 'target distribution'. Rejection sampling [62] builds on the inverse transform method. As we saw above this method turns a 2D shape that is difficult to uniformly sample from into a one-dimensional line which is easy to

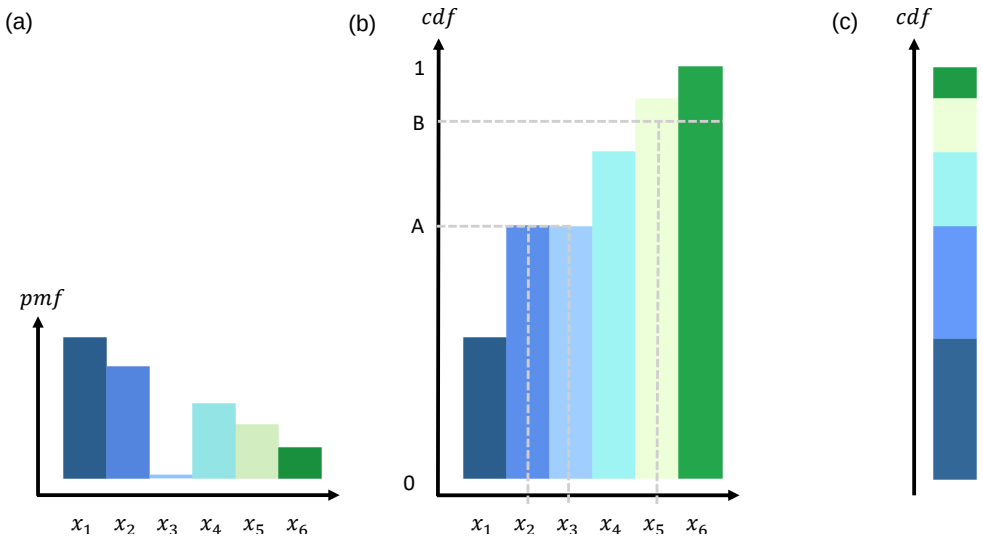

Figure 3.1: Inverse transform sampling. When sampling from a probability mass function (pmf) it is possible to convert it into a cumulative distribution function (cdf) and sample uniformly between 0 and 1. A variable's pmf and cdf are shown in plots (a) and (b) respectively. Plot (b) shows the mapping between the uniform variable and the outcome sampled. When there are two corresponding outcomes with a cdf equal to the uniformly sampled value, the first outcome is taken; see for example if the sampled value is u_1 the outcomes x_2 and x_3 both have this cdf, so in this case x_2 will be sampled. When no outcomes have the exact value sampled (which is always the case if the values can have infinite precision) the first outcome that has a greater cdf than the sampled value is chosen as shown with u_2 corresponding to x_5. Plot (c) provides another way of showing the correspondence between the random variable and the outcomes making it clearer that each outcome is sampled with probability given by the area of the pmf as expected.

sample from. Rejection sampling adds another outcome, 'reject' to turn the awkward 2D shape into one that can be sampled first in the x-coordinate and then the y-coordinate. Fig. (3.2) shows an example of this additional outcome in purple in (c) which turns the area to be uniformly sampled into a rectangle. A point inside a rectangle can be uniformly sampled simply by first sampling an x coordinate and then the y coordinate both from $U(0,1)$. Part (d) in the figure shows the equivalence to using the inverse sampling method on this new distribution (although it is not necessary). By adding the new outcome, the probabilities of all the others need to be scaled to renormalise the distribution. However, it is clear that the probability of sampling any of the original outcomes are in the same proportions, but there is also some probability of sampling this 'reject' option. Therefore by discarding any of these 'reject' outcomes and repeating until a sample is found that does not contain 'reject' outcomes, this will be equivalent to sampling from the original distribution.

Suppose $f(x)$ is completely unknown but the probability of any outcome, x, can be calculated. In rejection sampling, another distribution $g(x)$, commonly referred to as the 'proposal distribution' is introduced. The proposal distribution is multiplied by some constant H such that this scaled distribution is never smaller than the target distribution, $Hg(x) \geq f(x)\,\forall x$. We refer to this scaled proposal distribution as the 'envelope distribution'. The distribution can be sampled using rejection sampling as described in algorithm 1. The algorithm simply consists of two steps repeated until an

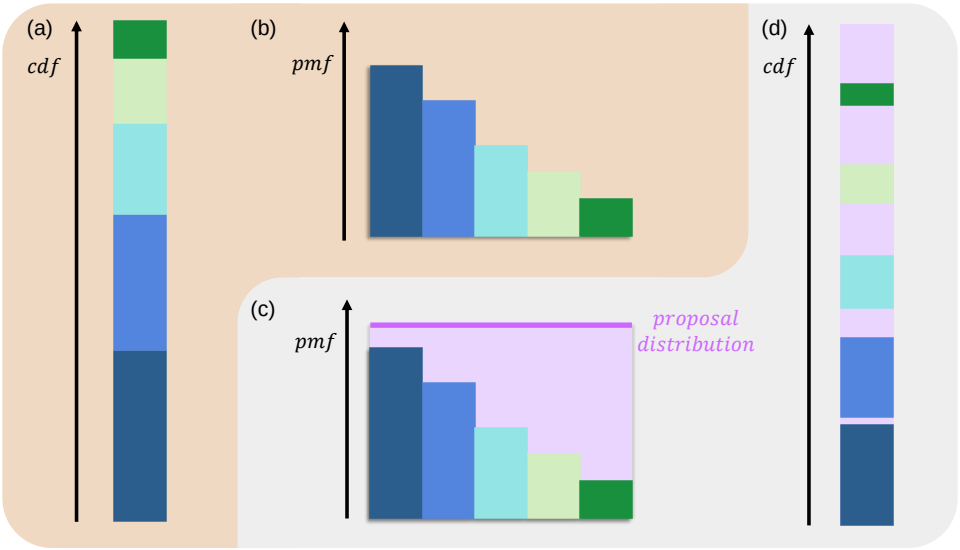

Figure 3.2: Inverse transform sampling and rejection sampling. A pmf is shown in (b) for a discrete random variable with the size of the sample space being 5. The height of the bars gives the probabilities of each outcome, in this case represented by the colours. (a) gives the mapping between the value of cdf (between 0 and 1) and the corresponding outcome (colour). The heights of each colour bar are the same in (a) and (b) and so uniformly sampling in either will sample according to the correct probabilities. It is more convenient to sample uniformly a number between 0 and 1 than a point in a 2D graph. Using the mapping between a random number between 0 and 1 to the cdf is known as inverse transform sampling. (c) shows another way of sampling uniformly from the area in (b) by adding another outcome given by the purple shading. Sampling by area in this graph will output the other outcomes in the same relative probabilities as before but with another option, purple. If we discard all purple outcomes, the remaining outcomes will be according to the original pmf. (d) shows the cdf now with the additional outcome included.

outcome is accepted:

1. Sample from $g(x) \to x$,

2. Accept x with probability $\frac{f(x)}{Hg(x)}$, otherwise reject.

Note that the first step samples an x-coordinate, and the second step is equivalent to sampling a y-coordinate between 0 and $Hg(x)$ and accepting if it is less than $f(x)$. Therefore, in this algorithm only one probability from the target distribution needs to be calculated in each repeated attempt and so for a sample of size 1 the number of probabilities to be calculated is the number of repeats until a proposed outcome is accepted.

To prove that this samples exactly from the target distribution, we need to show that the probability of an accepted outcome being x is $f(x)$. By Bayes' theorem, the probability of sampling

Algorithm 1 Rejection Sampling

Require: H s.t. $f(x) \leq Hg(x)\,\forall x$
  while True **do**
    Sample $x_i \leftarrow g(x)$
    Sample $u_i \leftarrow U(0,1)$
    if $u_i \leq \frac{f(x_i)}{Hg(x_i)}$ **then**
      return x_i

x conditional on the proposed state being accepted is given by

$$\Pr(x|\text{accept}) = \frac{\Pr(x \cap \text{accept})}{\Pr(\text{accept})} = \frac{\Pr(x)}{\Pr(\text{accept})}, \tag{3.2}$$

because if x is sampled, it must be true that the proposed state was accepted (and x) so the accept outcome is redundant in the numerator. First we look at the probability of accepting x. In step 1, the state x is proposed with probability $g(x)$. Then in step 2, it is accepted with probability $\frac{f(x)}{Hg(x)}$. So the probability of sampling x is the probability of proposing and accepting it:

$$\Pr(x) = g(x)\frac{f(x)}{Hg(x)} = \frac{f(x)}{H}. \tag{3.3}$$

Now we consider the overall probability of accepting independent of what is proposed in step 1. The overall probability of accepting is

$$\Pr(\text{accept}) = \sum_x \Pr(x) = \sum_x \frac{f(x)}{H} = \frac{1}{H}. \tag{3.4}$$

We note that the efficiency of the algorithm is given by the acceptance probability and hence is inversely proportional to the value of H. Assuming the proposal distribution is normalised, H is equal to the area under the envelope distribution, $Hg(x)$. Substituting eqs. (3.3) and (3.4) into (3.2) we recover

$$\Pr(x|\text{accept}) = \frac{f(x)}{H}H = f(x) \tag{3.5}$$

as expected.

Rejection sampling is a general sampling algorithm and can be used to sample from any distribution as long as it is possible to find a proposal distribution that is efficient to sample from and a value of H such that $Hg(x) \geq f(x)$ for all x. Note that $Hg(x) \geq f(x)$ implies that for all x with $f(x) > 0$, we require $g(x) > 0$ and so the support of the target distribution must be contained in the support of the proposal distribution. Choosing a distribution that is efficient to sample from is easy, for example take the uniform distribution. However, setting H to ensure $Hg(x) \geq f(x)$ is difficult given $f(x)$ is unknown. It could be set very high (eg. $Hg(x) \geq 1\,\forall x$ would always work), but a high value of H means a low rate of acceptance and therefore is less efficient. It is best to optimise H by setting it as low as possible, whilst still satisfying the condition. The closer $g(x)$ is to $f(x)$, the smaller the optimal value of H and the more efficient the algorithm. A poor proposal

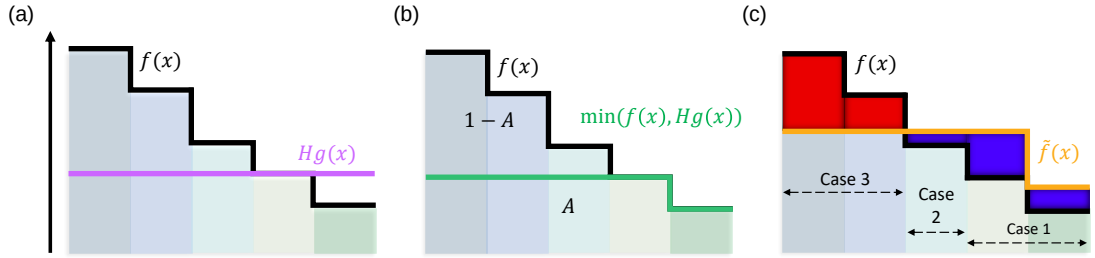

Figure 3.3: The error in approximate rejection sampling. In (a) the height of the envelope distribution (here uniform), $Hg(x)$, is too low such that for three of the outcomes they do not satisfy $f(x) \leq Hg(x)$. The area of $f(x)$ below the envelope distribution, A, and the area cut off, $1 - A$, is shown in (b). The green distribution is the minimum of the envelope and target distributions, and gives the relative probabilities of the outcomes sampled in approximate rejection sampling. As the cut off probability needs to be accounted for, the approximate distribution is renormalised in (c) to find the sampled distribution $\tilde{f}(x)$. This is the distribution that is actually being sampling from in this case. Case 1 corresponds to the outcomes where $f(x) \leq Hg(x)$, whereas cases 2 and 3 do not satisfy this exact constraint. Cases 2 (3) are the outcomes such that the sampled distribution is greater (less) than the target distribution: $\tilde{f}(x) > f(x)$ ($\tilde{f}(x) < f(x)$).

distribution or an overly generous value of H can have severe impact on the efficiency to the extent where the algorithm is no longer useful. Therefore, it is important to find a good candidate for the proposal distribution and a close estimate of the optimal H. These are the main challenges when implementing the algorithm.

3.1.2 Approximate rejection sampling

For exact rejection sampling as described above, the condition $Hg(x) \geq f(x)$ must be satisfied for all x. This is to ensure that in step 2 of the algorithm, the probability of accepting the proposed outcome is always less than or equal to 1. However, given $f(x)$ is generally unknown, finding H large enough to satisfy this and small enough for good efficiency is a non-trivial task. In many cases it may only be possible to *estimate* the minimum value of H required. In this case, the constraint can be removed as long as the algorithm is updated to upper bound the probability of accepting by 1 to ensure a meaningful probability. The acceptance rule in step 2 is adapted to

$$\Pr(\text{accept } x|\text{proposed } x) = \min\left\{1, \frac{f(x)}{Mg(x)}\right\}. \tag{3.6}$$

In the case where H is big enough to meet the requirement of exact sampling, this rule reduces back to the previous algorithm, and so this is a generalisation of the previous algorithm with no constraints. We now explore the consequences the relaxation of the constraint has on the efficiency and accuracy of the rejection sampling algorithm.

The efficiency is given by the overall probability of accepting any proposed outcome. This is

now given by:

$$\Pr(\text{accept}) = \sum_x \Pr(\text{propose } x)\ \Pr(\text{accept } x|\text{proposed } x) \tag{3.7}$$

$$= \sum_{x\in B} g(x)\frac{f(x)}{Hg(x)} + \sum_{x\notin B} g(x) = \frac{1}{H}\left(\sum_{x\in B} f(x) + \sum_{x\notin B} Hg(x)\right) = \frac{A}{H}, \tag{3.8}$$

where A is the area of $f(x)$ under the envelope distribution, $Hg(x)$, and $B = \{x : f(x) \leq Hg(x)\}$, the outcomes that satisfy the exact constraint. The area A is shown in fig. (3.3b) as the area under the green distribution which is the minimum of the target and envelope distributions. The outcomes in B are given by 'case 1' in fig. (3.3c) and those not in B are in 'case 2' and 'case 3' which will be distinguished later. Note that this equation is true even if $f(x)$ is not normalised, where A is the absolute area under the envelope distribution, but the proposal distribution $g(x)$ must be normalised. The area of $f(x)$ under the envelope distribution can never be greater than the total area of $f(x)$ and, assuming the target distribution is normalised, this means $A \leq 1$ with equality only when H is large enough to satisfy the exact condition for all outcomes. Therefore, from the above equation it can be seen that for a fixed value of H the efficiency of the rejection sampling algorithm is optimal when the exact criteria is met and gets worse the more area cut off by the envelope distribution. If H is fixed, the only way for more of the target distribution to be below the envelope distribution is if the target and proposal distributions are closer to each other. So this statement just reiterates that a proposal distribution that is closer to the target distribution is more efficient. If instead the target and proposal distributions are fixed and the value of H is scaled by a value $\alpha > 1$, it is less obvious whether the probability of accepting a proposed outcome increases or decreases, as both A and H change. The scaling of $H \to \alpha H$ implies that $A = \sum f(x) + \sum Hg(x) \to \sum f(x) + \alpha \sum Hg(x) < \alpha A$. Therefore $A/H \to \tilde{A}/\tilde{H} < \alpha A/(\alpha H) = A/H$ and so the efficiency decreases as H increases. So by reducing H the algorithm accepts the proposed state more often and is faster, but the more of the target distribution cut off by the envelope distribution. The increase in efficiency is a positive, but it is important to know the impact this has on the accuracy of the sampling algorithm.

When a proposed outcome is accepted, the probability of it being a particular outcome x should ideally be equal to $f(x)$. Using the adapted algorithm, the probability is given by

$$\Pr(x|\text{accept}) = g(x)\min\left(1, \frac{f(x)}{Hg(x)}\right)\frac{H}{A} = \min\left(\frac{Hg(x)}{A}, \frac{f(x)}{A}\right) = \begin{cases} \dfrac{Hg(x)}{A} & \text{if } f(x) > Hg(x) \\ \dfrac{f(x)}{A} & \text{if } f(x) \leq Hg(x) \end{cases} \tag{3.9}$$

A perhaps more intuitive way of finding this is shown in fig. (3.3). The target and envelope distributions are shown in (a) where the first three outcomes do not meet the exact condition. The probability above the envelope distribution is cut off and not included when sampling. The area sampled from is given by A in (b), the minimum of both distributions. However, for a distribution

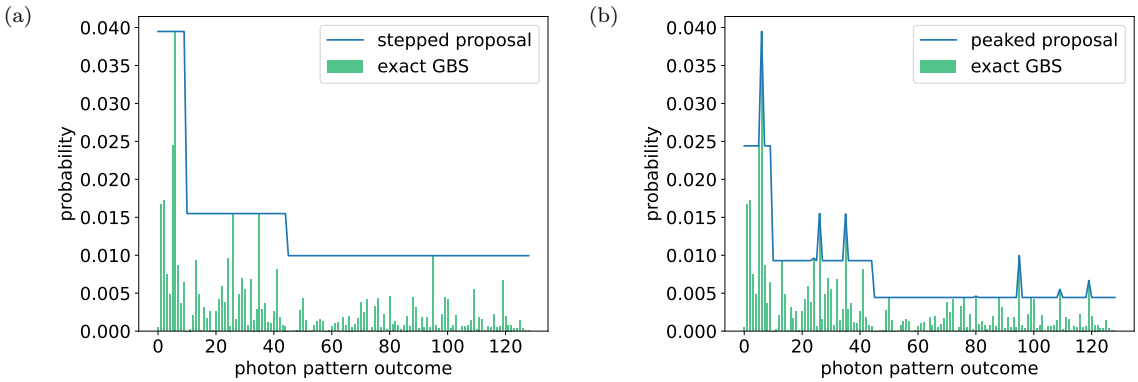

Figure 3.4: The proposal distributions for rejection sampling. For a Haar random unitary, the target distribution is shown by the green bars. The proposal distribution must be multiplied by a scaling factor so that it is always greater than the target. Two examples of the scaled proposal distributions are given by the blue lines. In (a), we show the stepped uniform proposal distribution which comprises a uniform distribution for each photon number set at the maximum probability. In (b), we have the peaked distribution which is based on the stepped uniform but where the height in each level is set as the nth maximum probability in that photon number. For the outcomes where the probability is greater than this value, the proposal takes the same value as the target distribution. This is beneficial as the area below the proposal distribution is smaller than for the stepped uniform.

to be sampled it must be normalised and clearly the area below both distributions will always sum to less than 1. Therefore this distribution is rescaled in (c) to find $\tilde{f}(x)$, the sampled distribution. Outcomes x that satisfy the exact condition are sampled with a higher probability than they should and outcomes that do not satisfy the condition are sampled with a probability determined by the envelope distribution only. As A approaches 1, which is equivalent to the exact condition always being met, these probabilities reduce to the correct probabilities as expected. The more probability cut off above the envelope distribution the further the distribution sampled from is from the target distribution.

In cases where $A < 1$ the algorithm samples only approximately from the target distribution. This is often good enough for many purposes, however it is important to know how approximate the sampling algorithm is by determining how close the two distributions are. To quantify this measure we use the total variation distance (TVD) between the target distribution and the distribution sampled from. The TVD between two distributions $q(x)$ and $r(x)$ is defined as

$$\mathrm{TVD}(q(x), r(x)) = \frac{1}{2} \sum_x |q(x) - r(x)|. \tag{3.10}$$

3.2 Rejection sampling applied to Gaussian boson sampling

Now we look at applying rejection sampling to simulating Gaussian boson sampling. Rejection sampling is beneficial to sampling problems where the desired sample size is significantly smaller than the size of the sample space. As discussed in the previous chapter, Gaussian boson sampling

(GBS) is such a problem where the sample space comprises the output patterns $(n_1, n_2, \ldots, n_M)$ and the target probabilities are given by eq. (2.110) or eq. (2.111) if there is no or some displacement in the input states respectively. The size of this sample space scales with the number of modes, M, and number of photons, N, as $\binom{N+M-1}{N}$. This combinatorial scaling grows extremely rapidly and so, for a reasonable sample size, rejection sampling is a good candidate for sampling from a GBS distribution. The challenge is to find a proposal distribution that is both efficient to sample from and a good approximation to the GBS distribution, and finding the scaling parameter H to ensure the envelope distribution is greater than the target distribution for most outcomes. We suggest a couple of proposal distributions and analyse the efficiency of them numerically.

In GBS, the number of photons is not fixed, and so it is possible to have a proposal distribution which incorporates all output patterns of any photon number or to first sample the number of photons and then perform rejection sampling on that subset of output patterns until a proposed outcome is accepted. We stress that if the whole sample space is considered in one proposal distribution, each time we reject or accept a proposed state the next proposed state can have any number of photons, whereas when first sampling the photon number, we must continue proposing outcomes with that photon number until a state is accepted. We now show that both methods have the same efficiency and accuracy. In the case of sampling the photon number first, each proposed state is not independent because we must propose the same photon number until it is accepted. Therefore, the most appropriate parameter to measure efficiency is the expected number of outcomes to propose until one is accepted. When the proposal distribution incorporates all photon numbers, the expected number of probabilities is given by $1/p = H$, for the exact case where p is the probability of accepting. When first sampling the photon number, we sample a photon number N with probability $\Pr(N) = F_N = \sum_{x \in N} f(x)$, where $x \in N$ means that the output pattern x has N photons. Whichever output pattern is proposed is then accepted with probability $F_N/(HG_N)$, where $G_N = \sum_{x \in N} g(x)$. The expected number of probabilities to calculate is given by the weighted sum of the expected value of the number of probabilities in each photon number:

$$\mathrm{E}(\#\text{probabilities}) = \sum_N \Pr(N)\mathrm{E}(\#\text{probabilities in } N) = \sum_N F_N \frac{HG_N}{F_N} = \sum_N HG_N = H, \quad (3.11)$$

where we use that the proposal probabilities G_N are normalised to sum to 1. Hence, the expected number of probabilities to calculate for one accepted outcome is the same in both cases. The accuracy is also the same in both cases which follows because the area cut off by the proposal distribution is the same so the TVD in each photon number is the same whether considered separately or not.

We generally find it is more useful to first sample the number of photons and then sample the output pattern from a proposal distribution dependent on the total number of photons. This also allows the number of photons to be fixed to restrict to a particular photon number which is equivalent to post-selecting from the quantum device. Therefore, we begin by describing a method to sample the number of photons.

3.2.1 Sampling the number of photons

To sample the total number of photons in the output photon pattern, we use the fact that the probability distribution of the number of photons in a state is conserved after passing through an interferometer under the assumption of no loss. In general, uniform loss across all modes in the interferometer can be included by moving the loss into the sources [25, 15]. So to sample the number of photons in the output pattern, we can instead sample the number of photons from the sources. The probability of n_i photons from a source in mode i is given in eq. (2.66) for a squeezed vacuum state or eq. (2.72) for a squeezed displaced state. Theoretically, the sources can produce an infinite number of photons but we restrict the total number of photons, N, to a chosen cut-off N_c such that $\Pr(N \leq N_c) < \epsilon$. To sample the number of photons from a source in mode i, the probabilities of all $n_i = 0$ to $n_i = N_c$ are calculated and this distribution can be sampled after renormalising. The sources are independent so each source can be sampled separately and the total photon number is simply given by the sum of the number of photons from each source, $N = \sum n_i$. As each source is allowed up to N_c photons, sometimes the sampled total photon number will be above this chosen cut-off and it is necessary to repeat until a photon number equal to or below the cut-off is sampled. However this will happen with low probability as long as ϵ is small and not have a detrimental effect on the efficiency of this method.

It can be important to find the probability of N photons in total, such as to determine the cut-off photon number to ensure $\Pr(N \leq N_c) < \epsilon$. To do this the probabilities of all the possible combinations of n_i such that $N = \sum n_i$ need to be summed. The number of combinations scales combinatorially $\binom{N+M-1}{N}$ which quickly becomes unmanageable. However, it is possible to find the *sum* of these probabilities without needing to calculate all the individual probabilities. For each mode, assign the probability of each n_i to an auxiliary variable x^{n_i}. So for mode i we form the polynomial

$$\sum_{n_i=0}^{N_c} \Pr(N_i = n_i) x^{n_i} = \Pr(N_i = 0) + \Pr(N_i = 1)x + \Pr(N_i = 2)x^2 + ... \tag{3.12}$$

Here the coefficients for each term are given by the probability of getting n_i photons where n_i is the order of that term. Given a polynomial for each mode as above, we want to find a new polynomial where the coefficients give the probability of getting N photons across all modes. Multiplying polynomials naturally collects together all the possible ways of getting N photons with the probability given as the coefficient of the x^N term. For example, if we want the probability of measuring 2 photons across 3 modes, we need to add the probabilities of detecting patterns $(0,0,2), (0,2,0), (2,0,0), (1,1,0), (1,0,1), (0,1,1)$ from the source. The probability for $(0,0,2)$ corresponds to the coefficient of the term $x_1^0 x_2^0 x_3^2$ (where the subscript labels the mode, but $x_1 = x_2 = x_3 = x$) so the sum of all the 2 photon probabilities naturally comes out as the coefficient of x^2. Mathematically, the probability of N total photons for any $N \leq N_c$ is given by

the coefficients in the following polynomial:

$$\sum_{N=0}^{MN_c} \Pr\left(\sum N_i = N\right) x^N = \prod_{i=1}^{M} \left(\sum_{n_i=0}^{N_c} P(N_i = n_i) x^{n_i} \right). \tag{3.13}$$

Note that the coefficients of orders greater than N_c do not give the correct probabilities for the total photon number because some terms are missing from the sum where more than N_c photons are measured in the same mode, but we are only interested in photon numbers below this. Hence, the problem of calculating the probabilities is reduced to multiplying together M polynomials of degree N_c. The naive way to multiply two polynomials of order N_c is to expand the brackets which results in N_c^2 elements. Hence the complexity of this is $O(N_c^2)$. To multiply M polynomials together we simply multiply the first two and then in turn multiply the result by the next polynomial. However, multiplying two polynomials of order N_c outputs a polynomial of order $2N_c$ and so this results in an increasingly high order of polynomial to multiply. In our case, we are only interested in the coefficients up to order N_c and therefore we can ignore any orders above this as we progress with the multiplication, cutting off the higher orders after each multiplication. Therefore, to calculate the final polynomial and hence the probability of any number of photons, it can be done by $M-1$ multiplications of polynomials of order N_c. The naive approach gives an overall complexity of $O(MN_c^2)$, which is polynomial with respect to both the number of photons and modes and the time taken to compute this is negligible compared to the time for a single probability as N_c increases assuming a reasonable M. Faster algorithms to multiply polynomials exist [63] and in particular the fast fourier transform can be used to find an algorithm with complexity of $O(N_c \log N_c)$ [64, 65] which reduces the overall complexity to $O(MN_c \log N_c)$.

3.2.2 Proposal distributions

One of the biggest challenges when applying rejection sampling to a task is to find a suitable proposal distribution that is efficient to sample from, the probability of any outcome can be found and gives a good acceptance rate. We introduce two proposal distributions that can be used for simulating GBS with rejection sampling: the stepped uniform and peaked distributions. These can be applied for both a fixed or variable total photon number.

3.2.2.1 The stepped uniform distribution

The first proposal distribution we try is simply the uniform one. For exact rejection sampling we need to find H such that $Hg(x) \geq f(x)$ for all x. For the uniform distribution, this corresponds to finding the maximum of the target distribution and setting H such that $Hg(x) = \max_x(f(x))$. In order to do this we need to know the maximum value in the target distribution, and set the envelope distribution at this height. To find the maximum value, we use an optimisation algorithm: hill climbing. We explain two variants of this algorithm in the following section and analyse their performance. Although they are not guaranteed to find the global maximum, we find numerically that they find close to the maximum generally and so they can be used in approximate rejection

sampling to sample from a distribution that is close to the desired GBS distribution.

As the probabilities of some photon numbers can be significantly higher than others, the maximum value in each photon number can be very different. So if we set the height of the envelope distribution at the global maximum, the area under it could be very large and hence the sampling extremely inefficient. To overcome this, it is more sensible to fix the height for each photon number separately, reducing the height of the distribution in most places and therefore the area under it. See fig. (3.4a) for a plot of an example of this distribution which we call the 'stepped uniform distribution' due to its appearance.

To sample from this distribution we simply fix or sample the photon number first and then sample an output pattern conditional on that photon number. When doing this, the target and proposal probabilities within the photon number are no longer normalised as they are normalised over the entire distribution. However, in rejection sampling the normalisation constant of the target distribution does not need to be known as long as the envelope distribution is always greater than the unnormalised target distribution. The proposal distribution must be normalised to sample from it, however the envelope distribution can be fixed by simply changing the value of H if the proposal distribution is renormalised.

We can sample the output pattern with the fixed/sampled photon number from the uniform distribution by assigning each pattern to an integer between 0 and $|\Omega|-1$, where $|\Omega|$ is the cardinality of the sample space, ie. the number of possible output patterns. Then it is as simple as sampling an integer uniformly. However, as the number of photons or modes increases, the number of possible output patterns increases very quickly and so storing the pairings between patterns and integers would cause memory errors. Yet, it is possible to order the patterns in such a way that a mapping can be found to convert between the pattern and an integer such that each integer corresponds to one and only one pattern. This mapping is as follows:

- Start with a pattern of N photons in M modes $\omega = (m_1, m_2, \ldots, m_N)$, where $0 \leq m_i \leq m_{i+1} \leq M-1$ (ie. in non-decreasing order).
- Remove the possibility of having more than one photon in the same mode by adding $i-1$ to m_i to get the new pattern $(\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_N) = (m_1, m_2+1, \ldots, m_N+N-1)$, where $\tilde{m}_i < \tilde{m}_{i+1}$ (ie. in increasing order). This pattern is now in the collision free space over $M+N-1$ modes, which removes problems with repeated modes being included for multiple orderings.
- $I = \sum_i \binom{\tilde{m}_i}{i}$.

Hence the output pattern ω can be uniquely mapped to the integer I. If we sample an integer and want to determine the corresponding pattern, we need to reverse this mapping. Given the sum of the binomial coefficients, we need to find the individual $\tilde{m}_i$. First, find the final $\tilde{m}_N$ by finding the largest $\tilde{m}_N$ such that $\binom{\tilde{m}_N}{N} \leq I$. Subtract the value of this binomial coefficient from I to leave the sum of the other binomial coefficients. Continue this process from the next last element until all $\tilde{m}_i$ have been found. To convert back to allow collisions, simply subtract $i-1$ from $\tilde{m}_i$. So if a random integer can be sampled, the corresponding pattern can be found.

However, it is not possible to sample a random integer between 0 and an arbitrarily large number due to memory constraints. For a 64-bit integer the largest maximum value is 2^{64}, which is not big enough for just 13 photons in 169 modes - a problem size for which the probabilities are quick to calculate. It is possible to generate an arbitrarily large number of k bits, but this does not allow for any desired upper limit. Therefore sampling from integers for the entire sample space is not generally a good choice for sampling from the uniform distribution for GBS. Because each element in a pattern is between 1 and M, this will always be a small enough range to sample, and is the basis for a memory efficient way to sample a pattern.

To sample a pattern in the collision free space (no repeated elements in the pattern), it is easiest to simply sample N of the modes at random in any order, and then order them. Each ordered pattern $(\tilde{m}_1, \ldots, \tilde{m}_N)$ where $\tilde{m}_i < \tilde{m}_{i+1}$ will correspond to $N!$ unordered patterns and therefore if the unordered patterns are generated uniformly randomly so will be the ordered patterns. However, if some of the elements are repeated, there are less unordered patterns that contribute to that pattern. For example, the state $(1, 2, 3)$ is sampled if any of the following are generated: $(1, 2, 3)$, $(2, 3, 1)$, $(3, 1, 2)$, $(1, 3, 2)$, $(3, 2, 1)$ and $(2, 1, 3)$. However for the state $(1, 1, 2)$, only the states $(1, 1, 2)$, $(1, 2, 1)$ and $(2, 1, 1)$ contribute. Therefore, $(1, 2, 3)$ is twice as likely to be sampled as $(1, 1, 2)$, and so this does not result in uniform sampling. To avoid this issue, we make use of the mapping between collision space with M modes and the collision free space with $M + N - 1$ modes. So we first sample a collision free pattern in $M + N - 1$ modes and then subtract $i - 1$ from m_i to find the desired pattern.

The advantage of using the stepped uniform proposal is that it is easy to sample from and in order to satisfy the exact constraint, only the global maximum for each photon number needs to be known or approximated. The drawback is that it is not necessarily close to the GBS distribution especially if just a few of the probabilities are much higher than the rest and so may not be the most efficient choice.

3.2.2.2 Peaked - rejection sampling

The second proposal distribution is aimed at GBS distributions where a lot of the probability is concentrated in just a few output patterns. In this case, the distribution is far from uniform and so that is not a good proposal distribution. Now we can separate the patterns into two groups where one includes the $n_{\max}$ outputs with the highest probability and the other contains all other patterns. The proposal distribution comprises of the exact values for the first group of states and a uniform distribution set at a lower height for the second group. Instead of fixing the height of the envelope distribution at the global maximum we would like to set the uniform level at the height of the $n_{\max}$th highest probability. This proposal distribution is a hybrid between the uniform distribution for the majority of the states and the target probability for just a few of the states with the largest probabilities. A plot of an example of this envelope distribution is shown in fig. (3.4b), and is named the 'peaked distribution' again due to its appearance. Similarly to the stepped uniform distribution, we fix the uniform part of the distribution at different heights for each photon number to reduce the area under the envelope distribution.

Sampling from this distribution requires the knowledge of $n_{\max}$ maxima not just the maximum. To do this we still use hill climbing algorithms, but we expect it to be more difficult to find all of the maxima values. The value of $n_{\max}$ is an arbitrary choice but we choose $n_{\max} = N$, the number of photons, because the number of outputs with unusually large probabilities is likely to increase as the size of the sample space does.

This distribution is easy to sample from as we can split the distribution in two: one part with the exact probabilities and the other with the uniform distribution. First we sample which half to choose. For this we need to know the relative probabilities of the two parts, which can be calculated easily: the probability of the exact values is just the sum of the known values, and the probability of the uniform group is given by its height multiplied by the total number of states in that group which is $\binom{N+M-1}{N} - N$. This is actually the area under the envelope which is not a normalised distributions and so these values will sum to H not 1, but the relative probabilities are unaffected by the scaling constant H. Then we sample from within the set we have selected. In the case of the uniform, we can sample as above and simply reject any time we select one of the patterns in the exact part. Given the number of maxima will be very small in comparison to the total number of patterns, the number of rejections will be negligible and not affect the efficiency. In the case of the exact values, we are simply sampling from a known distribution, and can use the inverse transform method. Alternatively, to avoid the need to post-select on the uniform sampling part, the probability can be split into two by including the uniform contribution of all states in one group and the *extra* probability of the exact values in the other group. In this case, when sampling from the uniform part all outcomes are valid, and when sampling the exact values it is the probability *above* the uniform level that is included for each outcome (with the probability below accounted for when sampling from the uniform part).

This peaked distribution is easy to sample from, although slightly slower than the uniform distribution, due to the extra step of sampling which group (uniform or exact values) to choose. However, the height of the uniform part can never be higher than in the stepped uniform case, but may be substantially lower thus increasing the probability of accepting a proposed pattern. The main concern is that it requires more knowledge of the highest values which might be more difficult.

3.2.2.3 Adaptive rejection sampling

A proposal distribution is better the closer it is to the target distribution. Each time we try to sample with rejection sampling, whether an outcome is accepted or rejected, we can use a different proposal distribution because each experiment is independent. When using rejection sampling, every time a state is rejected the target probability is calculated and this additional information can be used to improve the sampling rate. For each probability that is calculated, the proposal distribution can be updated to replace the proposal probability with the target probability [66]. Gradually over time the proposal distribution will become closer to the target distribution, and the probability of accepting the proposed state will increase. This will result in a hybrid envelope distribution with a subset of the sample space having probability $f(x)$ and the rest of the sample space given by $Hg(x)$. It is important that it is still easy to sample from this hybrid distribution

of $Hg(x)$ and $f(x)$. We note this is a generalisation of the peaked distribution where states with probabilities $f(x)$ are equivalent to the peaks and those with $Hg(x)$ is a more general proposal distribution than the uniform part of the peaked distribution. So in general it is always possible to first determine whether to sample the states given with envelope probability $f(x)$ or $Hg(x)$, and then sample from $f(x)$ states using the inverse transform method for the known distribution and from the $Hg(x)$ states as before. It may be possible to sample from $Hg(x)$ restricted to the states in that group, however in general that cannot be assumed and we need to post-select on outcomes in that group. For both the uniform and peaked distributions, we would need to post-select and this can substantially reduce the efficiency if the probability contained in the states in the $Hg(x)$ subset is small.

This technique can have a substantial speed-up if enough probabilities are replaced by the target probability and the unknown values can be sampled efficiently. However, if the majority of states are replaced this means the number of probabilities calculated is sufficiently large that rejection sampling is no more efficient than brute force sampling. Therefore we should assume that we will only be calculating a small number of the probabilities and in this regime the speed-up is at best very small. As such we choose not to adapt the proposal distribution to avoid complications arising from the sampling of the hybrid proposal distribution, given the speed-up will be minimal in the regime that rejection sampling is useful.

3.2.3 Hill climbing optimisation algorithms

For both of the suggested proposal distributions it is necessary to find at least an estimate of the maximum probability or in the case of the peaked distribution the N maxima. Hill climbing [67] is a local search algorithm to maximise a criterion and we employ it to maximise the probability of an output pattern by searching neighbours within the sample space. The choice of how to determine which output patterns are neighbours is open, however there are two natural ways to do this. For the first way, we write a pattern with notation corresponding to the first quantisation, where there are N elements that can take M values, $(m_1, \ldots, m_N)$. It is possible to find a neighbouring state by simply moving one photon to another mode by changing one element in the pattern. In this problem, the variable is N-dimensional with neighbours defined for each dimension; that is for dimension d, the corresponding variable is m_d and the neighbours are all M states with all other m_i fixed. The second way to define neighbours would be to instead work in the second quantisation, with M elements that can each take a value from 0 to N_c. A neighbour for each dimension can be defined in a similar way by changing one element, but this now corresponds to adding or removing photons from a particular mode. As such the number of photons is not conserved and this method would search over the whole sample space rather than for a fixed photon number. For the proposal distributions we need to know the maximum/maxima in each photon number and therefore we choose to use the first quantisation neighbours and run the algorithm for each photon number.

In hill climbing a starting state is chosen, and then each variable is improved in turn with all others fixed until they all have been improved at which point the process is repeated from the first variable but now with the other fixed variables potentially having different values compared to the

previous time. A **pass** is a single round of updating each variable in turn and the process is repeated for many passes until there is no improvement at which point it terminates. As this process is a local search (searches over only the neighbour states each time) it may not find the global maximum and get stuck in a local maximum instead. The final value depends on the starting state (and in some cases a sequence of random numbers). The general algorithm is outlined in algorithm (2).

Algorithm 2 Hill climbing

Input: M, N, the number of modes and photons
Output: $x = (m_1, \ldots, m_N)$
  Sample a starting pattern $x_0 = (m_1, \ldots, m_N)$ with probability p_0
  while True **do**
    for $n = 1$ to N **do**
      Update m_n, with all other m_i fixed, according to the update rule to find a new pattern x_n with corresponding probability p_n
    if $p_n = p_{n-1}$ **then**
      Break out of while loop
  return x_N

As the choice of starting state can determine whether a local or global maximum is found, it is beneficial to optimise the starting state. However, finding which state is the best to start at is not trivial and to overcome the problem of potentially starting in a poor choice, we can instead run the algorithm a few times starting at a different random state in each case. This is called random restart hill climbing, and the amount of times that a new starting state is chosen is a free parameter that should be chosen carefully to balance the trade-off in increased probability of finding the maximum with reduction in efficiency when increasing this number. The criteria for updating each variable can also be chosen and we consider two variants of hill climbing that we employ for GBS: steepest ascent and stochastic hill climbing.

3.2.3.1 Steepest ascent hill climbing

Steepest ascent is a variant of hill climbing that is a greedy algorithm always choosing the neighbour that maximises the probability. Hence the update rule in this case is to select the mode m_n such that $\Pr(m_1, \ldots, m_n, \ldots, m_N) > \Pr(m_1, \ldots, m_i, \ldots, m_N)$ for all $i \neq n$. In order to do this, it is necessary to calculate the probabilities for all m_i. This method is deterministic for a given starting state, always outputting the same end state. Since it continues to always choose the best improvement, once it has found the local maximum there will be no further improvements and hence the algorithm stops. This guarantees ending in a local maximum but not necessarily the global. Therefore we use random restarts to find a few local maxima increasing the probability of finding the global maximum but also meaning it is more likely to find more of the N maximum values, which may be local maxima or neighbours of various local maxima.

The number of probabilities we need to calculate in each pass is MN because for each of the N photons we need to know the probability of finding it in each of the M modes. The number of passes $|P|$ is unfixed and depends on how long it takes for the maximum found to remain unchanged,

and the number of restarts $|R|$ can be chosen arbitrarily. So the total number of probabilities to calculate is $|R||\bar{P}|MN$, where $|\bar{P}|$ is the average number of passes across all restarts.

3.2.3.2 Stochastic hill climbing

Stochastic hill climbing is a variation on steepest ascent hill climbing which climbs the curve non-deterministically and so will not always take the same path. Therefore we are less likely to become stuck in a local maximum. The non-determinism comes from when we select which mode to fix the photon in. In the previous hill climbing we always chose the mode which maximised the probability, however, here we will sample which mode to choose with a weighted probability based on the values for each pattern, such that we are more likely to choose the pattern with the highest GBS probability and we never choose a pattern which lowers the GBS probability. The exact function which maps the GBS values to the probability of accepting that pattern is not fixed and can be chosen based on each problem. We choose to accept a mode with probability proportional to the GBS probability of that pattern, restricted to patterns that have a greater probability than the current pattern.

The number of probabilities calculated here is $|R||\bar{P}_s|MN$, where $|\bar{P}_s|$ is the average number of passes in the stochastic case. In general $|\bar{P}_s| > |\bar{P}|$ because the stochastic variant will find higher probabilities more slowly as it doesn't necessarily accept the maximum. This wider search of the sample space also means that it is more likely to find the global maximum from any starting state which reduces the required number of restarts.

For the peaked distribution, we need to find the N maxima. To do this we can keep the highest N probabilities found at any point during the algorithm whether it was accepted or not. It is intuitive that to find more maxima it will require more probabilities to be calculated in general. Therefore we expect the hill climbing will need to run for longer for the peaked distribution.

3.3 Numerical analysis of the accuracy and efficiency of Gaussian boson sampling with rejection sampling

For both of the suggested proposal distributions above, stepped uniform and peaked, it is important to know the maximum or N maxima probabilities for each photon number respectively. The hill climbing algorithms presented above aim to find these and if they successfully do this, the rejection sampling algorithm is exact and the accuracy is perfect. In the case where they do not find the global maximum/maxima, some of the distribution is missed when sampling. This causes a non-zero total variation distance (TVD) between the target distribution $f(x)$ and the distribution we sample from, which we denote $\tilde{f}(x)$. It is important to know that $\tilde{f}(x)$ is a good approximation to the target distribution. There is a trade-off between accuracy and efficiency with hill climbing algorithms where the number of probabilities calculated can be chosen and increasing this will generally increase the probability of finding the maximum/maxima but also the time it takes to do so. Therefore care needs to be taken when fixing the number of probabilities such that there is a good balance between accuracy and efficiency. We wish to determine whether steepest ascent or

stochastic hill climbing is better for our application and how long they need to be run for any size GBS problem. We begin by analysing the accuracy of both algorithms for a range of problem sizes.

3.3.1 Comparison of hill climbing algorithms

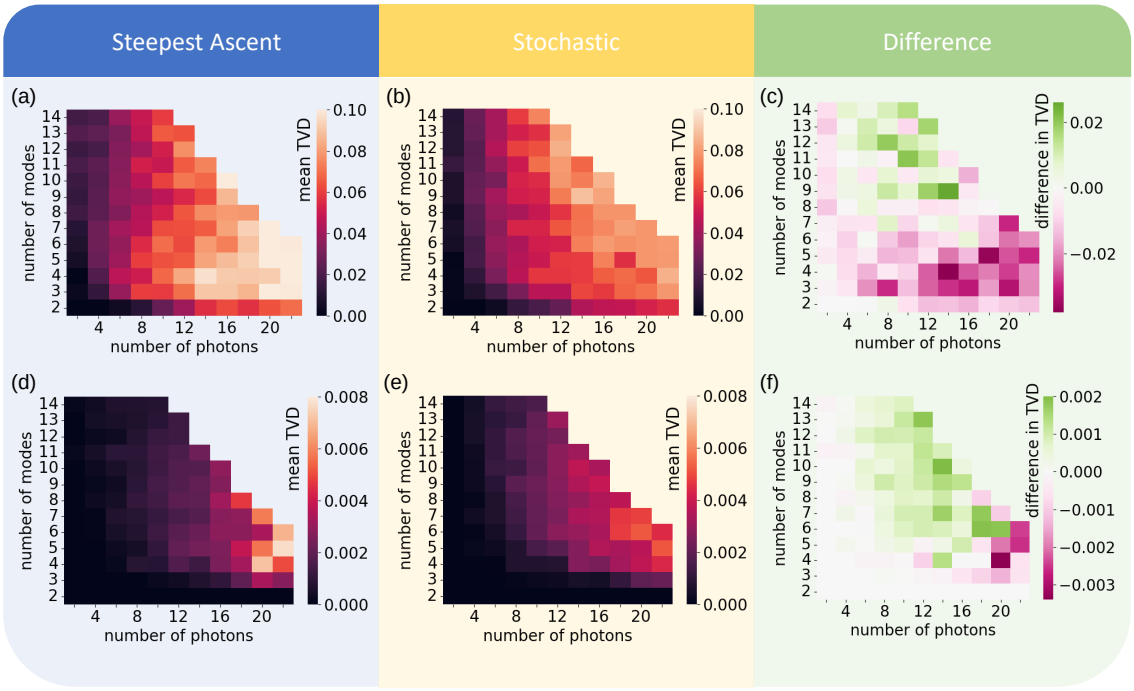

Figure 3.5: TVD for stepped uniform distribution. These heat maps represent the mean TVD between the target and sampled distributions as a function of the number of modes and photons when using the maximum as found by a hill climbing algorithm. The left column (plots (a) and (d)) shows the TVD when using the steepest ascent algorithm and the middle (plots (b) and (e)) for the stochastic algorithm. The right column (plots (c) and (f)) shows the difference (stochastic - steepest ascent) between these two graphs to highlight which algorithm performs better - green means that the steepest ascent has a lower TVD while pink represents where the stochastic algorithm is better. The top row ((a), (b) and (c)) are plots when terminating the hill climbing algorithms after finding 50 probabilities and the bottom row ((d), (e) and (f)) are for after 2000 probabilities. All data points are found by averaging over 100 Haar random unitaries.

We can measure the accuracy of the hill climbing algorithms in terms of either the probability of finding the maximum/maxima or the average TVD this causes between the target and sampled probability distributions. We look at both of these metrics and compare the accuracy for both hill climbing algorithms to determine which one is better for our application. For a fair comparison between steepest ascent and stochastic hill climbing, either the number of probabilities calculated are fixed and the accuracy is measured or vice versa. The first option is more convenient as it fixes the number of hafnians ensuring a finite time and it is not necessary to calculate the accuracy at every point. Therefore this is the option we choose.

The size of the sample space and the shape of the probability distribution (such as the number of

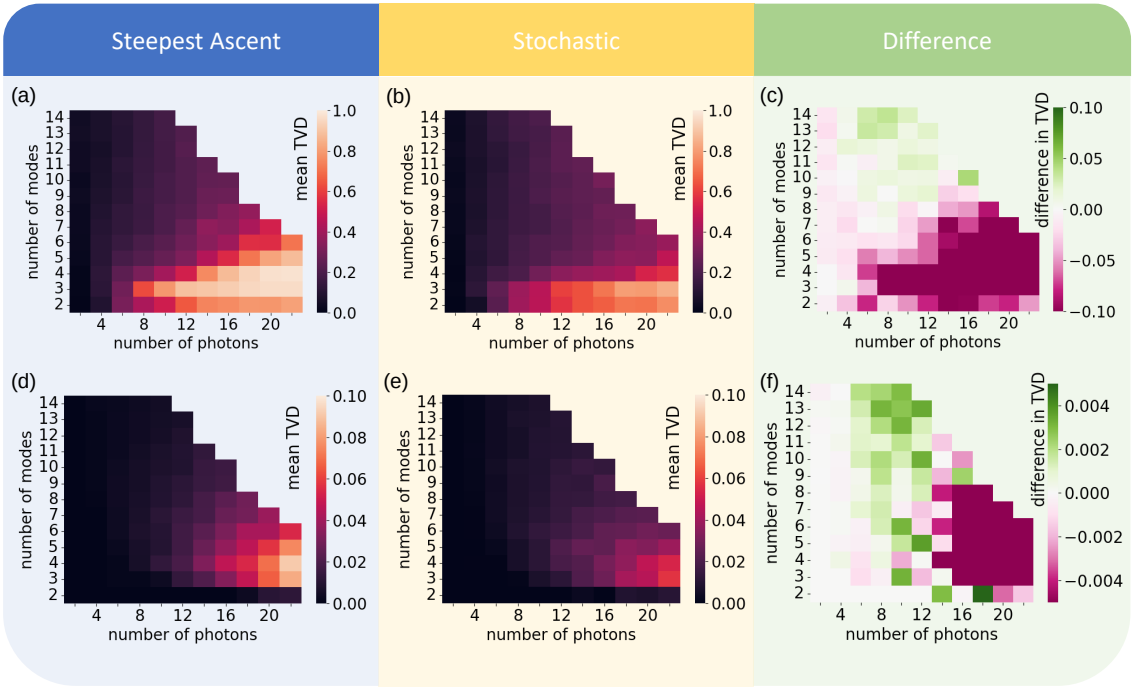

Figure 3.6: TVD for peaked distribution. These heat maps represent the mean TVD between the target and sampled distributions as a function of the number of modes and photons when using the maxima as found by a hill climbing algorithm. The plots correspond to the same scenarios as in fig. (3.5) but for the peaked distribution.

local maxima) depend on the number of modes and photons and so it is likely that the effectiveness of the hill climbing algorithms also depend on these variables. We are interested in rejection sampling, and therefore hill climbing, for GBS problem sizes where we cannot sample by brute force. However, it can be useful to estimate the required number of probability calculations for some accuracy for small problem sizes and extrapolate. Ensuring that the error does not increase with problem size in the regime that can be directly tested is important as it suggests that the error will not become large in the regime that cannot be tested. For the problem sizes that are small enough for the full distribution to be calculated, the accuracy of the hill climbing algorithms can be predicted by calculating the actual accuracy for many Haar random unitaries and finding the average. It is a reasonable assumption that although the number of probabilities required for the same accuracy will scale with the number of photons and modes, it will be relatively independent of the unitary. It is also possible that the relative squeezing and displacement input on each mode will affect the shape of the probability distribution. As it is impossible to test the accuracy for all infinite combinations of the squeezing and displacement, we test here for the case of no displacement and equal squeezing on all modes. Note that the value of the squeezing affects the total number of photons distribution but not the distribution of the patterns for a fixed photon number. This is because for equal squeezing on all modes, the squeezing can be factored out of the hafnian in the probability, which is the only part that has dependence on the output pattern. In all numerical

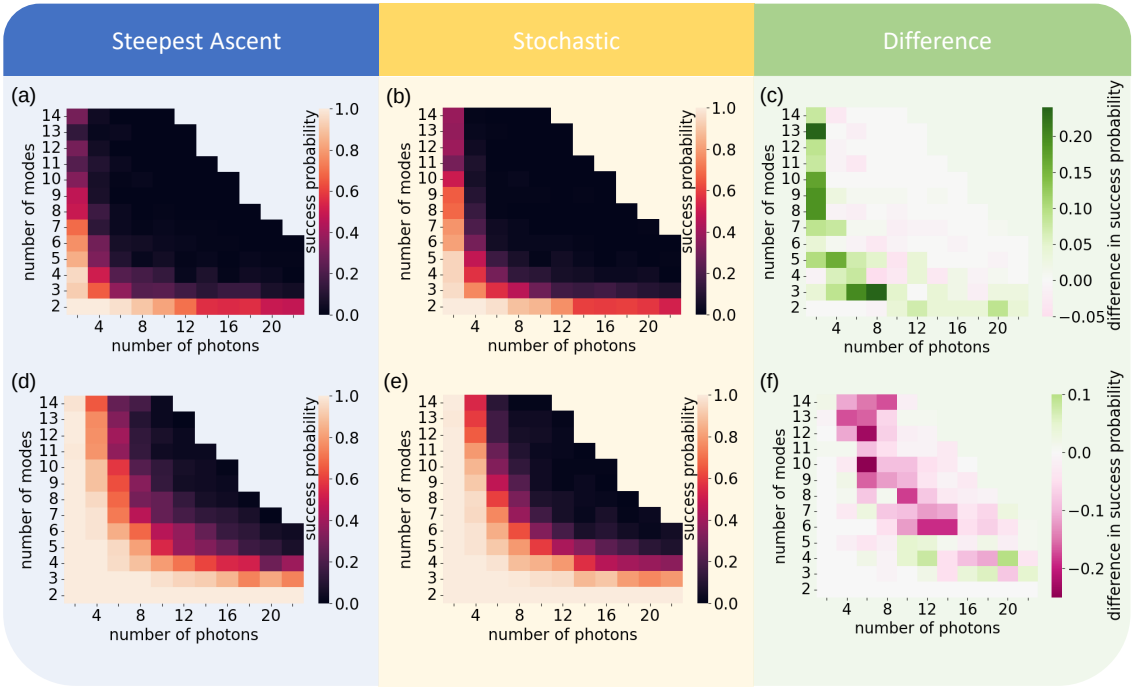

Figure 3.7: Success probability for stepped uniform proposal. These plots show the average probability of the hill climbing algorithms successfully finding the global maximum for a variety of problem sizes. As the number of modes or photons increases, this probability quickly falls to almost zero. The plots correspond to the same scenarios as in fig. (3.5).

simulations the squeezing parameter was set to 1.

We estimated the TVD between $f(x)$ and $\tilde{f}(x)$, and the probability of finding the maximum and N maxima for the stepped uniform and peaked distributions for various GBS problem sizes from 2 to 22 photons and 2 to 14 modes. The squeezing parameter was set as 1, but as mentioned above this is an arbitrary choice that does not affect the data. For each metric of accuracy and proposal distribution, the accuracy was found after calculating 50 and 2000 probabilities in both hill climbing algorithms. Where both the number of photons and modes are high, the problem size becomes too big to calculate the full distribution and the test becomes too slow. The results are shown in figs. (3.5)-(3.8). For each problem size, the values were calculated for 100 Haar random unitaries and the mean value is plotted.

First we consider the TVD between the target and sampled distributions. Fig. (3.5) shows the average TVD for the uniform distribution. Plots (a) and (d) are for steepest ascent hill climbing, plots (b) and (e) for stochastic hill climbing and plots (c) and (f) show the difference between them to highlight which algorithm performed better. The top row (plots (a), (b) and (c)) is the average TVD after calculating 50 probabilities in the hill climbing algorithms and the bottom row (plots (d), (e) and (f)) is for after 2000 probabilities. We wish to minimise the TVD and as can be seen from the graphs, both algorithms give fairly similar results. Therefore to determine which algorithm performs better, the graphs on the right give the mean value for the stochastic variant minus those

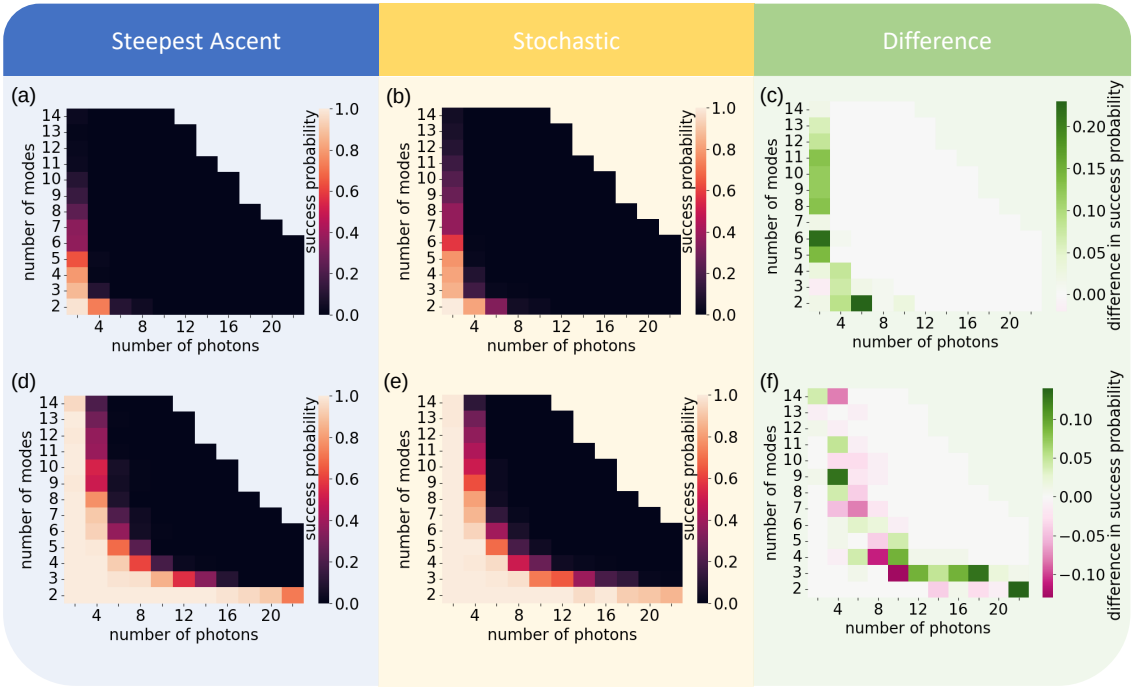

Figure 3.8: Success probability for peaked proposal. These plots show the average probability of the hill climbing algorithms successfully finding the N maxima for a variety of problem sizes. As the number of modes or photons increases, this probability quickly falls to almost zero, more quickly than for finding the global maximum as shown in fig. (3.7). The plots correspond to the same scenarios as in fig. (3.5).

for the steepest ascent. A positive value (in green) shows that the steepest ascent performed better whereas a negative value (in pink) means that the stochastic was better. Whilst neither colour is completely dominant in these graphs, there are two interesting observations. The first is that there is a grouping of the colours suggesting that in general the stochastic algorithm is preferable when there is a high density of photons per mode, otherwise the steepest ascent performs better. The second is that by comparing the top and bottom graphs, the pink is dominant after only 50 probabilities whereas the green is dominant after 2000 suggesting that stochastic is more effective more quickly but eventually the steepest ascent will find a better estimate of the maximum. We note that for 2 modes, both hill climbing algorithms are in theory exactly the same because there are only two probabilities to choose between in each step and an improved probability must be the maximum. Therefore the row for $M = 2$ in the difference graphs should be zero. This is seen well in the graph (f) but less so in graph (c). This suggests that the variance in the estimate of the TVD is too large to distinguish between the two algorithms after only 50 probabilities. There may also be differences between the two algorithms if both outputs have the same probability. Comparing the upper row to the lower row (note the difference in the scaling of the colourbars) shows that increasing the number of probabilities from 50 to 2000 reduces the TVD significantly as expected. There is an interesting pattern with regard to the number of modes and photons. In general the

TVD increases as the number of photons increases but may actually decrease with the number of modes. The increase with the number of photons can be explained as it increases the number of dimensions in the output pattern and therefore the number of local maxima will likely increase meaning it is easier to get stuck in one rather than finding the global maximum. The decrease with increasing number of modes is due to the increase in size of the sample space meaning that the average probability of each pattern is lower. Therefore in cases where the global maximum is not found, the next largest values are very close and so not as much probability is above the envelope distribution. Of course increasing the number of photons also increases the size of the sample space but it is counteracted by the increase in local maxima. It is important to note that the size of the sample space is $\binom{M+N-1}{N}$ and therefore the effect of increasing M or N is very dependent on the relative values of $M+N-1$ and N.

The same plots are given for the peaked proposal distribution in fig. (3.6). Here it is even more pronounced that the TVD is significantly different in the photon dense regime with both algorithms suffering when there is a high amount of photons per mode. A TVD of close to one occurs if not enough different probabilities have been calculated to assign all N peaks and the uniform distribution is set at the lowest which is zero. The difference plots, (c) and (f), show similar patterns to the stepped distribution but we have lower bounded the values by -0.1 and -0.005 in order that the smaller differences in the photon sparse regime can be seen. In the photon dense problem size, the stochastic algorithm is significantly better but in the photon sparse problems, there is a slight preference to the steepest ascent variant, again especially after 2000 probabilities. For the same number of probabilities calculated, when compared to the stepped uniform distribution the TVD is much larger which will result in a worse approximation when using the peaked distribution in rejection sampling. This is to be expected as this proposal distribution requires the knowledge of all N maxima and sets the uniform level at the smallest of these values. As there are more values to find it is more likely to miss a few of them and therefore the Nth maxima found is actually much lower than it should be.

As well as analysing how approximate the rejection sampling is when using the hill climbing algorithms, it it interesting to consider the probability that the rejection sampling is exact. Figs. (3.7) and (3.8) show the estimated probability of finding the maximum (for uniform proposal) and N maxima (for peaked proposal) respectively. As before, the left column (plots (a) and (d)) is the data when using steepest ascent, the middle (plots (b) and (e)) for stochastic and the right (plots (c) and (f)) for the difference between them. The top row (plots (a), (b) and (c)) is after allowing the hill climbing to run for 50 calculated probabilities and the bottom (plots (d),(e) and (f)) for 2000. Contrary to the previous plots, the higher the value (and therefore lighter the colour), the better the algorithm. In all cases as the number of modes or photons increases, the probability of exact rejection sampling goes quickly to zero. As expected, increasing the number of calculated probabilities does increase the probability of finding the maximum/maxima but it is still essentially zero for a moderate problem size after 2000 probabilities. The probability to find the maximum is better than to find all N maxima. This is to be expected as successfully finding the N maxima guarantees that the maximum has been found as well as the other $N-1$ maxima. We want to

determine which of the two hill climbing algorithms to use. For this data, if the difference is positive (green), the stochastic algorithm was better. Again, neither algorithm was always better than the other but for the uniform proposal there is a leaning towards pink with the knowledge of more probabilities suggesting that the steepest ascent is more likely to find the maximum. In the case of the peaked distribution, there are few problem sizes that don't have a probability of 0 or 1 of finding the maxima for both algorithms and therefore there is almost no difference between the algorithms in most cases. For those where there is a difference, the stochastic hill climbing algorithm performs better after 50 probabilities, but it is very equal after 2000 probabilities.

Overall it is less important to sample exactly as long as the approximation is good. Therefore when determining the hill climbing algorithm and the number of probabilities to find the maximum/maxima, the most important metric to consider is the TVD between the target and sampled distributions. The behaviour is very different depending on the ratio of photons to modes with the stochastic algorithm generally performing better in the photon dense regime and the steepest ascent preferred in the photon sparse regime. Therefore we choose to use the stochastic algorithm if $2M < N$ and the steepest ascent otherwise. We refer to this as the 'hybrid hill climbing' algorithm and use it for the rest of the numerical testing.

3.3.2 Efficiency of hybrid hill climbing

We wish to determine how many probabilities we need to calculate when running the hybrid hill climbing to ensure a small TVD between target and sampled distributions. As the plots in figs. (3.5) and (3.6) confirm, the effectiveness of the hill climbing algorithm depends on the number of modes and photons so we need to determine how to scale the number of probabilities calculated with these variables. To do this we run numerical tests to find the average number of probabilities required to reach a fixed TVD as a function of modes and photons.

We choose to find the minimum number of probabilities needed to be calculated in the hybrid hill climbing algorithm such that the TVD is less than 0.05, averaged over 100 Haar random unitaries. In order to find the number of probabilities at which the TVD is lower than a threshold value, it can require running the hill climbing for a long unknown time. Therefore it is necessary to put an upper limit on the number of probabilities to try, and we chose to set 10,000 as the limit for these simulations. For the problem sizes we tested, for the uniform distribution this was always high enough and for the peaked distribution there was only one case where the TVD did not reach 0.05. This was for 16 photons in 4 modes and the final value was 0.0527 and so any distortion to the data due to this upper limit is negligible. The data is plotted in fig. (3.9). The scaling of the required number of probabilities as a function of modes and photons is shown in the top and bottom rows respectively. The left column plots are when finding the global maximum for the stepped uniform distribution and the right for the N maxima for the peaked distribution. Each point is averaged over 100 instances with different Haar random unitaries and the error bars show the standard deviation.

First we consider the uniform distribution. As can be seen from the upper left plot, there is no significant pattern to how the number of required probabilities scales with the number of modes,

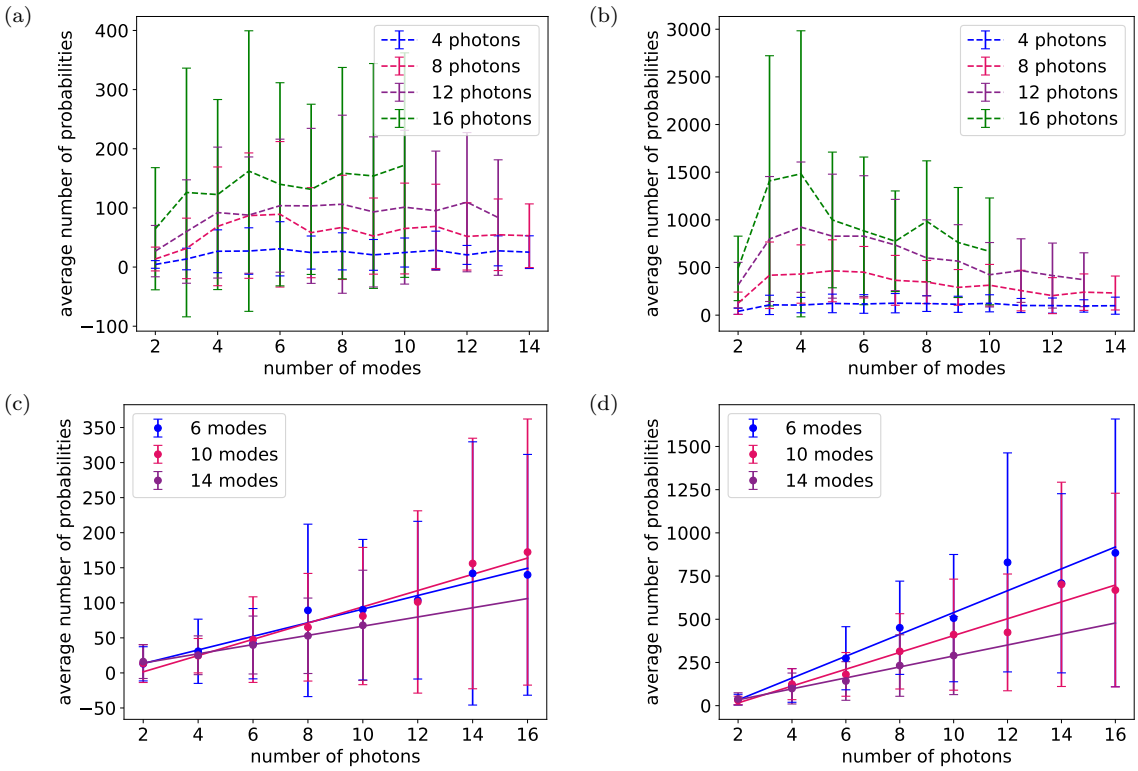

Figure 3.9: The efficiency of hill climbing. These plots show the average number of probabilities needed to be calculated for the TVD to fall below 0.05. Plots (a) and (b) show this as a function of the number of modes, while (c) and (d) give it as a function of the number of photons. The left column (a and c) is when using the stepped uniform proposal distribution and the right (b and d) is with the peaked proposal.

with it being approximately flat beyond 4 modes for all tested numbers of photons (4, 8, 12 and 16). The error bars are high which shows that for different Haar random unitaries and also different starting states for the hill climbing, its effectiveness varies significantly. There is however a clear pattern with an increase in photons causing an increase in the number of probabilities to calculate. This relationship is plotted in the lower left graph for several numbers of modes (6, 10 and 14). Despite the error bars being high, the mean values follow a linear trend quite closely and so we fit a line to each set of data points. These are given by $f(N) = 9.71N - 6.07$, $f(N) = 11.59N - 21.65$ and $f(N) = 6.57N + 0.98$ for $M = 6, 10$ and 14 respectively. The plots for $M = 6$ and $M = 10$ are similar but the gradient for $M = 14$ appears to be lower. This decrease in gradient with the increase in the number of modes shows more clearly that the top left plot must actually be decreasing beyond $M = 10$ even if it is not obvious by eye. It seems that $M = 10$ is approximately the most difficult number of modes and using the scaling found for $M = 10$ will result in a smaller TVD for higher modes. It is not possible to numerically find the relationship between the number of photons and the required number of probabilities for all M. Therefore, we choose the worst-case scenario and set the number of probabilities to calculate in the hill climbing for the uniform proposal distribution as $11.59N - 21.6$ for all other numbers of modes.

Now we look at the peaked distribution. From the upper right plot, there is a more clear pattern of rising until approximately 4 modes after which it decreases again. The lower right plot again shows a linear trend with increasing the number of photons with the steepest gradient now corresponding to the $M = 6$ case. The linear fit for the peaked distribution is given by $f(N) = 63.22N - 93.27$, $f(N) = 73.00N - 81.64$ and $f(N) = 31.75N - 29.82$ for $M = 6$, 10 and 14 respectively. This means that approximately 5 times as many probabilities need to be calculated to find the N maxima than the global maximum for the same TVD when using the respective proposal distributions. We again choose the worst-case scenario and set the number of probabilities to calculate in the hill climbing for the peaked proposal distribution as $63.22N - 93.27$ for all numbers of modes.

The above analysis is a good tool to predict the scaling of the accuracy of rejection sampling on average. However, the actual accuracy will depend a lot on the target distribution and the particular path taken in hill climbing, as shown by the large error bars in the plots. The target distribution depends on many factors: the unitary, squeezing and displacement parameters, and the path of the hill climbing depends on the starting state, and random sampling in the case of the stochastic variant. Therefore this prediction does not ensure that the TVD is guaranteed to be low if the derived parameters are used. We would like to be able to estimate post-sampling the accuracy of the sample found for a particular instance of simulating GBS with rejection sampling. We now investigate a method for estimating the TVD between the target and sampled distributions that can be used to check the accuracy after the sampling.

3.3.3 Post-sampling analysis of the accuracy of rejection sampling

The aim is to estimate the TVD between the target distribution $f(x)$ and the sampled distribution $\tilde{f}(x)$ after sampling. It is useful to formalise the link between the TVD and the area of the target distribution contained under the envelope distribution, A. For any two distributions, $q(x)$ and $r(x)$, the TVD between them is given by half the sum of the absolute differences for each outcome (eq. 3.10). So we can group the outcomes by whether $q(x) > r(x)$ or $q(x) < r(x)$ and given that both distributions sum to the same total probability, the total difference in the first group must be equal and opposite to the total difference in the second group. So the TVD is equivalent to the total difference in one of these groups. This is just the probability of one distribution not under the other one. In fig. (3.3), the TVD between the black $f(x)$ and orange $\tilde{f}(x)$ distributions in (c) is by definition half the sum of the blue and red shaded areas. These areas must be equal and so

the TVD is given by the area of either of them. We can also show this result mathematically:

$$\begin{aligned}\mathrm{TVD}(q,r) &= \frac{1}{2}\left(\sum_{x\in X^+}(q(x)-r(x)) + \sum_{x\in X^-}(r(x)-q(x))\right)\\ &= \frac{1}{2}\left(\sum_{x\in X^+}(q(x)-r(x)) + \sum_{x\in X^-}(q(x)-r(x)) - \sum_{x\in X^-}(q(x)-r(x)) + \sum_{x\in X^-}(r(x)-q(x))\right)\\ &= \frac{1}{2}\left(\sum_{x\in X}(q(x)-r(x)) + 2\sum_{x\in X^-}(r(x)-q(x))\right) = \sum_{x\in X^-}(r(x)-q(x)),\end{aligned} \tag{3.14}$$

where $X^+ = \{x : q(x) > r(x)\}$, $X^- = \{x : q(x) < r(x)\}$, and $X = X^+ \cup X^-$. In the last line, we used that both distributions are normalised so $\sum q(x) = \sum r(x) = 1$ and hence the first term cancels out. We can rewrite this in terms of the area of probability under both distributions, $A_{q,r}$:

$$\begin{aligned}\mathrm{TVD}(q,r) &= \sum_{x\in X^-}(r(x)-q(x)) = 1 - \sum_{x\in X^+} r(x) - \sum_{x\in X^-} q(x)\\ &= 1 - \sum_{x\in X}\min\{r(x),q(x)\} = 1 - A_{q,r}.\end{aligned} \tag{3.15}$$

This is equivalent to the probability in $r(x)$ above $q(x)$ and vice versa. So it is possible to find the TVD between two distributions if the area under both distributions is known. In our case, we want to find the TVD between the approximate distribution, $\tilde{f}(x)$, and the target distribution, $f(x)$, without knowing either distribution fully.

From section (3.1.2), we know that the area under both the target and *envelope* distributions, A, is related to the probability of accepting a proposed sample, $p_{\text{accept}} = A/H$. However, to calculate the TVD between the target and *sampled* distributions it is the area under these distributions $A_{f,\tilde{f}}$ that needs to be known. To find this area we need to consider for which outcomes the target distribution is greater than the sampled one. There are three different possible cases, which are shown in fig. (3.3). The first case is the outcomes which are sampled exactly (ie. $f(x) \leq Hg(x)$) for which the sampled probability is $f(x)/A$ according to eq. (3.11). Given $A \leq 1$, this implies $\tilde{f}(x) = f(x)/A \geq f(x)$. So the TVD contribution in these cases is $f(x)/A - f(x)$. We denote these outcomes as $x \in B$. The other cases are when the outcomes are sampled approximately (ie. $f(x) > Hg(x)$) for which the sampled probability is $Hg(x)/A$ which may be greater than or less than $f(x)$. So for cases 2 and 3, outcomes contribute $Hg(x)/A - f(x)$ (which we denote as $x \in C^-$) and $f(x) - Hg(x)/A$ (denoted $x \in C^+$) respectively to the TVD. The only case where the target distribution is greater than the sampled one is case 3 so the sum of the outcomes $x \in C^+$ gives the correct TVD. However, it is not possible to know the values of these outcomes or even which outcomes are in case 3 without knowing the entire target distribution which of course would defeat the purpose of using rejection sampling. Therefore, we look at a couple of ways of estimating the TVD by sampling.

3.3.3.1 Upper bound for the TVD between target and sampled distributions

First, we consider trying to find an upper bound to the TVD in terms of quantities that are easier to evaluate. We can upper bound the TVD as

$$\text{TVD}(f(x), \tilde{f}(x)) = \sum_{x \in C^+} f(x) - Hg(x)/A \leq \sum_{x \in C^+} f(x) - Hg(x) \leq \sum_{x \notin B} f(x) - Hg(x) = 1 - A, \tag{3.16}$$

where $B = \{x : f(x) \leq Hg(x)\}$, $C^+ = \{x \notin B : f(x) > Hg(x)/A\}$ and $C^- = \{x \notin B : f(x) < Hg(x)/A\}$. As before, A is the area under both the target and envelope distributions. The first inequality uses the fact $1/A \geq 1$ and is the equality only when $A = 1$ ie. in the exact case. The second inequality comes from the fact that $f(x) - Hg(x) \geq 0$ in C^- and so adding these terms can only increase the value. It is only the equality if $C^- = \varnothing$ (the empty set) which is more likely the closer the sampling is to being exact. So in approximate sampling TVD is bound by the probability that is missed in selecting the scaling of the proposal distribution. The value of A can be found using equation (3.8), if H and p, the probability of accepting, are known. We can assume H is already known as this is chosen when forming the envelope distribution, but p is unknown.

However it is possible to estimate the probability of accepting a state by taking a large sample size and recording the proportion of times the proposed state was accepted. In the limit of an infinite sample size, the fraction of times we accept, R_a, approaches the expected acceptance rate, $\langle R_a \rangle$, which is equal to the probability of accepting, p, ie. $R_a \approx \langle R_a \rangle = p$. We note that proposed samples that are rejected as well as those that are accepted contribute to the sample size for this test, rather than only accepted patterns contributing to the sample size when sampling from the target distribution. Hence the accuracy of p depends on the number of times we test a proposed pattern to be accepted, which may be substantially higher than the number of times we accept a pattern. So for a fixed GBS sample size, the sample size for this test is not fixed and sometimes the estimate will be more accurate than others.

When estimating the value of p, the error due to a finite sample size is given by

$$|p - R_a| \leq E = Z\sqrt{\frac{p(1-p)}{N_s}}, \tag{3.17}$$

where Z is the confidence level, and N_s is the sample size. We use the Wilson score interval technique [68] to solve for p in terms of R_a.

$$\begin{aligned} |p - R_a| &\leq Z\sqrt{\frac{p(1-p)}{N_s}} \\ (p - R_a)^2 &\leq Z^2\frac{p(1-p)}{N_s} \\ p^2 + R_a^2 - 2pR_a &\leq \frac{Z^2}{N_s}(p - p^2) \\ p^2(1 + Z^2/N_s) + p(-2R_a - Z^2/N_s) + R_a^2 &\leq 0 \end{aligned} \tag{3.18}$$

Solving this for the equality will give the upper and lower limits for p as

$$p = \frac{2R_a + Z^2/N_s \pm \sqrt{(2R_a + Z^2/N_s)^2 - 4(1 + Z^2/N_s)R_a^2}}{2(1 + Z^2/N_s)}. \tag{3.19}$$

We choose to find a worst-case scenario for the TVD between the target and sampled distributions and so find the upper limit on the estimate of the upper bound. To do this we need to find a lower limit on A so that TVD $\leq 1 - A \leq 1 - A_{\text{LL}}$. The subscripts UL and LL refer to the upper and lower limits respectively. A is estimated using the relation $A = Hp$ (eq. 3.8) and therefore a lower limit for the probability of accepting, p, must be determined. Explicitly,

$$\text{TVD} \leq 1 - A \leq 1 - A_{\text{LL}} = 1 - Hp_{\text{LL}}. \tag{3.20}$$

So the lower limit of p from eq. (3.19) can be found from the acceptance rate when sampling and then substituted into the above equation along with the known value of H to find an estimate of the upper bound for the desired TVD.

3.3.3.2 An estimation of the exact TVD between target and approximate distributions

In the above section we estimated an upper bound for the TVD which only depends on the area under the target and envelope distributions and the value of the scaling parameter H, avoiding needing to know anything about the sampled distribution. However, if we know A it is possible to calculate the sampled probabilities and in the previous method an estimate of A was found. This suggests we could use this estimate to estimate the exact value for the TVD. From eq. (3.15), the TVD between two distributions is given by $1 - A_{q,r}$, the area not below both of them and we describe a method here to estimate $A_{f,\tilde{f}}$.

Suppose we have rejection sampling with the proposal distribution as $g(x)$ but the scaling constant as $H/A = 1/p$ instead of H. This would have the acceptance probability

$$\Pr(\text{accept } x|\text{proposed } x) = \min\left\{1, \frac{f(x)}{g(x)/p}\right\}. \tag{3.21}$$

So the overall probability of accepting is

$$\begin{aligned} p_{f,\tilde{f}} &= \sum_x \Pr(\text{propose } x)\ \Pr(\text{accept } x|\text{proposed } x) = \sum_{x\in B\cup C^-} g(x)\frac{f(x)}{g(x)/p} + \sum_{x\in C^+} g(x) \\ &= \frac{1}{1/p}\left(\sum_{x\in B\cup C^-} f(x) + \sum_{x\in C^+} g(x)/p\right) = \frac{A_{f,\tilde{f}}}{1/p}, \end{aligned} \tag{3.22}$$

where $A_{f,\tilde{f}}$ is the area of $f(x)$ under the sampled distribution. So now we can find the TVD exactly

in terms of $p_{f,\tilde{f}}$ and p:

$$\mathrm{TVD}(f(x), \tilde{f}(x)) = 1 - A_{f,\tilde{f}} = 1 - \frac{p_{f,\tilde{f}}}{p}. \tag{3.23}$$

The value of p can be estimated from the acceptance rate in rejection sampling as in the previous section, but we also need to estimate $p_{f,\tilde{f}}$. This can be done by rejection sampling with the above acceptance rule. Note that we only use this rule for estimating the TVD, not for sampling from the target distribution. However, we will already need to calculate $f(x)$ for sampling from the target GBS distribution and the rest of the calculation is efficient. Therefore additionally keeping a count of the proportion of proposed outcomes that *would* be accepted under this rule has a negligible impact on the speed of the rejection sampling algorithm. As this acceptance rule requires knowledge of p, we first need to estimate p as before. The error on p is carried through when estimating $p_{f,\tilde{f}}$ and amplified. If the upper limit on p is used such that some value $p_{\mathrm{UL}} = \alpha p$, where $\alpha > 1$, only the outcomes in $B \cup C^-$ are scaled in $p_{f,\tilde{f}}$ and so

$$\begin{aligned} p_{f,\tilde{f},\mathrm{UL}} &= \sum_{x \in B \cup C^-} p_{\mathrm{UL}} f(x) + \sum_{x \in C^+} g(x) = \sum_{x \in B \cup C^-} \alpha p f(x) + \sum_{x \in C^+} g(x) \\ &\leq \sum_{x \in B \cup C^-} \alpha p f(x) + \sum_{x \in C^+} \alpha g(x) = \alpha p_{f,\tilde{f}}. \end{aligned} \tag{3.24}$$

Therefore, if an upper limit on p is used in the acceptance rule and in the denominator when estimating the TVD, we find an upper limit on the TVD:

$$\mathrm{TVD}_{\mathrm{UL}} = 1 - \frac{p_{f,\tilde{f},\mathrm{UL}}}{p_{\mathrm{UL}}} = 1 - \frac{p_{f,\tilde{f},\mathrm{UL}}}{\alpha p} \geq 1 - \frac{\alpha p_{f,\tilde{f}}}{\alpha p} = \mathrm{TVD}. \tag{3.25}$$

A disadvantage to this method is if we don't want to store all the values of $f(x)$ as we calculate them, we must divide the sample size in two, using the first group to estimate p and the second to estimate $p_{f,\tilde{f}}$ using the approximate value of p found. Reducing the sample size for each estimation further reduces the accuracy of both the estimates of p and $p_{f,\tilde{f}}$, compared to the method to estimate the upper bound of the TVD.

3.3.3.3 Comparison of the upper bound method and exact method

Both the above methods require estimating the probability of accepting which introduces some error due to finite sample size. In the first method, we find an upper bound to the TVD, so our estimate might be overly generous. However, in the second method, we also need to estimate the probability of accepting under the adapted rule which adds more error and again this might give an overly generous estimate for the TVD. In some situations, one method may give a tighter bound than the other. For example, when the upper bound is found, the first inequality uses $A \leq 1$ and so the upper bound is closer when A approaches 1. The second inequality is also closer to equality as A approaches 1. Hence this is a good estimate when A is close to 1 and therefore the TVD is small, but not if it is big. On the other hand, as the exact method relies on estimating two values

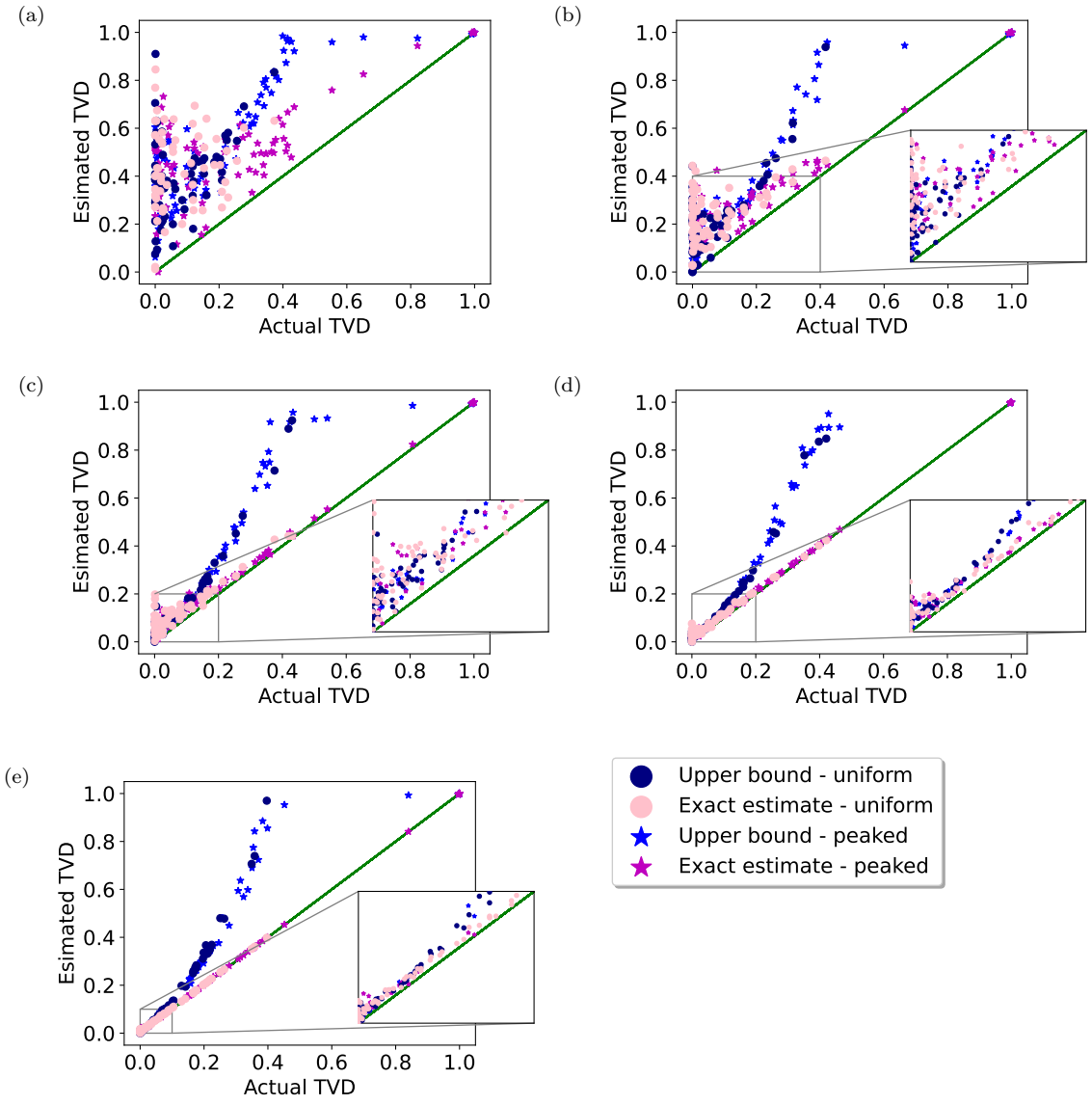

Figure 3.10: Testing the estimation of the TVD. Many data points were taken for various problem sizes to determine numerically how close the estimated TVD is to the correct value. These are plotted for the uniform distribution (circles) and peaked distribution (stars) for both the upper bound method (blues) and the exact estimate method (pinks). Plots (a), (b), (c), (d) and (e) correspond to sample sizes of 100, 1000, 10,000, 100,000 and 1,000,000 respectively. Increasing the sample sizes reduces the standard deviation as expected. The green line is $y = x$ which is the ideal relationship. The upper bound method is quite far from this for higher TVDs, but a good estimate at lower (and more useful) TVDs.

with a finite sample size, this estimate will be good for a large sample size but suffer from a small sample size. Generally we are interested in the case where the TVD is quite small, TVD < 0.1, otherwise the sampling approximation is not good enough. When the upper bound is close to the real value, the error from the upper bound method is dominated by the error due to a finite sample size in estimating p. For the exact method, we are estimating the actual TVD and so the errors always only arise from finite sample size. However, because in this method two variables need to be estimated with one depending on the other, the error due to finite sample size is likely higher for the exact method that the upper bound method. In addition, the sample size to estimate each of these variables is half compared to estimating p in the upper bound, further increasing the uncertainty.

We numerically tested the estimated TVD for various values of actual TVD and plot them for different sample sizes in fig. (3.10). In the above analysis nothing was assumed about the number of modes or photons or the proposal distribution and therefore the error in the estimation ought not to depend on these properties. The uniform proposal is marked with circles while the peaked is given by stars to distinguish between them. As expected they follow the same patterns. There is a higher concentration of the peaked events at higher TVDs, but this is only because the hill climbing was run for various numbers of probabilities (5, 10, 15, 20) in order to generate events with varying TVDs and hill climbing is less accurate for the peaked distribution as found in the previous section. Events were simulated for $N = 5$ and 10 with $M = 6$ and 12. For clarity they are not distinguished in the graphs but are equally distributed and clearly follow the same pattern.

In fig. (3.10), the green line marks where the estimated TVD is equal to the actual TVD and so the aim is to get as close to this line as possible. For a sample size of only 100 proposed states, the estimated TVD with both methods is often very far from the actual value and so cannot be used to determine the TVD. With a sample size of 1000, the spread is significantly reduced, with the deviation in the upper bound method smaller than in the exact method due to the larger sample size for estimating the variables. For the higher TVD values the upper bound is far from the correct value and so cannot be used in these cases. For a sample size of 10,000, the patterns are quite clear with the error due to sample size becoming small. For small TVDs the upper bound may be a slightly better estimate because the error due to upper bounding is very small. However, the exact estimate is also quite good for this sample size. As the sample size is increased further to 100,000 and 1,000,000, the patterns are very clear with almost no deviation for the exact method. There is still some variance in the upper bound method. This is because the tightness of the upper bound does not only depend on the actual TVD but the entire distribution.

It is not possible to know which method will guarantee the closest estimate so, given both methods do not significantly slow down the rejection sampling, we can calculate both estimates of the TVD and take the minimum value to give the best estimate. We choose 100,000 sample size to give a good estimate of the TVD. Although this is quite a lot of outcomes to test it is completely independent of the number of modes and photons and therefore does not suffer from scaling issues. Also, because this sample size takes into account all the rejected proposed outcomes, the sample size of the target distribution may be significantly lower and therefore it is reasonable that we will have this sample size in applications.

3.3.4 Complexity of Gaussian boson sampling with rejection sampling

The overall aim of this chapter is to estimate the efficiency of using rejection sampling while still sampling from a good approximation to the GBS distribution. In the previous sections we have tested the efficiency of the hill climbing algorithms and found a method to estimate the accuracy. The final piece of the puzzle is to estimate the efficiency of the accept-reject part of the algorithm. Again this can be estimated with numerical tests. The efficiency of the accept-reject part of the algorithm is the expected number of probabilities that need to be calculated per each accepted proposed state. We fixed the number of probabilities in the hill climbing as was found in the previous section for each number of modes and ran the rejection sampling algorithm to find a sample of size 100,000 for 100 instances with different Haar random unitaries. The number of probabilities calculated during the rejection sampling was recorded and from this the average number of probabilities to accept one proposed outcome was determined.

The results are plotted in fig. (3.11) for $M = 6$, 10 and 14. The left-hand side graph shows the estimated TVD using the method described previously as a function of the number of photons. Although the standard deviation is high, the average estimated TVD is approximately 0.05 for most of the data points which is the limit we fixed setting how long to run the hybrid hill climbing. This confirms that the chosen parameters for the hill climbing were good. For $M = 6$ for the peaked distribution, the TVD is a little high at approximately 0.1. This suggests that the hill climbing parameter found for this case was an underestimate, which is possible where parameters are determined numerically. The right-hand side graph gives the scaling with the number of photons of the average number of required probabilities to accept a proposed state. As expected the number of probabilities to be calculated is generally lower for the peaked distribution. For 6 modes, the average number of probabilities is very low, but as seen in the left-hand side plot, it is more approximate than the other cases and so is not a fair comparison. For 10 modes, the peaked distribution is more efficient but not significantly and for 14 modes, the distributions have very similar efficiencies. This suggests that the higher the number of modes, the less effective the peaked proposal distribution is compared to the stepped distribution.

In fig. (3.12), the same values are plotted but this time as a function of the number of modes. Because the scaling of the hill climbing parameter was only found for 6, 10 and 14 modes, for the others the worst-case of those three was used. In agreement with the plot scaling with the photon number, the peaked distribution gives a slightly high estimate of the TVD around 6 modes. The uniform seems to be more accurate with the estimated TVD closer to 0.05 as expected. The right-hand side plot again confirms that the peaked distribution is more efficient in the reject-accept part of the algorithm for all cases. However, unlike when running the hill climbing, there is a clear dependence of the efficiency on the number of modes, with more modes reducing the efficiency.

When deciding which proposal distribution is better the total number of probabilities must be taken into account. This depends on the number of photons and modes but also the sample size. If the number of probabilities in the hill climbing is h and the average number to accept one outcome is a, then the total number of probabilities t is $t = h + N_s a$. This number should be found for both proposal distributions to decide which is the best one to use.

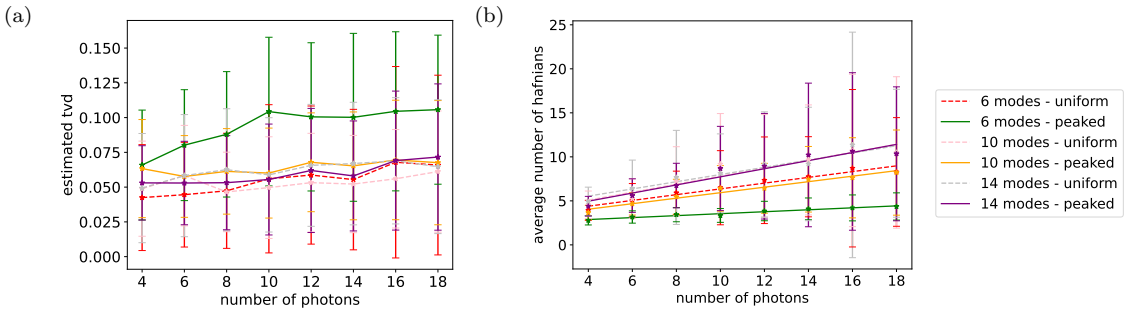

Figure 3.11: The scaling with photon number of the number of probabilities in rejection sampling. Plot (a) shows the estimated TVD when sampling with the hill climbing parameter set from previous numerical tests. The hill climbing parameter was fixed to ensure the TVD is approximately 0.05. This plot suggests that the majority of the parameters were chosen well except the peaked distribution with 6 modes which has an unexpected high TVD. The plot in (b) shows the average number of probabilities that needs to be found to output a single sample on average. For each probability a hafnian needs to be calculated and so this determines the efficiency of the algorithm. The efficiency generally follows a linear trend with the number of photons. The lines of best fit are given by the following equations: for the uniform distribution: $y = 0.33N + 3.09$, $y = 0.45N + 3.27$ and $y = 0.41N + 3.90$ for $M = 6$, 10 and 14 respectively; for the peaked distribution: $y = 0.11N + 2.43$, $y = 0.31N + 2.80$ and $y = 0.46N + 3.11$.

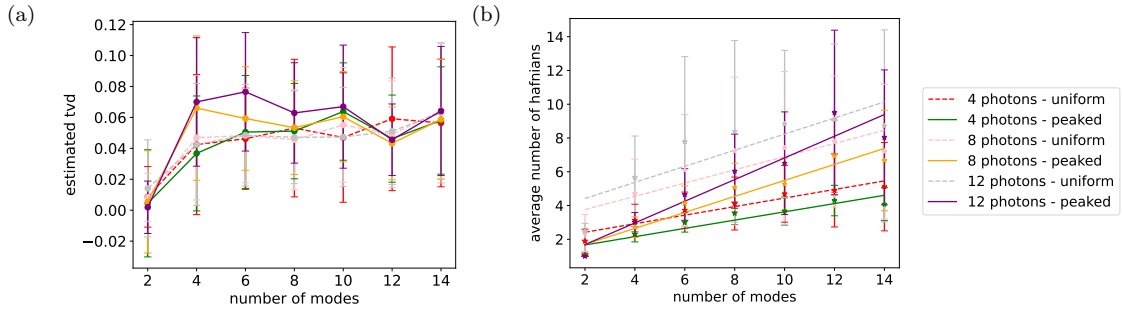

Figure 3.12: The scaling with number of modes of the number of probabilities in rejection sampling. Plot (a) shows the estimated TVD when sampling with the hill climbing parameter set from previous numerical tests. The hill climbing parameter was fixed to ensure the TVD is approximately 0.05, which is largely confirmed in this graph. The plot in (b) shows the average number of probabilities that needs to be found to output a single sample on average. For each probability a hafnian needs to be calculated and so this determines the efficiency of the algorithm. The efficiency generally follows a linear trend with the number of modes. The lines of best fit are given by the following equations: for the uniform distribution: $y = 0.25M + 1.91$, $y = 0.39M + 2.98$ and $y = 0.48M + 3.46$ for $N = 4$, 8 and 12 respectively; for the peaked distribution: $0.25M + 1.17$, $y = 0.47M + 0.75$ and $y = 0.64M + 0.40$.

3.4 Discussion

In this chapter, we have presented two possible proposal distributions for rejection sampling: the stepped distribution and the peaked distribution. From numerical testing, it seems that for both proposal distributions the maximum/maxima probabilities can be estimated in time that is linear in the number of photons and without scaling with the number of modes. The efficiency of the reject-accept part also scales linearly with the number of photons, but in addition there is linear scaling with the number of modes too. This means both proposal distributions are a feasible way to sample from a GBS distribution. In general, the peaked distribution loses its advantage in efficiency as the number of modes or photons grow and therefore it is easier to simply use the uniform distribution in general which is quicker to sample from and has a lower overhead in the hill climbing algorithm.

The accuracy and efficiency of this algorithm depend heavily on the particular instance of GBS, with the numerical data presented here having large error bars. However, the method to estimate the TVD after sampling is a good way to test the accuracy of the sample. In this way, although the accuracy cannot be guaranteed beforehand, the correctness of the algorithm can be verified in hindsight.

Further testing could be done to find the scaling for GBS distributions with displacement or unequal squeezing. However, it is unfeasible to test every possible scenario and reasonable to apply the relationships found above in these cases with the security that the accuracy can be tested afterwards. It is also possible that a better proposal distribution can be found to increase the efficiency. It is not clear how the exact condition $f(x) \leq Hg(x)$ can be ensured for most outcomes for a general proposal distribution and that is the reason it is not addressed in this thesis.

Acknowledgements and contributions

The author's contributions are performing all numerical tests and analysis, formalising the post-sampling check for the TVD, the peaked proposal distribution, applying stochastic hill climbing in this context and the method for finding the total photon number probabilities. The author is grateful to Anthony Laing for supervision, to Alex Neville for the introduction to rejection sampling, many useful ideas and conversations including the suggestion to use the acceptance rate to estimate the TVD, and to Levon Chakhmakhchyan for extensive theoretical support regarding GBS. The author also acknowledges Alex Jones for useful conversations.

Chapter 4

Simulating Gaussian Boson Sampling with Metropolised Independence Sampling

"A Wrackspurt... They're invisible. They float in through your ears and make your brain go fuzzy. I thought I felt one zooming around in here." - Luna Lovegood

In the previous chapter we looked at rejection sampling as a method to sample from the Gaussian boson sampling (GBS) distribution without needing to calculate the full probability distribution. The efficiency of rejection sampling is very dependent on the proposal distribution used. The need to know how to scale the proposal distribution such that it is always higher than the target distribution is very restricting to the proposal distributions we can consider. In this chapter we analyse a Markov chain Monte Carlo method which does not have this requirement and so there is more flexibility to the choice of proposal distributions. We begin by introducing the algorithm in a general setting.

4.1 Markov chain Monte Carlo

4.1.1 Markov chains

A discrete-time process can be described by a sequence of random variables, $\{X_1, X_2, ..., X_t, ...\}$, where X_t gives the state of the process at time t and each time a stochastic process is started, a different chain emerges according to some probability distribution. The set of all values each state can take is called the **state space**, denoted Ω. A **discrete-time Markov chain** describes such a stochastic process with the defining property that the probability of the state at time t is directly dependent only on the state at time $t-1$ and independent of all other previous states [69, 70]:

$$\Pr(X_t = x_t | X_1 = x_1, \ldots, X_{t-1} = x_{t-1}) = \Pr(X_t = x_t | X_{t-1} = x_{t-1}). \tag{4.1}$$

The probability of transitioning from the state at time t to the state at time $t+1$ is given by the **Markov kernel**. In the following, we assume a finite discrete state space (as is the case for GBS), but the treatment can be generalised to a continuous state space by using integrals instead of summations. Of course, because $\Pr(x_{t-1})$ depends on x_{t-2}, there is implicit dependence on previous values, but that information is contained in the last variable and hence knowledge of the path to get to that value is unnecessary. However, in order to predict the state multiple time steps ahead we can use recursion by summing over all the possible paths. For example, for two time steps we sum over all the possible intermediate points:

$$\Pr(x_{t+2}|x_t) = \sum_{x_{t+1}} \Pr(x_{t+2}|x_{t+1})\Pr(x_{t+1}|x_t). \tag{4.2}$$

This means that, given a starting state, it is possible to find the probability of the state at any time by the Markov kernels at each point in the chain:

$$\Pr(x_t|x_0) = \sum_{x_{t-1},\ldots,x_1} \Pr(x_t|x_{t-1})\Pr(x_{t-1}|x_{t-2})\cdots\Pr(x_1|x_0). \tag{4.3}$$

For a discrete state space, Ω, all the transition probabilities at each time point can be given by an $|\Omega| \times |\Omega|$ matrix $\boldsymbol{P}$ where

$$\boldsymbol{P}_{i,j}^{(t)} = \Pr(X_t = x_i | X_{t-1} = x_j). \tag{4.4}$$

This matrix is called the **transition matrix**. A Markov chain is time-homogeneous if the transition probabilities do not change over time and therefore $\boldsymbol{P}^{(t)} = \boldsymbol{P}^{(t')}$. If a vector of length $|\Omega|$, $\boldsymbol{\mu}^{(t)}$, describes the probability of being in each state at time t, the probability at any time can be given by It is convenient to write the probabilities of being in each state at time t in a vector of length $|\Omega|$, $\boldsymbol{\mu}^{(t)}$. For time-homogeneous chains, the probabilities at time t can be determined from the initial probabilities as

$$\boldsymbol{\mu}^{(t)} = \boldsymbol{P}^t \boldsymbol{\mu}^{(0)}. \tag{4.5}$$

For an eigenvector of $\boldsymbol{P}$, $\boldsymbol{v}$, with eigenvalue 1, $\boldsymbol{P}\boldsymbol{v} = \boldsymbol{v}$, the eigenvector describes a probability distribution that is stationary as applying the transition matrix any number of times leaves the probability distribution unchanged and fixed as time increases. Because the transition matrix describes probabilities, all elements are non-negative and the sum of the rows and columns must be 1. By the Perron-Frobenius theorem, this matrix must have such an eigenvector and the probability distribution converges to it over time. In a sense, as the chain progresses it 'forgets' the starting point and $P(X_t|X_0) \to P(X_t)$ as $t \to \infty$. When the probabilities are independent of the starting position, they are now time independent. So for $t > t_{\min}$ the probability distribution is stationary.

To ensure that the distribution converges to a stationary distribution, the chain must satisfy three properties [71]. These are

1. *Irreducible*

   The chain must be able to reach all possible states from any starting point. This means the

transition kernel must satisfy

$$\Pr(X_t = x_i | X_0 = x_j) > 0 \tag{4.6}$$

for some t and all i, j. This is trivially true if we allow transitions between any two states in the state space.

2. *Aperiodic*
   The chain must have some probability of returning to a state at random points in the chain, not cycling back to states with some regular occurrence. To be precise a chain is aperiodic if

$$\text{greatest common divisor } \{t > 0 : \Pr(X_t = x_i | X_0 = x_i) > 0\} \qquad \forall i \tag{4.7}$$

   This is trivially true if $\Pr(X_t = x_i | X_{t-1} = x_i) > 0$ for all t and i, ie. it is always possible to add the same state to a chain as the previous one.

3. *Positive recurrent*
   The chain is expected to return to its initial state in a finite time. This is trivially true if we allow any transition to be reversed (ie. if a transition in one direction is possible, it is also possible in the other direction). An equivalent definition is that a stationary distribution, $\pi(x)$, exists such that if the initial state is sampled from the stationary distribution the chain will continue to sample from that distribution:

$$\pi(x_j) = \sum_i \pi(x_i) \Pr(X_t = x_j | X_{t-1} = x_i). \tag{4.8}$$

4.1.2 Monte Carlo

We draw on methods in statistical analysis used to estimate the expectation values associated with probability distributions such as moments, interquartiles etc. The foundation of this method is that

$$E[\phi(x)] = \frac{\sum_i \phi(x_i) f(x_i)}{\sum_i f(x_i)} \approx \frac{1}{n} \sum_{q=1}^{n} \phi(x_q) \tag{4.9}$$

where $f(x_i)$ is the probability associated with drawing value x_i, and $\phi(x)$ is any function of x. Thus drawing n samples from a distribution can be used to estimate the expected value of a function $\phi(x)$. For our purposes we are interested in drawing samples from a distribution without using them to estimate an expectation value. So we focus here on how the samples can be drawn using Markov chain Monte Carlo methods, rather than why they are useful in statistics.

4.1.3 Metropolis-Hastings algorithm

The main idea for drawing samples from a distribution using a Markov chain is that a chain is formed where the stationary distribution is the target distribution. Once the distribution has converged,

states will be added to the chain with probability according to the target distribution. To build a Markov chain, it is necessary to define the transition probability.

The Metropolis-Hastings algorithm introduces a process for a state to transition [72, 73]. As in rejection sampling, a state is suggested from a proposal distribution that is efficient to sample from and this state is accepted or rejected with some probability. Here, if the state is accepted it is added to the Markov chain, whereas if it is rejected the previous state must be added again. The probability of accepting a proposed state y given the current state is x is given by

$$\Pr(\text{accept } y|x) = \min\left(1, \frac{f(y)g(y,x)}{f(x)g(x,y)}\right), \tag{4.10}$$

where $f(x)$ is the target probability of x and $g(x,y)$ is the proposal probability of y if the current state is x. Note that the proposal probability may now have dependence on the current state. So the probability of transitioning from x to y, with $x \neq y$ is

$$\Pr(X_t = y|X_{t-1} = x) = g(x,y)\min\left(1, \frac{f(y)g(y,x)}{f(x)g(x,y)}\right) = \min\left(g(x,y), \frac{f(y)g(y,x)}{f(x)}\right). \tag{4.11}$$

The probability of adding the same state to the chain is simply one minus the probability of adding a different state. So the probability of transitioning from state x to state x is

$$\Pr(x|x) = 1 - \sum_{y\neq x} \min\left(g(x,y), \frac{f(y)g(y,x)}{f(x)}\right). \tag{4.12}$$

So the Metropolis-Hastings algorithm forms a chain with transition matrix defined by eqs. (4.11) and (4.12) with the introduction of a proposal distribution. Similarly to rejection sampling, this proposal distribution must be efficient to sample from, any probability must be able to be calculated for the accept rule and the closer the proposal to the target distribution, the more likely the chain updates the state at each step and the faster it converges to the target distribution.

4.1.4 Metropolised independence sampling

Metropolised independence sampling (MIS) is a particular case of the Metropolis-Hastings algorithm where it is assumed that the proposal distribution is independent of the current state in the chain: $g(x,y) = g(y)$. This is the version we choose to use for simulating GBS and so we check under what conditions on the proposal distribution the Markov chain satisfies the three requirements to converge to a stationary distribution with this algorithm. We also confirm that the target distribution is the stationary distribution.

The first condition is irreducibility. This is satisfied if we can always transition to any possible state. The probability $\Pr(X_t = y|X_{t-1} = x)$ is only zero if $g(x) = 0$, $g(y) = 0$ or $f(y) = 0$. In the case that $f(y) = 0$, this means it is not possible to sample that outcome from the target distribution and hence that state should not be included in the state space. The state x must have previously been sampled from the proposal distribution (and accepted) and therefore $g(x) \neq 0$. So as long as $g(y) \neq 0$, the chain is irreducible. A valid proposal distribution for this algorithm must have

a positive probability for all states that have positive target probabilities. This is also true for rejection sampling.

The second condition is aperiodicity. This is satisfied if it is possible to add the same state twice in a row, ie. $\Pr(x|x) \neq 0$. The probability of remaining in the same state is given by the probability of rejecting the proposed state or proposing the same state. There will always be at least one value of x (with the smallest $f(x)/g(x)$ value) for which any proposed state is guaranteed to be accepted. So we require there to be a non-zero probability of proposing the current state, at least for these values. Given the proposal probabilities are independent of the current state and it must have been proposed already to have been added to the chain, this must always be true. Hence the second condition is naturally satisfied in this algorithm for any proposal distribution.

The final condition is positive recurrence, ie. that if a state is sampled from the stationary distribution, the next state will be drawn from the same distribution. We show this is always true, if the stationary distribution is the target as desired. The probability of sampling x at some point in the chain, X_t, is the probability that x is proposed and accepted or that x was the previous state in the chain and the proposed state was rejected:

$$\begin{aligned}\Pr(X_t = x) &= \sum_y f(y)g(x)\min\left\{1, \frac{f(x)g(y)}{f(y)g(x)}\right\} + \sum_y f(x)g(y)\left(1 - \min\left\{1, \frac{f(y)g(x)}{f(x)g(y)}\right\}\right) \\ &= \sum_y \min\left\{f(y)g(x), f(x)g(y)\right\} + \sum_{y\in B} f(x)g(y)\left(1 - \frac{f(y)g(x)}{f(x)g(y)}\right),\end{aligned} \tag{4.13}$$

where $B = \{y : f(y)g(x) < f(x)g(y)\}$. The first sum includes all possibilities for proposing x and accepting it, and the second sum encompasses when the previous state in the chain was x and the proposed state was rejected. From the first to second line we used the fact that the terms in the second sum where $y \notin B$ contribute zero to the sum and so can be removed. By further splitting the sums over $y \in B$ and $y \notin B$, the above expression can be simplified as

$$\begin{aligned}\Pr(X_t = x) &= \sum_{y\in B} f(y)g(x) + \sum_{y\notin B} f(x)g(y) + \sum_{y\in B} f(x)g(y)\left(1 - \frac{f(y)g(x)}{f(x)g(y)}\right) \\ &= \sum_{y\in B} f(y)g(x) + \sum_{y\notin B} f(x)g(y) + \sum_{y\in B} f(x)g(y) - \sum_{y\in B} f(y)g(x) \\ &= \sum_{y\notin B} f(x)g(y) + \sum_{y\in B} f(x)g(y) \\ &= \sum_y f(x)g(y) \\ &= f(x).\end{aligned} \tag{4.14}$$

Hence, once a state has been drawn from the target distribution at any point, any state afterwards will also be sampled from the target distribution. Therefore a stationary distribution exists and it is the target distribution.

The steps for building the Markov chain in MIS are:

1. Take a sample point x_1 from the proposal distribution and take this as the start of the chain;
2. For $i = 2$ to N_l (the desired length of the chain):
   (a) Sample a state x_i from the proposal distribution;
   (b) Evaluate the update probability, $p = \frac{f(x_i)g(x_{i-1})}{g(x_i)f(x_{i-1})}$;
   (c) Add x_i to the chain with the probability p or add x_{i-1};

This algorithm builds a chain that converges to sampling from the target distribution. Our aim is to sample from the target distribution for any sample size. This can be done by taking states from the chain but there are two things to consider. The first is that the states at the beginning of the chain have not yet converged to the stationary distribution and so must be discarded. The number of outcomes we discard is called the **burn-in time** or the **mixing time** and there is no exact method to calculate this but tests can be done to find a reasonable estimate. The second issue with this method is that although any particular state in the chain is sampled from the target distribution once it has converged, the probabilities are not independent of the previous states. This is the Markov property that the transition probability depends on the previous state and due to the algorithm adding the same state if the proposed one is rejected, it is much more likely that a state will be repeated in a Markov chain than if two independent samples were drawn. In Monte Carlo techniques, it is not necessary to have independent samples to approximate the average values, and so this correlation between neighbouring states is often ignored. However, for our purpose we generally do want independent samples and so we need to consider this when we form our chain. To remove this correlation we include a thinning interval τ such that only every τth state is kept when we take our sample. Both the burn-in time and thinning interval add inefficiencies into the algorithm and can be reduced by having a better proposal distribution. There is a trade-off between having a good convergence and lack of correlation with the speed of the sampling algorithm, which can be chosen by the user. We note that the proposal distribution does not need to satisfy the condition $f(x) \leq Hg(x)$ as for rejection sampling which allows for more options when considering the proposal distribution.

4.2 Metropolised independence sampling applied to Gaussian boson sampling

When applying MIS to any sampling problem, the three problems to fix are finding a good proposal distribution, and determining the burn-in time and the thinning interval. In this section we discuss several possible proposal distributions for GBS and in the following section numerically test the burn-in times and thinning intervals for some of these proposal distributions.

4.2.1 Stepped uniform proposal

As with rejection sampling we can use the stepped distribution for MIS, where we still want to take into account the probability of each photon number, with outcomes within each N being uniformly

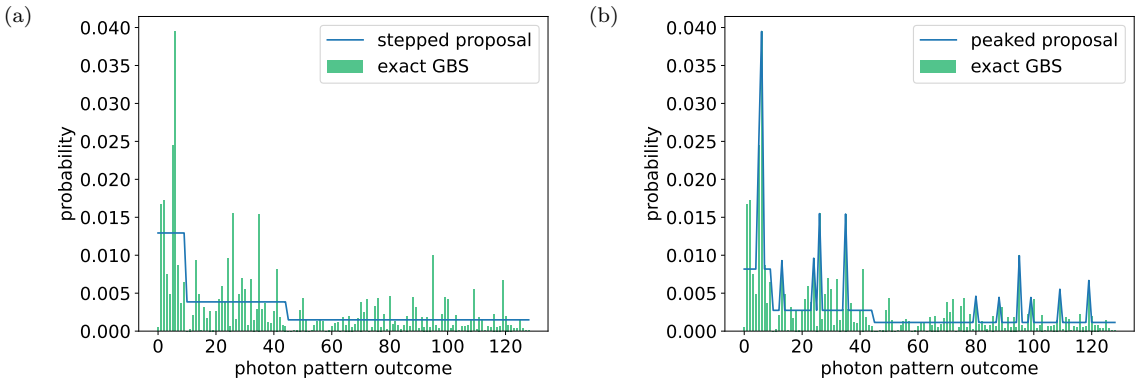

Figure 4.1: Proposal distributions for MIS. For an example Haar random unitary, the target distribution is shown by the green bars. Two proposal distributions are illustrated by the blue lines. The stepped proposal distribution is shown in (a). Here the height of each level is the average probability of patterns with that photon number. In (b), the peaked proposal is shown. For this, the n maxima patterns with n photons are set as the target probabilities and the remaining patterns take the average value of the other patterns for each photon number.

distributed, but the height different for each N. Although in rejection sampling this was to reduce the area underneath the proposal distribution as the maximum values in each photon number are potentially very different, in MIS we choose to do this because it is a closer distribution than the uniform to the target GBS distribution. It is also as easy to sample from the stepped distribution and to calculate its probabilities as long as the probability of each photon number is known. The height in each photon number is fixed simply by ensuring that the total probability of the outcomes with N photons is equal to the probability of sampling N photons from the target distribution. Therefore the height of the uniform distribution for N photons is given by the average probability:

$$g(x^{(N)}) = \frac{\Pr(N)}{|\Omega|}, \qquad \text{with } |\Omega| = \binom{N+M-1}{N}, \tag{4.15}$$

where the (N) superscript denotes an outcome with N photons and $\Pr(N)$ is the probability of getting N photons from the squeezed (and displaced) input states in the GBS device with M modes. This distribution is shown in fig. (4.1a). To sample from this probability distribution, first sample the photon number and then sample from the outcomes with N photons uniformly. The methods to sample the number of photons and photon patterns uniformly is given in sections (3.2.1) and (3.2.2.1) respectively.

4.2.2 Peaked proposal

By replacing some of the proposal values with the target values, the proposal distribution must always be closer to the target distribution. When one value is changed, all others must be rescaled so that it is still normalised. When the value is changed to the target value, that particular outcome is better approximated by the proposal and therefore reduces the TVD between the proposal and target distributions, but also the other outcomes must on average become better approximated

when they are renormalised further improving the TVD. As more values are replaced by the target probability the proposal distribution becomes closer and closer, where in the limit of all values being replaced the proposal distribution is exactly the target distribution. We can use this technique to suggest a proposal distribution where some of the outcomes are given by the target value. As this requires the calculation of the target probabilities which is time costly the distribution must only require a polynomial number of values to calculate. We note that the biggest reduction in TVD is found when there is the biggest difference between the proposal and target probabilities.

We choose to begin with the stepped uniform distribution as described above and therefore the outcomes with the biggest difference must be either the ones with the largest or smallest target probabilities. Because the probabilities are lower bounded by zero and upper bounded by one and the uniform probability will be increasingly close to zero as the sample space increases, it seems reasonable that the largest deviations from the uniform height will be for the outcomes with the greatest target probabilities. Therefore we choose to replace a polynomial number of the highest valued outcomes with their target probabilities. This is very similar to the peaked distribution in the previous chapter where we found the N highest probabilities. However, for rejection sampling the Nth maximum set the height of the uniform part whereas for MIS the height is set to ensure the proposal distribution is normalised. This distribution is shown in fig. (4.1b) So the proposal probability is given by

$$g(x^{(N)}) = \begin{cases} f(x), & \text{if } x \text{ is in } N \text{ maxima} \\ \frac{\Pr(N) - \sum f(x_{\text{peak}})}{\binom{N+M-1}{N} - N}, & \text{otherwise} \end{cases} \tag{4.16}$$

4.2.3 Coherent states

We can consider proposal distributions that are directly related to the GBS set-up in the hope that their distributions will be closer than the uniform distribution that is independent of the input states and interferometer. In particular we can consider the distribution from using states of light that are classically efficient to simulate as the input states instead. One example is the coherent state. So if we want to find a state that gives a similar probability distribution to our squeezed (and possibly displaced) states, we have a few options to try. The first is in the case that there is displacement in the input states. Here we can just ignore the squeezing and set the proposal distribution by using only the coherent component of the states. Intuitively this will give the biggest overlap with the target distribution when we have a lot of displacement compared to the squeezing. This distribution is shown in fig. (4.2c).

In the case where there is no displacement, this distribution cannot be used but instead a displacement parameter can be chosen such that this coherent state has the same mean number of photons in each mode as the input squeezed state. The mean number of photons in a squeezed state with squeezing parameter r is given by $\sinh^2 r$ and the mean number of photons in a coherent state with displacement parameter α is $|\alpha|^2$, and so by equating these values the correct displacement parameter can be found for the proposal distribution. Using this equivalent coherent state is applicable if there is displacement too. This distribution is shown in fig. (4.2d).

Recall from eq. (2.61) that a multimode coherent state passing through an interferometer is another multimode coherent state. Sampling a measurement of a coherent state in the Fock basis is efficient as it follows a Poisson distribution $N \sim \text{Pois}(|\alpha|^2)$ which can easily be sampled. Calculating the probability of a particular detection pattern is also efficient and given in eq. (2.54) for each mode.

We note that only the relative squeezing and displacement in different modes determine relative differences in the probability distribution of a particular photon number. That is, scaling the squeezing and displacement will only affect the probability distribution of the total number of photons and not the relative probabilities of each pattern conditional on the total photon number. Therefore, for equal squeezing and displacement, the values of the squeezing and displacement do not affect the target or proposal distributions and so the two methods of fixing the displacement parameter are completely equivalent, only when fixing the photon number.

In the event that the photon number is fixed, sampling from the Poisson distribution will not guarantee sampling a valid outcome with N photons. It is possible to keep sampling until a valid outcome is found, however it is possible to sample an outcome conditional on the total number of photons. If $n_i \sim \text{Pois}(|\beta_i|^2)$ for all modes i, we wish to find a sample $(n_1, \ldots, n_M)$ such that $\sum_i n_i = N$. From the relationship between Poisson distributions and multinomial distributions, $(n_1, \ldots, n_M) \sim \text{Mult}(N, \pi)$, where $\pi = (\pi_1, \ldots, \pi_M)$, and $\pi_i = |\beta_i|^2 / \sum_j |\beta_j|^2$.

4.2.4 Distinguishable squeezed states

We next take inspiration from the proposal distribution used in applying MIS to standard boson sampling, which was distinguishable photons. Many of the derivations in this section have been explored in a manuscript regarding partial indistinguishability [16]. First, we review indistinguishable and distinguishable photon probabilities in standard boson sampling. For boson sampling with an input state $\boldsymbol{s} \in \mathcal{Z}^M$, where we allow any integer number of indistinguishable photons in each input mode, the probability of an output pattern $\boldsymbol{t} \in \mathcal{Z}^M$, with $\sum \boldsymbol{s} = \sum \boldsymbol{t} = N$, is given by

$$\text{Prob}_{\text{indist}}(\boldsymbol{t}|\boldsymbol{s}) = \frac{1}{\boldsymbol{s}!\boldsymbol{t}!}|\text{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}})|^2, \tag{4.17}$$

where $\boldsymbol{s}! = \prod_i s_i!$, $\boldsymbol{U}^{\{r,c\}}$ is the submatrix of $\boldsymbol{U}$ made by keeping rows r and columns c, and $\tilde{\boldsymbol{s}} \in \mathcal{Z}^N$ is the pattern $\boldsymbol{s}$ in first quantisation. Now we consider the scenario where instead the photons are distinguishable to all photons in the other modes. We note that photons in the same mode behave the same regardless of whether they are distinguishable or indistinguishable and hence we can equivalently consider all photons to be distinguishable. The probability of an outcome $\boldsymbol{t}$ with input distinguishable photons is

$$\text{Pr}_{\text{dist}}(\boldsymbol{t}|\boldsymbol{s}) = \frac{1}{\boldsymbol{t}!}|\text{Perm}(|\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}}|^2), \tag{4.18}$$

where $|\boldsymbol{U}|^2$ is the elementwise modulus square of the matrix.

For a proposal distribution for GBS we instead suppose distinguishable squeezed states at the input. An example of this distribution is shown in fig. (4.2a). We consider distinguishable squeezed

states such that photons that come from different input modes are distinguishable. First, we look at how to calculate the probabilities of this distribution. Because distinguishable photons do not interfere, we can find the probability of an outcome by summing the probabilities (rather than probability amplitudes) of all the combinations of which mode each photon came from:

$$\Pr(\boldsymbol{t}) = \sum_{\substack{\boldsymbol{t}^{\{1\}},\dots\boldsymbol{t}^{\{M\}} \\ \sum \boldsymbol{t}^{\{i\}} = \boldsymbol{t}}} \prod_{i=1}^{M} \Pr(\boldsymbol{t}^{\{i\}}) \tag{4.19}$$

where $\Pr(\boldsymbol{t}^{\{i\}})$ is the probability of getting pattern $\boldsymbol{t}^{\{i\}}$ from the squeezer in input mode i. We can find the probability $\Pr(\boldsymbol{t}^{\{i\}})$ by assuming there is no squeezing in all modes apart from mode i. We can substitute this into eq. (2.122) to get

$$\Pr(\boldsymbol{t}^{\{i\}}) = \frac{1}{\boldsymbol{t}^{\{i\}}! \cosh r_i} \left| \mathrm{Haf}(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^{\top})^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}} \right|^2, \tag{4.20}$$

where $\boldsymbol{R}^{\{i\}}$ is the matrix with $R^{\{i\}}_{i,i} = \tanh r_i$ and 0 elsewhere. Due to the sparsity of the matrix $\boldsymbol{R}^{\{i\}}$, the expression for the product of matrices above can be simplified to

$$(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^{\top})_{j,k} = \sum_{v,w} U_{j,v} R^{\{i\}}_{v,w} U^{\top}_{w,k} = U_{j,i} R_{i,i} U^{\top}_{i,k} = \tanh r_i U_{j,i} U_{k,i}. \tag{4.21}$$

The hafnian is actually calculated for a submatrix of this and so when the corresponding rows and columns for the detected output pattern $\boldsymbol{t}^{\{i\}}$ are included, the row j in the submatrix corresponds to row $\tilde{\boldsymbol{t}}^{\{i\}}_j$ in the original matrix and column k to column $\tilde{\boldsymbol{t}}^{\{i\}}_k$:

$$[(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^{\top})^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}}]_{j,k} = (\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^{\top})_{\tilde{\boldsymbol{t}}^{\{i\}}_j,\tilde{\boldsymbol{t}}^{\{i\}}_k} = \tanh r_i U_{\tilde{t}^{\{i\}}_j,i} U_{\tilde{t}^{\{i\}}_k,i} \tag{4.22}$$

This can instead be written in terms of a submatrix of the unitary with rows $\tilde{\boldsymbol{t}}^{\{i\}}$ and all columns, $\boldsymbol{U}^{\{\boldsymbol{t}^{\{i\}}\}}$:

$$[(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^{\top})^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}}]_{j,k} = \tanh r_i U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{j,i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{k,i}. \tag{4.23}$$

We use the notation $\boldsymbol{U}^{\{r\}}$ to mean a submatrix of $\boldsymbol{U}$ formed by keeping rows in r and *all* columns, but the notation $\boldsymbol{U}^{\{r,c\}}$ to keep rows in r and only the columns in c. From the definition of the

hafnian, and denoting the number of photons detected from mode i as $n_i = \sum \boldsymbol{t}^{\{i\}}$, we find

$$\begin{aligned}
\mathrm{Haf}[(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^\top)^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}}] &= \frac{1}{(n_i/2)!2^{(n_i/2)}} \sum_{\sigma\in S_{n_i}} \prod_{j=1}^{n_i/2} \tanh r_i U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{\sigma(2j-1),i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{\sigma(2j),i} \\
&= \frac{\tanh^{n_i/2} r_i}{(n_i/2)!2^{(n_i/2)}} \sum_{\sigma\in S_{n_i}} \prod_{j=1}^{n_i/2} U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{\sigma(2j-1),i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{\sigma(2j),i} \\
&= \frac{\tanh^{n_i/2} r_i}{(n_i/2)!2^{(n_i/2)}} \sum_{\sigma\in S_{n_i}} \prod_{k=1}^{n_i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}_{\sigma(k),i} \\
&= \frac{\tanh^{n_i/2} r_i}{(n_i/2)!2^{(n_i/2)}} \sum_{\sigma\in S_{n_i}} \prod_{k=1}^{n_i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}_{\sigma(k),k} \\
&= \frac{\tanh^{n_i/2} r_i}{(n_i/2)!2^{(n_i/2)}} \mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}),
\end{aligned} \tag{4.24}$$

where $\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}$ is the matrix formed by taking the rows given by $\tilde{\boldsymbol{t}}^{\{i\}}$ and column i of $\boldsymbol{U}$ repeated n_i times. From line 2 to 3 the odd values of k correspond to the $\sigma(2j-1)$ terms and the even values of k to the $\sigma(2j)$ terms. From the third to fourth line, the column index does not change and so by replacing all columns in the unitary matrix by that column, all indices refer to the same column. In the final line, the definition of the permanent was used.

So the probability of outputting the pattern $\boldsymbol{t}^{\{i\}}$ with non-zero squeezing in only mode i can be found by substituting eq (4.24) into eq (4.20):

$$\begin{aligned}
\Pr(\boldsymbol{t}^{\{i\}}) &= \frac{1}{\boldsymbol{t}^{\{i\}}!\cosh r_i} \left|\mathrm{Haf}(\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^\top)^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}}\right|^2 \\
&= \left(\frac{n_i!}{(n_i/2)!^2 2^{n_i}} \frac{\tanh^{n_i} r_i}{\cosh r_i}\right) \left(\frac{1}{n_i!\boldsymbol{t}^{\{i\}}!} |\mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}})|^2\right) \\
&= \Pr(n_i)\Pr_{\mathrm{indist}}(\boldsymbol{t}^{\{i\}}|n_i).
\end{aligned} \tag{4.25}$$

Hence, the probability of this output is the same as the probability of getting n_i photons from the squeezed state multiplied by the probability of outputting this pattern from standard boson sampling with n_i photons input in mode i. We now show that the output pattern probability for indistinguishable photons input in one mode is the same as if the photons were distinguishable. The permanent of a matrix with all columns identical is a sum over product terms where each term in the sum is the same. Hence the permanent in this case can simply be written as one product term:

$$\mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}) = \sum_{\sigma\in S_{n_i}} \prod_{k=1}^{n_i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}_{\sigma(k),k} = n_i! \prod_{k=1}^{n_i} U^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}_{k,k}, \tag{4.26}$$

where $n_i!$ comes from the number of terms in the sum. When taking the modulus square of this

permanent, it can be moved to elementwise modulus square of the unitary:

$$|\mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}})|^2 = \left| n_i! \prod_{k=1}^{n_i} U_{k,k}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}} \right|^2 = n_i!^2 \prod_{k=1}^{n_i} \left| U_{k,k}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}} \right|^2 = n_i! \mathrm{Perm}\left(|\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{i\}},\boldsymbol{i}\}}|^2 \right). \quad (4.27)$$

Therefore the probability of the output pattern is the probability of producing n_i distinguishable photons and evolving them through the unitary:

$$\Pr(\boldsymbol{t}^{\{i\}}) = \left(\frac{n_i!}{(n_i/2)!^2 2^{n_i}} \frac{\tanh^{n_i} r_i}{\cosh r_i} \right) \left(\frac{1}{\boldsymbol{t}^{\{i\}}!} \mathrm{Perm}\left(|\boldsymbol{U}^{\{\tilde{\boldsymbol{t}}^{\{\boldsymbol{i}\}},\boldsymbol{i}\}}|^2 \right) \right) = \Pr(n_i) \Pr_{\mathrm{dist}}(\boldsymbol{t}^{\{i\}} | n_i). \quad (4.28)$$

This suggests we can sample from this distribution by first sampling how many photons the squeezed state produces and then sampling from standard boson sampling with distinguishable photons. Although this probability is written as a permanent, from the single product equivalence, it is possible to calculate in $O(1)$. This expression can be substituted into eq. (4.19) to find the probability of output pattern $\boldsymbol{t}$ from distinguishable squeezed states in M modes. The sum in eq. (4.19) comprises of all the ways of matching the photons in the output pattern to the input modes (ie. splitting the output pattern into groups depending on which squeezed state they originated). To examine how many ways there are of doing this, we first consider all the combinations of how t_i photons in output mode i were distributed among the input modes. This is simply all the ways of selecting t_i modes from M values with replacement: $\binom{t_i+M-1}{t_i}$. So across all output modes the number of ways of distributing the photons in the input modes is

$$\prod_{i=1}^{M} \binom{t_i + M - 1}{t_i}. \quad (4.29)$$

In the worst-case scenario every photon is detected in different modes and so $t_i = 1$ for N modes and 0 otherwise. In this case the binomial coefficient reduces to either M when $t_i = 1$ or 1 otherwise. Therefore the complexity is upper bounded by M^N. In the best-case scenario, all photons appear in one output mode and the complexity is simply $\binom{N+M-1}{N}$, which is the lower bound. However, because squeezing only produces photons in pairs in the ideal case, the number of photons from each input mode must be even and so this reduces the total number of ways of matching the output and input photons.

Rather than summing over all the ways of partitioning the photons according to the originating mode, it is possible to sum over all the possible input patterns. So we consider only *how many* photons came from each mode, not *which* photons originated in each mode. Clearly this will have a lot fewer terms to sum over if we do not assign all the photons to an input mode. We now show mathematically that this is a valid method.

We start with combining eqs. (4.28) and (4.19) and splitting the sum by first summing over all the ways of dividing the photons between the input modes and then over all possible patterns given

the division:

$$\Pr(\boldsymbol{t}) = \sum_{\substack{\boldsymbol{t}^{\{1\}},\ldots,\boldsymbol{t}^{\{M\}} \\ \sum \boldsymbol{t}^{\{i\}}=\boldsymbol{t}}} \prod_{i=1}^{M} \Pr(n_i)\Pr_{\text{dist}}(\boldsymbol{t}^{\{i\}}|n_i) = \sum_{\substack{n_1,\ldots,n_M \\ \sum n_i=N}} \sum_{\substack{\boldsymbol{t}^{\{1\}},\ldots,\boldsymbol{t}^{\{M\}} \\ \sum \boldsymbol{t}^{\{i\}}=\boldsymbol{t} \\ |\boldsymbol{t}^{\{i\}}|=n_i}} \prod_{i=1}^{M} \Pr(n_i)\Pr_{\text{dist}}(\boldsymbol{t}^{\{i\}}|n_i). \quad (4.30)$$

Here n_i is the number of photons detected from input mode i and $|\boldsymbol{t}^{\{i\}}|$ is the number of photons in the pattern $\boldsymbol{t}^{\{i\}}$ from input mode i, which must be equal. This expression can be rearranged to extract the probability of an output pattern from distinguishable input photons:

$$\Pr(\boldsymbol{t}) = \sum_{\substack{n_1,\ldots,n_M \\ \sum n_i=N}} \left(\prod_{i=1}^{M} \Pr(n_i)\right) \left(\sum_{\substack{\boldsymbol{t}^{\{1\}},\ldots,\boldsymbol{t}^{\{M\}} \\ \sum \boldsymbol{t}^{\{i\}}=\boldsymbol{t} \\ |\boldsymbol{t}^{\{i\}}|=n_i}} \prod_{i=1}^{M} \Pr_{\text{dist}}(\boldsymbol{t}^{\{i\}}|n_i) \right) = \sum_{\boldsymbol{s}} \Pr(\boldsymbol{s})\Pr_{\text{dist}}(\boldsymbol{t}|\boldsymbol{s}), \quad (4.31)$$

where $\boldsymbol{s}$ corresponds to the input pattern and gives the number of photons that originated from each mode. The sum is over all possible input patterns. This proves that the probability from distinguishable squeezed states is the probability of outputting that pattern from distinguishable photons weighted across all possible input patterns. The complexity of calculating the probability this way is given by the number of input patterns to sum over multiplied by the complexity of calculating a single probability. The calculation of the probability of the input state is efficient $O(1)$ and the probability for distinguishable single photons is given by the permanent of a positive $N \times N$ matrix which has complexity $O(N2^N)$ when restricting to the exact probability. In the case where both even and odd numbers of photons can be produced by each source, the number of potential input patterns is given by $\binom{N+M-1}{N}$ because we must choose N modes with possible repeated modes. However, when restricted to the case where photons come in pairs as they do with perfect squeezed states, we can now pair the photons together and consider where each *pair* is produced and so there are $\binom{N/2+M-1}{N/2}$ possible input states. So the overall complexity of calculating a probability from distinguishable squeezed states is $O(\binom{N+M-1}{N}N2^N)$ for general states or $O(\binom{N/2+M-1}{N/2}N2^N)$ when photons are produced in pairs. In reality this is a simplification of the complexity as photons bunched in the same input or output modes reduce the complexity, but we keep it general for the worst-case scenario. Also the matrix that the permanent is taken for is real and positive and there exists a polynomial algorithm to approximate this. However, the effect on MIS of the error due to this approximate algorithm is unknown and as such we choose to avoid this and find the exact probability.

In the above derivation, a zero displacement was assumed when the hafnian was reduced to the permanent. It can be extended to non-zero displacement input, where now the probability of a pattern from each squeezed state is given by

$$\Pr(\boldsymbol{t}^{\{i\}}) = \frac{\exp\left(\frac{1}{2}\text{Re}(\boldsymbol{\alpha}^\dagger \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top \boldsymbol{\alpha}^*) - |\boldsymbol{\alpha}|^2\right)}{\boldsymbol{t}^{\{i\}}! \cosh r_i} \left|\text{Lhaf}\left(\boldsymbol{\gamma}^{\{\tilde{\boldsymbol{t}}^{\{i\}}\}}, (\boldsymbol{U}\boldsymbol{R}^{\{i\}}\boldsymbol{U}^\top)^{\{\tilde{\boldsymbol{t}}^{\{i\}},\tilde{\boldsymbol{t}}^{\{i\}}\}}\right)\right|^2. \quad (4.32)$$

We do not show the derivation but the same treatment will find that the probability is still given as the sum over the possible input patterns with the probability of the input state now given by the probability for squeezed displaced states and the distinguishable photon probabilities are unchanged.

So we have found two ways to calculate the probabilities for this proposal distribution, but we also need to be able to sample from it efficiently. From the chain rule presented in eq. (4.31) the distribution from distinguishable input squeezed states can be sampled by first sampling the number of photons from each input mode and then sampling the output pattern from the input distinguishable single photons. As both of these are efficient to sample, the overall algorithm is also efficient. The input patterns can be sampled by calculating the probabilities of all number of photons up to some chosen cut-off photon number. The probabilities of n photons from squeezed vacuum and squeezed displaced states are given by eq. (2.66) and eq. (2.72) respectively. To sample from these input distinguishable single photons, the output mode of each photon can be sampled separately as they do not interfere. The probability of a photon input in mode s being detected in output mode t is given by the modulus square of a permanent of a 1×1 matrix, which is simply a number, and so sampling the output mode for each input photon involves sampling from a M-length known distribution.

When sampling the number of photons in each input mode, there are two complications to consider. The first is that the necessity to introduce a cut-off number of photons in each mode adds some error. This can be removed by allowing the cut-off in each mode to be that of the overall cut-off photon number and then repeatedly sampling until a pattern with less than the overall cut-off number of photons is produced. As long as the cut-off is sufficiently high that the distribution above this number of photons is negligible, the probability of sampling within the allowed number of photons is high and as such most samples will not need to be repeated and hence the efficiency of the sampling algorithm is not too adversely affected. The second complication is that the number of photons is not fixed when sampling the input pattern. The probability of N photons is the same for the proposal as target distribution and therefore sampling the input pattern naturally samples the photon number rather than needing to explicitly sample the photon number and then sampling the proposal pattern conditional on the number of photons as we have done with other proposal distributions. So in general this is not a problem. However, if it is desired to fix the photon number, for example for testing purposes, repeating until the correct number of photons are detected can be very inefficient if the photon number is not very probable. As the number of photons is conserved between the input and output modes, we can post-select the input patterns without needing to sample the output pattern until a valid input is found. Sampling the input patterns are very efficient and so repeating until success is not too problematic unless the probability of success is low. However, the probability of N can be extremely low and hence it is important to find a method to sample the input pattern conditional on the total number of photons.

Here we introduce such an algorithm. First by the definition of conditional probability

$$\Pr(s_i = n_i | S = N) = \frac{\Pr(s_i = n_i, S = N)}{\Pr(S = N)}, \tag{4.33}$$

where s_i is the number of photons in mode i, and S is the total photon number across all modes. The quantity $\Pr(S = N)$ can be efficiently found according to eq. (3.13), but it can also be ignored as it is constant for all values of n_i. So the quantity of interest is $\Pr(s_i = n_i, S = N)$. If the total number of photons across all modes is N and the number of photons in mode i is n_i, this condition is completely equivalent to n_i photons in mode i and $N - n_i$ photons in all modes except mode i. Because the modes are independent, the number of photons in mode i does not affect the number of photons in all other modes. Therefore the probability can be written as

$$\Pr(s_i = n_i, S = N) = \Pr(s_i = n_i, S_{\setminus i} = N - n_i) = \Pr(s_i = n_i)\Pr(S_{\setminus i} = N - n_i), \tag{4.34}$$

where $S_{\setminus i}$ is the total number of photons in all modes except i. We note that it is efficient to find the number of photons in any subset of the modes by using eq. (3.13) and also to find the probability of a particular pattern in a subset of the modes because the modes are independent at the input. We can use the chain rule of probability to iteratively sample each mode to get a sample with N photons in total. For each mode, the distribution to sample from is the probability conditional on the overall total number of photons and the sampled pattern in the previous modes:

$$\begin{aligned}
\Pr(s_i = n_i | S = N \cap \boldsymbol{s}_{<i} = \boldsymbol{n}_{<i}) &= \frac{\Pr(s_i = n_i \cap S = N \cap \boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})}{\Pr(S = N \cap \boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})} \\
&= \frac{\Pr(s_i = n_i \cap S_{>i} = N - n_i - N_{<i} \cap \boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})}{\Pr(S_{>i-1} = N - N_{<i} \cap \boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})} \\
&= \frac{\Pr(s_i = n_i)\Pr(S_{>i} = N - n_i - N_{<i})\Pr(\boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})}{\Pr(S_{>i-1} = N - N_{<i})\Pr(\boldsymbol{s}_{<i} = \boldsymbol{n}_{<i})} \\
&= \frac{\Pr(s_i = n_i)\Pr(S_{>i} = N - n_i - N_{<i})}{\Pr(S_{>i-1} = N - N_{<i})},
\end{aligned} \tag{4.35}$$

where the notation $\boldsymbol{s}_{<i}$ and $\boldsymbol{s}_{>i}$ is used to succinctly write the state of the modes before/after the ith mode respectively: $\boldsymbol{s}_{<i} = (s_1, \ldots, s_{i-1})$ and $\boldsymbol{s}_{>i} = (s_{i+1}, \ldots, s_M)$, and similarly $S_{<i}$ and $S_{>i}$ are the total number of photons in modes before/after the ith mode. Hence to sample the input pattern, start with the first mode and sample the number of photons in each mode in turn by evaluating the above probability for all n_i up to N. The probability for $n_i > N - N_{<i}$ will be zero because the second term in the numerator will be the probability of a negative number of photons which is impossible. Therefore, the probability distribution is finite and the sampling will be exact. The denominator is independent of the number of photons in mode i and therefore is a normalisation constant which doesn't need to be directly calculated but the distribution could instead be normalised once all the probabilities have been found. It is slightly more efficient to simply sum the probabilities to find this constant rather than calculate it directly. The full algorithm is outlined in algorithm (3).

We have shown an efficient method for sampling from this proposal distribution, however the two ways shown of calculating the probability are both much more inefficient than the target probability. Finding a method to calculate this proposal probability with a complexity no worse than the (loop) hafnian remains an open problem. Whilst it seems likely that a solution would exist, until it is

Algorithm 3 Sample input photon pattern

Input: N - desired number of photons
Output: $\boldsymbol{s} = (s_1, \ldots, s_M)$
  Begin with state $\boldsymbol{s} = (0, \ldots, 0)$, and $n_{\text{prev}} = 0$
  for $m = 1$ to M **do**
    for $n = 0$ to N **do**
      Find $\Pr(n) = \Pr(s_m = n)\Pr(\sum_{j=m+1}^{M} s_j = N - n - n_{\text{prev}})$
    Sample n_m from the normalised distribution $\Pr(n)$ and update $s_m = n_m$
    $n_{\text{prev}} \to n_{\text{prev}} + n_m$
  return $\boldsymbol{n}$

found, this proposal distribution is not useful for MIS and is only included and analysed in this work with the prospect that a solution may be found.

4.2.5 Independent pairs and singles

The problems with the previous proposal distribution were caused by the generation of photons in each state not being independent even though the photons did not interfere in the interferometer. Specifically the probability of measuring two photons in two input modes with squeezed states is not the same as four photons in one mode and none in the other. Here we introduce another proposal distribution to overcome this, motivated by the mapping of GBS to standard boson sampling with the interferometer given by $\boldsymbol{URU}^\top$. For standard boson sampling with distinguishable photons, the modulus square of the permanent is moved inside to act on the matrix elementwise. We can do the same thing with the probability for GBS where we relate the matrix inside the hafnian to the equivalent interferometer in standard boson sampling. Because each pair in GBS becomes an input and output photon in the mapping, each pair of photons behaves independently including in the generation.

The probability of generating an output pattern $\boldsymbol{t}$ for independent pairs of photons is

$$\Pr(\boldsymbol{t}) = \mathcal{N}_1 \frac{\text{Haf}\left((|\boldsymbol{URU}^\top|^2)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}}\right)}{\boldsymbol{t}!}, \tag{4.36}$$

for input squeezed vacuum states. If displacement is also included, individual photons are also generated which we also treat as independent. So the probability of a pattern is generalised for input squeezed coherent states to

$$\Pr(\boldsymbol{t}) = \mathcal{N}_2 \frac{\text{Lhaf}\left((|\boldsymbol{\gamma}|^2)^{\{\tilde{\boldsymbol{t}}\}}], (|\boldsymbol{URU}^\top|^2)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}}\right)}{\boldsymbol{t}!}. \tag{4.37}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are normalisation constants to be found. Because each pair or single is produced independently, the probability distribution for the total number of photons must be different to the target distribution. We wish to determine the above normalisation constants and also to find the probability of the total number of photons.

We begin with just displaced states which naturally act independently and so the probability distribution is already known. A photon is produced from a coherent state with probability proportional to $|\alpha|^2$. This implies that the probability of producing n photons is

$$\Pr(n) = \mathcal{N}_D \frac{|\alpha|^{2n}}{n!}, \tag{4.38}$$

where we multiplied the probabilities for each photon being produced and divided by $n!$ due to the number of ways of ordering the photons. This normalisation constant can be found by equating the sum of the probabilities of all photon numbers from zero to infinity with one, and employing the power series expansion of the exponential:

$$\sum_{n=0}^{\infty} \mathcal{N}_D \frac{|\alpha|^{2n}}{n!} = \mathcal{N}_D \exp\big(|\alpha|^2\big) = 1 \qquad \implies \Pr(n) = \exp\big(-|\alpha|^2\big) \frac{|\alpha|^{2n}}{n!}, \tag{4.39}$$

which agrees with the known result and the number of photons from a coherent state follows a Poisson distribution.

Now we apply the same treatment to squeezed states where each pair is independent. We assume that the probability of producing a pair from a squeezed state is proportional to $\tanh^2 r$. This implies that the probability of producing n pairs of photons is

$$\Pr(2n) = \mathcal{N}_S \frac{\tanh^{2n} r}{2^n n!}, \tag{4.40}$$

where we multiplied the probabilities for each photon pair being produced and divided by $n!$ to account for the number of ways to order each pair of photons and 2^n for the ordering of the two photons in each pair. This normalisation constant can be found using the same methods as above:

$$\begin{aligned} &\sum_{n=0}^{\infty} \mathcal{N}_S \frac{\tanh^{2n} r}{2^n n!} = \sum_{n=0}^{\infty} \mathcal{N}_S \frac{(\tanh^2 r/2)^n}{n!} = \mathcal{N}_S \exp\big(\tfrac{1}{2}\tanh^2 r\big) = 1 \\ &\implies \Pr(n) = \exp\big(-\tfrac{1}{2}\tanh^2 r\big) \frac{(\tanh^2 r/2)^n}{n!}, \end{aligned} \tag{4.41}$$

so that the number of photon pairs from a squeezed state also follow a Poisson distribution, $2n \sim \text{Pois}(\frac{1}{2}\tanh^2 r)$.

Finally, we look at squeezed coherent states by considering all combinations of photons from the squeezing part and the displacement part. For example, if four photons are detected, all four could be 'squeezing photons', or they could all be 'displacement photons' or they could be two of each. Note that the restriction such that the squeezing photons must come in pairs puts a restriction on whether an odd or even number of photons came from the displacement. So we find the probability of n independent photons from squeezed displaced states to be

$$\Pr(n) = \sum_{q=0}^{n/2} \mathcal{N}_{DS} \frac{\tanh^{2q} r}{2^q q!} \frac{|\alpha|^{2(n-2q)}}{(n-2q)!}, \tag{4.42}$$

which sums over getting $2q$ squeezing photons and $n-2q$ displacement photons. To find this normalisation constant, we again sum over all possible total photon numbers:

$$\begin{aligned}
\sum_{n=0}^{\infty}\sum_{q=0}^{n/2}\mathcal{N}_{DS}\frac{\tanh^{2q}r}{2^q q!}\frac{|\alpha|^{2(n-2q)}}{(n-2q)!} &= \sum_{q=0}^{\infty}\sum_{n=2q}^{\infty}\mathcal{N}_{DS}\frac{\tanh^{2q}r}{2^q q!}\frac{|\alpha|^{2(n-2q)}}{(n-2q)!} \\
&= \mathcal{N}_{DS}\sum_{q=0}^{\infty}\frac{\tanh^{2q}r}{2^q q!}\sum_{n=2q}^{\infty}\frac{|\alpha|^{2(n-2q)}}{(n-2q)!} \\
&= \mathcal{N}_{DS}\sum_{q=0}^{\infty}\frac{\tanh^{2q}r}{2^q q!}\sum_{i=0}^{\infty}\frac{|\alpha|^{2i}}{i!} \\
&= \mathcal{N}_{DS}\sum_{q=0}^{\infty}\frac{\tanh^{2q}r}{2^q q!}\exp\big(|\alpha|^2\big) \\
&= \mathcal{N}_{DS}\exp\big(\tfrac{1}{2}\tanh^2 r\big)\exp\big(|\alpha|^2\big),
\end{aligned} \tag{4.43}$$

where in line three we introduced the variable $i = n-2q$. So the probability of n photons from a squeezed coherent state is

$$\Pr(n) = \exp\big(-\tfrac{1}{2}\tanh^2 r - |\alpha|^2\big)\sum_{q=0}^{n/2}\frac{\tanh^{2q}r}{2^q q!}\frac{|\alpha|^{2(n-2q)}}{(n-2q)!}, \tag{4.44}$$

which is not Poissonian. To sample the number of photons in this case it is easy to sample the number of photons from the squeezing and separately the number of photons from the displacement and we do not need to worry about the probability above.

This analysis is for the case of the input state, but by conservation of photons through the interferometer these normalisation constants are the same at the output state. We can also generalise the Poisson description to the state at the output. To sample the photon pattern at the output, the squeezing and displacement photons can be sampled separately. The displacement can be placed after the interferometer by applying the unitary $\alpha \to \boldsymbol{U}\alpha$ and so sampling from this is just sampling from a Poisson distribution on each mode with a new mean value. The squeezing photons can be sampled at the input and then sampled as distinguishable single photons through the interferometer.

Sampling as described above does not fix the total number of photons. Because the probability distribution of the total number of photons is different to the target probability there would be a poor overlap between the target and proposal distributions. So we scale the probabilities such that the probability of n photons is the same for the target and proposal distributions. We have found the photon number probabilities above for a single mode. The sum of variables distributed according to a Poisson distribution is itself distributed by a Poisson distribution where the mean parameter is given by the sum of the individual parameters: $X_1 \sim \text{Pois}(\lambda_1), \ldots, X_m \sim \text{Pois}(\lambda_m) \implies \sum_i X_i \sim \text{Pois}(\sum_i \lambda_i)$. Therefore the photon number for all M modes for squeezed vacuum states is distributed by $2n \sim \text{Pois}(\frac{1}{2}\sum_i \tanh^2 r_i)$. Because the probability for squeezed displaced states is not Poissonian, we cannot use the same method. The reason it is not Poissonian is because

although the sum of Poissonian variables is also Poissonian, here it is the number of photon *pairs* from squeezing that is Poissonian and not the individual photons. However, we can use the polynomial method introduced in section (3.2.1) to find the probability of the number of photons from squeezed displaced states.

In order to sample from this scaled distribution, we need to first sample the number of photons from the same distribution as for the target and then sample the pattern conditional on this outcome. One possibility is to simply sample a pattern until one with the correct number of photons is output, but this could be slow as the photon number is unlikely in the proposal distribution. As both the distributions from squeezed states and coherent states are Poissonian, it is easy to sample from these distributions separately conditional on the outcome using the multinomial method explored previously. So the only piece of the puzzle left is how to sample the proportion of photons from the squeezing and displacement conditional on the total number of photons. This can be done by building a probability distribution for the probability of $2i$ photons from squeezing *and* $N-2i$ photons from displacement for all $i = 0, \ldots, \frac{N}{2}$. This is simply the probability of measuring $2i$ squeezing photons conditional on measuring a total of N photons

$$\begin{aligned}\Pr(N_{sq} = 2i | N_{total} = N) &= \frac{\Pr(N_{sq} = 2i, N_{total} = N)}{\Pr(N_{total} = N)} = \frac{\Pr(N_{sq} = 2i, N_{dis} = N - 2i)}{\Pr(N_{total} = N)} \\ &= \frac{\Pr(N_{sq} = 2i)\Pr(N_{dis} = N - 2i)}{\Pr(N_{total} = N)}.\end{aligned} \tag{4.45}$$

Sampling the number of squeezing photons from this distribution naturally fixes the number of displacement photons too.

Hence we have shown that this distribution can be sampled efficiently either with fixing the photon number (and with a different probability distribution for the photon number) or not. The probabilities are given by the hafnian of an $N \times N$ positive matrix which to calculate exactly has time complexity $O(N^3 2^{N/2})$. An example is shown in fig. (4.2b). For the case of no squeezing (as depicted in the figure), in the event that there is only two photons, there are no other pairs from which to be independent or not and therefore the proposal probabilities are exactly the same as the target distribution. If there is displacement, this is no longer true as squeezed coherent states produce the displacement and squeezing photons not independently and so the proposal distribution is not true to the input states.

4.3 Numerical analysis of the efficiency and accuracy of Metropolised independence sampling

When using MIS it is important to be able to fix the burn-in time and thinning interval to ensure the sampling approximation is good without knowledge of the distribution itself. Here we perform a few numerical tests to find where these parameters need to be set for problem sizes that are small enough to be directly tested. The patterns at small problem sizes can then be extrapolated beyond the region that is testable. We also want to compare the proposal distributions to determine which

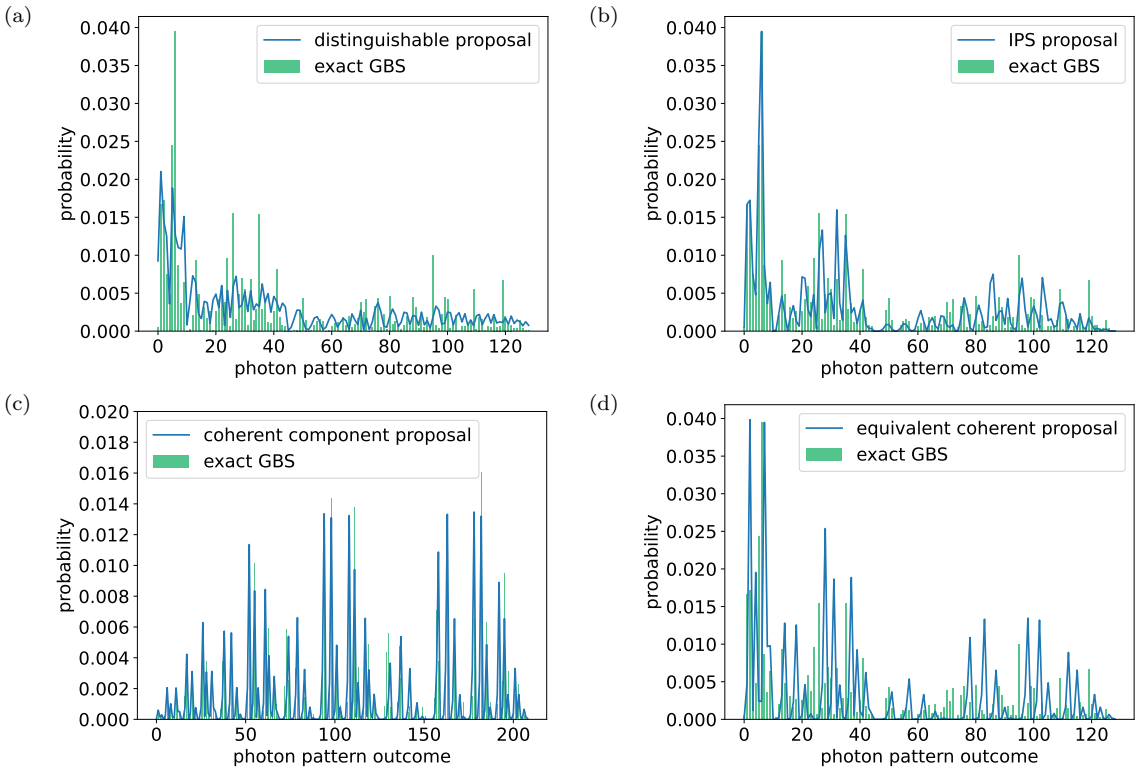

Figure 4.2: State dependent proposal distributions for MIS. For an example Haar random unitary, the target distribution is shown by the green bars. Four proposal distributions are illustrated by the blue lines. In (a), the proposal is distinguishable squeezed states where the probability of photons from the source remains the same as for the target but the photons act distinguishably through the unitary. The proposal distribution in (b) is independent pairs and singles (IPS). Here we assume the photons are produced independently in pairs from squeezing and individually from displacement. This changes the probabilities of n-photon events so we rescale the distribution such that the probability of each photon number is the same as in the target distribution. This graph depicts a GBS distribution with no displacement in the input states so the proposal is exactly the same for two-photon events. Graphs (c) and (d) depict a coherent state proposal where we ignore the squeezing in our state and use only the probability contribution from the displacement in (c) and find the equivalent displacement such that the average number of photons from each source is the same as for the squeezed state in (d). The proposal in (c) is only valid if there is displacement in the input states and so the graph plotted here is for equal contributions from squeezing and displacement. The other graphs are for the case of no displacement.

provide the most efficient sampling. For all testing, we fix the squeezing as equal in all modes (with a value of one) and sometimes include displacement which is also equal in all modes. We also test everything for a fixed number of photons, rather than allow the number of photons to be sampled, for simplicity. Because of this the two proposed coherent states are equivalent and we just test the one coherent proposal.

4.3.1 Acceptance rate test

When proposing a state y to add to a Markov chain, with previous state x, it is added with probability given by

$$\Pr(\text{accept } y|x) = \min\left(1, \frac{f(y)g(x)}{f(x)g(y)}\right). \tag{4.46}$$

So the overall probability of accepting a state independent of what state is proposed and the previous state is given by

$$\begin{aligned}\Pr(\text{accept}) &= \sum_{x,y} a(x)g(y)\min\left(1, \frac{f(y)g(x)}{f(x)g(y)}\right), \\ &= \sum_{x,y} \frac{a(x)}{f(x)}\min\left(f(x)g(y), f(y)g(x)\right),\end{aligned} \tag{4.47}$$

where $a(x)$ is the distribution from which the previous state x was sampled. The chain is started by sampling from the proposal distribution and so at the start the approximate distribution $a(x) = g(x)$. Once the chain has converged, the previous state was sampled from the target distribution: $a(x) = f(x)$. In the above equation $a(x)/f(x)$ is smallest on average when $a(x) = f(x)$ and largest when they are the most different. As the approximate distribution starts at the proposal and converges towards the target, the probability of accepting converges from its highest value when $a(x) = g(x)$ to its minimum value when $a(x) = f(x)$. Therefore the value of the probability of accepting a proposed state is indicative of how much it has converged. This will be used in the next section to determine the burn-in time. However, just the average probability of accepting is a good indication of how quickly the chain converges and how often a new state is added which determines the thinning interval. The higher the probability of accepting, the faster the chain is able to change and converge. Therefore we used this first to compare the proposal distributions to gauge which are generally better.

The probability of accepting can be numerically estimated by running the MIS chain many times and noting how many times the proposed state was accepted at each point in the chain. This requires running a lot of chains to get a good estimate but does not necessitate a brute force calculation of the whole probability space. This means we are able to test this beyond where we can brute force sample. Therefore we use this to compare the various proposal distributions.

We ran 100 chains for different Haar random unitaries of length 1000 and find the average rate of accepting for a range of numbers of modes and photons. These values are plotted in fig. (4.3). These give rather interesting results with the most notable observations being that the uniform and peaked distributions both perform better without displacement contrary to the IPS and coherent distributions. It is expected that the IPS and coherent distributions become closer to the target

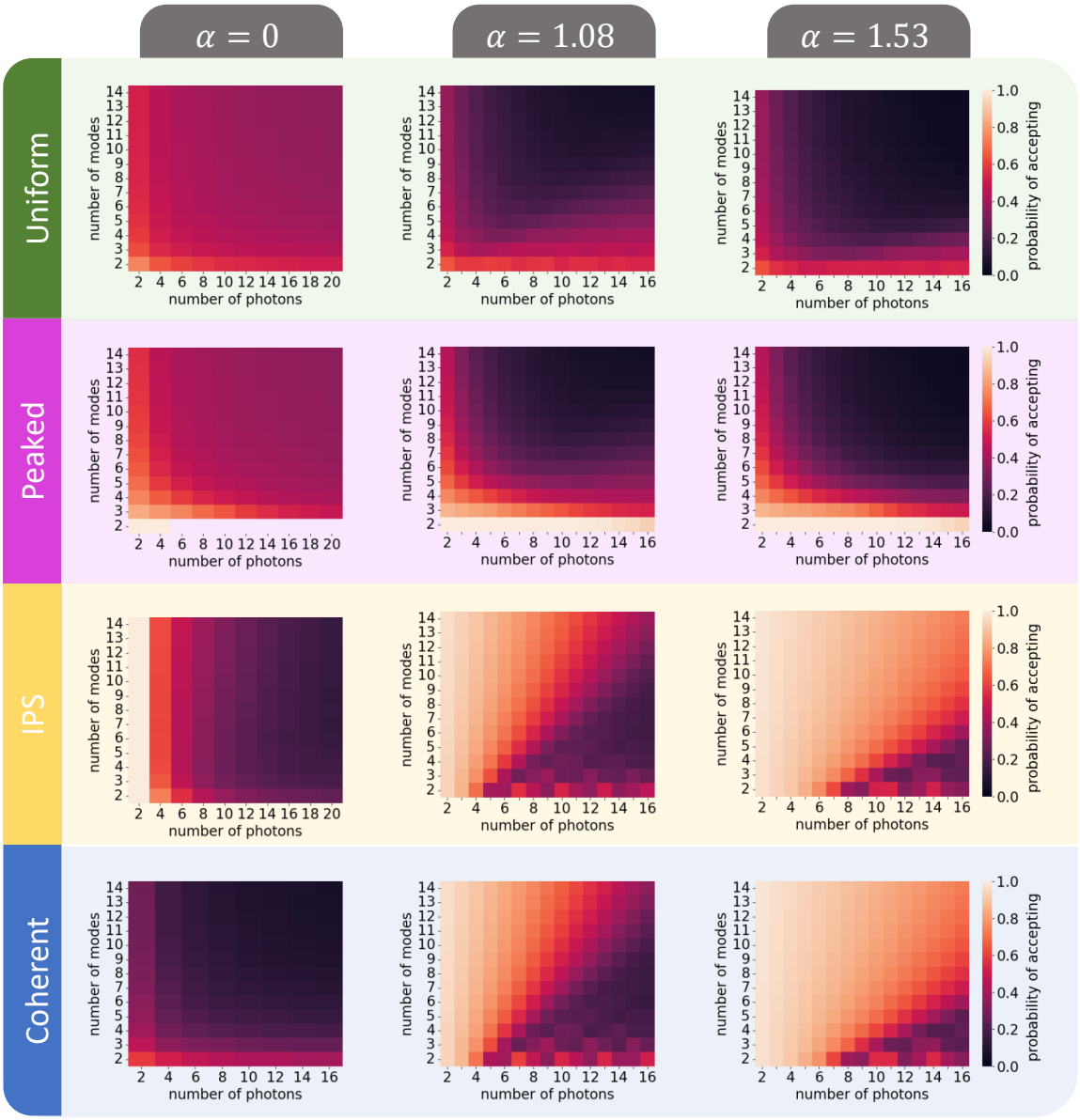

Figure 4.3: The probability of accepting in MIS. Plots are shown for the proposal distributions: uniform, peaked, IPS and coherent, and a selection of displacement parameters: no displacement, $\alpha = 1.08$ and $\alpha = 1.53$. These displacement parameters are chosen such that the mean number of photons ($\sinh^2 r + |\alpha|^2$) has no, less and more contribution from displacement ($|\alpha|^2$) than squeezing ($\sinh^2 r$) respectively (squeezing parameter $r = 1$). The heatmaps show the estimated probability of accepting. The uniform and peaked distributions are worse when displacement is increased, whereas the IPS and coherent distributions perform better. The IPS and coherent distributions are extremely similar when displacement is included.

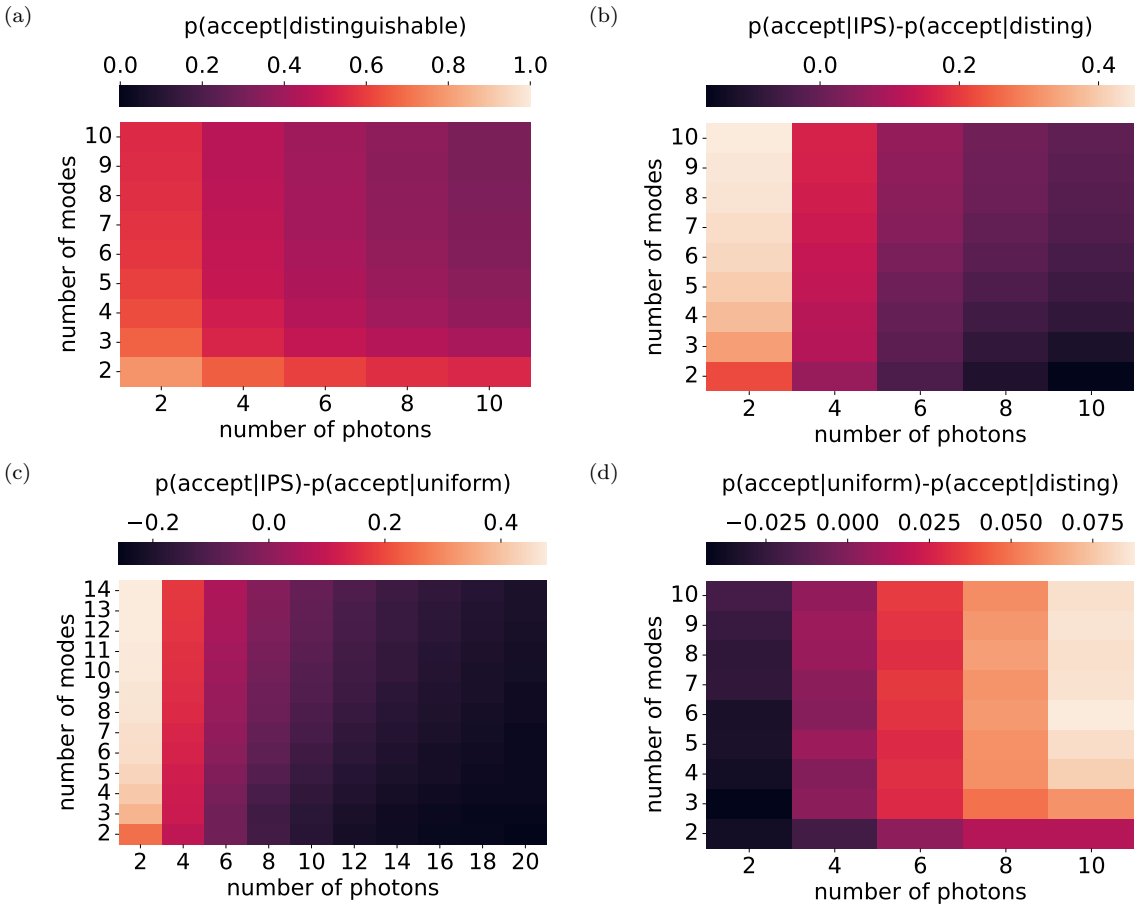

Figure 4.4: Comparison of the probability of accepting. Plot (a) shows the distinguishable proposal distribution with little variation in the plotted problem sizes, but getting slightly worse with increasing numbers of photons or modes. Plot (b) shows the acceptance rate from IPS minus that from the distinguishable proposal to more clearly show the comparison. The data suggests that IPS is better when there is a smaller density of photons per mode but the distinguishable proposal would be better otherwise. Plot (c) shows the acceptance rate from IPS minus the uniform. This follows a similar trend where the IPS is preferable in the low density region. Finally plot (d) shows the value for the uniform minus the distinguishable proposals. Generally this shows that the uniform is better, especially as the number of photons or modes increases.

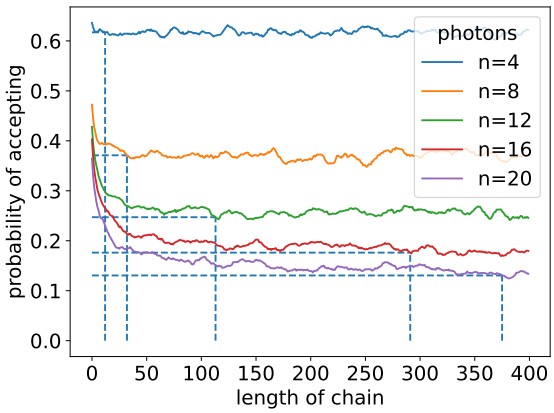

Figure 4.5: The probability of accepting over the length of the chain. This is an example plot for how the probability of accepting changes as the chain converges. Each point is found by the proportion of times the proposed state was accepted at that point in the chain out of 100 runs each with a different Haar random unitary. As the data is very noisy, we averaged the probability over 10 consecutive points in the chain. This example is for the IPS proposal in 10 modes with no displacement. From this plot the required burn-in is found to satisfy that the acceptance rate is within 0.001 of the final value, which is found by averaging over the last 500 states in the 1000 length chain. These points are indicated by the blue dashed lines.

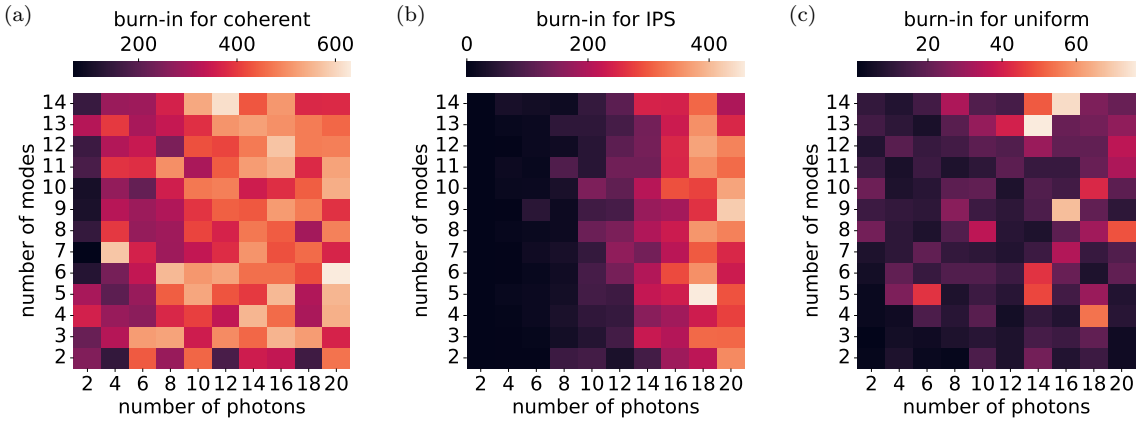

Figure 4.6: Determining the required burn-in. Plots (a), (b) and (c) show the estimated burn-in required for the coherent, IPS and uniform distributions respectively. There is some fluctuation in this data which may hide any patterns in the burn-in times. However, for the IPS distribution, it is clear that the burn-in does not seem to increase as the number of modes increase but is significantly affected as the number of photons increase. In general, it seems that the uniform distribution converges much faster than the others with the coherent distribution being the worst.

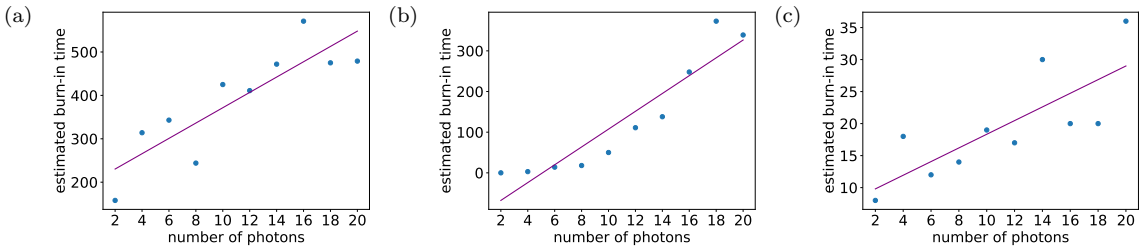

Figure 4.7: The burn-in time as a function of the number of photons. This is a plot of one row ($M = 10$) in fig. (4.6) to make clearer the relationship between the burn-in time and the number of photons for (a) coherent, (b) IPS and (c) uniform proposal distributions. The data points follow linearly increasing trends although there is a lot of variance. We plot the lines of best fit for each proposal distribution and the equations for the burn-in are $\tau = 17.7N + 195.00$, $\tau = 21.9N - 111.67$, and $\tau = 1.07N + 7.67$ for the coherent, IPS and uniform distributions respectively. This supports the previous observation that the uniform distribution is more efficient even as the number of photons increases beyond the testable region.

distribution as the displacement dominates and in the limit of infinite displacement, both become equal to the target distribution. However, when the displacement is set such that the mean number of photons from the displacement is just less than from the squeezing, the two plots look extremely similar which is somewhat surprising. Further, having a high density of photons per mode is detrimental to the efficiency of these proposal distributions. However for the uniform proposal, the high numbers of photons have better acceptance rates when there is displacement. It is worth noting that when adding displacement, the size of the sample space significantly increases as odd numbers of photons are also allowed and therefore a decrease in efficiency could be due to the increase in sample space rather than the direct effect of displaced states.

Calculating the proposal probabilities for the distinguishable squeezed states proposal is very slow and so we plot fewer modes and photons for only zero displacement in fig. (4.4). The acceptance rate is quite high but it is difficult to compare to the other distributions by eye. Therefore we also include comparison plots to show the difference between the distinguishable, IPS and uniform proposal distributions. We find that different proposals suit different combinations of the number of modes and photons.

The distinguishable distribution is slow to calculate and seems like it is often outperformed by just the uniform disrtribution and so we do not continue to analyse it for the burn-in and thinning interval. Also the peaked distribution does not show signs of notable improvement (in fact it sometimes looks worse) and due to the extra overhead of the hill climbing to estimate the maxima values, we do not consider this distribution worth investigating further.

4.3.2 Testing the burn-in time

As the Markov chain converges over time to the stationary distribution and we wish to take samples from this distribution, we need to know how long it takes for the chain to have converged. In theory, the chain will continue to converge forever getting asymptotically closer to the distribution, so we need to choose how far we allow our distribution to be from the target. This is an arbitrary choice,

where we can choose for example the maximum TVD we are willing to concede. We use several tests to determine how much of the chain we need to discard as a function of the number of modes and photons. We use this to predict the average runtime of MIS. Tests exist that can find the burn-in from the chain itself, however these methods scale with the size of the sample space which in this case is combinatorial and so we cannot apply these post-sampling tests [74, 75].

We can investigate the required burn-in time τ by estimating the probability of accepting at all burn-in times up to a maximum. This can be done by running many independent Markov chains and finding the fraction of times at each point in the chain the proposed state was accepted. Then we choose some ϵ such that we want to find the burn-in time at which the estimated acceptance probability is within ϵ of the minimum acceptance once the chain has converged. This is assumed to be given by the average acceptance probability over the end of the chain which is a good estimate as long as the acceptance probability is approximately constant by the end of the chain. Choosing a smaller ϵ enforces a better convergence to the target distribution.

An example of the estimated acceptance probability for each point in the Markov chain is shown in fig. (4.5). From the estimated probability of accepting at each point in the chain, we find the length of the chain required such that the probability of accepting at that point is within 0.001 of the final probability. From the graph it is clear that there is quite a lot of uncertainty likely caused by the finite sample size but also possibly by variation in different Haar random unitaries. This will have an impact on the accuracy of the determined burn-in times. This process is run for all numbers of modes and photons in our testing limits for the three proposal distributions: IPS, uniform and coherent and this is plotted in fig. (4.6). Although the data is noisy, there is a strong suggestion that the uniform distribution requires the smallest burn-in time and so is the most efficient. To see the scaling with the number of photons more clearly, we plot the acceptance rate as a function of the number of photons for 10 modes for each probability distribution in fig. (4.7). These show reasonable aligning with linear trends with the uniform having a much shallower gradient.

4.3.3 Testing the thinning interval

As mentioned previously, because there is a much higher probability of adding the same state to the chain than sampling that state independently there is a significant chance of repeating the same state multiple times in a chain which results in correlation between samples taken from the chain. It has been suggested that caching the samples and rearranging can be used to remove the correlation [76]. However, the order of the samples is not usually important and this method will still have correlation in the sample but not neighbouring ones. Similarly to testing the burn-in time, we run numerical tests in the region that is classically possible and look for scaling patterns to extrapolate.

The probability of accepting gives an indication of how likely two neighbouring states will be the same, but if the proposed state is the same as the current state it is guaranteed to accept it but the state does not change. This is not necessarily a reflection on this state having a high probability as if the proposal probability is high, it will be proposed often and always accepted regardless of the target probability for this state. Therefore a more appropriate metric to use to test the correlation is the probability of repeating the same state. Again this is likely to change with it settling once the

chain has converged. Therefore we take the average probability of repeating a state after a burn-in time of 500 which from the previous section is enough for most problem sizes to have comfortably converged.

The parameter we wish to fix is τ_{thin}, where every τ_{thin}th state is added to our sample. To do this we can choose a value r, the desired probability of repeating, and find r_1, the probability of repeating from one state to the next in the chain. So we need to find τ_{thin} such that $r_1^{\tau_{\text{thin}}} < r$, ie. the probability that a state is repeated all τ_{thin} times. The choice of r is arbitrary with the lower the number, the less correlation in the sample. We fix $r = 0.01$ and estimate the probability of repeating by averaging the proportion of times the state was the same as the previous one for all states after 500 in a chain of length 1000, repeated for 100 different Haar random unitaries. The results are shown in fig. (4.8). To more closely analyse the pattern as a function of the number of photons, we also plot the expected thinning interval for 10 modes for each proposal distribution in fig. (4.9). The data points follow linear trends quite closely and the gradient of the uniform distribution is much smaller than the IPS or the coherent proposals. However, the thinning intervals are reasonable for all problem sizes and extrapolating the worst performing distribution, the coherent proposal, to 100 photons (in 10 modes), we expect a thinning interval of less than 400 which is quite manageable, but the calculation of a hafnian of that size is already impossible. Therefore, even the poorer suggested proposal distributions are still quite useful and a viable option for simulating GBS.

4.4 Discussion

In this chapter we have reviewed several proposal distributions and numerically tested their efficiency whilst fixing the accuracy equally for all distributions. Perhaps surprisingly, the uniform distribution is a good proposal and often has the best convergence and least correlation. However, the best proposal depends on the number of photons and modes with the IPS distributions having a particularly strong reliance on the photon density. In all the cases tested, the scaling of the burn-in time and thinning interval appears to be approximately linear with the number of photons. In some cases increasing the number of modes will likely increase the efficiency further. The burn-in time need only add an overhead once regardless of the sample size and although some of the determined burn-in times were rather high an additional approximately 1000 probability calculations is still much more favourable than needing to calculate the entire distribution which grows very large very quickly. The thinning intervals of around 100 are quite reasonable and slow our sampling down by a factor of 100. When comparing with the acceptance rate in rejection sampling from the previous chapter, it seems as though rejection sampling might be the faster algorithm. However, one of the biggest problems with MIS is the inability to know or even estimate what the TVD is between the sampled and target distributions. We are able to estimate this with rejection sampling, but at present an efficient way to verify the TVD from a Markov chain with an insurmountable state space is an open problem. Therefore, we cannot be sure we are comparing the same accuracy of approximations. The choice of how close to wait for the acceptance probability to get to the final value and the desired probability of repeating are arbitrary choices which do not tell us about the

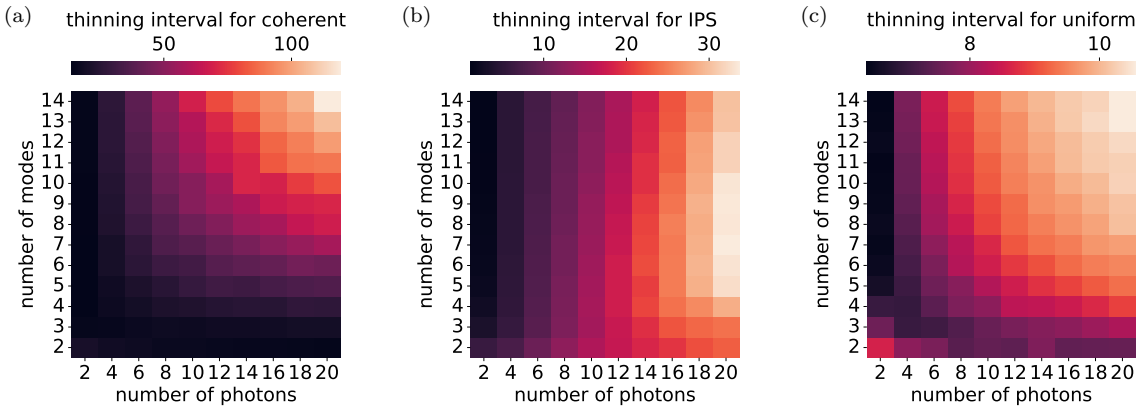

Figure 4.8: Thinning intervals for the MIS proposal distributions. The thinning interval was found such that the probability of repeating is less than 0.01, after a burn-in time of 500. All values are averaged over 100 Haar random unitaries. Plots (a), (b) and (c) show the expected thinning interval for the coherent, IPS and uniform distributions respectively. The coherent distribution seems to suffer from raising quite quickly as both the number of modes and photons are increased, reaching much higher thinning intervals than the other distributions. The IPS distribution again performs much better for regions with a lower density of photons per mode. It scales very well with the number of modes, however, and may be a good candidate in the case of many modes. The uniform distribution has small thinning intervals in all tested problem sizes. The appearance of a slight increase for the small problem sizes is due to there being a much smaller sample space and so if there are only three possible outcomes (as is the case for two photons in two modes), complete independent and uniform sampling should have a third probability of repeating. Requiring the probability to be 0.01 is very low in this case even if the samples were independent. The difference in the expected repeat probability in uncorrelated samples becomes negligible quite quickly as we increase the number of modes or photons.

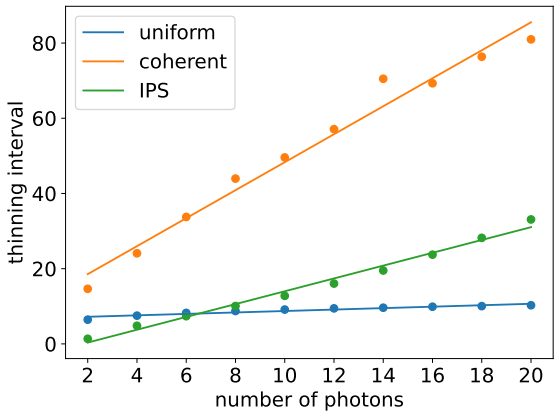

Figure 4.9: Thinning intervals as a function of the number of photons. This is a plot of one row ($M = 10$) in fig. (4.8) to make the relationship between the thinning interval and the number of photons clearer. A linear trend has been fitted to the data points which align quite well. The lines of best fit are given by $\tau_{\text{thin}} = 3.72N + 11.10$, $\tau_{\text{thin}} = 1.70N - 3.06$, and $\tau_{\text{thin}} = 0.19N + 6.82$ for coherent, IPS and uniform distributions respectively.

TVD that these choices provide.

Acknowledgements and contributions

The author's contributions are performing all numerical tests, finding the mathematical derivations for the probability distributions unless otherwise cited, and the sampling algorithm for distinguishable states with a fixed photon number. The independent pairs and singles distribution was independently devised by the author and Bryn Bell, who added the displacement, and the idea for the thinning interval test was courtesy of Jacob Bulmer. The author is grateful to Anthony Laing for supervision, Alex Neville for the introduction to MIS, many useful ideas and conversations and coding support and Levon Chakhmakhchyan for extensive theoretical support regarding GBS. The author also acknowledges Stefano Paesani and Alexandra Moylett for useful discussions.

Chapter 5

Chain-Rule Methods in Boson Sampling

"All good stories deserve embellishment." - Gandalf

The ultimate aim of this thesis is to find an algorithm that samples from the GBS distribution with the same time complexity as the calculation of a single probability for a pure state. Previously we used approximate Monte Carlo methods to do so, but it may also be possible to sample using exact techniques. In this chapter, we study various algorithms for both standard and Gaussian boson sampling using the chain rule of probability.

5.1 Sampling photon by photon in boson sampling

5.1.1 Chain rule of probability

In probability, the chain rule can be used when calculating the probability of a joint set of variables. It formulates the joint probability in terms of conditional probabilities:

$$\Pr(a,b) = \Pr(a)\Pr(b|a) = \Pr(b)\Pr(a|b), \tag{5.1}$$

where we use the notation (a,b) to denote the joint outcome $(a \cap b)$. The above rule can be extended to any number of joint variables:

$$\Pr(x_1,\ldots,x_q) = \Pr(x_1)\Pr(x_2|x_1)\Pr(x_3|x_1,x_2)\cdots\Pr(x_q|x_1,\ldots,x_{q-1}). \tag{5.2}$$

The probabilities of only a subset of the variables (eg. $\Pr(x_i)$) form the marginal probability distribution which completely describes the subset of the variables without knowledge of the other variables. It is equivalent to summing over the joint probabilities for all values of the unknown variables: $\Pr(a) = \sum_b \Pr(a,b)$. Essentially the marginal probability of a subset is the probability independent of what the other variables might be. In contrast, conditional probabilities (eg.

$\Pr(x_i|x_j)$) describe the probability of a subset dependent on the value of the conditional variables. It is possible to have a probability that is both marginal and conditional if it is conditional only on some of the other variables, for example for three variables $\Pr(x_1|x_2)$ is dependent on the outcome of x_2 yet independent of x_3.

When sampling from a joint probability distribution, the chain rule can be used to sample one variable at a time. The first variable is sampled without knowledge of any other variables; that is it is sampled from the marginal probability distribution. Then the next variable is sampled conditional on the outcome of the first variable but without knowledge of any of the remaining variables. This is repeated until the final variable is sampled conditional on all the values previously sampled. The advantage of this method is that it reduces the total number of probabilities needed to be calculated. To sample the joint outcome by calculating the entire distribution if each of the q variables can take r values, a total of r^q joint probabilities need to be calculated. However, by sampling one variable at a time using the chain rule, only qr marginal probabilities are calculated which are chosen depending on the sequence of outcomes.

5.1.2 Standard boson sampling

In boson sampling, the distribution is over joint variables where the variables can be either the mode of each photon in first quantisation or the number of photons in a mode in second quantisation. The size of the sample space is very large; for M modes, if the number of photons N is fixed, because the photons are not ordered, the sample space has $\binom{N+M-1}{N}$ sample points, and if the number of photons is not fixed it is even larger consisting of a sum of the binomial coefficients. Therefore, the chain rule seems like a good candidate for boson sampling. It was introduced as a method of exact sampling for standard boson sampling by Clifford and Clifford [26]. We briefly review this here as a starting point for the aim of classically simulating GBS.

In standard boson sampling the number of photons is fixed by the input state and so the first quantisation is more convenient. As the photons are indistinguishable the pattern corresponds to the sum over the symmetric permutations of the elements. When this state is measured, the pattern can be written in many ways and it is convenient to write it in a non-decreasing order. So a detection pattern of a photon in mode 1 and a photon in mode 2 is written as $\tilde{\boldsymbol{t}} = (1, 2)$ in first quantisation and $\boldsymbol{t} = (1, 1, 0, \ldots, 0)$ in second quantisation, and is measured with probability as determined by eq. (2.88). However, it is equivalent to sample from a distribution where all orderings are allowed in the first quantisation, but the number of ways of permuting the elements is accounted for in the probability. The number of distinct ways of ordering the elements in an output pattern is given by $N!/\tilde{\boldsymbol{t}}!$. If the probability of measuring the output pattern is given by $q(\tilde{\boldsymbol{t}})$, a new distribution that does not restrict the elements to be in non-decreasing order is found where each outcome now has probability $p(\tilde{\boldsymbol{t}}) = q(\tilde{\boldsymbol{t}})\tilde{\boldsymbol{t}}!/N!$. This means that the probability of sampling any permutation of an outcome from this new probability distribution is equal to the probability of sampling that outcome (which is forced to be in non-decreasing order) in the GBS distribution. So simply ordering the elements after they are sampled will result in sampling from the same distribution. This equivalent distribution is more convenient to sample from as the values that each element can have $(1, \ldots, M)$

is independent of the previous outcomes (because it does not need to be greater than or equal to the previous outcome) and the outcomes have more symmetry.

So the task of sampling the output modes of N photons naturally lends itself to using the chain rule to first sample the mode of the first photon and then sample the mode of each subsequent photon. This seems straightforward, however, it is important to remember that each probability using the chain rule is a marginal and conditional probability and so far we have only seen the joint probabilities of all photons. First we consider the definition of conditional probabilities:

$$\Pr(a|b) = \frac{\Pr(a,b)}{\Pr(b)}. \tag{5.3}$$

When sampling the mode m_i of photon i, the probabilities of all possible modes $\Pr(m_i|m_1,\ldots,m_{i-1})$ need to be calculated. The dependence on the mode m_i is only in the numerator in the above equation and the denominator can be found by normalising once the relative probabilities have been calculated for all modes. Therefore, the problem can be reduced to needing to know the marginal probabilities. Clifford and Clifford derived these as follows:

Suppose we want to find the probability of the first n photons in a particular pattern, with the other $N-n$ photons unknown. The marginal probability is given by summing over all the variables $\tilde{t}_i$, where $i > n$ and the probability $p(\tilde{\boldsymbol{t}}) = q(\tilde{\boldsymbol{t}})\tilde{\boldsymbol{t}}!/N!$ can be substituted in to find

$$p(\tilde{t}_1,\ldots,\tilde{t}_n) = \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} p(\tilde{t}_1,\ldots,\tilde{t}_N) = \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} \frac{\boldsymbol{t}!}{N!} \frac{|\mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}})|^2}{\boldsymbol{s}!\boldsymbol{t}!}, \tag{5.4}$$

where each $\tilde{t}_i$ can take any value between 1 and M to denote the mode in which the photon is measured. Using the definition of a permanent and then changing the order of the summations, the probability can be written as

$$\begin{aligned} p(\tilde{t}_1,\ldots,\tilde{t}_n) &= \frac{1}{N!\boldsymbol{s}!} \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} \left(\sum_{\sigma\in\pi(\tilde{\boldsymbol{s}})} \prod_{x=1}^{N} U_{\tilde{t}_x,\sigma(x)} \right) \left(\sum_{\tau\in\pi(\tilde{\boldsymbol{s}})} \prod_{y=1}^{N} U^*_{\tilde{t}_y,\tau(y)} \right) \\ &= \frac{1}{N!\boldsymbol{s}!} \sum_{\sigma,\tau\in\pi(\tilde{\boldsymbol{s}})} \prod_{j=1}^{n} U_{\tilde{t}_j,\sigma(j)} U^*_{\tilde{t}_j,\tau(j)} \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} \prod_{i=n+1}^{N} U_{\tilde{t}_i,\sigma(i)} U^*_{\tilde{t}_i,\tau(i)} \\ &= \frac{1}{N!\boldsymbol{s}!} \sum_{\sigma,\tau\in\pi(\tilde{\boldsymbol{s}})} \prod_{j=1}^{n} U_{\tilde{t}_j,\sigma(j)} U^*_{\tilde{t}_j,\tau(j)} \prod_{i=n+1}^{N} \sum_{k=1}^{M} U_{k,\sigma(i)} U^*_{k,\tau(i)}, \end{aligned} \tag{5.5}$$

where $\pi(\tilde{\boldsymbol{s}})$ is the set of the $N!$ permutations of the vector $\tilde{\boldsymbol{s}} = (\tilde{s}_1,\ldots,\tilde{s}_N)$. In the first line, the definition of a permanent is used and going to the second line the sum over elements in the output pattern and over the permutations of the input pattern are independent and so the order can be changed, and the elements of the unitary that depend only on the fixed modes in the output pattern are moved outside of the sum. To go to the third line, we use the fact that the elements in $\tilde{t}$ are summed over all integers from 1 to M which corresponds to all the ways of choosing $N-n$ ordered elements from 1 to M (with replacement). This is the reason using the distribution $p(\tilde{\boldsymbol{t}})$ rather than

$q(\tilde{\boldsymbol{t}})$ simplifies the problem. The matrix $\boldsymbol{U}$ is an $M \times M$ unitary and therefore $\sum_{m=1}^{M} U_{m,i} U_{m,j}^{*} = \delta_{ij}$. This implies that for the second product to be non-zero $\sigma(i) = \tau(i)$ for all $i > n$, in which case the product is one. This means the values of $\sigma(i)$ and $\tau(i)$ for $i \leq n$ are the same n elements chosen from $\tilde{\boldsymbol{s}}$ but in any order for all non-zero terms. Ignoring all the zero terms, the probability becomes

$$\begin{aligned} p(\tilde{t}_1, \ldots, \tilde{t}_n) &= \frac{1}{N! \boldsymbol{s}!} \sum_{\substack{\sigma,\tau \in \pi(\tilde{\boldsymbol{s}}) \\ \sigma(i)=\tau(i), i>n}} \prod_{j=1}^{n} U_{\tilde{t}_j,\sigma(j)} U_{\tilde{t}_j,\tau(j)}^{*} = \frac{(N-n)!}{N!} \sum_{\tilde{\boldsymbol{c}} \in \mathcal{C}_n} \frac{1}{\boldsymbol{c}!} \sum_{\sigma,\tau \in \pi(\tilde{\boldsymbol{c}})} \prod_{j=1}^{n} U_{\tilde{t}_j,\sigma(j)} U_{\tilde{t}_j,\tau(j)}^{*} \\ &= \frac{(N-n)!}{N!} \sum_{\tilde{\boldsymbol{c}} \in \mathcal{C}_n} \frac{1}{\boldsymbol{c}!} \left(\sum_{\sigma \in \pi(\tilde{\boldsymbol{c}})} \prod_{x=1}^{n} U_{\tilde{t}_x,\sigma(x)} \right) \left(\sum_{\tau \in \pi(\tilde{\boldsymbol{c}})} \prod_{y=1}^{n} U_{\tilde{t}_y,\tau(y)}^{*} \right) \\ &= \frac{(N-n)!}{N!} \sum_{\tilde{\boldsymbol{c}} \in \mathcal{C}_n} \frac{1}{\boldsymbol{c}!} |\mathrm{Perm}(\boldsymbol{U}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{c}}\}})|^2, \end{aligned} \tag{5.6}$$

where $\mathcal{C}_n$ denotes the set of all the ways of choosing n ordered elements from $\tilde{\boldsymbol{s}}$, and $\boldsymbol{c}!$ denotes the product of the factorials of $\boldsymbol{c}$ ($\tilde{\boldsymbol{c}}$ in the second quantisation). For example, if $\tilde{\boldsymbol{s}} = (1, 2, 2, 3)$, $n = 2$ and $M = 3$, we know $\boldsymbol{s} = (1, 2, 1)$ and there are 6 (non-unique) elements in $\mathcal{C}_n$ which are $(1, 2) \times 2$, $(1, 3)$, $(2, 2)$, $(2, 3) \times 2$. For $\tilde{\boldsymbol{c}} = (1, 2)$, $\boldsymbol{c}! = 1! \times 1! \times 0! = 1$ but for $\tilde{\boldsymbol{c}} = (2, 2)$, $\boldsymbol{c}! = 0! \times 2! \times 0! = 2$. In the first line, the sum is reduced to not include elements $\sigma(i)$ and $\tau(i)$ for $i > n$ as the product term is independent of these values. As all these terms contribute the same value, we instead include the factor $(N-n)!\boldsymbol{s}!/\boldsymbol{c}!$ which is the number of ways of arranging the elements such that the first n elements of σ and τ are a particular pattern ($\tilde{\boldsymbol{c}}$), while the last $N-n$ elements in both are the same as each other. The $(N-n)!$ comes from all the ways of permuting the last $N-n$ elements of both σ and τ simultaneously as the order doesn't matter but they must be the same. The $\boldsymbol{s}!$ is due to being able to permute values that are the same without changing anything. However, the $\boldsymbol{c}!$ must be included to remove the permuting of identical elements in $\tilde{\boldsymbol{c}}$ as they are also included in the sum over $\mathcal{C}_n$. In the second and third lines, the expression is simply rearranged to write the marginal probability in terms of a permanent.

The complexity of calculating a marginal probability using the expression above is $O(n3^n)$. Each permanent is of an $n \times n$ matrix and so takes time $O(n2^n)$ and there are $\binom{N}{n}$ elements in the set $\mathcal{C}_n$ which needs to be summed over. For photon n, the probabilities of that photon being in all of the M modes need to be found which involves the calculation of $O[\binom{N}{n}]$ $n \times n$ permanents. The process must be repeated for all N photons, with increasing size of permanents. Using an identity [77], the total time complexity to sample each photon is found to be

$$\sum_{n=0}^{N} \binom{N}{n} n 2^n M = \frac{2}{3} N 3^N M = O(N 3^N M). \tag{5.7}$$

This time complexity is significantly worse than the complexity of calculating a single joint probability which is $O(N2^N)$. However, the sum over the combination group $\mathcal{C}_n$ can be removed by effectively sampling that group first and further tricks using Gray code can be used to improve the complexity to $O(N2^N) + O(MN^2)$. This is derived in [26], but this step is unnecessary for this

thesis and so not detailed here.

5.1.3 Gaussian boson sampling

Sampling using the chain-rule method is almost as efficient as the calculation of one joint probability and as such it seems a promising route to simulate Gaussian boson sampling too. In GBS, there is an extra degree of freedom in the number of photons, but this is not a problem as this can be sampled first as described in section (3.2.1) and then a pattern can be sampled conditional on the total number of photons as done previously in rejection sampling and metropolised independence sampling. The photon pattern can be sampled photon by photon following the same procedure as above for standard boson sampling where the mode of each photon is sampled conditional on the modes of all the previous photons. However, the marginal probabilities need to be replaced to reflect the correct probabilities for GBS. We can search for an expression for the marginal probabilities by summing over the joint variables as before. For simplicity we take the case of pure states with squeezed vacuum states:

$$\begin{aligned} p(\tilde{t}_1,\ldots,\tilde{t}_n) &= \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} p(\tilde{t}_1,\ldots,\tilde{t}_N) = \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} \frac{\boldsymbol{t}!|\mathrm{Haf}(\boldsymbol{B}^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{t}}\}})|^2}{N!\boldsymbol{t}!\prod_m \cosh^2 r_m} \\ &= \frac{1}{N!\prod_m \cosh^2 r_m} \sum_{\tilde{t}_{n+1},\ldots,\tilde{t}_N} \left(\sum_{\sigma\in S_{\tilde{\boldsymbol{t}}}} \prod_{i=1}^{N/2} \frac{B_{\sigma(2i-1),\sigma(2i)}}{(N/2)!2^{N/2}}\right)\left(\sum_{\tau\in S_{\tilde{\boldsymbol{t}}}} \prod_{i=1}^{N/2} \frac{B^*_{\tau(2i-1),\tau(2i)}}{(N/2)!2^{N/2}}\right) \end{aligned} \tag{5.8}$$

where $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^\top$ and $\boldsymbol{R} = \oplus_m \tanh r_m$. Here a problem arises that was not present for the standard boson sampling case. The symmetric group depends now on the output pattern and so the summations cannot so easily be swapped and both indices of the matrix $\boldsymbol{B}$ depend on the element σ from the symmetric group which means the elements corresponding to the marginal elements $\tilde{t}_1,\ldots,\tilde{t}_n$ cannot be separated from the unknown variables $\tilde{t}_{n+1},\ldots,\tilde{t}_N$.

This means the marginal probabilities of a subset of photons cannot be found so easily as for the standard boson sampling case. A simplification of the marginal probabilities was found by Wu et al [78]. They found a time complexity of $O(M\sinh^2 r + \mathrm{poly}(N)2^{8N/3})$ which is reduced to $O(M\sinh^2 r + \mathrm{poly}(N)2^{2N})$ for an algorithm that uses a memory overhead of $O(M2^N)$. These are both improvements on the brute force approach but significantly slower than the time complexity to calculate one joint probability for a pure state which is $O(N^3 2^{N/2})$. In the case of the algorithm with exponential memory requirements this will cause its own limit for scaling due to a lack of memory and so is not useful beyond a certain size. Although a more efficient method for calculating the marginal probabilities is not obvious, there is no reason why it cannot be possible. It is still an open problem whether sampling a photon at a time could provide a method of simulating GBS with reasonable time complexity.

5.2 Sampling mode by mode in Gaussian boson sampling

Although the marginal probabilities of a subset of the *photons* is as of yet not useful in terms of time complexity, a Gaussian state across a subset of the modes is easy to describe and therefore the marginal probabilities of a subset of the *modes* are more convenient. The chain-rule sampling method can easily be applied in the second quantisation notation, where each variable sampled in turn is the occupancy number of each mode. This method was utilised by Quesada and Arrazola [79] to find an algorithm for simulating GBS in time complexity $O(MN^3 2^N)$, a significant improvement on the brute force approach and, apart from polynomial factors, the same complexity as the calculation of a *mixed* state probability. We review this algorithm in the following section as a starting point for the next algorithm.

5.2.1 Sampling with mixed probabilities

Recall that an M-mode Gaussian state can be completely characterised by its vector of means, $\bar{\boldsymbol{x}}$ (or $\bar{\boldsymbol{\zeta}}$), and covariance matrix, $\boldsymbol{V}$ (or $\boldsymbol{\Sigma}$), where the quadrature values $\boldsymbol{x} = \binom{\boldsymbol{q}}{\boldsymbol{p}} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{V})$ are distributed according to the multivariate normal distribution. It is a property of multivariate normal distributions that the marginal distribution over a subset of the variables is also normally distributed with the reduced vector of means and covariance matrix obtained by simply keeping the included variables in both the vector of means and the covariance matrix [80]. For example if variables X_2 and X_5 are included in the marginal distribution, the new vector of means comprises elements $\bar{\boldsymbol{x}}_2$ and $\bar{\boldsymbol{x}}_5$ while the covariance matrix is formed by keeping the second and fifth rows and columns. For Gaussian states, each mode m has two corresponding variables: q_m and p_m. In the notation used throughout this thesis, for mode m these are elements m and $M+m$ respectively in the vector of means and the corresponding rows and columns in the covariance matrix. The state can be equally described in the complex picture in terms of the creation and annihilation operators rather than the quadrature operators.

The probability of detecting a photon pattern $\boldsymbol{t}$ is given in eq. (2.110) for input squeezed vacuum states and eq. (2.111) for input squeezed coherent states which for convenience we repeat here:

$$\Pr(\boldsymbol{t}|\bar{\boldsymbol{x}} = \boldsymbol{0}) = \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \frac{\mathrm{Haf}(A^{\{\tilde{\boldsymbol{t}}',\tilde{\boldsymbol{t}}'\}})}{\prod t_m!}, \qquad \Pr(\boldsymbol{t}|\bar{\boldsymbol{x}} \neq \boldsymbol{0}) = \frac{\exp\left(\bar{\boldsymbol{\zeta}}^\dagger \boldsymbol{\Sigma}_Q^{-1} \bar{\boldsymbol{\zeta}}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_Q)}} \frac{\mathrm{Lhaf}(\boldsymbol{\gamma}^{\{\tilde{\boldsymbol{t}}'\}}, \boldsymbol{A}^{\{\tilde{\boldsymbol{t}}',\tilde{\boldsymbol{t}}'\}})}{\prod t_m!}, \tag{5.9}$$

where $\boldsymbol{\Sigma}_Q = \boldsymbol{\Sigma} + \frac{1}{2}\boldsymbol{I}$, $\boldsymbol{A} = \boldsymbol{X}(\boldsymbol{I} - \boldsymbol{\Sigma}_Q^{-1})$ and $\boldsymbol{\gamma} = \bar{\boldsymbol{\zeta}}^\dagger \boldsymbol{\Sigma}_Q^{-1}$. Clearly the probability of measuring an output photon pattern is a function of the pattern, the vector of means and covariance matrix only. Therefore, it is possible to calculate the marginal probability of an output pattern in a subset of modes, $\mathcal{A}$, by simply using the reduced vector of means $\bar{\boldsymbol{\zeta}}^{\{\mathcal{A}\}}$ and covariance matrix $\boldsymbol{\Sigma}^{\{\mathcal{A}\}}$ corresponding to the subset in the above equations. It is important to note that by taking the reduced covariance matrix, the matrix $\boldsymbol{A}$ is no longer the direct sum of two smaller matrices ($\boldsymbol{A} \neq \boldsymbol{B} \oplus \boldsymbol{B}^*$) in general, because the reduced state is no longer pure. As a result the (loop) hafnian of a $2N \times 2N$ rather than an $N \times N$ matrix needs to be found. So the time complexity for

a marginal probability is the same as for a mixed state joint probability ($O(N^3 2^N)$).

When using the chain rule to sample, for each variable the probability should be found for every value it can take and then it is easy to sample from a fully known distribution. In the case of sampling the output mode of a photon as done in the Clifford and Clifford algorithm for standard boson sampling, the number of modes is finite and there are no problems. However, in GBS when sampling the occupation number of a mode, in theory there is no upper bound to the number of photons detected from squeezed input states. This is not a significant problem though as for a high enough photon number the probability of detecting more photons is small and can be ignored. Therefore a cut-off photon number n_c should be chosen such that in each mode $\Pr(t_i > n_c) < \epsilon$. Of course it is always necessary to choose an overall cut-off number of photons across all modes N_c to be able to sample in a finite time. However the overall limit does not affect the relative probabilities of the events below the cut-off whereas a cut-off on each mode does because it removes only some events with each photon number. The algorithm is outlined below:
For mode $m = 1$ to M:

1. Find the reduced vector of means $\bar{\boldsymbol{\zeta}}^{(m)}$ and covariance matrix $\boldsymbol{\Sigma}^{(m)}$ for the first m modes.

2. Find the corresponding Husimi matrix $\boldsymbol{\Sigma}_Q^{(m)}$ and matrix $\boldsymbol{A}^{(m)}$.

3. With all previous outcomes fixed, calculate the marginal probabilities of mode m containing 0 to n_c photons conditional on the previous modes using the applicable expression in eq. (5.9):

$$\Pr(t_m | t_1^*, \ldots, t_{m-1}^*) = \frac{\Pr(t_1^*, \ldots, t_{m-1}^*, t_m)}{\Pr(t_1^*, \ldots, t_{m-1}^*)}, \tag{5.10}$$

   where the stars denote outcomes fixed in previous steps and t_m ranges from 0 to n_c. As the denominator is constant with respect to mode m, it is not necessary to calculate it and the distribution can be sampled by renormalising after evaluating the numerator for all values of t_m.

4. Sample the value of t_m according to the distribution found in the previous step.

This algorithm makes use of properties of Gaussian states to elegantly sample from a GBS distribution, but has three main drawbacks. The first is that by requiring a cut-off photon number in each mode a truncation error is introduced. This can be removed using the idea of 'overloading' introduced by Qi et al [81]. This is an extension of GBS such that a single mode can detect up to n_c photons or overload which is a separate event that does not distinguish between the number of photons above the cut-off in any mode. To sample this with the above method, step 3 needs to be adapted such that the probabilities are found up to n_c photons and the probability of any number above this cut-off is also found. The probability of a number of photons above the cut-off is the sum of an infinite number of probabilities (because there is an infinite number of possible photon numbers above the cut-off) which is of course not a practical way to determine the probability of overloading in a mode. However, using the fact that probabilities must sum to 1, if the normalised probabilities are found up to the cut-off photon number, the remaining probability is easily found

by subtracting the sum of the probabilities below from 1. This means that the denominator in eq. (5.10) must also be calculated, but this comes with no overhead as it was found in the previous step. Hence, this algorithm can be used to simulate overloaded GBS with no error or time overhead. It can also be used to simulate GBS up to some overall cut-off N_c without a truncation error. In this case, every time an overloaded mode is sampled, that sample is abandoned and the algorithm is repeated until a valid sample is found. Assuming the probability of overloading is small (because the cut-off in each mode is sufficiently high) this incurs only a small time penalty in occasionally needing to reject samples, but is the fastest completely exact simulation of GBS.

The second drawback is that the number of photons cannot be predetermined using this algorithm. Although this may not be considered a problem as the number of photons is not fixed in the quantum device and therefore sampling without a fixed photon number is true to the experiment, it may be useful for benchmarking or certain applications to be able to sample a set number of photons. This remains an open problem and an algorithm that sampled the photons sequentially rather than the modes would fix this. Otherwise, a similar algorithm to that presented in alg. (3) might be used to fix the number of photons. Here the probability of t_m photons in mode m must include the probability of the photon pattern up to that mode *and* the probability of $N - \sum_{j=1}^{m} t_j$ photons in the remaining modes, where they are now conditional on each other. This raises the problem of how to find the probability of n photons in a subset of the modes, conditional on the outcomes in the other modes. At the moment this is an open problem.

The final limitation of this algorithm is the time complexity. As the algorithm relies on the calculation of mixed state probabilities, even if the full mode state is pure, this limits the overall complexity of the algorithm to $O(N^3 2^N M)$, where the factor of M comes from the case where an $N \times N$ (loop) hafnian needed to be calculated in every mode. This limitation is addressed in the following section.

5.2.2 Sampling with pure probabilities

It is possible to reduce the problem of sampling a photon output pattern from a *mixed* state to sampling from a *pure* state. The covariance matrix of a mixed state can be split into two matrices $\boldsymbol{V} = \boldsymbol{T} + \boldsymbol{W}$, where $\boldsymbol{T}$ is the covariance matrix of a pure state and $\boldsymbol{W}$ is a positive semidefinite matrix, using the Williamson decomposition [82]. Using this, a mixed state ρ with vector of means $\bar{\boldsymbol{x}}$ and covariance matrix $\boldsymbol{V}$ can be written as a weighted integral over pure states [83],

$$\rho = \int d\boldsymbol{x}\, p(\boldsymbol{x}) \left|\psi_{x,T}\right\rangle \left\langle\psi_{x,T}\right|, \tag{5.11}$$

where $\left|\psi_{x,T}\right\rangle \left\langle\psi_{x,T}\right|$ is the density matrix of the pure state with vector of means $\boldsymbol{x}$ and covariance matrix $\boldsymbol{T}$ and $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{W})$ and so

$$p(\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^\top \boldsymbol{W}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}})\right)}{\sqrt{\det(2\pi\boldsymbol{W})}}. \tag{5.12}$$

Using this Williamson decomposition the probability of measuring outcome $\boldsymbol{t}$ is given by

$$\begin{aligned}\Pr(\boldsymbol{t}) &= \mathrm{Tr}(\rho\,|\boldsymbol{t}\rangle\langle\boldsymbol{t}|) = \mathrm{Tr}\left(\int d\boldsymbol{x}\, p(\boldsymbol{x})\,|\psi_{x,T}\rangle\langle\psi_{x,T}|\boldsymbol{t}\rangle\langle\boldsymbol{t}|\right)\\ &= \int d\boldsymbol{x}\, p(\boldsymbol{x})\mathrm{Tr}\left(|\psi_{x,T}\rangle\langle\psi_{x,T}|\boldsymbol{t}\rangle\langle\boldsymbol{t}|\right) = \int d\boldsymbol{x}\, p(\boldsymbol{x})\,\langle\boldsymbol{t}|\psi_{x,T}\rangle\langle\psi_{x,T}|\boldsymbol{t}\rangle\\ &= \int d\boldsymbol{x}\, p(\boldsymbol{x})|\langle\psi_{x,T}|\boldsymbol{t}\rangle|^2 = \int d\boldsymbol{x}\, p(\boldsymbol{x})\Pr(\boldsymbol{t}|\,|\psi_{x,T}\rangle).\end{aligned} \tag{5.13}$$

Hence to sample from the mixed distribution it is possible to first sample a displacement vector from $p(\boldsymbol{x})$ and then sample from the photon pattern distribution conditional on the displacement vector outputted. This method reduces the problem of sampling from a mixed state to that of sampling from a pure state. It does not solve the problem of sampling from a pure state as although each probability can now be calculated in $O(N^3 2^{N/2})$, a brute force approach would require a combinatorial number of probabilities to be calculated. Combining this decomposition technique with the mixed state chain-rule algorithm was first addressed by Quesada et al. In their pre-print manuscript they replaced the mixed state sampling step for each mode by sampling a displacement vector over the subset of modes followed by sampling the occupation number of that mode conditional on the pure covariance matrix, sampled displacement and sampled occupation numbers in the previous modes. The following proof was used to show that this method samples from the correct distribution: For each mode m the distribution sampled from $\tilde{p}$ is

$$\begin{aligned}\tilde{p}(t_m|t_1^*,\ldots,t_{m-1}^*) &= \int d\boldsymbol{x}\, p(\boldsymbol{x})\Pr(t_m|t_1^*,\ldots,t_{m-1}^*,\boldsymbol{x}) = \int d\boldsymbol{x}\, p(\boldsymbol{x})\frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}\\ &= \int d\boldsymbol{x}\, p(\boldsymbol{x})\frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*)} = \frac{\int d\boldsymbol{x}\, p(\boldsymbol{x})\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*)}\\ &= \frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*)}{\Pr(t_1^*,\ldots,t_{m-1}^*)} = \Pr(t_m|t_1^*,\ldots,t_{m-1}^*),\end{aligned} \tag{5.14}$$

where from the first to second line, the assumption is made that the probabilities of the previous values are independent of the sampled displacement vector because they were sampled previously and so cannot depend on something that is sampled later.

This algorithm could be applied to standard boson sampling as an alternative to the Clifford and Clifford algorithm. However, during the derivation and numerical testing of the standard boson sampling version, it became clear that the algorithm presented above was not correct. The error made in the derivation is the assumption that the probabilities are independent of the sampled displacement vector. Although the values of the occupation numbers in the previous modes are fixed before the displacement vector is sampled, the probability of measuring that pattern is very much dependent on the displacement. We suggest that the above problem could be fixed by sampling the displacement conditional on the pattern sampled in the previous modes. Then the above proof

becomes

$$\begin{aligned}
\tilde{p}(t_m|t_1^*,\ldots,t_{m-1}^*) &= \int d\boldsymbol{x}\,\Pr(\boldsymbol{x}|t_1^*,\ldots,t_{m-1}^*)\Pr(t_m|t_1^*,\ldots,t_{m-1}^*,\boldsymbol{x}) \\
&= \int d\boldsymbol{x}\,\Pr(\boldsymbol{x}|t_1^*,\ldots,t_{m-1}^*)\frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})} \\
&= \int d\boldsymbol{x}\,\left(\Pr(t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})\frac{p(\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*)}\right)\frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})} \\
&= \int d\boldsymbol{x}\,\frac{p(\boldsymbol{x})\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*)} = \frac{\int d\boldsymbol{x}\,p(\boldsymbol{x})\Pr(t_m,t_1^*,\ldots,t_{m-1}^*|\boldsymbol{x})}{\Pr(t_1^*,\ldots,t_{m-1}^*)} \\
&= \frac{\Pr(t_m,t_1^*,\ldots,t_{m-1}^*)}{\Pr(t_1^*,\ldots,t_{m-1}^*)} = \Pr(t_m|t_1^*,\ldots,t_{m-1}^*),
\end{aligned} \tag{5.15}$$

where from line 2 to 3, we used Bayes' theorem. So this would sample from the correct distribution but it introduces a new problem: how to sample from $\Pr(\boldsymbol{x}|t_1,\ldots,t_{m-1})$. This is in fact non-trivial. It is possible to post-select by sampling the vector $\boldsymbol{x}$ and accepting with probability $\Pr(t_1^*\ldots,t_{m-1}^*)$, however this is a mixed state probability and we have reintroduced the need to calculate quadratically slower probabilities. On top of this, the probability of accepting could be very small meaning the need to sample many times and calculate many mixed state probabilities. Therefore this is not a viable way to sample the conditional displacement. This modification to the algorithm does not provide an obvious way to sample with pure state probabilities. However, we present an algorithm that does not introduce mixedness when sampling mode by mode [84].

The main innovation we employ to maintain purity in the marginal output states is to perform pure Gaussian measurements on the modes not included in the marginal modes. A pure Gaussian measurement on a subset of modes in a pure Gaussian state will leave the unmeasured modes in another pure Gaussian state. Thus it is quadratically faster to calculate the probabilities conditional on the measurement outcome in the other modes than to calculate the marginal probabilities. This is an integral part of the algorithm and so we review heterodyne measurements (a particular pure Gaussian measurement), both of the full M-mode state and partial measurements on a subset, and the marginal state of the remaining modes after measuring.

Recall that a Gaussian state can be completely described by its vector of means $\bar{\boldsymbol{x}}$ and a covariance matrix, either symmetrical over the creation and annihilation operators $\boldsymbol{V}$, normally ordered $\boldsymbol{V}_P$ or antinormally ordered $\boldsymbol{V}_Q$. Heterodyne detection projects the state onto the quadrature basis $\boldsymbol{x}$. To simulate this measurement, an outcome is sampled from the multivariate normal distribution $\boldsymbol{x}\sim\mathcal{N}(\bar{\boldsymbol{x}},\boldsymbol{V}_Q)$. Therefore the probability of an outcome $\boldsymbol{x}$ from a state with vector of means $\bar{\boldsymbol{x}}$ and Q-matrix $\boldsymbol{V}_Q$ is

$$p(\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^\top\boldsymbol{V}_Q^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}})\right)}{\sqrt{\det(2\pi\boldsymbol{V}_Q)}}. \tag{5.16}$$

When measuring a subset of the modes in a multimode state, this is referred to as a partial heterodyne measurement. One way to simulate this is to sample a measurement on all modes and simply discard the outcomes on the modes not measured. Of course you can always sample from a marginal probability distribution by sampling all the variables and discarding the ones not included.

As the displacement is sampled according to a multivariate normal distribution, it is also possible to take advantage of their property that the marginal distribution is described by dropping the corresponding rows and columns in the mean vector and covariance matrix. Because each mode m has two variables q_m and p_m, rows and columns m and $M+m$ are removed for any mode which is not included.

Now we consider the conditional state of the subset of the modes, $\mathcal{A}$, when the other modes, $\mathcal{B}$, are measured with heterodyne detectors. We can again use the property of multivariate normal distributions of how the conditional state affects the mean vector and covariance matrix. As each mode has two corresponding variables, it is convenient to group variables for the modes in $\mathcal{A}$ together. This means the covariance matrix $\boldsymbol{V}$ is written in block form with modes in $\mathcal{A}$ and $\mathcal{B}$ in separate blocks, and similarly for the vector of means. This simply involves permuting the rows and columns in the covariance matrix and the elements of the vector of means, such that the ordering in both is consistent. So the covariance matrix and vector of means are rewritten in the ordering $(\boldsymbol{q}_\mathcal{A}, \boldsymbol{p}_\mathcal{A}, \boldsymbol{q}_\mathcal{B}, \boldsymbol{p}_\mathcal{B})$ as

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_{\mathcal{A}\mathcal{A}} & \boldsymbol{V}_{\mathcal{A}\mathcal{B}} \\ \boldsymbol{V}_{\mathcal{B}\mathcal{A}} & \boldsymbol{V}_{\mathcal{B}\mathcal{B}} \end{pmatrix}, \qquad \bar{\boldsymbol{x}} = \begin{pmatrix} \bar{\boldsymbol{x}}_\mathcal{A} \\ \bar{\boldsymbol{x}}_\mathcal{B} \end{pmatrix}. \tag{5.17}$$

The conditional state in modes $\mathcal{A}$ if the measurement outcome is $\boldsymbol{x}$ is given by [83]

$$\begin{aligned} \boldsymbol{V}_\mathcal{A}^{(\mathcal{B})} &= \boldsymbol{V}_{\mathcal{A}\mathcal{A}} - \boldsymbol{V}_{\mathcal{A}\mathcal{B}}(\boldsymbol{V}_{\mathcal{B}\mathcal{B}} + \tfrac{\hbar}{2}\boldsymbol{I})^{-1}\boldsymbol{V}_{\mathcal{B}\mathcal{A}}, \\ \boldsymbol{x}_\mathcal{A}^{(\mathcal{B})} &= \bar{\boldsymbol{x}}_\mathcal{A} + \boldsymbol{V}_{\mathcal{A}\mathcal{B}}(\boldsymbol{V}_{\mathcal{B}\mathcal{B}} + \tfrac{\hbar}{2}\boldsymbol{I})^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}_\mathcal{B}). \end{aligned} \tag{5.18}$$

So it is possible to sample a subset of the modes in the coherent basis leaving the state in the other modes as given above. Therefore, the state in the remaining subset of modes can now be sampled in the number basis to find a pattern in the modes of interest. Although we are only interested in the marginal photon pattern probabilities, not conditional on the displacement outcome, by first sampling the heterodyne measurement and then the conditional photon pattern, simply discarding the coherent state projection will result in a sample from the marginal probability. This is true because a marginal probability can be obtained by integrating the joint probabilities over the ignored variables, here the set of possible heterodyne outcomes $\boldsymbol{x}_\mathcal{B}$:

$$\Pr(\boldsymbol{t}_\mathcal{A}) = \int d\boldsymbol{x}_\mathcal{B} \Pr(\boldsymbol{t}_\mathcal{A}, \boldsymbol{x}_\mathcal{B}) = \int d\boldsymbol{x}_\mathcal{B} \Pr(\boldsymbol{x}_\mathcal{B}) \Pr(\boldsymbol{t}_\mathcal{A} | \boldsymbol{x}_\mathcal{B}). \tag{5.19}$$

In order to sample each mode sequentially conditional on the previous ones, we need to expand on the above principle. The algorithm is as follows:

1. If the state to be sampled is mixed, use the Williamson decomposition $\boldsymbol{V} = \boldsymbol{T} + \boldsymbol{W}$ to sample a new displacement from $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{W})$. Use this sampled vector as the new vector of means and the covariance matrix $\boldsymbol{T}$ in the following steps.

2. Sample a heterodyne measurement $\boldsymbol{x}$ outcome on all modes but the first. This is done by sampling from the normal distribution $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{V}_Q)$.

3. Find the state in the first mode conditional on the measurement outcome, $\bar{\boldsymbol{x}}_1$, $\boldsymbol{V}_1$, according to eq. (5.18).

4. Calculate the probabilities of the occupation number of the first mode being each value between zero and a cut-off photon number n_c from the state found in the previous step.

5. Sample the number of photons in the first mode from this distribution.

6. Discard x_2.

7. For modes $m = 2$ to M:

   (a) Find the state in the first m modes conditional on the remaining measurement outcome, $\bar{\boldsymbol{x}}_m$, $\boldsymbol{V}_m$.

   (b) Calculate the probabilities, from the state found in the previous step, of the photon pattern in the first m modes with the occupation number of the mth mode being each value between zero and the cut-off photon number n_c and the first $m-1$ occupation numbers fixed from the previous steps.

   (c) Sample the number of photons in the mth mode from this distribution.

   (d) Discard x_m.

We provide the following proof that this algorithm samples from the correct distribution denoted $p(t_1, \ldots, t_M)$. The distribution we are sampling from is given by

$$
\begin{aligned}
&\tilde{p}(t_1, \ldots, t_M) \\
&= \int dx_2 \cdots dx_M \, p(x_2, \ldots, x_M) p(t_1|x_2, \ldots, x_M) p(t_2|t_1, x_3, \ldots, x_M) \cdots p(t_M|t_1, \ldots, t_{M-1}) \\
&= \int dx_2 \cdots dx_M \, p(x_2, \ldots, x_M) \frac{p(t_1, x_2, \ldots, x_M)}{p(x_2, \ldots, x_M)} \frac{p(t_1, t_2, x_3, \ldots, x_M)}{p(t_1, x_3, \ldots, x_M)} \cdots \frac{p(t_1, \ldots, t_M)}{p(t_1, \ldots, t_{M-1})} \\
&= \int dx_2 \cdots dx_M \, \frac{p(t_1, x_2, \ldots, x_M)}{p(t_1, x_3, \ldots, x_M)} \frac{p(t_1, t_2, x_3, \ldots, x_M)}{p(t_1, t_2, x_4, \ldots, x_M)} \cdots \frac{p(t_1, \ldots, t_{M-1}, x_M)}{p(t_1, \ldots, t_{M-1})} p(t_1, \ldots, t_M) \\
&= \int dx_2 \cdots dx_M \, p(x_2|t_1, x_3, \ldots, x_M) p(x_3|t_1, t_2, x_4, \ldots, x_M) \cdots p(x_M|t_1, \ldots, t_{M-1}) p(t_1, \ldots, t_M) \\
&= p(t_1, \ldots, t_M),
\end{aligned}
$$

which is equal to the correct probability distribution as required. From the first to second line, we used the definition of conditional probabilities and simply rearranged in the third line. From the third to fourth line we again used the definition of conditional probabilities and the final step involves integrating over all the displacement parameters in order from x_2 to x_M. Integrating in this order means there is only one term containing the variable which is a conditional probability that must integrate to one over all possible outcomes as it is normalised to unity.

The necessity for a cut-off photon number in each mode introduces an error. This cut-off must be chosen high enough to ensure that the probability missed by this restriction is very small. Otherwise, due to the probability in each mode depending on the displacement vector sampled, this

truncation error can become amplified and has a worse effect on the accuracy than the same cut-off photon number in the previous algorithm that relied on mixed states and was quadratically slower. Similarly to the previous algorithm, it is possible to instead simulate the overloaded case to remove this error. However, this would require calculating a mixed state probability and lose the speed-up. This is because the normalised probability of an occupation number in mode m conditional on the displacement measurement and the previous modes is given by

$$\Pr(t_m|t_1,\ldots,t_{m-1},x_{m+1},\ldots,x_M)=\frac{\Pr(t_1,\ldots,t_m|x_{m+1},\ldots,x_M)}{\Pr(t_1,\ldots,t_{m-1}|x_{m+1},\ldots,x_M)}. \tag{5.20}$$

The numerator is a pure state which gives the probabilities calculated when the mode is sampled, but the denominator is mixed. Therefore it is much more efficient to normalise the distribution after calculating the values up to the cut-off than to calculate the normalisation constant explicitly. In the overload version, it is necessary to find this denominator as the probability of overloading the detector is given by one minus the probabilities of not overloading it. Therefore it is not generally useful to apply this algorithm to the overloading case, however it can be used to test that the algorithm is correct as there is no truncation error that could potentially hide a problem with the algorithm.

We provide numerical testing of all of the GBS algorithms covered in this chapter in fig. (5.2). We tested the algorithms by approximating the probability distribution by taking a very large sample size for numbers of modes and photons that are small enough for the whole distribution to be found. Because there will always be a distance between the correct distribution and the estimated distribution due to having a finite sample size, we include sampling from the correct distribution, known as brute force sampling, to benchmark the expected total variation distance. We test both the mixed state and pure state algorithms for both the case of overloading and not. When the squeezing is lower, the chosen cut-off is high enough that all algorithms quite closely agree. However, when the squeezing is higher, the cut-off is not high enough to avoid truncation errors and these are seen in both algorithms, with the pure state algorithm more affected. However, when applied to the overloaded scenario, all algorithms agree closely, providing evidence that both algorithms work well but can be adversely affected if the chosen cut-off is too low. We also provide a test for just the pure state chain-rule algorithm comparing it to brute force sampling, where we have averaged over more Haar random unitaries to provide less uncertainty. This is shown in fig. (5.1) and again the data points are completely overlapping as expected.

5.3 Sampling mode by mode in standard boson sampling

Given an algorithm can sample from a GBS distribution mode by mode, it seems that it ought to be possible to sample from a standard boson sampling distribution mode by mode too. The process of using the chain rule is the same but we need to derive the marginal probabilities for standard boson sampling instead. This problem is addressed in this section.

Scattershot boson sampling is a simplification of the more general GBS and as such can be

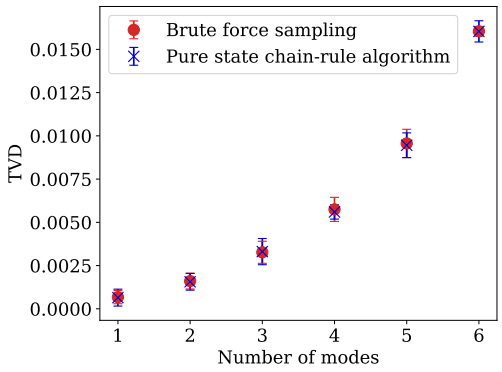

Figure 5.1: Total variation distance when brute force sampling and using the pure state chain-rule algorithm. For a range of number of modes from 1 to 6, the total variation distance (TVD) between two estimated probability distributions and the correct distribution are plotted. The estimated distributions are found by taking a large sample size of 100,000 and assigning probabilities to each outcome given by the proportion of samples with that value. The pure state chain-rule algorithm is the algorithm we have introduced here and the brute force sampling is sampled from the fully known distribution as a benchmark for the expected TVD which arises from a finite sample size. All points are averaged over ten instances with three different Haar random unitaries, and the error bars correspond to the standard deviation. The squeezing parameter was 0.5 with no displacement and the cut-off number of photons in each mode was three. The increase in TVD as the number of modes increases is expected due to the increase in the cardinality of the sample space - there are more possible outcomes so each has a smaller probability and so needs a larger sample size to estimate to the same precision, but here the sample size is consistent. The two distributions align closely providing evidence the algorithm is sampling from the correct distribution.

described as a particular instance of GBS as shown in section (2.4.3.2). Standard boson sampling is simply scattershot boson sampling where the photons were heralded in the correct modes. Thus it can be described as scattershot with post-selection. However, if we sample each mode at a time, rather than post-selecting on a valid output pattern in the heralding modes, we can simply fix the heralding modes first and sample the other modes conditional on this heralded pattern. Therefore, to sample from a standard boson sampling distribution mode by mode, the algorithm for GBS [79] can be used where the unitary is chosen to reduce it to scattershot and the input states are equal squeezed vacuum states, and the output from the heralded modes are fixed as the equivalent input photon pattern in the standard boson sampler. So this problem is reduced to finding the marginal probabilities for the specific case that describes scattershot boson sampling.

In section (2.4.3.2), the derivation is given for the reduction of the hafnian equation for GBS to

$$\Pr(\boldsymbol{t}) = \frac{\tanh^N r}{\cosh^{2M} r} \frac{|\mathrm{Perm}(\boldsymbol{U})^{\{\boldsymbol{t}_d,\boldsymbol{t}_h\}}|^2}{\prod_m t_m!}. \tag{5.21}$$

We follow similar steps to find the marginal probability, but where only the included modes are kept in the covariance matrix.

Consider a scattershot set-up with M heralding modes and M 'output' modes. This corresponds to a GBS set-up with $2M$ modes in total. To simulate standard boson sampling, we must first fix

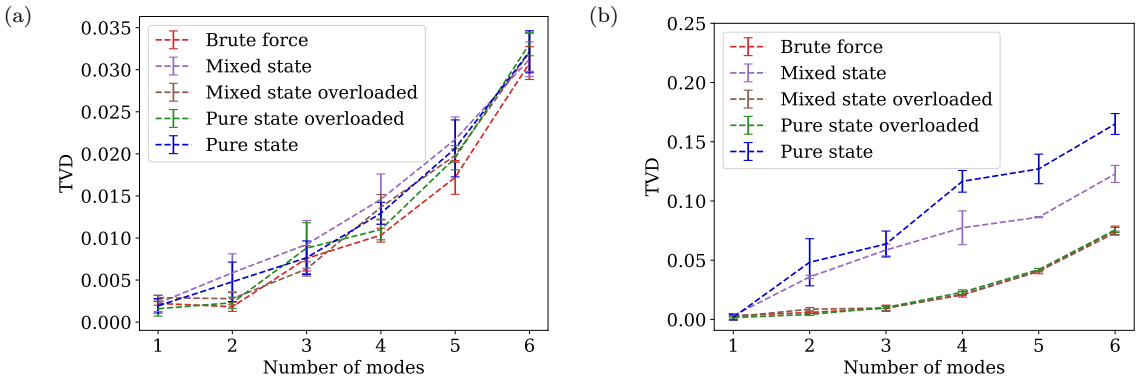

Figure 5.2: Comparison of the GBS chain-rule sampling algorithms. To show the accuracy of the algorithms covered in this chapter, the total variation distance (TVD) between the estimated distribution by sampling with a sample size of 20,000 and the correct distribution is plotted for 1 to 6 modes. All points were averaged over three instances with different Haar random unitaries, with the error bars showing the standard deviation. In (a), the squeezing parameter is 0.5 with an average number of photons per mode of 0.27, whereas in (b) the squeezing parameter is 1 with an average photon number of 1.38. The cut-off photon number was fixed as three for both. This is sufficiently high for the lower squeezing but has a non-negligible effect for the higher squeezing. This is seen in the plots as all algorithms have consistent TVDs in (a), whereas only the overloading algorithms are consistent with the brute force sampling plots in (b). The mixed state chain-rule algorithm has a higher TVD than the expected TVD when sampling from the correct distribution (brute force) due to a truncation error. The pure state chain-rule algorithm suffers even more from a truncation error due to being amplified by the coherent state projection. Both the pure and mixed state chain-rule algorithms with overloading allowed are extremely similar to the brute force sampling TVDs which confirms that the error in the versions without overloading is only due to the cut-off photon number being too low.

the heralding pattern as the desired input pattern $\boldsymbol{s}$ and so we start sampling the output state $\boldsymbol{t}$ from mode $M+1$ conditional on $\boldsymbol{s}$. We denote the mode we are sampling as m where $1 \leq m \leq M$, meaning the mth output mode, not including the heralding modes. For scattershot the squeezing should be equal and so the covariance matrix for the first $M+m$ modes is given by

$$\boldsymbol{\Sigma}_m = \frac{1}{2}\begin{pmatrix} \cosh(2r)\boldsymbol{I} & -\sinh(2r)(\boldsymbol{U}_{\mathrm{GBS}}\boldsymbol{U}_{\mathrm{GBS}}^{\top})_{M+m} \\ -\sinh(2r)(\boldsymbol{U}_{\mathrm{GBS}}\boldsymbol{U}_{\mathrm{GBS}}^{\top})^{*}_{M+m} & \cosh(2r)\boldsymbol{I} \end{pmatrix}, \tag{5.22}$$

where each block is an $(M+m)\times(M+m)$ matrix, $(\boldsymbol{U}\boldsymbol{U}^{\top})_{M+m}$ denotes taking the first $M+m$ rows and columns of the resulting matrix. Recall that the unitary required to simulate scattershot via GBS, $\boldsymbol{U}_{\mathrm{GBS}}$, satisfies the following:

$$\boldsymbol{U}_{\mathrm{GBS}}\boldsymbol{U}_{\mathrm{GBS}}^{\top} = -\begin{pmatrix} \boldsymbol{0} & \boldsymbol{U}^{\top} \\ \boldsymbol{U} & \boldsymbol{0} \end{pmatrix}, \tag{5.23}$$

where $\boldsymbol{U}$ is the unitary in the scattershot set-up. Using this, the marginal covariance matrix can

be expanded as

$$\boldsymbol{\Sigma}_m = \frac{1}{2}\left(\begin{array}{cc|cc} c\boldsymbol{I}^{(M)} & \mathbf{0} & \mathbf{0} & s\boldsymbol{U}_m^\top \\ \mathbf{0} & c\boldsymbol{I}^{(m)} & s\boldsymbol{U}_m & \mathbf{0} \\ \hline \mathbf{0} & s\boldsymbol{U}_m^\dagger & c\boldsymbol{I}^{(M)} & \mathbf{0} \\ s\boldsymbol{U}_m^* & \mathbf{0} & \mathbf{0} & c\boldsymbol{I}^{(m)} \end{array}\right), \tag{5.24}$$

where $\boldsymbol{I}^{(M)}$ is the $M \times M$ identity matrix and we introduce the shorthand $c = \cosh(2r)$ and $s = \sinh(2r)$. Here $\boldsymbol{U}_m$ denotes taking the first m *rows* of $\boldsymbol{U}$. Note that the same notation is used for a product of unitaries where in that case the subscript corresponds to both the included rows and columns. Using this notation allows the simple statement $(\boldsymbol{U}\boldsymbol{U}^\top)_m = \boldsymbol{U}_m\boldsymbol{U}_m^\top$. Although the order is not explicit, $\boldsymbol{U}_m^\top = (\boldsymbol{U}_m)^\top$ and taking the first m rows of $\boldsymbol{U}$ and then transposing or taking the Hermitian conjugate causes it to be the first m *columns* of these matrices.

For the probability, we need to know the $\boldsymbol{\Sigma}_Q$ and $\boldsymbol{A}$ corresponding to the marginal state including up to the mth output mode. To avoid excessive subscripts, we use $\boldsymbol{Q}$ for the Q-covariance matrix rather than $\boldsymbol{\Sigma}_Q$ introduced previously. This matrix is simply found by adding half the identity:

$$\boldsymbol{Q}_m = \boldsymbol{\Sigma}_m + \tfrac{1}{2}\boldsymbol{I} = \frac{1}{2}\left(\begin{array}{cc|cc} (c+1)\boldsymbol{I} & \mathbf{0} & \mathbf{0} & s\boldsymbol{U}_m^\top \\ \mathbf{0} & (c+1)\boldsymbol{I} & s\boldsymbol{U}_m & \mathbf{0} \\ \hline \mathbf{0} & sU_m^\dagger & (c+1)\boldsymbol{I} & \mathbf{0} \\ s\boldsymbol{U}_m^* & \mathbf{0} & \mathbf{0} & (c+1)\boldsymbol{I} \end{array}\right). \tag{5.25}$$

To determine $\boldsymbol{A}_m = \boldsymbol{X}_m(\boldsymbol{I}_m - \boldsymbol{Q}_m^{-1})$, we first can find the inverse of the Q matrix. We again make use of the inverse formula for block matrices repeated here for convenience:

$$\begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{pmatrix}^{-1} = \begin{pmatrix} (\boldsymbol{a}-\boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c})^{-1} & -\boldsymbol{a}^{-1}\boldsymbol{b}(\boldsymbol{d}-\boldsymbol{c}\boldsymbol{a}^{-1}\boldsymbol{b})^{-1} \\ -\boldsymbol{d}^{-1}\boldsymbol{c}(\boldsymbol{a}-\boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c})^{-1} & (\boldsymbol{d}-\boldsymbol{c}\boldsymbol{a}^{-1}\boldsymbol{b})^{-1} \end{pmatrix}, \tag{5.26}$$

Using this identity, the inverse of $\boldsymbol{Q}_m$ is found to be

$$\boldsymbol{Q}_m^{-1} = \left(\begin{array}{cc|cc} (1-t^2)\boldsymbol{I} + t^2\boldsymbol{U}_m^\top\boldsymbol{U}_m^* & \mathbf{0} & \mathbf{0} & -t\boldsymbol{U}_m^\top \\ \mathbf{0} & \boldsymbol{I} & -t\boldsymbol{U}_m & \mathbf{0} \\ \hline \mathbf{0} & -t\boldsymbol{U}_m^\dagger & (1-t^2)\boldsymbol{I} + t^2\boldsymbol{U}_m^\dagger\boldsymbol{U}_m & \mathbf{0} \\ -t\boldsymbol{U}_m^* & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{array}\right), \tag{5.27}$$

where we introduce the shorthand $t = \tanh r$. Note that here the argument of the hyperbolic function is only r rather than $2r$ hidden in c and s shorthands. From here the matrix $\boldsymbol{A}_m$ corresponding to the marginal state can easily be found. Because of the simple structure of both the identity matrix and the swap matrix $\boldsymbol{X}$, keeping only certain rows and columns in these matrices

simply reduces the size but leaves the structure unchanged. The matrix $\boldsymbol{A}_m$ is given by

$$\boldsymbol{A}_m = \boldsymbol{X}_m(\boldsymbol{I}_m - \boldsymbol{Q}_m^{-1}) = \left(\begin{array}{ccc|c} \boldsymbol{0} & t\boldsymbol{U}_m^\dagger & t^2(\boldsymbol{I} - \boldsymbol{U}_m^\dagger \boldsymbol{U}_m) & \boldsymbol{0} \\ t\boldsymbol{U}_m^* & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hline t^2(\boldsymbol{I} - \boldsymbol{U}_m^\top \boldsymbol{U}_m^*) & \boldsymbol{0} & \boldsymbol{0} & t\boldsymbol{U}_m^\top \\ \boldsymbol{0} & \boldsymbol{0} & t\boldsymbol{U}_m & \boldsymbol{0} \end{array}\right). \tag{5.28}$$

In the expression for the probability of a pattern, the hafnian is found for a submatrix of $\boldsymbol{A}_m$ depending on the heralding $\boldsymbol{s}$ and output patterns $\boldsymbol{t}$. However the block structure of $\boldsymbol{A}$ is unchanged when we include rows/columns according to these patterns. So the hafnian can be written as

$$\mathrm{Haf}(\boldsymbol{A}_m^{\{\tilde{\boldsymbol{s}}+\tilde{\boldsymbol{t}}\}}) = \mathrm{Haf}\left(\begin{array}{ccc|c} \boldsymbol{0} & t(\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} & t^2(\boldsymbol{I} - \boldsymbol{U}_m^\dagger \boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \\ t(\boldsymbol{U}_m^*)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hline t^2(\boldsymbol{I} - \boldsymbol{U}_m^\top \boldsymbol{U}_m^*)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} & \boldsymbol{0} & t(\boldsymbol{U}_m^\top)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ \boldsymbol{0} & \boldsymbol{0} & t(\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \end{array}\right), \tag{5.29}$$

where $\boldsymbol{M}^{\{r,c\}}$ denotes including the rows according to r and columns according to c.

Hafnians are unchanged under simultaneous permutations of rows and columns so we permute block rows and columns 2 and 4 (from the 4×4 block matrix) to simplify the hafnian to

$$\begin{aligned} \mathrm{Haf}(\boldsymbol{A}_m^{\{\tilde{\boldsymbol{s}}+\tilde{\boldsymbol{t}}\}}) &= \mathrm{Haf}\left(\begin{array}{cc|cc} \boldsymbol{0} & \boldsymbol{0} & t^2(\boldsymbol{I} - \boldsymbol{U}_m^\dagger \boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & t(\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ \boldsymbol{0} & \boldsymbol{0} & t(\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \\ \hline t^2(\boldsymbol{I} - \boldsymbol{U}_m^\top \boldsymbol{U}_m^*)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & t(\boldsymbol{U}_m^\top)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} & \boldsymbol{0} & \boldsymbol{0} \\ t(\boldsymbol{U}_m^*)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{array}\right) \\ &= \mathrm{Perm}\begin{pmatrix} t^2(\boldsymbol{I} - \boldsymbol{U}_m^\dagger \boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & t(\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ t(\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \end{pmatrix} \\ &= t^{2N}\mathrm{Perm}\begin{pmatrix} (\boldsymbol{I} - \boldsymbol{U}_m^\dagger \boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & (\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ (\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \end{pmatrix}. \end{aligned} \tag{5.30}$$

From the first to second line we used the relation between permanents and hafnians and from the second to third line we factor out t from the top N rows and the first N columns.

The final expression we need in order to evaluate the marginal probabilities is the determinant of the marginal Q matrix. This is given by

$$\begin{aligned} \det(\boldsymbol{Q}_m) =& \det\left(\frac{1}{2}\left(\begin{array}{cc|cc} (c+1)\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & s\boldsymbol{U}_m^\top \\ \boldsymbol{0} & (c+1)\boldsymbol{I} & s\boldsymbol{U}_m & \boldsymbol{0} \\ \hline \boldsymbol{0} & s\boldsymbol{U}_m^\dagger & (c+1)\boldsymbol{I} & \boldsymbol{0} \\ s\boldsymbol{U}_m^* & \boldsymbol{0} & \boldsymbol{0} & (c+1)\boldsymbol{I} \end{array}\right)\right) \\ =& \frac{\det((c+1)\boldsymbol{I}^{(M+m)})}{2^{2m+2M}}\det\left((c+1)\boldsymbol{I} - \begin{pmatrix} \boldsymbol{0} & s\boldsymbol{U}_m^\top \\ s\boldsymbol{U}_m & \boldsymbol{0} \end{pmatrix}\frac{\boldsymbol{I}}{c+1}\begin{pmatrix} \boldsymbol{0} & s\boldsymbol{U}_m^\dagger \\ s\boldsymbol{U}_m^* & \boldsymbol{0} \end{pmatrix}\right), \end{aligned} \tag{5.31}$$

where we've factored out the $\frac{1}{2}$ and used the block matrix determinant formula

$$\det\begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{pmatrix} = \det(\boldsymbol{d})\det(\boldsymbol{a} - \boldsymbol{b}\boldsymbol{d}^{-1}\boldsymbol{c}). \tag{5.32}$$

Further simplification finds

$$\begin{aligned} \det(\boldsymbol{Q}_m) &= \frac{(c+1)^{m+M}}{2^{2m+2M}}\det\left((c+1)\begin{pmatrix} \boldsymbol{I} - t^2\boldsymbol{U}_m^\top\boldsymbol{U}_m^* & \boldsymbol{0} \\ \boldsymbol{0} & (1-t^2)\boldsymbol{I} \end{pmatrix}\right) \\ &= \frac{(c+1)^{2m+2M}}{2^{2m+2M}}\det(\boldsymbol{I}^{(M)} - t^2\boldsymbol{U}_m^\top\boldsymbol{U}_m^*)\det((1-t^2)\boldsymbol{I}^{(m)}). \end{aligned} \tag{5.33}$$

Here we make use of Weinstein-Aronszajn's identity [85]

$$\det(\boldsymbol{I}^{(j)} + \boldsymbol{A}^{(j\times k)}\boldsymbol{B}^{(k\times j)}) = \det(\boldsymbol{I}^{(k)} + \boldsymbol{B}^{(k\times j)}\boldsymbol{A}^{(j\times k)}) \tag{5.34}$$

to change the order of $\boldsymbol{U}_m^\top$ and $\boldsymbol{U}_m^*$, and hence simplify the determinant to

$$\begin{aligned} \det(\boldsymbol{Q}_m) &= \frac{(c+1)^{2m+2M}}{2^{2m+2M}}\det(\boldsymbol{I}^{(m)} - t^2\boldsymbol{U}_m^*\boldsymbol{U}_m^\top)\det((1-t^2)\boldsymbol{I}^{(m)}) \\ &= \frac{(c+1)^{2m+2M}}{2^{2m+2M}}(1-t^2)^{2m} = (\cosh^2 r)^{2m+2M}\frac{1}{(\cosh^2 r)^{2m}} = \cosh^{4M} r, \end{aligned} \tag{5.35}$$

where from the first to second line we used that $\boldsymbol{U}_m^*\boldsymbol{U}_m^\top = (\boldsymbol{U}^*\boldsymbol{U}^\top)_m = (\boldsymbol{I})_m = \boldsymbol{I}^{(m)}$.

We can now input these values into the general equation to find a simpler expression for the marginal probability for GBS in this case:

$$\mathrm{Prob}(\boldsymbol{s}+\boldsymbol{t}) = \frac{\tanh^{2N} r}{\boldsymbol{s}!\boldsymbol{t}!\cosh^{2M} r}\mathrm{Perm}\begin{pmatrix} (\boldsymbol{I} - \boldsymbol{U}_m^\dagger\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & (\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ (\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0} \end{pmatrix}, \tag{5.36}$$

where this is the probability of the output pattern with $\boldsymbol{s}$ in the heralded modes and $\boldsymbol{t}$ in the output modes so the overall pattern is given by $\boldsymbol{s}+\boldsymbol{t}$. In standard boson sampling, the input state is fixed so it is actually the probability of detecting $\boldsymbol{s}+\boldsymbol{t}$ conditional on detecting $\boldsymbol{s}$ that is of interest. The probability of measuring the heralded pattern can be found by inputting $m = 0$ and $\boldsymbol{t} = \{\}$ (empty) in the above expression which reduces the permanent to be of a submatrix of the identity matrix. If there are no repetitions in $\tilde{\boldsymbol{s}}$, the submatrix of the identity is just a smaller identity. However, if there is bunching in the input modes, the submatrix becomes a block diagonal matrix with the diagonal blocks being the $s_m \times s_m$ all-ones matrix. The permanent of a block diagonal matrix is the product of the permanents of the block matrices and the permanent of the all-ones matrix is the factorial of its size. Therefore the permanent of the submatrix of the identity is $\prod_m s_m!$ and the probability of detecting $\boldsymbol{s}$ is

$$\Pr(\boldsymbol{s}) = \frac{\tanh^{2N} r}{\cosh^{2M} r}. \tag{5.37}$$

This probability could also have been found by simply calculating the probability of detecting s_m

photons from a two-mode squeezed state for each mode m. As expected, these expressions agree, providing reassurance for the derived marginal probability expression.

Putting everything together, we finally arrive at the expression for the marginal probability for standard boson sampling:

$$\begin{aligned}\text{Prob}(\boldsymbol{t}|\boldsymbol{s}) &= \text{Prob}(\boldsymbol{s}+\boldsymbol{t})/\text{Prob}(\boldsymbol{s}) \\ &= \frac{\tanh^{2N} r}{\boldsymbol{s}!\boldsymbol{t}!\cosh^{2M} r}\text{Perm}\begin{pmatrix} (\boldsymbol{I}-\boldsymbol{U}_m^\dagger\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & (\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ (\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0}\end{pmatrix}\Bigg/\frac{\tanh^{2N} r}{\cosh^{2M} r} \\ &= \frac{1}{\boldsymbol{s}!\boldsymbol{t}!}\text{Perm}\begin{pmatrix} (\boldsymbol{I}-\boldsymbol{U}_m^\dagger\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{s}}\}} & (\boldsymbol{U}_m^\dagger)^{\{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}\}} \\ (\boldsymbol{U}_m)^{\{\tilde{\boldsymbol{t}},\tilde{\boldsymbol{s}}\}} & \boldsymbol{0}\end{pmatrix}.\end{aligned} \tag{5.38}$$

This can be used to sample each mode sequentially, by sampling from the conditional probability $\text{Prob}(t_m|t_{m-1},\ldots,t_1,\boldsymbol{s})$ for $m=1$ to M. Because the number of photons is fixed, the probability only needs to be found for $t_m = 0,\ldots,N-n_{m-1}$ where n_{m-1} is the number of photons already sampled in the output pattern $t_1,\ldots,t_{m-1}$, and N is the number of input photons in $\boldsymbol{s}$. So unlike for GBS, this is an exact sampling algorithm. However it requires the calculation of the permanent of an $(N+n)\times(N+n)$ matrix for each probability with N input photons and n photons in the included output modes, which is much slower than the Clifford and Clifford algorithm for standard boson sampling.

5.4 Discussion

Chain-rule sampling algorithms can be used to sample from a multidimensional state by sampling either the modes or the photons sequentially. It seems that sampling photon by photon is more suited for standard boson sampling, while mode by mode naturally is incorporated into the Gaussian formalism. However, there is no obvious reason why both methods cannot be applied to either and each has its own advantages. We have provided a mode by mode algorithm for standard boson sampling, but it suffers from a worse complexity than the Clifford and Clifford algorithm. It remains to be shown whether the pure-state chain-rule algorithm for GBS can be applied to standard boson sampling as it would need to be adapted to start conditional on the heralded modes.

Acknowledgements and contributions

The author's contributions are devising the proof that the GBS pure-state algorithm samples from the correct distribution and performing numerical tests, and the derivation of the formula for the standard boson sampling case. The GBS sequential algorithm was a result of a collaboration. The original version of the algorithm and the foundation for the updated version was proposed by Nicolás Quesada with help from co-authors. The inclusion of the projection onto coherent states was due to Bryn Bell. The idea to apply the GBS sequential algorithm to standard boson sampling was from Jacob Bulmer, who also provided useful insight for the derivation provided.

Chapter 6

Conclusion

"In the end, everything will be okay. If it's not okay, it's not yet the end."
- Fernando Sabino

In this thesis we have explored several methods for simulating GBS in time proportional to the complexity of calculating a single probability for the distribution. This provides a quadratic speed-up on the previous leading algorithm when sampling from both pure and mixed states [79]. The numerical simulations were run assuming the ideal scenario with the application of quantum-inspired algorithms in mind. Experimental imperfections, such as loss, can readily be included [86]. We provide a comparison of the main types of boson sampling in table (6.1) highlighting the difference in complexity for the calculation of individual probabilities and the sampling problem. We have also added threshold GBS which is where the Gaussian state is measured using threshold detectors that only distinguish between whether a photon is present or not, rather than the number of photons. Although this type of boson sampling is not explored in this thesis, the algorithms presented can be extended to this case [86].

All three algorithms presented in this thesis for simulating GBS are viable options being efficient with regard to the number of probability calculations required to output a sample. However, they all suffer from some approximation. For rejection sampling this can be estimated from the sample itself given a large enough sample size, but predicting it beforehand is more uncertain. For MIS, it is not clear how to determine the total variation distance between the sampled and target distributions, but the numerical testing provides some assurance that for the parameters chosen the error will be small. The chain-rule sampling algorithm only suffers from a truncation error which can be reduced by choosing the cut-off photon number in each mode such that the probability of more photons is small. It is possible to tune this probability of getting more photons in any mode to some desired small ϵ by finding the corresponding cut-off photon number. However, the exact relationship between ϵ and the total variation distance is still unknown and so the TVD cannot be fixed or even determined.

The challenge to lower the boundary of quantum advantage by Gaussian boson sampling remains when experimental imperfections may be taken into account. In this case, loss and distinguishabil-

Variant	Matrix function	Complexity of probability	Complexity of sampling
Standard	permanent [6]	$O(N2^N)$	$O(N2^N) + O(MN^2)$ [26]
GBS	hafnian [7]	$O(N^3 2^N) \to O(N^3 2^{N/2})$	$O(MN^3 2^{N/2})$ [84]
GBS (displaced)	loop hafnian [51]	$O(N^3 2^N) \to O(N^3 2^{N/2})$	$O(MN^3 2^{N/2})$ [84]
Threshold GBS	Torontonian [87]	$O(N^3 2^N)$	$O(MN^3 2^{N/2})$ [86]

Table 6.1: A comparison of the complexity of boson sampling variants. For each variant of boson sampling, we list the matrix function that describes the probability distribution, the complexity of calculating a single probability and the complexity of sampling from the distribution. We include standard boson sampling, GBS with and without displacement in the input states and threshold GBS. Although the last variant has not been discussed in this thesis, it is included for completeness. The arrows show the reduction in complexity for a pure state compared to a mixed state.

ity cause approximations to the ideal distribution which means that classical simulations can also make use of these approximations to find a speed-up. Current experimental demonstrations have significant loss [10, 11] which allows the possibility for an approximate algorithm to exploit this. Another open problem is whether an exact chain-rule algorithm can be found similar to standard boson sampling. The error in the algorithm presented is small and so this would only be an incremental improvement. A more pressing open problem is finding a definitive method for validating experimental samples to provide evidence of sampling from the correct distribution.

Bibliography

[1] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11:369–395, 2020.

[2] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 2019.

[3] Pieter Kok, William J Munro, Kae Nemoto, Timothy C Ralph, Jonathan P Dowling, and Gerard J Milburn. Linear optical quantum computing with photonic qubits. *Reviews of modern physics*, 79(1):135, 2007.

[4] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[6] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011.

[7] Craig S Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian boson sampling. *Physical review letters*, 119(17):170501, 2017.

[8] Levon Chakhmakhchyan and Nicolas J Cerf. Boson sampling with gaussian measurements. *Physical Review A*, 96(3):032326, 2017.

[9] Austin P Lund, Anthony Laing, Saleh Rahimi-Keshari, Terry Rudolph, Jeremy L O'Brien, and Timothy C Ralph. Boson sampling from a gaussian state. *Physical review letters*, 113(10):100502, 2014.

[10] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[11] Han-Sen Zhong, Yu-Hao Deng, Jian Qin, Hui Wang, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Dian Wu, Si-Qiu Gong, Hao Su, et al. Phase-programmable gaussian boson sampling using stimulated squeezed light. *arXiv preprint arXiv:2106.15534*, 2021.

[12] Jelmer J Renema, Adrian Menssen, William R Clements, Gil Triginer, William S Kolthammer, and Ian A Walmsley. Efficient classical algorithm for boson sampling with partially distinguishable photons. *Physical review letters*, 120(22):220502, 2018.

[13] Alexandra E Moylett and Peter S Turner. Quantum simulation of partially distinguishable boson sampling. *Physical Review A*, 97(6):062329, 2018.

[14] Raúl García-Patrón, Jelmer J Renema, and Valery Shchesnovich. Simulating boson sampling in lossy architectures. *Quantum*, 3:169, 2019.

[15] Haoyu Qi, Daniel J Brod, Nicolás Quesada, and Raúl García-Patrón. Regimes of classical simulability for noisy gaussian boson sampling. *Physical review letters*, 124(10):100502, 2020.

[16] Junheng Shi and Tim Byrnes. Gaussian boson sampling with partial distinguishability. *arXiv preprint arXiv:2105.09583*, 2021.

[17] Joonsuk Huh, Gian Giacomo Guerreschi, Borja Peropadre, Jarrod R McClean, and Alán Aspuru-Guzik. Boson sampling for molecular vibronic spectra. *Nature Photonics*, 9(9):615, 2015.

[18] Joonsuk Huh and Man-Hong Yung. Vibronic boson sampling: generalized gaussian boson sampling for molecular vibronic spectra at finite temperature. *Scientific reports*, 7(1):1–10, 2017.

[19] Leonardo Banchi, Mark Fingerhuth, Tomas Babej, Christopher Ing, and Juan Miguel Arrazola. Molecular docking with gaussian boson sampling. *Science advances*, 6(23):eaax1950, 2020.

[20] Kamil Brádler, Pierre-Luc Dallaire-Demers, Patrick Rebentrost, Daiqin Su, and Christian Weedbrook. Gaussian boson sampling for perfect matchings of arbitrary graphs. *Physical Review A*, 98(3):032310, 2018.

[21] Kamil Brádler, Shmuel Friedland, Josh Izaac, Nathan Killoran, and Daiqin Su. Graph isomorphism and gaussian boson sampling. *Special Matrices*, 9(1):166–196, 2021.

[22] Maria Schuld, Kamil Brádler, Robert Israel, Daiqin Su, and Brajesh Gupt. A quantum hardware-induced graph kernel based on gaussian boson sampling. *arXiv preprint arXiv:1905.12646*, 2019.

[23] Kamil Bradler, Robert Israel, Maria Schuld, and Daiqin Su. A duality at the heart of gaussian boson sampling. *arXiv preprint arXiv:1910.04022*, 2019.

[24] Soran Jahangiri, Juan Miguel Arrazola, Nicolás Quesada, and Nathan Killoran. Point processes with gaussian boson sampling. *Physical Review E*, 101(2):022134, 2020.

[25] Alex Neville, Chris Sparrow, Raphaël Clifford, Eric Johnston, Patrick M Birchall, Ashley Montanaro, and Anthony Laing. Classical boson sampling algorithms with superior performance to near-term experiments. *Nature Physics*, 13(12):1153–1157, 2017.

[26] Peter Clifford and Raphaël Clifford. The classical complexity of boson sampling. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 146–155. SIAM, 2018.

[27] Jun John Sakurai and Eugene D Commins. Modern quantum mechanics, revised edition, 1995.

[28] Christopher C Gerry and Peter L Knight. *Introductory quantum optics*. Cambridge university press, 2005.

[29] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[30] Samuel L Braunstein and Peter Van Loock. Quantum information with continuous variables. *Reviews of modern physics*, 77(2):513, 2005.

[31] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2):621, 2012.

[32] Eugene P Wigner. On the quantum correction for thermodynamic equilibrium. In *Part I: Physical Chemistry. Part II: Solid State Physics*, pages 110–120. Springer, 1997.

[33] Kôdi Husimi. Some formal properties of the density matrix. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 22(4):264–314, 1940.

[34] Roy J Glauber. Coherent and incoherent states of the radiation field. *Physical Review*, 131(6):2766, 1963.

[35] ECG Sudarshan. Equivalence of semiclassical and quantum mechanical descriptions of statistical light beams. *Physical Review Letters*, 10(7):277, 1963.

[36] R Simon, N Mukunda, and Biswadeb Dutta. Quantum-noise matrix for multimode systems: U (n) invariance, squeezing, and normal forms. *Physical Review A*, 49(3):1567, 1994.

[37] Crispin Gardiner, Peter Zoller, and Peter Zoller. *Quantum noise: a handbook of Markovian and non-Markovian quantum stochastic methods with applications to quantum optics*. Springer Science & Business Media, 2004.

[38] A Holevo. Some statistical problems for quantum gaussian states. *IEEE Transactions on Information Theory*, 21(5):533–543, 1975.

[39] Alexander S Holevo. *Probabilistic and statistical aspects of quantum theory*, volume 1. Springer Science & Business Media, 2011.

[40] NN Bogoljubov, Vladimir Veniaminovic Tolmachov, and DV Širkov. A new method in the theory of superconductivity. *Fortschritte der physik*, 6(11-12):605–682, 1958.

[41] Maurice A De Gosson. *Symplectic geometry and quantum mechanics*, volume 166. Springer Science & Business Media, 2006.

[42] John F Cornwell. *Group theory in physics: An introduction*. Academic press, 1997.

[43] Célia Dantas, Norton G de Almeida, and B Baseia. Statistical properties of the squeezed displaced number states. *Brazilian journal of physics*, 28(4):00–00, 1998.

[44] Herbert John Ryser. *Combinatorial mathematics*, volume 14. American Mathematical Soc., 1963.

[45] Albert Nijenhuis and Herbert S Wilf. *Combinatorial algorithms: for computers and calculators*. Elsevier, 2014.

[46] David G Glynn. The permanent of a square matrix. *European Journal of Combinatorics*, 31(7):1887–1891, 2010.

[47] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[48] Henryk Minc. *Permanents*, volume 6. Cambridge University Press, 1984.

[49] Andreas Björklund. Counting perfect matchings as fast as ryser. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 914–921. SIAM, 2012.

[50] Marek Cygan and Marcin Pilipczuk. Faster exponential-time algorithms in graphs of bounded average degree. *Information and Computation*, 243:75–85, 2015.

[51] Andreas Björklund, Brajesh Gupt, and Nicolás Quesada. A faster hafnian formula for complex matrices and its benchmarking on a supercomputer. *Journal of Experimental Algorithmics (JEA)*, 24:1–17, 2019.

[52] Chong-Ki Hong, Zhe-Yu Ou, and Leonard Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Physical review letters*, 59(18):2044, 1987.

[53] Regina Kruse, Craig S Hamilton, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Detailed study of gaussian boson sampling. *Physical Review A*, 100(3):032326, 2019.

[54] VV Dodonov, OV Man'ko, and VI Man'ko. Multidimensional hermite polynomials and photon distribution for polymode mixed light. *Physical Review A*, 50(1):813, 1994.

[55] Saleh Rahimi-Keshari, Austin P Lund, and Timothy C Ralph. What can quantum optics say about computational complexity theory? *Physical review letters*, 114(6):060501, 2015.

[56] Louis Comtet. *Advanced Combinatorics: The art of finite and infinite expansions*. Springer Science & Business Media, 2012.

[57] Michael Hardy. Combinatorics of partial derivatives. *the electronic journal of combinatorics*, 13(R1):1, 2006.

[58] Taoufik Amri, Julien Laurat, and Claude Fabre. Characterizing quantum properties of a measurement apparatus: Insights from the retrodictive approach. *Physical review letters*, 106(2):020502, 2011.

[59] Levon Chakhmakhchyan and Nicolas J Cerf. Simulating arbitrary gaussian circuits with linear optics. *Physical Review A*, 98(6):062314, 2018.

[60] Jim Pitman. *Probability*. Springer, New York, NY, 1993.

[61] Luc Devroye. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006.

[62] L. Devroye. *Non-Uniform Random Variate Generation*. Springer New York, 2013.

[63] Robert T Moenck. Practical fast polynomial multiplication. In *Proceedings of the third ACM symposium on Symbolic and algebraic computation*, pages 136–148, 1976.

[64] W Morven Gentleman and Gordon Sande. Fast fourier transforms: for fun and profit. In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 563–578, 1966.

[65] John M Pollard. The fast fourier transform in a finite field. *Mathematics of computation*, 25(114):365–374, 1971.

[66] Walter R Gilks and Pascal Wild. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348, 1992.

[67] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[68] Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[69] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[70] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[71] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[72] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[73] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[74] Daniel Hsu, Aryeh Kontorovich, David A Levin, Yuval Peres, and Csaba Szepesvári. Mixing time estimation in reversible markov chains from a single sample path. *arXiv preprint arXiv:1708.07367*, 2017.

[75] Vitaly Kuznetsov and Mehryar Mohri. Generalization bounds for non-stationary mixing processes. *Machine Learning*, 106(1):93–117, 2017.

[76] Yong Liu, Min Xiong, Chunqing Wu, Dongyang Wang, Yingwen Liu, Jiangfang Ding, Anqi Huang, Xiang Fu, Xiaogang Qiang, Ping Xu, et al. Sample caching markov chain monte carlo approach to boson sampling simulation. *New Journal of Physics*, 22(3):033022, 2020.

[77] Jocelyn Quaintance and Henry W Gould. *Combinatorial identities for Stirling numbers: the unpublished notes of HW Gould*. World Scientific, 2015.

[78] Bujiao Wu, Bin Cheng, Fei Jia, Jialin Zhang, Man-Hong Yung, and Xiaoming Sun. Speedup in classical simulation of gaussian boson sampling. *Science Bulletin*, 65(10):832–841, 2020.

[79] Nicolás Quesada and Juan Miguel Arrazola. Exact simulation of gaussian boson sampling in polynomial space and exponential time. *Physical Review Research*, 2(2):023005, 2020.

[80] Yung Liang Tong. *The multivariate normal distribution*. Springer Science & Business Media, 2012.

[81] Haoyu Qi, Diego Cifuentes, Kamil Brádler, Robert Israel, Timjan Kalajdzievski, and Nicolás Quesada. Efficient sampling from shallow gaussian quantum-optical circuits with local interactions. *arXiv preprint arXiv:2009.11824*, 2020.

[82] John Williamson. On the algebraic problem concerning the normal forms of linear dynamical systems. *American journal of mathematics*, 58(1):141–163, 1936.

[83] Alessio Serafini. *Quantum continuous variables: a primer of theoretical methods*. CRC press, 2017.

[84] Nicolás Quesada, Rachel S Chadwick, Bryn A Bell, Juan Miguel Arrazola, Trevor Vincent, Haoyu Qi, Raúl García, et al. Quadratic speed-up for simulating gaussian boson sampling. *PRX Quantum*, 3(1):010306, 2022.

[85] Constantine Pozrikidis. *An introduction to grids, graphs, and networks*. Oxford University Press, 2014.

[86] Jacob FF Bulmer, Bryn A Bell, Rachel S Chadwick, Alex E Jones, Diana Moise, Alessandro Rigazzi, Jan Thorbecke, Utz-Uwe Haus, Thomas Van Vaerenbergh, Raj B Patel, et al. The boundary for quantum advantage in gaussian boson sampling. *Science Advances*, 8(4):eabl9236, 2022.

[87] Nicolás Quesada, Juan Miguel Arrazola, and Nathan Killoran. Gaussian boson sampling using threshold detectors. *Physical Review A*, 98(6):062322, 2018.