

QOLS-Kolthammer-2
MSci Project Report

The Role of Low-order Marginals in Gaussian Boson Sampling

CID: 01491000

Supervisor: Dr. Steve Kolthammer

Assessor: Dr. Kyle Major

May 1, 2023

Word count: 9625, excluding the title page, layperson's summary, contents page and references.

The Photonic Puzzle of a Light-Driven Quantum Computation

—Layperson’s Summary—

Quantum computers have been making headlines with their promise to solve complex problems in various fields, from climate change to medicine. However, the reality is that quantum computing is still in its infancy. A sobering look into this field will reveal (as of yet) limited applications, and few experimental demonstrations. As the scientific community continues to invest resources in this area, several demonstrations of calculations faster than regular, *classical* computers have emerged, pushing the boundaries of modern simulations. At the heart of the majority of these *quantum supremacy* claims lies a primitive model of quantum computing called Gaussian Boson Sampling (GBS). GBS leverages the intricate wave-particle relationship of light to create optical ‘qubits’. These qubits represent information differently than traditional binary one-or-zero bits, as they can exist in a combination of both zero and one. By manipulating light’s characteristics, GBS can generate an output pattern of photons that can be linked to various mathematical problems in chemistry, computer science, and the pharmaceutical industry. This full output is incredibly challenging to simulate using classical computers.

In 2022, Google proposed an algorithm that can approximately replicate the GBS output on powerful classical computers. To understand how it works, imagine the GBS output as a large puzzle forming an image. Google’s algorithm assembles small, correct puzzle pieces and fills in the remaining space with random pieces. While the resulting ‘image’ is surprisingly convincing, its accuracy is an ongoing debate. In this project, we investigated how much information these small puzzle pieces, called ‘low-order marginals’, contain about the overall puzzle. GBS is a physical experiment and so, inevitably, it interacts with the environment around it. This manifests as noise - think of it as a blurring of the puzzle image in this analogy. Our research aimed to explore the connection between noise and the information *not* carried in the low-order marginals. To do so, we coded Google’s algorithm and constructed three distinct noise models which we then used in conjunction with GBS simulations. We also aimed to identify the processes responsible for ‘creating’ the information that is absent in the low-order marginals. To accomplish this, we simulated quantum experiments that enhance a particular process that is not captured by the classical algorithm, namely the high-order correlations of interacting photons.

Our findings provide evidence that certain types of noise in GBS experiments make it easier for classical algorithms to simulate. We also discovered that it was more challenging for the classical algorithm to replicate experiments with enhanced high-order correlations, which implies that this process is not captured in the low-order marginals. Finally, we considered the following mathematical problem: imagine a set of points in which some are connected to each other with links. How can we find the smaller set of points that has the greatest number of links? This is a difficult problem, as it usually involves randomly selecting points until the densest subset is found. As an application of our findings, we applied the classical algorithm to this problem and demonstrated that it accelerates this process, in a manner similar to that of a physical GBS.

Contents

1	Introduction	4
2	Theory	5
2.1	Quantum Theory of Light	5
2.1.1	Light Field Quantisation	5
2.1.2	Phase space formulation and State types	6
2.2	Optical Quantum Computing	8
2.2.1	Overview of Quantum Computing	8
2.2.2	Gaussian Boson Sampling	8
2.2.3	Marginal Distributions of GBS	11
2.2.4	Noise in GBS	11
2.3	Classical Simulation of Gaussian Boson Sampling	12
2.3.1	Greedy Algorithm	12
2.3.2	Time complexity, High-order correlations	13
2.3.3	Boson Samplers with Arbitrary Input States	14
2.4	GBS Applications in Graph Search problems	15
2.4.1	Graphs & the Dense k -Subgraph Problem	15
2.4.2	Enhanced Stochastic Algorithms using GBS	16
3	Method	17
3.1	Bitstrings Statistical Measures	17
3.2	Computational Toolkit	17
3.2.1	Software Package for Photonic Quantum Circuit simulations	17
3.2.2	Greedy Algorithm Implementation tests	19
3.3	Noise Models Implementation	19
3.3.1	Optical Loss	20
3.3.2	Gate Error	20
3.3.3	Distinguishability Effects	21
3.4	High-order Photon Interference with Renema States Implementation	23
3.5	Graph Search Enhancement Implementation	24
4	Results	25
4.1	Marginal-Based Greedy Algorithm and Noise	25
4.2	High-order Photon Interference with Renema states in Boson Samplers	29
4.3	Graph Search Enhancement Results	30
5	Conclusion	31

Abstract - Gaussian Boson Sampling (GBS) is a limited form of a photonic quantum computer that was employed in recent quantum supremacy claims. The GBS output draws from a complex distribution pattern which is exponentially difficult to simulate on classical computers. In this study, we utilise a fully classical GBS algorithm as an instrument to assess the information-carrying capabilities of the low-order marginals of the distribution. We devise and construct three types of noise models onto GBS simulations, which are statistically compared against the classical algorithm output. We provide evidence that neglecting the high-order correlation in GBS is equivalent to the introduction of noise, as well as establish high-order photon interference as a potential source of high-order correlations. An application of this investigation is presented in the form of a classical algorithm inspired by GBS that speeds up the search for the densest subgraph problem.

1 Introduction

Recent developments in the field of quantum computing hold the potential to efficiently solve certain computational tasks that are beyond the capabilities of classical computers. Candidate tasks for near-term devices include simulated quantum many-body dynamics [1], quantum machine learning on hybrid devices [2] and quantum optimisation algorithms [3, 4]. At present, emerging quantum devices have limited resource capabilities and are prone to logical errors arising due to quantum decoherence in quantum computations. Nevertheless, there have been three instances of beyond-classical experimental demonstrations [5, 6, 7], referred to as ‘quantum advantage’. Two of these instances [6, 7] employed Gaussian Boson Sampling (GBS), a limited model of photonic quantum computation relying on passing Gaussian states through a linear interferometer and measuring their output configurations [8]. The single-photon detection pattern at the outputs samples from a probability distribution that is exponentially hard to simulate on classical computers. This has motivated the limited application of GBS primarily for algorithm speedup in the context of graph search [9] and quantum chemistry [10]. As the research endeavours on the development of error-correcting photonic quantum computers grow, understanding GBS’s information-bearing capacity and its link to computational complexity becomes vital.

Recently, Google Quantum AI [11] proposed a state-of-the-art classical algorithm to efficiently simulate the GBS model. This algorithm leverages the probability distribution of small subsets of GBS output modes, known as *low-order marginals*, to approximate the full output distribution. The local, low-order marginal distributions can be calculated classically with much greater efficiency compared to the full output distribution, which is exponentially expensive in the number of photon detections. Although Google’s research claimed to have successfully reproduced the first GBS quantum supremacy experiment using classical means, a subsequent paper by Xanadu Quantum Technologies presented a new quantum supremacy claim based on GBS that outperformed the classical algorithm [7]. This continuous discussion inspired the central questions of this project: (i) How much information about the full GBS distribution is encapsulated in the low-order marginals, and (ii) what, if any, is the association between GBS noise and the exclusion of high-order marginals, as experimental noise might be a contributing factor to the classical algorithm’s ability to replicate the GBS experiment [12, 13]. As such, we employ Google’s classical algorithm as a tool to address these questions, and in the process uncover the connection between physical phenomena in experiments and the source of complexity in GBS.

In this study, we utilised Google’s proposal to develop and evaluate the performance of the marginal-based classical algorithm. We then designed three distinct noise models to simulate GBS experiments under increasing levels of noise. The output distribution of the classical algorithm was numerically compared to both noise-induced GBS and an expanded group of boson

sampling circuits with heightened photon interference. As an application of our research, we also introduce a GBS-inspired classical algorithm that improves the performance of a particular graph search problem.

This report is organised in the following manner: Chapter 2 offers the necessary theoretical background for this project. Sections 2.1 and 2.2 cover fundamental aspects of quantum optics, while sections 2.3 and 2.4 delve into the inner workings of the classical algorithm and graph theory, respectively. In Chapter 3, the computational and statistical methods employed are presented in sections 3.1 and 3.2, the noise model implementation in section 3.3, the approach to increasing high-order photon interference in 3.4, and graph search applications in section 3.5. Chapter 4 outlines our results, initially for the performance of the classical algorithm against noise-injected GBS in section 4.1, then for the interference-enhanced circuits in section 4.2, and finally for our application to graph search in section 4.3.

2 Theory

2.1 Quantum Theory of Light

In this section, we present a concise overview of the quantum mechanical description of light, including important concepts such as modes of light, Gaussian states, and squeezed states. These form the necessary background for our investigation.

2.1.1 Light Field Quantisation

Starting from classical physics, we consider a radiation field confined to a one-dimensional cavity along the z -axis with boundaries at $z = 0$ and $z = L$. The single-mode fields that satisfy Maxwell's equations are given by

$$E_x(z, t) = \left(\frac{2\omega^2}{V\epsilon_0} \right)^{1/2} q(t) \sin(kz) \quad (1)$$

$$B_y(z, t) = \left(\frac{\epsilon_0\mu_0}{k} \right) \left(\frac{2\omega^2}{V\epsilon_0} \right)^{1/2} \dot{q}(t) \cos(kz) \quad (2)$$

where ω is the frequency of the mode, ϵ_0 and μ_0 is the permittivity and permeability of free space respectively, c is the speed of light, t is time, $k = \omega/c$ is the wave number, V is the effective volume of the cavity, $q(t)$ is a time-depended factor having with units of length and $\dot{q} = \frac{dq}{dt}$ [14]. It can then be showed that the classical field energy, or Hamiltonian H , of the single-mode field is given by

$$H = \frac{1}{2} (p^2 + \omega^2 q^2) \quad (3)$$

with $p = \dot{q}$. It is then apparent that the single-mode field is formally equivalent to a harmonic oscillator of unit mass, with q and p playing the roles of canonical position and momentum. Having identified the canonical variables for the classical system, we simply use the correspondence rule to swap them by their operator counterparts \hat{q} and \hat{p} , giving the Hamiltonian

$$\hat{H} = \frac{1}{2} (\hat{p}^2 + \omega^2 \hat{q}^2). \quad (4)$$

These are conjugate pairs, satisfying $[\hat{q}, \hat{p}] = i\hbar$, as well as Heisenberg's Uncertainty principle [14],

$$\Delta q \Delta p \geq \frac{\hbar}{2}. \quad (5)$$

Hence, the electromagnetic (EM) field of light in free space can be decomposed into a series of orthogonal modes, each analogous to an independent harmonic oscillator with operators \hat{q} and \hat{p} referred to as the quadrature operators. In this interpretation, a photon is an elementary excitation of a normal mode of the field [14]. It is convenient to introduce the non-Hermitian (and therefore non-observable) annihilation \hat{a} and creation \hat{a}^\dagger operators which act to lower or raise the number of excitation of the field. By letting $|n\rangle$ denote an energy eigenstate of the single mode field, we can write

$$\hat{a}|n\rangle = \sqrt{n}|n-1\rangle \quad (6)$$

$$\hat{a}^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle \quad (7)$$

providing a basis for the Hilbert space. The quadrature operators have the dimensionless forms \hat{Q} and \hat{P} , which are related to the annihilation and creation operators as

$$\hat{Q} = \frac{1}{2}(\hat{a} + \hat{a}^\dagger) \quad (8)$$

$$\hat{P} = \frac{1}{2i}(\hat{a} - \hat{a}^\dagger). \quad (9)$$

2.1.2 Phase space formulation and State types

Quantum mechanical states can be described in terms of state vectors, wavefunctions and density matrices. There are, however, other descriptions that cover all quantum mechanical phenomena. Directly following from the canonical formulation of the EM field with the quadrature operators \hat{q}, \hat{p} , the predominant representation in quantum optics is the Wigner function representation in phase-space. A system described by the density operator $\hat{\rho}$ has a Wigner function given by

$$W(q, p) = \int_{-\infty}^{\infty} da \langle q + a/2 | \hat{\rho} | q - a/2 \rangle e^{-ipa/\hbar} \quad (10)$$

where q and p are the canonical position and momentum quadratures defining phase space. Wigner functions are real-valued probability distributions, as they quantify the probability of finding the system in any given phase space volume V by integrating over that volume. They can, however, have negative values, which differentiates them from classical probability distributions.

Using the Wigner function representation, we are now able to discuss several types of quantum states that are significant for this investigation. First, *vacuum* states are the lowest energy harmonic oscillator states and are denoted by the density operator $\rho_{vacuum} = |0\rangle\langle 0|$. Fig.1 displays their Wigner function representation. Vacuum states are coherent states, meaning that the Heisenberg uncertainty relation of Eq.(5) between the canonical conjugate variables is minimised by distributing the uncertainty equally between the two.

Although it is unattainable to precisely determine both position and momentum simultaneously, it is feasible to decrease the uncertainty of one quantum variable while consequently increasing the uncertainty of its quantum conjugate. This concept is realised in the states known as *squeezed* states. The squeezing operation can be applied through squeezing gates, with their operator following $\rho_{squeezed} = \hat{S}(\chi)|0\rangle\langle 0|\hat{S}^\dagger(\chi)$. The Wigner function of a squeezed state is depicted in Fig.2.

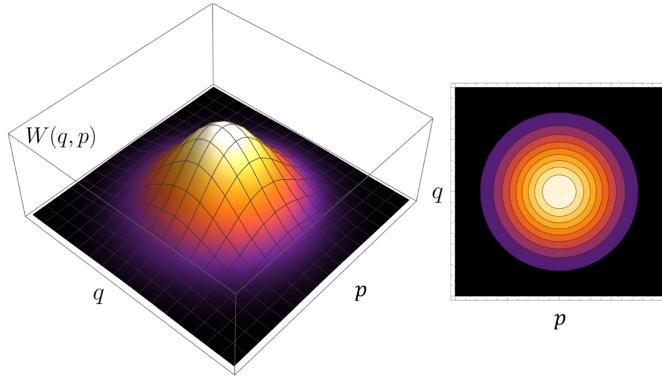


Figure 1: Wigner function of the vacuum state. The plot on the left is the quasi-probability surface $W(q, p)$. The plot on the right is the contour plot of this surface. Figure adapted from Ref.[15]

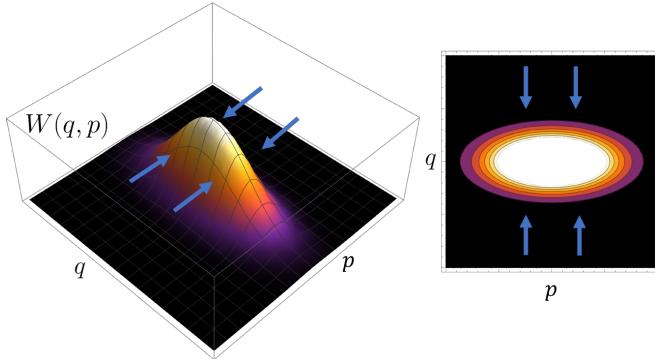


Figure 2: Wigner function of a squeezed vacuum state with $\rho = \hat{S}(\chi)|0\rangle\langle 0|\hat{S}^\dagger(\chi)$. The plot on the left is the quasi-probability surface $W(q, p)$. The uncertainty in position x is reduced while the uncertainty in p is increased. Figure adapted from Ref.[15]

Both vacuum and squeezed states belong in the family of Gaussian states. These states follow Gaussian statistics, and a comprehensive description of them can be achieved by considering their covariance matrix σ [14]. This matrix is built from the second moments associated with position and momentum quadratures. When displacement is not present, the covariance matrix can be expressed as follows [15]:

$$\sigma = \langle \{u_k, u_l\} \rangle - \langle u_k \rangle \langle u_l \rangle \quad (11)$$

where $\mathbf{u} = (\hat{q}_1, \dots, \hat{q}_n, \hat{p}_1, \dots, \hat{p}_n)^T$ is *mode vector*. Following this convention, the single-mode covariance matrix can be shown to follow

$$\sigma = \begin{pmatrix} \Delta\hat{q}^2 & \Delta\hat{q}\Delta\hat{p} \\ \Delta\hat{p}\Delta\hat{q} & \Delta\hat{p}^2 \end{pmatrix}. \quad (12)$$

Finally, complementary to the continuous Gaussian states are the discrete Fock states (or number states). These are eigenstates of the EM field meaning that, for example, the Fock state $|2\rangle_{Fock}$ corresponds to exactly 2 photons. Each of the Gaussian states considered before can be expanded in this discrete Fock basis.

2.2 Optical Quantum Computing

This section presents an overview of how light radiation fields can be exploited to encode information and perform computations. We then explain the primitive quantum computing setup of Gaussian Boson Sampling, along with the underlying equations governing its full and subset output distributions. Then follows a section on the main source of noise in optical systems. These form the basis for our algorithms and simulations.

2.2.1 Overview of Quantum Computing

Quantum computations involve computational processes that harness the properties of quantum systems. These physical features, known as observables, are denoted by their operators, and their continuity properties are encoded within the eigenvalue value spectrum of said operator. Discrete-variable encoding relies on two orthogonal states representing either 0 or 1 to relay information - known as *qubits*. The basis representation of a qubit is

$$\text{Qubit basis } |\Phi\rangle = c_0 |0\rangle + c_1 |1\rangle$$

where $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and c_0, c_1 are constants for amplitudes. A natural way to manipulate observables with discrete spectra is finite-dimensional linear algebra, with states represented by vectors and operations by matrices. While this is also the case for classical computations, the main difference lies in the space in which these states exist. Quantum operations take place in Hilbert space, which allows for superposition and entanglement. In this framework, problem-solving calculations are performed by exploiting the quantum correlations inherent in systems representing the problem. This contrasts with the truly discrete nature of classical binary ‘bit-space’, where each basic unit, the *bit*, is in a definite state of 0 or 1.

Many physical systems are intrinsically continuous, offering a paradigm in which the full infinite-dimensional Hilbert space can be used to encode information, referred to as continuous-variable (CV) encoding. The prototypical medium to explore this concept is the radiation field of light in free space, through its decomposition into individual harmonic oscillator states (as outlined in section 2.1.1). As such, the most elementary CV system is the bosonic harmonic oscillator, with the modes of the EM field acting as the data registers [16], represented as ‘wires’ in quantum circuits.

Operations on CV-encoded qubits can be performed through quantum logic gates, the analogue to classical logic gates. Quantum gates are unitary operators and are described as unitary matrices relative to some basis. A combination of the logical CV encoding and a set of logic gates are the building blocks for performing any mathematical operation, this set is referred to as universal quantum computation [17]. Despite the inevitable noise that plagues quantum operations, a physical error rate below a certain threshold can be suppressed to arbitrarily low levels through the aid of quantum error correction schemes [18], known as ‘fault-tolerance’. This project focuses on a non-universal, non-fault-tolerant quantum photonic model called Gaussian Boson Sampling, which is able to perform a specific set of mathematical operations. The model is outlined in the following section.

2.2.2 Gaussian Boson Sampling

Figure 3 depicts a GBS multimode linear-optical circuit. It involves sending separate Gaussian state inputs into a linear interferometer, characterised by a matrix \mathbf{U} , which transforms m input vacuum states $|0\rangle$ into m output modes. Each output mode is then measured by photon number resolving (PNR) detectors which count the number of photons in the measured mode.

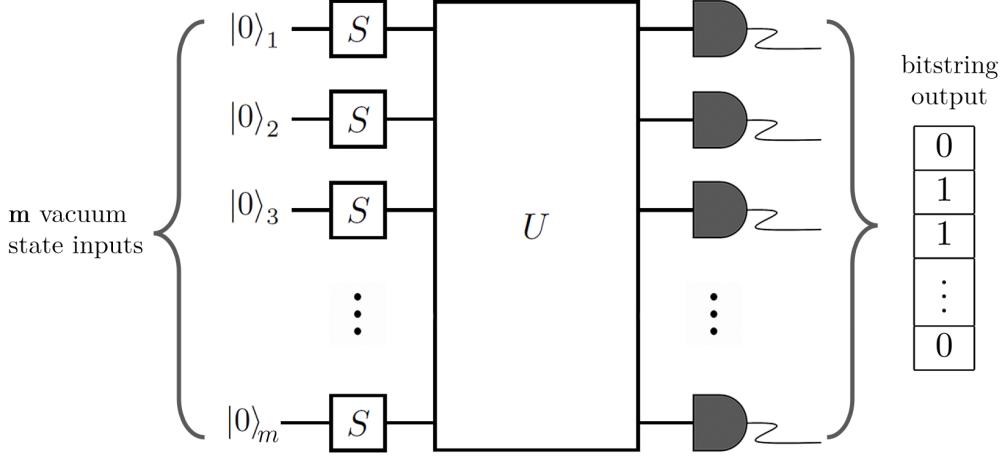


Figure 3: Gaussian Boson Sampling quantum circuit. An m -mode Gaussian state is passed into a linear interferometer described by the unitary operator \mathbf{U} . PNR or threshold detectors then follow at each output. In the case of threshold detectors, the output is in the form of a bit-string, with a '1' at modes where a photon is detected and '0' at modes where no photon is detected.

The Gaussian nature of the input is retained at the output, as the interferometer only consists of linear optical elements and loss, all of which are Gaussian transformations [19]. In the absence of displacement in the input states, and the probability of measuring a sample output pattern $\mathbf{n} = (n_1, n_2, \dots, n_m)$, where n_i is the number of photons detected in mode i , is given by

$$p(\mathbf{n}) = \frac{p_0}{\prod_i n_i!} \text{Haf}(\mathbf{A}_n) \quad (13)$$

where p_0 is the probability of not detecting any photons in any output (all-zero bit-string), Haf is the hafnian function, and matrix \mathbf{A} is related to the covariance matrix $\boldsymbol{\sigma}_{GBS}$ according to the expression

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix} \left(\mathbf{I} - \left(\boldsymbol{\sigma} + \frac{\mathbf{I}}{2} \right)^{-1} \right). \quad (14)$$

where (I) is the unit matrix. The reduced matrix \mathbf{A}_n is derived by selecting n_i times the i -th and $(i+m)$ -th columns and rows of matrix \mathbf{A} . The covariance matrix $\boldsymbol{\sigma}_{GBS}$ is obtained from the expression

$$\boldsymbol{\sigma}_{GBS} = \mathbb{I} - \frac{1}{2} \begin{pmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{U}^* \end{pmatrix} \begin{pmatrix} \mathbf{U}^\dagger & 0 \\ 0 & \mathbf{U}^T \end{pmatrix} + \begin{pmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{U}^* \end{pmatrix} \boldsymbol{\sigma}_{in} \begin{pmatrix} \mathbf{U}^\dagger & 0 \\ 0 & \mathbf{U}^T \end{pmatrix} \quad (15)$$

where \mathbf{U} is the unitary matrix describing the GBS interferometer and $\boldsymbol{\sigma}_{in}$ is the covariance matrix describing the input state to the interferometer. To obtain $\boldsymbol{\sigma}_{in}$, which is the tensor product of m two-mode squeezed vacua states, we calculate

$$\boldsymbol{\sigma}_{in} = \mathbf{S} \boldsymbol{\sigma}_{vac} \mathbf{S}_{TM}^\dagger \quad (16)$$

where $\sigma_{vac} = \frac{\mathbb{I}}{2}$ and the squeezing matrix \mathbf{S} is defined as

$$\mathbf{S} = \left(\begin{array}{ccc|ccc} \text{Ch}(r_1) & 0 & \dots & \text{Sh}(r_1) & 0 & \dots \\ 0 & \text{Ch}(r_2) & \dots & 0 & \text{Sh}(r_2) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \hline \text{Sh}(r_1) & 0 & \dots & \text{Ch}(r_1) & 0 & \dots \\ 0 & \text{Sh}(r_2) & \dots & 0 & \text{Ch}(r_2) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{array} \right) \quad (17)$$

with

$$\begin{aligned} \text{Ch}(r_k) &= \begin{pmatrix} \cosh(r_k) & 0 \\ 0 & \cosh(r_k) \end{pmatrix} \\ \text{Sh}(r_k) &= \begin{pmatrix} \sinh(r_k) & 0 \\ 0 & \sinh(r_k) \end{pmatrix} \end{aligned} \quad (18)$$

where $k = 1, 2, 3, \dots, m$ and denotes the specific GBS squeezing parameters.

In the case where a pure Gaussian state at the output has zero displacement, \mathbf{A} is further reduced to $\mathbf{A} = \mathbf{B} \oplus \mathbf{B}^*$, where \mathbf{B} is a $m \times m$ symmetric matrix. The probability of measuring an output sample \mathbf{n} then becomes

$$P(\mathbf{n}) = \frac{p_0}{\prod_i n_i!} |\text{Haf}(\mathbf{B}_n)|^2 \quad (19)$$

where the submatrix \mathbf{B}_n is derived by selecting n_i times the i -th row and column. However, the sampling distribution of Eq.(19) only occurs in the ideal lossless case, otherwise distribution of Eq.(13) is obtained. This is important as, in order to establish a correspondence to the symmetric adjacency matrix of an undirected graph (explained later in section 2.4.2), a symmetric matrix is needed - which is only the case for \mathbf{B} .

The hafnian of a matrix is defined as

$$\text{Haf}(\mathbf{A}) = \sum_{M \in \text{PMP}(M)} \prod_{(i,j) \in M} A_{i,j} \quad (20)$$

where $\text{PMP}(m)$ stands for the set of perfect matching of m objects. The concept of perfect matching is outlined further in section 2.4.1, as well as how they are related to perfect marching in graphs. It has been shown that calculating the Hafnian is exponentially expensive - classified as $\#P-hard$ - in the number of clicks (number of photon detections). Nevertheless, calculating probabilities marginalised over a small subset of modes is efficient as explained later.

If the PNR detectors in the GBS setup are replaced by the threshold detectors, then at each mode it will only be possible to differentiate between a vacuum state (no photons detected) or a higher Fock state (photons detected) - yielding a 0 or 1 respectively. Hence, The output of the GBS experiment with threshold detectors over m modes is a bit-string \mathbf{z} of length m . For a covariance matrix σ_{GBS} , given an output bit-string \mathbf{z} with threshold detectors clicking on modes in the set \mathcal{S} , the bit-string probability is given by

$$p(\mathbf{z}) = \frac{\text{Tor}(\mathbf{O}_{\mathcal{S}})}{\sqrt{\det(\sigma_{GBS})}}, \quad (21)$$

where $\mathbf{O}_{\mathcal{S}} = \mathbb{I} - (\sigma_{GBS}^{-1})_{\mathcal{S}}$ (reduced in the same way as in Eq.(14) and Tor is the Torontonian function defined as

$$\text{Tor}(A) = \sum_{Z \in P([\mathcal{S}])} \frac{(-1)^{|Z|}}{\sqrt{\det(\mathbb{I} - A_Z)}} \quad (22)$$

where $P([\mathcal{S}])$ is the power set of all $2^{\mathcal{S}}$ subsets of $[\mathcal{S}] = \{1, 2, \dots, \mathcal{S}\}$ and \mathcal{S} is the number of modes where a photon is detected. The Torontonian has been one of the main underlying machinery in this project, as we only considered GBS experiments with threshold detectors. Both Torontonian and the Hafnian belong to the same complexity class, and have been shown to be mathematically related [20]. Hence, the GBS setup is a non-universal optical computing model, which is able to efficiently calculate the Hafnians and Torontonians of matrices. While this constitutes a limited range of applications, it has been proposed to enhance the speed of these calculations for large matrices exponentially, a feat largely thought to be impossible for current classical computers.

2.2.3 Marginal Distributions of GBS

The *marginal* of a total distribution of random variables is the probability distribution of a subset of these variables [21]. In this project, the marginals of GBS refer to the output distribution of a subset of output modes, which are efficient to calculate classically. Despite the fact that calculating the full distribution is exponentially hard, the ideal two-mode marginals (that is the probability distribution over just 2 output modes) and three-mode marginals are efficient to compute. These marginals can be calculated using Eq.(15) and Eq.(21). However, rather than beginning with the complete covariance matrix σ_{GBS} , we utilise the reduced covariance matrix σ_R . This matrix can be acquired by selecting the i -th and $i + m$ -th rows and columns of the σ_{GBS} matrix, where i covers the mode indices of the desired marginal distribution.

2.2.4 Noise in GBS

All optical systems suffer from experimental imperfections, as opposed to the idealised theoretical models. The robustness of bosonic models is hindered by the effects of photon loss, fabrication imperfections on the linear-optical transformation gates, and the effects of partial photon distinguishability.

Optical Loss: One of the main sources of noise in linear optical systems is photon loss. While in ref.[22] it was concluded that GBS retains its complexity and hardness when a constant number of photons is lost, this falls short of the realistic regime where a *fraction* of photons is expected to be lost. Moreover, most current implementations are likely to suffer from exponential losses [23]. This can be concluded from the following argument: Each time a photon passes through a beamsplitter, it has a finite probability of being lost to the environment $p_{loss} = (1 - \eta)$. Hence, after passing through n beamsplitters in a bosonic network, the probability that it arrives at the output should decay as η^n .

Quantum Gate error: For a quantum gate described by its unitary operator \mathbf{U} , ‘quantum gate error’ describes the scenario in which the actually induced transformation deviates from $|\psi\rangle \mapsto \mathbf{U}|\psi\rangle$. For coherent gate errors, the transformation preserves the purity of the input state, but the unitary operation is perturbed to $|\psi\rangle \mapsto \tilde{\mathbf{U}}|\psi\rangle$, where $\tilde{\mathbf{U}} \neq \mathbf{U}$. These are prevalent in quantum systems due to imperfect classical control and parameter drift during the execution of quantum algorithms. Fluctuating environmental conditions can often lead to a slow drift of system parameters over the course of many experimental runs. Some of the most prevalent sources of gate error include drifts of the control laser frequency and amplitude, magnetic field drifts in the laboratory, and off-resonance scattering [24].

Photon Distinguishability effects: Non-classical photon interference is a fundamental feature of quantum linear circuits that must be retained for quantum supremacy and quantum calculations. The famous Hong-Ou-Mandel experiment [25] provided the first important example of non-classical two-photon interference. The degree of bunching behaviour exhibited at the output modes of a beamsplitter impinged by two independent photons depends on how similar the two photons are in all degrees of freedom such as time, frequency and polarisa-

tion. Distinguishable photons under a beamsplitter show no such correlation effects. Thus, it is important to consider the effects of partial distinguishability, especially in frequency, as experimental methods for producing a pair of entangled photons never produce two perfectly indistinguishable photons [26]. For GBS, partial distinguishability originates from imperfect input light that has minor shifts in time or frequency [27].

2.3 Classical Simulation of Gaussian Boson Sampling

In this section, we outline the recent, state-of-the-art algorithm for classically reproducing the bit-string outcomes of GBS using the classically-tractable marginal distributions of the experiment. We then delve into the topic of computational complexity in GBS, as well as the notion of high-order correlations.

2.3.1 Greedy Algorithm

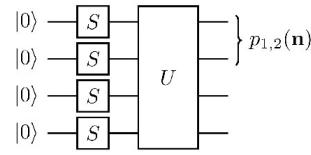
Despite the fact the full output distribution of GBS is exceedingly costly to simulate classically, it has been shown that the low-order marginals (the subsets of the distribution involving just two or three modes) can be efficiently calculated on a classical computer [11]. Furthermore, the marginal distributions of GBS are non-uniform, implying that they carry information about the whole distribution (as otherwise they would follow a uniform random distribution). Based on this premise, Google Quantum AI has recently proposed [11] a greedy, heuristic algorithm that generates bitstrings by approximating the output distribution of a GBS experiment with threshold detectors. It utilises the marginals of the distribution to construct an approximate GBS output. It makes the locally optimal choice at each stage (hence referred to as *greedy*), while the result is not guaranteed to be optimal but sufficient for a satisfactory approximation (hence referred to as *heuristic*). Moving forward, the approximate distribution generated by the greedy algorithm will be referred to as the ‘Greedy distribution’.

The goal of the algorithm targeting an m -mode GBS experiment is to generate a set of L , m -length bitstrings with approximately correct marginal probability distributions. First, we choose the order k of the algorithm approximation - this is essentially the size of the subset of modes from which the ideal marginals will be calculated. Higher k means a higher size subset of marginal probabilities (a subset of wires) will be computed, increasing the accuracy of the simulation at the expense of increased computational time. The number of required marginals is given by $\binom{m}{k}$ and is of order $\mathcal{O}(m^k)$. We then generate an empty matrix \mathbf{J} of size $L \times m$. We want to fill this matrix with all entries being a 0 or 1, thus each row corresponding to an m -length bit-string.

First, consider the first row and first k columns. We compute the ideal marginal probabilities of the first k modes using the method outlined in section 2.2.3. We then sample a k -length bit-string from ideal marginal probabilities. We fill the next rows with new samples and finally shuffle all rows randomly. Moving to the $(k + 1)$ th column, we compute the ideal marginal probabilities for output modes $1 : k + 1$. For each row, we sample an ideal bit, given the values of the previous $(k + 1)$ values. When we reach the final row, we shuffle all rows randomly. We continue iteratively until the matrix is filled. An overview of the algorithm for two-mode marginals ($k = 2$) is outlined in the diagram of figure 4.

1. Marginal distribution of modes 1,2:

$p_{1,2}(00)$	$p_{1,2}(01)$	$p_{1,2}(10)$	$p_{1,2}(11)$
0.21	0.32	0.38	0.09

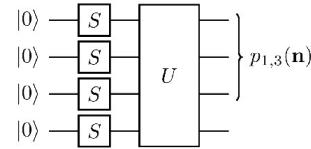


1.1 Select submatrix. **1.2** Add optimal bits. **1.3** Fill all rows. **1.4** Shuffle all rows.

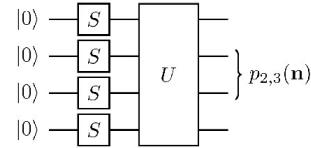
$$\left(\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 1 & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 1 & 0 & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 0 & 1 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \\ 1 & 0 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{array} \right)$$

2. Marginal distributions of modes 1,3 and modes 2,3:

$p_{1,3}(00)$	$p_{1,3}(01)$	$p_{1,3}(10)$	$p_{1,3}(11)$
0.24	0.26	0.31	0.19



$p_{2,3}(00)$	$p_{2,3}(01)$	$p_{2,3}(10)$	$p_{2,3}(11)$
0.25	0.25	0.25	0.25



2.1 Select submatrix. **2.2** Add optimal bit. **2.3** Fill all rows. **2.4** Shuffle all rows.

$$\left(\begin{array}{cccc} 0 & 1 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \\ 1 & 0 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 0 & 1 & 0 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot \\ 1 & 0 & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 0 & 1 & 0 & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot \\ 1 & 0 & 0 & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot \end{array} \right) \quad \left(\begin{array}{ccccc} 0 & 0 & 1 & \cdot & \cdot \\ 0 & 1 & 0 & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot \\ 1 & 0 & 0 & \cdot & \cdot \end{array} \right)$$

3. Repeat until all the columns are filled.

Figure 4: Outline of the greedy algorithm for two-mode marginal ($k = 2$) distributions as inputs. Here, by optimal bits we mean the bits that minimise the total variation distance (see section 3.1 for more details) between the ideal marginal and the appended bits. In steps 2.1-2.3, the red columns indicate that both the two marginal distributions for modes 1,3 and modes 2,3 need to be considered when generating the third column. Here, by $p_{i,j}(00)$ we refer to the (marginal) probability of sampling a ‘0,0’ bit-string from modes i and j .

2.3.2 Time complexity, High-order correlations

Time complexity is defined as the amount of time taken by an algorithm to run, as a function of the length of the input [28]. It was shown in ref.[11] that the time complexity for the greedy algorithm is $\mathcal{O}(m^k 2^k L)$. The first term, m^k , arises from the individual marginal distributions that need to be calculated to fill each column of the matrix \mathbf{J} . The 2^k term originates from the individual bit-strings that are considered at each step before appending the optimal bit. Finally, the L term is the repetitions for each row of the matrix (hence is equal to the number of rows). This assumes that the ideal marginal distributions calculations are efficient enough to be ignored in the time complexity - which is a valid assumption only for low orders. It is important to note the notion of quantum supremacy is said to be achieved when the time complexity of a quantum algorithm is lower than its classical counterpart [5].

One of the driving intuitions behind this investigation is that, in general, noisy GBS exper-

iments are easier to simulate classically than their ideal counterparts [12]. In this study, the greedy algorithm serves as a tool to examine the extent of information carried by low-order marginals, as it relies solely on these efficient calculations to approximate the output. When benchmarked against noisy GBS simulations, it can also establish a connection between noise and the omission of higher-order marginals of the full distribution.

A k^{th} order approximation Greedy algorithm does not consider the marginal distributions of more than k modes at a time. Moreover, it has been shown in Ref.[11] that k^{th} order marginals (marginal distributions of subsets of length k) of GBS contain information pertaining to the interference processes involving k or fewer photons. By this omission, the final probability distribution lacks what is referred to as *high-order correlations*. A k^{th} order correlation of a probability distribution is also known as the k^{th} order Ursell function - these are quantitative measures of the shape of the distribution similar to the modes of the distribution. For instance, the first two order Ursell functions are the mean and the variance. One of the aims of this project is to assess whether noise affects the high-order correlations of the final distribution; if that is the case, then the Greedy distribution would perform better under these noisy conditions. A formal mathematical definition of Ursell functions in GBS is outlined in Appendix D of ref.[11].

2.3.3 Boson Samplers with Arbitrary Input States

As discussed in Ref.[11], k^{th} order marginals (marginal distributions of subsets of length k) of GBS contain information about the interference process of k or fewer photons. It then follows that the marginal distributions of GBS are non-uniform (non-random), as they contain information about the full distribution. This finding holds true not only for GBS but also for boson samplers with arbitrary input states. Furthermore, as proposed in Ref.[29], the usage of non-Gaussian input states for boson samplers makes them harder to reproduce with classical marginal-based algorithms, such as the greedy algorithm. We refer to these input states as Renema states, after the author who introduced them. Renema states are product states of vacuum and single-photon states taking the following form for an even number of modes m :

$$|\Psi^{\text{Renema}}\rangle = \frac{1}{2}(|\psi\rangle_1 + |\psi\rangle_2) \quad (23)$$

where $|\psi\rangle_1$ has single-photon states $|1\rangle$ in modes $1, \dots, m/2$ and vacuum states $|0\rangle$ in modes $(m/2 + 1), \dots, m$, with $|\psi\rangle_2$ having the same arrangement with vacuum and single-photon states swapped. The Renema state for $m = 6$ is

$$|\Psi\rangle = \frac{1}{2}(|111000\rangle + |000111\rangle). \quad (24)$$

Although an analysis of boson samplers with Renema state inputs deviates from the initial investigation into GBS, it remains relevant as it delves into the physical processes that render boson sampling with arbitrary input states computationally hard for classical simulations. In Renema states, their construction enhances the $\frac{m}{2}$ th order photon interference exponentially [13]. Consequently, classical algorithms relying on marginal distributions of order $k < \frac{m}{2}$ should be unable to fully reproduce the information contained within the complete distribution, leading to a considerable reduction in their performance.

2.4 GBS Applications in Graph Search problems

2.4.1 Graphs & the Dense k -Subgraph Problem

Graph theory aims to describe complex systems through the bilateral relationships between the systems' components [30]. It consists of a set of objects, referred to as *vertices* or *nodes*, which are connected in pairs by *edges*, acting as links. Graph theory is particularly useful in clustering problems spanning a plethora of fields, including bioinformatics [31], migration pattern detection in financial markets [32], and data mining [33].

Mathematically, a graph \mathcal{G} consists of a set of vertices \mathcal{V} and a set of edges \mathcal{E} which are pairs of vertices taken from \mathcal{V} . The edge between vertices i and j is denoted as (i, j) . If a vertex connects to itself, it is a *self-loop*. The *degree* of a vertex is the number of edges attached to that vertex. Graphs can be *directed* if edges are allowed to have a direction, or *undirected* if all edges connect symmetrically. In *weighted* graphs, each edge has a positive integer value associated with it, called the *edge weight*. Graphs can be represented in a multitude of ways, but a particularly insightful representation for this project is using the *Adjacency Matrix*. An adjacency matrix Δ captures all the information in a graph and is defined such that Δ_{ij} is the weight of the edge running from vertex j to vertex i . Naturally, the dimension of Δ will be equal to the number of vertices in the graph that it represents. In the case where the graph is undirected, the Δ matrix is symmetric. A widely used model for generating random graphs is the Erdős–Rényi model, in which all possible pairs of vertices have a fixed probability $p_{\text{E.R}}$ of forming an edge.

A *subgraph* $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is made of a subset of the vertex and edge sets of the original graph \mathcal{G} . Critically, all the edges of the subgraph must be made from the vertices in the subgraph, meaning that if $(i, j) \in \mathcal{E}'$ then $i, j \in \mathcal{V}'$ [34]. An example graph and subgraphs are shown in figure 5.

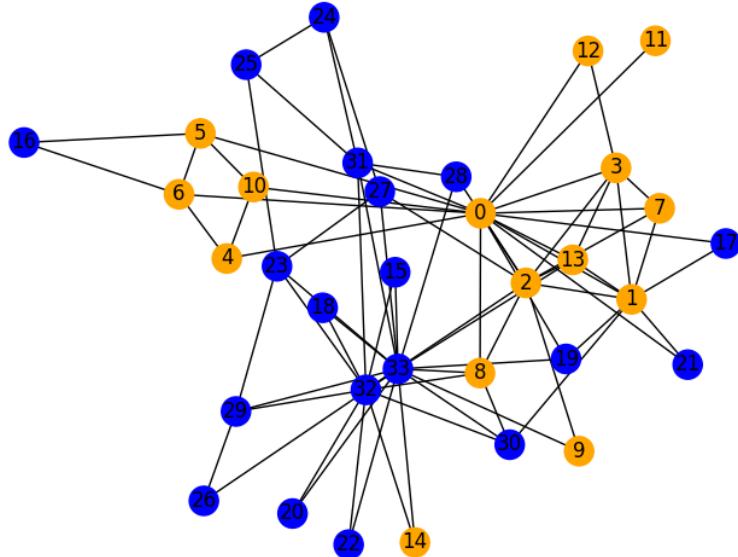


Figure 5: Example of undirected, unweighted graph with number of vertices $|\mathcal{V}| = 33$ generated on the Python package *networkx*. No self-loops were allowed. When considering vertices of a particular colour only, it would constitute a subgraph of the full graph.

A *matching* is a subset of edges in which no two edges share a common vertex. A *perfect matching* is an independent edge set in which every vertex is incident to exactly one edge of the matching [35]. This is illustrated intuitively in figure 6 which shows three perfect matchings in a complete graph with 4 vertices.

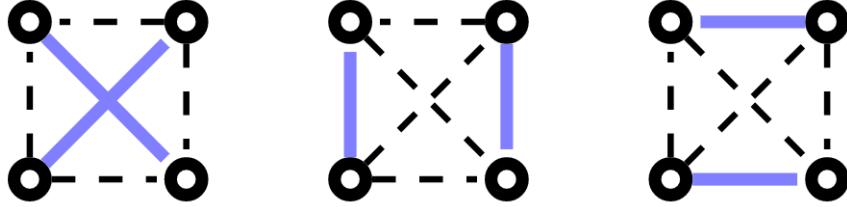


Figure 6: Perfect matchings in a fully connected graph of 4 vertices, indicated by the purple lines. Adapted from Ref.[36].

In this study, we applied the Greedy algorithm to speed-up classical stochastic algorithms for the densest k -subgraph (DkS) problem. DkS consists of the following algorithm-based optimisation task: given a graph \mathcal{G} with $|\mathcal{V}|$ vertices and a parameter k , find the subgraph with $k < |\mathcal{V}|$ vertices with the highest density [37]. The density of a weighted graph can be defined as

$$d = 2 \frac{\sum_{i,j} \Delta_{ij}}{|\mathcal{V}|(|\mathcal{V}| - 1)} \quad (25)$$

summing over all the edges of the graph and where $|\mathcal{V}|$ is the number of vertices in the graph [38]. The DkS problem is NP-Hard, which means that it is currently suspected that there are no algorithms that can solve this problem in polynomial-time or less. Here, polynomial-time refers to the case where the execution time of a computation is no more than a polynomial function of the size of the input; in this case the size of the graph [39]. While not yet formally proven, it is for this reason that stochastic algorithms, which generate and use random variables, are employed.

2.4.2 Enhanced Stochastic Algorithms using GBS

As established in Eq.(19), the probability of a GBS outcome is directly proportional to the hafnian of the submatrix \mathbf{B}_n of the larger matrix \mathbf{B} describing the GBS output. It is also the case that the hafnian of an adjacency matrix describing a graph is equal to the number of its perfect matchings [36]. Crucially, it is shown quantitatively in Ref.[40] that a graph with many perfect matchings is expected to have many edges, thus higher density according to Eq.(25).

Hence we arrive at the vital insight: by constructing a GBS experiment with a matrix \mathbf{B} which corresponds to the adjacency matrix Δ of a graph, the samples drawn at the highest probabilities will correspond to the subgraphs with the highest number of perfect matching. These photon number samples will denote the densest subgraphs, effectively biasing the stochastic algorithm towards higher densities, in place of drawing samples from a uniform distribution. This is possible because both Δ and \mathbf{B} are symmetric matrices. To construct the GBS parameters, \mathbf{B} is decomposed using Takagi-Autonne factorisation into a vector \mathbf{v} and a unitary matrix Λ . The vector determines the squeezing parameters of the input Gaussian states, while Λ determines the interferometer parameters [41].

3 Method

3.1 Bitstrings Statistical Measures

The probability distribution of GBS consists of the individual probabilities for each unique bit-string outcome $p(\mathbf{z})$. For a total of m modes there are a total of 2^m possible unique bit-strings of length m . Using the convention

$$\begin{aligned} p_0 &= p(\mathbf{z} = [0, 0, 0, \dots, 0]) \\ p_1 &= p(\mathbf{z} = [1, 0, 0, \dots, 0]) \\ p_2 &= p(\mathbf{z} = [0, 1, 0, \dots, 0]) \\ &\vdots \\ p_{(2^N-1)} &= p(\mathbf{z} = [1, 1, 1, \dots, 1]) \end{aligned} \tag{26}$$

we can define a statistical metric that quantifies the *difference* between the simulated GBS output distribution p^{GBS} and the Greedy distribution p^{Greedy} . In this project, we employ the widely used Total Variation Distance \mathcal{D} and Kullback-Leibler (KL) divergence \mathcal{KL} [21]. The statistical distance metrics for our discrete probability distributions are given by

$$\mathcal{D}(\text{Greedy}, \text{GBS}) = \frac{1}{2} \sum_i |p_i^{\text{Greedy}} - p_i^{\text{GBS}}| \tag{27}$$

$$\mathcal{KL}(\text{Greedy}, \text{GBS}) = \sum_i p_i^{\text{Greedy}} \ln \left(\frac{p_i^{\text{Greedy}}}{p_i^{\text{GBS}}} \right). \tag{28}$$

KL divergence is also a measure of relative informational entropy between the two probability distributions, giving a definite measure of information loss between the approximation and empirical distribution in units of information. For \log_2 in Eq.(28), these are in *bits*. for the natural log, these are in *nats* [21].

3.2 Computational Toolkit

We developed and executed all algorithms and simulations in Python, leveraging specialised quantum photonic packages when necessary. This section presents the details of implementation and performance evaluations.

3.2.1 Software Package for Photonic Quantum Circuit simulations

In order to assess the performance of the marginal-based greedy algorithm, we benchmarked its performance against the full output distribution of Gaussian Boson Sampling (GBS). To accomplish this, we simulated GBS for various mode numbers using Xanadu's recently released Python package `StrawberryFields`[42]. This comprehensive, full-stack Python library facilitates the design of photonic quantum circuits, implementation of quantum gates, and execution of measurements on quantum photonic platforms. Photonic quantum circuits are represented as `Program` objects. The program can be simulated on various available backends, followed by the application of quantum operations and the performance of measurements. Within the context of a program, circuit components can be applied using built-in functions (for instance `Vacuum()`, `Bsgate()`, `PNR()`), along with their required input parameters. A specific unitary matrix \mathbf{U} for the GBS interferometer, with the desired dimensions, was generated using the `unitary_group()` function from the `scipy.stats` package and was consistently employed throughout a given simulation.

The Fock backend is one of the virtual quantum computer backends available. It is built on Python’s package NumPy and represents quantum states using their vector representation in a truncated photon number basis (superposition of superpositions of the Harmonic Simulator energy states, as outlined in the Fock basis of section 2.1.2). In the Fock backend, a Fock cut-off is required to specify the number state at which the representation will be truncated. This backend was utilised for the investigation into arbitrary states for boson samplers.

The Gaussian backend employs the covariance matrix representation of states detailed in section 2.1.2, resulting in considerably faster performance compared to the Fock backend. This was used for the majority of the investigation, with the exception of the Renema arbitrary states, as it is incapable of simulating non-Gaussian states. In this backend, a Fock cut-off is still a necessary input, which specifies the highest number state for which the output probabilities of the GBS simulations are computed.

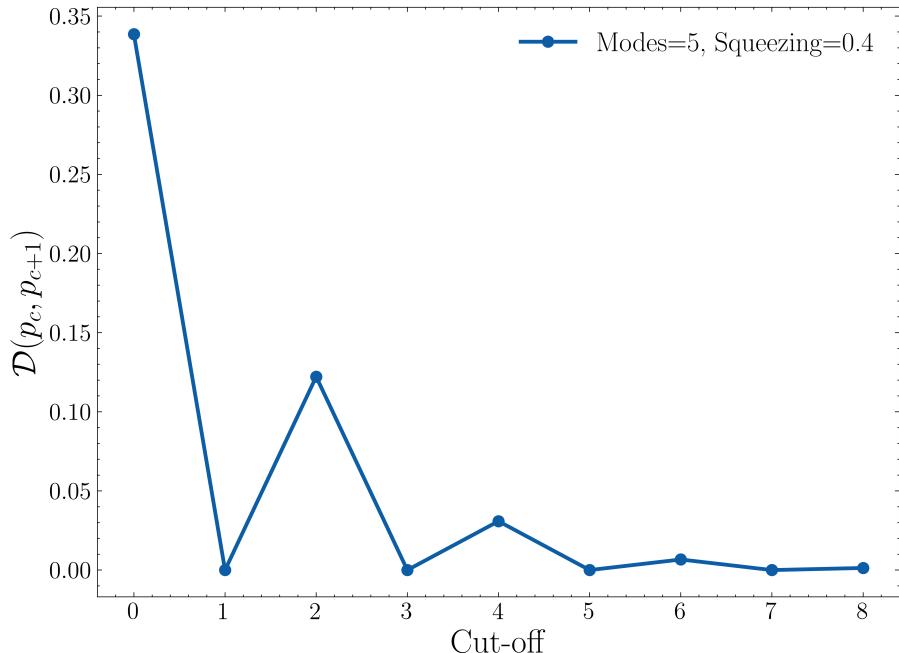


Figure 7: Decrease in cut-off error with increase in cut-off value c . The total variation distance on the y-axis, $\mathcal{D}(p_c, p_{c+1})$, is the measure of the difference between the output probability distributions for a 5-mode GBS simulation with a Fock cut-off of c and an identical simulation with a cut-off of $c + 1$. A randomly generated unitary matrix was used for the interferometer, along with a fixed squeezing parameter of 0.4 for all modes.

It is therefore evident that a primary source of error in our simulations is the cut-off value. Utilising a higher cut-off results in a more accurate state representation and outcomes; however, it could quickly overwhelm computer memory. This is primarily due to circuit operations, where the states are represented as unitary matrices with dimensions equal to the number of Fock coefficients. An inadequate cut-off value leads to significant portions of the Hilbert space being neglected, which affects the final probability distribution. To quantify the cut-off error for a specific simulation, we executed the simulation with a maximum cut-off value c that our computational resources allowed concerning computer memory and run-time. We then conducted the same simulation with a cut-off of $c - 1$. The target cut-off error is subsequently derived from the differences between the two probability distributions at each corresponding point.

While this approach might initially appear asymmetric, as we are not considering the cut-off of $c + 1$. Nonetheless, it is justified by the observation that the error decreases with increasing cut-off, as demonstrated in Fig.7. Consequently, in a GBS simulation, the difference between a cut-off of $c + 1$ and c will always be smaller than the difference between c and $c - 1$.

3.2.2 Greedy Algorithm Implementation tests

The greedy algorithm served as a vital tool in this investigation. It was therefore important to benchmark our execution to ensure its accurate performance. First, the statistical distance between the Greedy algorithm's output and GBS experiment simulations is expected to decrease as the approximation order of the greedy algorithm increases. This is because, for higher approximation orders, the algorithm uses higher-order marginal distributions (larger subsets of modes) to construct the full distribution. Secondly, the statistical distance in the marginal orders up to the approximation order should be approximately zero, indicating that the distribution generated by the greedy algorithm utilises accurate marginals up to the approximation order.

To test these features, we determined the mean total variation distance $\overline{D}(\text{Greedy}, \text{GBS})$ and the mean KL divergence $\overline{KL}(\text{Greedy}, \text{GBS})$ for increasing marginal orders by comparing the ideal marginal distributions of a specific order with their corresponding distributions in the greedy algorithm. The results depicted in Fig.8 confirmed the aforementioned expected behaviour for the greedy algorithm implementation.

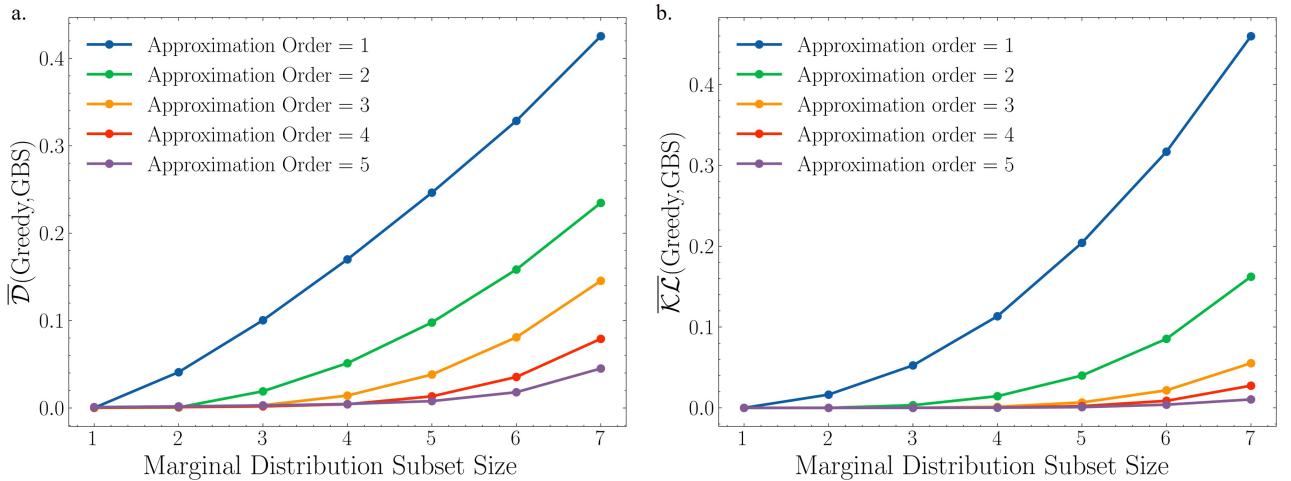


Figure 8: Average statistical distance between the greedy algorithm distribution and the corresponding simulated GBS distribution for increasing marginal orders. A 7-mode experiment with fixed squeezing parameters of 0.5 was used throughout, with a random unitary matrix for the interferometer input. $L = 2000$ bitstring samples were generated for the greedy algorithm. The y-axes display the mean variational distance and mean KL divergence for Fig.(a) and Fig.(b) respectively. Here, approximation order refers to the order of the marginal distributions used as inputs to the Greedy algorithm.

The fundamental mechanism of the greedy algorithm is the accurate determination of the low-order marginal distributions, as outlined in section 2.2.3. To test our Python code for these calculations, we compared them with the corresponding empirical distributions obtained from state vectors in the StrawberryFields GBS simulations. It was important to ensure that the GBS simulations were executed for a sufficient duration to obtain enough samples for an accurate comparison.

3.3 Noise Models Implementation

A central objective of this project is to establish a connection between noise and the complexity of GBS. Determining whether any specific type of noise is related to the omission of high-order correlations in GBS can be revealed by testing if the low-order marginal-based Greedy algorithm can reproduce a target GBS under noisy conditions. This section outlines the noise models employed for injection into GBS simulations. The theory behind each noise model is outlined in section 2.2.4.

3.3.1 Optical Loss

Simulating noise from optical loss involves deliberately losing a fraction of the photons from the idealised simulation. To achieve this, we couple each output mode to an ancillary ‘environmental’ mode initialised in the vacuum state $|0\rangle$. The coupling is achieved through beamsplitters $\Lambda(\phi)$, with transmissivity angle $t = \cos^2 \phi$. In this manner, each photon exiting the interferometer has a probability of reflecting on the beamsplitter and subsequently becoming ‘lost’ to the environment. This setup is shown in Fig.9. The fraction of lost light is then given by the loss parameter $l = 1 - t$.

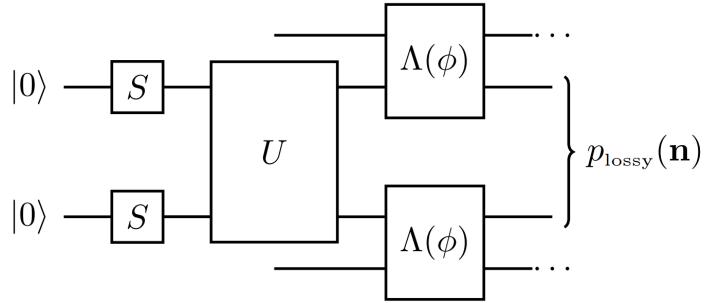


Figure 9: Optical loss noise model for an example 2-mode GBS experiment. Each output mode is coupled to an appended environmental mode using a beamsplitter with transmissivity angle ϕ . The original output modes (without the extra environmental modes) constitute the lossy distribution p_{lossy} .

To obtain the distribution that incorporates optical loss p_{lossy} , we run the GBS simulation with the appended environmental modes in StrawberryFields using the Gaussian backend. After tracing out the environmental modes, we obtain p_{lossy} as marginals of the total distribution of the simulation state vector. By varying the transmission angle ϕ from 0 to 2π , the loss fraction l increases from 0 to 1.

3.3.2 Gate Error

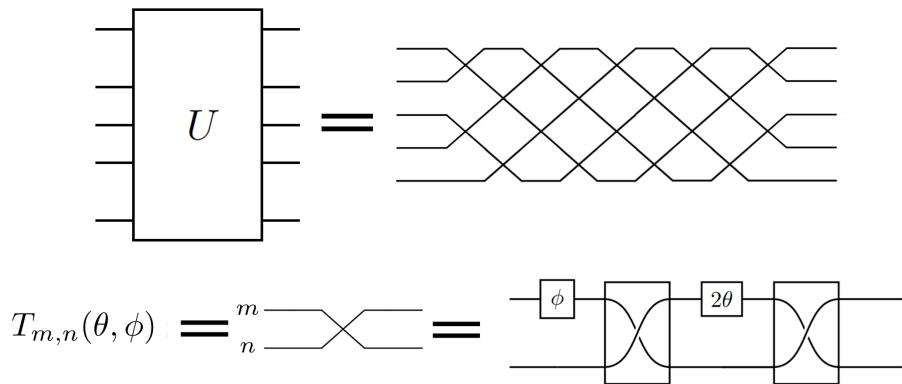


Figure 10: Gate error noise model. Top diagram: The GBS unitary interferometer is decomposed rectangularly into a mesh of beamsplitters and rotation gates. Each crossing to a smaller circuit, shown at the bottom, of two 50:50 beamsplitters and two rotation gates with parameters θ and ϕ . Each crossing corresponds to a transformation matrix $T_{m,n}$, with m, n denoting the two modes it acts on.

To implement gate error in our GBS simulation, we needed to perturb interferometer parameters in our simulation to the deviated unitary \tilde{U} . First, we rectangularly decompose the interferometer unitary U into a mesh arrangement of beamsplitters and rotation gates, as illustrated

in Fig.10 for an interferometer with 5 modes. This is based on the scheme of ref.[43], giving:

$$U = D \left(\prod_{(m,n) \in W} T_{m,n}(\theta, \phi) \right). \quad (29)$$

where W is a defined configuration of beamsplitters and phase shifters, $T_{m,n}$ is a transformation matrix acting on modes m and n constructed by two 50:50 beamsplitters and two rotation gates with shift parameters θ and ϕ , and D is a diagonal matrix implemented through rotation gates (phase shifts) on all individual output modes of the interferometer.

The decomposition process is implemented using the algorithm suggested in ref.[43], with a 5-mode example yielding

$$U = T_{3,4}^{-1} T_{4,5}^{-1} T_{1,2}^{-1} T_{2,3}^{-1} T_{3,4}^{-1} T_{4,5}^{-1} D T_{1,2} T_{2,3} T_{3,4} T_{1,2}. \quad (30)$$

We then apply a fixed deviation δ^σ to each rotation parameter that is sampled from a normal distribution $\mathcal{N}(\mu = 0, \sigma)$ with mean $\mu = 0$ and standard deviation σ in the following way:

$$\phi_i \longrightarrow \phi'_i = \phi_i + \delta_i^\sigma \quad \theta_i \longrightarrow \theta'_i = \theta_i + \delta_i^\sigma \quad (31)$$

after which we construct the deviated unitary

$$U(\vec{\theta}, \vec{\phi}) \rightarrow \tilde{U}(\vec{\theta}', \vec{\phi}') \quad (32)$$

according to Eq.(30). Note here that by the vector notation $\vec{\theta}$ we mean all the θ in the whole mesh circuit. We then use \tilde{U} in our GBS simulation in Strawberryfields with the Gaussian backend. The value of σ is the noise parameter in this model, as it determines the size of the induced deviations. Since the deviation itself is sampled from a normal distribution, it can lead to random fluctuations in the applied deviations. To overcome this, for each σ we perform

$$p'_i = \int_{\mathcal{N}(\mu=0, \sigma)} p_i(\theta + \delta, \phi + \delta) p(\delta) d\delta \quad (33)$$

where $\{p_i\}$ are the component of the noisy GBS distribution, $p(\delta)$ is the probability of sampling deviation δ , and $\{p'_i\}$ are the components of the noisy GBS distribution after averaging out the fluctuations. To solve the integral we perform Monte Carlo integration [44]:

$$p'_i = \frac{1}{J} \sum_j^R p_i(\theta + \delta^{\sigma,j}, \phi + \delta^{\sigma,j}) \quad (34)$$

where R is the number of repetitions.

3.3.3 Distinguishability Effects

To account for the partial distinguishability of GBS experiment modes, we need to compute the output state vector for each photon degree of freedom separately. Here we only considered wavelength differences, since these arise as the main source of distinguishability. However, this is an inherently challenging task, as there are infinitely many wavelengths that would need to be considered.

To simplify the computational complexity, we employed a simplified model in which the imperfection of each mode - the ‘segment’ of light with a different wavelength - is orthogonal to all other imperfections. This means that they do not interfere with each other, which allowed us to assume that the imperfection of each mode is made up of a single wavelength. These

assumptions are motivated by real experiments, as in many cases the imperfection of a single mode primarily falls within a narrow range of wavelengths [27]. The various imperfections across different modes often fall on different ranges and thus do not overlap. A diagram illustrating an example of this model is shown in Fig. 11.

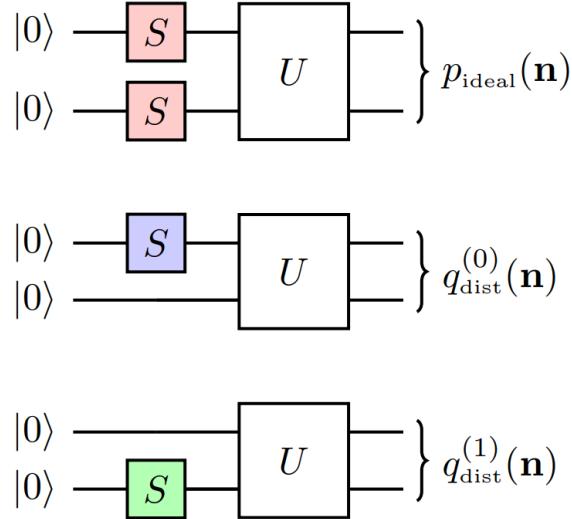


Figure 11: Distinguishability model for a 2-mode GBS circuit. The top circuit has perfectly indistinguishable light, which gives the ideal output distribution p_{ideal} . The next two circuits are the imperfection circuits, representing the wavelength imperfection in each mode. The imperfection circuits give the probability distributions $q_{\text{dist}}^{(0)}$ and $q_{\text{dist}}^{(1)}$ which are subsequently convolved with the ideal distribution.

First, we compute the individual probability distribution for each wavelength independently of the rest. Following this, we carry out a convolution of all the individual distributions. Referring to the example case of Fig. 11, we initially perform a convolution between the distributions p_{ideal} and $q_{\text{dist}}^{(0)}$, which represent the ideal wavelength (depicted in red) and the imperfection of the first mode (depicted in blue), respectively. This results in the following calculations for the probability components:

$$\begin{aligned} p'(00) &= p(00) \cdot q^{(0)}(00) \\ p'(01) &= p(01) \cdot q^{(0)}(00) + p(00) \cdot q^{(0)}(01) \\ p'(10) &= p(10) \cdot q^{(0)}(00) + p(00) \cdot q^{(0)}(10) \\ p'(11) &= p(11) \cdot q^{(0)}(00) + p(01) \cdot q^{(0)}(10) + p(10) \cdot q^{(0)}(01) + p(00) \cdot q^{(0)}(11) \end{aligned}$$

where $p'(00) = p'(\mathbf{z} = [0, 0])$ in accordance with the convention of section 3.1. We subsequently perform a further convolution of p' with the next imperfection mode $q_{\text{dist}}^{(1)}$ (depicted in green):

$$\begin{aligned} p''(00) &= p'(00) \cdot q^{(1)}(00) \\ p''(01) &= p'(01) \cdot q^{(1)}(00) + p'(00) \cdot q^{(1)}(01) \\ p''(10) &= p'(10) \cdot q^{(1)}(00) + p'(00) \cdot q^{(1)}(10) \\ p''(11) &= p'(11) \cdot q^{(1)}(00) + p'(01) \cdot q^{(1)}(10) + p'(10) \cdot q^{(1)}(01) + p'(00) \cdot q^{(1)}(11) \end{aligned} .$$

This results in the final probability distribution p'' , which includes distinguishability effects. It is important to note that the parameter governing noise levels in this context is the squeezing of the imperfections. Higher squeezing corresponds to a greater photon number and, as a

result, an increased degree of distinguishability. In our implementation, we assigned the same squeezing parameter to all wavelength imperfections within our simulation. We apply an analogous procedure to incorporate distinguishability effects into an m -mode GBS experiment. We initially generate $m+1$ replicas of our circuit corresponding to one for the ideal wavelength and one for each mode imperfection. We then acquire the state vectors for all of them by utilising the Gaussian backend of our simulation. From these state vectors, we calculate the respective probability distributions and subsequently perform the convolution in the previously described manner.

3.4 High-order Photon Interference with Renema States Implementation

By utilising the proposed Renema states as inputs to boson samplers, we aimed to enhance the high-order interference of these experiments and investigate the effects on the greedy algorithm performance. We simulated three experiments of boson samplers with $m = 6$ modes, each with different levels of disjunct input states. The first sampler received inputs of Renema states $|\Psi\rangle_R = \frac{1}{2}(|111000\rangle + |000111\rangle)$. The $(\frac{m}{2})^{\text{th}} = 3^{\text{rd}}$ order photon interference is expected to be enhanced in this sampler. The second boson sampler experiment had the input state $|\Psi\rangle_{SR} = \frac{1}{2}(|111000\rangle + |011100\rangle)$ as input, which is similar to a Renema state and will be referred to as semi-Renema state. Finally, the third sampler had an input state of $|\Psi\rangle_{BS} = \frac{1}{2}(|111000\rangle)$, the normally used Fock state for boson sampling experiments. We then approximate the output of these circuits with the Greedy algorithm and measure the distance between the approximate and simulation distribution for each circuit. Utilising the output state vectors of each simulation, we determine all the 2nd-order marginals (two-mode subset of output modes) for each of the three experiments, which are also the inputs in the Greedy algorithm. We then measured the statistical distance between the Greedy distribution and the simulated output distribution for each of the boson samplers.

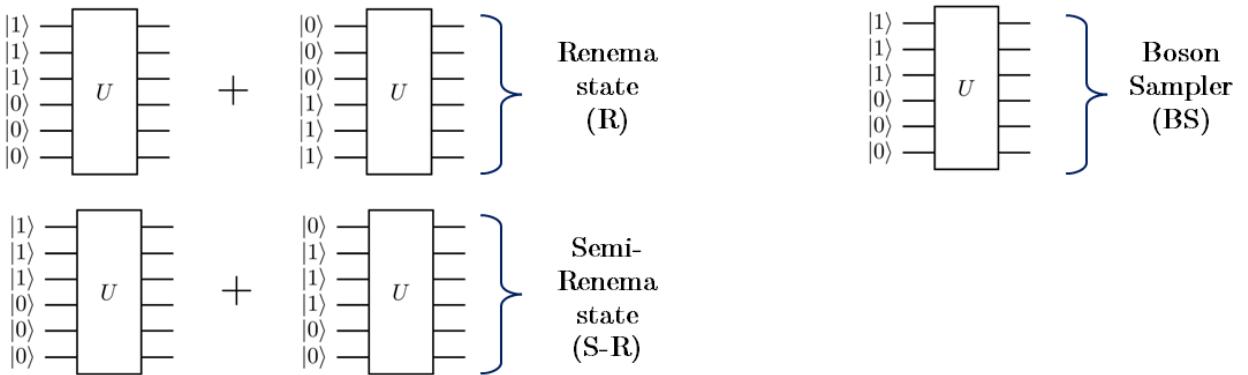


Figure 12: Renema, Semi-Renema and Boson Sampling circuits. The circuit addition represents their state superposition.

For these simulations, the appropriate backend to use was the Fock backend, as these are highly non-Gaussian states. To create the superposition needed for the Renema and Semi-Renema states, we created two bosonic sampling circuits: one for each of $|\psi\rangle_1$ and $|\psi\rangle_2$ of Eq.(23), as depicted in Fig.12. We then added their output state vectors linearly and obtained the total output distribution of the superposition. Note that to prepare these Fock states, we adopted the `Ket` state preparation method of `StrawberryFields`, as opposed to the more natural built-in `Fock` method for creating single-mode Fock states. This is because the latter entangles the state as it traces out each prepared mode, swapping it with the specified Fock state.

3.5 Graph Search Enhancement Implementation

The goal of this implementation is to compare the performance of uniform random search against the Greedy-enhanced algorithm in the D_kS problem. An Erdős–Rényi graph of size $\mathcal{V} = 6$ was initialised with connection probability $p_{\text{E.R.}} = 0.7$ using the python package **NetworkX**[45]. The generated graph was unweighted; hence we appended random weights derived from a continuous uniform distribution between 0.01 and 0.02 on each edge. The graph shown in Fig.13 was used throughout the implementation by fixing the random seed of the Python built-in pseudo-random number generator.

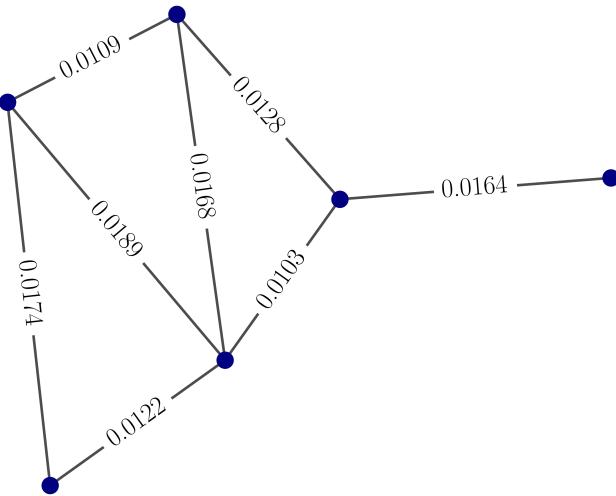


Figure 13: Graph used throughout the implementation. The blue dots represent nodes and the lines represent undirected edges. The edge weights are shown on each edge.

The adjacency matrix Δ of this graph was determined using a **NetworkX** built-in function, giving

$$\Delta = \begin{pmatrix} 0 & 0.0164 & 0 & 0 & 0 & 0 \\ 0.0164 & 0 & 0 & 0.0103 & 0.0128 & 0 \\ 0 & 0 & 0 & 0.0122 & 0 & 0.0174 \\ 0 & 0.0103 & 0.0122 & 0 & 0.0168 & 0.0189 \\ 0 & 0.0128 & 0 & 0.0168 & 0 & 0.0109 \\ 0 & 0 & 0.0174 & 0.0189 & 0.0109 & 0 \end{pmatrix} \quad (35)$$

which is diagonally symmetric as expected since the graph is undirected. We used Autonne-Takagi decomposition on Δ using the built-in function of **StrawberryFields** `decompositions.takagi` to obtain the squeezing parameters vector \mathbf{v} and the interferometer unitary matrix \mathbf{U} that map the graph onto a GBS experiment. We then computed the marginal probabilities for a GBS experiment with these parameters using the procedure of section 2.2.3. Using the greedy algorithm, we generate L bit-string samples corresponding to subgraphs in the associated graph. The number of photon detections corresponds to the number of vertices k of its associated subgraph, while the detection positions correspond to the vertices. For instance, the bit-string ‘110010’ would correspond to the subgraph of nodes {1,2,5}. To find the density of the subgraph corresponding to each bit-string sample, we reduce the adjacency matrix to Δ_R that only contained the detected vertices. Using Eq.(25) with Δ_R we calculate the density of the subgraph. We chose to scale the squeezing parameters $r_i = \tanh^{-1}(cv_i)$ according to the constant $c > 0$ such that the average number of detection in each bit-string is on average equal to

the size of the subgraph k . The mean photon number $\langle n \rangle$ is then given by:

$$\langle n \rangle = \sum_{j=1}^m \frac{(cv_j)^2}{1 - (cv_j)^2} \quad (36)$$

This scaling factor does not affect the relative probability distribution [41]. The procedure employed for choosing subgraph samples is as follows:

1. Choose the number of subgraph samples n .
2. Generate and store L greedy algorithm samples of size \mathcal{V} .
3. Filter the samples that have k detections (bit-string ‘1’ appearing k times).
4. Calculate the density of each sample and save the maximum density obtained
5. Discard these samples, choose another set of n subgraph samples and repeat from step 1 for 100 iterations.
6. Calculate the average maximum density.
7. Increase the number of subgraph samples n and repeat.

The same procedure was employed for uniform random search, where instead of the greedy algorithm in step 2 we use NumPy’s `random.randint`. Finally, we compared the average maximum densities at each value of n .

4 Results

4.1 Marginal-Based Greedy Algorithm and Noise

Fig.14 depicts the plots for total variation distance and KL divergence between the 2nd order Greedy algorithm and GBS simulation distributions for increasing levels of optical loss. The simulations were run for 5 modes with a cutoff of 7, and a greedy matrix \mathbf{J} size of $L = 2000$. For moderate levels of optical loss, both distance metrics \mathcal{D} and \mathcal{KL} decrease. This means that the marginal-based Greedy algorithm, which only utilises the noiseless 2-mode marginal distributions as inputs, more closely resembles a noisy GBS distribution than the ideal distribution it is targeting. This is evidence that the omission of higher-order marginals in GBS is related to introducing noise to the distribution. After the initial drop, the statistical distance rapidly increases for higher levels of optical loss. This phenomenon can be explained as, at high levels of loss, most photons in the GBS simulation get lost to the appended environmental modes. This means that most of the output probability mass is shifted to the all-zero bit-string $p_0 = [0, 0, 0, \dots, 0]$. In contrast, the omission of high-order interference process in the Greedy algorithm has the opposite effect, shifting the distribution from the all-zero bit-string to outcomes with photon detections. It can then be inferred that introducing moderate levels of optical loss to the ideal GBS simulation has a similar effect on the shift in the distribution as ignoring the high-order photon interference effects.

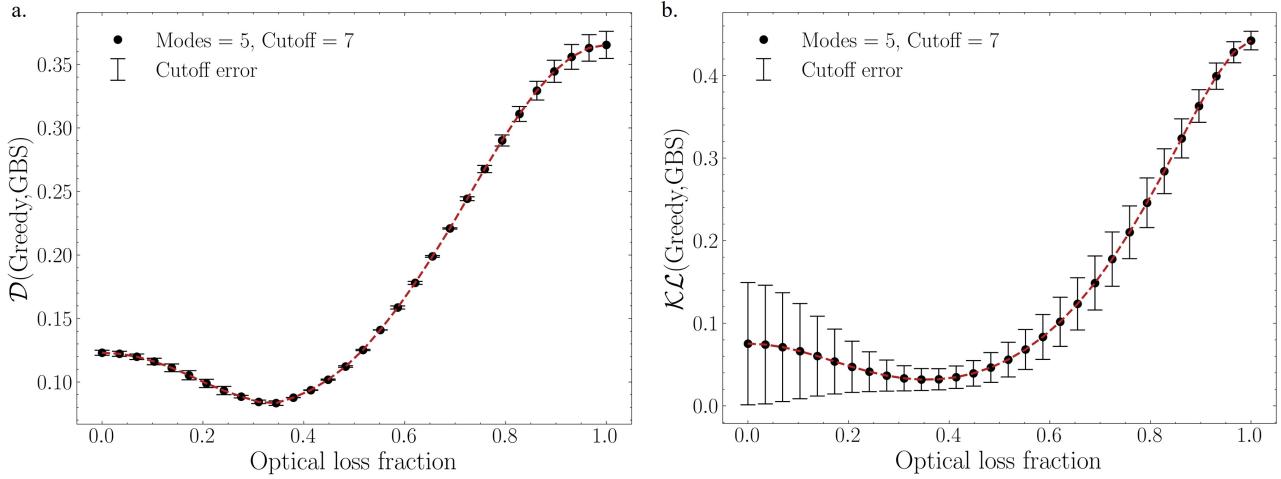


Figure 14: Statistical distance between Greedy distribution and output distribution of a 5-mode GBS simulation under increasing levels of optical loss. The optical loss fraction denotes the probability of each photon getting lost to the environment in each output mode. A Fock cut-off of 7 was used, with the error bars denoting the Cut-off error. The red dotted line is the χ^2 fit through all the data points. Fig.(a) depicts the total variation distance and Fig.(b) depicts the KL divergence.

The results for the distinguishability noise model follow a similar trend to those observed in the optical loss scenario. Fig.15 depicts the resulting variation distance and KL divergence between the 2nd order Greedy distribution and the GBS distribution leveraged by increasing levels of distinguishability imperfections. The initial decrease in statistical distance suggests that the effect of blurring induced by the convolution of additional distinguishable wavelengths causes the GBS distribution to more closely align with the Greedy distribution, establishing a correlation between the distinguishability effects and the exclusion of high-order photon correlations.

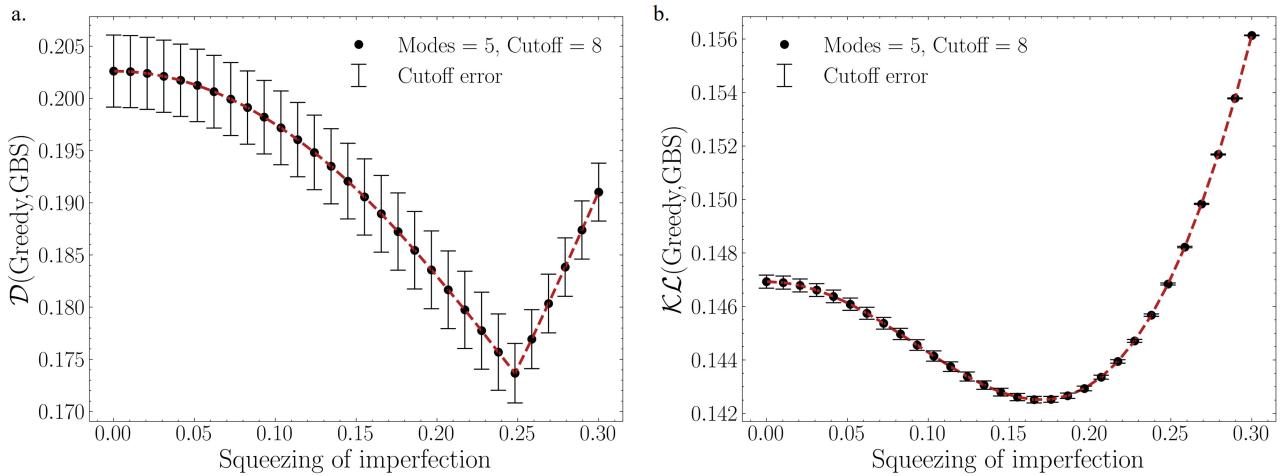


Figure 15: Statistical distance between the Greedy distribution and the output distribution of a 5-mode GBS simulation under increasing levels of distinguishability imperfections. The Greedy algorithm was run with 2-mode marginal inputs for $L = 2000$ bit-string samples. The primary squeezing parameter was set to 0.5, with Fock cut-off of 8. The red dotted line is the χ^2 fit through all the data points. Fig.(a) depicts the total variation distance and Fig.(b) depicts the KL divergence.

Similar to the optical loss model, the statistical distance increases rapidly at higher levels of distinguishability noise. This can be attributed to the rise of the mean photon number of the circuit at each convolution with an imperfection circuit. The mean photon number was not kept constant, as the original (primary) wavelength squeezing was not scaled down in proportion to the addition of new wavelengths. At high squeezing of imperfections, the rise in mean photon

number becomes the dominant factor in the GBS outcome distribution, overall increasing the statistical distance. In spite of this, the statistical distance still decreases initially which implies that the blurring of the convolution with the additional wavelengths has a greater effect than the overall increase in mean photon number.

This general trend was not observed in the results for the gate error model. As seen in Fig.16, the statistical distance increases as the gate deviation increases, without the otherwise observed initial drop in distance. Each data point was repeated 100 times for the Monte Carlo integration as outlined in section 3.3.2. A plausible explanation for not observing the same phenomenon in the gate error model might be the substantial fluctuations it exhibits. This could be addressed by increasing the integration samples in each Monte Carlo routine; however, that would considerably push the computational limitations and under the time constraints of this investigation. It is also the case that the deviations applied were equal for both the beam-splitter and rotation gate angle parameters $\vec{\theta}$ and $Vec\phi$. It is possible that one of these parameters has a more dominant impact on the deviated circuit. As a future extension, it could be valuable to examine the individual contributions of each gate separately, in particular determining whether this would result in an initial decrease in distance.

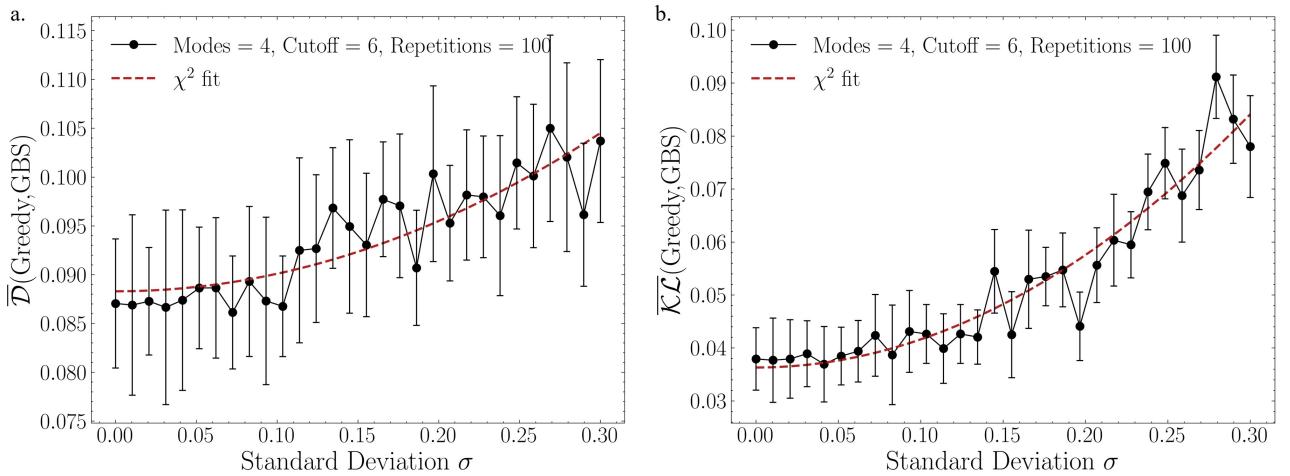


Figure 16: Statistical distance between the Greedy distribution and the output distribution of a 4-mode GBS simulation with gate errors of increasing deviation parameters Δ . The Greedy algorithm was run with 2-mode marginal inputs for $L = 2000$ bit-string samples. The Monte Carlo integration repetitions were set to 100, with Fock cut-off of 6. The red dotted line is the χ^2 fit through all the data points. Fig.(a) depicts the total variation distance and Fig.(b) depicts the KL divergence.

Considering again the optical loss model, we devised two approaches to isolating the statistical distance effects that cause the initial decrease. In the first method, we adjusted the squeezing parameters according to Eq.(36) using a scaling factor to maintain a constant total mean photon number throughout all the loss levels. This way, when photons are lost to environmental modes, the increased squeezing compensates for the lost photon by adding more to maintain a constant number. We then measured the statistical distance between the adjusted GBS experiments and the Greedy distribution with marginal inputs derived accounting for the scaled squeezing and fixed amount of loss. The result is depicted in Fig.17. While the general trend is decreasing as expected, the adjusted squeezing results in fluctuations in the data points, since for each data point the Greedy algorithm is targeting a different GBS experiment, with different marginal distributions. It was also computationally prohibitive to move to higher loss fractions, as increased squeezing requires higher computational resources.

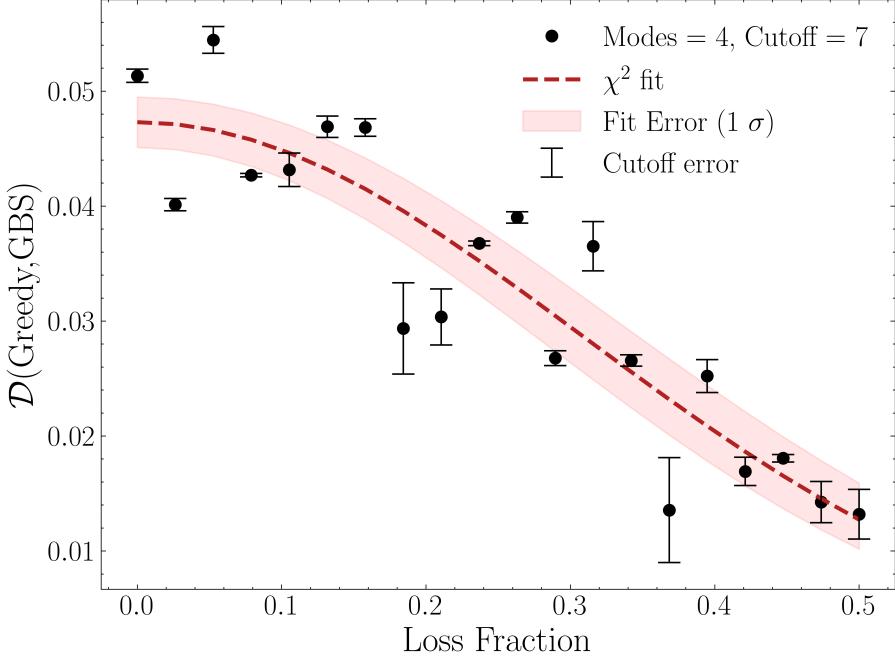


Figure 17: Total variation distance between the Greedy distribution and the output distribution of a 4-mode GBS simulation under increasing optical loss and scaled squeezing parameters. The Greedy distribution was calculated from $L = 2000$ bit-string samples.

As a second method of isolating the decrease, we considered only examining outcome distributions conditioned on having a certain fixed number of detections. We anticipated isolating factors causing a reduction in statistical distance this way, as the shift of probability mass towards the all-zero bitstring outcome was not expected to influence the distribution of outcomes with a low number of detections. Despite this, an increase in statistical distance was still observed in these conditional probability distributions, with Fig. 18 illustrating one such distribution for bit-strings with only 1 detection.

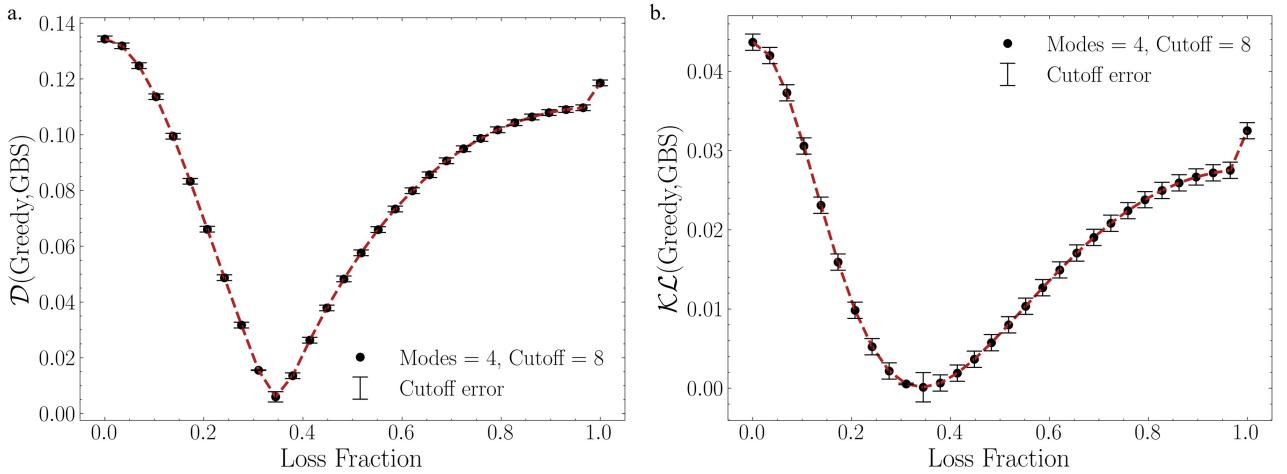


Figure 18: Statistical distance between the Greedy distribution and a 4-mode GBS outcome distribution under increasing levels of optical loss, filtered only to those outcomes with 1 photon detection. The Fock cut-off was set to 8. Fig.(a) depicts the total variation distance and Fig.(b) depicts the KL divergence.

We now consider the point at which the Greedy distribution is closest to the ideal GBS distribution out of all noise models. This is at 0.25 squeezing of the imperfection on the distinguishability noise model. To reveal the point at which the Greedy distribution deviates from the ideal, we measure the mean statistical distance of all the marginal subsets between

the Greedy distribution and the noisy GBS distribution for increasing marginal sizes. The results, depicted in Fig.19, reveal that the low-order marginals (subset distribution sizes of 1-3) of the Greedy distribution are closer to the ideal GBS simulation. This can be attributed to the fact that the greedy algorithm is supplied these low-order marginals at initiation. It is after a marginal subset size of 4 that the greedy resembles a noisy GBS more than the ideal. This suggests that the high-order marginal distributions of the noisy GBS experiment closely resemble the workings of the greedy algorithm, and hence GBS is closer to the computational complexity of the Greedy algorithm when it is injected with noise.

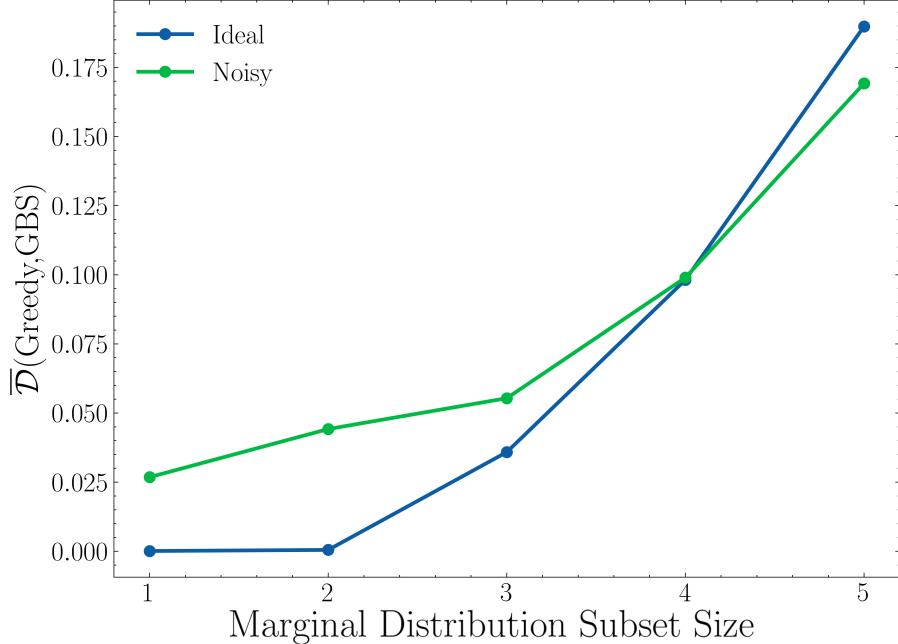


Figure 19: Mean total variation distance between increasing sizes of marginals of the Greedy distribution and the simulated GBS distribution with fixed distinguishability noise of 0.25 (squeezing of the imperfections). The Greedy distribution is of 2nd order approximation and was obtained after $L = 2000$ bit-string samples.

4.2 High-order Photon Interference with Renema states in Boson Samplers

The resulting statistical distance between the three boson samplers having input states as Renema states (R), semi-Renema states (S-R) and boson sampling Fock states (BS), and the greedy algorithm approximation is shown in Fig.20. The highest statistical distance (for both total variation distance and KL divergence) is for the circuit with the pure Renema input states, as theorised. The statistical distance with the circuit with BS states is higher than that of the semi-Renema states, which is not what we initially predicted. One possible reason for this could be that for the SR circuit, by moving the positions of the single-photon states in the 2nd part of the wavefunction, it causes destructive interference effects which in turn suppress the enhances high-order photon interference. Nevertheless, these results act as further evidence for the connection of high-order correlations and the increase in computational complexity of general boson samplers, which includes GBS. These findings establish a link between high-order correlations and high-order photon interference in GBS. Through our previous noise model implementations, we posited that introducing noise and neglecting high-order correlations in GBS experiments share similarities. The results of this section imply a further connection: namely that the introduction of noise in GBS experiments serves to suppress high-order photon interference processes.

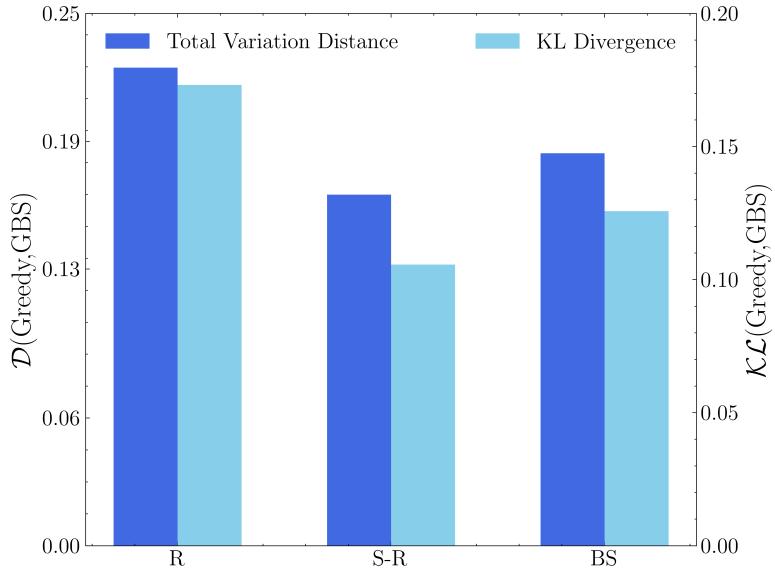


Figure 20: Statistical distance between the Greedy distribution and the boson samplers with the following input states: Renema (R), semi-Renema (S-R), Boson Sampling (BS) Fock states. The left-hand side y-axis shows total variation distance, while the right-hand side y-axis shows KL divergence.

4.3 Graph Search Enhancement Results

Given the results of the previous sections, we concluded that the low-order marginals of the GBS distribution carry a significant amount of information about the full distribution. We therefore anticipated that the marginal-based classical Greedy algorithm would improve graph search algorithms, despite the fact that it only utilises 2nd-order marginals. Our results, illustrated in Fig.21, supported this expectation.

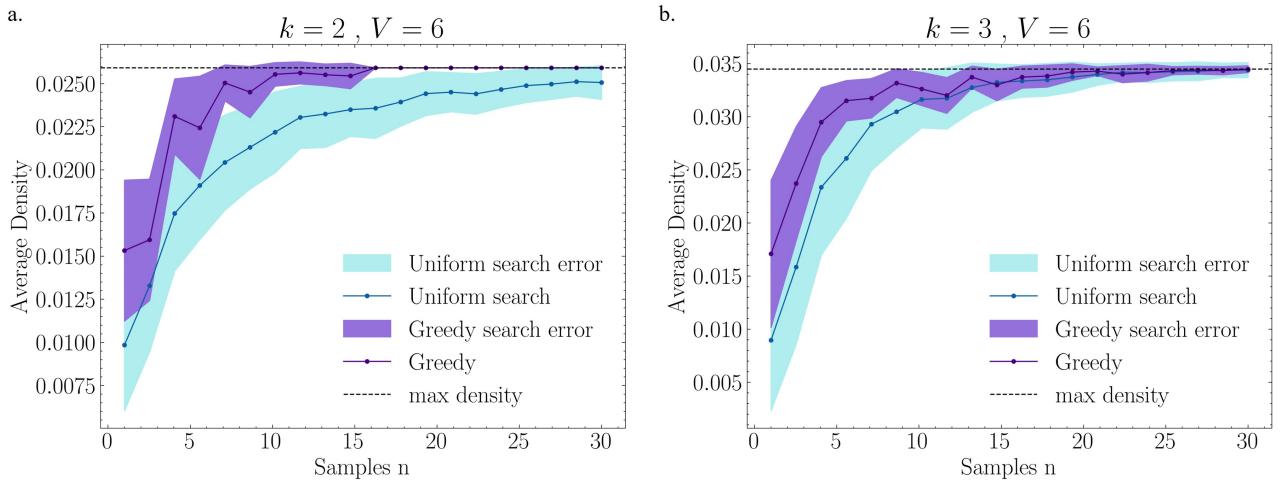


Figure 21: Uniform random search versus Greedy-enhanced search for the DkS problem. The graphs display the average maximum density obtained from both algorithms for increasing number of samples n , as described in section 3.5. Fig.(a) displays the results for subgraphs of size $k = 2$ while Fig.(b) displays the results for subgraphs of size $k = 3$. In both cases, the Greedy search algorithm converges to the maximum faster than the uniform random search.

It can be seen that for both subgraphs sizes of $k = 2$ and $k = 3$, the subgraph of the highest density is reached with fewer samples from the Greedy-search algorithm than uniform random search. Additionally, for the majority of sample sizes n , the Greedy-search algorithm manages to converge to higher maximum averages than its uniform random counterpart. This confirms that the samples obtained from the Greedy search are biased towards the maximum

density, even in cases where only the two-mode marginals were used instead of the full output distribution. Since low-order marginals are easy and efficient to compute, this could serve as a pathway towards enhanced, fully classical algorithms for graph search problems.

5 Conclusion

This investigation uncovers several connections related to the processes of GBS. First, we establish a link between introducing noise into a GBS experiment and the omission of high-order marginal distributions. This was achieved by comparing the output distribution of the low-order marginal-based Greedy algorithm with our GBS simulation outputs affected by optical noise. We further demonstrate that, for moderate levels of optical loss and distinguishability effects, the Greedy distribution more closely resembles the noisy GBS rather than the ideal one. Second, by analyzing boson samplers with Renema states as inputs, we reveal an additional connection between the high-order photon interference processes and high-order marginal correlations in GBS. These initial findings suggest a deeper relationship between the introduction of noise and the suppression of high-photon interference processes in GBS. Third, based on the conclusion that low-order marginals possess significant information-carrying attributes, we demonstrate that a GBS-inspired graph search algorithm enhances the search for the DkS graph problem.

This work offers several avenues for future extension. By leveraging high-performance computing resources, larger GBS experiments with a greater number of modes could be simulated. Additionally, a formal mathematical connection between noise and high-order marginal distributions could be established. Finally, broadening the scope of graph search applications to encompass a wider range of problems could potentially lead to significant speedups in various mathematical challenges, impacting fields such as computer science, machine learning, and other scientific domains.

Acknowledgements

First, I would like to thank Dr Steve Kolthammer for his inspiring supervision of this project and for his eagerness to delve into and introduce new areas of quantum physics in every meeting. Second, I would like to thank my exceptional project partner, with the hope that we will work together again sometime in the future. Finally, I would like to thank Dr. Segio Boixo and Dr. Benjamin Villalonga of Google Quantum AI for our discussion on this project and for clarifying the details of their Greedy algorithm proposal.

References

- [1] Adam Smith, M. S. Kim, Frank Pollmann, and Johannes Knolle. Simulating quantum many-body dynamics on a current digital quantum computer. *npj Quantum Information*, 5(1):1–13, November 2019. Number: 1 Publisher: Nature Publishing Group.
- [2] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. Schuyler Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, Colm A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, Robert S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, Blake R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti. Unsupervised Machine Learning on a Hybrid Quantum Computer, December 2017. arXiv:1712.05771 [quant-ph].
- [3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm, November 2014. arXiv:1411.4028 [quant-ph].
- [4] Edward Farhi and Aram W. Harrow. Quantum Supremacy through the Quantum Approximate Optimization Algorithm, October 2019. arXiv:1602.07674 [quant-ph].
- [5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brando, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michelsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. Number: 7779 Publisher: Nature Publishing Group.
- [6] 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, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, December 2020. Publisher: American Association for the Advancement of Science.
- [7] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob F. F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, Adriana E. Lita, Thomas Gerrits, Sae Woo Nam, Varun D. Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, June 2022. Number: 7912 Publisher: Nature Publishing Group.

- [8] Bryan T. Gard, Keith R. Motes, Jonathan P. Olson, Peter P. Rohde, and Jonathan P. Dowling. An introduction to boson-sampling. pages 167–192. August 2015. arXiv:1406.6767 [quant-ph].
- [9] Juan Miguel Arrazola and Thomas R. Bromley. Using Gaussian Boson Sampling to Find Dense Subgraphs, July 2018. arXiv:1803.10730 [quant-ph].
- [10] Nicolás Quesada. Franck-Condon factors by counting perfect matchings of graphs with loops. *The Journal of Chemical Physics*, 150(16):164113, April 2019. arXiv:1811.09597 [physics, physics:quant-ph].
- [11] Benjamin Villalonga, Murphy Yuezhen Niu, Li Li, Hartmut Neven, John C. Platt, Vadim N. Smelyanskiy, and Sergio Boixo. Efficient approximation of experimental Gaussian boson sampling, February 2022. arXiv:2109.11525 [quant-ph].
- [12] 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, March 2020. Publisher: American Physical Society.
- [13] Jelmer J. Renema. Marginal probabilities in boson samplers with arbitrary input states, December 2020. arXiv:2012.14917 [quant-ph].
- [14] Christopher Gerry and Peter Knight. *Introductory Quantum Optics*. Cambridge University Press, Cambridge, 2004.
- [15] Timothy Heightman. *MSc Thesis: Benchmarking Gaussian Boson Sampling devices with Correlation Functions*. Imperial College London, September 2022.
- [16] Daniel Gottesman, Alexei Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Physical Review A*, 64(1):012310, June 2001. arXiv:quant-ph/0008040.
- [17] J. Eli Bourassa, Rafael N. Alexander, Michael Vasmer, Ashlesha Patil, Ilan Tzitrin, Takaya Matsuura, Daiqin Su, Ben Q. Baragiola, Saikat Guha, Guillaume Dauphinais, Krishna K. Sabapathy, Nicolas C. Menicucci, and Ish Dhand. Blueprint for a Scalable Photonic Fault-Tolerant Quantum Computer. *Quantum*, 5:392, February 2021. arXiv:2010.02905 [quant-ph].
- [18] John Preskill. Fault-tolerant quantum computation. In *Introduction to quantum computation and information*, pages 213–269. World Scientific, 1998.
- [19] Regina Kruse, Craig S. Hamilton, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. A detailed study of Gaussian Boson Sampling. *Physical Review A*, 100(3):032326, September 2019. arXiv:1801.07488 [quant-ph].
- [20] Nicolás Quesada, Juan Miguel Arrazola, and Nathan Killoran. Gaussian Boson Sampling using threshold detectors. *Physical Review A*, 98(6):062322, December 2018. arXiv:1807.01639 [quant-ph].
- [21] L.J. Anthony. The Cambridge Dictionary of Statistics (2nd ed.). *Reference Reviews*, 17(1):29–30, January 2003. Publisher: Emerald Group Publishing Limited.
- [22] Scott Aaronson and Daniel J. Brod. BosonSampling with lost photons. *Physical Review A*, 93(1):012335, January 2016. Publisher: American Physical Society.

- [23] Michał Oszmaniec and Daniel J. Brod. Classical simulation of photonic linear optics with lost particles. *New Journal of Physics*, 20(9):092002, September 2018. Publisher: IOP Publishing.
- [24] Bichen Zhang, Swarnadeep Majumder, Pak Hong Leung, Stephen Crain, Ye Wang, Chao Fang, Dripto M. Debroy, Jungsang Kim, and Kenneth R. Brown. Hidden Inverses: Coherent Error Cancellation at the Circuit Level. *Physical Review Applied*, 17(3):034074, March 2022.
- [25] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Physical Review Letters*, 59(18):2044–2046, November 1987. Publisher: American Physical Society.
- [26] Dan Cogan, Zu-En Su, Oded Kenneth, and David Gershoni. Deterministic generation of indistinguishable photons in a cluster state. *Nature Photonics*, 17(4):324–329, April 2023. Number: 4 Publisher: Nature Publishing Group.
- [27] Junheng Shi and Tim Byrnes. Effect of partial distinguishability on quantum supremacy in Gaussian Boson sampling. *npj Quantum Information*, 8(1):1–11, May 2022. Number: 1 Publisher: Nature Publishing Group.
- [28] Michael Sipser. *Introduction to the theory of computation*. The PWS series in computer science. PWS Pub. Co, Boston, prelim. ed edition, 1996.
- [29] J.J. Renema, A. Menssen, W.R. Clements, G. Triginer, W.S. Kolthammer, and I.A. Walmsley. Efficient Classical Algorithm for Boson Sampling with Partially Distinguishable Photons. *Physical Review Letters*, 120(22):220502, May 2018.
- [30] Tim S Evans. *Network Notes, Complexity and Networks Course*. Imperial College London Press, 2022.
- [31] T. Charlie. Hodgman. *Bioinformatics*. Bios Instant notes. Taylor & Francis, London, 2nd ed. edition, 2010. Publication Title: Bioinformatics.
- [32] Nitesh Chawla and Wei Wang, editors. *Proceedings of the 2017 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Philadelphia, PA, June 2017.
- [33] Ming-Syan Chen, Jiawei Han, and P.S. Yu. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883, December 1996. Conference Name: IEEE Transactions on Knowledge and Data Engineering.
- [34] J. A. (John Adrian) Bondy. *Graph theory*. Graduate texts in mathematics, 244. Springer, New York, 2008. Publication Title: Graph theory.
- [35] Eric W. Weisstein. Perfect Matching. Publisher: Wolfram Research, Inc.
- [36] Brajesh Gupt, Josh Izaac, and Nicolás Quesada. The Walrus: a library for the calculation of hafnians, Hermite polynomials and Gaussian boson sampling. *Journal of Open Source Software*, 4(44):1705, December 2019.
- [37] U. Feige, D. Peleg, and G. Kortsarz. The Dense k -Subgraph Problem. *Algorithmica*, 29(3):410–421, March 2001.

- [38] Juan Miguel Arrazola and Thomas R. Bromley. Using Gaussian Boson Sampling to Find Dense Subgraphs. *Physical Review Letters*, 121(3):030503, July 2018. arXiv:1803.10730 [quant-ph].
- [39] Paul Black. Dictionary of Algorithms and Data Structures (DADS), 2017.
- [40] M Aaghabali, Saieed Akbari, Shmuel Friedland, Klas Markström, and Zakeieh Tajfirouz. Upper bounds on the number of perfect matchings and directed 2-factors in graphs with given number of vertices and edges. *European Journal of Combinatorics*, 45:132–144, 2015. Publisher: Elsevier.
- [41] S. Sempere-Llagostera, R. B. Patel, I. A. Walmsley, and W. S. Kolthammer. Experimentally finding dense subgraphs using a time-bin encoded Gaussian boson sampling device, June 2022. arXiv:2204.05254 [physics, physics:quant-ph].
- [42] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry Fields: A Software Platform for Photonic Quantum Computing. *Quantum*, 3:129, 2019. eprint: 1804.03159.
- [43] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. An Optimal Design for Universal Multiport Interferometers, February 2017. arXiv:1603.08788 [physics, physics:quant-ph].
- [44] Mark Scott and Paul Dauncey. *3rd Year Computational Physics Course Notes*. Imperial College London, 2021.
- [45] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.