frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 20 November 2024
DOI 10.3389/fcomp.2024.1441879

Biclustering a dataset using photonic quantum computing

Ajinkya Borle* and Ameya Bhave

CSEE Department, University of Maryland Baltimore County, Baltimore, MD, United States

Biclustering is a problem in machine learning and data mining that seeks to group together rows and columns of a dataset according to certain criteria. In this work, we highlight the natural relation that quantum computing models like boson and Gaussian boson sampling (GBS) have to this problem. We first explore the use of boson sampling to identify biclusters based on matrix permanents. We then propose a heuristic that finds clusters in a dataset using Gaussian boson sampling by (i) converting the dataset into a bipartite graph and then (ii) running GBS to find the densest sub-graph(s) within the larger bipartite graph. Our simulations for the above proposed heuristics show promising results for future exploration in this area.

KEYWORDS

biclustering, quantum computing, boson sampling, Gaussian boson sampling, block clustering, co-clustering, two mode clustering, data mining

OPEN ACCESS

EDITED BY
Kamil Khadiev,
Kazan Federal University, Russia

REVIEWED BY
Mingxing Luo,
Southwest Jiaotong University, China
Iskender Yalcinkaya,
Czech Technical University in Prague, Czechia

*CORRESPONDENCE
Ajinkya Borle
✉ aborle1@umbc.edu

RECEIVED 31 May 2024
ACCEPTED 09 October 2024
PUBLISHED 20 November 2024

CITATION
Borle A and Bhave A (2024) Biclustering a dataset using photonic quantum computing. *Front. Comput. Sci.* 6:1441879.
doi: 10.3389/fcomp.2024.1441879

1 Introduction

Quantum machine learning is an emerging field of study that is at the intersection of quantum physics and machine learning. It contains research problems that span from leveraging quantum computing for machine learning, to the use of machine learning methods to model quantum physics. Broadly speaking, our work is in the former set of problems. We study the use of computational models enabled by photonics, specifically boson sampling and Gaussian boson sampling (GBS), for an unsupervised learning problem: biclustering.

Biclustering or Co-clustering is the selection of rows and columns of a matrix based on a given criteria (largest values, similar values, constant values, etc.) (Mirkin, 1997). It currently has applications in (but not limited to) bioinformatics (Pontes Balanza et al., 2015; Xie et al., 2019; Castanho et al., 2022), text mining (de Castro et al., 2007; Orzechowski and Boryczko, 2016), recommender systems (Choi et al., 2018; Sun and Zhang, 2022), and even fields like malware analysis (Raff et al., 2020). While the time complexity of biclustering depends on the exact formulation of the problem (i.e., the criteria for the biclusters), the problems of particular interest are the ones for which the decision problems that are NP-complete in nature; e.g., does this matrix have a bicluster of size $b_1 \times b_2$? An answer to such a question can be verified in polynomial time, but not found in polynomial time (Cormen et al., 2022). Therefore, meta-heuristics are often used for this task (José-García et al., 2022), to get good solutions fast, but without theoretical guarantees.

Among the many models of quantum computing, certain metaheuristic based models have also been proposed on which relevant machine learning and data mining problems can be mapped onto (Adachi and Henderson, 2015; Kumar et al., 2018; Schuld et al., 2020; Bonaldi et al., 2023). These include quantum annealing (Kadowaki and Nishimori, 1998), Boson sampling (Aaronson and Arkhipov, 2011), and GBS (Hamilton et al., 2017). While a method to apply quantum annealing to the task of biclustering already exists (Bottarelli et al., 2018), our work, to the best of our knowledge, is the first one that applies boson sampling and GBS to this problem.

Boson Sampling is a restricted model of quantum computing most easily implemented with photonic quantum computing, in particular, with linear optics (Aaronson and Arkhipov, 2011). GBS is a variant of the above that generates photons by squeezing light (Hamilton et al., 2017). Both models solve #P-hard problems[1] with proposed applications in the fields of graph theory (Mezher et al., 2023), machine learning (Schuld et al., 2020; Bonaldi et al., 2023), and optimization (Arrazola et al., 2018) among others. These are quantum computing models that when applied with photonic quantum computing, are feasible right in the NISQ era of quantum computing but can have utility even beyond it (Madsen et al., 2022; Deshpande et al., 2022).

The contributions of this paper are as follows:

1. We propose and explore the application of boson sampling (and Gaussian boson sampling) to the problem of biclustering in machine learning. To the best of our knowledge, this is the first work to do so.
2. For boson sampling, we applied the unitary dilation theorem (Halmos, 1950; Mezher et al., 2023) to embed our dataset in a unitary matrix.
3. We propose a simulated annealing (SA) technique that uses boson sampling as a subroutine for finding biclusters.
4. For GBS, we show how to embed a dataset as a unitary matrix. This is done by first considering the dataset as a bipartite graph (Karim et al., 2019) and then using the Autonne-Takagi decomposition (Takagi, 1924) on it.
5. We performed preliminary simulations to study the basic characteristics of both boson sampling and GBS for the task of detecting biclusters in a dataset.

Since our work is the first one for this topic, our focus was on establishing the basics that would be crucial for any follow-up research done in the field. Our results show conditions for when boson sampling and GBS do well (e.g., for binary datasets) and certain situations to watch out for (some problems may need lot more samples than others). We believe that our results can give useful insights for future work in this direction.

2 Background

In this section, we will cover the topics and notations that will be used in this work.

2.1 Permanent

A Permanent of an $N \times N$ matrix A is an operation that can mathematically be defined as:

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

Where σ is a permutation of the symmetric group S_N. In other words, a permanent is the summation of the products of all the possible elements with all permutations of index values for rows and columns.[2] This is similar to calculating the determinant using the Leibinz's rule (Miller, 1930) except that the sign of each summand is positive.

Calculating the permanent is a #P-hard problem that takes exponential time even for the best classical algorithms (Ryser, 1963; Glynn, 2013). It is an important component in the analysis of the probabilities from boson sampling.

2.2 Boson sampling

Boson sampling is a non-universal model of quantum computation, pioneered by Aaronson and Arkhipov (2011) on the observations of Troyansky and Tishby (1996). A boson is a subatomic particle that has an integer spin number (e.g., Higgs boson, photon, gluon, etc). More importantly for us, the most popular and feasible approach for its realization is by the use of photons. For the remainder of the paper, we would be using terminology from quantum optics for this process.

Unlike other models of quantum computing, instead of using qubits as the building blocks of quantum information, we use linear optical modes (through which photons can traverse) for carrying and manipulating information. Here, we assume m modes carrying n photons (where $m > n$), are plugged into a linear-optical network made up of beamsplitters and phaseshifters. The linear-optical network can be represented as a Unitary matrix $U \in \mathbb{C}^{m \times m}$. Here we define the input state as

$$|\psi_0\rangle = |n_1, n_2, n_3, ..., n_m\rangle \tag{2}$$

$$\text{such that } n = \sum_{i=1}^{m} n_i \tag{3}$$

$$\text{and } n_i \geq 0 \tag{4}$$

Where Equation 2 is a Fock state that denotes the number of photons in the input state for each of the modes involved. After passing through the linear-optical network, we measure the number of photons in each mode. Let us denote this state[3] by $|\psi'\rangle$

$$|\psi'\rangle = |n'_1, n'_2, n'_3, ..., n'_m\rangle \tag{5}$$

$$\text{such that } n = \sum_{i=1}^{m} n'_i \tag{6}$$

$$\text{and } n'_i \geq 0 \tag{7}$$

Meaning that the number of photons in each mode may have changed at the end of the computation. The probability of a particular Fock state to be measured at the end of a single run (or sample) of this process (given an input state) is denoted by

$$P(\psi'|\psi_0) = \frac{|\mathrm{Per}(U_{\psi',\psi_0})|^2}{n_1!n_2!n_3!...n_m!n'_1!n'_2!n'_3!...n'_m!} \tag{8}$$

Where U_{ψ',ψ_0} is an $n \times n$ sub-matrix constructed from U by taking n_i times the ith column on U, and n'_j times the jth row of U, $\forall\, 1 \leq i, j \leq m$ (Aaronson and Arkhipov, 2011; Mezher et al., 2023).

[1] A complexity class which has problems that are at least as hard as the hardest problems in NP.

[2] In any given product, no row or column can be repeated.

[3] Assuming there is no photon loss.

Sub-matrices that have larger permanents will have a higher probability of being measured. While for a mode, any number of photons larger than 1 will have an adverse effect on the probability. With enough sampling, we get the probability distribution described by Equation 8. Typically for boson sampling, it is often assumed that our starting state $|\psi_0\rangle$ has 1 photon at maximum per mode. And for a variety of problems, it is also expected that the states we are interested in have 1 photon at maximum for a mode in the output (Aaronson and Arkhipov, 2011). Thus, in order for us to apply boson sampling to real world problems, we need to encode them as unitary matrices where the solutions can be decoded from sub-matrices with the largest permanent values.

2.3 Hafnian

Related to the permanent, the Hafnian (Termini, 2006) of a symmetric matrix $A \in \mathbb{C}^{N\times N}$ is a value that is calculated by the following equation

$$\mathrm{Haf}(A) = \sum_{\rho\in P} \prod_{\{i,j\}\in\rho} A_{i,j} \tag{9}$$

Where P is the set of perfect matchings for a fully-connected graph of N vertices.[4] The permanent of a matrix C and its hafnian are connected by the following relationship

$$\mathrm{Per}(C) = \mathrm{Haf}\left(\begin{pmatrix} 0 & C \\ C^T & 0 \end{pmatrix}\right) \tag{10}$$

2.4 Gaussian boson sampling

One of the biggest challenges in the implementation of boson sampling is the production of synchronized single-photons on a large scale. Different schemes of producing photons have been suggested in order to address this, such as Gaussian boson sampling (Hamilton et al., 2017).

In the physical setup for GBS (implemented using photonics), the linear interferometer for m modes is prefixed with squeezing operators on all the modes individually. This does not produce an exact number of photons but depending on the squeezing parameter, can produce photons with an average count per mode from a Gaussian distribution[5] (henceforth known as mean number of photons $\overline{n}$ per mode). The unitary matrix that defines the linear interferometer is typically constructed from a symmetric matrix $A \in \mathbb{C}^{N\times N}$ (where $m = N$)[6] after using a process known as Autonne-Takagi decomposition (Takagi, 1924).

At the end of the computation, the number of photons that appear at each mode are read out $|\psi'\rangle = |n'_1 n'_2 ... n'_m\rangle$. The probability of reading a particular $|\psi'\rangle$ is proportional to

$$P(\psi') \propto c^n \frac{\mathrm{Haf}(A_{\psi'})}{n'_1! n'_2! ... n'_m!} \tag{11}$$

Where c is a scaling parameter from squeezing and n is the sum of all observed photons (Equation 6). The matrix $A_{\psi'}$ is an $n \times n$ matrix constructed by taking n_i times the ith column and ith row of A, for $1 \le i \le m$. For many computational problems however, as a simplified heuristic, the submatrix at the end is constructed from the unique rows and columns indicated by the ith mode for which $n_i > 0$. This is typically done with threshold detectors for measurement and the exact probability distribution for such a setup depends on a matrix function called torontonian (Quesada et al., 2018; Deng et al., 2023a) (a function that is analogous to hafnian). The task then becomes to encode real world problems for which the submatrix with a high hafnian (or torontonian to be more specific) value would yield us the solution.

Readers who want to learn about GBS in depth are recommended to read the original paper (Hamilton et al., 2017).

2.5 The biclustering problem

Like previously mentioned, the problem of biclustering is one where the rows and columns of a matrix are clustered together (called biclusters) depending on a criteria. Biclusters can be formed by different criteria (Hochreiter et al., 2010), some of the popular ones are :

1. Biclusters with a constant value populating all the cells, constant value for each row or constant value for each column.
2. Biclusters where values are unusually high or low (with respect to the rest of the matrix).
3. Biclusters that have low variance.
4. Biclusters that have correlated rows and columns.

The exact time complexity of the biclustering problem is dependent on the biclustering criteria. Many versions of the biclustering problem are NP-Hard in nature. And while polynomial approximation algorithms exist; most of them being quadratic or cubic in their time and space complexity (Cheng and Church, 2000; Prelić et al., 2006), heuristic based methods have been gaining popularity (Maâtouk et al., 2021; Cui et al., 2020; José-García et al., 2022). These approaches attempt to get optimal (or near-optimal) solutions but it is hard to do theoretical analysis for time and space complexities. Our boson and Gaussian boson sampling based methods are in the latter category.

For our work, we will focus on biclusters for two criteria (i) biclusters that have high values and (ii) biclusters that have maximum number of ones in a binary matrix (see "Figure 1" for an example of each).[7] We choose these since they are quite relevant to problems in machine learning and bioinformatics (Wang et al., 2016; Castanho et al., 2024; Prelić et al., 2006). All

[4] This does not mean that A is fully-connected, it just means that the number of matchings considered for the calculation of the Hafnian are from a fully connected graph (that has the same number of vertices as A does).

[5] To be considered as an hyperparameter.

[6] Not considering any hardware restrictions.

[7] Assuming all values lie in the range $[0, 1]$, we can also potentially find (iii) biclusters with low values and (iv) biclusters with majority zeros for a binary matrix. We can do this by subtracting current value of each cell in the matrix from 1.

matrices (representing datasets) and sub-matrices (representing biclusters) considered in our simulations are square shaped, but we also comment on how our work can be extended for rectangular datasets and biclusters (in the following sections).

2.5.1 Biclustering example on a machine learning dataset

Figure 2 shows a hypothetical example of how biclustering might be used to gain insights from a dataset, such as the ones relevant to machine learning. For this example, consider a dataset from an ecommerce website that shows the dollar amount spent by various customers on products of six categories for a given month. Here the customers (from C1 to C8) are the records of the dataset and the categories (from Electronics to Beauty) are the features. We want to analyze the buying patterns of users with respect to categories. This example does not have any explicitly stated target feature/variable since the primary task is to extract correlations between features and records (i.e unsupervised learning).

If we consider biclusters based on high values, we can see that customers C1, C2, C5, and C7 and categories Electronics and Books form a 4 × 2 bicluster β_1. While customers C3, C4, C6, and C8 spent a lot on Clothing, Home and Garden, Sports and Beauty. This forms the second bicluster β_2 of size 4 × 4. From these biclusters we can infer things like (i) β_1 indicates tech-savy customers who spend more times on electronics and books, (ii) β_2 may represent customers who are lifestyle-oriented and spend more on personal and home items. Using insights like these, the ecommerce company may be able to do targeted advertisements of products to particular customers, and also make recommendations to customers to products of other categories (e.g., customers who like books may also like electronics).

Here, if we had used ordinary (traditional) clustering on customers, we may have grouped C1, C2, C5, and C7 together based on their similar general spending patterns, but it wouldn't necessarily have highlighted the fact that their similarity is primarily in Electronics and Books. Likewise, traditional column clustering may group Electronics and Books together based on overall spending across all customers, but it wouldn't capture that this relationship is strongest for a specific subset of customers.

While the problems we consider for simulations in the current work are synthetic and simple in nature (without being non-trivial). We believe it forms the foundation for future work(s) that would ultimately be able to solve problems with photonic quantum computing like the one mentioned above at scale[8] better than classical computing.

2.6 Simulated annealing

Simulated annealing (SA) is a family of heuristics that aim to optimize a cost function by randomly sampling from the solution space of an objective function and then accepting or rejecting a new sample based on the current sample and a temperature parameter (Bertsimas and Tsitsiklis, 1993). In the context of our work, we use it as a black box optimization method with boson sampling as a subroutine for finding the best columns for a bicluster. Section 6.1 and Algorithm 1 (lines 16, 21 and 24 in particular) describe how SA is implemented in our work. For the readers who want to know more about the fundamentals of the original technique, we recommend the original paper by Kirkpatrick et al. (1983).

3 Related work

The field of biclustering has been well-studied from the perspective of classical computing. A lot of progress that has been made in creating or improving biclustering algorithms has come from the field of bioinformatics (Madeira and Oliveira, 2004; Ayadi et al., 2009; Castanho et al., 2022). While the most well-known algorithms for finding biclustering are based on spectral decomposition of matrices (Dhillon, 2001; Kluger et al., 2003), plenty of other heuristics have also been explored for this task (Hochreiter et al., 2010; José-García et al., 2022).

Previously, the application of quantum annealing to the problem of biclustering was proposed (Bottarelli et al., 2018) where the problem of finding biclusters was encoded in the quadratic unconstrained binary optimization (QUBO) form and then solved on a D-wave$^{\text{TM}}$ 2X annealer.[9] The authors performed experiments on datasets of size 100 × 50 but with a focus on a 10 × 10 moving window with biclusters upto 6 × 6 in size.

As far as photonic quantum computing is concerned, our work is the first that attempts to apply boson sampling and GBS to this problem. At the time of writing this paper, photonic quantum computing is still in its nascent stage,[10] and our primary aim is to gain preliminary insights about the relationship between photonic quantum computing and biclustering. For this, we worked with dataset of size 12 × 12 and biclusters of sizes 4 × 4 and 6 × 6.

The other work that is related to ours is the work done on using GBS for clustering (Bonaldi et al., 2023), where the authors showed promising results (on simulators) when compared to results produced by classical methods. In their work, Bonaldi et al. created a graph for GBS whose edges represented inverse distances between datapoints (vertices) and found clusters by setting the weight of the edge to 0 or 1 according to a threshold. In our work, we do not use inverse distances as a concept and have a different method for encoding the datasets into linear interferometers.

4 Boson sampling for biclustering

4.1 Introduction

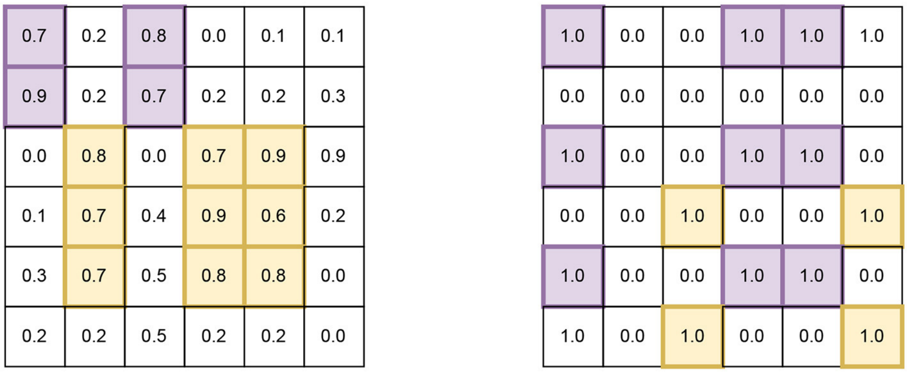
Let us consider a dataset represented by a matrix $D \in \mathbb{R}^{d_1 \times d_2}$ in which we need to find k biclusters $\mathcal{B} = \{\beta_1, \beta_2, ..., \beta_k\}$ of size[11] $b \times b$ each. Like mentioned in Section 2.5, we are looking for biclusters with large values. Our method is based on the conjecture that large

[8] Assuming the hardware scales as well.

[9] Technically, QUBO problems are first converted to their Ising model (Cipra, 1987) equivalents and then run on a quantum annealer.

[10] But with the potential of scaling up in a noise-resistant manner.

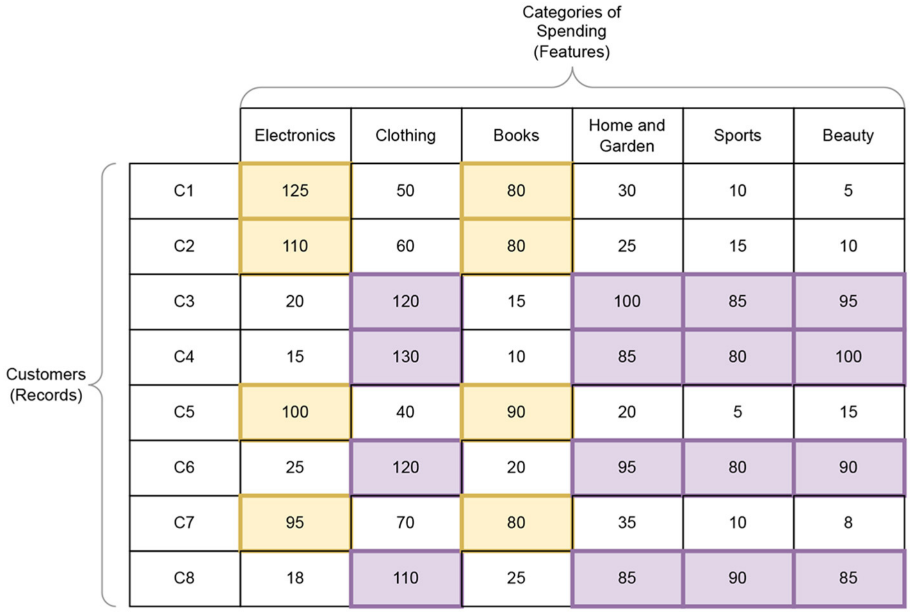
[11] We will mention how to embed rectangular matrices and search for rectangular biclusters later in this work.



0.7	0.2	0.8	0.0	0.1	0.1
0.9	0.2	0.7	0.2	0.2	0.3
0.0	0.8	0.0	0.7	0.9	0.9
0.1	0.7	0.4	0.9	0.6	0.2
0.3	0.7	0.5	0.8	0.8	0.0
0.2	0.2	0.5	0.2	0.2	0.0

1.0	0.0	0.0	1.0	1.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	1.0	1.0	0.0
0.0	0.0	1.0	0.0	0.0	1.0
1.0	0.0	0.0	1.0	1.0	0.0
1.0	0.0	1.0	0.0	0.0	1.0

FIGURE 1
Examples of square biclusters in a larger square matrix, representing a dataset. **(Left)** Is an example of a dataset that has elements in the range [0, 1] and **(Right)** is an example of a dataset that has binary elements. In each dataset, there exist two biclusters that are distingushable by different colors.



Categories of Spending (Features)
Customers (Records)

	Electronics	Clothing	Books	Home and Garden	Sports	Beauty
C1	125	50	80	30	10	5
C2	110	60	80	25	15	10
C3	20	120	15	100	85	95
C4	15	130	10	85	80	100
C5	100	40	90	20	5	15
C6	25	120	20	95	80	90
C7	95	70	80	35	10	8
C8	18	110	25	85	90	85

FIGURE 2
Example of a dataset of customers and the amount of money (in dollars in a month) they spent on an ecommerce website on different categories of products. Here we show two biclusters that shows (i) a subset of customers who spend a lot on Electronics and Books and (ii) another subset that spends on Clothing, Home and Garden, Sports and Beauty. For details, refer Section 2.5.1.

values in a bicluster would imply that the bicluster has a large permanent.

We use boson sampling as a subroutine to find out which b rows would give the highest permanent value for a given set of b columns. We do this over many different sets of columns to finally end up with a bicluster matrix that has the largest permanent. We can then set all the values of that bicluster to 0 (in the dataset) and repeat the process again for new biclusters (up until a termination criteria is met[12]).

4.2 Approach

In order to use boson sampling, we first need to embed our matrix D as a unitary operator. For simplification, we assume that all of the elements are real values scaled to the interval of $[0, 1]$. We first scale this matrix by $s = \sigma_{max}(D)$, the largest singular value of D to produce $D_s = \frac{1}{s}D$. We then embed this matrix in a greater unitary matrix by the unitary dilation theorem (Halmos, 1951; Mezher et al., 2023).

$$U_D = \begin{pmatrix} D_s & \sqrt{I_{d_1 \times d_1} - D_s(D_s)^{\dagger}} \\ \sqrt{I_{d_2 \times d_2} - (D_s)^{\dagger} D_s} & -(D_s)^{\dagger} \end{pmatrix} \tag{12}$$

[12] The termination criteria can be anything: from a maximum number of biclusters to a threshold value of the bicluster's norm.

$U_D \in \mathbb{C}^{(d_1+d_2)\times(d_1+d_2)}$ is a unitary matrix that can now be converted into a linear interferometer using efficient methods[13] (Reck et al., 1994; Clements et al., 2016). Since we know the rows and columns where D_s has been embedded, we can directly focus on those rows and columns for this computation (namely, the first d_2 modes). After the computation, we can do a procedure called post selection, where we filter out unwanted samples based on some criteria E, to then calculate the conditional probability of ψ': $P(\psi'|\psi_0, E)$. For our case, E can be denoted by

$$E = E_1 \wedge E_2 \tag{13}$$

$$\text{where } E_1 = \bigwedge_{i=1}^{d_1} (n_i' \leq \tau) \tag{14}$$

$$\text{and } E_2 = \bigwedge_{i=d_1+1}^{d_1+d_2} (n_i' = 0) \tag{15}$$

In other words, at the end of boson sampling, each of the first d_1 modes can have upto τ photons[14] but the next d_2 modes must have exactly 0 photons in them. This is because the first d_1 modes would represent the rows of D_s.

Let $\mathcal{M} = \{1, 2, ..., (d_1 + d_2) - 1, d_1 + d_2\}$ be a set of index positions from 1 to $d_1 + d_2$. $\mathcal{C}' \subset M, |C'| = b$ is a set of the column index positions of our choice (from the first d_2 index positions). Essentially, $\mathcal{C}'$ is the set of columns of a submatrix β' that serves as a candidate for a potential bicluster.

Thus, our starting state can be described as $|\psi_0\rangle = |\mathcal{N}\rangle$ where

$$\mathcal{N} = \{n_i | (n_i = 1 \wedge i \in \mathcal{C}') \vee (n_i = 0 \wedge i \in \mathcal{M} - \mathcal{C}')\} \tag{16}$$

In other words, we generate a photon in the modes specified by $\mathcal{C}'$. After boson sampling, we extract the set of selected rows $\mathcal{R}'$ from the samples of the state with the highest probability $P(\psi'|\psi_0, E)$.

Once we have $\mathcal{R}'$ and $\mathcal{C}'$, we can construct our candidate bicluster β'. We can then evaluate the quality of this candidate by a cost function $f()$ like the permanent[15] or the matrix norm of β'. Figure 3 represents a visual version of this workflow.

Repeating this process for different choices of $\mathcal{C}'$ may give us different choices of $\mathcal{R}'$. And at the end, the candidate β' with the best cost function score (largest permanent or norm) would be considered as a bicluster β with columns $\mathcal{C}$ and $\mathcal{R}$. The process of figuring out the best choice of $\mathcal{C}$ can be viewed as a blackbox optimization process. In our work, we use SA for this task.

Once a bicluster β is chosen and added to the set $\mathcal{B}$, we make the values corresponding to its rows and columns equal to 0 in D_s and repeat the entire process again. We can do this for k times (if the goal is to get k biclusters) or use some other criteria for stoppage. This heuristic is encapsulated in Algorithm 1 in detail.

```
procedure MAIN($D, num\_samples, b, k, \varepsilon, f(), E, T$)   ▷ $f()$:cost function, $k$:no. of biclusters, $\varepsilon$: termination criteria (satisfiability expression), $E$: postselection criteria, $T$: annealing temperature schedule
  Create $D_s \leftarrow \frac{1}{s}D$ where $s = \sigma_{max}(D)$
  Initialize $i \leftarrow 0$, set $\mathcal{B} \leftarrow \emptyset$ and a set $\mathcal{I} \leftarrow \emptyset$
  while $i < k$ and $\varepsilon \neq$ True do
    Get $\beta, \mathcal{C}, \mathcal{R}$ ← FINDBICLUSTER_SA($D_s, b, num\_samples, f(), E, T$)
    Add bicluster $\beta$ to the set of all biclusters, $\mathcal{B} \leftarrow \mathcal{B} \cup \beta$
    Append $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{R} \times \mathcal{C}$   ▷ Here, ×: cartesian product
    Set all elements in $D_s$ to 0 for positions described in $\mathcal{R} \times \mathcal{C}$
    $i \leftarrow i + 1$
  end while
  return $\mathcal{B}, \mathcal{I}$
end procedure
procedure FINDBICLUSTER_SA($D_s, b, num\_samples, f(), E, T$)
  Randomly select $b$ columns and create a set $\mathcal{C}'$. Intialize $cost \leftarrow 0$
  repeat
    Get next $t$ from annealing schedule $T$
    Get $\mathcal{R}' \leftarrow$ GETROWS($D_s, b, num\_samples, \mathcal{C}', E$)
    Create candidate bicluster $\beta'$ by taking elements from $D_s$ indexed by $\mathcal{R}' \times \mathcal{C}'$
    Evaluate $cost' \leftarrow f(\beta')$
    $\Delta cost \leftarrow cost' - cost$
    if $\Delta cost > 0$ or $U(0,1) < exp(\frac{\Delta cost}{t})$ then
      $\beta \leftarrow \beta'$, $cost \leftarrow cost'$, $\mathcal{C} \leftarrow \mathcal{C}'$ and $\mathcal{R} \leftarrow \mathcal{R}'$
    end if
    Generate a new $\mathcal{C}'$ based on the neighborhood of $\mathcal{C}$
  until all $t$s in $T$ are covered
  return $\beta, \mathcal{C}, \mathcal{R}$
end procedure
procedure GETROWS($D_s, b, num\_samples, \mathcal{C}', E$)
  Create unitary matrix $U_D$ using $D_s$ by Equation 12
  Prepare state $|\psi_0\rangle = |\mathcal{N}\rangle$ by using $\mathcal{C}'$ and Equation 16
  Do boson sampling for $num\_samples$ times
  Get the state $|\psi'\rangle$ with the highest observed $P(\psi'|\psi_0, E)$ where E is defined by Equation 13
  Extract the set of rows $\mathcal{R}'$ from $|\psi'\rangle$
  return $\mathcal{R}'$
end procedure
```

Algorithm 1. Boson sampling for finding biclusters of size $b \times b$ (simulated annealing for col. selection).

[13] This is unlike the challenge of decomposing unitary matrices into quantum gates for the gate-model.

[14] Traditionally, $\tau = 1$ in some versions of boson sampling.

[15] Since calculating the permanent is #P-Hard, the use of boson sampling (Mezher et al., 2023) is suggested for this task as well.

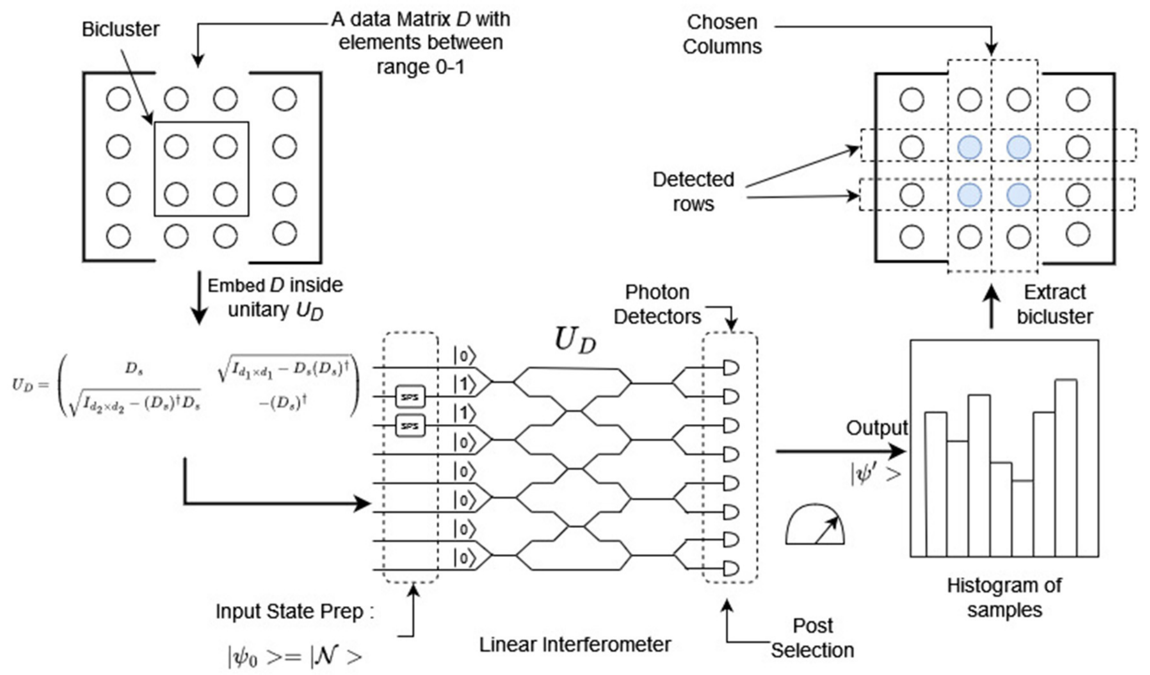

Bicluster
A data Matrix D with elements between range 0-1
Chosen Columns
Detected rows
Embed D inside unitary U_D
$U_D = \begin{pmatrix} D_s & \sqrt{I_{d_1\times d_1} - D_s(D_s)^\dagger} \\ \sqrt{I_{d_2\times d_2} - (D_s)^\dagger D_s} & -(D_s)^\dagger \end{pmatrix}$
Photon Detectors
Extract bicluster
U_D
SPS
SPS
$|0\rangle$
$|1\rangle$
$|1\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
Output
$|\psi' >$
Histogram of samples
Input State Prep :
$|\psi_0 >= |\mathcal{N} >$
Linear Interferometer
Post Selection

FIGURE 3
Workflow of the boson sampling approach for biclustering. Here, the user chooses the columns and boson sampling returns corresponding rows; from which the candidate bicluster is constructed. By Equation 8, it is expected that the bicluster that has the highest permanent value would also have the highest probability of being obtained (for the initial choice of columns). This figure is for illustrative purposes only.

4.2.1 An example of a linear optical circuit for boson sampling

Here we will present an example of a linear optical circuit as it relates to boson sampling (and the application of biclustering). While this example cannot go into all the details of such a circuit, we hope that it is still informative for the readers of this paper. Consider the following non-unitary matrix

$$D = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \tag{17}$$

While such a small matrix is trivial for an actual biclustering problem (or for boson sampling), the focus of this section is to show how an arbitrary matrix D gets translated into a optical circuit. Using Equation 12 we embed D into the unitary matrix U, which in this case is:

$$U = \begin{pmatrix} 0.38 & 0.45 & 0.65 & -0.47 \\ 0.53 & 0.60 & -0.48 & 0.35 \\ 0.57 & -0.49 & -0.38 & -0.53 \\ -0.49 & 0.42 & -0.45 & -0.61 \end{pmatrix} \tag{18}$$

$$\text{and } s = 13.19 \tag{19}$$

Like mentioned before, unlike gate-model quantum computers, it is fairly efficent to convert an arbitrary unitary matrix into a linear optical circuit (Reck et al., 1994). The resultant circuit is known as a Reck's type circuit and has been implemented for Python in packages such as `Perceval`. The implementation typically involves using R_X beamsplitters with phase shifters $e^{i\phi}$ where

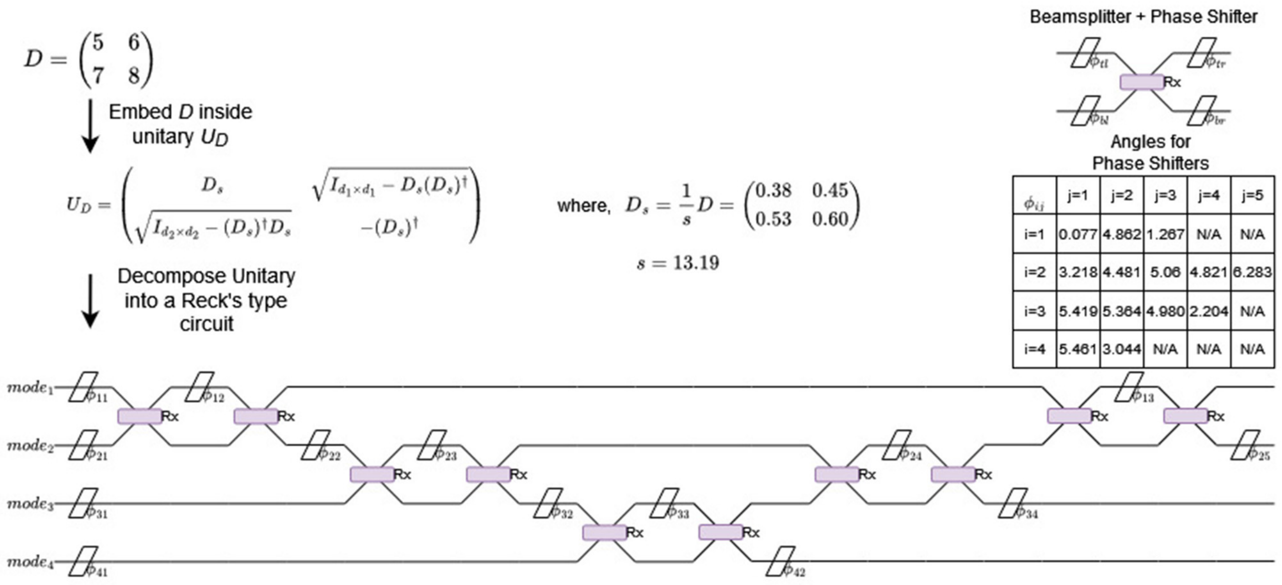
$$R_X = \begin{pmatrix} exp(i(\phi_{tl} + \phi_{tr}))\cos\theta/2 & iexp(i(\phi_{bl} + \phi_{tr}))\sin\theta/2 \\ iexp(i(\phi_{tl} + \phi_{br}))\sin\theta/2 & exp(i(\phi_{bl} + \phi_{br}))\cos\theta/2 \end{pmatrix} \tag{20}$$

with $\theta = \pi/2$, Each R_X operation acts on two modes (depicted in Figure 4 as the "X" intersection with a rectangle overlay). The phaseshift operations are depicted as diagonal bars and can be included inside the nearest R_X matrix by proximity. Here, ϕ_{tl},ϕ_{tr},ϕ_{bl} and ϕ_{br} are the top-left, top-right, bottom-left and bottom-right phase shift angles with respect to a R_X operation (see "Beamsplitter + Phase Shifter" in Figure 4). For further details we recommend the paper on `Perceval` package by the team at Quandela$^{\text{TM}}$ (Heurtel et al., 2023). For the rest of the paper, we will assume that if you have an arbitrary dataset matrix D, it is possible to convert it into a linear optical circuit for boson sampling.

4.3 Dealing with rectangular biclusters

While this work is primarily concerned with square biclusters, rectangular biclusters can also be preprocessed to be accommodated in this scheme. This step would essentially involve padding the matrix with columns of 1s.

We begin here with the precursor to matrix D; the matrix $D' \in \mathbb{R}^{d_1' \times d_2'}$ with each value in the interval $[0, 1]$. We will describe a scheme that would allow us to find a bicluster $\beta \in \mathbb{R}^{b_1 \times b_2}$ with $b_1 > b_2$. This preprocessing method is restricted to finding biclusters where number of columns are fewer than the number of


$D = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$
Embed *D* inside unitary U_D
$U_D = \begin{pmatrix} D_s & \sqrt{I_{d_1 \times d_1} - D_s(D_s)^\dagger} \\ \sqrt{I_{d_2 \times d_2} - (D_s)^\dagger D_s} & -(D_s)^\dagger \end{pmatrix}$
where, $D_s = \frac{1}{s} D = \begin{pmatrix} 0.38 & 0.45 \\ 0.53 & 0.60 \end{pmatrix}$
$s = 13.19$
Decompose Unitary into a Reck's type circuit
Beamsplitter + Phase Shifter
ϕ_{tl}
ϕ_{tr}
ϕ_{bl}
ϕ_{br}
Rx
Angles for Phase Shifters
$mode_1$
$mode_2$
$mode_3$
$mode_4$
ϕ_{11}
ϕ_{12}
ϕ_{13}
ϕ_{21}
ϕ_{22}
ϕ_{23}
ϕ_{24}
ϕ_{25}
ϕ_{31}
ϕ_{32}
ϕ_{33}
ϕ_{34}
ϕ_{41}
ϕ_{42}

ϕ_{ij}	j=1	j=2	j=3	j=4	j=5
i=1	0.077	4.862	1.267	N/A	N/A
i=2	3.218	4.481	5.06	4.821	6.283
i=3	5.419	5.364	4.980	2.204	N/A
i=4	5.461	3.044	N/A	N/A	N/A

FIGURE 4
Example of a matrix *D* being embedded inside a unitary matrix U_D that is converted into a linear optical circuit using the methodology in Reck's seminal work (Reck et al., 1994) (and implemented in `Perceval`). Each line indicates a mode (not a qubit), the diagonal bars indicates phase shifters and the "X" intersection of modes are beamsplitters. The table denotes the phase shift angles for the circuit (ϕ_{ij} is the *i*th mode's *jth* phase shift angle).

rows.[16] So if our desired rectangular bicluster has $b_2 > b_1$, then we would need to start with $(D')^T$ such that we would be searching for $\beta^T \in \mathbb{R}^{b_2 \times b_1}$.

The general approach is to pad matrix D' with Δb columns[17] of all 1s to create matrix D. This is done to have "anchored" columns $\mathcal{C}_{anchor}, |\mathcal{C}_{anchor}| = \Delta b$ where we send 1 photon each in their corresponding modes. The rest of the columns $\mathcal{C}_{choice}, |\mathcal{C}_{choice}| = b_2$ would be the actual choices for the columns of our bicluster (picked from the first d_2 modes). Together they would make $\mathcal{C}'$ (where $\mathcal{C}' \leftarrow \mathcal{C}_{choice} \cup \mathcal{C}_{anchor}$)

By sending photons through modes of "anchored" columns, we can make them available for row selection (after the computation) while still keeping track of a smaller number of columns for our actual bicluster. Since the anchored columns all have 1s, they would have a constant effect on the value of the permanent (even after scaling down by s) and can be ignored in the final creation of the bicluster. Algorithm 2 describes this preprocessing process in detail.

```
procedure MAIN($D', d_1', d_2', b_1, b_2$)
    Initialize variable $transpose\_flag = $ False
    if $b_2 > b_1$ then
        Take its transpose $D' \leftarrow (D')^T$
        Swap values in $b_1$ and $b_2$
        Set $transpose\_flag \leftarrow$ True ▷ This lets users know if they should transpose their biclusters
    end if
    Compute $\Delta b \leftarrow b_1 - b_2$
    Create matrix $A_1$ of all 1s of size $d_1 \times \Delta b$, Create $D \leftarrow \begin{pmatrix} D' & A_1 \end{pmatrix}$, where $D \in \mathbb{R}^{d_1 \times d_2}$ with $d_1 \leftarrow d_1'$ and $d_2 \leftarrow d_2' + \Delta b$
    return $D, d_1, d_2, transpose\_flag$
end procedure
```

Algorithm 2. Preprocessing data for finding rectangular biclusters (boson sampling).

5 Gaussian boson sampling for biclustering

The process of using GBS for the task of biclustering is fairly straightforward. Additional preprocessing and post processing can be added to this method to make it more efficient, but we are going to focus on the basics of this application in our work.

Here, we will first convert our dataset $D \in \mathbb{R}^{d_1 \times d_2}$ (assumed to have values in the range $[0, 1]$) into an adjacency matrix $D_{adj} \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$ by the following transformation

$$D_{adj} = \begin{pmatrix} 0 & D \\ D^T & 0 \end{pmatrix} \tag{21}$$

This is done to prepare the matrix for the Autonne-Takagi decomposition[18] from which we will get the unitary matrix $U_{D_{adj}}$

$$D_{adj} = U_{D_{adj}} \text{diag}(\lambda_1, \lambda_2, ..., \lambda_{d_1+d_2}) U_{D_{adj}}^T \tag{22}$$

[16] D' can have dimensions where $d_1 = d_2$ or $d2 > d1$ or $d1 > d2$.

[17] $\Delta b = |b_1 - b_2|$.

[18] This process only works on symmetric matrices and D_{adj} is one.

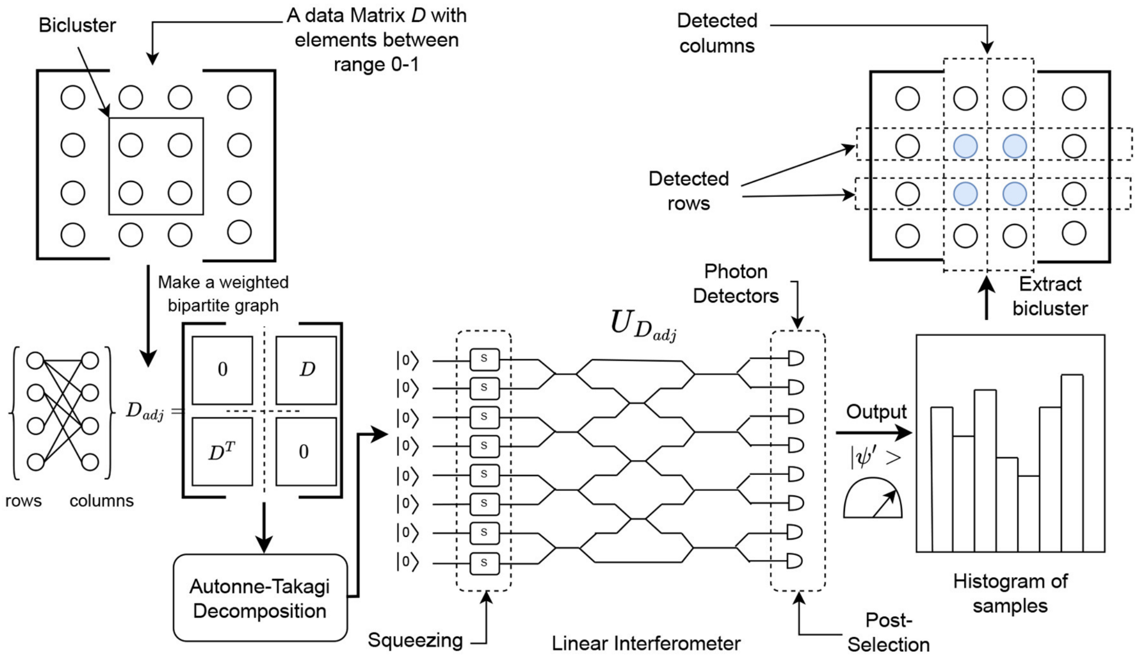


FIGURE 5
Workflow of the Gaussian boson sampling (GBS) approach for biclustering. Here, GBS returns the rows as well as the columns of potential biclusters (ideally with large hafnian or torontonian values). This figure is for illustrative purposes only.

The λ_is are used in the calculation of the squeezed states.[19] Here it is assumed that the matrix D_{adj} is scaled with the parameter $c > 0$ to make sure that $0 \leq \lambda_i \leq 1$. Together c and $\lambda_i s$ are used in calculating the squeezing parameters $r_i = \tanh^{-1}(c\lambda_i)$. They are also related to the mean number of photons to be generated by the following equation

$$\overline{n} = \sum_{i=1}^{d_1+d_2} \frac{(c\lambda_i)^2}{1-(c\lambda_i)^2} \tag{23}$$

The unitary matrix $U_{D_{adj}}$ would then be converted to a linear interferometer and the squeezing parameters r_i would be applied to their respective modes. After the computation, the columns $\mathcal{C}'$ and rows $\mathcal{R}'$ can be extracted from their corresponding modes in $|\psi'\rangle$ where one or more photons were detected.

Each GBS sample gives back a candidate bicluster β'. These candidate biclusters can be evaluated by a function $f()$ such as a matrix norm, based on which (and a threshold value) it can be accepted or rejected. The values in D corresponding to the accepted bicluster are then set to zero.[20] The process can then repeat k times or until a termination criteria ε is reached.

Finding biclusters can be represented as the process of locating dense sub-graphs in a bipartite graph (Karim et al., 2019). Searching for N-dense subgraphs using GBS is a topic that is being actively explored (Arrazola and Bromley, 2018; Solomons et al., 2023). The major restriction is that the sub-graphs (or biclusters) that GBS finds will have a even numbered dimension (N is even). In order to find biclusters for an odd numbered N, and for biclusters where specific dimensions $b_1 \times b_2$ are needed, you would need to post-process the results.[21] Figure 5 illustrates a general idea of how GBS can be used to find biclusters and Algorithm 3 details this heuristic more formally.

6 Simulations and results

In order to do preliminary validation of our ideas, we performed a few simulations to get an initial estimate of our proposed approaches. All our simulations were done on classical computers with most of them being done on a high performance server with a `AMD Ryzen Threadripper 3970X 32-Core` processor with a memory size of ∼`256.7 GB`. The programs were written in the Python programming language and the simulation packages used were `Perceval` for boson sampling (Heurtel et al., 2023) and `Strawberry Fields` for GBS (Killoran et al., 2019; Bromley et al., 2020). All our simulations were noiseless.

The method for simulation that was used for boson sampling (implemented in `Perceval`) is Clifford and Clifford algorithm (Clifford and Clifford, 2018). This is an exact algorithm which is one of the fastest methods to simulate boson sampling classically, the time complexity for which is exponential in nature at $O(n2^n + mn^2)$ (where m is the number of modes and n are the number of photons). For Gaussian boson sampling, although the exact method for its simulation is exponential at $O(mN^3 2^N)$

[19] Here, λs are not to be taken as the eigenvalues of D_{adj}.

[20] In this work, we only deal with non-overlapping biclusters. For getting overlapping clusters, we would need a more refined approach that includes (i) not setting the values to 0 after extracting a bicluster and (ii) a larger number of samples.

[21] Since in GBS, both the rows and columns are observed after the computation.

```
procedure MAIN($D, num\_samples, \varepsilon, \overline{n}, k, f()$)
    Initialize $\mathcal{B} \leftarrow \emptyset$, $\mathcal{I} \leftarrow \emptyset$ and $i \leftarrow 0$
    while $i < k$ and $\varepsilon \neq$ True do
        Create $D_{adj} \leftarrow \begin{pmatrix} 0 & D \\ D^T & 0 \end{pmatrix}$
        Figure out scaling parameter $c$ based on $D_{adj}$ and $\overline{n}$
        Scale the matrix $D_{adj} \leftarrow cD_{adj}$
        Do Autonne-Takagi decomposition to get $U_{D_{adj}}$ and $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_{(d_1+d_2)}\}$
        Get a set of squeezing parameters $R = \{r_1, r_2, ..., r_{(d_1+d_2)}\}$ by $r_i \leftarrow \tanh^{-1}(c\lambda_i), \forall r_i \in R$
        Do GBS using $U_{D_{adj}}$, $R$ to get $num\_samples$ number of samples
        while samples remain to be processed do
            Get the candidate rows $\mathcal{R}'$ and columns $\mathcal{C}'$ from the current sample.
            Get the corresponding bicluster candidate $\beta'$ and
            if $\beta'$ has a good value of $f()$ then
                $\beta \leftarrow \beta'$, $\mathcal{R} \leftarrow \mathcal{R}'$ and $\mathcal{C} \leftarrow \mathcal{C}'$
                Store it in $\mathcal{B} \leftarrow \mathcal{B} \cup \beta$ and append $\mathcal{I} \leftarrow \mathcal{R} \times \mathcal{C}$
                Set all elements in $D$ to 0 for positions described in $\mathcal{R} \times \mathcal{C}$
                $i \leftarrow i + 1$
            end if
        end while
    end while
    return $\mathcal{B}, \mathcal{I}$
end procedure
```

Algorithm 3. Gaussian boson sampling for finding biclusters (non-overlapping).

(N is the number of total photons), non-negative matrices can be approximately simulated in polynomial time with the time complexity of $O(mMN^3)$ (M being the number of approximate matrices used for approximation). We recommend the work of Quesada and Arrazola for readers interested in more information (Quesada and Arrazola, 2020).

We also wanted to experiment on real photonic quantum processing units (QPU) but at the time of the project, there were no publicly available QPUs that could handle problems of a large enough size. With the preliminary simulations, our aim was to select problems that are not intractable to simulate,[22] but are also non-trivial in nature.

We took a total of four problems : two for boson sampling and two for GBS with 12×12 sized datasets. The reason for choosing the above dimensions was to have a matrix in which biclusters of non-trivial sizes[23] could be embedded in. In other words, like mentioned above, because this is the first work of its kind, the chosen problems were designed to have a balance of simplicity and non-triviality. Another thing to note is that our simulations were based on a simplified version of the proposed heuristics in Algorithms 1, 3. This was done (i) partially due to the nature of these problems being simple and straightforward but also because of (ii) the simulation times involved. The exact details are mentioned in the following sections. We hope that the insights from these simulations would be useful for future implementation on real photonic hardware.

6.1 Boson sampling-problem 1

6.1.1 Setup

In our first simulation, we are going to have the following assumptions:

1. There is only one bicluster β in our dataset D.
2. We know the set of columns $\mathcal{C}$ for the β.

Our task here is to use boson sampling with input created from $\mathcal{C}$ to calculate the probability of observing the correct set of rows $\mathcal{R}$. For this, we created a 6×6 matrix that will act as our bicluster β that has values randomly sampled from $\{0.7, 0.8, 0.9\}$. Then we created $D^{(1)}, D^{(2)}, ..., D^{(5)}$ datasets of size 12×12 where we embed β at $\mathcal{C}, \mathcal{R} = \{4, 5, 6, 7, 8, 9\}$. All other elements in the datasets have values from 0 to $0.1 \times \alpha$:

$$\underset{i \notin \mathcal{R}, j \notin \mathcal{C}}{D_{ij}^{(\alpha)}} \in \{(0.1)h | 0 \leq h \leq \alpha, h \in \mathbb{Z}\} \tag{24}$$

We created one more matrix $D^{(6)}$ in which we embed a 6×6 bicluster of all 1s (for the same $\mathcal{R}$ and $\mathcal{C}$). The rest of the values are all 0s. This is done to study boson sampling's performance for binary matrices, albeit a very simple one in this case.

For each dataset and our fixed $\mathcal{C}$ we performed boson sampling with 10^5 and 10^6 samples. We then analyzed the success probability $\mathcal{P}$ of getting our ideal bicluster for three different conditions:

1. No postselection (raw results)
2. Postselection as defined by Equation 13 with $\tau = 1$.
3. Postselection as defined by Equation 13 with $\tau = 3$.

For points 1 and 2 in the list above, in order to estimate $\mathcal{P}$, a success is to observe if the modes corresponding to $\mathcal{R}$ receive 1 photon each. Here, the success probability $\mathcal{P}$ is based on finding (single) photons in all rows described in $\mathcal{R}$ and no others. For #3, we are fine with up to three photons in the same mode, but we only count each unique row once. Since the number of photons are limited, in this case, success is defined as finding photons in a subset of the rows $\mathcal{R}$ of the ideal bicluster, but no others.

6.1.2 Results and discussion

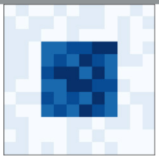
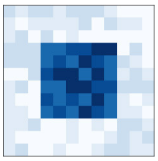
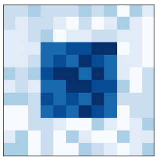
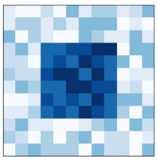
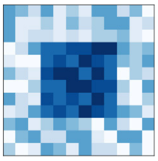
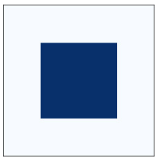
The results of the simulations on our first problem can be best described by Table 1. This table shows us the results for cases where 1 million samples were taken. It should be noted that our results for 100k samples were comparable, but we chose to publish the former since the calculated probabilities would be closer to their exact values (due to the larger sample size).

The first observation we can make is the effect that the values in $(D_{ij}^{(\alpha)})_{i \notin \mathcal{R}, j \notin \mathcal{C}}$ can have on finding the bicluster β. As the values in the other parts of the dataset go up (even if its lower than the values

[22] While not being impossible, the simulation process was still slow for the resources we had.

[23] 4×4 and 6×6.

TABLE 1 Success probabilities $\mathcal{P}$ for boson sampling-problem 1 of datasets $D^{(1)}$ to $D^{(6)}$, when calculated for 1 million samples.

Dataset	Heatmap	No-postselect			Postselect $\tau = 1$			Postselect $\tau = 3$		
		Num.	Den.	$\mathcal{P}$	Num.	Den.	$\mathcal{P}$	Num.	Den.	$\mathcal{P}$
$D^{(1)}$		219	10^6	2.1×10^{-4}	219	245	0.893	13, 269	13, 593	0.976
$D^{(2)}$		185	10^6	1.8×10^{-4}	185	262	0.706	11, 323	12, 289	0.921
$D^{(3)}$		152	10^6	1.5×10^{-4}	152	366	0.415	9, 101	11, 277	0.807
$D^{(4)}$		101	10^6	1×10^{-4}	101	514	0.196	5, 909	9, 130	0.647
$D^{(5)}$		49	10^6	4.9×10^{-5}	49	559	0.087	3, 007	6, 618	0.454
$D^{(6)}$		255	10^6	2.5×10^{-4}	255	255	1.0	14, 686	14, 686	1.0

The heatmaps depicts the areas of the table where the values are the highest (dark blue) and lowest (white or very light blue). For each of the three settings (no-postselection, postselection with $\tau = 1$, and postselection with $\tau = 3$), the numerator (no. of samples containing the ideal result), denominator (no. of samples filtered through the postselection criteria E), and $\mathcal{P}$ (numerator/denominator) is listed.

in the rows of β), the probability of those rows to be sampled also goes up (even if the probability of finding the rows of β may still be the highest). For example, for the data of postselect $\tau = 1$, we can see how the results degrade when going from 0.89 for $D^{(1)}$ to 0.087 for $D^{(5)}$. This seems to suggest that such a technique would do better if there is enough difference in the magnitude of the values of the bicluster and the rest of the dataset. This is similar to how in quantum annealing, finding the ground state solutions becomes worse when there are plenty of other good solutions in the landscape (in other words, when a landscape is not "rugged") (King et al., 2019). Similarly, the equivalent of a lot of ruggedness in boson sampling terms would seem to be a larger difference in the magnitude of β and $(D^{(\alpha)}_{ij})_{i \notin \mathcal{R}, j \notin \mathcal{C}}$.

The other important observation to make is the role postselection plays for this task. In the million samples, the majority of them have photons that may be (i) incident on modes $> d_1$ and/or (ii) aggregate in modes in quantities > 1. After applying our strictest postselection criteria (postselect $\tau = 1$ column), the samples that remain are only in the hundreds. The situation is somewhat better for our other postselection criteria (postselec $\tau = 3$) where the samples that remain are in the thousands. Regardless, based on these simulations, it would appear that in order to even conduct meaningful postselection, we need a very large number of samples. Depending on the hardware equipment being used, the time taken to get one sample in a linear interferometer for boson sampling and GBS can be very fast (Madsen et al., 2022; Deng et al., 2023b).

Like mentioned before, the success probability for when postselection is done with $\tau = 3$ is altered to be the probability of samples where one or more photons are received on a mode in $\mathcal{R}$.[24] Therefore, not only are we working with a higher denominator when $\tau = 3$, but also a comparitively high numerator value. And while $\mathcal{P}$ does decrease from $D^{(1)}$ to $D^{(5)}$, it is higher than its $\tau = 1$ counterpart. Essentially it indicates that, the traditional way to run boson sampling (with single photons in the input mode and a requirement of only allowing one photon in a mode at the output) may not be best suited for practical applications. Indeed,

[24] And also like mentioned before, if we were to extract rows from such a sample (to construct β'), we would only take the unique ones.

along with the lower probability, this stringent approach will face issues with noise once we would try and implement it on an actual photonic device (Brod et al., 2019) (photon loss being a major concern among them).

Finally, the results for $D^{(6)}$ shows us how boson sampling may behave for a bicluster of all 1s with all other parts of the dataset containing zeros. Here we can see that the postselected $\mathcal{P}$ is observed to be 1. The success probabilities of a realistic binary dataset however, may not be expected to be so high as there would most likely be presence of 1s outside of the actual biclusters. While typical binary datasets would probably not have such a simple spread of values across the entire matrix, the value of our simple simulation is to show how extreme contrast in values can affect the probabilities.

6.2 Boson sampling-problem 2

6.2.1 Setup

For our second simulation, we use dataset $D^{(2)}$ from the first problem[25] but shuffle its rows and columns randomly to make it into a slightly more challenging problem. The objective is to find a $\beta \in \mathbb{R}^{6\times6}$ where we do not know its $\mathcal{R}$ or $\mathcal{C}$. Here, our assumption is that there is one and only one 6 × 6 bicluster in the dataset.

In order to do this, we use Algorithm 1 with the following parameters:

1. $D = P_R(D^{(2)}P_C)$ where P_R and P_C are the randomly generated row and column permutation matrices, respectively.
2. $num\_samples = 10^5, b = 6$ and $k = 1$
3. $\varepsilon = \emptyset$, essentially we don't have any separate termination critera.
4. $f()$ is the permanent
5. T is an exponential decay annealing schedule: by $T = \{t_i | t_i = t_0\left(\frac{t_f}{t_0}\right)^{i/p}, 0 \le i \le b-1, i \in \mathbb{Z}\}$ with $t_0 = 100$ and $t_f = 0.01$

In each iteration of the SA process, the rows are chosen from the sample that appears the most after postselection. We use the postselection criteria E as mentioned in Equation 13 with initial $\tau = 1$. If we do not find any samples that satisfy this criteria, we iteratively increase τ upto $b-1$. If no samples are selected even after this, we assign $\mathcal{R}' \leftarrow \emptyset$ and $cost' = -1000$. This means that the selection of $\mathcal{C}'$ was very bad and should not be accepted. For samples where $\tau > 1$, it means that $|\mathcal{R}'| \le 6$. For these simulations, whenever $|\mathcal{R}'| < 6$, we chose to reduce the $|\mathcal{C}'|$ to match $|\mathcal{R}'|$ by dropping the columns in $\mathcal{C}'$ with the lowest individual L2 norms,[26] which will give us β', our bicluster candidate. Essentially, for these simulations on problem 2, our bicluster can potentially be smaller than 6 × 6 (but not rectangular) if a bicluster of the original size cannot be found.

For SA, the neighborhood of $\mathcal{C}$ is defined as any column's index position in D outside of $\mathcal{C}$ which can be swapped for a index position inside $\mathcal{C}$. In other words, the process of generating new candidate $\mathcal{C}'$ is as follows:

TABLE 2 Approximate success probabilities $\mathcal{P}$ for problem 2 of boson sampling for corresponding number of steps that simulated annealing (SA) was run for (boson sampling being used as a subroutine).

Anneal steps p	20	50	100	150	200
Success probability $\mathcal{P}$	0.17	0.34	0.85	0.95	0.99

1. Copy $\mathcal{C}' \leftarrow \mathcal{C}$
2. Randomly select a column index position i of D where $i \in \mathcal{C}'$
3. Randomly select a column index position j of D where $j \notin \mathcal{C}'$
4. Delete i from $\mathcal{C}' : \mathcal{C}' \leftarrow \mathcal{C}' - \{i\}$
5. Add j to $\mathcal{C}'$: $\mathcal{C}' \leftarrow \mathcal{C}' \cup j$

We ran our simulations for p steps that span over the entire annealing schedule T for $p \in \{20, 50, 100, 150, 200\}$ for a hundred trials per p (500 total). Each SA step only involves one Monte Carlo sweep (i.e., only one iteration per temperature value in T).

6.2.2 Results and discussion

The results for the simulation of problem 2 as encapsulated by Table 2 do show a favorable scaling of finding a square bicluster for the number of SA steps p involved. Of course, since we have just considered one problem over here, a more thorough study in the future is warranted.

One of the most glaring challenges is the number of samples that we had to do for this process to work. As mentioned previously, we ran boson sampling for 10^5 times per each anneal step. This also constrained us in limiting the number of times we run the simulation per p. However, it should also be kept in mind that actual devices are going to be way faster than the simulation work we did. A different black box technique to SA could also be considered.

6.3 Gaussian boson sampling-problem 1

6.3.1 Setup

Since the approach of finding biclusters using GBS yields us both the rows and columns of the bicluster simultaneously, for our first experiment, we consider the same dataset D as the one for boson sampling-problem 2 (see Figure 6). We also consider a binary version D_{bin} that makes all values ≥ 0.7 from the original equal to one and zero if otherwise. Our goal is to find the same 6×6 bicluster as the one in Section 6.2.

In GBS, because we cannot control the exact number of photons we send through the circuit, the mean number of photons $\overline{n}$ per mode (or just $\overline{n}$ per mode) is an important hyperparameter for the GBS process. As we can see from Equation 11, having either too low or too high a value of $\overline{n}$ per mode may affect the results in a negative way.[27]

Our simulations were done with 1, 2, 4, 6, and 10 mean photons per mode. We took 10^4 samples for each setting of $\overline{n}$ per mode. The primary reason for taking fewer samples than boson sampling is that the GBS approach seems to require a smaller

[25] $D^{(2)}$ was chosen over the others for a decent contrast of values in the dataset (useful for a good $\mathcal{P}$) while still being challenging enough for boson sampling.

[26] Higher elements in a column would correlate with higher (vector) norm values.

[27] Too low : photons may not be generated and/or lost to photon loss. Too high: the probability of finding good biclusters will go down factorially.

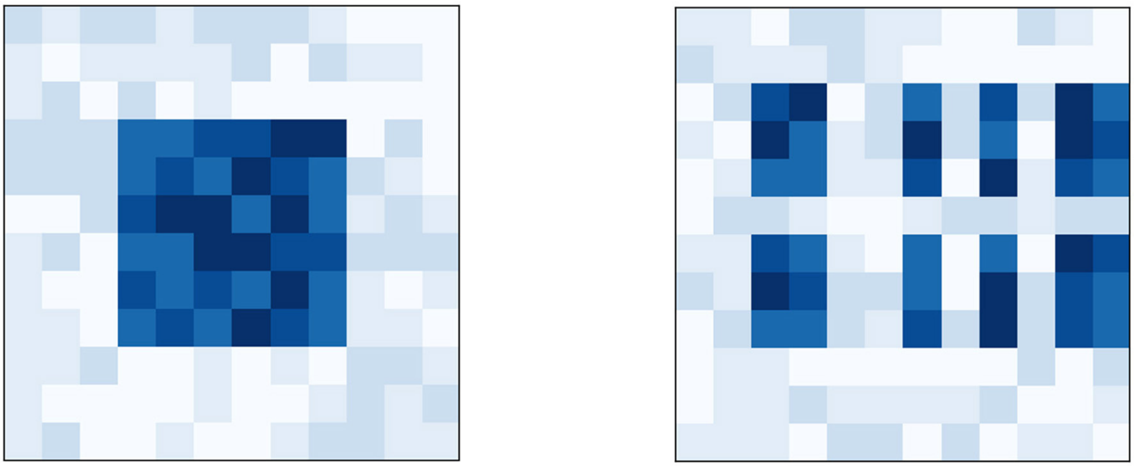


FIGURE 6
(Right) Heatmap of the dataset used in boson sampling-problem 2. This problem was generated by taking $D^{(2)}$ from boson sampling-problem 1 whose heatmap is on the **(Left)** and then performing a random permutation on its rows and columns.

TABLE 3 Approximate success probabilities $\mathcal{P}$ for problem 1 of Gaussian boson sampling (GBS) for corresponding mean number of photons ($\overline{n}$) per mode.

$\overline{n}$ per mode	1	2	4	6	10
No. of correct samples	735	1,038	1,099	945	701
Success probability $\mathcal{P}$	0.0735	0.1038	0.1099	0.0945	0.0701

This is for the real valued dataset D.

TABLE 4 Approximate success probabilities $\mathcal{P}$ for problem 1 of Gaussian boson sampling (GBS) for corresponding mean number of photons ($\overline{n}$) per mode.

$\overline{n}$ per mode	1	2	4	6	10
No. of correct samples	1,949	4,125	6,180	7,305	8,234
Success probability $\mathcal{P}$	0.1949	0.4125	0.6180	0.7305	0.8234

This is for the binary dataset D_{bin}.

number of samples for other applications (Arrazola and Bromley, 2018; Bonaldi et al., 2023). The secondary reason is that the cost of simulating a GBS process is far larger for a 24-mode linear interfermetor than for a boson sampling circuit of the same size. This is because a classical process has to keep track of squeezed light states across a large number of modes in GBS.

After the process, photons in the first d_1 modes would represent the rows that have been selected for a candidate bicluster and the next d_2 modes would represent the selected columns for the same. For measurement, we used the threshold detection process as implemented in the `strawberryfields` package. So even if we had more than one photon in a mode, we would still count that row/column only once. Finally, once we decipher the proposed biclusters from the samples, we then compare them against the correct result in order to calculate the success probability $\mathcal{P}$.

6.3.2 Results and discussion

From Tables 3, 4, the very first thing the data seems to suggest, is that GBS, like boson sampling, seems more effective at the task of finding binary biclusters. But we don't need success probabilities $\geq 2/3$ in order for GBS to be useful. As long as we have samples from which the part or whole of a bicluster can be extracted,[28] there may be some utility of GBS for this application. Another thing to note is that the success probability of a random sampling procedure to find the correct bicluster is in the order of.[29] 10^{-6}, which is lower than the success probabilities from GBS. It should also be mentioned that except in one case (real valued dataset, $\overline{n}$ per mode $= 1$), the correct samples were in the majority (i.e., the statistical mode).

The other important observations with GBS are that (i) we figure the rows and columns of the bicluster simultaneously and (ii) we may be able to get meaningful results with fewer number of samples when compared to our approach with boson sampling. Of course, further studies and a thorough comparison with classical methods is needed to make stronger claims.

The final observation for this problem is how the different datasets performed for different values of $\overline{n}$ per mode. For D, 4 mean photons per mode gave us the best success probability and for D_{bin} it was 10 (from the limited information we have). Regardless, the initial data indicates that the best value for $\overline{n}$ per mode could be different from problem to problem.

[28] This may be done by evaluating the candidate bicluster of each sample using a cost metric $f()$ like a matrix norm, and selecting the bicluster with the highest cost value.

[29] There are a total of 24 modes. For simplifying the calculation, consider the values the modes can take to be either 0 or 1 (photons are either there or not). Total number of possible solutions (for a 6×6 bicluster): $\binom{26}{6} = 134596$ and probability of sampling the best solution (assuming there is only one) becomes $1/134,596 \approx 7.43 \times 10^{-6}$.

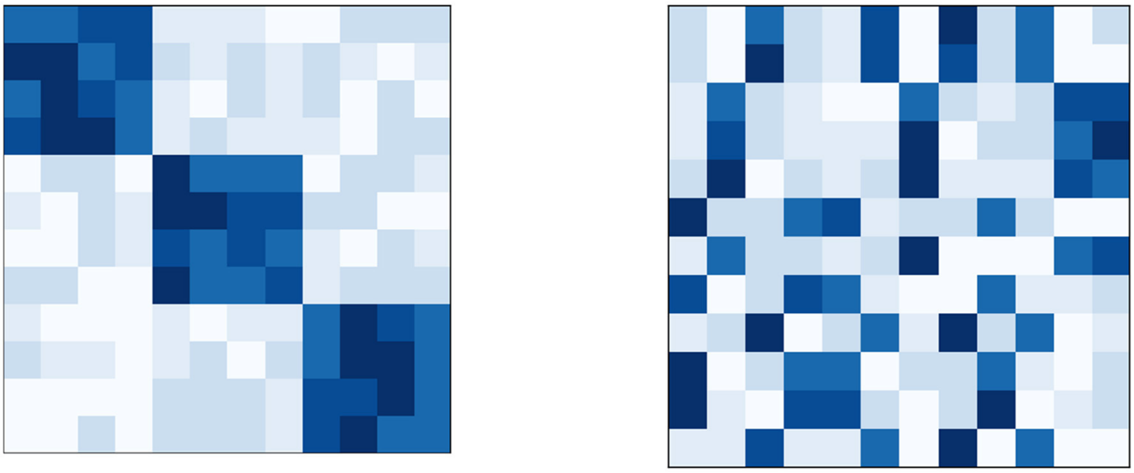


FIGURE 7
(Right) Heatmap of the dataset with three biclusters used in GBS-problem 2. This problem was generated by first creating the dataset whose heatmap is on the **(Left)** and then performing a random permutation on its rows and columns. For further details, refer Section 6.4.1.

6.4 Gaussian boson sampling-problem 2

6.4.1 Setup

For the second simulation of GBS, we chose a problem that has three biclusters $\beta_1, \beta_2, \beta_3 \in \mathbb{R}^{4\times4}$ that has values in the interval $[0.7, 0.9]$. These three biclusters are placed along the diagonal of a 12×12 dataset D, the rest of which has values ≤ 0.2. After this, a random permutation was applied on the rows and columns of the dataset to make the problem more challenging and realistic.[30] Figure 7, shows the heatmaps of the dataset during its initial and final creation phases (Left and Right parts of the figure, respectively). Finally a binary dataset D_{bin} was created by the same criterion as mentioned in Section 6.3.1.

The aim of this simulation was to see how effective GBS would be for detecting multiple biclusters. We chose 2 as the $\overline{n}$ per mode value after doing some preliminary simulations of different values (for small number of samples). Here, we applied a simplified version of the heuristic from Algorithm 3 to reduce the number of simulations we need to do while still being able to extract useful results. Essentially, after performing GBS once for 10^4 samples, we remove two of the correct biclusters that have the highest counts in all the samples by replacing their values with 0s in D (or D_{bin}). The GBS process is then done once more for the same number of samples (10^4) to see the change in measuring the last remaining bicluster.

6.4.2 Results and discussion

After applying GBS to our 12×12 datasets, we notice that the number of times the correct biclusters were successfully observed were in single digits for the real-valued dataset and in the low double-digits for the binary one (see Table 5). At first glance, these results look very underwhelming. But we would like to bring to attention two points: firstly, even for the real-valued dataset, these results are still orders of magnitude better than results we may expect from random sampling.[31] This is corroborated by the 2013 theoretical work[32] by Aaronson and Arkhipov (2013).

Secondly, the standard way that the GBS process is set up is as a fixed circuit that runs in constant time (for a given size), unlike a gate-model quantum algorithm or a quantum annealing process. Thus, to an extent, taking more samples for GBS is more tolerable than taking more samples in the other types of models.[33]

Table 6 shows the results of GBS once the two most observed biclusters (from the previous run) were removed from the datasets. Here, we see a significant increase of observing the remaining bicluster, with the largest rise seen for the binary dataset. This suggests that a full-fledged GBS heuristic that detects and removes biclusters iteratively may improve its chances for finding one or more biclusters in the i^{th} iteration than in the $(i-1)^{\text{th}}$ iteration. Though further studies are needed before we can say more, we hope that the insight from this result is useful to the research community at large.

6.5 A short summary of the results

Following is the summary of the results from our simulations:

[30] Since one may not expect to find an ideal bicluster located entirely continuosly across the dataset.

[31] As mentioned in Section 6.3.2, random sampling has a probability in the order of 10^{-6}, the worst of our GBS results suggest probabilities in the order of 10^{-4}.

[32] While that work is based on traditional boson sampling, it would also broadly apply for the comparison in question.

[33] Of course, in order to make a stronger case for GBS to be used for this application, it would still have to yield meaningful results that have to be better than the alternatives in some way(s). What we are suggesting is that our results do not take GBS out of contention for being a candidate for biclustering.

1. Both boson sampling and GBS can be applied to real-valued and binary datasets to detect biclusters within larger datasets.
2. Both boson sampling and GBS perform better when the contrast between the values that are inside and outside the bicluster is large.
3. The GBS approach seems to produce comparable or better results for fewer number of samples (against our current way of performing boson sampling).
4. The GBS approach provides us with both the rows and columns of the bicluster simultaneously.
5. If multiple biclusters are present within a dataset, then their chances of being detected goes down. We hope this can be somewhat alleviated in the future by the use of preprocessing and postprocessing methods[34] to extract meaningful results.

7 Future work

Based on the outcomes of Section 6, we can now comment on the potential future work that would help to better understand the utility of these restricted models of quantum computation (enabled by photonics) for the problems of biclustering.

1. **Better method for boson sampling**: In our simulations, GBS outperformed boson sampling in terms of the number of samples that it needed to work effectively. But since both methods use mostly similar components (beamsplitters and phaseshifters), the difference may be more to do with how we encode the problems for boson sampling. One naive solution could be to use the same technique to make unitary matrices that GBS uses, but by using single photon sources rather than squeezing light. However, this suggestion would probably need further refinement since (i) producing synchronized single photons at scale is a major engineering hurdle and (ii) it ignores the possibility of photon loss.
2. **Implementation on real photonic hardware:** Another potential future work that can be done is a comparison on real photonic hardware for boson sampling and GBS. At the time of writing, there was no publicly available hardware that would support problems of the sizes like the ones in this work.
3. **Experimentation on real-world datasets:** With better simulation software and hardware, it would be useful to test these photonic methods on datasets representing non-synthetic data that come from domains where biclustering is the most relevant.
4. **Comparison against classical methods:** Like we mentioned in the section before, comparison against industry-standard classical methods is also part of the work that needs to be done for making stronger claims.
5. **Develop hybrid quantum-classical techniques:** This is a direction of research that we believe will make a significant impact for this application. It is quite possible that classical methods can outperform a direct application of restricted models of quantum computing (like boson sampling and GBS), at least in the short term, when the latter has to be moderately adapted to an application problem. This is because classical methods can often exploit their Turing completeness to implement a range of solutions rather than be restricted to just one type. For example, currently, it is still challenging for quantum annealing (another restricted model of quantum computing) to solve k-SAT problems better than classical methods (Gabor et al., 2019).

   Another reason for investigating hybrid quantum-classical techniques is the fact that quantum circuits in the short to medium term (even for photonic-based hardware) will remain relatively small. Taking these two points together, we believe that in the best case scenario, boson sampling or GBS may be best used as a smaller (but effective) sub-routine inside a larger method for solving problems in biclustering. This is essentially to counter disadvantages of either approaches.

TABLE 5 Results of GBS-problem 2 for 10^4 samples for real-valued and binary datasets (D and D_{bin}, respectively).

Dataset	No. of correct samples		
	Bicluster β_1	Bicluster β_2	Bicluster β_3
D	3	5	6
D_{bin}	41	34	24

We can see that GBS performs better for D_{bin} than for D.

TABLE 6 Results on GBS-problem 2 datasets after the top two results in each category were removed (see Table 5).

Dataset	No. of correct samples		
	Bicluster β_1	Bicluster β_2	Bicluster β_3
D	1,030	0	0
D_{bin}	0	0	5,966

For D, it was β_2 and β_3 and for D_{bin} it was β_1 and β_2.

8 Conclusion

In this work, we proposed the use of two computational models from photonics: namely boson sampling and Gaussian boson sampling (GBS), for the problem of biclustering. Being the first work in this research direction, we conducted four preliminary tests where we simulated the application of these quantum computing models on synthetic datasets for biclustering. We found that these models are best suited for binary datasets and datasets where the contrast in the values is very high. We also found that the direct application of GBS has two main advantages over the direct application of boson sampling: (i) fewer number of samples needed and (ii) the ability two locate both rows and columns of a bicluster simultaneously. Based on our findings, we recommend a list of future work, primarily to do with (a) better encoding of biclustering problems in boson sampling, (b) experiments on photonic hardware and on (c) non-synthetic data, (d) comparison with classical methods and (e) the development of hybrid quantum-classical methods. We hope that the results of our preliminary

[34] For example, for preprocessing, you can use a threshold value to generate a binary version of a real-valued dataset (Bonaldi et al., 2023). Similarly, postprocessing methods that build on the raw solutions in order to produce better solutions can also be considered (Arrazola and Bromley, 2018).

simulations are useful to the research community for all future work in this domain.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) are found here: https://github.com/aborle1/Photonic_bicluster.

Author contributions

ABo: Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. ABh: Conceptualization, Data curation, Software, Visualization, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

The authors would like to thank Dr Charles Nicholas and the DREAM Laboratory in the University of Maryland Baltimore County (UMBC), for providing access to their high performance compute servers for our simulations. The initial ideas for the example for application of biclustering in machine learning for Section 2.5.1 and Figure 2 were generated by the Claude Sonnet 3.5 large language model (LLM). The final text in Section 2.5.1 and the illustration in Figure 2 were written and drawn respectively by the authors (manually) based on suggestions provided by the above mentioned AI model.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aaronson, S., and Arkhipov, A. (2011). "The computational complexity of linear optics," in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (New York, NY: ACM), 333–342.

Aaronson, S., and Arkhipov, A. (2013). Bosonsampling is far from uniform. *arXiv* [preprint]. doi: 10.48550/arXiv.1309.7460

Adachi, S. H., and Henderson, M. P. (2015). Application of quantum annealing to training of deep neural networks. *arXiv* [preprint]. doi: 10.48550/arXiv.1510.06356

Arrazola, J. M., and Bromley, T. R. (2018). Using gaussian boson sampling to find dense subgraphs. *Phys. Rev. Lett.* 121:030503. doi: 10.1103/PhysRevLett.121.030503

Arrazola, J. M., Bromley, T. R., and Rebentrost, P. (2018). Quantum approximate optimization with gaussian boson sampling. *Phys. Rev. A* 98:012322. doi: 10.1103/PhysRevA.98.012322

Ayadi, W., Elloumi, M., and Hao, J.-K. (2009). A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data. *BioData Min.* 2, 1–16. doi: 10.1186/1756-0381-2-9

Bertsimas, D., and Tsitsiklis, J. (1993). Simulated annealing. *Stat. Sci.* 8, 10–15. doi: 10.1214/ss/1177011077

Bonaldi, N., Rossi, M., Mattioli, D., Grapulin, M., Fernández, B. S., Caputo, D., et al. (2023). Boost clustering with gaussian boson sampling: a full quantum approach. *arXiv* [preprint]. doi: 10.1007/s42484-024-00185-w

Bottarelli, L., Bicego, M., Denitto, M., Di Pierro, A., Farinelli, A., and Mengoni, R. (2018). Biclustering with a quantum annealer. *Soft Comp.* 22, 6247–6260. doi: 10.1007/s00500-018-3034-z

Brod, D. J., Galvão, E. F., Crespi, A., Osellame, R., Spagnolo, N., and Sciarrino, F. (2019). Photonic implementation of boson sampling: a review. *Adv. Phot.* 1, 034001–034001. doi: 10.1117/1.ap.1.3.034001

Bromley, T. R., Arrazola, J. M., Jahangiri, S., Izaac, J., Quesada, N., Gran, A. D., et al. (2020). Applications of near-term photonic quantum computers: software and algorithms. *Quant. Sci. Technol.* 5:034010. doi: 10.1088/2058-9565/ab8504

Castanho, E. N., Aidos, H., and Madeira, S. C. (2022). Biclustering fMRI time series: a comparative study. *BMC Bioinformatics* 23, 1–30. doi: 10.1186/s12859-022-04733-8

Castanho, E. N., Aidos, H., and Madeira, S. C. (2024). Biclustering data analysis: a comprehensive survey. *Brief. Bioinform.* 25:bbae342. doi: 10.1093/bib/bbae342

Cheng, Y., and Church, G. M. (2000). Biclustering of expression data. *Intell. Syst. Mol. Biol.* 8, 93–103.

Choi, S., Ha, H., Hwang, U., Kim, C., Ha, J.-W., and Yoon, S. (2018). Reinforcement learning based recommender system using biclustering technique. *arXiv* [preprint]. doi: 10.48550/arXiv.1801.05532

Cipra, B. A. (1987). An introduction to the ising model. *Am. Math. Monthly* 94, 937–959. doi: 10.1080/00029890.1987.12000742

Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S., and Walmsley, I. A. (2016). Optimal design for universal multiport interferometers. *Optica* 3, 1460–1465. doi: 10.1364/OPTICA.3.001460

Clifford, P., and Clifford, R. (2018). "The classical complexity of boson sampling," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA: SIAM), 146–155.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to Algorithms*. Cambridge, MA: MIT Press.

Cui, L., Acharya, S., Mishra, S., Pan, Y., and Huang, J. Z. (2020). Mmco-clus-an evolutionary co-clustering algorithm for gene selection. *IEEE Trans. Knowl. Data Eng.* 34, 4371–4384. doi: 10.1109/TKDE.2020.3035695

de Castro, P. A., de França, F. O., Ferreira, H. M., and Von Zuben, F. J. (2007). "Applying biclustering to text mining: an immune-inspired approach," in *Artificial Immune Systems: 6th International Conference, ICARIS 2007, Santos, Brazil, August 26-29, 2007. Proceedings* (Berlin; Heidelberg: Springer), 83–94.

Deng, Y.-H., Gong, S.-Q., Gu, Y.-C., Zhang, Z.-J., Liu, H.-L., Su, H., et al. (2023a). Solving graph problems using gaussian boson sampling. *Phys. Rev. Lett.* 130:190601. doi: 10.1103/PhysRevLett.130.190601

Deng, Y.-H., Gu, Y.-C., Liu, H.-L., Gong, S.-Q., Su, H., Zhang, Z.-J., et al. (2023b). Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage. *Phys. Rev. Lett.* 131:150601. doi: 10.1103/PhysRevLett.131.150601

Deshpande, A., Mehta, A., Vincent, T., Quesada, N., Hinsche, M., Ioannou, M., et al. (2022). Quantum computational advantage via high-dimensional gaussian boson sampling. *Sci. Adv.* 8:eabi7894. doi: 10.1126/sciadv.abi7894

Dhillon, I. S. (2001). "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 269–274.

Gabor, T., Zielinski, S., Feld, S., Roch, C., Seidel, C., Neukart, F., et al. (2019). "Assessing solution quality of 3sat on a quantum annealing platform," in *Quantum Technology and Optimization Problems: First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings 1* (Berlin; Heidelberg: Springer), 23–35.

Glynn, D. G. (2013). Permanent formulae from the veronesean. *Designs Codes Cryptogr.* 68, 39–47. doi: 10.1007/s10623-012-9618-1

Halmos, P. (1950). Summa. brasil. math. *Norm. Dilat. Extens. Operat.* 2, 125–134.

Halmos, P. R. (1951). Normal dilations and extensions of operators. *Bull. Am. Math. Soc.* 57, 294.

Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C., and Jex, I. (2017). Gaussian boson sampling. *Phys. Rev. Lett.* 119:170501. doi: 10.1103/PhysRevLett.119.170501

Heurtel, N., Fyrillas, A., De Gliniasty, G., Le Bihan, R., Malherbe, S., Pailhas, M., et al. (2023). Perceval: a software platform for discrete variable photonic quantum computing. *Quantum* 7:931. doi: 10.22331/q-2023-02-21-931

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., et al. (2010). Fabia: factor analysis for bicluster acquisition. *Bioinformatics* 26, 1520–1527. doi: 10.1093/bioinformatics/btq227

José-García, A., Jacques, J., Sobanski, V., and Dhaenens, C. (2022). Biclustering algorithms based on metaheuristics: a review. *Metaheurist. Mach. Learn.* 39–71. doi: 10.1007/978-981-19-3888-7_2

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Phys. Rev. E* 58:5355. doi: 10.1103/PhysRevE.58.5355

Karim, M. B., Huang, M., Ono, N., Kanaya, S., and Altaf-Ul-Amin, M. (2019). Bicluso: a novel biclustering approach and its application to species-voc relational data. *IEEE/ACM Transact. Comp. Biol. Bioinf.* 17, 1955–1965. doi: 10.1109/TCBB.2019.2914901

Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., and Weedbrook, C. (2019). Strawberry fields: a software platform for photonic quantum computing. *Quantum* 3:129. doi: 10.22331/q-2019-03-11-129

King, J., Yarkoni, S., Raymond, J., Ozfidan, I., King, A. D., Nevisi, M. M., et al. (2019). Quantum annealing amid local ruggedness and global frustration. *J. Phys. Soc. Jpn.* 88:061007. doi: 10.7566/JPSJ.88.061007

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.* 13, 703–716. doi: 10.1101/gr.648603

Kumar, V., Bass, G., Tomlin, C., and Dulny, J. (2018). Quantum annealing for combinatorial clustering. *Quant. Inf. Process.* 17:1–14. doi: 10.1007/s11128-017-1809-2

Maâtouk, O., Ayadi, W., Bouziri, H., and Duval, B. (2021). Evolutionary local search algorithm for the biclustering of gene expression data based on biological knowledge. *Appl. Soft Comput.* 104:107177. doi: 10.1016/j.asoc.2021.107177

Madeira, S. C., and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transact. Comp. Biol. Bioinf.* 1, 24–45. doi: 10.1109/TCBB.2004.2

Madsen, L. S., Laudenbach, F., Askarani, M. F., Rortais, F., Vincent, T., Bulmer, J. F., et al. (2022). Quantum computational advantage with a programmable photonic processor. *Nature* 606, 75–81. doi: 10.1038/s41586-022-04725-x

Mezher, R., Carvalho, A. F., and Mansfield, S. (2023). Solving graph problems with single-photons and linear optics. *arXiv* [preprint]. doi: 10.1103/PhysRevA.108.032405

Miller, G. (1930). On the history of determinants. *Am. Math. Monthly* 37, 216–219. doi: 10.1080/00029890.1930.11987058

Mirkin, B. (1997). Mathematical classification and clustering. *J. Operat. Res. Soc.* 48, 852–852. doi: 10.1057/palgrave.jors.2600836

Orzechowski, P., and Boryczko, K. (2016). "Text mining with hybrid biclustering algorithms," in *International Conference on Artificial Intelligence and Soft Computing* (Berlin; Heidelberg: Springer), 102–113.

Pontes Balanza, B., Giráldez, R., and Aguilar Ruiz, J. S. (2015). Biclustering on expression data: a review. *J. Biomed. Inf.* 57, 163–180. doi: 10.1016/j.jbi.2015.06.028

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., et al. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22, 1122–1129. doi: 10.1093/bioinformatics/btl060

Quesada, N., and Arrazola, J. M. (2020). Exact simulation of gaussian boson sampling in polynomial space and exponential time. *Phys. Rev. Res.* 2:023005. doi: 10.1103/PhysRevResearch.2.023005

Quesada, N., Arrazola, J. M., and Killoran, N. (2018). Gaussian boson sampling using threshold detectors. *Phys. Rev. A* 98:062322. doi: 10.1103/PhysRevA.98.062322

Raff, E., Zak, R., Lopez Munoz, G., Fleming, W., Anderson, H. S., Filar, B., et al. (2020). "Automatic yara rule generation using biclustering," in *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security* (New York, NY: ACM), 71–82.

Reck, M., Zeilinger, A., Bernstein, H. J., and Bertani, P. (1994). Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* 73:58. doi: 10.1103/PhysRevLett.73.58

Ryser, H. J. (1963). Combinatorial mathematics. *Am. Math. Soc.* 14:147. doi: 10.5948/UPO9781614440147

Schuld, M., Brádler, K., Israel, R., Su, D., and Gupt, B. (2020). Measuring the similarity of graphs with a gaussian boson sampler. *Phys. Rev. A* 101:032314. doi: 10.1103/PhysRevA.101.032314

Solomons, N. R., Thomas, O. F., and McCutcheon, D. P. (2023). Effect of photonic errors on quantum enhanced dense-subgraph finding. *Phys. Rev. Appl.* 20:054043. doi: 10.1103/PhysRevApplied.20.054043

Sun, J., and Zhang, Y. (2022). Recommendation system with biclustering. *Big Data Mining Anal.* 5, 282–293. doi: 10.26599/BDMA.2022.9020012

Takagi, T. (1924). On an algebraic problem reluted to an analytic theorem of carathéodory and fejér and on an allied theorem of landau. *Jpn. J. Math.* 1, 83–93. doi: 10.4099/jjm1924.1.0_83

Termini, S. (2006). *Imagination and Rigor: Their Interaction Along the Way to Measuring Fuzziness and Doing Other Strange Things*. Berlin; Heidelberg: Springer-Verlag.

Troyansky, L., and Tishby, N. (1996). On the quantum evaluation of the determinant and the permanent of a matrix. *Proc. Phys. Comput.* 96.

Wang, B., Miao, Y., Zhao, H., Jin, J., and Chen, Y. (2016). A biclustering-based method for market segmentation using customer pain points. *Eng. Appl. Artif. Intell.* 47, 101–109. doi: 10.1016/j.engappai.2015.06.005

Xie, J., Ma, A., Fennell, A., Ma, Q., and Zhao, J. (2019). It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. *Brief. Bioinf.* 20, 1450–1465. doi: 10.1093/bib/bby014