Sample efficient graph classification using binary Gaussian boson sampling

Amanuel Anteneh [*]
Department of Computer Science, University of Virginia, Charlottesville, Virginia 22903, USA

Olivier Pfister [†]
Department of Physics, University of Virginia, Charlottesville, Virginia 22903, USA

(Received 20 September 2023; accepted 16 November 2023; published 11 December 2023)

We present a variation of a quantum algorithm for the machine learning task of classification with graph-structured data. The algorithm implements a feature extraction strategy that is based on Gaussian boson sampling (GBS), a near-term model of quantum computing. However, unlike the currently proposed algorithms for this problem, our GBS setup requires only binary (light or no light) detectors, as opposed to photon-number-resolving detectors. Binary detectors are technologically simpler and can operate near room temperature, making our algorithm much less complex and costly to implement physically. We also investigate the connection between graph theory and the Torontonian matrix function which characterizes the probabilities of binary GBS detection events.

DOI: 10.1103/PhysRevA.108.062411

I. INTRODUCTION

Graphs are one of the most versatile data structures used in computing, and developing machine learning methods for working with graph-structured data has been a growing subfield of machine learning research. Graph classification, in particular, has useful applications in fields such as bioinformatics, network science, and computer vision as many of the objects studied in these fields can easily be represented as graphs. However, using graph-structured data with machine learning models is not a straightforward task. This is because one of the most common ways of representing a graph for computational applications, i.e., as an adjacency matrix, cannot easily be used as an input for machine learning classifiers which primarily take vector-valued data as their inputs. Therefore, a common way of working with graph-structured data is to define a feature map ϕ that maps a graph G to a vector in a Hilbert space called a feature space. From there, a function κ, called a kernel, is defined that measures the similarity of two graphs in the feature space. An example of a feature map from $\mathbb{R}^2 \to \mathbb{R}^3$ is shown in Fig. 1.

Kernel methods refer to machine learning algorithms that learn by comparing pairs of data points using this similarity measure. In our context we have a set of graphs $\mathbb{G}$, and we call a kernel κ a *graph kernel* if it is a function of the form $\kappa : \mathbb{G} \times \mathbb{G} \to \mathbb{R}$ [1,2]. The most common example of a kernel function is the feature space's inner product $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$. The goal of such methods is to construct mappings to feature vectors whose entries (the features) relate to relevant information about the graphs. Using a Gaussian boson sampling (GBS) device to construct graph kernels was an idea first proposed by Schuld *et al.* [3].

Boson sampling was first proposed by Aaronson and Arkhipov [4] as a task—generating the photon-counting outcomes of the "quantum Galton board" constituted by an $M \times M$ optical interferometer fed single photons in some of its input ports—that is strongly believed to be intractable to classical computers. The reason for this intractability is that calculating the probability distribution for generating random outcomes using Monte Carlo simulations requires calculating the permanent of an $M \times M$ matrix. Calculating the permanent of a general matrix is known to be #P-complete [5], which is a class of problems comparable to the class of NP-complete problems in difficulty. Gaussian boson sampling [6] is a variant of boson sampling in which the single-photon inputs are replaced with single-mode squeezed states, as produced, for example, by two-photon-emitting optical parametric amplifiers [7]. The GBS probability distribution is governed by the Hafnian of an $M \times M$ matrix. Calculating the Hafnian of a general square matrix can be reduced to the task of calculating permanents; therefore, calculating the Hafnian is also #P-complete. In both cases, a quantum machine implementing boson sampling or GBS can easily sample from these hard-to-calculate probability distributions just because they are "wired in," and this constitutes the "quantum advantage" that was recently demonstrated in optical experiments [8,9]. Note also that the initial "quantum supremacy" result obtained by Google on a superconducting qubit array [10] was a quantum (circuit) sampling result as well.

Beyond these necessary initial steps of demonstrating that quantum hardware can, indeed, reach regions inaccessible to classical hardware, a subsequent question is that of the utility of a sampling task. Whereas the usefulness of sampling in and of itself is far from established, we know that the histograms produced by statistically significant sampling constitute

[*]asa2rc@virginia.edu
[†]olivier.pfister@gmail.com

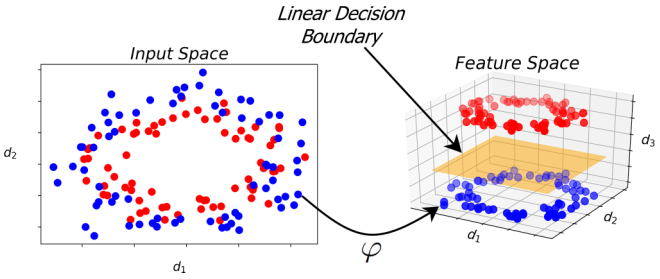


FIG. 1. In the original input space $\mathbb{R}^2$ the data points, which belong to either the red or blue class, are not separable by a linear function (the decision boundary), but after mapping the points to feature vectors in a higher-dimensional space $\mathbb{R}^3$, a linear function is able to separate the two classes. This linear decision boundary can be calculated by supervised machine learning models such as a support vector machine. In our case the input space is the set of all undirected graphs, which we denote as $\mathbb{G}$.

empirical probability distributions that tend toward the true, classically intractable probability distributions for sample numbers linear in the number of possible outcomes [11]. The problem is that this very number of possible outcomes grows exponentially with M in an M-qubit quantum circuit in general [12] and exponentially or superexponentially with M in an M-optical-mode boson or Gaussian boson sampler, which dispels any notion of quantum advantage for calculating the corresponding quantum probability distributions.

One direction that has been explored in relation to this conundrum is the binning of GBS measurements, which results in outcome classes whose cardinality scales favorably (e.g., polynomially) with the problem size (the GBS mode number). The immediate downside of such an approach is the loss of information it entails, which impacts usefulness. However, graph classification using feature vectors and coarse-graining might provide advantageous GBS applications. This was first pointed out by Schuld *et al.* [3].

In this paper, we show that a technologically simpler version of GBS, which we term binary GBS, can achieve comparable or better performance. This paper is structured as follows. In Sec. II we give broad reminders about GBS and graph theory (with details given in Appendix A) and the current GBS graph kernel from Ref. [3]. We then present our graph kernel in Sec. III, along with results from numerical experiments and analyses of its complexity, features, and advantages.

II. REMINDERS ABOUT GAUSSIAN BOSON SAMPLING AND GRAPH THEORY

A. Gaussian boson sampling

As mentioned above, an M-mode GBS device comprises M single-mode-squeezing inputs, an $M \times M$ optical interferometer, and M photon-number-resolving (PNR) detectors; see Fig. 2 for an example. The latter have come of age in superconducting devices such as transition-edge sensors [15] and superconducting nanowire single-photon detectors [16]. Both were recently used to make PNR measurements of as many as 100 photons [17,18].

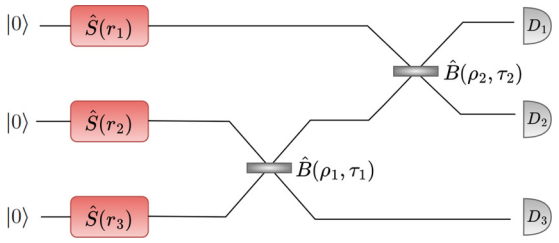


FIG. 2. Example of a three-mode Gaussian boson sampler. Mode $i \in \{1, 2, 3\}$ starts in the vacuum state $|0\rangle$, is then squeezed by $\hat{S}(r_i)$, and passes through the network of two beam splitters (the interferometer) before the number of photons in each mode is measured by the detectors $D_{i\in\{1,2,3\}}$.

An M-mode Gaussian boson sampler prepares a Gaussian (Wigner function) quantum state using the M squeezers and the interferometer. The squeezers output squeezed light into the interferometer, and the photons are then passed through the interferometer, after which the M detectors detect what modes the photons end up in, resulting in a detection event. We denote a detection event as $\mathbf{n} = (n_1, \ldots, n_M)$, where n_i is the photon count in the ith mode and the total number of photons is $n = \sum_{i=1}^{M} n_i$.

We now consider binary detectors, such as single-photon avalanche photodiodes, which are single photon sensitive but are not PNR and give the same signal regardless of how many photons were absorbed. In this case, we have $n_i \in \{0, 1\}$, where $n_i = 0$ indicates zero photons were detected in that mode and $n_i = 1$ indicates that at least one photon was detected. When using binary detectors, we no longer know the total photon number n, so we use N to denote the number of detectors that detect photons leading to $\sum_{i=1}^{M} n_i = N \leqslant M$.

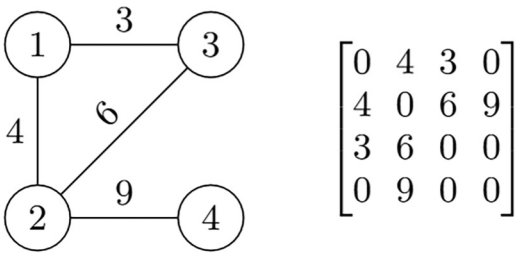
An M-mode Gaussian state is fully described by a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{2M\times 2M}$ and a displacement vector $\mathbf{d} \in \mathbb{R}^{2M}$ [19].

B. Graph theory

In this paper we define a graph $G = (V, E)$ as a set of vertices $V = \{v_1, v_2, \ldots\}$ and a set of edges $E = \{(v_1, v_1), (v_1, v_2), \ldots, (v_i, v_j), \ldots\}$ that connect vertices if the edge value is not zero. A graph can be unweighted, with all nonzero edge weights equal to 1, or weighted, for example, with real edge weights in GBS. For undirected graphs, which is what we will exclusively work with in this paper, $(v_i, v_j) = (v_j, v_i)\ \forall\, i, j$. The size of a graph is equal to the cardinality $|V|$ of its vertex set. The degree of a vertex v is the number of edges that are connected to it. The maximum degree of a graph is the largest degree of a vertex in its vertex set.

Graphs can be represented in a number of ways, such as a diagram [Fig. 3(a)] or a more computationally useful adjacency matrix [Fig. 3(b)]. The adjacency matrix of an undirected graph G with $|V|$ vertices is a $|V| \times |V|$ symmetric matrix A with entries a_{ij}, where a_{ij} is the weight of the edge connecting vertices i and j.

A substantial amount of work has been done on the connection between graph theory and Gaussian boson sampling with PNR detectors [20–22]. In Appendix A, we summarize some



(a)

$$\begin{bmatrix} 0 & 4 & 3 & 0 \\ 4 & 0 & 6 & 9 \\ 3 & 6 & 0 & 0 \\ 0 & 9 & 0 & 0 \end{bmatrix}$$

(b)

FIG. 3. Four-vertex graph and its corresponding adjacency matrix. (a) Undirected weighted four-vertex graph with four edges. (b) Adjacency matrix of the graph.

details of this work, namely, how a given graph adjacency matrix can be encoded in a GBS experiment.

C. Sample complexity of GBS

1. The problem with using GBS beyond sampling

The sample complexity of a machine learning algorithm refers to the number of samples or amount of data required to learn some target function. In the case of GBS applications it refers to the number of samples we need to generate from the GBS device to learn or approximate a probability distribution over some set of the photon detection events. This complexity type is extremely important to examine for any applications of GBS because it could potentially render certain applications of GBS intractable for larger problem sizes.

For example, it was shown that the GBS device utilizing PNR detectors can encode the graph isomorphism problem [21]. This is done by encoding two graphs into two GBS devices and sampling each S times. The S samples could then be used, in principle, to reconstruct the probability distribution over all possible detection events $\mathbf{n}$ for a given M and n. However, this cannot be done efficiently enough to provide a quantum advantage. Indeed, we know from Refs. [11,23] that to reconstruct a probability distribution D over a discrete finite set Ω of cardinality $|\Omega|$ from an empirical distribution $\hat{D}$ constructed from samples from D we require

$$S = \left\lceil \frac{2\left(\ln(2)|\Omega| + \ln\left(\frac{1}{\delta}\right)\right)}{\epsilon^2} \right\rceil \tag{1}$$

samples to guarantee that

$$p(||D - \hat{D}||_1 \geqslant \epsilon) \leqslant \delta, \tag{2}$$

where $||D - \hat{D}||_1$ denotes the L_1 distance between D and $\hat{D}$. In other words we require

$$O\left(\frac{|\Omega| + \ln\left(\frac{1}{\delta}\right)}{\epsilon^2}\right) \tag{3}$$

samples to ensure with a probability of at most δ that the sum of the absolute values of the errors on the empirical probability distribution is ϵ or greater.

This means that the number of samples we need to approximate a probability distribution scales linearly with the number of elements in its sample space, i.e., the number of outcomes. In the case where D is the probability distribution over the set of all possible PNR detection events, the number of such events for a given number of modes M and maximum number of photons n is

$$|\Omega| = \binom{n+M-1}{M-1} = \frac{(n+M-1)!}{n!(M-1)!}, \tag{4}$$

which in number theory is also known as the formula for the number of weak compositions of an integer n in M parts. As shown in Appendixes B and C, under the assumption that the number of modes scales quadratically with the number of photons, $M \in O(n^2)$, this quantity grows superexponentially with M and, in general, scales as $O((n+M-1)^{M-1})$, meaning that as the size of the graphs increases, and therefore as the number of modes of the GBS device increases, we require an exponential number of samples to ensure the algorithm can give us the correct result within a certain probability. Therefore, while the algorithm may, in principle, be able to decide graph isomorphism, it is sample inefficient to an exponential degree, making it intractable to implement even with a fault-tolerant quantum computer.

2. Coarse graining of sample distributions

However, a method was suggested in [21] to coarse grain the probability distribution by combining outcomes in groups called orbits. Coarse graining in this sense means constructing a new probability distribution over the set of these groups, the cardinality of which is less than the original set of all possible detection events. An orbit $\mathrm{O}_{\mathbf{n}}$ consists of a detection event $\mathbf{n}$ and all of its permutations. For example, the orbit that contains the detection event $\mathbf{n} = (1, 2, 2)$ also contains the detection events (2,1,2) and (2,2,1). The number of orbits for a four-mode GBS device is equal to the number of ways one can write $n_1 + n_2 + n_3 + n_4 = n$, where the order of the summands does not matter. This is called the number of partitions of integer n into M parts, and from the number-theory literature [24] it is known to behave asymptotically as

$$|\Omega| \approx \frac{e^{\pi\sqrt{\frac{2(n-M)}{3}}}}{4\sqrt{3}(n-M)}, \quad M \leqslant n \leqslant 2M. \tag{5}$$

If we assume the number of photons grows linearly with the number of modes, $n \in \Theta(M) \to n = 2M$, we have the following asymptotic bound on the number of orbits:

$$\frac{1}{4\sqrt{3}M} e^{\pi\sqrt{\frac{2M}{3}}} \in O\left(\frac{e^{\pi\sqrt{\frac{2M}{3}}}}{M}\right). \tag{6}$$

This means the number of orbits, which is now the number of outcomes $|\Omega|$ from Eq. (1), grows as $M^{-1}e^{\pi\sqrt{2M/3}}$, meaning we would have a sample complexity of

$$O\left(\frac{M^{-1}e^{\pi\sqrt{2M/3}} + \ln\left(\frac{1}{\delta}\right)}{\epsilon^2}\right), \tag{7}$$

which is subexponential but still intractable for large M.

3. Sample complexity of previously proposed GBS graph kernels

The first GBS-based graph kernel proposed in [3] maps a graph G to feature vectors in a feature space

$\phi : G \to \mathbf{f} = (f_1, f_2, \ldots, f_D) \in \mathbb{R}^D$, where $f_i = p(\mathrm{O}^i_{\mathbf{n}})$ is the probability of detecting a detection event from the orbit $\mathrm{O}^i_{\mathbf{n}}$. This kernel was shown to perform well against three of the four classical kernels we use as benchmarks in this paper. However, a shortcoming of this method is that the sample complexity is $O(M^{-1}e^{\pi\sqrt{2M/3}})$.

The second GBS kernel is of the form $\phi : G \to \mathbf{f} = (f_1, f_2, \ldots, f_D) \in \mathbb{R}^D$, with $f_i = p(\mathcal{M}^i_{n,\Delta_s})$, where $p(\mathcal{M}^i_{n,\Delta_s})$ is the probability of detecting a detection event that belongs to the "metaorbit" $\mathcal{M}^i_{n,\Delta_s}$. A metaorbit $\mathcal{M}_{n,\Delta_s}$ is uniquely defined by a total photon number n and Δ_s, which is defined as

$$\Delta_s = \left\{ \mathbf{n} : \sum_i n_i = n \ \wedge \ \forall i : n_i \leqslant s \right\}. \tag{8}$$

Therefore a metaorbit consists of all detection events in which the total photon number is equal to n, where no detector counts more than s photons. It is claimed that this strategy partitions the set of all PNR detection events into a polynomial number of subsets in n [25].

III. THE ALGORITHM

A. GBS with binary detectors and its relation to graph theory

While the relationship between GBS with PNR detectors and graph theory has been thoroughly explored, there has been little exploration of how GBS with binary detectors fits into the picture. In this section we shed some light on the relationship between the two. As stated before, when using binary detectors, the detection outcomes are of the form $\mathbf{n}_{\text{bin}} = (n_1, \ldots, n_M)$, where $n_i \in \{0, 1\}\ \forall\, i$ and $n_i = 1$ indicates the ith detector detected one or more photons. The probability of detecting a detection outcome with binary detectors is characterized by a matrix function called the Torontonian, to which the same arguments for classical intractability as for the Hafnian can be extended [26]. The probability of a given binary detection event $\mathbf{n}_{\text{bin}}$ is given by

$$p(\mathbf{n}_{\text{bin}}) = \frac{\mathrm{Tor}(O_{\mathbf{n}_{\text{bin}}})}{\sqrt{\det(Q)}} = \frac{\mathrm{Tor}(X\tilde{A}_{\mathbf{n}_{\text{bin}}})}{\sqrt{\det(Q)}}, \tag{9}$$

where

$$\tilde{A} = (A \oplus A), \tag{10}$$

$$X = \begin{bmatrix} 0 & \mathbb{I} \\ \mathbb{I} & 0 \end{bmatrix}, \tag{11}$$

$$Q = (\mathbb{I}_{2M} - X\tilde{A})^{-1}, \tag{12}$$

$$O = \mathbb{I} - Q^{-1} \tag{13}$$

and $\mathrm{Tor}(\cdot)$ is the Torontonian of a $2N \times 2N$ matrix A, defined as

$$\mathrm{Tor}(A) = \sum_{Z \in P([N])} (-1)^{|Z|} \frac{1}{\sqrt{\det(\mathbb{I} - A_Z)}}, \tag{14}$$

where $P([N])$ is the power set, the set of all possible subsets, of the set $[N] = \{1, 2, \ldots, N\}$. The probability of a PNR detection event $\mathbf{n}$ can be written in terms of the matrix O as

$$p(\mathbf{n}) = \frac{1}{\sqrt{\det(Q)}} \frac{\mathrm{Haf}(\tilde{A}_{\mathbf{n}})}{\mathbf{n}!} = \frac{1}{\sqrt{\det(Q)}} \frac{\mathrm{Haf}(XO_{\mathbf{n}})}{\mathbf{n}!}. \tag{15}$$

The probability of a binary GBS detection event is simply the sum of all probabilities of the corresponding PNR detection events. A useful example to illustrate this is a four-mode Gaussian boson sampler programed according to some adjacency matrix A of a graph G. Suppose we use binary detectors and measure the detection event $\mathbf{n}_{\text{bin}} = (1, 0, 1, 0)$. The corresponding detection events when using PNR detectors would be of the form $\mathbf{n} = (n_1, 0, n_3, 0)$, where $n_1, n_3 > 0$ and $n_1 + n_3$ is even. We will define $\mathcal{N}$ to be the set of all possible four-mode PNR detection events with zeros in the second and fourth indices; that is, only the second and fourth detectors detect no photons. From this we have

$$p(1, 0, 1, 0) = \frac{\mathrm{Tor}(X\tilde{A}_{(1,0,1,0)})}{\sqrt{\det(Q)}} = \sum_{\mathbf{n} \in \mathcal{N}} p(\mathbf{n}) = \sum_{\mathbf{n} \in \mathcal{N}} \frac{\mathrm{Haf}^2(A_{\mathbf{n}})}{\mathbf{n}!\sqrt{\det(Q)}}. \tag{16}$$

This means that the Torontonian of $X\tilde{A}$ is proportional to an infinite sum of Hafnians because there are an infinite number of integer lists of the form $(n_1, 0, n_3, 0)$ where $n_1, n_3 > 0$. In a real GBS experiment, however, the energy is finite, and therefore, the measured probabilities of these events would be equal to a finite version of this sum in which all detection events with a total photon number greater than some cutoff photon number vanish from the series.

In terms of graph theory this means the probability of detecting $\mathbf{n}_{\text{bin}} = (1, 0, 1, 0)$ is proportional to the sum of the squared Hafnians of all possible subgraphs of G of unbounded and even size with their second and fourth vertices removed. But again, in practice, the maximum size of the subgraphs will always be bounded by some maximum photon number for a real GBS experiment. More generally, we have

$$p(\mathbf{n}_{\text{bin}}) = \frac{\mathrm{Tor}\left(O_{\mathbf{n}_{\text{bin}}}\right)}{\sqrt{\det(Q)}} = \frac{\mathrm{Tor}\left(X\tilde{A}_{\mathbf{n}_{\text{bin}}}\right)}{\sqrt{\det(Q)}} = \sum_{\mathbf{n} \in \mathcal{N}} \frac{\mathrm{Haf}^2(A_{\mathbf{n}})}{\mathbf{n}!\sqrt{\det(Q)}}, \tag{17}$$

where $\mathcal{N}$ is the set of all PNR events that correspond to the binary detection event $\mathbf{n}_{\text{bin}}$ [27].

B. Constructing the feature vectors

Once the GBS device is programmed, we generate S samples from the device. For our algorithm we use binary detectors, so each sample is a list of length M with entries being either 0 or 1. Once we have these samples, we use them to construct the feature vector, for which we have two definitions based on two coarse-graining strategies.

The first is based on what we call the μ coarse-graining strategy in which we group together detection events that contain exactly i detector "clicks" or ones. For example, the detection events (1,0,0) and (0,0,1) would be grouped together since they both contain exactly one detector click. These groups can also be thought of as "binary orbits" since they contain a detection event and all its permutations. This strategy partitions the set of all binary detection events into a linear number of disjoint subsets in N. Using this strategy, we

TABLE I. Graph dataset statistics after preprocessing. A more detailed description of these datasets can be found in Appendix B of Ref. [3].

Dataset	No. of graphs	No. of classes	Average No. of vertices	Average No. of edges
AIDS	1723	2	11.11	11.29
BZR_MD	257	2	20.10	197.69
COX2_MD	118	2	23.90	274.40
ENZYMES	204	6	18.56	36.30
ER_MD	357	2	19.27	185.15
FINGERPRINT	1080	3	10.58	9.10
IMDB-BINARY	806	2	15.98	63.32
MUTAG	179	2	17.48	19.23
NCI1	1853	2	19.77	21.27
PROTEINS	515	2	15.77	29.37
PTC_FM	284	2	13.64	13.99

can define the feature map as $\phi : G \to \mathbf{f} = (f_0, f_1, \ldots, f_N) \in \mathbb{R}^N$, where N is the maximum number of detector clicks and $f_i = \frac{S_i}{S}$, with S_i being the number of samples which contain exactly i ones. Equivalently, this is the probability of detecting an event where exactly i detectors detect a photon.

The second feature map is based on what we call the ν coarse-graining strategy. For a five-mode boson sampler utilizing binary detectors with a maximum click number of 5 there are $|\Omega| = 32$ possible detection outcomes. This coarse-graining strategy groups together detection events whose first five modes are one of these 32 outcomes. For example, the detection event $\mathbf{n}_{\text{bin}} = (0, 1, 0, 0, 1, 0, 1)$ belongs in the group associated with the detection event (0,1,0,0,1) since they are equal if one is only concerned with the first five modes. This strategy partitions the set of all detection events of five or more modes into a constant number of subsets, i.e., 32. The feature map based on this strategy is defined as $\phi : G \to \mathbf{f} = (f_{[0,0,0,0,0]}, f_{[1,0,0,0,0]}, \ldots, f_{\mathbf{n}}) \in \mathbb{R}^{32}$, where $f_{\mathbf{n}}$ is the probability of detecting an event where the first five modes correspond to one of the 32 possible detection outcomes. For example, $f_{[1,0,0,0,0]}$ is the probability that the first detector detects photons and the following four detectors detect vacuum.

Once we construct the feature vector for each graph in the dataset (Table I), we input them into a machine learning classifier such as a support vector machine.

C. Complexity analysis

In this section we discuss, in addition to the time and space complexity, the sample complexity of our algorithm.

1. Sample complexity

Since n_i for binary detection events can be either 0 or 1, we can think of the detection outcomes as binary strings of length M with at most M ones. The number of binary strings of length M with exactly i ones is $\binom{M}{i}$. So the number of possible binary detection events, the number of binary strings of length M with at most M ones, is given by

$$|\Omega| = \sum_{i=0}^{M} \binom{M}{i}. \tag{18}$$

We can show this function grows like 2^M using the binomial expansion

$$2^M = (1+1)^M = \sum_{i=0}^{M} \binom{M}{i} 1^{M-i} 1^i = \sum_{i=0}^{M} \binom{M}{i}. \tag{19}$$

Therefore, we could not simply use the probability of the individual detection events as features without coarse graining even when using binary detectors because we would still need a prohibitively large number of samples to approximate their probabilities to within a constant error. This was the reason for introducing the ν and μ coarse-graining strategies.

Since the number of outcomes of the μ distribution scales linearly with N, which is $\leqslant M$, the sample complexity of approximating the μ coarse-grained probability distribution is

$$O\left(\frac{M + \ln\left(\frac{1}{\delta}\right)}{\epsilon^2}\right), \tag{20}$$

which reduces to $O(M)$ for constant ϵ and δ. The sample complexity of approximating the ν coarse-grained probability distribution is

$$O\left(\frac{32 + \ln\left(\frac{1}{\delta}\right)}{\epsilon^2}\right), \tag{21}$$

which reduces to $O(1)$ for constant ϵ and δ.

2. Space complexity

The size of the ν feature vectors is constant with respect to the graph size, so the space required is $O(1)$, and for the μ feature vectors the size grows linearly with N, which is $\leqslant M$, so the space required is $O(M)$. However, storing the adjacency matrix of the graphs requires $O(M^2)$ space complexity.

3. Time complexity

The time complexity is determined by the most computationally time intensive step of the algorithm, which is encoding the adjacency matrix in the GBS device. This is the case because the encoding process requires taking the Takagi decomposition of matrix A, which for an $M \times M$ matrix has time complexity $O(M^3)$ because it is a special case of the singular-value decomposition [28]. However, quantum algorithms for computing the singular-value decomposition of a matrix with

TABLE II. Average test accuracies of the support vector machine with different datasets and graph kernels. The values in bold in the top and bottom sections of the table are the best accuracies obtained for that section. The values in parentheses are the standard deviations across the 10 repeats of double cross validation. GS, RW, SM, and SP refer to the graphlet-sampling, random-walk, subgraph-matching, and shortest-path kernels, respectively. $\mathrm{GBS}_{\nu}^{\mathrm{bin}}$ and $\mathrm{GBS}_{\mu}^{\mathrm{bin}}$ denote our GBS kernel with binary detectors that use the ν and μ coarse-graining strategies to construct the feature vectors, respectively. $\mathrm{GBS}_{\nu}^{\mathrm{bin+}}$ indicates that the feature associated with detecting vacuum [0,0,0,0,0] in the first five modes was dropped from all feature vectors. $\mathrm{GBS}^{\mathrm{PNR}}$ and $\mathrm{GBS}^{\mathrm{PNR+}}$ refer to the original GBS kernels with PNR detectors that use orbit and metaorbit probabilities as features, respectively, with a displacement of d on each mode.

Dataset	$\mathrm{GBS}_{\nu}^{\mathrm{bin+}}$	$\mathrm{GBS}_{\nu}^{\mathrm{bin}}$	$\mathrm{GBS}_{\mu}^{\mathrm{bin}}$	GS	RW	SM	SP
AIDS	98.47 (0.10)	98.74 (0.20)	**99.53** (0.05)	99.30 (0.07)	53.11 (11.90)	77.85 (2.44)	99.34 (0.09)
BZR_MD	60.14 (1.28)	61.73 (0.89)	58.79 (1.17)	51.42 (3.51)	**64.54** (0.36)	time out[a]	50.82 (1.76)
COX2_MD	**51.62** (2.76)	50.18 (2.96)	51.30 (3.86)	49.01 (3.18)	48.98 (4.78)	time out[a]	48.11 (4.30)
ENZYMES	**48.10** (1.18)	41.75 (2.35)	19.83 (1.43)	34.59 (2.54)	19.50 (2.29)	37.38 (1.60)	22.15 (1.88)
ER_MD	67.74 (0.94)	69.19 (0.33)	68.84 (0.50)	48.88 (4.53)	**70.32** (0.02)	time out[a]	45.23 (4.35)
FINGERPRINT	64.45 (0.78)	**65.53** (0.86)	63.56 (0.67)	65.25 (1.30)	33.63 (3.57)	46.89 (0.56)	46.22 (1.02)
IMDB-BINARY	60.69 (0.84)	61.35 (0.98)	67.34 (0.38)	**68.49** (0.63)	67.78 (0.38)	time out[a]	65.50 (0.27)
MUTAG	84.63 (0.91)	**85.94** (0.98)	81.37 (0.90)	80.80 (0.91)	83.22 (0.04)	83.24 (1.27)	82.74 (1.65)
NCI1	**63.45** (0.57)	56.99 (1.69)	59.09 (1.02)	50.34 (3.22)	50.96 (3.58)	time out[a]	53.40 (2.25)
PROTEINS	**65.95** (1.03)	63.38 (0.73)	63.11 (0.55)	65.75 (0.94)	56.91 (1.39)	62.93 (0.83)	63.63 (0.41)
PTC_FM	52.63 (3.95)	57.47 (2.72)	59.17 (1.58)	**60.74** (1.48)	50.95 (3.68)	56.36 (2.66)	55.38 (4.04)

Dataset	$\mathrm{GBS}^{\mathrm{PNR}}$ ($d=0$)	$\mathrm{GBS}^{\mathrm{PNR}}$ ($d=0.25$)	$\mathrm{GBS}^{\mathrm{PNR+}}$ ($d=0$)	$\mathrm{GBS}^{\mathrm{PNR+}}$ ($d=0.25$)
AIDS	99.60 (0.05)	**99.62** (0.03)	99.58 (0.06)	99.61 (0.05)
BZR_MD	62.73 (0.71)	62.13 (1.44)	62.01 (1.43)	**63.16** (2.11)
COX2_MD	44.98 (1.80)	50.11 (0.97)	57.84 (4.04)	**57.89** (2.62)
ENZYMES	22.29 (1.60)	28.01 (1.83)	25.72 (2.60)	**40.42** (2.02)
ER_MD	70.36 (0.78)	70.41 (0.47)	71.01 (1.26)	**71.05** (0.83)
FINGERPRINT	65.42 (0.49)	**65.85** (0.36)	66.19 (0.84)	66.26 (4.29)
IMDB-BINARY	64.09 (0.34)	**68.71** (0.59)	68.14 (0.71)	67.60 (0.75)
MUTAG	**86.41** (0.33)	85.58 (0.59)	85.64 (0.78)	84.46 (0.44)
NCI1	**63.61** (0.00)	62.79 (0.00)	63.59 (0.17)	63.11 (0.93)
PROTEINS	**66.88** (0.22)	66.14 (0.48)	65.73 (0.69)	66.16 (0.76)
PTC_FM	53.84 (0.96)	52.45 (1.78)	**59.14** (1.72)	56.25 (2.04)

[a]Run time $>$ 7 days.

complexity that is polylogarithmic in the size of the matrix do exist [29]. In particular, the quantum singular-value estimation algorithm for an $m \times n$ matrix presented in [30] has complexity $O(\mathrm{polylog}(mn)/\epsilon)$, where ϵ is an additive error.

IV. NUMERICAL EXPERIMENTS

A. Implementation details

We used THE WALRUS PYTHON library to classically sample from the GBS output distribution when running our experiments and the GRAKEL PYTHON library to fetch the datasets and simulate the classical graph kernels [31,32]. Classically sampling from a GBS output distribution is very time intensive even when using binary detectors, so we choose to follow the choice made in [3] and discard graphs with greater than 25 and fewer than 6 vertices for each dataset. Before sampling from the GBS device we have four parameters we can set: the maximum number of detector clicks allowed N, the average photon number $\bar{n}$, the displacement on each mode of the GBS device d, and, last, the number of samples generated by the GBS device S. We set $N = 6, \bar{n} = 5$, and $d = 0$ for our results reported here, leading to a probability distribution of 32 outcomes using the ν coarse-graining strategy and 7 outcomes using the μ coarse-graining strategy. Using Eq. (1) with $\delta = 0.01$ and $\epsilon = 0.06$, we require about $S = 15\,000$ samples for the ν feature vectors and about $S = 6000$ samples for the μ feature vectors.

For the machine learning classifier we use a support vector machine with an radial basis function (RBF) kernel κ_{rbf}. We obtain the accuracies in Table II by running a double 10-fold cross validation 10 times. The inner fold performs a grid search through the discrete set of values $[10^{-4}, 10^{-3}, \ldots, 10^{2}, 10^{3}]$ on the C hyperparameter of the SVM which controls the penalty on misclassifications. Last, due to the class imbalance present in some of the benchmark datasets we use class weights to assign more importance to training examples from classes that appear less frequently in the data, which is standard procedure in such cases.

B. Numerical results from GBS simulation and subsequent classification

We tested our graph kernel on the same datasets used in [3]. We also ignored vertex labels, vertex attributes, and edge attributes and converted all adjacency matrices to be unweighted.

Four classical graph kernels were used as a benchmark for our algorithm's classification accuracy: the subgraph-matching kernel with time complexity $O(kM^{k+1})$, where M

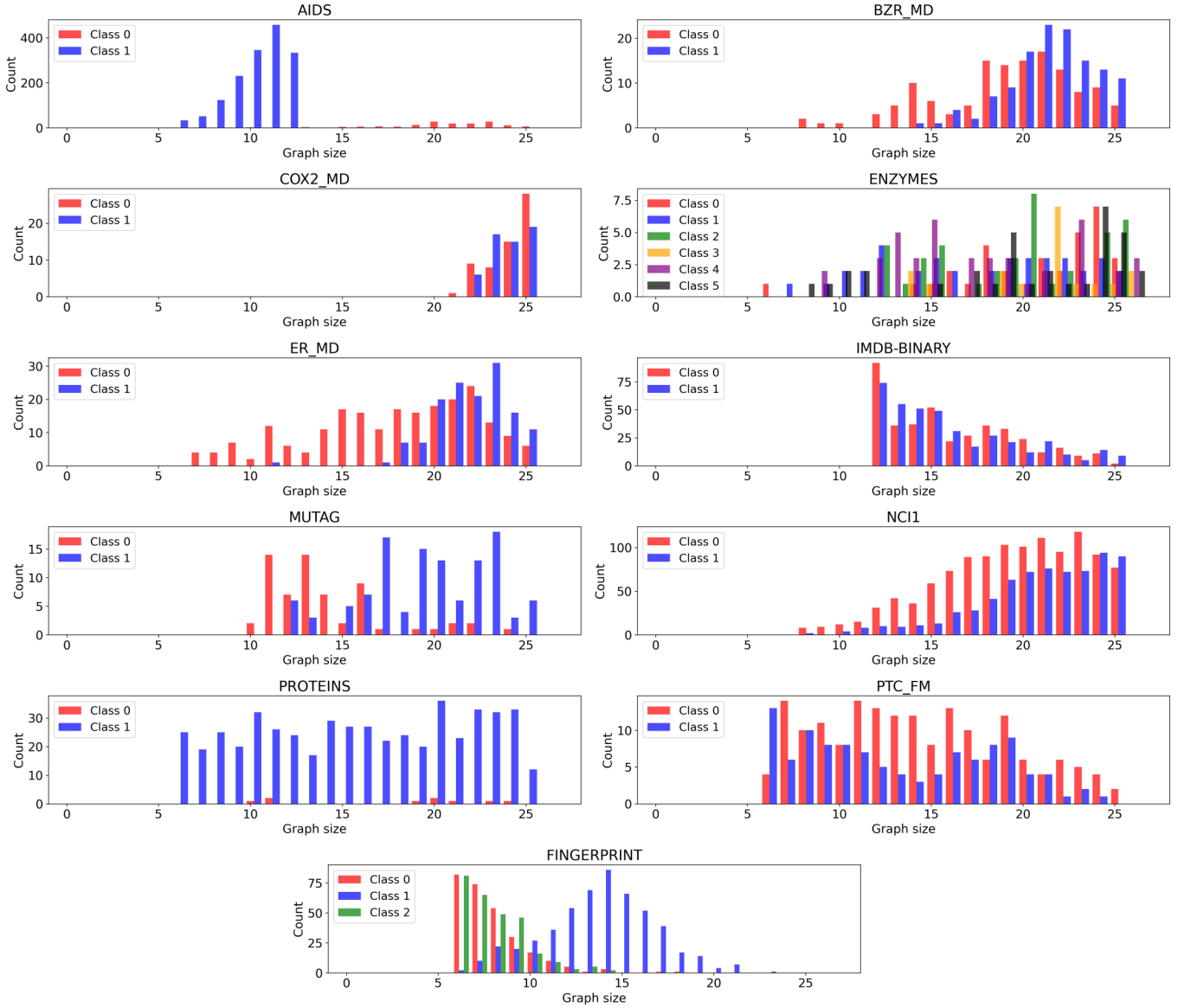


FIG. 4. Distribution of graph sizes according to class for each dataset.

is the number of vertices and k is the size of the subgraphs being considered [33]; the graphlet-sampling kernel with the worst-case time complexity $O(M^k)$, which can be optimized to $O(Md^{k-1})$ for graphs of bounded degree with the restriction that $k \in \{3, 4, 5\}$, where k is the graphlet size and d is the maximum degree of the graph [34]; the random-walk kernel with time complexity $O(M^3)$ [35]; and the shortest-path kernel with time complexity $O(M^4)$ [36]. For the graphlet-sampling kernel we set maximum graphlet size to $k = 5$ and draw 5174 samples; for the random-walk kernel we use fast computation and a geometric kernel type with the decay factor set to $\lambda = 10^{-3}$. For the subgraph-matching kernel we set maximum subgraph size to $k = 5$, and for the shortest-path kernel we used the Floyd-Warshall algorithm to calculate shortest paths. The accuracies of all four classical kernels and our kernel are shown in the top part of Table II. The values in the bottom part of Table II are the accuracies of the original GBS graph kernels and are taken from [3], where the feature vectors were constructed with $n = 6$. Some accuracies for the subgraph-matching kernel are not reported due to its $O(M^6)$ time complexity for $k = 5$, which required longer than 7 days of computation time for datasets with a high average number of vertices.

We can see from Table II that our kernel is very competitive with both the classical and PNR-based GBS graph kernels and, in fact, achieves the highest accuracy for the ENZYMES dataset. The PNR-based kernel obtains a test accuracy significantly higher than random guessing (≈16%) for ENZYMES only when displacement is applied. Our kernel can be seen as even more feasible in this regard since we do not require the extra operation of displacement to reach our level of accuracy. From Fig. 4 we see that for some datasets such as AIDS and MUTAG there is a strong size imbalance among graphs of different classes. For the AIDS dataset graphs belonging to class 1 are much smaller in size than those in class 0, while for MUTAG the converse is true. This size imbalance also exists to a lesser extent for the ER_MD, BZR_MD and PTC_FM datasets, and for the FINGERPRINT dataset class 1 has a

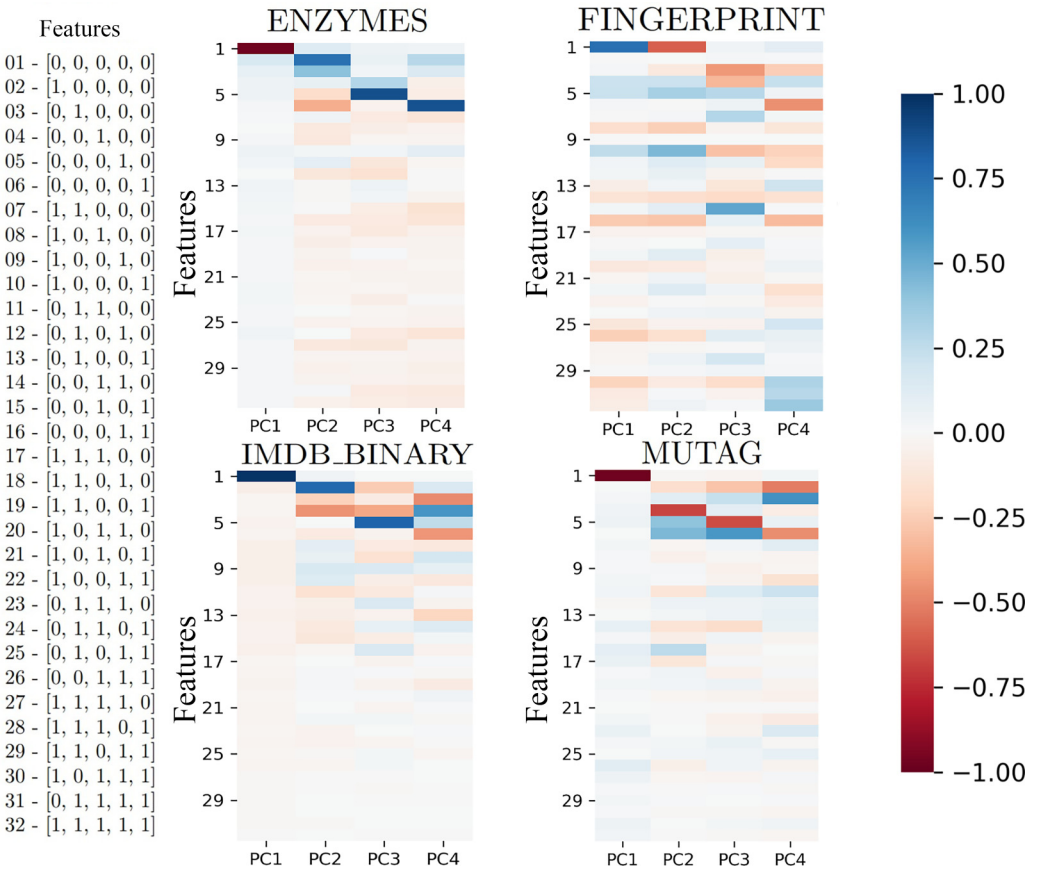


FIG. 5. Results of the principal component analysis (PCA) of the ν feature vector entries for the ENZYMES, MUTAG, IMDB_BINARY, and FINGERPRINT datasets. The heat maps show the weight or coefficient associated with each feature with regard to the first four principal components.

graph size distribution significantly different from the other two. Both our GBS kernel and the PNR-based kernel perform well on these size-imbalanced datasets, which indicates that graph size is a property that GBS-based kernels are sensitive to and that this sensitivity persists even when binary detectors are used.

C. Feature analysis

Figure 5 shows the results of performing a principal component analysis of the feature vectors generated using the ν coarse-graining strategy for various datasets. The analysis shows that the feature associated with vacuum [0,0,0,0,0] contributes by far the most in support of the first principal component. The analysis also suggests that in some cases the first 10 or so features contribute the most to the support of all of the first four principal components, but in other cases, such as that for FINGERPRINT, most features contribute more or less equally.

D. Comparison to classical kernels

Our graph kernel has a time complexity that is equivalent to the random-walk kernel and better than the shortest-path kernel by a factor of M while outperforming both on most datasets. Furthermore, the time complexity of our kernel is not exponential in the size of the subgraphs we are probing like the subgraph-matching kernel. The graphlet-sampling kernel does have a more favorable complexity of $O(Md^{k-1})$ for graphs with maximum degree d. However, it is important to note that many real-world graphs are what are called scale-free networks, and from the network-science literature [37] the maximum degree of these graphs grows polynomially with the graph size. Therefore, it is possible that the maximum degree of these graphs grows linearly with the graph size, i.e., $d \in O(M)$, which would lead to a complexity of $O(M^k)$ for the graphlet-sampling kernel. What is also interesting is that GBS kernels seems to provide more distinguishing power than some classical kernels for graphs with no vertex and edge labels like those used in our simulations. Take, for example, the ENZYMES dataset, for which the binary GBS kernel achieves a classification accuracy of $\approx$48% while the shortest-path kernel reaches about 23%. If we instead choose not to ignore vertex labels, we find the shortest-path kernel gives a classification accuracy of about 50%. Since the GBS features are related to Hafnians, this suggests that features related to the number of perfect matchings of a graph could be more useful for distinguishing graphs of different classes when one has no information about the attributes of the graph nodes.

V. CONCLUSION

We proposed a variation of an algorithm for the machine learning task of classification with graph-structured data that

uses a Gaussian boson sampler utilizing only binary detectors. We showed that our algorithm outperforms four classical graph kernels in the task of graph classification on many datasets. This is most evident with regard to the ENZYMES dataset, for which our ν feature map outperforms all methods. The feature corresponding to detecting vacuum in the first five modes plays a particularly important role, as shown by the principal component analysis, because it is related to the Hafnian of all possible subgraphs of G with their first five vertices removed. We also showed that the kernel is sample efficient, a major issue for applications of GBS, and has a time complexity that is comparable to those of the classical strategies.

The fact that a GBS kernel using only binary (light or no light) detectors produces such accuracies was a surprise to us, as it was far from intuitive and/or expected that binary detection could ever achieve a performance level comparable to—and, in some cases, better than—photon-number-resolving detection. Moreover, the binary photodetection approach relies on the technology of single-photon avalanche photodiodes, a technology which is well established, near room temperature, and cheap and can be extended to the near-infrared wavelengths compatible with low-loss integrated optics, thereby enabling compact on-chip implementations. This stands in stark contrast to photon-number-resolving detection, which requires expensive cryogenic superconductor-based systems.

A number of questions remain open for investigation such as how vertex and edge labels can be encoded in the GBS device. Also, as stated earlier, it is known that the existence of a polynomial-time classical algorithm for exact sampling from the output probability distribution of a boson sampling or Gaussian boson sampling device would imply the collapse of the polynomial hierarchy to the third level, and thus, the existence of such an algorithm is believed to be very unlikely [38]. This result can also be extended to GBS with binary detectors [26]. However, it is not known, even though some work has been done in this area [25], whether such arguments exist for algorithms that sample from coarse-grained versions of these probability distributions such as those defined in [3] or our work. It is important to know whether such arguments exist because they would imply these quantum kernels are also likely hard to simulate classically.

ACKNOWLEDGMENTS

We thank M. Schuld, K. Brádler, S. Aaronson, I. Cirac, M. Eaton, N. Quesada, A. Blance, and S. Maereg for useful advice and discussions. We thank Research Computing at the University of Virginia for providing access to, and support for, the Rivanna computing cluster. This work was supported by National Science Foundation under Grant No. PHY-2112867.

APPENDIX A: REMINDERS ABOUT STANDARD GBS

1. GBS with PNR detectors

Substantial work has been done already on the connection between graph theory and Gaussian boson sampling with PNR detectors [20–22]. Here we present the important concepts. Any undirected graph G with no self-loops and $|V| = M$ vertices can be encoded in an M-mode GBS setup consisting of a set of M squeezers followed by an interferometer of beam splitters according to its adjacency matrix A. Once the graph is encoded in the GBS device, the probability of detecting a specific detection event $\mathbf{n} = (n_1, \ldots, n_M)$ is equal to

$$p(\mathbf{n}) = \frac{1}{\sqrt{\det(Q)}} \frac{\mathrm{Haf}(\tilde{A}_\mathbf{n})}{\mathbf{n}!} = \frac{1}{\sqrt{\det(Q)}} \frac{\mathrm{Haf}^2(A_\mathbf{n})}{\mathbf{n}!}, \tag{A1}$$

with

$$Q = (\mathbb{I}_{2M} - X\tilde{A})^{-1}, \quad X = \begin{bmatrix} 0 & \mathbb{I} \\ \mathbb{I} & 0 \end{bmatrix}, \tag{A2}$$

$\mathbf{n}! = n_1! \times \cdots \times n_M!$, $\tilde{A} = (A \oplus A)$, and $\mathrm{Haf}(\cdot)$ denoting the Hafnian of a $2M \times 2M$ matrix. The Hafnian is a matrix function defined mathematically as

$$\mathrm{Haf}(A) = \sum_{\pi \in S_M} \prod_{(u,v)\in\pi} A_{u,v}, \tag{A3}$$

where S_M is the partition of the set $\{1, 2, \ldots, 2M\}$ into unordered disjoint pairs. For example, if $M = 2$, then $S_M = (\{(1, 2), (3, 4)\}, \{(1, 4), (2, 3)\}, \{(1, 3), (2, 4)\})$. If A is the adjacency matrix of an unweighted graph, then the Hafnian is equal to the number of perfect matchings of the vertices of the graph. A perfect matching is a partition of the vertex set of a graph into pairs such that each vertex is connected to exactly one edge from the edge set. All perfect matchings of a complete four-vertex graph are shown in Fig. 6.

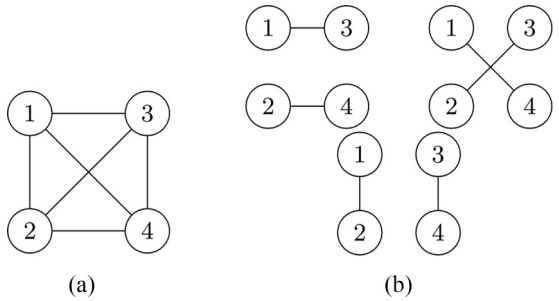


FIG. 6. (a) The complete graph of four vertices and (b) the three corresponding perfect matchings of the complete four-vertex graph.

$A_\mathbf{n}$ is the $n \times n$ submatrix of A induced according to the photon detection event $\mathbf{n}$. $A_\mathbf{n}$ is obtained by repeating the ith row and column according to the measurement pattern $\mathbf{n}$. If $n_i = 0$, then the ith row and column are deleted from A, but if $n_i > 0$, then the ith row and column are repeated n_i times. For example, the probability of detecting the event where each mode has exactly one photon $\mathbf{n} = (1, 1, \ldots, 1)$ would be proportional to the Hafnian of the original matrix A since $A_\mathbf{n} = A$. What this means in terms of the graph is that vertex i and all its edges are either deleted if $n_i = 0$ or duplicated n_i times if $n_i > 0$. Therefore, the probability of a detection event $\mathbf{n}$ is proportional to the squared Hafnian of the subgraph $G_\mathbf{n}$ corresponding to the induced adjacency matrix $A_\mathbf{n}$. Examples of different detection events and their corresponding induced subgraphs are shown in Fig. 7.

These induced subgraphs are of even size since the number of photons detected is always even due to the fact that the inputs are squeezed states. However, when displacement is applied to the modes of the GBS, the probability of detecting an

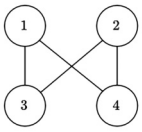
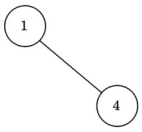
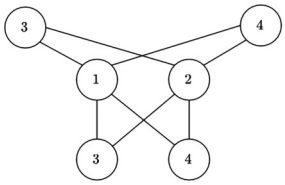
$\mathbf{n}$	$G_{\mathbf{n}}$	$\mathrm{Haf}^2(A_{\mathbf{n}})$
$(1,1,1,1)$		4
$(1,0,0,1)$		1
$(1,1,2,2)$		0

FIG. 7. Different photon detection events $\mathbf{n}$ and the corresponding subgraphs $G_{\mathbf{n}}$ they induce and the value of the squared Hafnians of those subgraphs. The probability of the detection event where each detector detects one photon corresponds to the Hafnian of the graph encoded in the GBS. We can see in the third graph from the top that when a detector detects two photons, the corresponding vertices and their respective edges are duplicated.

odd number of photons is, in general, not zero anymore, and the probability of individual detection events is characterized by the loop Hafnian lHaf$(\cdot)$ as opposed to the Hafnian [40,41]. We do not apply displacement for the numerical experiments done in this paper.

2. Encoding a graph into a GBS device

To map a graph to a feature vector we must first program the GBS device by setting the squeezing parameters and beam-splitter angles of the device according to the adjacency matrix A of the graph. Any adjacency matrix $A \in \mathbb{R}^{M\times M}$ of an undirected graph of M vertices can be mapped to a symmetric, positive-definite $2M \times 2M$ covariance matrix $\mathbf{\Sigma}$ of a pure Gaussian state of M modes via the following procedure. First, a doubled adjacency matrix $\tilde{A}$ is constructed,

$$\tilde{A} = c\begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} = c(A \oplus A), \tag{A4}$$

where c is a rescaling constant chosen such that $0 < c < 1/\lambda_{\max}$, where $\lambda_{\max}$ is the maximum singular value of A [3]. We use $\tilde{A}$ because, unlike A, it is guaranteed to map to a covariance matrix of a pure Gaussian state, which is easier to prepare than a mixed one [20]. This also has the advantage of allowing us to utilize the identity $\mathrm{Haf}(A \oplus A) = \mathrm{Haf}^2(A)$ to relate $\tilde{A}$ to A. To map $\tilde{A}$ to a covariance matrix $\mathbf{\Sigma}$ we use the following matrix equations:

$$\mathbf{\Sigma} = Q - \mathbb{I}_{2M}/2,\ Q = (\mathbb{I}_{2M} - X\tilde{A})^{-1}, \quad X = \begin{bmatrix} 0 & \mathbb{I} \\ \mathbb{I} & 0 \end{bmatrix}. \tag{A5}$$

To program the GBS device to sample from the probability distribution corresponding to the covariance matrix $\mathbf{\Sigma}$ of the pure Gaussian state we need the unitary matrix U that characterizes the interferometer of the device as well as the squeezing parameters $r_1, \ldots, r_M$ of each of the M squeezers. We can obtain these values by taking the Takagi decomposition of A, which is of the form

$$A = U\,\mathrm{diag}(\lambda_1, \ldots, \lambda_M)U^T. \tag{A6}$$

The squeezing parameters are determined by the singular values $\lambda_1, \ldots, \lambda_M$ and c via the relationship $r_i = \tanh^{-1}(c\lambda_i)$. The singular values and c also uniquely determine the mean photon number $\bar{n}$ of the device according to

$$\bar{n} = \sum_{i=1}^{M} \frac{(c\lambda_i)^2}{1-(c\lambda_i)^2} = \sum_{i=1}^{M} \sinh^2(r_i). \tag{A7}$$

The rescaling constant c can be used to adjust $\bar{n}$ because multiplying A by c scales its singular values without changing the structure of the graph other than scaling all edge weights by c. The matrix U can be decomposed to give the parameters of the beam-splitter gates of the interferometer [42].

The GBS device, if we are using PNR detectors, now samples from the probability distribution

$$p(\mathbf{n}) = \frac{1}{\sqrt{\det(Q)}}\frac{\mathrm{Haf}(\tilde{A}_{\mathbf{n}})}{\mathbf{n}!} = \frac{1}{\sqrt{\det(Q)}}\frac{\mathrm{Haf}^2(A_{\mathbf{n}})}{\mathbf{n}!}. \tag{A8}$$

APPENDIX B: SUPEREXPONENTIAL GROWTH OF GBS DETECTION EVENTS FOR $M \in O(n^2)$

Lemma 1. $\frac{(n+M-1)!}{n!(M-1)!} \in \omega(\lfloor\sqrt{M}\rfloor^{\lfloor\sqrt{M}\rfloor})$ for $n = \lfloor\sqrt{M}\rfloor$.

Proof.

$$\frac{(n+M-1)!}{n!(M-1)!} \xrightarrow{n=\lfloor\sqrt{M}\rfloor} \frac{(\lfloor\sqrt{M}\rfloor+M-1)!}{(\lfloor\sqrt{M}\rfloor)!(M-1)!},$$

$$\begin{aligned}
\frac{(\lfloor\sqrt{M}\rfloor+M-1)!}{(\lfloor\sqrt{M}\rfloor)!(M-1)!} &= \frac{\left[\prod_{i=1}^{\lfloor\sqrt{M}\rfloor}(M-1+i)\right](M-1)!}{\left[\prod_{i=1}^{\lfloor\sqrt{M}\rfloor} i\right](M-1)!} \\
&= \frac{\left[\prod_{i=1}^{\lfloor\sqrt{M}\rfloor}(M-1+i)\right]}{\left[\prod_{i=1}^{\lfloor\sqrt{M}\rfloor} i\right]} \\
&= \prod_{i=1}^{\lfloor\sqrt{M}\rfloor}\left[\frac{M-1}{i}+1\right] \\
&> \prod_{i=1}^{\lfloor\sqrt{M}\rfloor}\left[\frac{M-1}{\lfloor\sqrt{M}\rfloor}+1\right] \\
&= \left(\frac{M-1}{\lfloor\sqrt{M}\rfloor}+1\right)^{\lfloor\sqrt{M}\rfloor} \\
&= \left(\frac{M}{\lfloor\sqrt{M}\rfloor}+1-\frac{1}{\lfloor\sqrt{M}\rfloor}\right)^{\lfloor\sqrt{M}\rfloor} \\
&\geqslant \lfloor\sqrt{M}\rfloor^{\lfloor\sqrt{M}\rfloor}.
\end{aligned}$$

Therefore, $\frac{(n+M-1)!}{n!(M-1)!} \in \omega(\lfloor\sqrt{M}\rfloor^{\lfloor\sqrt{M}\rfloor})$ for $n = \lfloor\sqrt{M}\rfloor$. ■.

APPENDIX C: INDUCTION PROOF FOR $\binom{n}{k} \in \Theta(n^k)$

Lemma 2. $\binom{n}{k} \in \Theta(n^k)$.

Proof. For the base case $k = 2$,

$$\binom{n}{2} = \frac{n(n-1)}{2!},$$

$$\lim_{n\to\infty} \frac{\frac{n(n-1)}{2!}}{n^2} = \frac{1}{2!},$$

$$0 < \frac{1}{2!} < \infty,$$

$$\therefore \binom{n}{2} \in \Theta(n^2).$$

Assume the result holds up to $k = \ell$:

$$\binom{n}{\ell} = \frac{n(n-1)(n-2)\cdots(n-\ell+1)}{\ell!} \in \Theta(n^\ell).$$

In the inductive step, $k = \ell + 1$:

$$\binom{n}{\ell+1} = \frac{n(n-1)(n-2)\cdots(n-\ell)}{(\ell+1)!},$$

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\frac{n(n-1)(n-2)\cdots(n-\ell)}{(\ell+1)!}}{n^{\ell+1}} \\
&= \lim_{n\to\infty} \frac{\frac{n(n-1)(n-2)\cdots(n-\ell+1)}{\ell!}}{n^\ell} \frac{\frac{(n-\ell)}{\ell+1}}{n} \\
&= \lim_{n\to\infty} \frac{\frac{n(n-1)(n-2)\cdots(n-\ell+1)}{\ell!}}{n^\ell} \lim_{n\to\infty} \frac{\frac{(n-\ell)}{\ell+1}}{n} \\
&\equiv \frac{1}{\ell!}\frac{1}{\ell+1} \\
&= \frac{1}{(\ell+1)!}, \quad 0 < \frac{1}{(\ell+1)!} < \infty, \\
&\therefore \binom{n}{\ell+1} \in \Theta(n^{\ell+1}).
\end{aligned}$$

■

[1] G. Nikolentzos, G. Siglidis, and M. Vazirgiannis, Graph kernels: A survey, J. Artif. Intell. Res. **72**, 943 (2021).

[2] N. M. Kriege, F. D. Johansson, and C. Morris, A survey on graph kernels, Appl. Network Sci. **5**, 6 (2020).

[3] M. Schuld, K. Brádler, R. Israel, D. Su, and B. Gupt, Measuring the similarity of graphs with a Gaussian boson sampler, Phys. Rev. A **101**, 032314 (2020).

[4] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, Electronic Colloquium on Computational Complexity, https://eccc.weizmann.ac.il.

[5] L. G. Valiant, The complexity of computing the permanent, Theor. Comput. Sci. **8**, 189 (1979).

[6] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Gaussian boson sampling, Phys. Rev. Lett. **119**, 170501 (2017).

[7] H.-A. Bachor and T. C. Ralph, *A Guide to Experiments in Quantum Optics*, 3rd ed. (Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2019).

[8] H.-S. Zhong *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[9] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, Quantum computational advantage with a programmable photonic processor, Nature (London) **606**, 75 (2022).

[10] F. Arute *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[11] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, Inequalities for the l_1 deviation of the empirical distribution, Hewlett-Packard, Technical Report No. HPL-2003-97, R1, 2003 (unpublished).

[12] Note that this is not related to the number of possible output quantum states, which scales with the number of parameters governing the quantum evolution, e.g., the parameters of a simulated Hamiltonian. Obviously, no quantum advantage can be obtained for M-qubit Hamiltonians that have $O(2^M)$ parameters, but all classically intractable M-qubit Hamiltonians of physical interest are local and have parameter numbers polynomial in M [13], which validates Feynman's proposed advantage for quantum simulation [14]. An $M \times M$ optical interferometer has M^2 parameters, for example. However, even though any useful quantum computer will explore only an $O(M^k)$-dimensional region of an $O(2^M)$-dimensional Hilbert space, the number of measurement outcomes will still scale like $O(2^M)$ *a priori*, simply because we do not know the adequate measurement basis that best contains the $O(M^k)$ output states. This is the well-known exponential overhead of quantum state tomography.

[13] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[14] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[15] A. E. Lita, A. J. Miller, and S. W. Nam, Counting near-infrared single-photons with 95% efficiency, Opt. Express **16**, 3032 (2008).

[16] C. Cahall, K. L. Nicolich, N. T. Islam, G. P. Lafyatis, A. J. Miller, D. J. Gauthier, and J. Kim, Multi-photon detection using a conventional superconducting nanowire single-photon detector, Optica **4**, 1534 (2017).

[17] M. Eaton, A. Hossameldin, R. J. Birrittella, P. M. Alsing, C. C. Gerry, H. Dong, C. Cuevas, and O. Pfister, Resolution of 100 photons and quantum generation of unbiased random numbers, Nat. Photonics **17**, 106 (2023).

[18] R. Cheng, Y. Zhou, S. Wang, M. Shen, T. Taher, and H. X. Tang, A 100-pixel photon-number-resolving detector unveiling photon statistics, Nat. Photonics **17**, 112 (2023).

[19] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Gaussian quantum information, Rev. Mod. Phys. **84**, 621 (2012).

[20] K. Brádler, P.-L. Dallaire-Demers, P. Rebentrost, D. Su, and C. Weedbrook, Gaussian boson sampling for perfect matchings of arbitrary graphs, Phys. Rev. A **98**, 032310 (2018).

[21] K. Brádler, S. Friedland, J. Izaac, N. Killoran, and D. Su, Graph isomorphism and Gaussian boson sampling, Spec. Matrices **9**, 166 (2021).

[22] J. M. Arrazola and T. R. Bromley, Using Gaussian boson sampling to find dense subgraphs, Phys. Rev. Lett. **121**, 030503 (2018).

[23] C. L. Canonne, A short note on learning discrete distributions, arXiv:2002.11457.

[24] A. Y. Oruç, On number of partitions of an integer into a fixed number of positive integers, J. Number Theory **159**, 355 (2016).

[25] K. Bradler, R. Israel, M. Schuld, and D. Su, A duality at the heart of Gaussian boson sampling, arXiv:1910.04022.

[26] N. Quesada, J. M. Arrazola, and N. Killoran, Gaussian boson sampling using threshold detectors, Phys. Rev. A **98**, 062322 (2018).

[27] The Torontonian was also shown to be a generating function for the Hafnian given by $\mathrm{Haf}(A) = \frac{1}{M!}\frac{d^M}{dz^M}\mathrm{Tor}(zXA)|_{z=0}$, where A is a $2M \times 2M$ matrix.

[28] T. Hahn, Routines for the diagonalization of complex matrices, arXiv:physics/0607103.

[29] L. Gu, X. Wang, and G. Zhang, Quantum higher order singular value decomposition, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)* (IEEE, Piscataway, NJ, 2019), pp. 1166–1171.

[30] I. Kerenidis and A. Prakash, Quantum recommendation systems, arXiv:1603.08675.

[31] B. Gupt, J. Izaac, and N. Quesada, The Walrus: A library for the calculation of Hafnians, Hermite polynomials and Gaussian boson sampling, J. Open Source Software **4**, 1705 (2019).

[32] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis, GraKeL: A graph kernel library in python, J. Mach. Learn. Res. **21**, 1 (2020).

[33] N. Kriege and P. Mutzel, Subgraph matching kernels for attributed graphs, arXiv:1206.6483.

[34] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, Efficient graphlet kernels for large graph comparison, in *Artificial Intelligence and Statistics* (PMLR, Clearwater Beach, Florida, 2009), pp. 488–495.

[35] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, Graph kernels, J. Mach. Learn. Res. **11**, 1201 (2010).

[36] K. M. Borgwardt and H.-P. Kriegel, Shortest-path kernels on graphs, in *Fifth IEEE International Conference on Data Mining (ICDM'05)* (IEEE, Piscataway, NJ, 2005), pp. 74–81.

[37] A.-L. Barabási and M. Pósfai, *Network Science* (Cambridge University Press, Cambridge, 2016).

[38] Although this has been proven rigorously for the exact sampling case [4,39], the proof pertaining to the approximate sampling case rests on the assumption of two conjectures known as the permanent-of-Gaussians conjecture and the permanent anticoncentration conjecture, which are, as of now, still unproven.

[39] R. Kruse, C. S. Hamilton, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Detailed study of Gaussian boson sampling, Phys. Rev. A **100**, 032326 (2019).

[40] N. Quesada, Franck-Condon factors by counting perfect matchings of graphs with loops, J. Chem. Phys. **150**, 164113 (2019).

[41] J. F. F. Bulmer, S. Paesani, R. S. Chadwick, and N. Quesada, Threshold detection statistics of bosonic states, Phys. Rev. A **106**, 043712 (2022).

[42] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optimal design for universal multiport interferometers, Optica **3**, 1460 (2016).