

# A quantum hardware-induced graph kernel based on Gaussian Boson Sampling

Maria Schuld,<sup>1,\*</sup> Kamil Brádler,<sup>1</sup> Robert Israel,<sup>1</sup> Daiqin Su,<sup>1</sup> and Brajesh Gupt<sup>1</sup>

<sup>1</sup>*Xanadu, Toronto, Canada*

(Dated: May 31, 2019)

A device called a ‘Gaussian Boson Sampler’ has initially been proposed as a near-term demonstration of classically intractable quantum computation. As recently shown, it can also be used to decide whether two graphs are isomorphic. Based on these results, we construct a feature map and graph similarity measure or ‘graph kernel’ using samples from the device. We show that the kernel performs well compared to standard graph kernels on typical benchmark datasets, and provide a theoretical motivation for this success, linking the distribution of a Gaussian Boson Sampler to the number of matchings in subgraphs. Our results contribute to a new way of thinking about kernels as a (quantum) hardware-efficient feature mapping, and lead to an interesting application for near-term quantum computing.

## I. INTRODUCTION

Machine learning algorithms can be understood as a way to determine similarity between data points, mapping inputs to similar classes or clusters if they show matching patterns. But how can we learn from data where it is even difficult to determine if two data points are the same? Graphs are such data structures. A graph  $G = \{V, E\}$  can be described by a set of nodes  $V$  and a set of edges  $E$  that are pairwise connections between nodes. An equivalent representation is an adjacency matrix  $A$  whose element  $A_{ij}$  is one if node  $i$  and  $j$  are connected by an edge and zero otherwise. Both representations are not unique, since different ways of identifying the nodes can give rise to the same graph.<sup>1</sup>

There are three major strategies to classify<sup>2</sup> graph-structured data: First, one can construct a so called *graph neural network* that mirrors the connectivity of the graph [4]. Second, one can extract a feature vector from each graph and run a conventional machine learning algorithm on the new data set of feature vectors [5]. The feature vector can be an *embedding* of the graph in Euclidean space while preserving a property of choice [6–8], or it can be a higher-level representation of the graph [9]. Third, one can compute a *graph kernel* [10], a similarity measure between graphs which can be used in kernel methods [11]. Sometimes the third method draws on the second and extracts feature vectors which are then used in a standard kernel [12]. In this paper we follow this strategy, but suggest a new way to map graphs to feature vectors: by making use of quantum hardware. Hardware-induced feature maps have been proposed in

the context of random features [13], and can lead to very efficient implementations. Kernels computed by quantum computers have been prominently suggested in [14, 15], where the question of concrete applications was left wide open. Here we provide an example of a “quantum kernel” for graph-structured data based on the technique of Gaussian Boson Sampling (GBS) [16–18]. GBS is a generalization of Boson Sampling [19, 20], which has originally been proposed as a classically intractable task that demonstrates the power of near-term quantum hardware [21].

As we will explain in more detail in Sec. II, an optical GBS device processes information using the continuous degrees of freedom of a quantum system, realized by the modes of the electromagnetic field. It has been previously shown that the photon measurement statistics of a GBS device that encodes a graph in its optical state give rise to a complete set of graph isomorphism invariants [22]. Here we demonstrate that these results are highly relevant for machine learning by interpreting these statistics as a feature vector that can be further processed by both kernel-based and other machine learning methods. Sec. III shows that the “GBS feature map” works similarly to a class of graph kernels which count subgraphs [9, 23]). The probabilities of drawing samples from the quantum device contain information about the number of perfect matchings in all subgraphs, as well as in extended subgraphs created by duplicating nodes and edges. The features extracted from the graph are therefore related to so called  $r$ -matchings. Another interpretation associates the features with higher-order moments of a multivariate Gaussian distribution whose covariance matrix is proportional to the doubled adjacency matrix of the graph. Numeric experiments presented in Sec. IV reveal that the GBS kernel yields very competitive results compared to other graph kernels when used in combination with a support vector machine on standard benchmark datasets; it also motivates the importance of low-photon events and an optical mechanism called *displacement*.

---

\* maria@xanadu.ai

<sup>1</sup> The complexity of deciding whether two graphs are isomorphic is unknown; neither a polynomial-time algorithm nor NP-completeness proof has been discovered yet [1]. Many algorithms (for example, [2]) perform reasonably well in their average time complexity, but may take time exponential in the graph size in some instances.

<sup>2</sup> Other machine learning tasks look at how to predict new nodes and edges in a graph, see for example, [3].

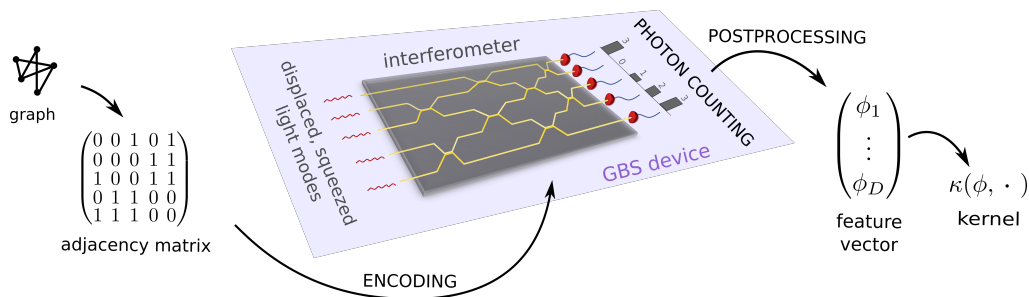


FIG. 1. Idea of the quantum hardware-induced feature map. A graph’s adjacency matrix is embedded into the interferometer of a Gaussian Boson Sampler (GBS). The probabilities of measuring certain photon counting events are interpreted as a feature vector in  $\mathbb{R}^D$ , which gives rise to a kernel.

## II. GAUSSIAN BOSON SAMPLING AS A FEATURE MAP

In this section we describe the mathematical details of the quantum hardware-induced feature map (see also Figure 1). The scheme works for *simple graphs*, i.e. undirected graphs without self-loops or multiple edges. While edge weights can be treated on the same footing as unweighted edges, we leave the inclusion of categorical edge labels or node labels for future studies. Mindful of potential readers without a technical background in quantum mechanics we will only highlight some important aspects of Gaussian Boson Sampling and refer to Refs. [16, 18, 24] for more detail. The GBS kernel is summarized in Eq. (6).

### A. Encoding graphs into the GBS device

An optical Gaussian Boson Sampler is a device where a special quantum state (a so-called *Gaussian state*) is prepared by the *optical squeezing* of  $M$  displaced light modes, followed by an *interferometer* of beamsplitters. Such a Gaussian state is fully described by a covariance matrix  $\sigma \in \mathbb{R}^{2M \times 2M}$  as well as a displacement vector  $\mathbf{d} \in \mathbb{R}^{2M}$  [25]. Photon number resolving detectors count the photons in each mode.

As outlined in [24], a quantum state prepared by a GBS device can encode a graph  $G = (V, E)$  with an adjacency matrix  $A$  of entries  $A_{ij}$  that are one if the edge  $(i, j)$  exists in  $G$  and zero else. The entries of  $A$  can also represent continuous “edge weights” that denote the strength of a connection. In the latter case we will speak of a “weighted adjacency matrix”.

In order to associate  $A$  with the symmetric, positive definite  $2M$ -dimensional covariance matrix of a Gaussian state of  $M$  modes, we have to construct a “doubled adjacency matrix”

$$\tilde{A} = c \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} = c(A \oplus A), \quad (1)$$

where the rescaling constant  $c$  is chosen so that  $0 < c < 1/s_{\max}$ , and  $s_{\max}$  is the maximum singular value

of  $A$  [22, 24].<sup>3</sup> For simplicity we will always rescale all adjacency matrices with a factor  $1/(s_{\max}^{\{G\}} + 10^{-8})$  where  $s_{\max}^{\{G\}}$  is the largest singular value among all graphs in the data set under consideration. As a result we will assume that  $c = 1$  and  $\tilde{A} = A \oplus A$  can be encoded into a GBS device. We call this the “doubled encoding strategy”.

The matrix  $\tilde{A}$  can now be associated with a quantum state’s covariance matrix  $\sigma$  by setting the squeezing as well as the beamsplitter angles of the interferometer so that

$$\sigma = Q - \mathbb{1}/2, \quad \text{with } Q = (\mathbb{1} - X\tilde{A})^{-1}, \quad X = \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix}. \quad (2)$$

### B. Photon counting

After embedding  $A$  via  $\tilde{A}$  into the quantum state of the GBS, each measurement of the photon number resolving detectors returns a photon event  $\mathbf{n} = [n_1, \dots, n_M]$ , with  $n_i \in \mathbb{N}$  indicating the number of photons measured in the  $i$ -th mode. Assuming for now that the displacement  $\mathbf{d}$  is zero (nonzero displacement is discussed in Appendix A), the probability of measuring a given photon counting event is

$$p(\mathbf{n}) = \frac{1}{\sqrt{\det(Q)} \mathbf{n}!} \text{Haf}^2(A_{\mathbf{n}}), \quad (3)$$

where  $\mathbf{n}! = n_1!n_2! \dots n_M!$ .

Let us go through this nontrivial equation bit by bit. The Hafnian  $\text{Haf}()$  is a matrix operation similar to the determinant or permanent. For a general symmetric matrix  $C \in \mathbb{R}^N \times \mathbb{R}^N$  it reads

$$\text{Haf}(C) = \sum_{\pi \in P_N^{\{2\}}} \prod_{(u,v) \in \pi} C_{u,v}. \quad (4)$$

<sup>3</sup> As long as it fulfills the above inequality,  $c$  can be treated as a hyperparameter of the feature map, which may also be influenced by hardware constraints since it relates ultimately to the amount of squeezing required.

$O_{\mathbf{n}^*}$	$\mathbf{n}$	$ \mathbf{n} $	$G_{\mathbf{n}}$	Haf( $A_{\mathbf{n}}$ )
$O_{[0,0,0]}$	[0, 0, 0]	0		0
	[1, 0, 0]			0
$O_{[1,0,0]}$	[0, 1, 0]	1		0
	[0, 0, 1]			0
$O_{[1,1,0]}$	[1, 1, 0]			1
	[1, 0, 1]	2		1
	[0, 1, 1]			1
$O_{[2,0,0]}$	[2, 0, 0]			0
	[0, 2, 0]	2		0
	[0, 0, 2]			0

$O_{\mathbf{n}^*}$	$\mathbf{n}$	$ \mathbf{n} $	$G_{\mathbf{n}}$	Haf( $A_{\mathbf{n}}$ )
$O_{[1,1,1]}$	[1, 1, 1]	3		0
	[2, 1, 0]			0
	[2, 0, 1]			0
$O_{[2,1,0]}$	[1, 2, 0]	3		0
	[1, 0, 2]			0
	[0, 2, 1]			0
	[0, 1, 2]			0
	[3, 0, 0]			0
$O_{[3,0,0]}$	[0, 3, 0]	3		0
	[0, 0, 3]			0

TABLE I. Photon events  $\mathbf{n}$ , total photon number  $|\mathbf{n}|$ , extended induced subgraph  $G_{\mathbf{n}}$  (indicated by red/black nodes and edges) and Hafnian Haf( $A_{\mathbf{n}}$ ) for the orbits of an original fully connected simple graph of three nodes, and up to  $|\mathbf{n}|_{\max} = 3$ . The Hafnian corresponds to the number of perfect matchings in the extended induced subgraph. Note that the red nodes are not mutually connected.

Here,  $P_N^{\{2\}}$  is the set of all  $N!/((N/2)!2^{N/2})$  ways to partition the index set  $\{1, 2, \dots, N\}$  into  $N/2$  unordered pairs of size 2, such that each index only appears in one pair. The Hafnian is zero for odd  $N$ . As an example, for the index set  $\{1, 2, 3, 4\}$  we have  $P_4^{\{2\}} = \{(1, 2), (3, 4)\}, \{(1, 3), (2, 4)\}, \{(1, 4), (2, 3)\}$ .

If  $C$  is interpreted as an adjacency matrix containing the edges of a graph, the set  $P_N^{\{2\}}$  contains edge-sets of all possible perfect matchings on  $G$ . A perfect matching is a subset of edges such that every node is covered by exactly one of the edges. The Hafnian therefore sums the products of the edge weights in all perfect matchings. If all edge weights are constant, it simply counts the number of perfect matchings in  $G$ . Note that in Eq. (3) we used the fact that for real and symmetric  $A$ ,  $\text{Haf}(\tilde{A}) = \text{Haf}(A \oplus A) = \text{Haf}^2(A)$ . In other words, the doubled encoding strategy leads to a square factor which will play a profound role in the quantum feature map we are aiming to construct.

Eq. (3) does not depend on the Hafnian of the adjacency matrix  $A$ , but on a matrix  $A_{\mathbf{n}}$ .  $A_{\mathbf{n}}$  contains  $n_j$  duplicates of the  $j$ th row and column in  $A$ . If  $n_j = 0$ , the  $j$ th row/column in  $A$  does not appear in  $A_{\mathbf{n}}$ . Effectively, this constructs a new graph  $G_{\mathbf{n}}$  from  $A$  according to the following rules (see also Table I):

1. If all  $n_j, j = 1, \dots, M$  are one (i.e., each detector counted exactly one photon),  $A_{\mathbf{n}} = A$ .
2. If some  $n_j$  are zero and others one (i.e., these detectors report no photons),  $A_{\mathbf{n}}$  describes an *induced subgraph*  $G_{\mathbf{n}}$  of  $G$ , in which nodes that correspond to detectors with zero count were deleted together with any edge that connected them to other nodes.

3. If some  $n_j$  are larger than one (i.e., these detectors count more than one photon),  $A_{\mathbf{n}}$  describes what we call an *extended induced subgraph* in which the corresponding nodes and all their connections are duplicated  $n_j$  times.

In short, the probability of a photon event to be measured by the GBS device is proportional to the square of the (weighted) number of perfect matchings in a -possibly extended - induced subgraph of the graph encoded into the interferometer.

Computing the Hafnian of a general matrix is in complexity class #P, and formally reduces to the task of computing permanents [26]. If no entry in the matrix is negative, efficient approximation heuristics are known, although their success is only guaranteed under specific circumstances [27, 28].

### C. Post-processing and features

We follow Brádler et al.'s [22] fruitful strategy to “coarse-grain” the distribution of photon counting events by summarizing them to sets called *orbits* (see Table I). An orbit  $O_{\mathbf{n}^*} = \{\text{perm}(\mathbf{n}^*)\}$  contains permutations of the detection event  $\mathbf{n}^*$ . For example,  $[2, 1, 1, 0]$  is in the same orbit as  $[0, 1, 2, 1]$ , but not  $[2, 2, 0, 0]$ . The photon counting event  $\mathbf{n}^*$  is therefore an arbitrary “representative” of the photon counting events in an orbit. The probability of detecting a photon counting event of orbit  $O_{\mathbf{n}^*}$  is given by the sum of the individual probabilities,

$$p(O_{\mathbf{n}^*}) = \sum_{\mathbf{n} \in O_{\mathbf{n}^*}} p(\mathbf{n}). \quad (5)$$

The number of orbits  $O_{\mathbf{n}}$  containing events of up to  $k$  photons in total is equal to the number of ways that the integers of  $1, \dots, k$  can be partitioned into a sum of at most  $M$  terms. In practice we usually have  $k \ll M$ , in which case there are 2, 4, 7, 12, 19, 30, 45, 67 orbits for  $k = 1, \dots, 8$ , respectively<sup>4</sup>. In a real GBS setup, the energy is finite and high photon counts therefore become very unlikely.<sup>5</sup> It is only a natural choice to limit the maximum number of photons  $|\mathbf{n}|_{\max}$  to a constant  $k$  and call  $\mathcal{O}^{k,M}$  the set of all orbits where  $M$  detectors detect  $k$  photons or less.

In practice, the orbit probabilities are estimated by drawing a sufficiently large number of samples of photon counting events from the GBS device. In Ref [9] we find that we can approximate a probability distribution on  $D$  possible outcomes, with probability at most  $\delta$  that the sum of absolute values of the errors in the empirical probabilities of the outcomes is  $\epsilon$  or more, using

$$S = \left\lceil \frac{2(\log(2)D + \log(\frac{1}{\delta}))}{\epsilon^2} \right\rceil$$

samples. For  $k = 8$ ,  $D = 67$ ,  $\epsilon = 0.05$  and  $\delta = 0.05$ , we need 39,550 samples. Since current-day photon number resolving detectors can accumulate about  $10^5$  samples of photon counting events per second [29], it takes in principle only a fraction of a second for the orbit probabilities to be estimated by the physical hardware, irrespective of the graph size.

While hardware implementations of Gaussian Boson Samplers are rapidly advancing, in this paper we still have to resort to simulations. Sampling from photon event distributions is still a topic of active research, and to ensure that the results are not influenced by approximation errors we will use exact calculations here. This limits the scope of the experiments to graphs of the order of 25 nodes.

#### D. Feature map and graph kernel

Summarizing the above, the feature map implemented by a GBS device maps a graph to a feature vector,  $G \rightarrow \mathbf{f} \in \mathbb{R}^D$ , where the entries  $f_i, i = 1, \dots, D$  of  $\mathbf{f}$  are the probabilities of detecting certain types of photon events that we called orbits,

$$f_i = p(O_{\mathbf{n}_i^*}), \quad (6)$$

and the probability of the  $i$ 'th orbit is fully defined by Eqs. (5) and (3) (the ordering of the orbits does not

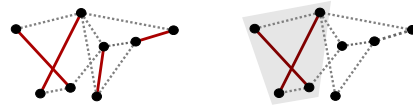


FIG. 2. Example of a perfect matching (left) and a 2-matching (right). The 2-matching is at the same time a perfect matching of the subgraph highlighted in grey.

matter). The relation of these features to those of a Graphlet Sampling kernel are highlighted in Appendix B.

Assuming that  $k \leq M$  for all graphs,  $D$  is solely determined by the maximum photon number  $k$ , which is a hyperparameter of the feature map. Two more hyperparameters are the displacement that can be applied to the light modes, as well as the amount of squeezing which is determined by the normalization constant  $c$ . We will assume here that the displacement applied to all modes is a constant  $d$ .

The feature vectors can be used as inputs to common machine learning models such as neural networks. Here we are interested in constructing a similarity measure or kernel that computes the similarity between two graphs  $G$  and  $G'$ . In the simplest case of a ‘linear’ and ‘rbf’ kernel we get (with a hyperparameter  $\delta$ )

$$\begin{aligned} \kappa_{\text{lin}}(G, G') &= \langle \mathbf{f}, \mathbf{f}' \rangle, \\ \kappa_{\text{rbf}}(G, G') &= \exp\left(-\frac{\|\mathbf{f} - \mathbf{f}'\|^2}{2\delta^2}\right), \end{aligned}$$

both of which are well known to be positive semi-definite so that the results of kernel theory apply to the ‘‘GBS kernel’’ constructed here.

### III. INTERPRETING THE GBS GRAPH FEATURES

The previous section suggested that the GBS feature map is related to graph properties such as matchings. In this section we will analyze the features in more detail.

#### A. Single-photon features and $r$ -matchings

It turns out that the probabilities of ‘single-photon’ orbits (i.e., each detector counts either zero or one photon) are related to a graph property called the ‘‘matching polynomial’’ of  $G$  [30–32],

$$\mu(G) = \sum_{r=0}^{\lceil M/2 \rceil} (-1)^r m(G, r) x^{M-2r}. \quad (7)$$

The coefficients  $m(G, r)$  of the matching polynomial count the number of  $r$ -matchings or ‘‘independent edge sets’’ in  $G$  – sets of  $r$  edges that have no vertex in common (see Figure 2). In our language, this could be

<sup>4</sup> See also A000070 in the Online Encyclopedia of Integer Sequences, <https://oeis.org/A000070>.

<sup>5</sup> The energy of a Gaussian quantum state, and hence the average photon number, is determined by the squeezing and displacement operations.

written as  $m(G, r) = \sum_{\mathbf{n} \in O_{[1, \dots, 1, 0, \dots]}} \text{Haf}(A_{\mathbf{n}})$  (where  $[1, \dots, 1, 0, \dots]$  has  $2r$  single photon detections). Hence, if it were not for the square of the Hafnian in Eq. (3), the probability  $p(O_{\mathbf{n}^*})$  of a single-photon orbit would be proportional to an  $|\mathbf{n}|/2$ -matching  $m(G, |\mathbf{n}|/2)$  of  $G$ . The square gives rise to a new object

$$g(G, r) = \sum_{\mathbf{n} \in O_{[1, \dots, 1, 0, \dots]}} \text{Haf}^2(A_{\mathbf{n}}).$$

Replacing  $m$  with  $g$  in Eq. (7) leads to a new type of polynomial  $\gamma(G)$  which one could call a *GBS polynomial*.

An interesting observation occurs for the feature corresponding to orbit  $O_{[1, 1, 0, \dots]}$  (see for example Table I). Since there are only two options – the two nodes are connected and have therefore exactly one perfect matching, or they are not and have none – the square does not have any effect, and the probability of the orbit is proportional to the number of 1-matchings of this graph, which is in turn equal to its number of edges. Hence, we have that  $p(O_{[1, 1, 0, \dots]}) \propto |E|$ , and the hardware natively returns an “edge counting” feature.

### B. Higher-order moments

The probability of measuring a given photon counting event  $\mathbf{n} = [n_1, \dots, n_M]$  can also be interpreted from a slightly different, more physically motivated viewpoint. The  $M$  nodes of a graph can be associated with  $M$  random variables drawn from a multivariate normal distribution  $N(\xi, \Sigma)$ , where the covariance matrix  $\Sigma$  corresponds to the doubled adjacency matrix  $\tilde{A}$ , and  $\xi$  is the mean vector related to displacement via  $\xi = Q^{-1} \mathbf{d}^\dagger$ . The higher-order moments  $E[X_1^{(1)} \dots X_1^{(n_1)} \dots X_M^{(1)} \dots X_M^{(n_M)}]$  of this distribution are proportional to  $\text{Haf}(A_{\mathbf{n}})$ , which in turn is related to the probability of a photon event via Eq. (3). This result follows from *Isserlis’ theorem* [33], which decomposes the higher order moments into sums of products of covariances  $E[X_a X_b]$ . In short, the GBS device turns a graph into a multivariate normal distribution and samples from its moments.

Using this picture, the first-order moments of the ‘graph-induced distribution’ correspond to photon events of the form  $[1, 0, \dots]$  and their probability is indeed proportional to the mode means as apparent from Appendix A. The second-order moments correspond to photon events of the form  $[1, 1, 0, \dots]$  and their probability is proportional to the entries of the adjacency matrix – the edge weights. Consistent with this observation, we stated before that orbits with 2 non-zero detectors “measure” the edge count of a graph.

While the doubled encoding strategy as well as the presence of multi-photon events somewhat obscure interpretations of features in terms of  $r$ -matchings and higher-order moments, we motivate in Appendix C that they can be a blessing in disguise, making very similar graphs distinguishable by smaller maximum photon numbers  $k$ .

## IV. EXPERIMENTS

Finally, we provide some numerical results to investigate the GBS graph kernel in practice. Benchmarks suggest that it is well competitive to standard “classical” graph kernels. We furthermore show that displacement may improve classification accuracy by shifting weight into the higher-order orbits, and that orbits with photon numbers smaller or equal to 2 contribute most to the result.

### A. Benchmarking

To benchmark the GBS feature map, we use a setup that has become a standard in testing graph kernels: A C-Support Vector Machine (SVM) with a precomputed kernel. The test accuracies in Table II are obtained by running 10 repeats of a double 10-fold cross-validation. The inner fold extracts the best model by adjusting the  $C$ -parameter of the SVM – which controls the penalty on misclassifications – via grid search between values  $[10^{-4}, 10^3]$ , and the best model is then used to get the accuracy of the test set in the outer cross-validation loop. The GBS feature vectors were used in conjunction with a ‘rbf’ kernel  $\kappa_{\text{rbf}}$ .

For the GBS graph kernel, we chose a gentle displacement of  $d = 0.25$  on every mode and  $k = 6$ , leading to 30-dimensional feature vectors. We used exact simulations based on the hafnian library [34]. These are computationally very expensive, which is why we only consider small datasets. Three classical graph kernels are benchmarked for comparison: The Graphlet Sampling kernel [9] (GS) with maximum graphlet size of  $k = 5$  and 5174 samples drawn, the Random Walk kernel [35] (RW) with fast computation and a geometric kernel type, and the Subgraph Matching kernel (SM) [23]. The three classical kernels were simulated using Python’s *grakel* library [36].<sup>6</sup>

The datasets are taken from the repository of the Technical University of Dortmund [37] (see Figure 3). Data preprocessing for all experiments consists of a) excluding graphs which have less than 6 or more than 25 nodes, b) extracting binary adjacency matrices and c) rescaling by a constant  $c$  as explained in Section II A. More details regarding preprocessing can be found in Appendix D.

As Table II shows, the GBS kernel performs well and outperforms the other methods visibly for MUTAG and NCI1, while still leading for AIDS, BZR\_MD, ER\_MD, FINGERPRINT and PROTEINS. Displacement increases the performance of the GBS kernel significantly for COX2\_MD, ENZYMES and IMDB-BIN, but not for other data sets. The GBS kernel does well on datasets where

<sup>6</sup> Experiments were run on IBM’s cloud platform using four 2.8GHz Intel Xeon-IvyBridge Ex (E7-4890-V2-PentadecaCore) processors with 15 CPU cores each, as well as on Oak Ridge’s Titan super-computer.

Dataset	GBS ( $d = 0.0$ )	GBS ( $d = 0.25$ )	GS	RW	SM
AIDS	99.60 $\pm$ 0.05	<b>99.62</b> $\pm$ 0.03	98.44 $\pm$ 0.09	56.95 $\pm$ 7.99	79.20 $\pm$ 0.68
BZR_MD	<b>62.73</b> $\pm$ 0.71	62.13 $\pm$ 1.44	60.60 $\pm$ 1.77	49.88 $\pm$ 3.74	61.90 $\pm$ 1.21
COX2_MD	44.98 $\pm$ 1.80	50.11 $\pm$ 0.97	55.04 $\pm$ 3.33	57.72 $\pm$ 3.26	<b>66.94</b> $\pm$ 1.22
ENZYMES	22.29 $\pm$ 1.60	28.01 $\pm$ 1.83	35.87 $\pm$ 2.19	21.13 $\pm$ 1.91	<b>36.70</b> $\pm$ 2.83
ER_MD	70.36 $\pm$ 0.78	<b>70.41</b> $\pm$ 0.47	65.65 $\pm$ 1.06	68.75 $\pm$ 0.53	68.21 $\pm$ 0.99
FINGERPRINT	65.42 $\pm$ 0.49	<b>65.85</b> $\pm$ 0.36	64.10 $\pm$ 1.52	47.69 $\pm$ 0.21	47.14 $\pm$ 0.62
IMDB-BIN	64.09 $\pm$ 0.34	68.71 $\pm$ 0.59	68.37 $\pm$ 0.62	66.38 $\pm$ 0.21	out of time*
MUTAG	<b>86.41</b> $\pm$ 0.33	85.58 $\pm$ 0.59	81.08 $\pm$ 0.93	83.02 $\pm$ 1.08	83.14 $\pm$ 0.24
NCI1	<b>63.61</b> $\pm$ 0.00	62.79 $\pm$ 0.00	49.96 $\pm$ 3.27	52.36 $\pm$ 2.63	51.36 $\pm$ 1.88
PROTEINS	<b>66.88</b> $\pm$ 0.22	66.14 $\pm$ 0.48	65.91 $\pm$ 1.29	56.27 $\pm$ 1.23	63.03 $\pm$ 0.84
PTC_FM	53.84 $\pm$ 0.96	52.45 $\pm$ 1.78	<b>59.48</b> $\pm$ 1.95	51.97 $\pm$ 2.68	54.92 $\pm$ 2.94

TABLE II. Mean test accuracy of the Support Vector Machine with different datasets and different graph kernels, with the standard deviation between 10 repetitions of the double cross-validation. GS, RW, and SM are three standard classical graph kernels described in the text. \*Runtime > 20 days.

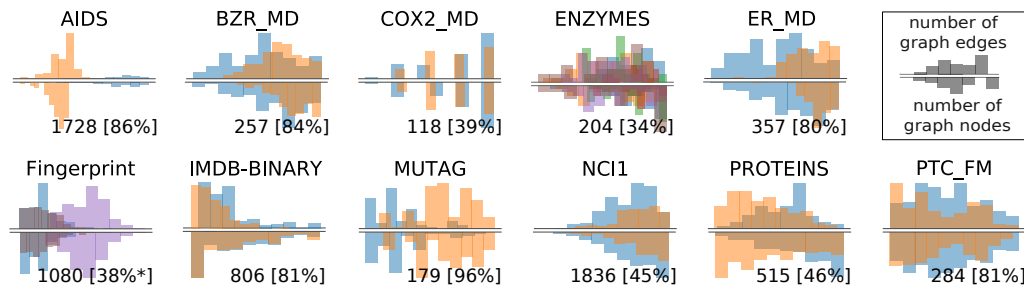


FIG. 3. Histograms of node and edge numbers of graphs in the benchmark datasets. The number of graphs as well as its percentage with respect to the original data are shown below each plot.

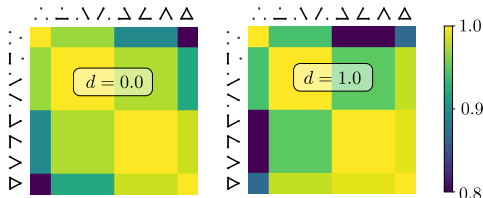


FIG. 4. GBS kernel for all size-3 graphs and displacements  $d = 0, 1$ .

the distribution of node and edge numbers differs strongly between classes. However, we confirmed that excluding the ‘edge counting features’  $[1, 1, 0..], [2, 2, 0..], \dots$  does not influence classification performance. While the graph size is considered by the GBS kernel, it seems to be only one of many properties that enters the notion of similarity.

### B. Displacement and feature importance

The hyperparameters of the GBS graph kernel are the constant displacement  $d$  which administered to each node, as well as the normalization constant  $c$  and the maximum photon number  $k$ . Since simulations restrict the value of  $k$  at this stage and we keep  $c$  fixed as a preprocessing

constant, we focus on the effect of displacement. Displacement can change the similarity measure significantly. For example, comparing graphs of size  $|V| = 3$  in Figure 4, one finds that the fully disconnected graph is closer to the fully connected graph than a graph with two edges for  $d = 1$ , but vice versa for  $d = 0$ .

Figure 5 uses the example of IMDB-BIN and MUTAG to investigate the GBS features for  $d = 0, d = 0.25$  and  $d = 1$ . The feature averages show that the general distribution of the feature vector is similar for both classes, but still visually distinguishable.<sup>7</sup> Consistent with the theory, increasing displacement shifts the features towards higher-order orbits, and populates features that are zero when  $d = 0$ . Features associated with orbits  $[1, 1, 0, \dots], [1, 1, 1, 1, 0, \dots]$  and  $[1, 1, 1, 1, 1, 1]$ , as well as  $[2, 1, 0, \dots]$  and  $[2, 1, 1, 1, 0, \dots]$  seem to be particularly important in the support of principal components, and get high weights when training a perceptron on the GBS features. Where displacement renders them nonzero, uneven orbits such as  $[1, 1, 1, 0, \dots], [1, 1, 1, 1, 0, \dots]$  follow suit. During our investigations we confirmed that drop-

<sup>7</sup> Standardization of the feature vectors to emphasize their mutual differences improved classification accuracy in some cases, but deteriorated it in others.



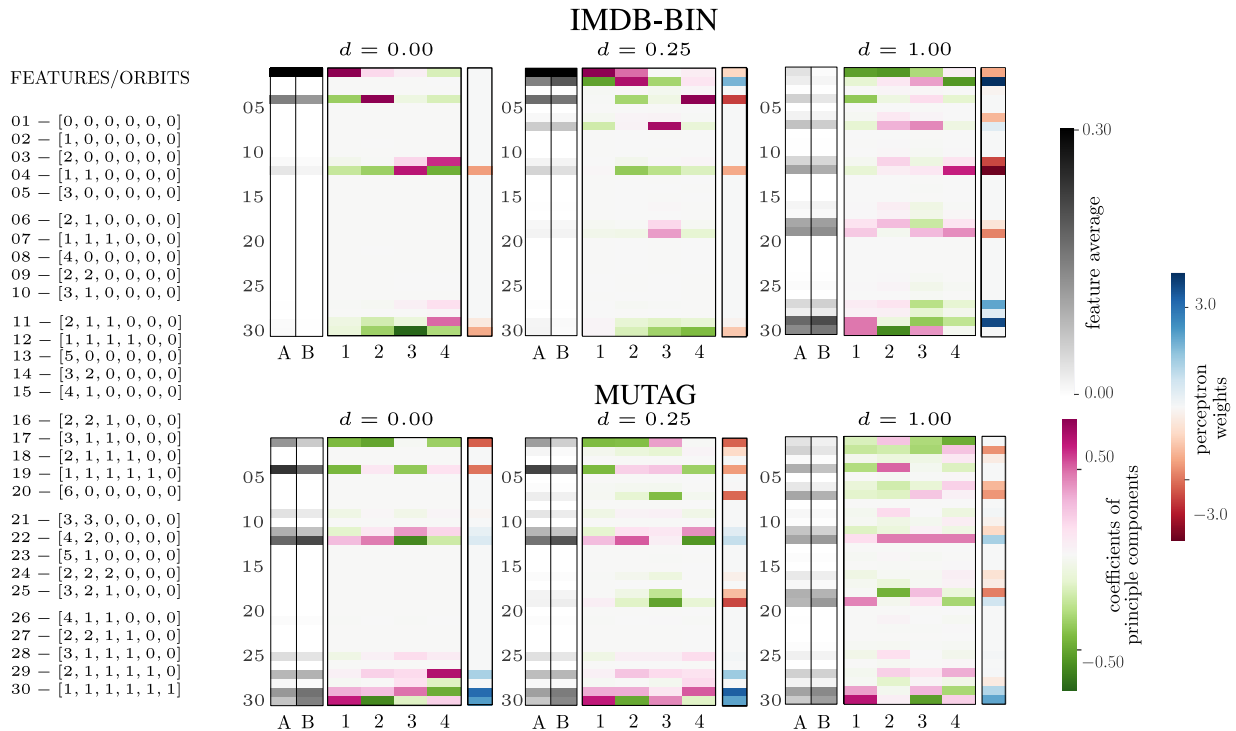


FIG. 5. Three measures for feature importance for IMDB-BINARY (top row) and MUTAG (bottom row) using  $k = 6$  and for  $d = 0, d = 0.25$  and  $d = 1$ . The 3 + 3 heatmaps consist of three columns each. The leftmost column (gray color map) shows the average of each feature for the two different classes, here labeled  $A$  and  $B$ . The center column shows the coefficients with which each feature contributes to the four first principal components in the PCA analysis. The third column shows the weights which a perceptron attributes to each feature when trained to classify the target labels.

ping features with high single-detector photon numbers did not have a huge influence on classification. Consistent with the results from Table II, MUTAG has ‘richer’ features for  $d = 0$  than IMDB-BIN for classification with a perceptron, an advantage that IMDB-BIN equalizes with growing displacement.

The feature analysis suggests that features related to subgraphs of all sizes (here 1 to 6) are important for the classification results, and that duplication of a single node in the subgraphs may be beneficial – a feature that Graphlet Sampling kernels do not explore. The effect of displacement varies with the dataset, and  $d$  should therefore be kept as a hyperparameter for model selection.

## V. CONCLUSION

We proposed a new type of feature extraction strategy for graph-structured data based on the quantum technique of Gaussian Boson Sampling. We suggested that the success of the method is related to the fact that such a system samples from distributions that are related to useful graph properties. For classical machine learning,

this method presents a potentially powerful extension to the gallery of graph kernels, each of which has strengths on certain data sets. For quantum machine learning, this proposes the first application of a “quantum kernel”.

A lot of questions are still open for further investigation, for example regarding the role and interpretation of displacement, how GBS performs with weighted adjacency matrices, how node and edge labels can be considered, as well as whether the feature vectors are useful in combination with other methods such as neural networks. We expect that the rapid current development of numeric GBS samplers as well as quantum hardware will help answering these questions in the near future.

## ACKNOWLEDGEMENTS

We thank Christopher Morris and Nicolas Quesada for valuable advice, as well as the authors of Python’s GraKel library. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

[1] Johannes Kobler, Uwe Schöning, and Jacobo Torán, *The graph isomorphism problem: its structural complexity*

(Springer Science & Business Media, 2012).

- [2] Brendan D McKay *et al.*, *Practical graph isomorphism* (Department of Computer Science, Vanderbilt University Tennessee, USA, 1981).
- [3] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich, “A review of relational machine learning for knowledge graphs,” *Proceedings of the IEEE* **104**, 11–33 (2016).
- [4] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu, “A comprehensive survey on graph neural networks,” arXiv preprint arXiv:1901.00596 (2019).
- [5] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov, “Learning convolutional neural networks for graphs,” in *International conference on machine learning* (2016) pp. 2014–2023.
- [6] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang, “Network representation learning: A survey,” *IEEE transactions on Big Data* (2018).
- [7] Palash Goyal and Emilio Ferrara, “Graph embedding techniques, applications, and performance: A survey,” *Knowledge-Based Systems* **151**, 78–94 (2018).
- [8] Aditya Grover and Jure Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2016) pp. 855–864.
- [9] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt, “Efficient graphlet kernels for large graph comparison,” in *Artificial Intelligence and Statistics* (2009) pp. 488–495.
- [10] Swarnendu Ghosh, Nibaran Das, Teresa Gonçalves, Paulo Quaresma, and Mahantapas Kundu, “The journey of graph kernels through two decades,” *Computer Science Review* **27**, 88–111 (2018).
- [11] Bernhard Scholkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond* (MIT press, 2001).
- [12] Nils Kriege, Marion Neumann, Kristian Kersting, and Petra Mutzel, “Explicit versus implicit graph feature maps: A computational phase transition for walk kernels,” in *Data Mining (ICDM), 2014 IEEE International Conference on* (IEEE, 2014) pp. 881–886.
- [13] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala, “Random projections through multiple optical scattering: Approximating kernels at the speed of light,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2016) pp. 6215–6219.
- [14] Maria Schuld and Nathan Killoran, “Quantum machine learning in feature Hilbert spaces,” *Physical review letters* **122**, 040504 (2019).
- [15] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature* **567**, 209 (2019).
- [16] Craig S Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex, “Gaussian boson sampling,” *Physical review letters* **119**, 170501 (2017).
- [17] AP Lund, A Laing, S Rahimi-Keshari, T Rudolph, Jeremy L O’Brien, and TC Ralph, “Boson sampling from a gaussian state,” *Physical review letters* **113**, 100502 (2014).
- [18] Regina Kruse, Craig S Hamilton, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex, “A detailed study of Gaussian Boson Sampling,” arXiv preprint arXiv:1801.07488 (2018).
- [19] Max Tillmann, Borivoje Dakić, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther, “Experimental boson sampling,” *Nature Photonics* **7**, 540 (2013).
- [20] Matthew A Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C Ralph, and Andrew G White, “Photonic boson sampling in a tunable circuit,” *Science* **339**, 794–798 (2013).
- [21] Scott Aaronson and Alex Arkhipov, “The computational complexity of linear optics,” in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (ACM, 2011) pp. 333–342.
- [22] Kamil Brádler, Shmuel Friedland, Josh Izaac, Nathan Killoran, and Daiqin Su, “Graph isomorphism and Gaussian boson sampling,” arXiv preprint arXiv:1810.10644 (2018).
- [23] Nils Kriege and Petra Mutzel, “Subgraph matching kernels for attributed graphs,” arXiv preprint arXiv:1206.6483 (2012).
- [24] Kamil Brádler, Pierre-Luc Dallaire-Demers, Patrick Rebentrost, Daiqin Su, and Christian Weedbrook, “Gaussian boson sampling for perfect matchings of arbitrary graphs,” *Physical Review A* **98**, 032310 (2018).
- [25] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd, “Gaussian quantum information,” *Reviews of Modern Physics* **84**, 621 (2012).
- [26] Leslie G Valiant, “The complexity of computing the permanent,” *Theoretical computer science* **8**, 189–201 (1979).
- [27] Alexander Barvinok, “Approximating permanents and hafnians,” arXiv preprint arXiv:1601.07518 (2016).
- [28] Mark Rudelson, Alex Samorodnitsky, Ofer Zeitouni, *et al.*, “Hafnians, perfect matchings and gaussian matrices,” *The Annals of Probability* **44**, 2858–2888 (2016).
- [29] VD Vaidya, B Morrison, LG Helt, R Shahrokhshahi, DH Mahler, MJ Collins, K Tan, J Lavoie, A Repington, M Menotti, *et al.*, “Broadband quadrature-squeezed vacuum and nonclassical photon number correlations from a nanophotonic device,” arXiv preprint arXiv:1904.07833 (2019).
- [30] E.J Farrell, “An introduction to matching polynomials,” *Journal of Combinatorial Theory, Series B* **27**, 75–86 (1979).
- [31] Chris D. Godsil and Ivan Gutman, “On the theory of the matching polynomial,” *Journal of Graph Theory* **5**, 137–144 (1981).
- [32] Ole J Heilmann and Elliott H Lieb, “Theory of monomer-dimer systems,” in *Statistical Mechanics* (Springer, 1972) pp. 45–87.
- [33] Leon Isserlis, “On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables,” *Biometrika* **12**, 134–139 (1918).
- [34] Andreas Björklund, Brajesh Gupta, and Nicolás Quesada, “A faster hafnian formula for complex matrices and its benchmarking on the titan supercomputer,” arXiv preprint arXiv:1805.12498 (2018).
- [35] Thomas Gärtner, Peter Flach, and Stefan Wrobel, “On graph kernels: Hardness results and efficient alternatives,” in *Learning theory and kernel machines* (Springer, 2003) pp. 129–143.



- [36] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis, “Grakel: A graph kernel library in python,” arXiv preprint arXiv:1806.02193 (2018).
- [37] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann, “Benchmark data sets for graph kernels,” (2016).
- [38] Nicolás Quesada, “Franck-condon factors by counting perfect matchings of graphs with loops,” The Journal of chemical physics **150**, 164113 (2019).
- [39] E Spence, “Regular graphs with four different eigenvalues,” [Http://www.maths.gla.ac.uk/es/1.24.gz](http://www.maths.gla.ac.uk/es/1.24.gz).

### Appendix A: Adding displacement

The Gaussian Boson Sampling setup underlying Eq. (3) consists of squeezing and interferometers. But a Gaussian quantum state can also be manipulated by a third operation: displacement. Displacement changes the mean of the  $M$ -mode Gaussian state while leaving the covariance matrix (and therefore the encoding strategy) as before. A non-zero mean changes Eq. (3) in an interesting, but non-trivial manner.

Without going into the details [18], if considering nonzero displacement, instead of summing over  $P_N^{\{2\}}$  in Eq. (4), we have to sum over  $P_N^{\{1,2\}}$ , or the set of partitions of the index set  $\{1, \dots, N\}$  into subsets of size *up to* 2. For the index set  $\{1, 2, 3, 4\}$ , we had

$$P_4^{\{2\}} = \{(1, 2), (3, 4)\}, \{(1, 3), (2, 4)\}, \{(1, 4), (2, 3)\},$$

which now becomes

$$\begin{aligned} P_4^{\{1,2\}} = & \{ \{(1, 2), (3), (4)\}, \{(1, 3), (2), (4)\}, \{(1, 4), (2), (3)\}, \\ & \{(2, 3), (1), (4)\}, \{(2, 4), (1), (3)\}, \{(3, 4), (1), (2)\}, \\ & \{(1, 2), (3, 4)\}, \{(1, 3), (2, 4)\}, \{(1, 4), (2, 3)\}, \\ & \{(1), (2), (3), (4)\} \end{aligned}$$

Instead of the Hafnian in Eq. (3), we therefore get a mixture of Hafnians of  $A_{\mathbf{n}}$ 's submatrices (stemming from the pairs) and other factors (stemming from the size-1 sets).

Assume that displacement is applied to both the  $\hat{x}$  and  $\hat{p}$  quadratures of each mode, described by a  $2M$ -dimensional displacement vector  $\mathbf{d} = (d_1, \dots, d_M, d_1^*, \dots, d_M^*)^T$ . The effect on Eq. (3) is as follows. Let  $Q$  the  $2M \times 2M$  matrix from Eq. (2),  $\mathbf{b} = \mathbf{d}^\dagger Q^{-1}$ , and

$$\tilde{A}_{\mathbf{n}} = c \begin{pmatrix} A_{\mathbf{n}} & 0 \\ 0 & A_{\mathbf{n}} \end{pmatrix}$$

similar to Eq. (1). We get

$$\begin{aligned} p(\mathbf{n}) &= \frac{e^{-\frac{1}{2}\mathbf{d}^\dagger Q^{-1}\mathbf{d}}}{\sqrt{\det(Q)} \mathbf{n}!} \left[ \text{Haf}(\tilde{A}_{\mathbf{n}}) + \sum_{i \neq j}^{2M} b_i b_j \text{Haf}(\tilde{A}_{\mathbf{n}-\{i,j\}}) + \dots + \prod_j^{2M} b_j \right], \\ &= \frac{e^{-\frac{1}{2}\mathbf{d}^\dagger Q^{-1}\mathbf{d}}}{\sqrt{\det(Q)} \mathbf{n}!} \sum_{n=0}^M \sum_{\{i_1 \dots i_{2n}\} \subseteq \mathcal{I}_{2M}} b_{i_1} \dots b_{i_{2n}} \text{Haf}(\tilde{A}_{\mathbf{n}-\{i_1, \dots, i_{2n}\}}), \end{aligned} \quad (\text{A1})$$

where  $\mathcal{I}_{2M}$  is the index set  $\{1, \dots, 2M\}$ . In this notation we assume  $\{i_1, \dots, i_0\} = \{\}$  and  $b_{i_1} \dots b_{i_0} = 1$ . The “reduced” Hafnians of the form  $A_{\mathbf{n}-\{i,j\}}, A_{\mathbf{n}-\{i,j,k,l\}} \dots$  are constructed by “deleting” rows and columns  $\{i, j\}, \{i, j, k, l\}, \dots$  in  $\tilde{A}_{\mathbf{n}}$ . The expression in the brackets of Eq. (S1) is also known as a “loop Hafnian” of a matrix  $\tilde{A}_{\mathbf{n}}$  that carries  $b_1, \dots, b_{2M}$  on its diagonal [38].

Equivalently to the square rule  $\text{Haf}(\tilde{A}) = \text{Haf}(A)^2$  in the regime of zero displacement, Eq. (A1) simplifies to

$$\begin{aligned} p(\mathbf{n}) &= \frac{e^{-\frac{1}{2}\mathbf{d}^\dagger Q^{-1}\mathbf{d}}}{\sqrt{\det(Q)} \mathbf{n}!} \left[ \text{Haf}(A_{\mathbf{n}}) + \sum_i^M b_i \text{Haf}(A_{\mathbf{n}-\{i\}}) + \dots + \prod_j^M b_j \right]^2 \\ &= \frac{e^{-\frac{1}{2}\mathbf{d}^\dagger Q^{-1}\mathbf{d}}}{\sqrt{\det(Q)} \mathbf{n}!} \left[ \sum_{n=0}^M \sum_{\{i_1 \dots i_n\} \subseteq \mathcal{I}_M} b_{i_1} \dots b_{i_n} \text{Haf}(A_{\mathbf{n}-\{i_1, \dots, i_n\}}) \right]^2. \end{aligned} \quad (\text{A2})$$

To show that (A1) = (A2), one uses the fact that for  $\tilde{A}$  being a direct sum  $A \oplus A$ , the index set  $i_1, \dots, i_{2n} \in \mathcal{I}_{2M}$  that Eq. (A1) sums over can be divided into two index sets:  $j_1, \dots, j_s$  which contains all  $s$  indices from the ‘first subspace’ (i.e., the first  $M$  dimensions) of  $\tilde{A}$ , and  $k_1, \dots, k_{s'}$  containing the  $s'$  indices from the ‘second subspace’, and  $s + s' = 2n$ . The fact that  $\text{Haf}(A \oplus B) = \text{Haf}(A)\text{Haf}(B)$ , allows us to express the Hafnian of reduced versions of  $\tilde{A}_{\mathbf{n}}$  as a product of reduced versions of matrix  $\tilde{A}_{\mathbf{n}}$ ,

$$\text{Haf}(\tilde{A}_{\mathbf{n}-\{i_1, \dots, i_{2n}\}}) = \text{Haf}(A_{\mathbf{n}-\{j_1, \dots, j_s\}})\text{Haf}(A_{\mathbf{n}-\{k_1, \dots, k_{s'}\}}).$$

Altogether, we can therefore write:

$$\begin{aligned} p(\mathbf{n}) &\propto \sum_{n=0}^M \sum_{\{i_1 \dots i_{2n}\} \subseteq \mathcal{I}_{2M}} b_{i_1} \dots b_{i_{2n}} \text{Haf}(\tilde{A}_{\mathbf{n}-\{i_1 \dots i_{2n}\}}) \\ &= \sum_{s, s'=0}^M \sum_{\{j_1 \dots j_s\} \subseteq \mathcal{I}_M} \left( b_{j_1} \dots b_{j_s} \text{Haf}(A_{\mathbf{n}-\{j_1 \dots j_s\}}) \right) \\ &\quad \times \sum_{\{k_1 \dots k_{s'}\} \subseteq \mathcal{I}_M} \left( b_{k_1} \dots b_{k_{s'}} \text{Haf}(A_{\mathbf{n}-\{k_1 \dots k_{s'}\}}) \right) \\ &= \left( \sum_{n=0}^M \sum_{\{i_1 \dots i_n\} \subseteq \mathcal{I}_M} b_{i_1} \dots b_{i_n} \text{Haf}(A_{\mathbf{n}-\{i_1 \dots i_n\}}) \right)^2 \end{aligned}$$

In the main paper we always consider  $\mathbf{d} = (d, \dots, d, d, \dots, d)^T$  for a constant value  $d$ .

One can see that displacement explores substructures of doubled extended subgraphs (doubled, since we consider  $\tilde{A}_{\mathbf{n}}$  and not  $A_{\mathbf{n}}$ ). An important effect of displacement is that  $p(\mathbf{n})$  for odd total photon numbers  $|\mathbf{n}|$  is not necessarily zero any more, since the sum in Eq. (A2) contains Hafs of even-sized subgraphs.

## Appendix B: Comparison to Graphlet Sampling kernel

Counting subgraphs in a larger graph is a concept used in various classical graph kernels. Graphlet Sampling kernels [9] bear the most striking similarity to GBS feature maps, since the features count how often graphlets of size  $|V| = 3, 4, 5, \dots$  appear in a graph  $G$ . In the language developed here we can express the feature  $f_g$  which counts graphlet  $g$  via

$$f_g \propto \sum_{\mathbf{n} \in \mathcal{O}_{[1, \dots, 1, 0, \dots]}} \mathbb{1}_{g \cong G_{\mathbf{n}}}, \quad (\text{B1})$$

using an indicator function  $\mathbb{1}_{g \cong G_{\mathbf{n}}}$  that is one if graphlet  $g$  is isomorphic to the subgraph  $G_{\mathbf{n}}$  and zero else, as well as the orbit represented by  $[1, \dots, 1, 0, \dots]$  counting  $|V|$  single photons. In comparison, rewriting Eq. (6) in a similar way, the GBS features are

$$f_i = f_{\mathbf{n}_i^*} \propto \sum_{\mathbf{n} \in \mathcal{O}_{\mathbf{n}^*}} \left( \sum_{g \in \mathcal{P}^{|\mathbf{n}|}} \mathbb{1}_{g \cong G_{\mathbf{n}}} \right)^2, \quad (\text{S4})$$

where  $\mathcal{P}^{|\mathbf{n}|}$  is the set of all perfect matchings of size  $|\mathbf{n}|$ .

As a result, instead of counting graphlets, the GBS feature map sums squares of perfect matching counts in graphlets. Figure 6 shows the number of perfect matchings in all graphs up to size  $|V| = 6$ .

## Appendix C: Benefits of multi-photon events and GBS polynomial

We remarked in the paper that both multi-photon events and the strategy of encoding a direct sum of an adjacency matrix into a GBS device somewhat obscures simple interpretations of the GBS feature map, but that this could be a blessing in disguise. Here we want to illustrate this with an example (leaving rigorous analysis for future work).

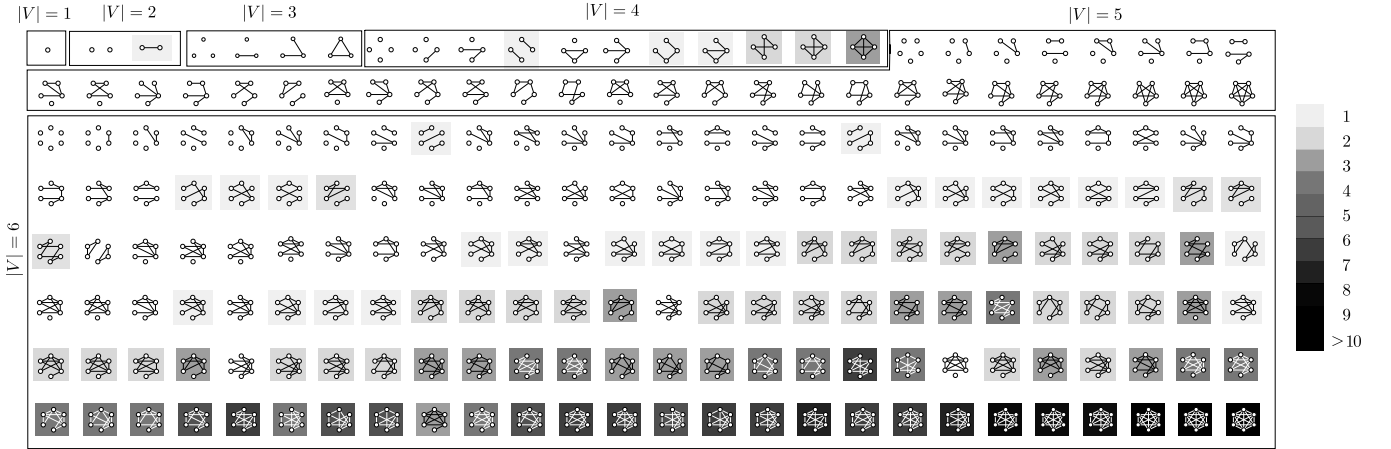


FIG. 6. All non-isomorphic graphs up to size  $|V| = 6$  and the number of perfect matchings they contain (grey shading scale).

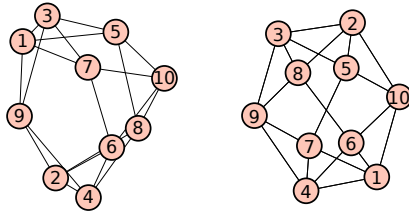


FIG. 7. This pair of regular isospectral non-isomorphic graphs on ten vertices is easier to distinguish with the double encoding strategy than if we only consider their matching polynomials.

Consider the isospectral pair of graphs shown in Figure 7 and  $r \leq 10$ . Their matching polynomials from Eq. (7) up to order  $r = 10$  coincide:

$$\mu(G^{(1)}) = \mu(G^{(2)}) = x^{10} - 20x^8 + 130x^6 - 312x^4 + 229x^2 - 24. \quad (\text{C1})$$

However, extending  $G^{(1)}$  and  $G^{(2)}$  by copying all nodes and their edges once, we find that the matching polynomial of the extended graphs differ:

$$\mu(G_{[2,2,\dots,2]}^{(1)}) - \mu(G_{[2,2,\dots,2]}^{(2)}) = -1536x^4 + 3840x^2 - 768. \quad (\text{C2})$$

In other words, studying the structure of extended graphs can increase distinguishability in the context of graph isomorphism.

As laid out in detail in Section 3.1 of the main paper, GBS does not compute the coefficients of matching polynomials, but a slightly different quantity which we termed the ‘GBS polynomial’  $\gamma(G)$ . It turns out that the GBS polynomials  $\gamma(G_{[2,2,\dots,2]}^{(1,2)})$  differ in orders of  $4 \leq r \leq 10$  (meaning that the difference between the two graphs can already be detected by a GBS device with maximum 8 photons), while the matching polynomials  $\mu(G_{[2,2,\dots,2]}^{(1,2)})$  from Eq (C2) differ only for orders  $8 \leq r \leq 10$ .

Our numeric examinations of families of regular and strongly regular graphs suggest that the situation in (3), where the whole matching polynomials coincide, is not typical. For example, taking the family of 87 isospectral regular graphs on 24 vertices from [39] with the spectrum  $\{[9]^1, [3]^4, [1]^9, [-3]^{10}\}$ , none of their matching polynomials up to  $r = 10$  is identical. Again, by investigating the GBS polynomial we find that they become different ‘sooner’, that is for a lower total photon number, or a lower order  $r$ . The same conclusion holds the family of four strongly regular graphs on 28 vertices.

We conclude that the GBS polynomial may have interesting properties to characterize graphs, which reflect onto but extend applications of graph kernels.

#### Appendix D: Data and data preprocessing

Preprocessing of the benchmarking datasets includes these three steps:

1. *Graph selection*: Graphs which have less than 6 or more than 25 nodes are excluded to keep the feature vectors constant and to limit the time of simulations. The share of excluded graphs is displayed in Figure (3) in the main paper, and ranges from 5% to 55%.
2. *Labels and attributes*: Potential node labels, node attributes and edge attributes are ignored. The edge labels in BZR\_MD, COX2\_MD, ER\_MD, MUTAG and PTC\_FM were translated to the following weights: 0 - no chemical bond, 1 - single bond/double bond/triple bond/aromatic bond. The edge labels in AIDS were translated into the weights: 0 - no edge 1 - valence of zero, one or two. In FINGERPRINT, only graphs of the three dominant classes 0, 4, 5 were considered, since the other classes did not contain a sufficient number of samples after graph selection.
3. *Rescaling*: The final (weighed or unweighed) adjacency matrix is divided by a normalization constant  $c = 1/(\lambda_{\max}^{\{G\}} + 10^{-8})$  that is slightly larger than the largest singular value  $s_{\max}^{\{G\}}$  of any adjacency matrix in the dataset, as explained in Section 2.1 of the main paper.

All datasets were chosen *before* the first experiments were run, to avoid a post-selection bias in favour of the GBS kernel.