Quantum-Inspired Classical Algorithm for Graph Problems by Gaussian Boson Sampling

Changhun Oh,[1,2,*] Bill Fefferman,[3] Liang Jiang,[1] and Nicolás Quesada[4]

[1] *Pritzker School of Molecular Engineering, University of Chicago, Chicago, Illinois 60637, USA*
[2] *Department of Physics, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea*
[3] *Department of Computer Science, University of Chicago, Chicago, Illinois 60637, USA*
[4] *Department of Engineering Physics, École Polytechnique de Montréal, Montréal, Quebec H3T 1J4, Canada*

(Received 23 April 2023; revised 12 November 2023; accepted 11 April 2024; published 23 May 2024)

We present a quantum-inspired classical algorithm that can be used for graph-theoretical problems, such as finding the densest k subgraph and finding the maximum weight clique, which are proposed as applications of a Gaussian boson sampler. The main observation from Gaussian boson samplers is that a given graph's adjacency matrix to be encoded in a Gaussian boson sampler is non-negative and that computing the output probability of Gaussian boson sampling restricted to a non-negative adjacency matrix is thought to be strictly easier than general cases. We first provide how to program a given graph problem into our efficient classical algorithm. We then numerically compare the performance of ideal and lossy Gaussian boson samplers, our quantum-inspired classical sampler, and the uniform sampler for finding the densest k subgraph and finding the maximum weight clique and show that the advantage from Gaussian boson samplers is not significant in general. We finally discuss the potential advantage of a Gaussian boson sampler over the proposed quantum-inspired classical sampler.

DOI: 10.1103/PRXQuantum.5.020341

I. INTRODUCTION

Over the last few years, we have seen the first plausible quantum computational advantages from random circuit sampling with superconducting qubits [1–6] and Gaussian boson sampling [7–12]. While there are numerous interesting debates on the claimed quantum advantage, such as the effect of noise and the verification [13–19], the computational cost of classically simulating the existing quantum devices is still enormous. Therefore, an obvious next step beyond proof-of-principle experiments is to take advantage of the computational power of such noisy intermediate-scale quantum (NISQ) devices to solve more practical problems.

An interesting and crucial observation to exploit the potential quantum advantage of Gaussian boson sampling is that one can embed a graph in the circuit so that a Gaussian boson sampler can be programmed for various graph-theoretical problems [20–23]. Such an observation has led many theoretical proposals to solve problems [23], such as finding the densest k subgraph [21] and finding the maximum weight clique [22]. These problems have attracted much attention because of their potential applications in a wide range of fields such as data mining [24–28] and bioinformatics [29–31] (see Ref. [23] for more applications). Due to the importance of solving the problems, such theoretical proposals have recently started to be experimentally implemented [22,32–34], which opens the possibility of taking advantage of quantum computational advantage from NISQ devices to solve practical problems. In particular, the quantum advantage was claimed by comparing it with the uniform distribution, which was referred to as the classical algorithm.

Meanwhile, potential applications of quantum devices sometimes turn out to be classically simulable. For example, the molecular vibronic spectra problem has been considered as an application of Gaussian boson sampling [35–37]. Very recently, classical algorithms have been developed to solve the problem as accurately as the Gaussian boson sampler for many cases, including the Fock-state version of the problem [38]. Therefore, to claim and exploit the potential quantum advantage more rigorously, it is also crucial to scrutinize the problems' complexity and potential ways of simulating the quantum algorithm using a classical counterpart. Similarly, while the graph-theoretical problems we consider, such as finding the densest k subgraph and finding the maximum weight

[*] Corresponding author: changhun0218@gmail.com

clique, have been considered applications of Gaussian boson sampling, it has been a longstanding open question whether they provide a provable quantum computational advantage [39].

In this work, we present a quantum-inspired classical sampler that can be used for graph-theoretical problems, such as finding the densest k subgraph and finding the maximum weight clique. The algorithm is inspired by the proposals of using Gaussian boson samplers for the problems [21–23,40]. The main idea behind exploiting a Gaussian boson sampler to solve such problems is that the density of a graph is typically proportional to the number of perfect matchings and that the output probability of a Gaussian boson sampler is proportional to the latter. Therefore, a Gaussian boson sampler enables us to more frequently sample subgraphs with many perfect matchings, implying a high density. Our proposed classical algorithm has the same property as a Gaussian boson sampler in that the sampler more frequently samples subgraphs with many perfect matchings. Hence, the crucial property enjoyed by the Gaussian boson sampler to solve such problems can also be similarly attained by our classical algorithm. Thus, our proposal questions the quantum advantage of the Gaussian boson sampler for solving graph problems.

The key observation that makes our classical algorithm efficient is that the adjacency matrix of a graph to be embedded in Gaussian boson-sampling circuits is always non-negative. In addition, it is often the case that computational problems associated only with non-negative quantities are easier than more general cases. For example, computing the output probability of Gaussian boson sampling restricted to a non-negative adjacency matrix in multiplicative error is in $\mathrm{BPP}^{\mathrm{NP}}$, which is thought to be strictly easier than general cases, the complexity of which is #P-hard [41,42]. Using this observation, we present a method that embeds the adjacency matrix of a given graph into two-photon boson-sampling circuits that can be classically efficiently simulated. We emphasize that a Gaussian boson sampler and our classical algorithm depend differently on the number of perfect matchings; the former is proportional to its square while the latter is to itself. To see the effect of such a difference, we then compare the performance of the Gaussian boson sampler, the quantum-inspired classical sampler, and the uniform sampler. We consider Erdős-Rényi random graphs with different parameters and show that the average density and maximum density of subgraphs obtained by a Gaussian boson sampler and our classical sampler do not exhibit a significant difference. We also compare the performance of a heuristic classical algorithm equipped with each sampler for finding the maximum weighted clique for the graphs used in Refs. [22,32] and show again that the difference between our classical sampler and the Gaussian boson sampler is not significant. We finally discuss whether or not a Gaussian boson sampler has a potential exponential advantage over our quantum-inspired classical algorithm.

The paper is organized as follows: In Sec. II, we review the relation between Gaussian boson sampling and graph-theoretic problems. In Sec. III, we provide our quantum-inspired classical algorithm and how to program graph-theoretic problems to the algorithm. In Sec. IV, we numerically analyze the algorithm's performance by comparing it with the Gaussian boson sampler and the uniform sampler. In Sec. V, we discuss the potential advantage of the Gaussian boson sampler over our classical algorithm.

II. GAUSSIAN BOSON SAMPLING AND ITS APPLICATION TO GRAPH-THEORETIC PROBLEMS

Gaussian boson sampling is a sampling task, which is proven to be hard to classically simulate under plausible assumptions [11,12,43]. It can be experimentally implemented by injecting squeezed states with squeezing parameters $\{r_i\}_{i=1}^M$ into M linear-optical circuit, characterized by an $M \times M$ unitary matrix, and measuring the photon-number distribution over the output modes. The output probability of obtaining the photon-number outcome $\mathbf{n} \in \mathbb{Z}_{\geq 0}^M$ can be expressed by the hafnian of a relevant matrix A as [11]

$$p(\mathbf{n}) = \frac{|\mathrm{haf} A_{\mathbf{n}}|^2}{\mathbf{n}!\sqrt{|\Sigma + \mathbb{1}/2|}}, \tag{1}$$

where Σ is the covariance matrix of the output quantum state [44,45] and $A = UDU^{\mathrm{T}}$ with a diagonal matrix $D = \mathrm{diag}(\{\tanh r_i\}_{i=1}^M)$, and $A_{\mathbf{n}}$ is obtained by selecting rows and columns of matrix A corresponding to the outcome $\mathbf{n}$. Here, the hafnian of a $2n \times 2n$ matrix X is defined as

$$\mathrm{haf}(X) = \frac{1}{2^n n!} \sum_{\sigma \in \mathcal{S}_{2n}} \prod_{i=1}^{n} X_{\sigma(2i-1),\sigma(2i)}, \tag{2}$$

where $\mathcal{S}_{2n}$ is the $2n$-element permutation group. The key idea of using Gaussian boson sampling for various applications is that one can embed an arbitrary complex symmetric matrix A by using the Takagi decomposition of $A = UDU^{\mathrm{T}}$ with an appropriate rescaling to make the diagonal component less than one, which is to assure that $\tanh r_i$'s are within their range [20]. We note that there is freedom of choosing the rescaling factor as long as $\tanh r_i$'s are smaller than one, which changes the total photon-number distributions but not the relative weight in the same total photon-number outcome sector. Hence, for a given adjacency matrix A, one can construct a Gaussian boson-sampling circuit with squeezing parameters obtained by the diagonal matrix D and a linear-optical circuit with the unitary matrix U.

Most of the proposed applications are related to graph-theoretic problems due to the hafnian and its relation to the output probability of Gaussian boson sampling [23,46]. Consider an undirected graph $G = (V, E)$, where V is the set of vertices and $E \subset V \times V$ is the set of edges. The graph can be represented by its $|V|$-dimensional adjacency matrix A whose matrix element $A_{ij} = 1$ when $(i,j) \in E$, i.e., when there is an edge between the ith and jth vertices. The key observation is that the hafnian of a matrix A is the number of perfect matchings of a graph whose adjacency matrix is A. Therefore, by embedding the adjacency matrix A into Gaussian boson sampling, one can obtain a sampler that favorably samples outcomes corresponding to a subgraph whose number of perfect matching is large. Such a property has been used to apply a Gaussian boson sampling for finding dense subgraphs based on the observation that the density of a graph is likely to be large when its number of perfect matchings is large [21]. Hence, it can lead to an acceleration of many heuristic classical algorithms, which employ sampling a subgraph as their subroutines, such as finding the densest k-subgraph problem [21] or finding the maximum clique [22].

In the following section, we will show that a similar acceleration is possible using a classical algorithm. The critical observation is that the adjacency matrix A does not contain negative elements and that many computational problems restricted to non-negative elements are often easier than more general cases. For example, computing the permanent of non-negative-element matrices in a multiplicative error is easy for a classical computer [47] while the corresponding problem for general matrices is hard (#P-hard) [47]. Our intuition from a physical perspective is that Gaussian boson sampling associated with non-negative-element matrix A might not necessarily require a complicated multiphoton interference. Indeed, this is true for Fock-state boson sampling; if we use distinguishable particles as an input of Fock-state boson sampling, which is obviously easy to simulate instead of indistinguishable bosons, the output probability is expressed by the permanent of non-negative-element matrices [42].

III. CLASSICAL SAMPLING ALGORITHM FOR GRAPH-THEORETIC PROBLEMS

A. Multiple two-photon boson sampling

Before we provide our classical algorithm, let us consider a Fock-state boson sampling with input $|20\cdots 0\rangle$, i.e., two-photon boson sampling. After a linear-optical circuit U, the probability of obtaining photons at the ith and jth modes with $i \neq j$ is given by

$$p_{ij} = \frac{|\mathrm{Per} U_{1,(i,j)}|^2}{2} = 2|U_{1i}|^2|U_{1j}|^2, \tag{3}$$

where $U_{1,(i,j)}$ is the submatrix of U selecting the first rows twice and the ith and jth columns. When $i = j$, $p_{ii} = |U_{1i}|^4$. More simply, the probability p_{ij} can be thought of as that of obtaining two photons at i and j's modes after two independent trials with probability $p_i = |U_{1i}|^2$. Hence, two-photon boson sampling can be easily simulated by rolling a dice that produces output i with probability $\{|U_{1i}|^2\}_{i=1}^M$ twice independently and then ignoring the order of two outcomes.

Now, consider K different two-photon boson sampling circuits with different linear-optical circuits $\{U^{(k)}\}_{k=1}^K$. We will denote $p_{ij}^{(k)}$ as the kth circuit's probability of obtaining photons at the ith and jth modes. Let us consider a sampling such that for N trials, we randomly choose one circuit out of the set of circuits with probability q_k with $\sum_{k=1}^K q_k = 1$ and inject two photons in the first mode. Finally, we always obtain a $2N$ number of photons after finishing N trials. Let us compute the probability of obtaining outcome $\mathbf{n} = (1, \ldots, 1, 0, \ldots, 0)$, i.e., $2N$ photons are detected on the first $2N$ modes. One can intuitively see that this is related to the perfect matchings, i.e., to hafnian, because we need to find matchings of photon pairs that originate from the same circuit. More precisely and formally, the probability is the sum of possible configurations that provide the corresponding output (see Appendix A for more details):

$$p(\mathbf{n}) = \frac{1}{2^N} \sum_{\sigma \in \mathcal{S}_{2N}} \prod_{i=1}^N \sum_{k=1}^K q_k p^{(k)}_{\sigma(2i-1),\sigma(2i)} \tag{4}$$

$$= \sum_{\sigma \in \mathcal{S}_{2N}} \prod_{i=1}^N \sum_{k=1}^K q_k |U^{(k)}_{1,\sigma(2i-1)}|^2 |U^{(k)}_{1,\sigma(2i)}|^2 \tag{5}$$

$$= 2^N N! \mathrm{haf}(A_{\mathbf{n}}), \tag{6}$$

where A is a non-negative $M \times M$ matrix with its elements defined as

$$A_{ij} \equiv \sum_{k=1}^K q_k |U^{(k)}_{1,i}|^2 |U^{(k)}_{1,j}|^2 = VQV^{\mathrm{T}} = WW^{\mathrm{T}}, \tag{7}$$

and

$$V_{ik} \equiv |U^{(k)}_{1,i}|^2, \quad W_{ik} \equiv \sqrt{q_k}|U^{(k)}_{1,i}|^2, \quad Q \equiv \mathrm{diag}(q_1, \ldots, q_K). \tag{8}$$

Also, $A_{\mathbf{n}}$ is a $2N \times 2N$ submatrix of A with selecting the part of ones of $\mathbf{n}$. The expression of the probability suggests that if we implement the routine as described above, the corresponding sampler's output probability is proportional to the hafnian of the submatrix of A. Thus, such a construction allows us to find a classical sampler whose output probability is written as a submatrix of the hafnian of a matrix A, which can be obtained by U and Q. The remaining challenge is to prove that we can embed an arbitrary non-negative matrix A into a sampler whose output probability is proportional to the hafnian of submatrices of A, which will be addressed in the following section.

It is worth emphasizing that by focusing on the outcome vector $\mathbf{n} = (1, \ldots, 1, 0, \ldots, 0)$ instead of the individual outcomes of each trial, we assumed that we forget the information of the unitary operation, i.e., the index k, for each trial. In principle, we have the index information when we run such a procedure, but ignoring the index information (much like marginalization of the corresponding information) renders the desired property that the output probability is expressed by a hafnian (see Appendix A for more details).

B. Mapping a general graph to a circuit

Consider an $M \times M$ general non-negative symmetric matrix A, which corresponds to a graph of M vertices. Although we will call this matrix an adjacency matrix for simplicity, we require only the matrix to be non-negative and symmetric. We first note that the hafnian of a matrix does not depend on its diagonal elements. Thus, we can freely add an arbitrary diagonal matrix without changing its hafnian so that the matrix becomes diagonally dominant, i.e., $A_{ii} \geq \sum_{j\neq i} |A_{ij}|$ for all i's. Hence, without loss of generality, we will assume that a given adjacency matrix A satisfies

$$A_{ii} = \sum_{j\neq i} |A_{ij}| \quad \text{for all } i'\text{s}. \tag{9}$$

It is known that a non-negative symmetric diagonally dominant matrix is completely positive [48], i.e., $A = HH^{\mathrm{T}}$, where H is not necessarily a square matrix. Thus, once we find the matrix H, we can construct W in Eq. (7). Now we show how to find H such that $A = HH^{\mathrm{T}}$, which is the result from Ref. [48].

For a given $M \times M$ adjacency matrix, after adding the diagonal matrix to satisfy Eq. (9), we can always decompose the matrix A as

$$A = \sum_{1\leq j<i\leq M} B^{(i,j)}, \tag{10}$$

where the $M \times M$ matrices $B^{(i,j)}$ have A_{ij} in positions ii, ij, ji and jj and 0 elsewhere. Then, we can rewrite it as $B^{(i,j)} = \mathbf{b}^{(i,j)}(\mathbf{b}^{(i,j)})^{\mathrm{T}}$ with the M-dimensional vector $\mathbf{b}^{(i,j)}$ whose ith and jth elements are $\sqrt{A_{ij}}$. We then construct $M \times M^2$ matrix H such that H's $(M(i-1)+j)$th column is $\mathbf{b}^{(i,j)}$. Here, we set $(M(i-1)+i)$th columns to be zero. Then, we can easily check that $A = HH^{\mathrm{T}}$.

So far, we have shown how to find H, such that $A = HH^{\mathrm{T}}$. Now we will show how to construct two-photon boson sampling circuits $\{U^{(k)}\}_{k=1}^{M^2}$ to program H. We first define an $M^2 \times M^2$ diagonal matrix D such that D's each element is the sum of H's ith column i.e., $D_{ii} = \sum_{k=1}^{M} H_{ki}$ for all $i \in [M^2]$. We then divide H's ith column by it, so that the resultant $M \times M^2$ matrix V satisfies $\sum_{k=1}^{M} V_{ki} = 1$ for all i's and $HH^{\mathrm{T}} = VD^2V^{\mathrm{T}}$. Finally, we rescale the matrix D^2 so that its trace becomes 1, i.e., $Q \equiv D^2/\mathrm{Tr}[D^2]$. Finally, we obtain the same form as Eq. (7),

$$A = \mathrm{Tr}[D^2]VQV^{\mathrm{T}}, \tag{11}$$

with a coefficient $\mathrm{Tr}[D^2]$. Here, the meaning of the diagonal matrix Q is manifest that Q's kth diagonal element is q_k, i.e., the probability of selecting the kth circuit. Also, V's kth column represents $|U_{1,i}^{(k)}|^2$. Thus, accordingly, for each $k \in [M^2]$ we construct $U_{1,i}^{(k)}$, which can be easily done because we require only a condition for a single row of the unitary matrix and V is a non-negative matrix. (This is

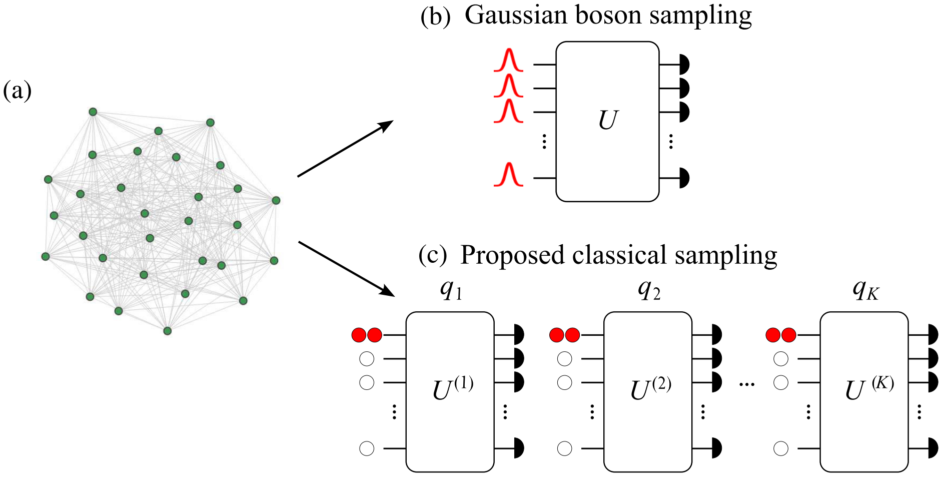

FIG. 1. (a) Graph related to a given graph-theoretical problem such as finding the densest k subgraph or the maximum weighted clique. (b) One can embed such a graph into a Gaussian boson sampler [20] (c) We propose a classical sampler, which can solve the problem in a similar manner to a Gaussian boson sampler.

to show the correspondence; in practice, one can directly use $|U_{1,i}^{(k)}|^2$ as sampling probabilities without constructing two-photon boson sampling.)

To summarize, we have shown that for a given $M \times M$ adjacency matrix A, we can construct a set of linear-optical circuits $\{U^{(k)}\}_{k=1}^{M^2}$ with associated probabilities $\{q_k\}_{k=1}^{M^2}$, which samples outcomes with probabilities proportional to the hafnian of the corresponding $2N \times 2N$ submatrix of A, i.e., of subgraphs of $2N$ vertices. Here, we can freely choose N depending on the size of subgraphs we want. Since two-photon boson sampling can be easily simulated by a classical computer, the entire procedure can be simulated using a classical computer efficiently.

One interesting difference between our classical algorithm and a Gaussian boson sampler is that the latter's output probability is proportional to the square of the hafnian, $\propto |\text{haf}(A_{\mathbf{n}})|^2$ instead of the hafnian itself $\propto \text{haf}(A_{\mathbf{n}})$, which may provide better performance (see Sec. V for more discussion). In the following section, we compare their performances for finding the densest k-subgraph problem and finding the maximum clique problem. One advantage of our quantum-inspired classical algorithm is that we can choose the output photon number as we want, while Gaussian boson sampling has a fixed total photon-number distribution for a given setup. Thus, when one wants to focus on a particular size of subgraphs, Gaussian boson samplers cost additional overhead for postselection.

We emphasize that since we require the matrix A to be non-negative, our algorithm cannot be used for the standard Gaussian boson sampling where the matrix A is generally complex. Also, for the relevant problems we consider, we need only to consider collision-free outcomes, since only collision-free outcomes correspond to proper subgraphs. Due to this and since the constructed $U_{1,i}^{(k)}$ is only nonzero for at most two i's for each k [see Eq. (10) and the construction of B, H, and V], we need only to simply generate two photons for such i's modes instead of sampling after we sample a circuit from $\{q_k\}_{k=1}^{M^2}$.

Finally, our sampling algorithm can also be used to estimate the hafnian of a non-negative matrix with an additive error, which is comparable with the algorithms in Refs. [49–51]. It is worth emphasizing that conversely, the algorithms that estimate the hafnian of a non-negative matrix with an additive error do not immediately derive an *exact* sampling algorithm proposed in this work because of the additive error. For instance, Ref. [51] employed the approximate estimation of the hafnian of a non-negative matrix to construct a sampling algorithm. However, the approximation algorithm entails a systematic error due to the additive error induced by the small number of samples, which may become exponentially significant as the system size grows; in stark contrast, our algorithm is exact.

IV. PERFORMANCE COMPARISON

Many applications of the Gaussian boson-sampling routine are based on generating subgraphs that have high density. Typically, the performance of Gaussian boson sampling has been compared to the uniform sampler, which was treated as a representative classical sampler [21, 22,32,33]. Our quantum-inspired algorithm clearly provides a better way of exploiting a classical computer than a simple uniform sampler. In this section, we quantitatively compare the performances for two different tasks.

A. Finding the densest *k* subgraph

One of the particularly interesting problems proposed to use Gaussian boson samplers is finding the densest k-subgraph problem, the definition of which is as follows: given a graph G with M vertices, find the subgraph of $k < M$ vertices with the largest density. This problem is known to be NP-hard [52]. Thus, generally, it is not believed to be efficiently solved using quantum devices. Nevertheless, in Ref. [21], it was shown that a Gaussian boson sampler can accelerate the performance of a heuristic classical algorithm using the fact that the density of a graph is proportional to the number of perfect matchings and that the output probability of a Gaussian boson sampler is proportional to the square of the number of perfect matchings. As we have shown in the previous section, our classical sampler also enjoys the same property, namely, its output probability is proportional to the number of perfect matchings. Therefore, the key property of exploiting a Gaussian boson sampler to solve this problem can also be obtained using our classical algorithm.

For the simulation with the Gaussian boson sampler, for a given graph, we program the Gaussian boson-sampling circuit using the method in Sec. II and exactly simulate it using a classical algorithm [53]. Here, we choose the covariance matrix whose average photon number is equal to the target subgraph's size k. As mentioned before, for our quantum-inspired classical algorithm, we can fix the output photon number, corresponding to the size of subgraphs. The uniform sampler is also implemented over the k photon subspace.

For practical consideration, we also consider lossy Gaussian boson sampling, where we replace the lossless squeezed states with squeezing parameter r_i by new squeezing parameter r_i' such that the average photon number of the new squeezed states after loss equals the mean photon number of the lossless squeezed states, i.e., $\sinh^2 r_i = \eta \sinh^2 r_i'$ where η is the transmission rate and $1 - \eta$ is the loss rate. Thus, we compensate for the effect by increasing the input squeezing parameter so that the output Gaussian state has the average photon number to be k. Otherwise, the probability of detecting k photons is highly reduced due to the loss. From the experimental perspective, such an adjustment inevitably increases the

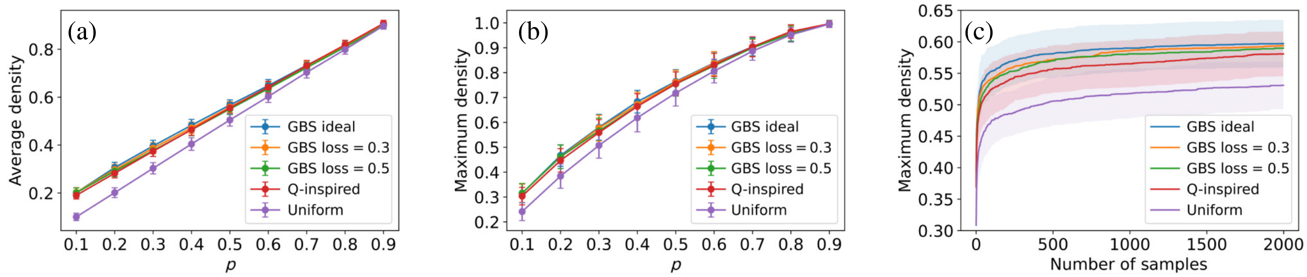

FIG. 2. Density of samples from ideal and lossy Gaussian boson sampler, our classical sampler, and the uniform sampler for random graphs with $n = 30$ vertices and $k = 10$ size subgraphs and various p. (a) Average density for different regimes. (b) Maximum density for different regimes. We use 10^2 different graphs with 10^3 number of samples for each p. (c) The maximum density achieved by a different number of samples with $p = 0.3$. We analyze other choices of $n = 20, 40, 100$ in Appendix B and show that the behaviors are similar to the presented case of $n = 30$.

input squeezing parameter, making the experiment more demanding.

To compare the performance between different strategies, the ideal and lossy Gaussian boson samplers, our quantum-inspired classical sampler, and the uniform sampler, we generate Erdős-Rényi random graphs with different probability p of having edges between vertices and present the behavior of the density of generated samples from each sampler in Fig. 2. The figure shows that the Gaussian boson sampler and our quantum-inspired classical sampler do not have a significant difference for a wide range of p, while the uniform sampler tends to generate lower density graphs than the former two, especially for low p. We also compare the maximum density obtained for a different number of samples. Again, the uniform sampler clearly shows poorer performance than other samplers. Although our quantum-inspired classical algorithm generates a slightly lower maximum density, the difference is not very significant.

To compare lossy Gaussian boson samplers with the lossless Gaussian boson sampler, we observe that the performance degradation due to loss is insignificant. Hence, for the task of generating dense subgraphs, adjusting the input squeezing parameters maintains the performance of the ideal Gaussian boson sampler, while preparing squeezed states with large squeezing becomes more demanding experimentally and may cause additional loss. Recently, such noise robustness has been used to provide another classical way of generating dense graphs in Ref. [54].

In Fig. 3, we compare the probability distributions of the ideal Gaussian sampler (normalized to the postselected sector), the quantum-inspired classical sampler, and the uniform sampler for Erdős-Rényi random graphs with $p = 0.3$ and $p = 0.75$ cases. For the instance that has the largest probability, i.e., the largest perfect matching, the probability ratio between the Gaussian boson sampler and the uniform sampler is large as 280 and 10, respectively, which can be a large advantage. However, when compared to our quantum-inspired classical algorithm, the ratio becomes only about 9 and 3. It implies that we need only approximately 10 times more samples to obtain the instance. Such a difference comes from the different proportionality of the probabilities to hafnian. We discuss this difference and the potential advantage of Gaussian boson samplers over our algorithm in Sec. V. In Refs. [21,40], the additional heuristic classical algorithm is applied to find the densest k subgraph. In the next section for finding the maximum weighted clique, we will compare the performance incorporated with the additional algorithm.

B. Finding the maximum weighted clique

Another relevant interesting problem is finding the maximum weighted clique [55,56]. In Ref. [22], the quantum-classical hybrid algorithm equipped with Gaussian boson samplers has been proposed to accelerate a heuristic classical algorithm to solve molecular docking problems, which is related to drug design [22,57,58]. The problem takes an input of an undirected graph with vertices and its adjacency matrix with an additional vector, which is the weight

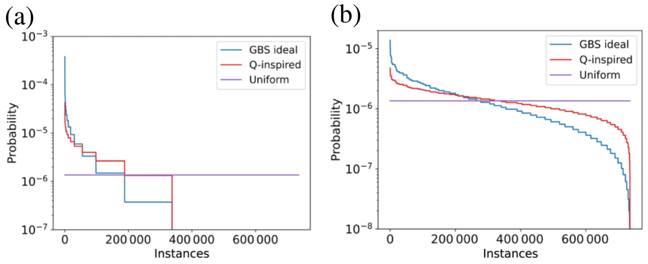

FIG. 3. Examples of probability distributions of samplers for Erdős-Rényi random graphs with (a) $p = 0.3$ and (b) $p = 0.75$ with $n = 24$, $k = 8$. The ratio of the largest probability between the Gaussian boson sampler and the uniform sampler is about 280 and 10, while that between the Gaussian boson sampler and the quantum-inspired classical algorithm is only about 9 and 3, respectively.

vector associated with vertices. Reference [22] provides a way of embedding the problem to a Gaussian boson sampler, which favorably samples the outcome corresponding to a high-weight clique. To embed the weight of vertices in the sampler, we first construct a Gaussian boson sampler corresponding to a matrix B, which is given by

$$B = \Omega(D - A)\Omega, \tag{12}$$

where Ω is a suitable diagonal matrix and D is degree matrix ($D - A$ is graph Laplacian). The degree matrix is the diagonal matrix with the degree of each vertex. Then, it can be shown that the probability distribution is given by

$$p(\mathbf{n}) \propto [\det(\Omega_{\mathbf{n}})\mathrm{haf}(A_{\mathbf{n}})]^2. \tag{13}$$

Therefore, by choosing Ω as a diagonal matrix with the weight vector as its diagonal elements, we can introduce the effect of the weight in the probabilities, so that the sampler favors the outcomes having a large weight. For collision-free cases, which is our main interest for graph problems, the dependence from D disappears and we can set $B = \Omega A \Omega$ without loss of generality.

Using the fact that for $\Omega = \mathrm{diag}(w_1, \ldots, w_n)$ [59],

$$\mathrm{haf}(\Omega A \Omega) = \left(\prod_{i=1}^{n} w_i\right)\mathrm{haf}(A), \tag{14}$$

we can apply the same classical algorithm with the matrix $\Omega A \Omega$ because if A is completely positive, then so is $\Omega A \Omega$. As pointed out in Ref. [22], one might choose Ω as a weight matrix, but one can also choose $\Omega_{ii} = 1 + \alpha w_i$ to give some freedom that might weigh more on the hafnian than weight. For simplicity, we choose $\alpha = 1$ for our numerical results.

We consider two graphs corresponding to different types of protein, which are studied in Ref. [22]: binding interaction between the tumor necrosis factor-α converting enzyme (TACE) and a thiol-containing aryl sulfonamide compound (AS) and a different protein structure (PBD ID: 1ow7), corresponding to Paxillin LD4 motif bound to the focal adhesion targeting (FAT) domain of the focal adhesion kinase.

With the graphs and weights used in Ref. [22], we first obtain 10^5 samples from each sampler and count the number of samples corresponding to the maximum weighted clique. While the Gaussian boson sampler found approximately 20 samples corresponding to the maximum weighted clique for both cases, our quantum-inspired classical sampler and the uniform sampler found only one or two samples for both cases, which implies that Gaussian boson sampling indeed performs better than the other samplers. The difference in the performance from the previous densest subgraph case is that the problems we consider for the maximum weight clique have only a single subgraph solution over many subgraphs. In contrast, the previous density problems may have many subgraphs having the same density. To be clearer, in Fig. 4, we compare the probability distributions of the ideal Gaussian sampler, the quantum-inspired classical sampler, and the uniform sampler. For the instance that has the largest probability, i.e., the largest perfect matching, the probability ratio between the Gaussian boson sampler and the uniform sampler is large as 323 and 385, respectively, which can be a large advantage. However, when compared to our quantum-inspired classical algorithm, the ratio becomes only about 11 and 15. It implies that we need only about approximately 10 times more samples to obtain the instance.

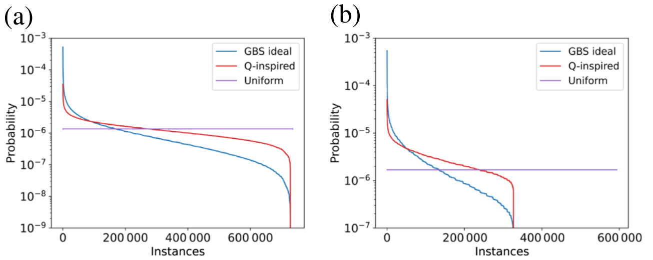

FIG. 4. Probability distributions of samplers for (a) TACE-AS and (b) 1ow7, used in Ref. [22]. The ratio of the largest probability between the Gaussian boson sampler and the uniform sampler is about 323 and 385, while that between the Gaussian boson sampler and the quantum-inspired classical algorithm is only about 11 and 15, respectively.

After sampling from each sampler, we have also implemented additional heuristic procedures, so-called shrinking and searching [60,61]. The basic idea is to shrink the sampled subgraphs to a smaller clique by truncating vertices that do not constitute a clique and to locally search larger cliques by adding vertices that compose cliques maximizing the weight. After the procedures, we compare the success probability of finding the maximum weighted cliques for the graphs used in Ref. [22] and a 32-node graph used in an experiment to claim an advantage over a classical uniform sampler [32], which is shown in Fig. 5. One can see that after the additional step, the difference is not as dramatic as in the first step and that our quantum-inspired classical sampler performs better than the uniform sampler, which has frequently been used as a benchmark.

It is worth emphasizing that the experiment in Ref. [32] using a Gaussian boson sampler and the shrinking and searching postprocessing with 30 iterations obtained a success probability around 16% while we obtain at least 40% of success probability even with the uniform sampler. The difference is that for the searching postprocessing, we aim to find a clique that maximizes the weight, whereas Ref. [32] aims to maximize the size of the clique without

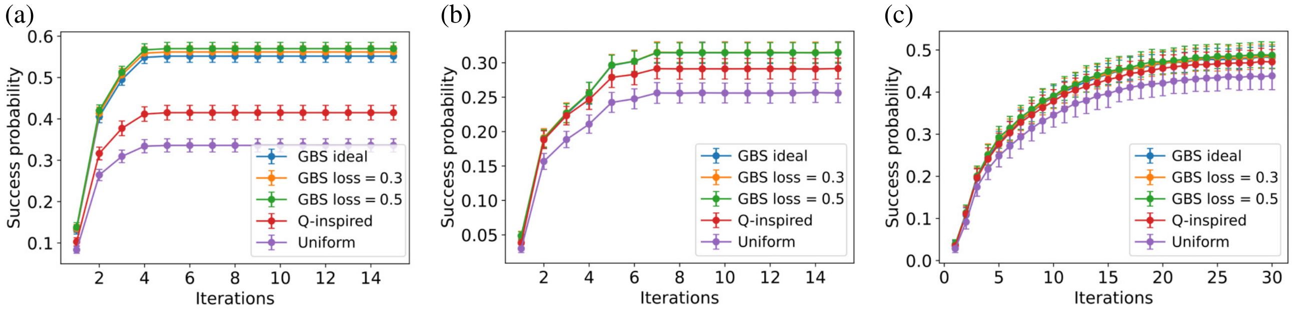

FIG. 5. Success probability of finding the maximum weighted clique with different iteration steps for shrinking and searching with the graphs (a) TACE-AS, (b) 1ow7, used in Ref. [22] and (c) 32-node graph used in a recent experiment [32].

taking into account the weight. Such a different postprocessing leads to a significant gap. Also, for this case, we used only 300 samples, unlike 10^5 samples for the previous cases, to make the number of samples the same as the experiment. Consequently, the very small gap between the performances of the Gaussian boson sampler and our sampler implies that the advantage over a classical sampler does not exist or is insignificant in this case.

Now we compare the maximum squeezing parameters required for ideal and lossy Gaussian boson samplers. For the ideal case, for the first graph, the maximum squeezing parameter is $r_{\max} = 1.380$. On the other hand, for the lossy case, when the loss rate is $1-\eta = 0.3$, $r_{\max} = 1.557$ and when the loss rate is $1-\eta = 0.5$, $r_{\max} = 1.725$. For the second graph, for the ideal case and lossy cases with $1-\eta = 0.3, 0.5$, the maximum squeezing parameters are given by $r_{\max} = 1.121, 1.312, 1.499$, respectively. For the third graph, the maximum squeezing parameters are given by $r_{\max} = 1.200, 1.381, 1.555$, respectively. Therefore, although theoretically, the lossy case is comparable to or even better than the lossless case, the initial squeezing parameters to compensate for the realistic photon loss become more demanding in experiments.

V. DISCUSSION ON POTENTIAL QUANTUM ADVANTAGE FROM THE GAUSSIAN BOSON SAMPLER

We now discuss how much improvement a Gaussian boson sampler might achieve over our classical algorithm and if a Gaussian boson sampler can provide an exponential speedup over our algorithm. We emphasize that finding the densest k subgraph or the maximum weighted clique relies on the heuristic argument for the proportionality between the number of perfect matchings and the density. To avoid the subtlety due to this, in this section, we will focus on the problem that finds the subgraphs with the maximum number of perfect matchings, which is directly related to the output probabilities [40]. To do this, we consider the problem of obtaining a specific sample $\mathbf{n}^*$ having the largest number of perfect matchings, say the solution of the problem, and compare how much time cost is required for each sampler with fixing a certain photon-number sector.

Recall that for a given adjacency matrix A, a Gaussian boson sampler generates samples following the probability distribution (normalized within a particular photon number sector):

$$p_Q(\mathbf{n}) = \frac{|\mathrm{haf}(A_\mathbf{n})|^2}{Z_Q} = \frac{\mathrm{haf}(A_\mathbf{n})^2}{Z_Q}, \tag{15}$$

and the proposed quantum-inspired classical algorithm generates samples following the probability:

$$p_C(\mathbf{n}) = \frac{\mathrm{haf}(A_\mathbf{n})}{Z_C}. \tag{16}$$

Here, Z_Q and Z_C are normalization factors. Then the two distributions are related as

$$p_C(\mathbf{n}) = \frac{\sqrt{p_Q(\mathbf{n})}}{\sum_\mathbf{n}\sqrt{p_Q(\mathbf{n})}}. \tag{17}$$

First, we assume that the solution $\mathbf{n}^*$ is unique, which can be generalized to polynomially many solutions. In this case, when $p_Q(\mathbf{n}^*)$ is inverse-exponentially small, the Gaussian boson sampler already takes exponential time to obtain the corresponding sample. Thus, we will focus on the case that $p_Q(\mathbf{n}^*)$ is only inverse-polynomially small, which guarantees that a Gaussian boson sampler can generate the corresponding sample in a polynomial time. Now, suppose that the probabilities $p_Q(\mathbf{n})$ are concentrated on polynomially many outcomes including $\mathbf{n}^*$, i.e., $p_Q(\mathbf{n}) = O(1/\mathrm{poly}(M))$ for some polynomially many $\mathbf{n}$ and $p_Q(\mathbf{n}) = 0$ otherwise. Then, using the Cauchy-Schwarz inequality,

$$\sum_\mathbf{n}\sqrt{p_Q(\mathbf{n})} \le \sqrt{\sum_\mathbf{n} p_Q(\mathbf{n})\sum_\mathbf{n} 1^2} = \sqrt{\sum_\mathbf{n} 1^2} \\ = O(\mathrm{poly}(M)), \tag{18}$$

where the sum is over the support of $p_Q(\mathbf{n})$, we obtain for $\mathbf{n}$ such that $p_Q(\mathbf{n}) = O(1/\text{poly}(M))$, $p_C(\mathbf{n})$ is at least inverse-polynomially large. Therefore, the classical sampler also takes only polynomial time to obtain the sample $\mathbf{n}^*$, which implies that the Gaussian boson sampler provides at most a polynomial speedup.

Now, suppose that $p_Q(\mathbf{n}) = O(1/\text{poly}(M))$ for polynomially many $\mathbf{n}$ including $\mathbf{n}^*$ and $p_Q(\mathbf{n}) = O(1/\exp(M))$ for other exponentially many outcomes. For simplicity, let us consider an example: $p_Q(\mathbf{n}^*) = 3/4$ and $p_Q(\mathbf{n}) = 1/4^{M+1}$ for 4^M number of $\mathbf{n}$. For such a distribution, the corresponding probability $p_C(\mathbf{n}^*)$ becomes

$$p_C(\mathbf{n}^*) = \frac{\sqrt{3}/2}{\sqrt{3}/2 + 1/2^{M+1} \times 4^M} = O(1/\exp(M)), \tag{19}$$

which is inverse-exponentially small. Therefore, for this example, running the quantum-inspired classical algorithm we proposed would take an exponential time to generate $\mathbf{n}^*$, which shows an exponential separation between the Gaussian boson sampling and our quantum-inspired classical algorithm.

The pertinent question is whether such a nontrivial problem with the above distribution exists. Unfortunately, this might not be the case. Let us consider a similar problem in the Fock-state boson-sampling setting [42], where the probability is expressed by the permanent of the submatrices of a unitary matrix. Mathematically, it was shown that if the output probability (permanent) is large as 1/poly, the relevant matrix has to be very close to a trivial matrix, such as the identity matrix, or permutation matrices [62,63]. We expect that a similar result holds for the hafnian and the Gaussian boson sampling, although it has not been proven to the best of our knowledge. (If it turns out to be false for hafnian, it can provide an exponential speedup over our algorithm, as shown above.) In addition, since the relevant graph-theoretic problems, such as finding the densest k subgraphs and finding the maximum weight clique, are NP-hard, such instances are not the hardest instances of the problems unless quantum devices can solve NP-hard problems efficiently. Nevertheless, the above example implies that if a relevant problem associated with an adjacency matrix A has an outcome $\mathbf{n}^*$ with large $|\text{haf}(A_{\mathbf{n}^*})|^2$ with other exponentially many outcomes having very small probabilities, running a Gaussian boson sampler can provide a significant improvement over our algorithm.

Another more interesting possibility for quantum advantage is the problem in which there exist exponentially many solutions among a much larger number of possible instances, instead of a unique solution. Suppose that we have $4^N/2$ number of solutions that have probability $p_Q(\mathbf{n}) = 1/4^N$ and $4^M/2$ number of instances that are not solutions whose probabilities are $1/4^M$, where N is a positive integer. Then, the probability of obtaining a solution is given by 1/2 for the quantum case. If we translate this into our classical algorithm, the probability of obtaining a solution is given by

$$\sum_{\mathbf{n}:\text{sol}} p_C(\mathbf{n}) = \frac{\sum_{\mathbf{n}:\text{sol}} \sqrt{p_Q(\mathbf{n})}}{\sum_{\mathbf{n}} \sqrt{p_Q(\mathbf{n})}} = \frac{2^N}{2^N + 2^M} = \frac{1}{1 + 2^{M-N}}. \tag{20}$$

Hence, when $M - N = \Omega(N)$, e.g., $M = 2N$, the probability of obtaining a solution becomes exponentially small. Therefore, in this case, whereas the Gaussian boson sampler can efficiently obtain a solution, our classical algorithm takes exponential samples, 2^{M-N}, to obtain a solution. Unlike the unique-solution case, the Gaussian boson sampler may still provide an advantage over our algorithm.

VI. DISCUSSIONS

In this work, we have presented a quantum-inspired classical algorithm for finding the densest k subgraph and finding the maximum weighted clique. We numerically show that although the Gaussian boson sampler may provide an advantage over our algorithm for a fixed number of samples, the advantage is generally not very significant. On the other hand, we provide a potential advantage of a Gaussian boson sampler over our classical algorithm in Sec. V.

However, it might be possible to find a better classical algorithm than our algorithm; in principle, a classical algorithm following the same probability distribution as the Gaussian boson sampler associated with a non-negative matrix A is not prohibited because computing the hafnian of a non-negative matrix within a multiplicative error is in BPP$^{\text{NP}}$ not #P-hard, which is crucial for hardness proof of boson sampling [42,64]. Furthermore, since finding the densest k subgraph and the maximum weighted clique relies on the proportionality between the density and the number of perfect matchings, there may exist a classical sampler whose output probability is proportional to a quantity that is easy to sample from but is still highly proportional to the density. Again, we emphasize that our algorithm is irrelevant to the standard Gaussian boson-sampling hardness [11,65] because the associated matrix for the standard setting is not generally non-negative. In addition, analyzing and generalizing our algorithm for other applications [23,66], such as optimization [67] and graph similarity [33,68–70], would be an interesting future work.

Moreover, in general, more extensive studies on whether the sampler-based algorithms, such as Gaussian boson sampler or our sampler, can outperform other existing methods are another important future work. In

Appendix C, we consider an ant-colony-optimization (ACO) algorithm [71] as an example of classical algorithms for solving the densest k-subgraph problem and show that for typical cases, the ACO algorithm outperforms the sampling-based algorithms (even with the simulated annealing method proposed in Ref. [21]). Nevertheless, there may be some ensemble of graphs for which the sampler-based method surpasses existing classical algorithms, which we leave an open question. Also, it may be possible to incorporate the samplers into existing classical algorithms to enhance the performance such as the ACO algorithm (see Appendix C for more analysis).

In Ref. [72], the so-called independent pairs and singles (IPS) distribution, which was developed for a different purpose, is shown to be classically efficiently samplable. For our purpose, we focus only on pairs. For a given $M\times M$ non-negative matrix A, i.e., $A_{jk}\geq 0$, the distribution is defined as follows:

$$Q(\mathbf{n})=\frac{e^{-\frac{1}{2}\sum_{j,k=1}^{M}A_{jk}}}{\prod_i n_i!}\text{haf}(A_{\mathbf{n}}), \tag{21}$$

where $\mathbf{n}$ corresponds to the output photon-number vector. The sampling algorithm is implemented by generating photon pairs for each mode pairs (j,k) (with $j\leq k$) from a Poisson distribution with mean given by $A_{j,k}$ and combining the photon numbers. First, if we postselect collision-free outcomes for a fixed total photon number, then the probability distribution is equivalent to our proposed algorithm. However, for a given matrix A, the total photon-number distribution of the IPS sampler is fixed, which potentially costs an additional overhead. (This is the case for Gaussian boson samplers as well.) More specifically, when one wants to focus on subgraphs with a fixed number of vertices, the postselection overhead is additionally required. On the other hand, our algorithm can generate the desired photon number only, which does not cause any postselection cost (except for collision-free postselection, which applies to Gaussian boson samplers and IPS samplers as well).

ACKNOWLEDGMENTS

We thank L. Banchi, M. Fingerhuth, T. Babej, C. Ing, and J.M. Arrazola for providing the information of graphs from Ref. [22]. We acknowledge support from the ARO MURI (Grants No. W911NF-16-1-0349, No. W911NF-21-1-0325), AFOSR MURI (Grants No. FA9550-19-1-0399, No. FA9550-21-1-0209), DoE Q-NEXT, NSF (Grants No. OMA-1936118, No. ERC-1941583, No. OMA-2137642), NTT Research, and the Packard Foundation (Grant No. 2020-71479). N.Q. acknowledges support from the Ministère de l’Économie et de l’Innovation du Québec and the Natural Sciences and Engineering Research Council of Canada. B.F. acknowledges support from AFOSR (FA9550-21-1-0008). This material is based upon work partially supported by the National Science Foundation under Grant CCF-2044923 (CAREER) and by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers (Q-NEXT), as well as by DOE QuantISED grant DE-SC0020360. The authors are also grateful for the support of the University of Chicago Research Computing Center for assistance with the numerical simulations carried out in this paper. We acknowledge The Walrus python library for the open source of Gaussian boson-sampling algorithms [53].

APPENDIX A: OUTPUT PROBABILITY OF QUANTUM-INSPIRED CLASSICAL ALGORITHM

Let us first consider a probability of obtaining $\{(k_i,\mathbf{n}^{(i)})\}_{i=1}^N$, where $1\leq k_i\leq K$, i.e., we obtain k_i from the probability distribution $\{q_i\}_{i=1}^K$ and $\mathbf{n}^{(i)}$ from the two-photon boson sampling from $U^{(k_i)}$ for ith trial,

$$p((k_1,\mathbf{n}^{(1)}),\ldots,(k_N,\mathbf{n}^{(N)}))=\prod_{i=1}^N p((k_i,\mathbf{n}^{(i)})) \tag{A1}$$

$$=\prod_{i=1}^N\left(2q_{k_i}|U^{k_i}_{1,\mathbf{n}_1^{(i)}}|^2|U^{k_i}_{1,\mathbf{n}_2^{(i)}}|^2\right), \tag{A2}$$

where $\mathbf{n}_{1,2}^{(i)}$ denotes the indices of the first and the second ones in the vector, respectively. The second line is from Eq. (3) with q_{k_i} for the probability of obtaining k_i.

If we marginalize the information of which two-photon boson sampling we used and individual two-photon outcomes, i.e., k_i’s and $\mathbf{n}^{(i)}$’s, we obtain the probability distribution

$$p(\mathbf{n})=\sum_{k_1,\ldots,k_N=1}^{K}\sum_{\substack{\mathbf{n}^{(1)},\ldots,\mathbf{n}^{(N)}:\\ \sum_{i=1}^N\mathbf{n}^{(i)}=\mathbf{n}}}p((k_1,\mathbf{n}^{(1)}),\ldots,(k_N,\mathbf{n}^{(N)})) \tag{A3}$$

$$=\sum_{k_1,\ldots,k_N=1}^{K}\sum_{\substack{\mathbf{n}^{(1)},\ldots,\mathbf{n}^{(N)}:\\ \sum_{i=1}^N\mathbf{n}^{(i)}=\mathbf{n}}}\prod_{i=1}^N\left(2q_{k_i}|U^{(k_i)}_{1,\mathbf{n}_1^{(i)}}|^2|U^{(k_i)}_{1,\mathbf{n}_2^{(i)}}|^2\right) \tag{A4}$$

$$=\sum_{\substack{\mathbf{n}^{(1)},\ldots,\mathbf{n}^{(N)}:\\ \sum_{i=1}^N\mathbf{n}^{(i)}=\mathbf{n}}}\prod_{i=1}^N\sum_{k_i=1}^K\left(2q_k|U^{(k)}_{1,\mathbf{n}_1^{(i)}}|^2|U^{(k)}_{1,\mathbf{n}_2^{(i)}}|^2\right) \tag{A5}$$

$$=\frac{1}{2^N}\sum_{\sigma\in\mathcal{S}_{2N}}\prod_{i=1}^N\sum_{k=1}^K q_k p^{(k)}_{\sigma(2i-1),\sigma(2i)}, \tag{A6}$$

where the sum over $\mathbf{n}^{(i)}$ stands for the sum over all possible vectors $\{\mathbf{n}^{(i)}\}_{i=1}^{N}$ such that each vector $\mathbf{n}^{(i)}$ is composed of only two ones with other elements being zeros and the sum of the vectors is equal to $\mathbf{n}$. Here, the first line is the marginalization of the information of k_i's and $\mathbf{n}^{(i)}$'s, and the second line is from Eq. (A2), and the third line is obtained by changing the order of the product and the summation, and the last line is by changing the summation for $\mathbf{n}^{(i)}$ by partitioning $2N$ ones in $\mathbf{n}$ into N pairs. Hence, we derived Eq. (4) in the main text.

APPENDIX B: ADDITIONAL PLOTS

In this Appendix, we present additional plots to compare the performance of the Gaussian boson sampler, our quantum-inspired sampler, and the uniform sampler with different random graphs, the number of vertices to be $n = 20, 40, 100$ with fixing $k = 10$, which are shown in Fig. 6. Although we focused on the case of $n = 30, k = 10$ in the main text, the additional figure clearly shows that the performance difference is not significant for different choices of the size of graphs n.

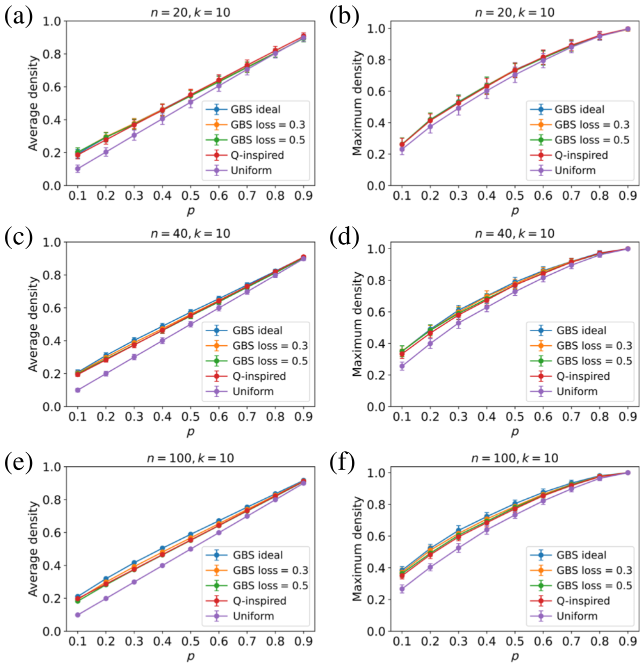

FIG. 6. Density of samples from ideal and lossy Gaussian boson sampler, our classical sampler, and the uniform sampler for random graphs with (a),(b) $n = 20$, (c),(d) $n = 40$, (e),(f) $n = 100$ vertices and $k = 10$ size subgraphs and various p. (a),(c),(e) Average density for different regimes. (b),(d),(f) Maximum density for different regimes. We use 10^2 different graphs with 10^3 number of samples for each p.

APPENDIX C: DISCUSSION ON EXISTING CLASSICAL ALGORITHMS

Besides the algorithm using Gaussian boson sampler [21,40], there have been numerous studies to find classical algorithms for solving the densest k-subgraph problem or finding the maximum weight clique problem due to their importance for many applications. We refer readers to Refs. [55,56,73–77] for the survey.

In this Appendix, we discuss the performance of the algorithm using the Gaussian boson sampler and uniform sampler, our quantum-inspired classical algorithm against a particular classical algorithm focusing on finding the densest k-subgraph problem [71]. The classical algorithm we consider is based on an ant-colony-optimization approach [78], which has been successful for various optimization problems such as the traveling salesman problem [79]. We chose this particular algorithm because of its efficiency despite its simplicity, while we emphasize that we do not claim that this is the best-known classical algorithm and there are various classical algorithms to be compared against.

The high-level idea of this algorithm is based on the behavior of ants to find a path; ants communicate with pheromones, and based on the information obtained by the communication, they can find the shortest path to the destination (e.g., the food source). Therefore, the quality of the previous solutions changes the configuration of pheromones, and the subsequent trials adaptively change the path based on it.

More specifically, suppose ants move along a graph along its edges. When an ant is at the ith vertex, the probability of moving to jth vertex is calculated as

$$p(i,j) = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k\in\mathcal{N}_i}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}}, \quad \text{if } j \in \mathcal{N}_i, \tag{C1}$$

where τ_{ij} is called the pheromone level between nodes i and j, and η_{ij} is the heuristic information, and $\mathcal{N}_i$ is the set of the neighbors of the vertex i. Here, we choose the heuristic information as the density of the jth vertex in the subgraph obtained by subtracting the vertices the ant has moved through from the entire graph (one may choose it differently). After an ant finishes, i.e., the ant moves through k vertices, we obtain a solution and update the pheromone using the quality of the obtained solution:

$$\tau_{ij} \rightarrow \rho\tau_{ij} + Q, \tag{C2}$$

where ρ is called the evaporation coefficient and Q is the quality of the solution, i.e., the density of the obtained solution. Thus, the algorithm is stochastic and adaptive. For the performance comparison, we fix the parameters as the number of ants to be 10 and $\alpha = 1$, $\beta = 3$, $\rho = 0.9$ while the choice of parameters may be further optimized for different graphs.

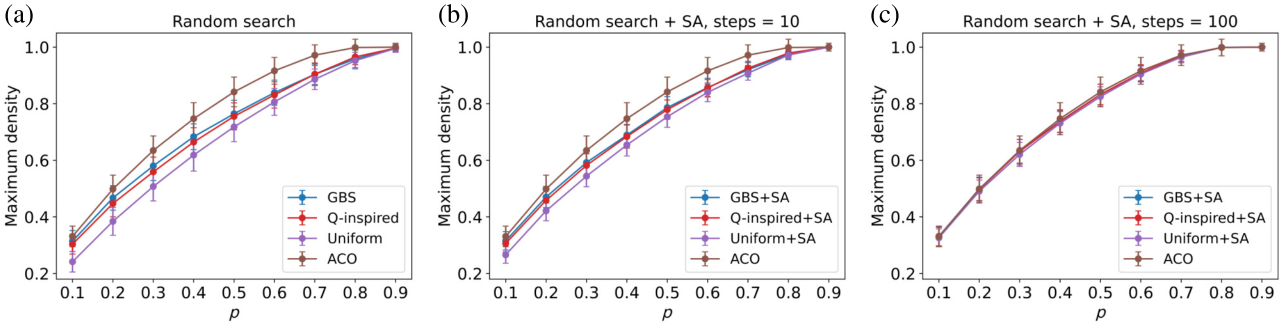

FIG. 7. Maximum density obtained by Gaussian boson sampler, our classical sampler, and the uniform sampler for random graphs of $n = 30$ and $k = 10$ (a) without and (b),(c) with the simulated annealing (SA) of the number of steps to be 10 and 100, respectively.

We first compare the Gaussian, quantum-inspired, uniform samplers to the ACO algorithm for Erdős-Rényi random graphs with $n = 30$ and $k = 10$, which is shown in Fig. 7(a). Clearly, the ACO algorithm's results are significantly denser than the simple sampling methods, namely, random search. However, Refs. [21,40] suggested that such a random search algorithm can be employed for a more sophisticated stochastic method, such as exploration, tweaking, and simulated annealing. The simulated annealing algorithm runs as follows [21]. (1-1) Generate a random l-vertex subgraph R of an input subgraph S of $l < k$ vertices. (1-2) Extend R by picking a uniform random number $0 \leq m \leq k - l$ of the vertices remaining from S, along with the corresponding edges, which yields an $l + m$-vertex graph. (2-1) Generate a random subgraph T of $k - l$ vertices using a sampler. (2-2) Reduce T by randomly picking m vertices. (3) Attach the graph R and T if R and T are disjoint; otherwise, redo until the condition is satisfied. We then use the above step as a subroutine of the so-called simulated annealing; the latter is a stochastic method that replaces the candidate S with a new candidate with probability determined by the temperature and the quality of the candidates. (In our case, density corresponds to the quality.) When the temperature is high, it is likely to replace the candidate if the new candidate is low quality. When the temperature is low, it is unlikely to replace the candidate if the new candidate is low quality, and the temperature decreases as the running time increases. The initial candidate S is chosen by a sampler, and the simulated annealing processes the candidate afterwards. Thus, we introduce an additional classical algorithm to further process samples generated by Gaussian, quantum-inspired, uniform samplers to enhance the performance beyond the simple random search.

With certain parameters, the initial temperature $T = 0.01$ and linear cooling schedule with the number of steps 10 and 100, we again compare with the ACO algorithm in Figs. 7(b) and 7(c). Whereas the improvement by the simulated annealing with the number of steps 10 is not sufficiently significant, we achieve almost the same performance when the number of steps is 100. To analyze the difference more extensively, we also consider structured graphs, i.e., not a generic random graph. We employ the same graph used in Ref. [21], which was generated by (i) generating a random graph of 20 vertices with $p = 0.5$, (ii) generating another random graph of ten vertices with $p = 0.875$, and (iii) selecting eight vertices at random in both graphs and adding an edge between them. We compare the maximum density obtained for different numbers of samples by each method, random search and simulated annealing with different samplers and the ACO algorithm, which are shown in Fig. 8. We can observe from the figure that although adding more steps for the simulated annealing improves the performance for this structured graph, the ACO algorithm finds a higher-density subgraph much faster than other methods. Therefore, overall, the simulated annealing algorithm with Gaussian or quantum-inspired classical samplers does not outperform the ACO algorithm for this case. It is worth noting that the uniform sampler with simulated annealing performs better than other samplers when the number of steps is 100, which may be attributed to the sufficiently large number of steps for the simulated annealing that makes more exploration by uniform sampler beneficial over Gaussian or quantum-inspired sampler weighs on certain patterns.

Since the ACO algorithm seems very powerful, we may also attempt to incorporate the Gaussian boson sampler or

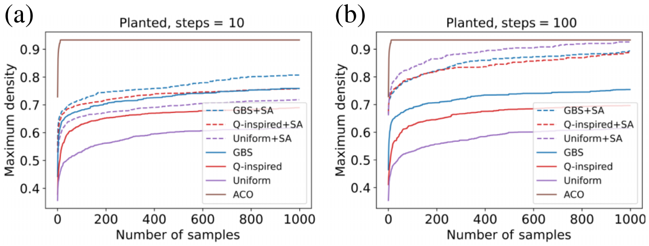

FIG. 8. Maximum density obtained by different schemes for a structured graph of $n = 30$ and $k = 10$ with the number of steps (a) 10 and (b) 100 for simulated annealing (SA).

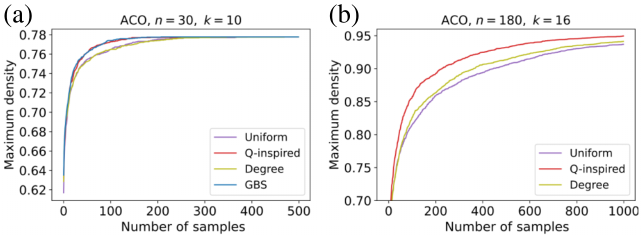

FIG. 9. Maximum density obtained for (a) $n = 30, k = 10$ and (b) $n = 180, k = 16$ structured graphs by the ACO algorithm with different methods of choosing the initial positions of ants. Here, we generate 100 samples to determine the initial positions of ants for the cases of the Gaussian boson sampler and quantum-inspired classical sampler.

our classical sampler into the ACO algorithm. Much like the simulated annealing method, one way to do this is to use the samplers to choose the initial positions of ants. The intuition is that if the ants start from the vertices of dense subgraphs (more precisely, subgraphs with a large hafnian) instead of a random choice of the initial vertices, there may be a higher chance of finding the densest subgraph. To realize the intuition, we first generate a number of samples of k subgraph using a Gaussian boson sampler or the quantum-inspired classical sampler and pick the subgraph with the largest density, and then ants begin at the vertices of the subgraph.

To see whether it actually enhances the performance, we again compare the performance between the ACO algorithms with uniform random initial vertices, degree-weighted random initial vertices, and sampler-assisted initial vertices. Here, the degree-weighted method randomly chooses the initial vertices with probabilities weighted by each vertex's degree, inspired by the sampler-assisted method. We again consider structured graphs and present the results in Fig. 9. For the $n = 30$ and $k = 10$ case, the graph was generated by (i) generating a random graph of n vertices with $p = 0.4$ and (ii) replacing 1–5, 11–15, 21–25 subgraphs by random graphs of five vertices with $p = 0.8$. For the $n = 180$ and $k = 16$ case, the graph was generated by (i) generating a random graph of n vertices with $p = 0.4$ and (ii) replacing 1–30, 116–135, 156–180 subgraphs by random graphs of $30, 20, 25$ vertices with $p = 0.8$. For those cases, we can indeed observe that the initial vertices chosen using a Gaussian boson sampler and our classical sampler enhance the performance of the ACO algorithm. (For the latter, we did not implement the Gaussian boson sampler due to the large computational cost.) Thus, it also shows that the sampling methods may be able to enhance the performance of existing stochastic classical algorithms other than the simulated annealing proposed in Refs. [21,40].

Although we compared the sampling-based algorithms to a specific classical algorithm and provided a possibility of extending it to a hybrid method, we emphasize that more extensive studies of how the Gaussian boson sampler or our classical sampler compares to other existing classical algorithms and whether such a hybrid method can be extended to other classical algorithms are necessary.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, and D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[2] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, and D. Fan, *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. **127**, 180501 (2021).

[3] Q. Zhu, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, and M. Gong, *et al.*, Quantum computational advantage via 60-qubit 24-cycle random circuit sampling, Sci. Bull. **67**, 240 (2022).

[4] A. Morvan, B. Villalonga, X. Mi, S. Mandra, A. Bengtsson, P. Klimov, Z. Chen, S. Hong, C. Erickson, and I. Drozdov, *et al.*, Phase transition in random circuit sampling, arXiv:2304.11119.

[5] A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani, On the complexity and verification of quantum random circuit sampling, Nat. Phys. **15**, 159 (2019).

[6] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nat. Phys. **14**, 595 (2018).

[7] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, and Y. Hu, *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[8] H.-S. Zhong, Y.-H. Deng, J. Qin, H. Wang, M.-C. Chen, L.-C. Peng, Y.-H. Luo, D. Wu, S.-Q. Gong, and H. Su, *et al.*, Phase-programmable Gaussian boson sampling using stimulated squeezed light, Phys. Rev. Lett. **127**, 180502 (2021).

[9] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, and M. J. Collins, *et al.*, Quantum computational advantage with a programmable photonic processor, Nature **606**, 75 (2022).

[10] Y.-H. Deng, *et al.*, Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage, Phys. Rev. Lett. **131**, 150601 (2023).

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

[12] A. Deshpande, A. Mehta, T. Vincent, N. Quesada, M. Hinsche, M. Ioannou, L. Madsen, J. Lavoie, H. Qi, and J. Eisert, *et al.*, Quantum computational advantage via high-dimensional Gaussian boson sampling, Sci. Adv. **8**, eabi7894 (2022).

[13] B. Villalonga, M. Y. Niu, L. Li, H. Neven, J. C. Platt, V. N. Smelyanskiy, and S. Boixo, Efficient approximation of experimental Gaussian boson sampling, arXiv:2109.11525.

[14] X. Gao, M. Kalinowski, C.-N. Chou, M. D. Lukin, B. Barak, and S. Choi, Limitations of linear cross-entropy as a measure for quantum advantage, PRX Quantum **5**, 010334 (2024).

[15] A. Bouland, B. Fefferman, Z. Landau, and Y. Liu, in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Denver, CO, USA, 2022), pp. 1308–1317.

[16] C. Oh, L. Jiang, and B. Fefferman, Spoofing cross-entropy measure in boson sampling, Phys. Rev. Lett. **131**, 010401 (2023).

[17] D. Aharonov, X. Gao, Z. Landau, Y. Liu, and U. Vazirani, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (STOC, New York, NY, USA, 2023).

[18] C. Oh, L. Jiang, and B. Fefferman, On classical simulation algorithms for noisy boson sampling, arXiv:2301.11532.

[19] C. Oh, M. Liu, Y. Alexeev, B. Fefferman, and L. Jiang, Classical algorithm for simulating experimental Gaussian boson sampling, arXiv:2306.03709.

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

[22] L. Banchi, M. Fingerhuth, T. Babej, C. Ing, and J. M. Arrazola, Molecular docking with Gaussian boson sampling, Sci. Adv. **6**, eaax1950 (2020).

[23] T. R. Bromley, J. M. Arrazola, S. Jahangiri, J. Izaac, N. Quesada, A. D. Gran, M. Schuld, J. Swinarton, Z. Zabaneh, and N. Killoran, Applications of near-term photonic quantum computers: Software and algorithms, Quantum Sci. Technol. **5**, 034010 (2020).

[24] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, Trawling the web for emerging cyber-communities, Comput. Netw. **31**, 1481 (1999).

[25] V. Boginski, S. Butenko, and P. M. Pardalos, Mining market data: A network approach, Comput. Oper. Res. **33**, 3171 (2006).

[26] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava, Dense subgraph maintenance under streaming edge weight updates for real-time story identification, Proc. VLDB Endow. **5**, 574 (2012).

[27] B. Balasundaram, S. Butenko, and I. V. Hicks, Clique relaxations in social network analysis: The maximum k-plex problem, Oper. Res. **59**, 133 (2011).

[28] J. Pattillo, N. Youssef, and S. Butenko, in *Handbook of Optimization in Complex Networks: Communication and Social Networks* (Springer, Berlin, 2011), p. 143.

[29] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou, Motifcut: Regulatory motifs finding with maximum density subgraphs, Bioinformatics **22**, e150 (2006).

[30] B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang, in *Annual International Conference on Research in Computational Molecular Biology* (Springer, Berlin, 2010), p. 456.

[31] N. Malod-Dognin, R. Andonov, and N. Yanev, in *International Symposium on Experimental Algorithms* (Springer, Berlin, 2010), p. 106.

[32] S. Yu, Z.-P. Zhong, Y. Fang, R. B. Patel, Q.-P. Li, W. Liu, Z. Li, L. Xu, S. Sagona-Stophel, and E. Mer, *et al.*, A universal programmable Gaussian boson sampler for drug discovery, Nat. Comput. Sci. **3**, 839 (2023).

[33] S. Sempere-Llagostera, R. B. Patel, I. A. Walmsley, and W. S. Kolthammer, Experimentally finding dense subgraphs using a time-bin encoded Gaussian boson sampling device, Phys. Rev. X **12**, 031045 (2022).

[34] Y.-H. Deng, S.-Q. Gong, Y.-C. Gu, Z.-J. Zhang, H.-L. Liu, H. Su, H.-Y. Tang, J.-M. Xu, M.-H. Jia, and M.-C. Chen, *et al.*, Solving graph problems using Gaussian boson sampling, Phys. Rev. Lett. **130**, 190601 (2023).

[35] J. Huh, G. G. Guerreschi, B. Peropadre, J. R. McClean, and A. Aspuru-Guzik, Boson sampling for molecular vibronic spectra, Nat. Photonics **9**, 615 (2015).

[36] H. Jnane, N. P. Sawaya, B. Peropadre, A. Aspuru-Guzik, R. Garcia-Patron, and J. Huh, Analog quantum simulation of non-condon effects in molecular spectroscopy, ACS Photonics **8**, 2007 (2021).

[37] S. Jahangiri, J. M. Arrazola, N. Quesada, and A. Delgado, Quantum algorithm for simulating molecular vibrational excitations, Phys. Chem. Chem. Phys. **22**, 25528 (2020).

[38] C. Oh, Y. Lim, Y. Wong, B. Fefferman, and L. Jiang, Quantum-inspired classical algorithms for molecular vibronic spectra, Nat. Phys. **20**, 225 (2024).

[39] There was a claim that there exists an efficient classical algorithm to solve the problems (e.g., Refs. [80,81]), but to the best of our knowledge, the algorithm and its details have not been presented.

[40] J. M. Arrazola, T. R. Bromley, and P. Rebentrost, Quantum approximate optimization with Gaussian boson sampling, Phys. Rev. A **98**, 012322 (2018).

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

[42] S. Aaronson and A. Arkhipov, in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (Association for Computing Machinery, New York NY USA, 2011), p. 333.

[43] D. Grier, D. J. Brod, J. M. Arrazola, M. B. de Andrade Alonso, and N. Quesada, The complexity of bipartite Gaussian boson sampling, Quantum **6**, 863 (2022).

[44] 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).

[45] A. Serafini, *Quantum Continuous Variables: A Primer of Theoretical Methods* (CRC Press, Boca Raton, FL, 2017).

[46] C. Oh, Y. Lim, B. Fefferman, and L. Jiang, Classical simulation of boson sampling based on graph structure, Phys. Rev. Lett. **128**, 190501 (2022).

[47] M. Jerrum, A. Sinclair, and E. Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries, J. ACM **51**, 671 (2004).

[48] A. Berman and N. Shaked-Monderer, *Completely Positive Matrices* (World Scientific, Singapore, 2003).

[49] A. Barvinok, Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor, Random Struct. Algor. **14**, 29 (1999).

[50] M. Rudelson, A. Samorodnitsky, and O. Zeitouni, Hafnians, perfect matchings and Gaussian matrices, Ann. Probab. **44**, 2858 (2016).

[51] A. Uvarov and D. Vinichenko, On randomized estimators of the hafnian of a nonnegative matrix, arXiv:2312.10143.

[52] U. Feige, D. Peleg, and G. Kortsarz, The dense k-subgraph problem, Algorithmica **29**, 410 (2001).

[53] 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 Softw. **4**, 1705 (2019).

[54] N. R. Solomons, O. F. Thomas, and D. P. S. McCutcheon, Effect of photonic errors on quantum enhanced dense-subgraph finding, Phys. Rev. Appl. **20**, 054043 (2023).

[55] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, in *Handbook of Combinatorial Optimization* (Springer, Boston, MA, 1999), p. 1.

[56] Q. Wu and J.-K. Hao, A review on algorithms for maximum clique problems, Eur. J. Oper. Res. **242**, 693 (2015).

[57] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, A geometric approach to macromolecule-ligand interactions, J. Mol. Biol. **161**, 269 (1982).

[58] F. S. Kuhl, G. M. Crippen, and D. K. Friesen, A combinatorial algorithm for calculating ligand binding, J. Comput. Chem. **5**, 24 (1984).

[59] A. Barvinok, *Combinatorics and Complexity of Partition Functions* (Springer, Berlin, 2016), Vol. 9.

[60] W. Pullan and H. H. Hoos, Dynamic local search for the maximum clique problem, J. Artif. Intell. Res. **25**, 159 (2006).

[61] W. Pullan, Phased local search for the maximum clique problem, J. Comb. Optim. **12**, 303 (2006).

[62] S. Aaronson and H. Nguyen, Near invariance of the hypercube, arXiv:1409.7447.

[63] R. Berkowitz and P. Devlin, A stability result using the matrix norm to bound the permanent, Israel J. Math. **224**, 437 (2018).

[64] L. Stockmeyer, in *Proceedings of the fifteenth annual ACM symposium on Theory of computing* (Association for Computing Machinery, New York, NY, USA, 1983) p. 118.

[65] A. Deshpande, A. Mehta, T. Vincent, N. Quesada, M. Hinsche, M. Ioannou, L. Madsen, J. Lavoie, H. Qi, J. Eisert, D. Hangleiter, B. Fefferman, and I. Dhand, Quantum computational advantage via high-dimensional Gaussian boson sampling, Sci. Adv. **8**, eabi7894 (2022).

[66] R. Mezher, A. F. Carvalho, and S. Mansfield, Solving graph problems with single photons and linear optics, Phys. Rev. A **108**, 032405 (2023).

[67] J. Arrazola, V. Bergholm, K. Brádler, T. Bromley, M. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, and L. Helt, *et al.*, Quantum circuits with many photons on a programmable nanophotonic chip, Nature **591**, 54 (2021).

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

[69] 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).

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

[71] Z. Tasnádi and N. Gaskó, in *2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (IEEE, Linz, Austria, 2022), p. 208.

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

[73] P. M. Pardalos and J. Xue, The maximum clique problem, J. Glob. Optim. **4**, 301 (1994).

[74] S. Khuller and B. Saha, in *International Colloquium on Automata, Languages, and Programming* (Springer, Berlin, 2009), p. 597.

[75] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal, *A Survey of Algorithms for Dense Subgraph Discovery* (Springer, Boston, MA, 2010), p. 303.

[76] R. Sotirov, On solving the densest k-subgraph problem on large graphs, Optim. Methods Softw. **35**, 1160 (2020).

[77] T. Lanciano, A. Miyauchi, A. Fazzone, and F. Bonchi, A survey on the densest subgraph problem and its variants, arXiv:2303.14467.

[78] M. Dorigo and G. Di Caro, in *Proceedings of the 1999 Congress on Evolutionary Computation-Cec99 (Cat. No. 99TH8406)*, Vol. 2 (IEEE, Washington, DC, USA, 1999), p. 1470.

[79] T. Stützle and M. Dorigo, ACO algorithms for the traveling salesman problem (John Wiley & Sons, Inc., New York, NY United States, 1999).

[80] S. Aaronson, https://scottaaronson.blog/?p=5159.

[81] S. Aaronson and S.-H. Hung, Certified randomness from quantum supremacy, arXiv:2303.01625.