nature physics

Supplementary information

https://doi.org/10.1038/s41567-024-02535-8

Classical algorithm for simulating experimental Gaussian boson sampling

In the format provided by the authors and unedited

CONTENTS

S1. SINGLE-MODE DECOMPOSITION

In this Supplemental Material, we consider the decomposition of the simplest example, a single-mode Gaussian state with photon loss. Suppose we have prepared a squeezed vacuum state with squeezing parameter $r \geq 0$, which we assumed nonnegative without loss of generality, whose covariance matrix is given by

$$V_0 = \begin{pmatrix} e^{2r} & 0 \\ 0 & e^{-2r} \end{pmatrix}. \tag{S1}$$

After the photon loss channel, whose transmission rate is given by η and loss rate is thus $1-\eta$, the covariance matrix transforms to

$$V = \eta V_0 + (1-\eta)\mathbb{1}_2 = \begin{pmatrix} \eta e^{2r} + 1 - \eta & 0 \\ 0 & \eta e^{-2r} + 1 - \eta \end{pmatrix}. \tag{S2}$$

By setting $\eta e^{-2r} + (1-\eta) = e^{-2s}$, we can decompose the covariance matrix V as

$$V = \begin{pmatrix} e^{2s} & 0 \\ 0 & e^{-2s} \end{pmatrix} + \begin{pmatrix} \eta e^{2r} + 1 - \eta - e^{2s} & 0 \\ 0 & 0 \end{pmatrix} \equiv V_p + W, \tag{S3}$$

where W's first matrix element is nonnegative when $r \geq 0$. Here we call s an actual squeezing parameter because this is the squeezing parameter that is actually involved in complexity, which will be clear below. We note that such a decomposition is not unique. For our purpose, however, we maximize the matrix W and minimize the energy of the pure Gaussian state V_p to minimize the quantum resource (see below for another decomposition from the Williamson decomposition.). We emphasize that a similar decomposition has been used to accelerate the classical algorithm for simulating Gaussian boson sampling quadratically in Ref. [1]. However, the purpose of the decomposition was to avoid simulating the density matrix rather than a pure state, not to take advantage of the fact that the thermal part, namely, the random displacement part, should not significantly increase the complexity. Furthermore, the optimization we implement significantly minimizes the simulation cost. As an example, the Williamson decomposition from Ref. [1] gives the following decomposition:

$$V = \begin{pmatrix} e^{t} & 0 \\ 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 2n_{\mathrm{nth}} + 1 & 0 \\ 0 & 2n_{\mathrm{nth}} + 1 \end{pmatrix} \begin{pmatrix} e^{t} & 0 \\ 0 & e^{-t} \end{pmatrix} \tag{S4}$$

$$= \begin{pmatrix} e^{t} & 0 \\ 0 & e^{-t} \end{pmatrix} \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 2 \begin{pmatrix} n_{\mathrm{th}} & 0 \\ 0 & n_{\mathrm{th}} \end{pmatrix} \right] \begin{pmatrix} e^{t} & 0 \\ 0 & e^{-t} \end{pmatrix} \tag{S5}$$

$$= \begin{pmatrix} e^{2t} & 0 \\ 0 & e^{-2t} \end{pmatrix} + 2 \begin{pmatrix} n_{\mathrm{th}} e^{2t} & 0 \\ 0 & n_{\mathrm{th}} e^{-2t} \end{pmatrix}. \tag{S6}$$

Here, the squeezing parameter of the pure state is

$$t = \frac{1}{4} \log \left(\frac{\eta e^{2r} + 1 - \eta}{\eta e^{-2r} + 1 - \eta} \right), \tag{S7}$$

and

$$n_{\mathrm{th}} = \frac{1}{2} \left(\sqrt{(\eta e^{2r} + 1 - \eta)(\eta e^{-2r} + 1 - \eta)} - 1 \right). \tag{S8}$$

One can easily show that $t > s$, namely, our minimization, gives a smaller photon number for the quantum part.

S2. EXISTING CLASSICAL ALGORITHMS AND OUR CLASSICAL ALGORITHM

In this Supplemental Material, we compare our classical algorithm with existing algorithms simulating Gaussian boson sampling. First, we compare it with the *exact* simulation algorithms from Refs. [1–3]. These exact simulation algorithms are typically used for estimating the running time for classical computers to simulate the experiments [4–6]. One significant limitation of such algorithms is that their complexity increases substantially even if we simply introduce a large displacement before the photon number measurement. This is because the large displacement will introduce more output photons on average, which increases the matrix size of which we compute the (loop) hafnian for sampling. It implies that the classical algorithms do not discriminate between quantum resources, which are photons

from squeezed states, and classical resources, which are photons from displacement, as long as there exist quantum resources. Therefore, even when there are a large number of thermal photons due to photon loss, which is the case in the current experiments (see Table I in the main text), the complexity analysis based on these types of algorithms inevitably overestimates the complexity of the corresponding experiments. Furthermore, the appropriate target of hardness is typically an approximate simulation instead of an exact simulation [7–9]. Therefore, the running time from these exact algorithms overestimates the complexity of the existing experiments, taking into account additional experimental noise.

Let us now introduce *approximate* classical algorithms designed to simulate *lossy* boson sampling. Various types of algorithms exist for this purpose. First, we consider efficient classical algorithms using classical states. The key idea of these algorithms is to find the closest classical state, such as a thermal state [10–12] or a separable state [13], which are easy to efficiently sample from. One significant drawback of these algorithms is that they cannot improve the accuracy by increasing the running time because the closest classical state is fixed for each algorithm for given parameters of a lossy boson sampler, and thus we cannot enhance the quality of the approximate state by increasing the running time. Therefore, although these approximations provide a boundary of efficient classical sampling and an important benchmark for demonstrating quantum advantage, they are not appropriate to simulate lossy boson samplers or characterize their complexity unless the loss rate is very large.

Another relevant type of classical algorithm that takes advantage of photon loss is matrix product operators [14–16]. The main idea is that since external noise, such as photon loss, reduces the amount of the entanglement of the output state, such a tensor network method may be able to reduce the complexity of simulating the corresponding lossy boson samplers. However, although the required bond dimension does not necessarily increase rapidly for the high loss regime, simulating the recent experiments is still too expensive as the output photon number increases. Another drawback is that since the matrix product operators simulate density matrices, the cost is at least quadratically larger than simulating a pure state, which is the case in the present work.

Our algorithm addresses such drawbacks of the existing algorithms. Although our algorithm is generally an approximate sampler because of the truncation of singular values, the complexity of our algorithm does not increase by introducing a displacement on the output state. Therefore, one can significantly reduce the complexity when a given boson sampler contains a large number of thermal photons. In addition, our algorithm can efficiently improve its accuracy from a thermal-state approximation to the ideal distribution by increasing the running time depending on the target approximation error, which is due to the fact that the MPS method enables us to take into account the contribution from V_p in contrast to the thermal state approximation that approximates V_p by a trivial vacuum state's covariance matrix $\mathbb{1}_{2M}$. As emphasized repeatedly in the main text, an interesting feature of our algorithm is that in one extreme case where we set the MPS part to be a vacuum state, it becomes a thermal state approximation, which recovers the concept of the previous results in Refs. [10, 11]. Also, our new algorithm has a significant advantage over the previous matrix product operator algorithms [15, 16] in that we remove the thermal photons W, which enables a drastic reduction in resource utilization for the MPS algorithm.

At a high level, our algorithm resembles another type of classical algorithm designed for approximate noisy samplers exploiting the fact that the output probability of noisy quantum devices typically converges to an easy distribution [17–24]. More formally, the noisy output probabilities can be expanded as polynomials with different degrees in terms of the noise parameter, with the lowest degree of polynomials being a distribution that is easy to sample from (e.g., uniform distribution in Refs. [23, 24]). Thus, by appropriately choosing the degree, one can control the accuracy of the approximate samplers by increasing the running time. Our approximate sampler resembles this type of algorithm in that one extreme case gives a distribution that is easy to sample from, and as we increase the running time, we can improve the algorithm's accuracy. To the best of our knowledge, however, this type of classical algorithm requires a large amount of noise to be efficient and so has not been implemented for finite-size experiments yet.

Still another type of classical algorithm is specialized to spoof benchmarking but not necessarily to simulate the ground-truth sampler [25]. We emphasize that the proposed algorithm in this work is not designed to spoof benchmarking but truly to approximately simulate the ground-truth distribution.

S3. MPS CONSTRUCTION

In this Supplemental Material, we provide the method for MPS construction of Gaussian states based on Ref. [26]. While the method in that reference is typically inefficient, we employ the property of Gaussian states to make it efficient so that we can efficiently find the reduced density matrix of a bipartition $A : B$ and its spectral decomposition because the reduced density matrix is still a Gaussian state and Gaussian states can always be written as a product of thermal

states $\hat{\rho}_T$ followed by a Gaussian unitary operation [27]:

$$\hat{\rho}_B = \mathrm{Tr}_A[|\psi\rangle\langle\psi|] = \hat{U}_B\hat{\rho}_T\hat{U}_B^\dagger = \sum_{\boldsymbol{n}=\boldsymbol{0}}^{\infty} p_T(\boldsymbol{n})\hat{U}_B|\boldsymbol{n}\rangle\langle\boldsymbol{n}|\hat{U}_B^\dagger, \tag{S9}$$

where $p_T(\boldsymbol{n}) = \prod_{i\in B}\bar{n}_i^{n_i}/(\bar{n}_i+1)^{n_i+1}$ and $\bar{n}_i$ is the mean photon number of the ith mode's thermal state. One can easily find $\hat{U}_B$ and $\{\bar{n}_i\}_{i\in B}$ by Williamson decomposition of the covariance matrix of the state in the B part [27]. Hence, the eigenstates of the reduced density matrix are always a number state followed by a Gaussian unitary operation.

Here we recall the method of constructing an MPS proposed in Ref. [26] with adapting Gaussian states' properties. First, we apply the singular value decomposition along the first mode and the rest of the modes with a prechosen bond dimension χ:

$$|\psi\rangle \approx \sum_{\alpha_1=0}^{\chi-1}\lambda_{\alpha_1}^{[1]}|\Phi_{\alpha_1}^{[1]}\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle = \sum_{n_1=0}^{d-1}\sum_{\alpha_1=0}^{\chi-1}\Gamma_{\alpha_1}^{[1]n_1}\lambda_{\alpha_1}^{[1]}|n_1\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle = \sum_{n_1=0}^{d-1}\sum_{\alpha_1=0}^{\chi-1}A_{\alpha_1}^{[1]n_1}|n_1\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle, \tag{S10}$$

where

$$A_{\alpha_1}^{[1]n_1} = \langle n_1^{[1]}|\langle\Phi_{\alpha_1}^{[2\cdots M]}|\psi\rangle = \langle n_1^{[1]}|\langle\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}|\hat{U}^{[2\cdots M]\dagger}\hat{U}|\boldsymbol{n}=\boldsymbol{0}\rangle. \tag{S11}$$

Here the approximation is due to truncation from the predetermined bond dimension, and the state $|\psi\rangle$ can always be written as $|\psi\rangle = \hat{U}|\boldsymbol{n}=\boldsymbol{0}\rangle$ by Williamson decomposition of a pure Gaussian state; and, as emphasized before, the singular values $\lambda_{\alpha_1}^{[1]}$ can be easily found by performing the Williamson decomposition for the marginal covariance matrix over the bipartition between the first mode and the rest of the modes as in Eq (S9). Also, the eigenstates $\{|\Phi_{\alpha_1}^{[1]}\rangle\}$, $\{|\Phi_{\alpha_1}^{[2\cdots M]}\rangle\}$ are always photon number states followed by Gaussian unitary operations. Thus, we can characterize the eigenstate $|\Phi_{\alpha_1}^{[2\ldots M]}\rangle = \hat{U}^{[2\cdots M]}|\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\rangle$ as $\{\hat{U}^{[2\cdots M]}, \boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\}$. We then rewrite it in number basis $|n_2^{[2]}\rangle$ as

$$|\Phi_{\alpha_1}^{[2\cdots M]}\rangle = \sum_{n_2=0}^{d-1}|n_2^{[2]}\rangle|\tau_{\alpha_1 n_2}^{[3\cdots M]}\rangle, \tag{S12}$$

where

$$|\tau_{\alpha_1 n_2}^{[3\cdots M]}\rangle = \langle n_2^{[2]}|\Phi_{\alpha_1}^{[2\cdots M]}\rangle, \tag{S13}$$

where we expand by the eigenstates of the reduced density matrix $\{|\Phi_{\alpha_2}^{[3\cdots M]}\rangle\}_{\alpha_2=0}^{\chi-1}$

$$|\tau_{\alpha_1 n_2}^{[3\cdots M]}\rangle \approx \sum_{\alpha_2=0}^{\chi-1}A_{\alpha_1\alpha_2}^{[2]n_2}|\Phi_{\alpha_2}^{[3\cdots M]}\rangle = \sum_{\alpha_2=0}^{\chi-1}\Gamma_{\alpha_1\alpha_2}^{[2]n_2}\lambda_{\alpha_2}^{[2]}|\Phi_{\alpha_2}^{[3\cdots M]}\rangle \tag{S14}$$

$$A_{\alpha_1\alpha_2}^{[2]n_2} = \langle n_2^{[2]}|\langle\Phi_{\alpha_2}^{[3\cdots M]}|\Phi_{\alpha_1}^{[2\cdots M]}\rangle, \tag{S15}$$

where $A_{\alpha_1\alpha_2}^{[2]n_2} = \Gamma_{\alpha_1\alpha_2}^{[2]n_2}\lambda_{\alpha_2}^{[2]}$, and $|\Phi_{\alpha_2}^{[3\cdots M]}\rangle$ is the eigenstate of the reduced density matrix of the $[3\cdots M]$ part and $\lambda_{\alpha_2}^{[2]}$ are the singular values, which can be easily identified. Practically, we need only to compute matrices A by

$$A_{\alpha_1\alpha_2}^{[2]n_2} = \langle n_2^{[2]}|\langle\Phi_{\alpha_2}^{[3\cdots M]}|\Phi_{\alpha_1}^{[2\cdots M]}\rangle = \langle n_2^{[2]}|\langle\boldsymbol{n}_{\alpha_2}^{[3\cdots M]}|\hat{U}^{[3\cdots M]\dagger}\hat{U}^{[2\cdots M]}|\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\rangle = \langle n_2^{[2]}|\langle\boldsymbol{n}_{\alpha_2}^{[3\cdots M]}|\hat{V}^{[2\cdots M]}|\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\rangle, \tag{S16}$$

where $\hat{V}^{[2\cdots M]} \equiv \hat{U}^{[3\cdots M]\dagger}\hat{U}^{[2\cdots M]}$. By iterating this procedure, we obtain all the matrix elements, which is summarized as

$$A_{\alpha_1}^{[1]n_1} = \langle n_1^{[1]}|\langle\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}|\hat{U}^{[2\cdots M]\dagger}\hat{U}^{[1\cdots M]}|\boldsymbol{n}=\boldsymbol{0}\rangle, \tag{S17}$$

$$A_{\alpha_{k-1}\alpha_k}^{[k]n_k} = \langle n_k^{[k]}|\langle\boldsymbol{n}_{\alpha_k}^{[(k+1)\cdots M]}|\hat{U}^{[(k+1)\cdots M]\dagger}\hat{U}^{[k\cdots M]}|\boldsymbol{n}_{\alpha_{k-1}}^{[k\cdots M]}\rangle \quad \text{for } \ 1<k<M, \tag{S18}$$

$$A_{\alpha_{M-1}}^{[M]n_M} = \langle n_M^{[M]}|\hat{U}^{[M]}|n_{\alpha_{M-1}}^{[M]}\rangle, \tag{S19}$$

and

$$\Gamma_{\alpha_1}^{[1]n_1} = A_{\alpha_1}^{[1]n_1}/\lambda_{\alpha_1}^{[1]}, \tag{S20}$$

$$\Gamma_{\alpha_{k-1}\alpha_k}^{[k]n_k} = A_{\alpha_{k-1}\alpha_k}^{[k]n_k}/\lambda_{\alpha_k}^{[k]} \ \text{ for } \ 1<k<M, \tag{S21}$$

$$\Gamma_{\alpha_{M-1}}^{[M]n_M} = A_{\alpha_{M-1}}^{[M]n_M}. \tag{S22}$$

Therefore, the remaining calculation to obtain all the matrix elements is $\langle \boldsymbol{n}_1|\hat{V}|\boldsymbol{n}_2\rangle$, for number states $|\boldsymbol{n}_1\rangle$ and $|\boldsymbol{n}_2\rangle$ and a Gaussian unitary operator $\hat{V}$. This quantity has already been studied in Refs. [28, 29] by noting that any Gaussian unitary operation can be decomposed as $\hat{V} = \hat{U}_2\hat{S}(\boldsymbol{r})\hat{U}_1$, where passive unitary operations $\hat{U}_1$ and $\hat{U}_2$ and single-mode squeezers $\hat{S}(\boldsymbol{r}) = \otimes_i \hat{S}(r_i)$:

$$\langle \boldsymbol{n}_1|\hat{U}_2\hat{S}(\boldsymbol{r})\hat{U}_1|\boldsymbol{n}_2\rangle = \frac{\mathrm{haf}(\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2})}{\sqrt{\boldsymbol{n}_1!\boldsymbol{n}_2!\prod_i \cosh r_i}}, \tag{S23}$$

where U_2 and U_1 correspond to the unitary matrix that characterize the unitary operators $\hat{U}_2$ and $\hat{U}_1$, and Σ is a matrix obtained by

$$\Sigma = \begin{pmatrix} U_2 & 0 \\ 0 & U_1^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \tanh \boldsymbol{r} & \operatorname{sech} \boldsymbol{r} \\ \operatorname{sech} \boldsymbol{r} & -\tanh \boldsymbol{r} \end{pmatrix} \begin{pmatrix} U_2^{\mathrm{T}} & 0 \\ 0 & U_1 \end{pmatrix}, \tag{S24}$$

where $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is obtained by repeating Σ's block matrices by $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ times. Hence, the complexity of obtaining all the matrix elements of A, or equivalently Γ and λ, is $O(Md\chi^2 \times (\text{hafnian of } \Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}))$, and the complexity of computing the hafnian of $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is $\tilde{O}(2^{(\boldsymbol{n}_1+\boldsymbol{n}_2)/2})$ [30, 31]. Therefore, two crucial factors determine the complexity: the bond dimension χ and the maximum of $|\boldsymbol{n}_1 + \boldsymbol{n}_2| \equiv \sum_i((\boldsymbol{n}_1)_i + (\boldsymbol{n}_2)_i)$. While we study the scaling of the bond dimension χ in Methods of the main text more comprehensively, both the bond dimension and the maximum $|\boldsymbol{n}_1+\boldsymbol{n}_2|$ are affected by the amount of entanglement. The former is evident, and the latter is because for a pure multimode Gaussian state, the reduced state on part B over a bipartition $A:B$ becomes more thermalized when the parties A and B are highly entangled. For example, if they are a product state, the reduced state is still a pure state. Also, we emphasize that the matrix size of $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is much smaller than the matrix size for computing the output probability of the actual output state, which includes the random displacement. Hence, our MPS construction is, in general, much more efficient than directly sampling from the output state using the best-known classical algorithm [1, 3] when the loss rate is large.

S4. SINGULAR VALUES OF OUTPUT STATE OF GAUSSIAN BOSON SAMPLING

In this Supplemental Material, we analyze the distribution of singular values of MPS more quantitatively. By analysis, we show that we need only the bond dimension $\chi = \mathrm{poly}\log(1/\epsilon)$ for a fixed number of squeezed states and squeezing parameters and that we may not efficiently reduce the simulation error for the transmission rate scaling as $\eta = O(1/\sqrt{N})$. To this end, we consider K two-mode squeezed states out of M mode for fixed K and M and provide a way to truncate singular values as in Methods of the main text. We directly study K two-mode squeezed states' singular values for a bipartition instead of entanglement entropy. Then the reduced density matrix of one part of two-mode squeezed states can be written as a product of thermal states

$$\hat{\rho}_T(\bar{n})^{\otimes K} = \bigotimes_{i=1}^{K}\left(\sum_{n_i=0}^{\infty} p_{\bar{n}}(n_i)|n_i\rangle\langle n_i|\right) = \sum_{\boldsymbol{n}=0}^{\infty} p_{\bar{n}}(\boldsymbol{n})|\boldsymbol{n}\rangle\langle\boldsymbol{n}| = \sum_{k=0}^{\infty}\sum_{\boldsymbol{n}:\sum_i n_i=k}(1-R)^K R^k|\boldsymbol{n}\rangle\langle\boldsymbol{n}|, \tag{S25}$$

where $\bar{n} = \sinh^2 s$ is the mean photon number of thermal states $\hat{\rho}_T$ obtained by partial trace, $p_{\bar{n}}(n_i) \equiv \bar{n}_i^{n_i}/(\bar{n}_i+1)^{n_i+1}$, and $p_{\bar{n}}(\boldsymbol{n}) \equiv \prod_{i=1}^{K} p_{\bar{n}}(n_i)$. Here, we have decomposed the sum into the total photon number sector k and outcomes that have the same photon number, namely, $\boldsymbol{n}$ such that $\sum_i n_i = k$. For the last expression, we define $R \equiv \bar{n}/(\bar{n}+1)$. Then for each sector k, we have eigenvalues $(1-R)^K R^k$ for $\binom{K+k-1}{k}$ terms. Thus, the distribution of the sum of eigenvalue for each sector k follows a negative binomial distribution. Now we analyze the error when we truncate this up to l photon sector. For this case the number of singular values we keep is given by

$$\chi_l \equiv \sum_{k=0}^{l}\binom{K+k-1}{k}, \tag{S26}$$

which becomes the bond dimension we use for MPS. The probability we lose by the truncation is

$$\epsilon_l \equiv \sum_{k=l+1}^{\infty}\binom{K+k-1}{k}(1-R)^K R^k = 1 - I_{1-R}(K, l+1), \tag{S27}$$

where I_R is the regularized incomplete beta function. Here, since $(1-R)^K R^k$ is a monotonically decreasing function in k, the best way of choosing the singular values to minimize the error is to choose from the lowest photon number sectors. We also note that the truncation number l determines the largest $|\boldsymbol{n}_1+\boldsymbol{n}_2|$ that appeared in the direction construction of MPS of the output Gaussian state (see Methods in the main text). Hence, if l needs to increase at most logarithmically, the part for computing hafnians for the direct construction of the MPS is efficient.

Note that the tail probability of binomial distribution with probability p of obtaining at least $k+1$ success out of n trials, or, equivalently, multiple Bernoulli trials, is $1-I_{1-p}(n-k,1+k)$. Thus, the right-hand side is equivalent to obtaining at least $l+1$ successes from the binomial with $K+l$ trials with success probability R. Using concentration inequality for $X\equiv\sum_{i=1}^{K+l}X_i$ with Bernoulli random variable X_i with probability R [32], we have

$$\Pr[X>(1+\delta)\mathbb{E}[X]]\leq\exp\left(-\frac{\delta^2}{3}\mathbb{E}[X]\right), \tag{S28}$$

and choosing $\delta=l/\mathbb{E}[X]-1=l/[(K+l)R]-1>0$ (thus, $l>KR/(1-R)$), the probability we lose by truncation is upper bounded as

$$\epsilon_l=1-I_{1-R}(K,l+1)=\Pr[X>l]\leq\exp\left(-\frac{1}{3}\left(\frac{l}{(K+l)R}-1\right)^2(K+l)R\right)\leq\exp\left(-\frac{1}{3}c^2lR\right), \tag{S29}$$

Now let us consider a fixed-size circuit, that is, one in which K, $\bar{n}$, and R are fixed. Then

$$\epsilon_l\leq\exp\left(-\frac{1}{3}\left(\frac{l}{(K+l)R}-1\right)^2(K+l)R\right)\leq\exp\left(-\frac{1}{3}c^2lR\right), \tag{S30}$$

where c is a nonzero constant lower bound of δ for $l>KR/(1-R)$. Thus, in order to bound the error by ϵ, it is sufficient to choose

$$l\geq\frac{-3\log\epsilon}{c^2R}=O(\log\epsilon^{-1}), \tag{S31}$$

which leads to the upper bound of the bond dimension

$$\chi_l=\sum_{k=0}^{l}\binom{K+k-1}{k}\leq(l+1)\binom{K+l-1}{l}\leq(l+1)\frac{(K+l-1)^{K-1}}{(K-1)!}=O(l^K)=O\left(\left(\log\epsilon^{-1}\right)^K\right). \tag{S32}$$

Therefore, the bond dimension to achieve error ϵ is $\chi=\operatorname{poly}\log\left(\epsilon^{-1}\right)$. It also guarantees that the computation of hafnians is efficient for the direct construction of MPS since l can be chosen to be $O(\log\epsilon^{-1})$.

Now let us turn our attention to the case where $\eta=\Theta(1/\sqrt{K})$. When $\eta=\Theta(1/\sqrt{K})$, $s=\Theta(1/\sqrt{K})$, $\bar{n}=\Theta(1/K)$, and $R=\bar{n}/(\bar{n}+1)\approx 1/K$. First, we note that if we select $l=0$, the number of singular values is $\chi_0=1$ and the error is also $\epsilon_0=1-(1-R)^K\approx 1-(1-1/K)^K\approx 1-1/e$. Thus, it recovers the same scaling as Refs. [10, 11, 13].

Again, using concentration inequality for $X\equiv\sum_{i=1}^{K+l}X_i$ with Bernoulli random variable X_i with probability R and choosing $\delta=l/\mathbb{E}[X]-1=l/[(K+l)R]-1>0$ (thus, $l>KR/(1-R)$), we have

$$\epsilon_l\leq\exp\left(-\frac{1}{3}\left(\frac{l}{(K+l)R}-1\right)^2(K+l)R\right). \tag{S33}$$

The difference from the fixed-size circuit case is that we are interested in the regime where $R\approx\bar{n}\approx 1/K$. Note that we chose l to be larger than $KR/(1-R)=K\bar{n}=O(1)$ for $R=O(1/K)$. Then the bound we obtained gives, for large K,

$$\exp\left(-\frac{1}{3}\left(\frac{l}{(K+l)R}-1\right)^2(K+l)R\right)\approx\exp\left(-\frac{1}{3}\left(\frac{Kl}{(K+l)}-1\right)^2\frac{K+l}{K}\right)\approx\exp\left(-\frac{1}{3}(l-1)^2\right), \tag{S34}$$

and thus the cost to achieve ϵ is

$$\chi_l=\sum_{k=0}^{l}\binom{K+k-1}{k}\leq(l+1)\binom{K+l-1}{l}\leq(l+1)\frac{(K+l-1)^l}{l!}=O(l^K)=O\left(\left(\log\frac{1}{\epsilon}\right)^{K/2}\right). \tag{S35}$$

Therefore, the upper bound of the cost from the concentration inequality increases exponentially in K.

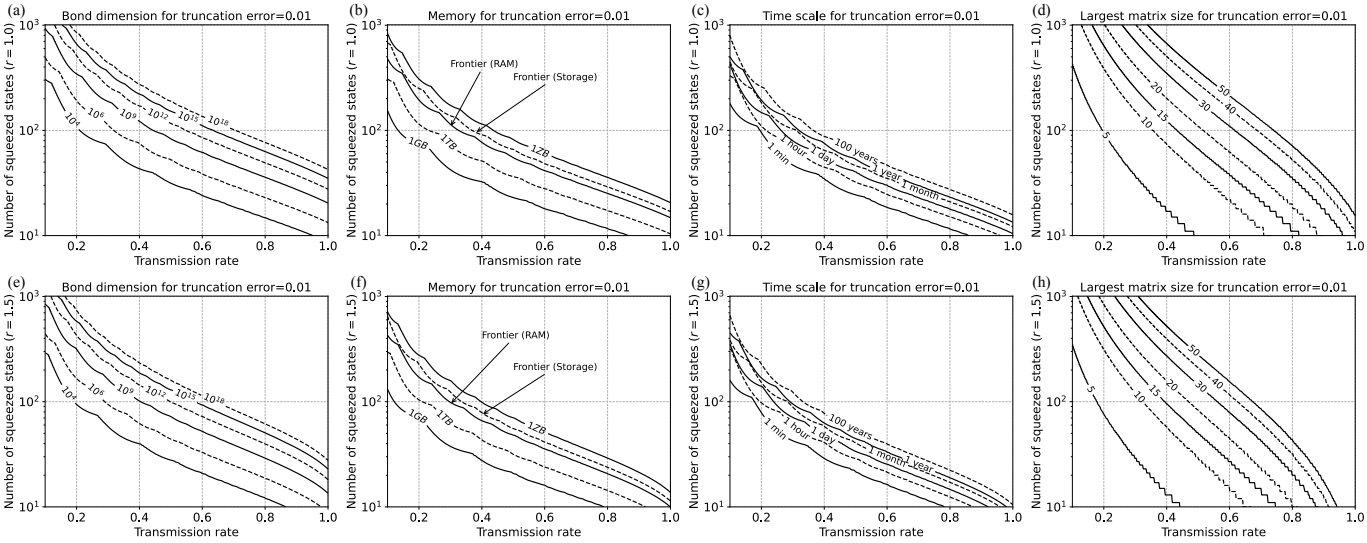

FIG. S1. Computational cost estimate for constructing an MPS that achieves truncation error 0.01 with squeezing parameter (a-d) $r = 1.0$ and (e-h) $r = 1.5$. The costs are characterized by (a)(e) MPS bond dimension, (b)(f) memory usage, (c)(g) time scale, and (d)(h) the matrix size for which we have to compute the hafnian.

S5. IMPLEMENTATION

We implement our simulation algorithm using Python. Specifically, GPU computations are optimized by using the CuPy library, and distributed computation is achieved by using the Message Passing Interface (MPI) using the MPI for Python library. We also develop a custom CUDA kernel for a minor subroutine in CUDA C++, which interfaces with Python through CuPy. Part of the algorithm is taken or modified from the source code of the Strawberry Fields library [33] as well as The Walrus library [34].

The simulation algorithm uses a single GPU for the computation and storage of a single-mode MPS tensor. During the MPS calculation, all tensors are computed independently on different GPUs, which fills tensor entries with appropriately calculated hafnian values. Hafnians of equal-sized square matrices are computed in parallel for numerical efficiency, and this is possible because the hafnian calculation algorithm is data-independent. To avoid impractical memory costs, we limit the maximum number of parallel hafnians to a value dependent on the size of the input matrices, and we loop over subbatches to complete all hafnians. After a tensor is completed, the tensor is saved to the local SSD for later use during sampling.

Additionally, since the computational cost for different modes varies dramatically, GPUs that completed the designated tensor calculations are used to accelerate the computation of more costly tensors. Specifically, when challenging tensors have too many parallel hafnians to compute and other GPUs are available, subbatches are sent via the communication fabric for computation, and the results are later collected.

For the sampling algorithm, the computed single-mode MPS tensors are loaded to each GPU. During the sampling procedure, a vector is passed from one mode to the next via the communication fabric after some operations with local tensors for sampling. After the vector is sent to the next GPU, a new tensor is received from the previous GPU for new samples, resulting in a stream of samples that propagate through the chain. This process is also performed in a parallel manner via batch parallel random displacement generation, matrix multiplications, and weighted random choices. For the largest simulations, 100 samples are processed in parallel on a single GPU for numerical efficiency and reasonable memory costs. Therefore, $100 \times M$ samples are processed in parallel on M GPUs.

S6. COMPUTATIONAL COST ESTIMATION

In this Supplemental Material, we further provide estimates of the computational cost of our algorithm for simulating larger systems and that for simulating systems that take into account the experimental architecture. First, as in the main text, to make the analysis simple and consistent, we again assume the same conditions for the setup: (i) input squeezing parameters and loss rates are uniform across the modes, (ii) all the inputs are squeezed states without any vacuum states although they are not necessarily satisfied in the current experiments. We emphasize that while the USTC's experiments used a different number of squeezed state inputs, 50, from the number of modes, 144, i.e., many

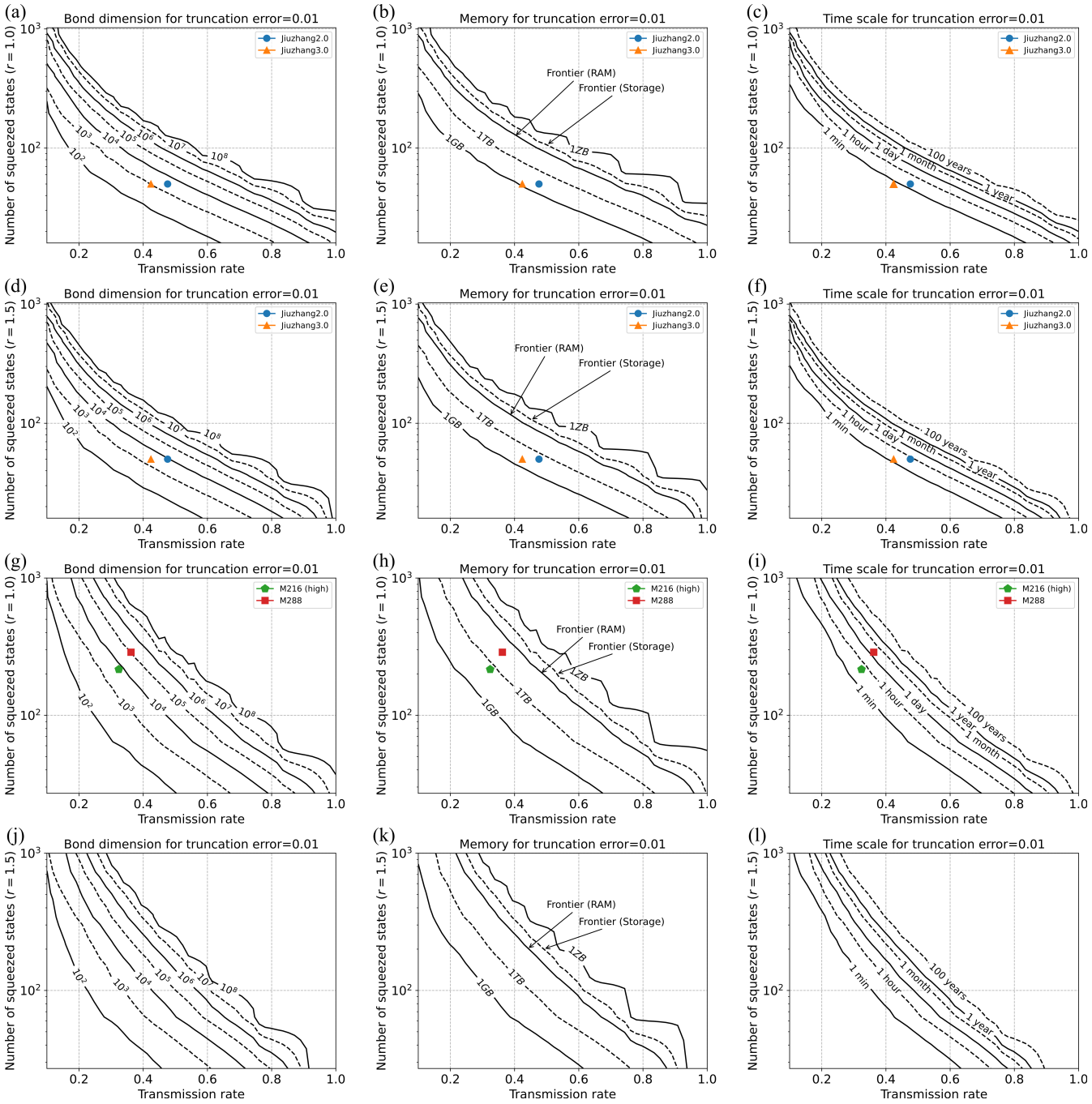

FIG. S2. Computational cost estimate for constructing an MPS that achieves truncation error 0.01 with squeezing parameter (a-c) and (g-i) $r = 1.0$ and (d-f) and (j-l) $r = 1.5$. We consider the USTC's setup in (a-f) and Xanadu's setup in (g-l). The costs are characterized by (a)(d)(g)(j) MPS bond dimension, (b)(e)(h)(k) memory usage, and (c)(f)(i)(l) time scale. The dots represent the relevant parameters in each experiment, and we do not present them in (j-l) because Xanadu's largest squeezing parameter is lower than $r = 1.5$. We averaged over ten different random circuits.

of the inputs were the vacuum, in our analysis, we set them to be equal to remove the dependence on the position of input squeezed states.

We again consider three different circuit ensembles: (a) the worst-case circuits as in Extended Data Fig. 2, (b) the circuits similar to the USTC's experiments, and (c) the circuits similar to Xanadu's experiments. More specifically, for (b) and (c), we model the USTC's experiment using the same structure with local beam splitters and assume that local beam splitters follow Haar-random unitaries. We also model Xanadu's experiment using the same 3D structure

with local Haar-random random unitary beam splitters.

We present the estimate for the worst-case circuit in Fig. S1 and the one for the circuit modeling the experiments in Fig. S2. Here, the required bond dimension for achieving a certain truncation error is obtained by considering the bipartition in Extended Data Fig. 2(c) and using Eq. (S27) for the worst-case circuits analytically and directly numerically computing all the largest singular values for the USTC and Xanadu's setup. The required memory usage is estimated by

$$(8 \text{ bytes}) \times (\text{bond dimension})^2 \times (\text{the number of modes}) \times (\text{local Hilbert space dimension}), \tag{S36}$$

where 8 bytes is for Complex64 variables, and we set local Hilbert space dimension 4 as our simulation. The time scale represents the time cost to prepare the MPS by counting the number of matrix elements without taking into account the complexity of computing each element (see below), which is estimated by

$$(600 \text{ secs}) \times \left(\frac{\text{bond dimension}}{10000}\right)^2 \times (\text{local Hilbert space dimension}), \tag{S37}$$

where 600 secs is to use our simulation time as a benchmark, which used $\chi = 10^4$ bond dimension. We note that the time scale can vary for different computing devices or be significantly reduced by employing more resources. Finally, we obtain the largest matrix size for which we have to compute the hafnian to construct an MPS, which is equal to $2l$ in Eq. (S27). This corresponds to the cost of computing each element of MPS that was not taken into account in the time scale. In fact, as presented in the figure, for the relevant regimes for the current experiments, the matrix size is not too large to compute in a reasonable time. Also, Ref. [31] shows that computing the hafnian of matrix size less than 50×50 is still possible with a reasonable computational cost. Note that although we do not present the matrix size estimate in Fig. S2 because of numerical uncertainty, based on the worst-case circuit results, we expect that the matrix size is not an immediate obstacle for the current experimental parameters.

S7. BAYESIAN TEST

As mentioned in the main text, Refs. [5, 6, 35] employ the Bayesian test as a benchmark. In this Supplemental Material, we show that the test is not suitable for our purpose and our sampler because our sampler is closer to the ground-truth distribution than the experimental sampler is but our sampler is far from the experimental sampler. To see this, we compute the TVD of all the pairs between the ideal, MPS, and experimental samplers, as illustrated in Fig. S3(a). The figure clearly shows that while the MPS sampler is a good approximation of the ground-truth distribution, it still has a large distance to the experimental sampler. Therefore, as shown in the figure, even though our sampler approximates the ideal distribution better than the experiments in TVD, it fails to pass the Bayesian test, where the Bayesian score is defined as

$$\text{score} \equiv \frac{1}{N_s} \sum_{j=1}^{N_s} \log \frac{\Pr^{(G)}(\boldsymbol{m}_j) \Pr^{(s)}(N)}{\Pr^{(s)}(\boldsymbol{m}_j) \Pr^{(G)}(N)}, \tag{S38}$$

where G stands for the ground truth and $\Pr^{(i)}(\boldsymbol{m})$ is the output probability of obtaining an outcome $\boldsymbol{m}$ a sampler i, which is either ideal or a mock-up sampler, and $\Pr^{(i)}(N)$ is the probability of obtaining total N photons. Thus, a positive score means that the ground-truth distribution is a better approximation than the mock-up distribution. Consequently, the failure of our sampler to pass the Bayesian test does not imply that our sampler is not a good approximation of the ground-truth distribution.

S8. INTERMEDIATE-SCALE EXPERIMENTS

In this Supplemental Material, we display an additional plot for intermediate cases: Borealis's $M = 216$ (low) case and Jiuzhang2.0's $M = 144$ with P65-1 and P65-2. For each case we choose $\chi = 600, 1000, 2000$ with $d = 4$ ($d = 6$ is used for sampling by patching the tensors.), respectively. We numerically demonstrate that the bond dimensions are sufficient to achieve better scores than experiments in XEB and two-point correlation benchmarking, as exhibited in Fig. S4. Also, each case entails an error of $0.0114, 0.0078, 0.0162$, respectively. For all cases with the chosen bond dimensions, our MPS simulator achieves comparable XEB scores and two-point correlation functions.

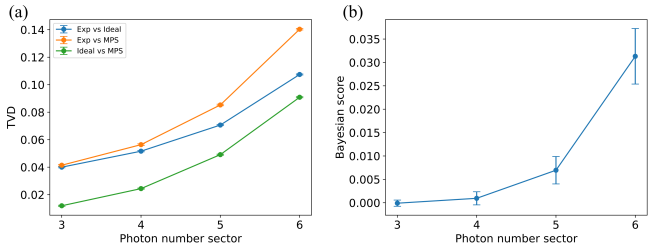

FIG. S3. (a) TVD between the experimental sampler, MPS sampler ($\chi = 20$), and the ideal sampler. (b) The Bayesian score of the MPS sampler. The error bars are the standard deviation obtained by 1,000 bootstrapping resamples.

S9. GROUND-TRUTH DISTRIBUTION OF JIUZHANG3.0

We note that Ref. [35] used a slightly different ground-truth distribution from the one we used for benchmarking. More specifically, Jiuzhang3.0 applied 8-fold local beam splitters to each output mode for implementing pseudo-photon-number-resolving detection (PPNRD), which makes $M = 144$ output modes into $M = 1152$ modes. In our simulation we treat the experiment as $M = 144$ modes instead of $M = 1152$ because the purpose of additional linear-optical circuits is to implement PPNRD and they are local beam splitters. In other words, after generating samples by photon numbers, we transform the photon numbers to photon clicks for both the experiment and our sampler to compute the correlation functions of photon clicks for 144 modes. We consider threshold detectors instead of PNRDs because the photon number for each output port is not dilute enough to mimic the true PNRD in experiment, especially for the high case.

Also, we remark that Ref. [35] modeled the ground-truth distribution as a lossy and partial distinguishable GBS, additionally incorporating the partial distinguishability noise into the ground truth, while we still employ the lossy GBS as the ground truth. Hence, for our analysis, the partial distinguishability causes a deviation between the experimental and ground-truth samplers, which allows the MPS to have a larger truncation error. We expect that if we include additional noises into the ground-truth distribution, the truncation error of MPS has to decrease further; that is, the bond dimension has to be chosen larger than the present simulation because the ground-truth distribution becomes closer to the experimental sampler. However, the partial distinguishable GBS model used in Ref. [35] can be easily implemented by constructing two independent MPSs for two distinguishable input sources. In addition, since a single-mode squeezed state is split into two smaller squeezed states, the cost of each MPS is smaller than the indistinguishable GBS model.

[1] N. Quesada, R. S. Chadwick, B. A. Bell, J. M. Arrazola, T. Vincent, H. Qi, R. García, *et al.*, Quadratic speed-up for simulating Gaussian boson sampling, PRX Quantum **3**, 010306 (2022).

[2] N. Quesada and J. M. Arrazola, Exact simulation of Gaussian boson sampling in polynomial space and exponential time, Physical Review Research **2**, 023005 (2020).

[3] J. F. Bulmer, B. A. Bell, R. S. Chadwick, A. E. Jones, D. Moise, A. Rigazzi, J. Thorbecke, U.-U. Haus, T. Van Vaerenbergh, R. B. Patel, *et al.*, The boundary for quantum advantage in Gaussian boson sampling, Science Advances **8**, eabl9236 (2022).

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

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

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

[7] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (2011) pp. 333–342.

[8] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Gaussian boson sampling, Physical Review Letters **119**, 170501 (2017).

[9] 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, Science Advances **8**, eabi7894 (2022).

[10] R. García-Patrón, J. J. Renema, and V. Shchesnovich, Simulating boson sampling in lossy architectures, Quantum **3**, 169 (2019).

[11] H. Qi, D. J. Brod, N. Quesada, and R. García-Patrón, Regimes of classical simulability for noisy Gaussian boson sampling,

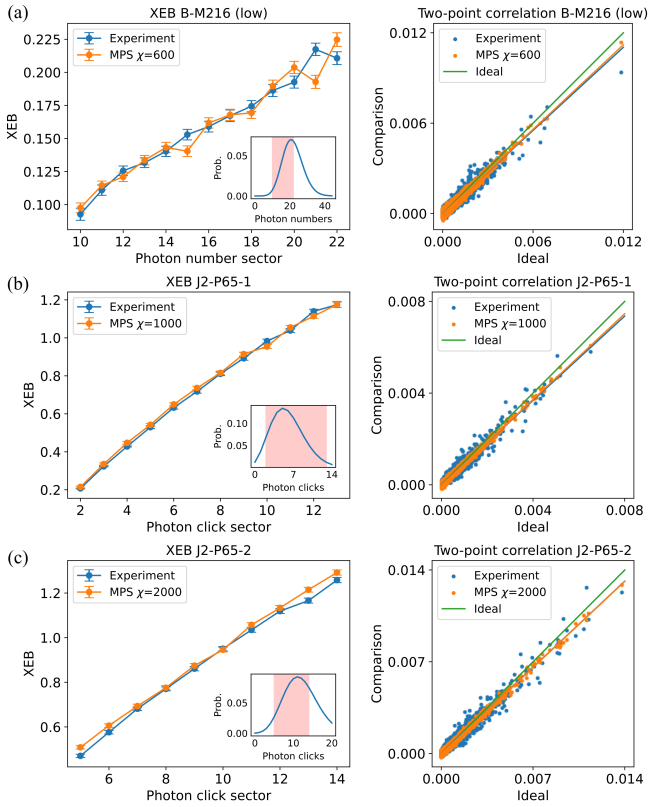

FIG. S4. XEB and two-point correlation function for (a) Borealis $M = 216$ (low), (b) Jiuzhang2.0's P65-1 with $M = 144$, and (c) Jiuzhang2.0's P65-2 with $M = 144$. The error bars are the standard deviation obtained by 1,000 bootstrapping resamples.

Physical Review Letters **124**, 100502 (2020).

[12] J. Martínez-Cifuentes, K. Fonseca-Romero, and N. Quesada, Classical models may be a better explanation of the jiuzhang 1.0 gaussian boson sampler than its targeted squeezed light model, Quantum **7**, 1076 (2023).

[13] M. Oszmaniec and D. J. Brod, Classical simulation of photonic linear optics with lost particles, New Journal of Physics **20**, 092002 (2018).

[14] H.-L. Huang, W.-S. Bao, and C. Guo, Simulating the dynamics of single photons in boson sampling devices with matrix product states, Physical Review A **100**, 032305 (2019).

[15] C. Oh, K. Noh, B. Fefferman, and L. Jiang, Classical simulation of lossy boson sampling using matrix product operators, Physical Review A **104**, 022407 (2021).

[16] M. Liu, C. Oh, J. Liu, L. Jiang, and Y. Alexeev, Simulating lossy gaussian boson sampling with matrix-product operators, Physical Review A **108**, 052604 (2023).

[17] G. Kalai and G. Kindler, Gaussian noise sensitivity and bosonsampling, arXiv preprint arXiv:1409.3093 (2014).

[18] J. J. Renema, A. Menssen, W. R. Clements, G. Triginer, W. S. Kolthammer, and I. A. Walmsley, Efficient classical algorithm for boson sampling with partially distinguishable photons, Physical Review Letters **120**, 220502 (2018).

[19] J. Renema, V. Shchesnovich, and R. Garcia-Patron, Classical simulability of noisy boson sampling, arXiv preprint arXiv:1809.01953 (2018).

[20] V. S. Shchesnovich, Noise in boson sampling and the threshold of efficient classical simulatability, Physical Review A **100**, 012340 (2019).

[21] A. E. Moylett, R. García-Patrón, J. J. Renema, and P. S. Turner, Classically simulating near-term partially-distinguishable and lossy boson sampling, Quantum Science and Technology **5**, 015001 (2019).

[22] 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 preprint arXiv:2109.11525 (2021).

[23] D. Aharonov, X. Gao, Z. Landau, Y. Liu, and U. Vazirani, A polynomial-time classical algorithm for noisy random circuit sampling, arXiv preprint arXiv:2211.03999 (2022).

[24] C. Oh, L. Jiang, and B. Fefferman, On classical simulation algorithms for noisy boson sampling, arXiv preprint arXiv:2301.11532 (2023).

[25] C. Oh, L. Jiang, and B. Fefferman, Spoofing cross-entropy measure in boson sampling, Physical Review Letters **131**, 010401 (2023).

[26] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Phys. Rev. Lett. **91**, 147902 (2003).

[27] A. Serafini, *Quantum continuous variables: a primer of theoretical methods* (CRC press, 2017).

[28] N. Quesada, Franck–Condon factors by counting perfect matchings of graphs with loops, The Journal of Chemical Physics **150**, 164113 (2019).
[29] C. Oh, Y. Lim, Y. Wong, B. Fefferman, and L. Jiang, Quantum-inspired classical algorithms for molecular vibronic spectra, Nature Physics **20**, 225 (2024).
[30] A. Björklund, Counting perfect matchings as fast as Ryser, in *Proceedings of the twenty-third annual acm-siam symposium on discrete algorithms* (SIAM, 2012) pp. 914–921.
[31] A. Björklund, B. Gupt, and N. Quesada, A faster hafnian formula for complex matrices and its benchmarking on a supercomputer, Journal of Experimental Algorithmics (JEA) **24**, 1 (2019).
[32] D. P. Dubhashi and A. Panconesi, *Concentration of measure for the analysis of randomized algorithms* (Cambridge University Press, 2009).
[33] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, Strawberry fields: A software platform for photonic quantum computing, Quantum **3**, 129 (2019).
[34] B. Gupt, J. Izaac, and N. Quesada, The Walrus: A library for the calculation of hafnians, Hermite polynomials and Gaussian boson sampling, Journal of Open Source Software **4**, 1705 (2019).
[35] Y.-H. Deng, Y.-C. Gu, H.-L. Liu, S.-Q. Gong, H. Su, Z.-J. Zhang, H.-Y. Tang, M.-H. Jia, J.-M. Xu, M.-C. Chen, *et al.*, Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage, Phys. Rev. Lett. **131**, 150601 (2023).