PHYSICAL REVIEW A **96**, 022301 (2017)

Exact boson sampling using Gaussian continuous-variable measurements

A. P. Lund,[1] S. Rahimi-Keshari,[1,2] and T. C. Ralph[1]
[1]*Centre for Quantum Computation and Communications Technology, School of Mathematics and Physics, The University of Queensland, St. Lucia, Queensland 4072, Australia*
[2]*School of Physics, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5531, Tehran, Iran*
(Received 29 May 2017; published 2 August 2017)

Boson sampling is a quantum mechanical task involving Fock basis state preparation and detection and evolution using only linear interactions. A classical algorithm for producing samples from this quantum task cannot be efficient unless the polynomial hierarchy of complexity classes collapses, a situation believed to be highly implausible. We present a method for constructing a device which uses Fock state preparations, linear interactions, and Gaussian continuous-variable measurements for which one can show that exact sampling would be hard for a classical algorithm in the same way as boson sampling. The detection events used from this arrangement do not allow a similar conclusion to be drawn for the classical hardness of approximate sampling. We discuss the details of this result outlining some specific properties required by approximate sampling hardness.

DOI: 10.1103/PhysRevA.96.022301

I. INTRODUCTION

Boson sampling is the task of producing statistical samples from Fock basis measurements of a bosonic M-mode linear scattering network with an input consisting of N modes prepared with a single boson and the remaining $M-N$ modes prepared in the vacuum state. This task, while not universal for quantum computing, has been shown to not be efficiently computable by any classical algorithm (or the polynomial hierarchy of complexity classes collapses, which is believed to be extremely unlikely) [1]. However, a quantum implementation is efficient as one merely needs to build the scattering device as described within a sufficiently small error budget. This is many orders of magnitude easier than the construction of a fully universal quantum computer, but is still a challenge for current technology.

Attempts have been made to identify scenarios where the proof of classical hardness of boson sampling can be used or adapted to other sampling problems. One particular scenario that is of experimental interest is in the use of continuous-variable (CV) Gaussian states or measurements. For another restricted computational model based on sampling from qubit circuits involving commuting coherent rotations, it has been shown that CV variants are hard to simulate classically [2]. It is known that for linear networks with Gaussian state inputs and Gaussian measurements it is efficient for classical algorithms to not only produce samples but also compute the entire output distribution [3]. Nevertheless it has been shown that a hybrid approach which involves linear networks with input two-mode squeezed vacuum states and Fock basis detection has a similar classical hardness proof to the original boson sampling problem [4]. There is also evidence for the classical hardness of a more general construction involving squeezed-vacuum inputs, linear optics, and Fock basis detection [5,6]. The question this paper addresses is the reverse situation: Fock state inputs to linear networks and homodyne detection.

An important aspect of the hardness proof for boson sampling is that the output probability distribution contains probabilities which are proportional to matrix permanents from submatrices of the matrix describing the linear scattering network. A matrix permanent is a quantity which is computed like a matrix determinant without the alternating addition and subtraction. In fact, when sampling with Fock state inputs and detection, all detection probabilities are proportional to submatrices derived from the linear scattering network [7]. This special situation, given that two plausible conjectures hold, allows the hardness proof of "approximate" sampling to be shown [1]. This is because an allowed error budget's effect can be spread over all detection events provided the linear network appears randomly distributed.

Here we show that using single-photon input states and a particular Gaussian measurement one can extract submatrix permanents to within an exponentially small error. Therefore one can show that exact sampling from this distribution is hard. It is not necessarily the case that approximate sampling is still hard and we discuss this in relation to our construction.

In Sec. II we present some of the background behind the hardness arguments for boson sampling. Then in Sec. III we present the CV-n detector model using a Fock state $|n\rangle$ input with CV measurements. In Sec. IV we will then describe exact sampling using the CV-1 model and the technical details involved in showing the hardness of computing samples from the CV output distribution. Finally we will discuss the issues preventing the hardness result from being used in this model to make definitive conclusions about the hardness of approximate sampling.

II. CLASSICAL HARDNESS OF BOSON SAMPLING

A problem in the class boson sampling is one where the statistical samples can be generated by an M mode linear interaction between N singly occupied bosonic modes (and $M-N$ bosonic vacua) which is subsequently detected in the Fock basis. Boson sampling is either inefficient using classical computational resources (not in P), or the "polynomial hierarchy" of complexity classes collapses to the third level, a situation believe to be implausible. It is not our goal to present in full the background and subsequent arguments toward the truth of this statement because this has been done elsewhere [1]. We will, however, outline some of the key aspects used in

this paper that are needed to understand what is required for the proof presented in [1] to hold.

A. Polynomial hierarchy

The polynomial hierarchy of complexity classes is a nested structure defined by the use of oracles. An oracle is essentially an assumption on the resources available to an algorithm which can greatly assist in proving statements in computational complexity. The hierarchy has a complex definition and we will concentrate on a simplified version.

The class NP is the set of decision (yes/no) problems whose satisfying input can be verified efficiently. This defines the first level of the polynomial hierarchy. The second level is then the class NP with access to an oracle from the first level. Subsequent levels are defined by continuing this recursion, e.g., the third level is the class NP with access to an oracle from the second level.

This structure has a strong connection to similarly defined hierarchies within number theory and set theory. In those cases each level of the hierarchy is strictly larger than lower levels. If two levels were to coincide, then the addition of levels stops growing and the hierarchy is said to collapse. In terms of the computational complexity structure, a collapse of the hierarchy to the first level means that $\mathsf{P} = \mathsf{NP}$ or that NP problems can be efficiently solved deterministically, a situation believed to be highly implausible. A collapse to the second level would mean $\mathsf{P}^{\mathsf{NP}} = \mathsf{NP}^{\mathsf{NP}}$ which is the same statement relative to an NP oracle. Being relative to the oracle means that the statement is slightly more plausible, but it is still believed to be not possible. The prevailing belief is that the polynomial hierarchy does not collapse to any level and this is the assumption on which the hardness of boson sampling can be proven.

B. Stockmeyer's approximate counting algorithm

Critical to the hardness proof of boson sampling is the use of Stockmeyer's approximate counting algorithm [8]. This algorithm computes estimates of a quantity defined as

$$F = \sum_{x \in Q} f(x), \tag{1}$$

where $f : Q \to \{0,1\}$ is a boolean function from length l bit strings $Q = \{0,1\}^l$ onto a single bit. In other words F is the number of inputs that result in an output of 1, a set we will call Q_1. The computed estimate of this quantity is multiplicative, which means the estimate of F that the algorithm produces is $\tilde{F}$ satisfying

$$F g^{-1} \leqslant \tilde{F} \leqslant F g, \tag{2}$$

where $g > 1$ and for Stockmeyer's algorithm is lower bounded by $1 + 1/poly(l)$.

Stockmeyer's algorithm computes the estimate $\hat{F}$ by finding the smallest output with no collisions for a randomly chosen hash function on Q_1. That is, choose randomly a function $h : Q_1 \to \{0,1\}^p$ (for $p \leqslant l$) and if there exist no elements $a,b \in Q_1$ such that $h(a) = h(b)$ (a hash collision) then we have made an upper-bounded estimate on the size of Q_1 of 2^p.

Here we can see the elements which make up this algorithm: the estimate of size is multiplicative, it requires finding hash collisions, which is an NP problem, and involves random choices. This results in Stockmeyer's algorithm being contained within a class $\mathsf{FBPP}^{\mathsf{NP}}$ (if the function f is efficiently computable). The superscript notation describes access to an NP oracle, which in the case of Stockmeyer's algorithm is used to find hash collisions. BPP means the algorithm proceeds by making random (probabilistic) choices with a probability of success at least 2/3. The prefix F is used to describe the output from this algorithm, which is a number (function) rather than a decision or yes/no output. This is therefore not an algorithm one would expect to be efficiently computable when realistically implemented. But it is used here to establish if a problem lives within the "polynomial" hierarchy of complexity classes. In this case, this algorithm lives within the third level as $\mathsf{BPP} \subset \mathsf{NP}^{\mathsf{NP}}$ [9,10].

This algorithm is used to make multiplicative estimations of the underlying probabilities of a distribution from samples of the distribution. In the model of classical computation, unlike quantum mechanical models, there is no inherent randomness. Randomness is introduced by external means and can be regarded as an input to the algorithm. If the function $f(x)$ above represents a sampling algorithm for a two-outcome distribution, then it can be thought of as converting input random bits x distributed uniformly into samples of the desired probability distribution. One can then estimate the probability of the outcome 1 by counting how many inputs produce the outcome 1 relative to the total number of possible inputs. This would mean dividing the estimate $\hat{F}$ by 2^l which will also produce a multiplicative estimate even though it has been divided by an exponentially increasing factor.

A polynomial hierarchy collapse is triggered if f is efficient to compute and the probability it produces samples from is a quantity for which multiplicative estimation is known to be outside the third level of the polynomial hierarchy.

C. Multiplicative and additive errors

Producing approximations within multiplicative errors is particularly powerful. It is more natural to consider the case of additive errors. An additive estimate of F would be one satisfying

$$F - g \leqslant \tilde{F} \leqslant F + g. \tag{3}$$

This kind of estimation will arise from more natural models of errors within a computation. For example, the function f above most likely admits errors which are of an additive nature and hence the estimations performed using this noisy model will also be additive. Only in the case of admitting exactly zero noise to f, or in the sampling case a situation called "exact sampling", can one generate a multiplicative estimation.

The crucial outcome of [1] is that the polynomial hierarchy collapse can be shown to occur in boson sampling even with a given level of total variation distance between the ideal and actual distributions. The total variation distance is an additive quantity and will generate additive errors in estimates. But as the probabilities in boson sampling tend to decrease to zero exponentially, the introduction of additive errors will overwhelm the magnitude of the quantity being estimated. The trick of [1] is to use the structure of the probabilities in boson sampling to encode the estimated quantity within

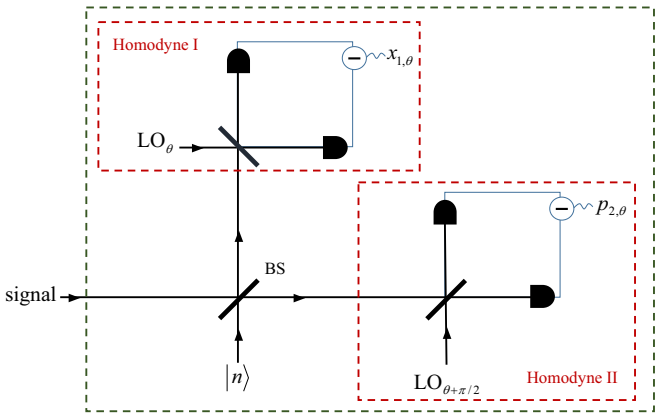

FIG. 1. CV-n measurement device. The outer green dashed box encloses the whole CV-n device, which measures the input signal in the displaced Fock state basis. The inner red boxes represent two homodyne detections performed simultaneously that give two CV outcomes $x_{1,\theta}$ and $p_{2,\theta}$ from the measurement device.

an exponentially large set of possible outcomes in a way that without knowledge of where the problem is encoded, looks like a Haar-random unitary matrix (i.e., the choice is hidden from the implementation). This means that the additive estimate generated will have bounds determined by the average case which turns out to be sufficient to additively estimate matrix permanents. Using this estimate it is then possible to trigger a polynomial hierarchy collapse to the third level, given that two plausible conjectures about the nature of estimating permanents of Gaussian random matrices hold true.

III. DETECTOR

We will now describe our continuous-variable detection model and how we use it to construct a probability distribution which can be used in the arguments of [1]. The model is based on a measurement device for measuring in the displaced number state basis, which we refer to it as CV-n measurement, and two variations of this measurement, phase-randomized CV-n (PRCV-n) and discretized-phase-randomized CV-n (DPRCV-n). Only a brief outline of the model is presented here. The details of this calculation are given in the Appendix.

As depicted in Fig. 1, the measurement device works as follows. The input signal is overlapped on a 50:50 beamsplitter with a number state $|n\rangle$, and the outputs of the beamsplitter are measured by two conjugate homodyne detections whose local oscillators have $\pi/2$ phase difference. As shown in the Appendix, the positive operator-valued measure (POVM) elements of this measurement are

$$\Pi^n(x_{1,\theta}, p_{2,\theta}) = \frac{1}{2\pi} D(\alpha)\,|n\rangle\langle n|\,D^\dagger(\alpha). \tag{4}$$

The two real numbers $x_{1,\theta}$ and $p_{2,\theta}$ form the results of a simultaneous measurement of two orthogonal quadratures. Notice that for $n = 0$, CV-0, we have heterodyne measurement.

Phase-randomized CV-n (PRCV-n). If the phase θ of the local oscillators is randomized, while the relative phase is fixed, we have PRCV-n measurement. As discussed in the Appendix, the outcome of this measurement is a single non-negative parameter $R = r^2 = x_{1,\theta}^2 + p_{2,\theta}^2$ with the corresponding POVM element

$$\Pi^n(R) = n!e^{-R}\sum_{k=0}^{\infty}\frac{R^{k-n}}{k!}\left[L_n^{k-n}(R)\right]^2|k\rangle\langle k|\,, \tag{5}$$

with $L_k^m(.)$ being the generalized Laguerre polynomials. Though this measurement has less information, we will concentrate on it since we are only interested in events where $R \approx 0$ and having the POVM diagonal in the Fock basis greatly simplifies many calculations.

Discretized-phase-randomized CV-1 (DPRCV-1). The measurement model outputs continuous variables (i.e., R) and hence these POVMs are describing probability *densities*. To be able to use the methods of [1] to make some definitive statement about computational hardness, one must work in probabilities and not probability densities. Hence, we will utilize the discretized version of the phase-randomized CV-1 (DPRCV-1) measurement through its ability to distinguish between the single-photon state and other number states. We divide the range of $0 \leqslant R \leqslant \infty$ for the POVM elements in Eq. (5) for $n = 1$, to two parts: $t = \{R|0 \leqslant R \leqslant t\}$ which represents the smallest discrete region and $\bar{t} = \{R|t \leqslant R \leqslant \infty\}$ which represents all the others. Thus, we have two POVM elements for each interval:

$$\begin{aligned}\Pi_t^1 &= \sum_{k=0}^{\infty}\frac{1}{k!}\int_0^t dR\, e^{-R}R^{k-1}(k-R)^2\,|k\rangle\langle k| \\ &= \sum_{k=0}^{\infty}G(t,k)\,|k\rangle\langle k|\,,\end{aligned} \tag{6}$$

where

$$G(t,k) = \frac{1}{k!}[k^2\gamma(k,t) - 2k\gamma(1+k,t) + \gamma(2+k,t)], \tag{7}$$

with $\gamma(k,t) = \int_0^t dR\, e^{-R}R^{k-1}$ being the lower incomplete Γ function. As this is a two-outcome measurement and each of the elements is bounded above by 1 the POVM for all other outcomes can then be written

$$\Pi_{\bar{t}}^1 = I - \Pi_t^1. \tag{8}$$

The POVM element in Eq. (6) can be expanded in a power series as

$$\begin{aligned}\Pi_t^1 = &\left(\frac{t^2}{2} - \frac{t^3}{3} + O(t^4)\right)|0\rangle\langle 0| \\ &+ \left(t - \frac{3t^2}{2} + \frac{7t^3}{6} + O(t^4)\right)|1\rangle\langle 1| \\ &+ \left(t^2 - \frac{4t^3}{3} + O(t^4)\right)|2\rangle\langle 2| \\ &+ \left(\frac{1}{2}t^3 + O(t^4)\right)|3\rangle\langle 3| + O(t^4)\,|4\rangle\langle 4| + \cdots.\end{aligned} \tag{9}$$

Therefore, if t is small $t = \epsilon$ such that $\epsilon^2 \approx 0$, the POVM element associated with detecting $R \in [0,\epsilon]$ is

$$\Pi_\epsilon^1 = \epsilon\,|1\rangle\langle 1|\,. \tag{10}$$

It is this form of the POVM which allows a measurement that distinguishes a single-photon Fock state from the remainder of the Hilbert space.

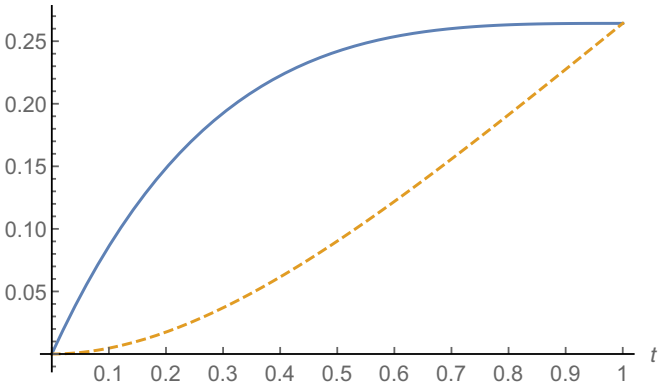

FIG. 2. Comparison between $\eta(t)$ (solid line) and $p_D(t)$ (dashed line) for the CV-1 detector.

To demonstrate what this measurement is detecting consider the case of an input state being either $|0\rangle$ or $|1\rangle$. Then when detecting the single-photon state (and for any value of t),

$$\eta(t) = \mathrm{Tr}\big[\,|1\rangle\langle 1|\,\Pi_t^1\big] = 1 - e^{-t}(1+t^2) \tag{11}$$

can be thought of as the efficiency of the detector. Also

$$p_D(t) = \mathrm{Tr}\big[\,|0\rangle\langle 0|\,\Pi_t^1\big] = 1 - e^{-t}(1+t) \tag{12}$$

can be thought of as the dark count probability. Figure 2 compares these two quantities.

IV. BOSON SAMPLING

We now consider the problem of sampling from the output probability distribution of a linear-optical network (LON) using CV-1, PRCV-1, and DPRCV-1 measurements (where the ancillary Fock state is $n = 1$) introduced in the previous section. We first derive the output probability distribution for each measurement scheme, and then discuss the complexity of sampling from the probability distribution.

A. Probability distributions

In this setup, the input state to the LON is

$$|1_N\rangle = |\underbrace{1,1,\cdots,1}_{N},\underbrace{0,0,\cdots,0}_{M-N}\rangle\,. \tag{13}$$

Each output mode is measured by a CV-1 measurement; hence, the overall POVM elements are

$$\Pi^1(\boldsymbol{\alpha}) = \frac{1}{(2\pi)^M} D(\boldsymbol{\alpha})\,|1_M\rangle\langle 1_M|\,D^\dagger(\boldsymbol{\alpha}), \tag{14}$$

where $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\ldots,\alpha_M)$. As shown in Fig. 3, this setup is equivalent to injecting single photons into a larger network and performing homodyne measurements at the output, i.e., boson sampling with homodyne measurements.

Therefore, the output probability density is given by

$$\begin{aligned} P(\boldsymbol{\alpha}) &= \frac{1}{(2\pi)^M}\,|\,\langle 1_M|\,D^\dagger(\boldsymbol{\alpha})\mathcal{U}_{\mathrm{LON}}\,|1_N\rangle\,|^2 \\ &= \frac{1}{(2\pi)^M}\left|\sum_{\boldsymbol{n}}\langle 1_M|\,D^\dagger(\boldsymbol{\alpha})\,|\boldsymbol{n}\rangle\,\langle\boldsymbol{n}|\,\mathcal{U}_{\mathrm{LON}}\,|1_N\rangle\right|^2. \end{aligned} \tag{15}$$

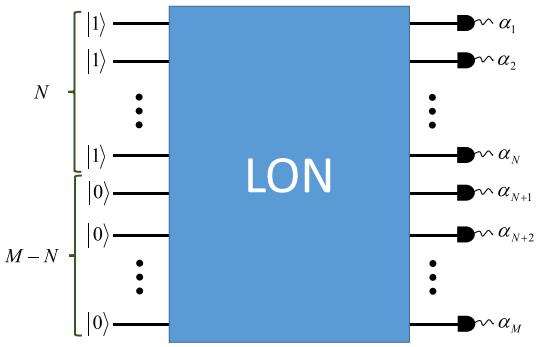

FIG. 3. Boson sampling using CV-1 measurements.

Here $\mathcal{U}_{\mathrm{LON}}$ is the unitary operation associated with the LON, and $|\boldsymbol{n}\rangle = |n_1,n_2,\ldots,n_M\rangle$ is the multimode Fock state. As a LON preserves the number of photons, the sum is restricted to $\boldsymbol{n}$'s which satisfy $\sum_{i=1}^M n_i = N$.

We have [7]

$$\langle\boldsymbol{n}|\,\mathcal{U}_{\mathrm{LON}}\,|1_N\rangle = \mathrm{Per}\big(U_{1_N\times\boldsymbol{n}}\big), \tag{16}$$

which is the permanent of an $N\times N$ submatrix of the unitary matrix of LON, $U_{1_N\times\boldsymbol{n}}$ that is corresponding to the first N rows and columns with multiplicity of n_i. The unitary matrix U is defined through the relation

$$b_j^\dagger = \mathcal{U}_{\mathrm{LON}} a_j^\dagger\,\mathcal{U}_{\mathrm{LON}}^\dagger = \sum_{i=1}^M U_{ij} a_i^\dagger, \tag{17}$$

where $b_j^\dagger$ and $a_i^\dagger$ are modal creation operators for output j and input i, respectively. Also, using the expression for $\langle k|\,D(\alpha)\,|m\rangle$ in [11], one can simply verify that

$$\langle 1_M|\,D^\dagger(\boldsymbol{\alpha})\,|\boldsymbol{n}\rangle = \prod_{j=1}^M e^{-|\alpha_j|^2/2}\frac{(\alpha_j^*)^{n_j-1}}{\sqrt{n_j!}}(n_j - |\alpha_j|^2).$$

By using these relations, the output probability density (15) becomes

$$P(\boldsymbol{\alpha}) = \frac{e^{-\boldsymbol{\alpha}\boldsymbol{\alpha}^\dagger}}{(2\pi)^M}\left|\sum_{\boldsymbol{n}}\mathrm{Per}\big(U_{1_N\times\boldsymbol{n}}\big)\prod_{j=1}^M\frac{(\alpha_j^*)^{n_j-1}}{\sqrt{n_j!}}(n_j - |\alpha_j|^2)\right|^2. \tag{18}$$

From this expression we can see that if $\alpha_j = 0$, the probability density $P(\boldsymbol{\alpha})$ is zero unless $n_j = 1$.

We define $\boldsymbol{n}_N = (n_{N,1},n_{N,2},\ldots,n_{N,M})$ as a M-tuple whose elements are either zero or one and the number of ones is N, and define $\boldsymbol{\alpha}_{\boldsymbol{n}_N}$ a vector of length M whose element are given by $[\boldsymbol{\alpha}_{\boldsymbol{n}_N}]_j = (1-n_{N,j})\alpha_j$, i.e., it has N zero elements corresponding to $n_{N,j} = 1$. Using these notations, we can see that

$$P(\boldsymbol{\alpha}_{\boldsymbol{n}_N}) = \frac{1}{(2\pi)^M}\big|\mathrm{Per}\big(U_{1_N\times\boldsymbol{n}_N}\big)\big|^2\prod_{n_{N,j}=0}e^{-|\alpha_j|^2}|\alpha_j|^2, \tag{19}$$

where the product is over $M-N$ nonzero elements of $\boldsymbol{\alpha}_{\boldsymbol{n}_N}$. This implies that if N of M α's are zero, the probability density $P(\boldsymbol{\alpha})$ is proportional to the absolute value square of permanent

of a submatrix of U. For example, for $\boldsymbol{n}_N = 1_N$

$$P(\boldsymbol{\alpha}_{1_N}) = P(0,\boldsymbol{\alpha}_{M-N}) = \frac{1}{(2\pi)^M}\left|\mathrm{Per}\left(U_{1_N\times 1_N}\right)\right|^2 \prod_{j=N+1}^{M} e^{-|\alpha_j|^2}|\alpha_j|^2. \quad (20)$$

If we use PRCV-1 measurements at the output of LON, using the POVM in Eq. (A36), the output probability density is given by

$$P_{\mathrm{P}}(\boldsymbol{R}) = e^{-|\boldsymbol{R}|}\sum_{\boldsymbol{n}}\left|\mathrm{Per}\left(U_{1_N\times\boldsymbol{n}}\right)\right|^2 \prod_{j=1}^{M}\frac{1}{n_j!}R_j^{n_j-1}(n_j-R_j)^2, \quad (21)$$

where $\boldsymbol{R} = (R_1, R_2, \ldots, R_M)$ and $|\boldsymbol{R}| = R_1 + R_2 + \cdots + R_M$. Defining $\boldsymbol{R}_{\boldsymbol{n}_N}$ similar to $\boldsymbol{\alpha}_{\boldsymbol{n}_N}$, we have

$$P_{\mathrm{P}}(\boldsymbol{R}_{\boldsymbol{n}_N}) = \left|\mathrm{Per}\left(U_{1_N\times\boldsymbol{n}_N}\right)\right|^2 \times \prod_{n_{Nj}=0}\frac{1}{n_j!}e^{-R_j}R_j^{n_j-1}(n_j-R_j)^2. \quad (22)$$

For DPRCV-1 measurements at the output, the probability of detecting outcome $\boldsymbol{m}_N$, that is, N clicks within interval t and $M-N$ clicks within $\bar{t}$, using Eq. (7), is given by

$$P_{\mathrm{D}}(\boldsymbol{m}_N) = \sum_{\boldsymbol{n}}\left|\mathrm{Per}(U_{1_N\times\boldsymbol{n}})\right|^2 \times \prod_{m_{Nj}=1} G(t,n_j)\prod_{m_{Nj}=0}[1-G(t,n_j)]. \quad (23)$$

If t is very small, this probability can be expanded to leading order in a powers of t as

$$P_{\mathrm{D}}(\boldsymbol{m}_N) = \left|\mathrm{Per}\left(U_{1_N\times\boldsymbol{m}_N}\right)\right|^2[t^N + O(t^{N+1})] + \sum_{\boldsymbol{m}=\boldsymbol{m}_N^1}|\mathrm{Per}\left(U_{1_N\times\boldsymbol{m}}\right)|^2 O(t^{N+2}), \quad (24)$$

where $\boldsymbol{m}_N^1$ are M-tuples whose elements are equal to $\boldsymbol{m}_N$ except one 0 and one 1 are interchanged.

Notice that the probability distribution (23) corresponding to DPRCV-1 measurements is a coarse-grained version of probability density (21), and that itself is a phase-randomized version of Eq. (18). Therefore, if exact sampling from the probability distribution (23) is classically hard, exact sampling from the other probability distributions must be classically hard as well. In the following two sections we discuss exact and approximate sampling from the probability distribution (23).

B. Exact boson sampling

To leading order in t, Eq. (24) gives a distribution for which hardness of exact sampling can be determined, as we shall show in this section. Merely setting the $O(t^{N+1})$ terms to zero results in a probability

$$P_{\mathrm{D}}(1_N) = |\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2 t^N. \quad (25)$$

A probability of this form does not pose any problems for the argument of hardness in exact sampling. However, it is important to consider higher-order terms because exact sampling allows sampling using any matrix even those with vanishingly small permanents. For example, one could consider the case where the permanent $|\mathrm{Per}(U_{1_N\times 1_N})|^2 = O(t^2)$ but the unwanted higher-order terms have $|\mathrm{Per}(U_{1_N\times\boldsymbol{n}})|^2 = O(1)$. Furthermore, as $|\mathrm{Per}(U_{1_N\times\boldsymbol{n}})|^2$ defines a probability distribution over all $\boldsymbol{n}$ (i.e., the boson sampling distribution) we have

$$\sum_{\boldsymbol{m}=\boldsymbol{m}_N^1}|\mathrm{Per}\left(U_{1_N\times\boldsymbol{m}}\right)|^2 \leqslant 1. \quad (26)$$

This means that, using Eq. (24), the probability can be written as

$$P_{\mathrm{D}}(1_N) = t^N\left[(1-O(t))\left|\mathrm{Per}\left(U_{1_N\times 1_N}\right)\right|^2 + E\right], \quad (27)$$

where $E = O(t^2)$ is an "error" over the desired probability. Unfortunately E contributes to the probability in an additive sense. However, as we have used the upper bound on the matrix permanents that contribute to E we know that this bound is independent of N. So if we choose E to be a small constant, we need to quadratically vary t to achieve that constant. Furthermore, with constant E, the sizes of submatrix permanents that can be estimated must be lower bounded with a bound that depends on E.

To make this more explicit consider the case where $0 < L \leqslant |\mathrm{Per}(U_{1_N\times 1_N})|^2 \leqslant 1$. If we used the approximate counting algorithm using $P_{\mathrm{D}}(1_N)$ from Eq. (27) then we would have an estimate $\tilde{p}$ which satisfies

$$\left[|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2 + E\right]g^{-1} < \tilde{p} < \left[|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2 + E\right]g, \quad (28)$$

where $g > 1$ is the multiplicative error factor from the estimation. We can also write

$$\left[|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2 - |E|\right]g^{-1} < \tilde{p} < \left[|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2 + |E|\right]g, \quad (29)$$

and using $|\mathrm{Per}(U_{1_N\times 1_N})|^2 > L$

$$|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2(1-|E|/L)g^{-1} < \tilde{p} < |\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2(1+|E|/L)g. \quad (30)$$

Now if $|E|/L < 1/2$ (arbitrarily, it could be any constant less than 1), we have

$$|\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2(1+2|E|/L)^{-1}g^{-1} < \tilde{p} < |\mathrm{Per}\left(U_{1_N\times 1_N}\right)|^2(1+2|E|/L)g, \quad (31)$$

where we now have a multiplicative error estimate with error $g' = (1+2|E|/L)g$.

The #P-hardness of approximating $\mathrm{Per}(X)^2$ (Theorem 28 from [1]) where X is an $N\times N$ matrix, requires a choice of g that is polynomially dependent on N. This will be achieved here if the extra $(1+2|E|/L)$ term maintains a polynomial scaling.

At this point, we consider the origin of the added error term E to argue how L must scale in terms of N. All of the schemes described in Sec. III depend on a continuous output parameter based on the CV-1 style measurement. However, any experimental realization will output values to within some precision. One particular way to do this, with a connection to discrete computational outputs, is to consider a discretization of the results into b-bit integers. The outcomes R from PRCV-1 are non-negative, so in this case consider b-bit non-negative

integers. R is unbounded for large values, but in our scheme we are only interested in small values so we can choose an arbitrary fixed boundary above which all values are assigned the same bit-string output. With these requirements, results can be partitioned into 2^b regions such that the largest result corresponds to all outcomes above some fixed value, say 1. In the simplest case, if $b = 1$ then we would have the one-bit binary string with '0' representing all results of the continuous value R between 0 and 1 and the binary string '1' representing values greater than 1. Increasing b and continuing this division of the results, we find that the range of values covered by equal size partitions of the values between 0 and 1 we have $t = 2/(2^b - 1) = O(2^{-b})$. It is the scaling in b, and not t, that is required to be considered when analyzing the complexity of the problems modeled on this device.

We will proceed with the analysis below assuming this simple partitioning of the values of R. It should be noted that other discretization strategies can be considered. For example, we could change the discretization thresholds such that smaller regions around zero could be considered by using a common technique called "companding." However, one must be careful here not to sacrifice too much probability by having the zero region scale super-exponentially. Any scaling which results in the region at zero scaling as $O(2^{-poly(b)})$ is sufficient for the exact sampling argument below. This is due to the nature of Stockmeyer's approximate counting algorithm utilized in Ref. [1]. As these strategies do not change the complexity hardness result we present, we will not consider them further in this paper.

Now we will return to considering the error term $|E|$. This quantity, as stated above, scales as $O(t^2)$. But in the number of bits used for discretization it will scale as $O(2^{-2b})$. To achieve a $poly(N)$ scaling of $|E|/L$, a scaling in L of $O(2^{-2b})$ would counteract the error, leaving a $O(1)$ overhead.

So our procedure will achieve a multiplicative estimate with a very quickly decaying lower bound L on the matrix permanents that can be estimated. But what actually determines the choice of the lower bound L that is permitted? The proof of the #P-hardness for approximating $\mathrm{Per}(X)^2$ proceeds in [1] by using this estimation polynomially many times to compute $\mathrm{Per}(X)$ given that X was a zero-one matrix. As part of the proof one needs to use the estimation procedure in a binary search toward a matrix whose permanent is zero and the procedure stops when sufficient precision has been achieved to determine the permanent of the zero-one matrix, as it must be an integer value. Introducing a lower bound on the permanents for which the estimation is valid would have to be compatible with this final precision.

Nevertheless, in the hardness proof for approximate boson sampling, it is assumed that the input matrices X have matrix elements that have a Gaussian distribution. If one is to accept the conjectures of [1], then there is a low probability of randomly having $\mathrm{Per}(X)$ near zero. Specifically, using the words from [1],

> ...if $X \sim \mathcal{G}^{n\times n}$ is Gaussian then ...a $1 - 1/poly(n)$ fraction of [Per(X)'s] probability mass is greater than $\sqrt{n!}/poly(n)$ in absolute value, $\sqrt{n!}$ being the standard deviation.

Using this we could choose the lower bound of $\mathrm{Per}(X)$ to scale as $\sqrt{N!}/poly(N)$. However, we have written the bound for our construction in terms of the submatrix of the unitary matrix U. Following the embedding procedure from [1] this reduces the size of X by a factor involving the matrix norm $||X||$. Therefore the probability reduces by a factor of $||X||^{2N} \leqslant 2^{poly(N)}$ as N (the number of input photons) is a proxy for the problem input size (i.e., the matrix X) which must be efficiently represented in N. This results in a scaling of $L = O(||X||^{-2N} N!/poly(N))$. In terms of the discretization bits the scaling is $b = O(N \log N + poly(N))$, where the $poly(N)$ in this scaling is from the polynomial which bounds $||X||^{-2N}$. So a polynomial scaling in the size of the discretization allows the approximation of permanents which are highly probable when using an approximate boson sampling algorithm. This argument shows that the exact sampling problem presented here, which permits multiplicative permanent estimation with permanents with an exponentially decreasing lower bound, must also be hard to compute with classical resources.

C. Approximate sampling

The arguments just made do not permit one to continue the hardness argument through to the case of approximate sampling as was done in [1] for the CV distributions we have constructed above. The reason is quite simple: only an exponentially small subset of events (i.e., detections near the origin) are used to make the exact sampling argument. Conversely, for the same reason, one also cannot definitively conclude that approximate sampling from this distribution is an efficient classical task.

The approximate sampling criterion for this CV distribution would be

$$\int |P(\boldsymbol{\alpha}) - Q(\boldsymbol{\alpha})| d\vec{\alpha} < \beta, \tag{32}$$

where P is the probability density for our distribution of CV events from the device described above and Q is the computable approximation. The event used for sampling above is the set of $\boldsymbol{\alpha}$, where N α's are around a ball of radius t around the origin. Hence all of this error could potentially be concentrated on our event. This would dominate any exponential prefactors of the matrix permanent in the probability, exponentially reducing the signal that is being fed into the approximate counting algorithm.

Approximate sampling is shown in [1] by potentially utilizing all possible (or more precisely the collision-free subspace of) events to perform the approximate counting. The algorithm cannot know which event is being probed, so a concentrate of error as described above would result in the approximate sampling algorithm being an exact sampling algorithm for almost all results.

So what is required in the CV case is more events. To show classical hardness for this distribution, one needs to argue that events away from the origin are also #P-hard when used exactly. This is difficult for this construction because these events do not reduce to Fock basis measurements under any approximation. Also, there must be combinatorially enough events so that the approximate sampling error can be considered low enough on average over all the events so that hardness is maintained. Even if this specific criterion cannot be met, this does not mean that the distribution is efficient for a classical computation. To show classical efficiency,

either a constructive proof demonstrating efficiency (e.g., the methodology used in [12]) or some other proof technique ruling out classical hardness would be required.

V. CONCLUSION

We have constructed a continuous-variable measurement model we call CV-n which measures in a displaced number state basis. The measurement model can be achieved by mixing an n-photon state with the input on a 50:50 beamsplitter then making homodyne measurements on the output. We have shown how an n Fock basis state measurement can be approximately achieved using this measurement utilizing those cases when the homodyne measurement outcomes are simultaneously small.

We then showed that this measurement model is compatible with the exact boson sampling problem, a computing task that has been shown to be inefficient for any classical device to simulate. We have discussed how this model as presented here is not compatible with approximate boson sampling since the number of detection events utilized, when compared with the whole event space, is too small.

Note added in proof. Recently, we became aware that a related approach to this problem has been described in Ref. [13].

ACKNOWLEDGMENTS

This work was supported by the Australian Research Council Centre of Excellence for Quantum Computation and Communications Technology (Project No. CE110001027). S.R.-K. also acknowledges support from Iran's National Elites Foundation (INEF).

APPENDIX: CV-N MEASUREMENT: CONTINUOUS-VARIABLE MEASUREMENT IN THE DISPLACED NUMBER STATE

In this appendix we obtain the POVM elements of the CV-n measurement; see Fig. 1. The POVM elements of this measurement are of the form

$$\Pi^n(x_{1,\theta},p_{2,\theta}) = \frac{1}{c}\,|\Psi(x_{1,\theta},p_{2,\theta})\rangle\langle\Psi(x_{1,\theta},p_{2,\theta})|\,, \tag{A1}$$

where $x_{1,\theta}$ and $p_{2,\theta}$ are the outcomes of the first and second homodyne measurements, respectively, and c is the normalization constant such that $\int dx_{1,\theta}\,dp_{2,\theta}\Pi(x_{1,\theta},p_{2,\theta}) = I$.

The POVM elements of the first homodyne measurement are

$$|x_{1,\theta}\rangle\langle x_{1,\theta}| = e^{-i\hat{P}_{1,\theta}x_{1,\theta}}\,|x_{1,\theta}=0\rangle\langle x_{1,\theta}=0|\,e^{i\hat{P}_{1,\theta}x_{1,\theta}}, \tag{A2}$$

where

$$\hat{P}_{1,\theta} = -\hat{X}_1\sin\theta + \hat{P}_1\cos\theta. \tag{A3}$$

Notice that $|x_{1,\theta}=0\rangle$ is an infinitely squeezed vacuum state, and can be written as

$$|x_{1,\theta}=0\rangle = \lim_{r\to\infty}\hat{S}(re^{i2\theta})\,|0\rangle\,, \tag{A4}$$

with $\hat{S}(\xi) = \exp[(\xi^*\hat{a}^2 - \xi\hat{a}^{\dagger 2})/2]$ being the squeezing operator and $|0\rangle$ being the vacuum state.

For the second homodyne measurement we have

$$|p_{2,\theta}\rangle\langle p_{2,\theta}| = e^{i\hat{X}_{2,\theta}p_{2,\theta}}\,|p_{2,\theta}=0\rangle\langle p_{2,\theta}=0|\,e^{-i\hat{X}_{2,\theta}p_{2,\theta}}, \tag{A5}$$

where

$$\hat{X}_{2,\theta} = \hat{X}_2\cos\theta + \hat{P}_2\sin\theta. \tag{A6}$$

Similarly, we have

$$|p_{2,\theta}=0\rangle = \lim_{r\to\infty}\hat{S}(-re^{i2\theta})\,|0\rangle\,. \tag{A7}$$

Now using Eqs. (A2) and (A5), $|\Psi(x_{1,\theta},p_{2,\theta})\rangle$ in Eq. (A1) is given by

$$|\Psi(x_{1,\theta},p_{2,\theta})\rangle = \frac{1}{\mathcal{N}}\,\langle n|\underbrace{\hat{\mathcal{U}}_{\rm BS}e^{-i\hat{P}_{1,\theta}x_{1,\theta}}e^{i\hat{X}_{2,\theta}p_{2,\theta}}}_{A}\,|x_{1,\theta}=0,p_{2,\theta}=0\rangle\,, \tag{A8}$$

where $\mathcal{N}$ is the normalization constant. Using the following relations for the 50:50 beamsplitter

$$\hat{\mathcal{U}}_{\rm BS}\begin{pmatrix}\hat{X}_1\\ \hat{X}_2\end{pmatrix}\hat{\mathcal{U}}_{\rm BS}^\dagger = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}\hat{X}_1\\ \hat{X}_2\end{pmatrix}, \tag{A9}$$

$$\hat{\mathcal{U}}_{\rm BS}\begin{pmatrix}\hat{P}_1\\ \hat{P}_2\end{pmatrix}\hat{\mathcal{U}}_{\rm BS}^\dagger = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}\hat{P}_1\\ \hat{P}_2\end{pmatrix}, \tag{A10}$$

the expression A in Eq. (A8) becomes

$$\hat{\mathcal{U}}_{\rm BS}e^{-i\hat{P}_{1,\theta}x_{1,\theta}}e^{i\hat{X}_{2,\theta}p_{2,\theta}} = D_1(\beta_1)D_2(\beta_2)\hat{\mathcal{U}}_{\rm BS}, \tag{A11}$$

where

$$\beta_1 = \tfrac{1}{2}(x_{1,\theta}\cos\theta - p_{2,\theta}\sin\theta) + i\,\tfrac{1}{2}(x_{1,\theta}\sin\theta + p_{2,\theta}\cos\theta), \tag{A12}$$

$$\beta_2 = \tfrac{1}{2}(-x_{1,\theta}\cos\theta - p_{2,\theta}\sin\theta) + i\,\tfrac{1}{2}(-x_{1,\theta}\sin\theta + p_{2,\theta}\cos\theta), \tag{A13}$$

and $D(\beta)$ is the displacement operator. Thus, Eq. (A8) becomes

$$|\Psi(x_{1,\theta},p_{2,\theta})\rangle = \frac{1}{\mathcal{N}}\,\langle n|\,D_1(\beta_1)D_2(\beta_2)\hat{\mathcal{U}}_{\rm BS}\,|x_{1,\theta}=0,p_{2,\theta}=0\rangle\,. \tag{A14}$$

Using

$$\hat{\mathcal{U}}_{\rm BS}\,|x_{1,\theta}=0,p_{2,\theta}{=}0\rangle = \lim_{r\to\infty}\hat{\mathcal{U}}_{\rm BS}\hat{S}(re^{i2\theta})\hat{S}(-re^{i2\theta})\,|0,0\rangle \tag{A15}$$

$$= \lim_{r\to\infty}\frac{1}{\cosh r}\sum_{k=0}^{\infty}(e^{2i\theta}\tanh r)^k\,|k,k\rangle \tag{A16}$$

$$= \lim_{r\to\infty}\frac{e^{2i\theta\hat{n}_2}}{\cosh r}\sum_{k=0}^{\infty}(\tanh r)^k\,|k,k\rangle\,, \tag{A17}$$

we can write Eq. (A14) as

$$|\Psi(x_{1,\theta},p_{2,\theta})\rangle = \frac{1}{\mathcal{N}}D_1(\beta_1)\lim_{r\to\infty}\frac{1}{\cosh r}\times\sum_{k=0}^{\infty}(\tanh r)^k\,|k\rangle\,\langle n|\,D_2(\beta_2)e^{2i\theta\hat{n}_2}\,|k\rangle\,. \tag{A18}$$

It can be shown that

$$\langle n|\, D_2(\beta_2) e^{2i\theta \hat{n}_2}\, |k\rangle = e^{2i\theta n}\, \langle n|\, D_2(\beta_2')\, |k\rangle = e^{2i\theta n}\, \langle k|\, D_2(-\beta_2'^*)\, |n\rangle\,, \tag{A19}$$

where

$$\beta_2' = \tfrac{1}{2}(-x_{1,\theta}\cos\theta + p_{2,\theta}\sin\theta) + i\,\tfrac{1}{2}(x_{1,\theta}\sin\theta + p_{2,\theta}\cos\theta), \tag{A20}$$

and the second equality can be seen using this expression

$$\langle n|\, D(\alpha)\, |k\rangle = \sqrt{\frac{k!}{n!}} e^{-|\alpha|^2/2} \alpha^{n-k} L_k^{n-k}(|\alpha|^2), \tag{A21}$$

with $L_k^m(.)$ being the generalized Laguerre polynomials.

By substituting from Eq. (A19) into Eq. (A18), we get

$$|\Psi(x_{1,\theta}, p_{2,\theta})\rangle = \frac{1}{\mathcal{N}} D_1(\beta_1) e^{2i\theta n} \lim_{r\to\infty} \frac{1}{\cosh r} \times \sum_{k=0}^{\infty} (\tanh r)^k\, |k\rangle\, \langle k|\, D_2(-\beta_2'^*)\, |n\rangle\,. \tag{A22}$$

As this state must be normalized in the limit of $r \to \infty$, it is straightforward to see that

$$\mathcal{N}^{-1} = \cosh r. \tag{A23}$$

As $\lim_{r\to\infty} \tanh r = 1$, we now have

$$|\Psi(x_{1,\theta}, p_{2,\theta})\rangle = e^{2i\theta n} D_1(\beta_1) \left(\lim_{r\to\infty} \sum_{k=0}^{\infty} (\tanh r)^k\, |k\rangle\langle k| \right) D_2(-\beta_2'^*)\, |n\rangle \tag{A24}$$

$$= e^{2i\theta n} D_1(\beta_1) D_2(-\beta_2'^*)\, |n\rangle \tag{A25}$$

$$= e^{2i\theta n} e^{(-\beta_1\beta_2' + \beta_1^*\beta_2'^*)/2} D(\beta_1 - \beta_2'^*)\, |n\rangle \tag{A26}$$

$$= e^{2i\theta n} e^{(-\beta_1\beta_2' + \beta_1^*\beta_2'^*)/2} D(\alpha)\, |n\rangle\,, \tag{A27}$$

where

$$\alpha = (x_{1,\theta} + i p_{2,\theta}) e^{i\theta} = r e^{i\theta + i\phi}. \tag{A28}$$

Therefore, POVM elements of the CV-n measurement are

$$\Pi^n(x_{1,\theta}, p_{2,\theta}) = \frac{1}{2\pi} D(\alpha)\, |n\rangle\langle n|\, D^\dagger(\alpha). \tag{A29}$$

Notice that it can be simply checked that

$$\frac{1}{2\pi} \int d^2\, \alpha D(\alpha)\, |n\rangle\langle n|\, D^\dagger(\alpha) = I. \tag{A30}$$

Next we describe the phase-randomized CV-n measurement. If the phase θ of the local oscillator is randomized, with the relative phase $\pi/2$ being fixed, we have

$$\Pi^n(R) = \frac{1}{2\pi} \int d\theta\, D(\sqrt{R} e^{i\theta + i\phi})\, |n\rangle\langle n|\, D^\dagger(\sqrt{R} e^{i\theta + i\phi}), \tag{A31}$$

where

$$R = r^2 = x_{1,\theta}^2 + p_{2,\theta}^2. \tag{A32}$$

The matrix elements of this operator in the Fock basis are

$$\langle k|\, \Pi^n(R)\, |l\rangle = \frac{1}{2\pi} \int d\theta\, \langle k|\, D(\sqrt{R} e^{i\theta + i\phi})\, |n\rangle\langle n|\, D^\dagger(\sqrt{R} e^{i\theta + i\phi})\, |l\rangle \tag{A33}$$

$$= \frac{1}{2\pi} e^{-R} \frac{n!}{\sqrt{k!l!}} (\sqrt{R})^{k+l-2n} L_n^{l-n}(R) L_n^{k-n}(R) \times \int d\theta\, e^{i(\theta+\phi)(k-l)} \tag{A34}$$

$$= \frac{1}{2\pi} e^{-R} \frac{n!}{\sqrt{k!l!}} (\sqrt{R})^{k+l-2n} L_n^{l-n}(R) L_n^{k-n}(R) 2\pi \delta_{k,l}. \tag{A35}$$

Therefore, the POVM elements of the phase-randomized measurement are given by

$$\Pi^n(R) = n! e^{-R} \sum_{k=0}^{\infty} \frac{R^{k-n}}{k!} \left[L_n^{k-n}(R)\right]^2 |k\rangle\langle k|\,. \tag{A36}$$

It can be simply verified that

$$\int_0^\infty dR\; \Pi^n(R) = I. \tag{A37}$$

For $n = 1$, using

$$L_n^{k-n}(R) = \sum_{j=0}^{n} \frac{(-1)^j k!}{j!(k-n+j)!(n-j)!} R^j, \tag{A38}$$

we have

$$L_1^{k-1}(R) = k - R; \tag{A39}$$

so the POVM elements (A36) become

$$\Pi^1(R) = e^{-R} \sum_{k=0}^{\infty} \frac{R^{k-1}}{k!} (k-R)^2\, |k\rangle\langle k|\,. \tag{A40}$$

Notice that

$$\Pi^1(0) = |1\rangle\langle 1| \tag{A41}$$

as expected.

[1] S. Aaronson and A. Arkhipov, Theory Comput. **9**, 143 (2013).
[2] T. Douce, D. Markham, E. Kashefi, E. Diamanti, T. Coudreau, P. Milman, P. van Loock, and G. Ferrini, Phys. Rev. Lett. **118**, 070503 (2017).
[3] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto, Phys. Rev. Lett. **88**, 097904 (2002); S. D. Bartlett and B. C. Sanders, J. Mod. Opt. **50**, 2331 (2003).
[4] A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. O'Brien, and T. C. Ralph, Phys. Rev. Lett. **113**, 100502 (2014).
[5] S. Rahimi-Keshari, A. P. Lund, and T. C. Ralph, Phys. Rev. Lett. **114**, 060501 (2015).
[6] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, arXiv:1612.01199.
[7] S. Scheel, arXiv:quant-ph/0406127.
[8] L. J. Stockmeyer, in *Proceedings of the 15th Annual ACM Symposium on Theory of Computing* (ACM, New York, 1983), pp. 118–126.
[9] M. Sipser, in *Proceedings of the 15th ACM Annual Symposium on Theory of Computing* (ACM, New York, 1983), pp. 330–335.
[10] C. Lautemann, Inf. Proc. Lett. **17**, 215 (1983).
[11] K. E. Cahill and R. J. Glauber, Phys. Rev. **177**, 1857 (1969).
[12] S. Rahimi-Keshari, T. C. Ralph, and C. M. Caves, Phys. Rev. X **6**, 021039 (2016).
[13] L. Chakhmakhchyan and N. J. Cerf, arXiv:1705.05299 [quant-ph].