



# Training Gaussian boson sampling by quantum machine learning

Claudio Conti<sup>1,2,3</sup>

Received: 3 March 2021 / Accepted: 22 September 2021  
© The Author(s) 2021

## Abstract

We use neural networks to represent the characteristic function of many-body Gaussian states in the quantum phase space. By a pullback mechanism, we model transformations due to unitary operators as linear layers that can be cascaded to simulate complex multi-particle processes. We use the layered neural networks for non-classical light propagation in random interferometers, and compute boson pattern probabilities by automatic differentiation. This is a viable strategy for training Gaussian boson sampling. We demonstrate that multi-particle events in Gaussian boson sampling can be optimized by a proper design and training of the neural network weights. The results are potentially useful to the creation of new sources and complex circuits for quantum technologies.

**Keywords** Machine learning · Gaussian Boson sampling

## 1 Introduction

The development of new models and tools for machine learning (ML) is surprisingly affecting the study of many-body quantum systems and quantum optics (Huang et al. 2021). Neural networks (NN) enable representations of high-dimensional systems and furnish a universal ansatz for many purposes, like finding the ground state of many-body Hamiltonians (Carleo et al. 2019), including dissipative systems (Vicentini et al. 2019; Mangini et al. 2021).

Unsupervised and supervised learning endow new designs for quantum circuits (Marquardt 2021), metrology and cryptography (Lumino et al. 2018; Fratalocchi et al. 2021), multilevel gates (Marcucci et al. 2020), and Bell tests (Melnikov et al. 2020). NN are also triggering new fundamental investigations in quantum neuromorphic and wave computing (Marcucci et al. 2020; Hughes et al. 2019; Ballarini et al. 2020; Nokkala et al. 2020; Marković and Grollier 2020; Silva et al. 2021), quantum

thermodynamics (SgROI et al. 2021), and topological photonics (PiloZZi et al. 2021).

The impact of ML in quantum optics and many-body physics is related to the versatile representation that the NN models furnish for functions of an arbitrary number of variables. Also, the powerful application programming interfaces (APIs), as TensorFlow, enable many new features and tools to compute and design many-body Hamiltonians or large-scale quantum gates (Broughton et al. 2020).

Here, we show that NN models are also useful when considering representations in the phase space, as the characteristic functions  $\chi$  or the Q-representation (Barnett and Radmore 1997). Unitary operators, as squeezers or displacers, act on the phase-space as variable transformations that correspond to layers in the NN model. Hence, a multilayer NN may encode phase-space representations of complex many-body states. This encoding has two main advantages: on the one hand, one can quickly build complex quantum states by combining NN layers; on the other hand, one can use the automatic graph building and API differentiation technology to compute observables. Also, graphical and tensor processing units (GPU and TPU) may speed up the computation.

In the following, we show how to compute the probability of multi-particle patterns when Gaussian states propagate in a system made of squeezers and interferometers. This problem corresponds to the renowned Gaussian Boson sampling (Hamilton et al. 2017; Quesada et al. 2018),

✉ Claudio Conti  
claudio.conti@uniroma1.it

<sup>1</sup> Department of Physics, University of Sapienza, P.le Aldo Moro 5, 00185 Rome, Italy

<sup>2</sup> Institute for Complex Systems, National Research Council (ISC-CNR), Via dei Taurini 19, 00185 Rome, Italy

<sup>3</sup> Research Center Enrico Fermi, Via Panisperna 89a, 00184 Rome, Italy

which recently demonstrated the quantum advantage at an impressing scale (Zhong et al. 2020), following earlier realizations (Tillmann et al. 2012; Broome et al. 2013; Spring et al. 2013; Spagnolo et al. 2014; Carolan et al. 2014; Wang et al. 2019) of the original proposal by Aharanson and Arkhipov (Aaronson and Arkhipov 2013). The theory of Gaussian Boson sampling (GBS) heavily relies on phase-space methods (Kruse et al. 2019), making it an exciting NN test-bed supported by recently reported trainable hardware (Arrazola et al. 2021; Hoch et al. 2021; Zhong et al. 2021).

A notable outcome of adopting NN models in the phase space is the possibility of training multi-particle statistics (Banchi et al. 2020) and other features as the degree of entanglement. Indeed, most of the reported investigations in quantum ML focus either on using NN models as a variational ansatz or tailoring the input/output response of a quantum gate. On the contrary, ML in the phase space permits optimizing many-particle features, for example, to increase the probability of multi-photon events. NN may open new strategies to generate non-classical light or enhance the probability of observing large-scale entanglement with relevance in many applications. Here, we derive the NN representing the characteristic function of the Gaussian boson sampling setup. Proper NN training increases the photon-pair probability by orders of magnitude.

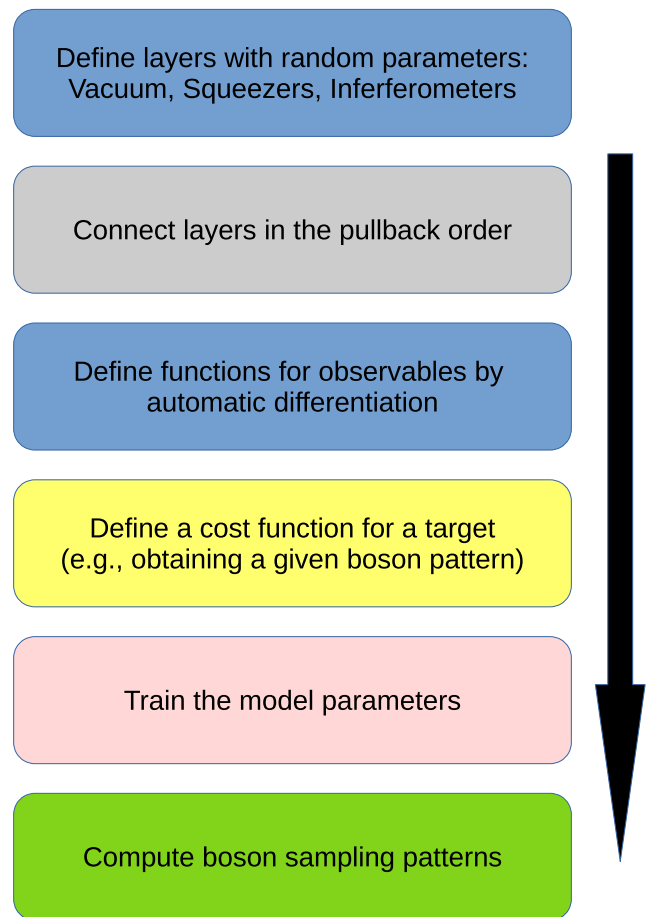
Figure 1 shows the general workflow of the proposed methodology, the different steps enable to define a trainable model for optimizing Gaussian boson sampling. In Section 2, we introduce the way we adopt a neural network to compute the characteristic function. In Section 3, we detail how to compute the observable as derivatives of the characteristic function neural network. In Section 4, we show how to compute the Gaussian boson sampling patterns. In Section 5, we introduce the loss function and describe the training of the model to optimize specific patterns. Conclusions are drawn in Section 6.

## 2 Characteristic function as a neural network

In the phase space, we represent a  $n$ -body state by the complex characteristic function  $\chi(\mathbf{x}) = \chi_R(\mathbf{x}) + i\chi_I(\mathbf{x})$  of a real vector  $\mathbf{x}$  (Gardiner and Zoller 2004; Barnett and Radmore 1997).  $\mathbf{x}$  has dimension  $1 \times N$  with  $N = 2n$ . For Gaussian states (Wang et al. 2007)

$$\chi(\mathbf{x}) = e^{-\frac{1}{4}\mathbf{g}\mathbf{x}\mathbf{x}^\top + i\mathbf{d}\mathbf{x}} \tag{1}$$

with  $\mathbf{g}$  the real covariance  $N \times N$  matrix, and  $\mathbf{d}$  the real displacement  $N \times 1$  vector. In our notation, we omit the



**Fig. 1** Workflow of the proposed methodology to train Boson sampling by representing the characteristic function as a neural network

symbols of the dot product such that  $\mathbf{x}\mathbf{d}$  and  $\mathbf{x}\mathbf{g}\mathbf{x}^\top$  are scalars. One has  $(j, k = 0, 1, 2, \dots, N - 1)$

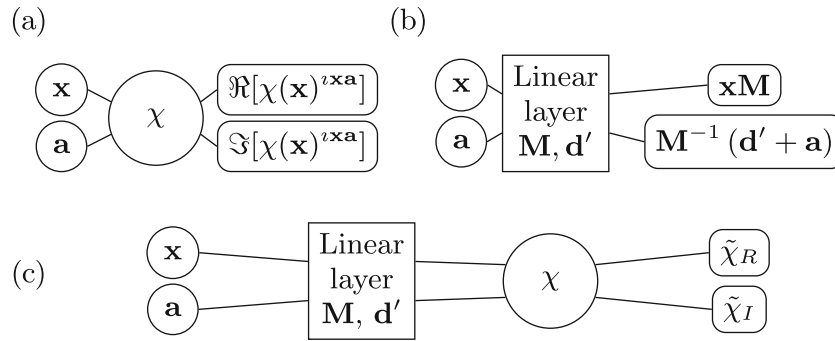
$$\langle \hat{R}_j \rangle = d_j = \left. \frac{\partial \chi}{\partial x_j} \right|_{\mathbf{x}=0}, \tag{2}$$

and

$$g_{jk} = 2\langle (\hat{R}_j - d_j)(\hat{R}_k - d_k) \rangle - iJ_{jk}, \tag{3}$$

being  $\mathbf{J} = \bigoplus_{j=0}^{n-1} \mathbf{J}_1$ ,  $\mathbf{J}_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$  (Wang et al. 2007). In Eq. 2, the canonical variables,  $\hat{q}_j = \hat{R}_{2j}$  and  $\hat{p}_j = \hat{R}_{2j+1}$ , with  $j = 0, 1, \dots, n - 1$ , are organized in the  $N \times 1$  operator array  $\hat{\mathbf{R}}$ . As shown in Fig. 2a, the characteristic function is a NN layer with two real outputs  $\chi_R$  and  $\chi_I$ . The  $\chi$  layer has two inputs:  $\mathbf{x}$ , and a auxiliary bias  $N \times 1$  vector  $\mathbf{a}$ , for later convenience.

The vacuum state is a Gaussian state with  $\mathbf{g} = \mathbf{1}$  and  $\mathbf{d} = \mathbf{0}$ . From the vacuum, one can generate specific states by unitary transformations, as displacement or squeezing operators. These transform the canonical variables as  $\hat{\tilde{\mathbf{R}}} = \mathbf{M}\hat{\mathbf{R}} + \mathbf{d}'$ , where the symplectic matrix  $\mathbf{M}$  and the vector  $\mathbf{d}'$  depend on the specific operator (detailed, e.g., in Wang



**Fig. 2** (a) A neural network model for the characteristic function. Two inputs, a data vector  $\mathbf{x}$  with shape  $1 \times N$  and a bias vector  $\mathbf{a}$  with shape  $N \times 1$  feed the model that compute  $\chi$  and returns the real and imaginary parts of  $\chi(\mathbf{x})e^{i\mathbf{x}\mathbf{a}}$ . (b) A layer representing a linear transformation by a symplectic  $N \times N$  matrix  $\mathbf{M}$  and a displacement  $N \times 1$  vector  $\mathbf{d}'$ . With such a definition

et al. (2007)). The characteristic function changes as

$$\tilde{\chi}(\mathbf{x}) = \chi(\mathbf{x}\mathbf{M})e^{i\mathbf{x}\mathbf{d}'+i\mathbf{x}\mathbf{a}} = \chi(\mathbf{x}\mathbf{M})e^{i(\mathbf{x}\mathbf{M})\mathbf{M}^{-1}(\mathbf{d}'+\mathbf{a})} \tag{4}$$

We represent the linear transformation as a NN layer with two inputs  $\mathbf{x}$  and  $\mathbf{a}$  and two outputs  $\mathbf{x}\mathbf{M}$  and  $\mathbf{M}^{-1}(\mathbf{d}'+\mathbf{a})$  (Fig. 2b). By this definition, Eq. 4 is as a two-layer NN.

Figure 2c shows  $\tilde{\chi}$  as the “pullback” of the linear layer from the  $\chi$  layer. The two layers form a NN that can be implemented with common APIs.<sup>1</sup> Given the vacuum state with characteristic function  $\chi$ , one can build the NN model of an arbitrary state by multiple pullbacks. Indeed, we defined the linear layers in a way that they can be cascaded. Figure 4a below shows a  $n$ -mode squeezed vacuum as a multiple pullback of single mode squeezers, each acting on a different mode.

### 3 Observables

Observables are computed as derivatives of the NN model. For example, the mean photon number per mode is related to the derivatives of the characteristic function. The mean photon number for mode  $j$ , is

$$\langle \hat{n}_j \rangle = -\frac{1}{2} \left( \nabla_j^2 + 1 \right) \chi \Big|_{\mathbf{x}=0} \tag{5}$$

being  $\nabla_j^2 = \partial_{q_j}^2 + \partial_{p_j}^2$  and  $q_j = x_{2j}$  and  $p_j = x_{2j+1}$ . The differential photon number of modes  $j$  and  $k$  is

$$\langle (\hat{n}_j - \hat{n}_k)^2 \rangle = \left[ \frac{1}{4} \left( \nabla_j^2 - \nabla_k^2 \right)^2 - \frac{1}{2} \right] \chi \Big|_{\mathbf{x}=0} \tag{6}$$

Automatic differentiation packages enables an efficient computations of the derivatives of the NN model.

<sup>1</sup>A TensorFlow implementation in a Jupyter notebook is available at <https://github.com/nonlinearwaves/BosonSampling>

layers can be cascaded, and one can represent single mode squeezers, interferometers, and other unitary operators. (c) A model representing a state with characteristic function  $\chi$ , subject to a unitary transformation. This is a pullback of a linear transform from the original state, which produces a new state with characteristic function  $\tilde{\chi}$  [see Eq. 4]

### 4 Gaussian boson sampling with the neural network model

In the GBS protocol, one considers a many-body squeezed vacuum state propagating in an Haar interferometer, which distributes the photons in the output modes. For modelling GBS, we hence need squeezing layers and a layer representing the transmission through random interferometers. The squeezing layers are realized by a proper design of the corresponding symplectic matrices  $\mathbf{M}$  with  $\mathbf{d} = 0$ . We implement the Haar matrix operator by QuTiP software (Johansson et al. 2013). Figure 3 shows a pseudo-code to build the neural network model by composing different layers.

Figure 4b is a graphical representation of the GBS NN model (Hamilton et al. 2017).

Boson sampling corresponds to computing the probability  $\Pr(\bar{\mathbf{n}})$  of finding  $\bar{n}_0$  photons in mode 0,  $\bar{n}_1$  photons in mode 1, and so forth.  $\bar{\mathbf{n}} = (\bar{n}_0, \bar{n}_1, \dots, \bar{n}_{n-1})$  is a given photon pattern. Letting  $\hat{\rho}$  the density matrix, one has

$$\Pr(\bar{\mathbf{n}}) = \text{Tr}[\hat{\rho}|\bar{\mathbf{n}}\rangle\langle\bar{\mathbf{n}}|],$$

with

$$|\bar{\mathbf{n}}\rangle\langle\bar{\mathbf{n}}| = \otimes_{j=0}^{n-1} |\bar{n}_j\rangle\langle\bar{n}_j|.$$

Correspondingly (Kruse et al. 2019),

$$\Pr(\bar{\mathbf{n}}) = \frac{1}{\bar{\mathbf{n}}!} \prod_{j=0}^{n-1} \left( \frac{\partial^2}{\partial \alpha_j \partial \alpha_j^*} \right)^{\bar{n}_j} e^{\sum_j |\alpha_j|^2} Q_\rho(\alpha, \alpha^*) \Big|_{\alpha=0} \tag{7}$$

where  $\bar{\mathbf{n}}! = \bar{n}_0! \bar{n}_1! \dots \bar{n}_{n-1}!$  and

$$Q_\rho = \pi^n \langle \alpha | \rho | \alpha \rangle$$

is the Q-representation of the density matrix (Gardiner and Zoller 2004; Barnett and Radmore 1997) with  $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$  complex displacements.

```

Input = input layer                                ▷ Create input layer
V = vacuum layer                                  ▷ Create vacuum layer

for i=0,1,2,...,n-1 do
    S[i] = squeezer layer for mode i                ▷ Create a squeezing layer per mode
end for

R = random interferometer                          ▷ Create a random interferometer
x, a= Input                                       ▷ Define input tensors x and a
x, a= R (x,a)                                     ▷ Connect interferometer in pullback order

for i=0,1,2,...,n-1 do
    x, a= S[i](x,a)                                ▷ Connect squeezing layers in pullback order
end for
x, a= V(x,a)                                       ▷ Connect vacuum layer in pullback order
    
```

Fig. 3 Pseudo-code for the creation of a neural network representing a Gaussian boson sampling experiment

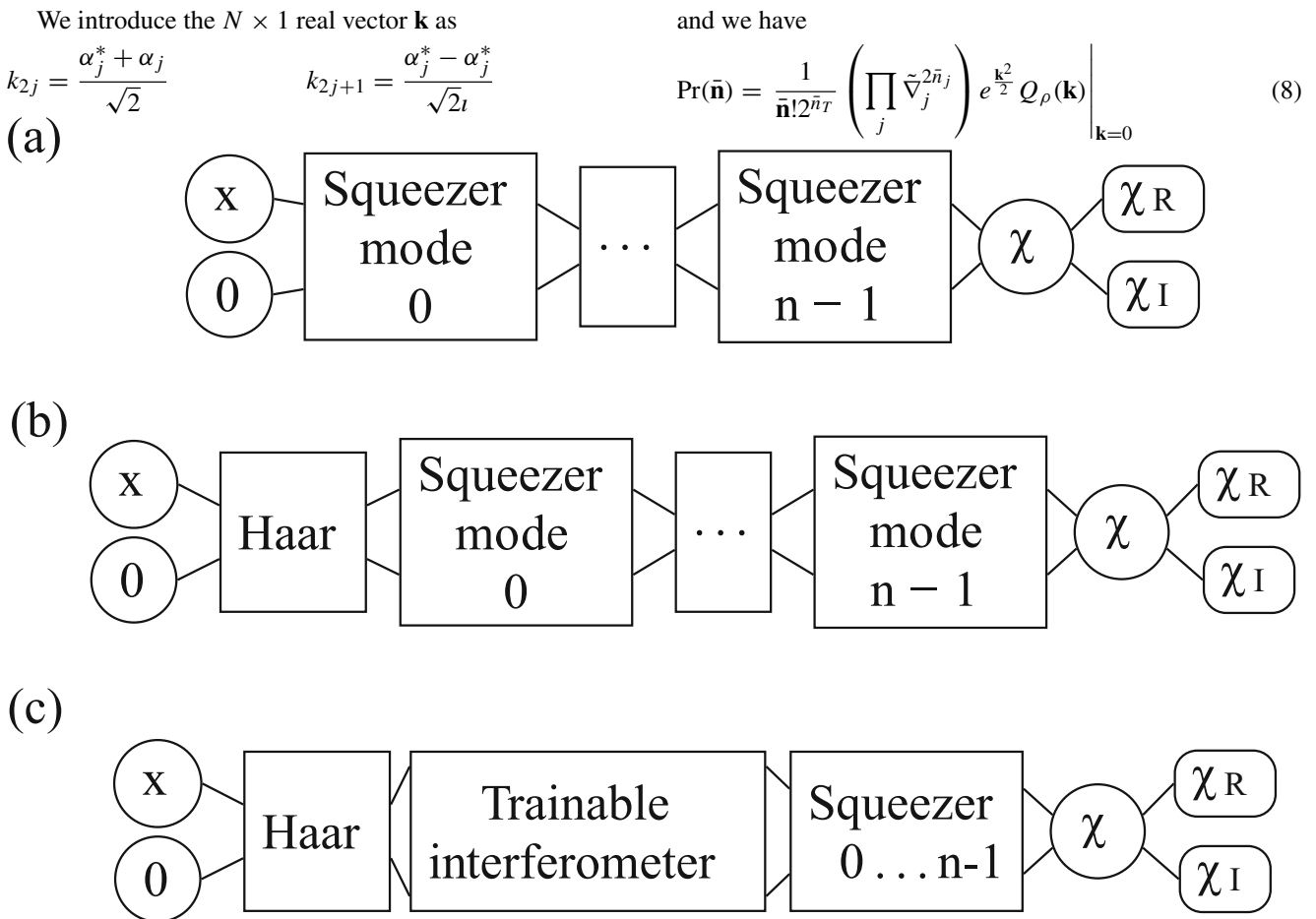


Fig. 4 (a) A multiple pullback that represents a many-body squeezed vacuum, obtained by a vacuum state  $\chi$  by cascading  $n$  identical single mode squeezers. The order of the squeezers is not relevant as they act of different modes.  $\chi_R$  and  $\chi_I$  are the real and imaginary part of the resulting characteristic function. (b) GBS setup, a  $n$ -body squeezed vacuum enters an Haar interferometer. Note that the order of the

operators, from the vacuum to the interferometer goes from right to left. (c) GBS setup including a trainable random interferometer before entering the Haar interferometer. The multiple squeezers are represented as a single block. The trainable interferometer can optimize the probability of pair generation. The right panel shows the architecture of the TensorFlow model for  $n = 6$

with  $\tilde{\nabla}_j^2 = \partial^2/\partial k_{2j} + \partial^2/\partial k_{2j+1}$  and  $\bar{n}_T = \sum_{j=0}^{n-1} \bar{n}_j$ .  $\mathcal{Q}_p$  in Eq. 8 can be evaluated explicitly as a multidimensional Gaussian integral:

$$\Pr(\bar{\mathbf{n}}) = \frac{1}{\bar{\mathbf{n}}! 2^{\bar{n}_T}} \left( \prod_j \tilde{\nabla}_j^{2\bar{n}_j} \right) \mathcal{Q}(\mathbf{k}) \Big|_{\mathbf{k}=0} \tag{9}$$

with  $(p, q = 0, 1, \dots, N - 1)$

$$\mathcal{Q}(\mathbf{k}) = \frac{1}{\sqrt{2^n \det A}} e^{\frac{1}{2} \mathbf{k}^2 e^{-\frac{1}{2} \sum_{pq} A_{pq}^{-1} (k_p - d_q)(k_p - d_q)}} \tag{10}$$

being  $A_{pq} = \frac{1}{2} (g_{pq} + \delta_{pq})$ . Equations 9 and 10 can be implemented as further layers of the NN, and the probability of a given pattern computed by running the model. Figure 5a shows an example of the pattern probability distribution with  $n = 6$ , obtained by using the NN model in Fig. 4b with squeezing parameters  $r_j = 0.88$  and  $\phi_j = \pi/4$ , such that all the single mode squeezers are identical, each with mean photon number  $\sinh(r_j)^2 \simeq 1$ . As in Hamilton et al. (2017), we consider patterns with  $\langle \hat{n}_j \rangle = \{0, 1\}$ .

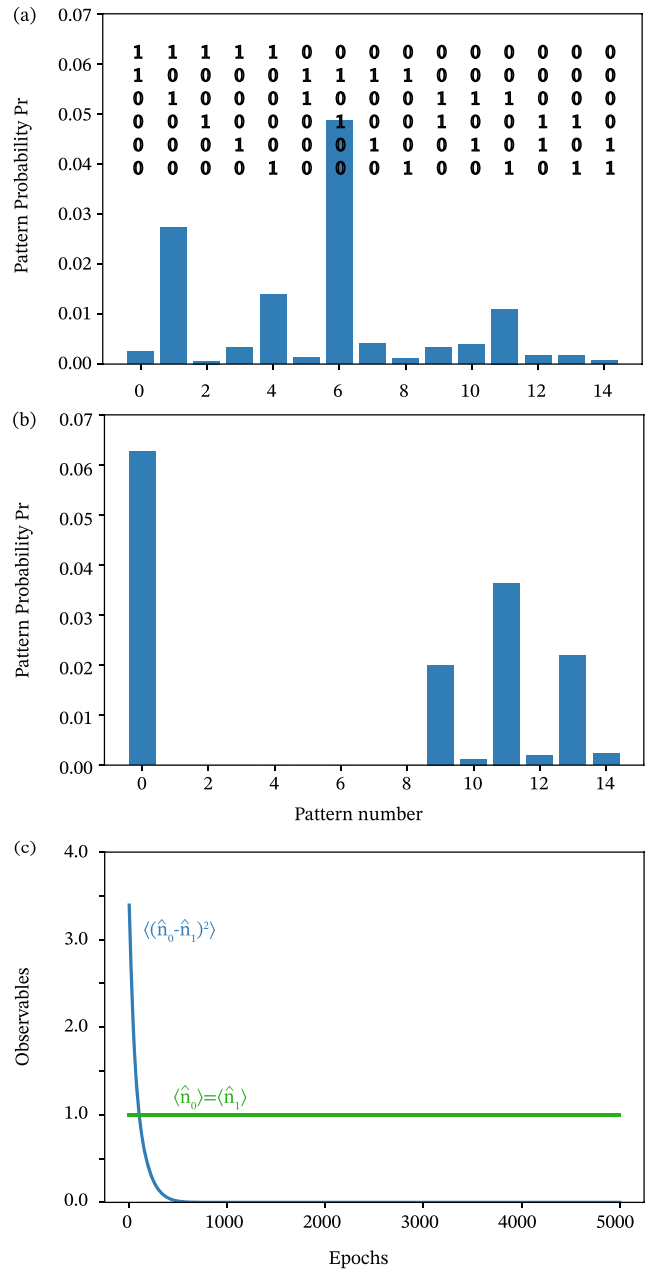
### 5 Training Gaussian boson sampling

Our interest is understanding if we can train the model to maximize the generation of specific patterns, e.g., a photon pair in modes 0 and 1. Using complex media to tailor linear systems is a well renowned technique as, for example, to synthesize specific gates (Leedumrongwatthanakun et al. 2020; Taballione et al. 2019) or taming entanglement (Valencia et al. 2020). Here, we use the NN model in the phase space to optimize multi-particle events.

One could use the squeezing parameters in the model in Fig. 4b as training parameters. However, the degree of squeezing affects the number of particles per mode, and we want to alter the statistical properties of states without changing the average number of particles. We hence consider a GBS setup with an additional trainable interferometer as in Fig. 4c, which is typically realized by amplitude or phase modulators.

In Fig. 4c,  $n$  squeezed vacuum modes impinge on a trainable interferometer and then travel through a Haar interferometer. Instead of two distinct interferometers, one could use a single device (i.e., combine the Haar interferometer with the trainable interferometer), but we prefer to distinguish the trainable part from the mode-mixing Haar unitary operator.

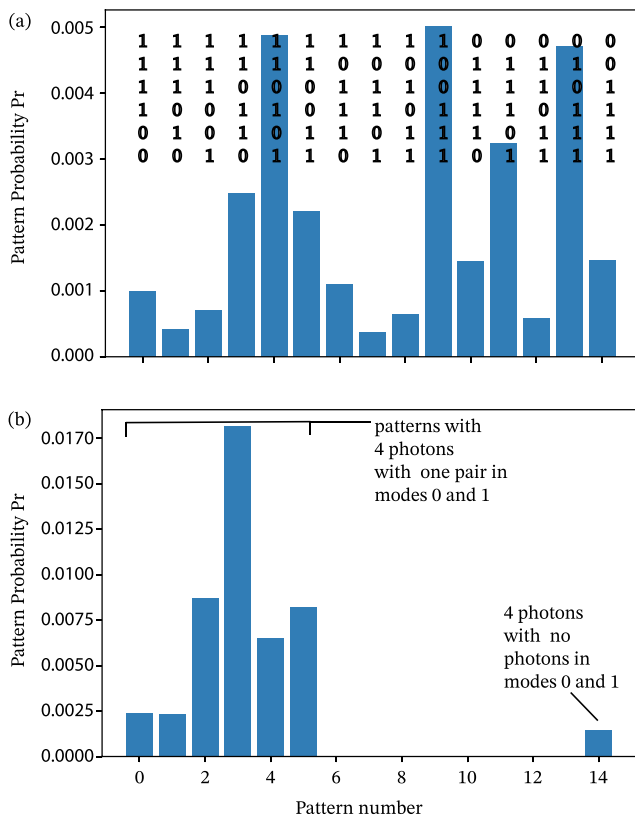
Given  $n$  modes, our goal is to maximize the probability of patterns that contains a pair of photons in the mode 0 or 1. For example, for  $n = 6$ , this means maximizing the



**Fig. 5** (a) Probability distribution of patterns with two photons for  $n = 6$  in the model in Fig. 4c, before training. The insets detail the particle distribution in the patterns. (b) As in (a) after training, the probability of finding a pair in mode 0 and 1 is enhanced by more than one order of magnitude. (c) Mean photon number in mode 0 and 1 during the training epochs (green), and expected differential photon number  $\langle (\hat{n}_0 - \hat{n}_1)^2 \rangle$  in the two modes, which vanishes after thousands of epochs. The statistical distribution of pairs changes at a constant photon number per mode. Data generated by the code in <https://github.com/nonlinearwaves/BosonSampling>

probability of  $\bar{\mathbf{n}} = (1, 1, 0, 0, 0, 0)$  with respect to  $\bar{\mathbf{n}} = (1, 0, 0, 1, 0, 0)$ . We use as loss function

$$\mathcal{L} = e^{\langle (\hat{n}_0 - \hat{n}_1)^2 \rangle} \tag{11}$$



**Fig. 6** (a) Probability distribution of patterns with 4 photons ( $n = 6$ ) in the model in Fig. 4c before training. The insets detail the particles in each pattern. (b) As in (a) after training; the probability of patterns with two photons in modes 0 and 1 is maximized. Data generated by the code in <https://github.com/nonlinearwaves/BosonSampling>

which is minimal when the expected differential number of photons in mode 0 and mode 1 vanishes. This is the case when the state has a particle pair in mode 0 and mode 1. We stress the difference in using other cost functions, which involve the expected number of photons per mode as, e.g.,

$$\mathcal{L}_0 = e^{(\langle \hat{n}_0 \rangle - \langle \hat{n}_1 \rangle)^2}.$$

The linear interferometer does not affect the average number of photons (which are mixed by the Haar layer). Correspondingly, training using  $\mathcal{L}_0$  is not effective to generate entangled pairs. On the contrary,  $\mathcal{L}$  in Eq. 11 contains  $\langle \hat{n}_0 \hat{n}_1 \rangle$ , which is maximal with a photon pair in modes 0 and 1.

Figure 5a shows the computed probabilities of pairs for the model in Fig. 4c, with a random instance of the Haar and the linear interferometers. Training strongly alters this statistical distribution, as shown in Fig. 5b.

Figure 5c shows the trend during the training epochs of  $\langle (\hat{n}_0^2 - \hat{n}_1^2) \rangle$ , which goes to zero while the mean photon numbers  $\langle \hat{n}_0 \rangle$  and  $\langle \hat{n}_1 \rangle$  remain unaltered.

Training also maximizes higher photon events, as in the pattern  $\bar{\mathbf{n}} = (1, 1, 1, 1, 0, 0)$  with 4 photons and  $n = 6$ . Fig. 6a shows the pattern probability with 4 photons. After training with the loss function in Eq. 11,  $\Pr(\bar{\mathbf{n}})$  substantially increases for the patterns with four photons containing 1 pair in modes 0 and 1 (Fig. 6b).

## 6 Conclusions

We have shown that a many-body characteristic function may be reformulated as a layered neural network. This approach enables to build complex states for various applications, as gate design or boson sampling.

A common argument in criticizing quantum neural networks is that the linear quantum mechanics does not match with the nonlinearity-eager NN models. However, recent investigations show that nonlinearity may be introduced in quantum neural networks (Zhao and Gao 2021). Our remark is that if we formulate quantum mechanics in the phase space, nonlinearity arises in the characteristic function (or other representation). We analyzed this strategy in the simplest case of Gaussian states. The resulting model is universal and may be trained for different purposes. For this reason, phase space models allow naturally in dealing with non-classical states and computing observables by derivatives. This formulation opens many opportunities. For example, the optimization of multi-particle events can be extended to fermionic fields. As a drawback, computing boson patterns probabilities by NN APIs is not expected to be competitive with highly optimized algorithms running on large-scale clusters (Quesada and Arrazola 2020; Li et al. 2020). Still, it appears to be a versatile and straightforward methodology.

Here, we have shown many-body quantum state design and engineering by TensorFlow. We have demonstrated how to enhance multi-particle generation, with many potential applications in quantum technologies. In addition, the proposed method enables training Boson sampling without explicitly computing derivatives of the Hafnian (Banchi et al. 2020; Broughton et al. 2020), but resorting to automatic computational packages. We have tested the algorithm with a conventional workstation with a single commercial GPU (NVIDIA QUADRO RTX 4000), with a computational time of the order of few minutes with 6 modes.

The method can be generalized to other boson sampling setups, as including Glauber layers and multi-mode squeezers. Also, it readily allows to test different loss functions for tailoring the boson sampling patterns. Extension beyond Gaussian states can be envisaged by using a general machine learning networks with an arbitrary number of layers and different nonlinearity.



**Funding** Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement. We acknowledge support from Horizon 2020 Framework Programme QuantERA grant QUOMPLEX, by National Research Council (CNR), Grant 731473.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Aaronson S, Arkhipov A (2013) The computational complexity of linear optics. *Theory Comput* 9:143
- Arazola JM et al (2021) Quantum circuits with many photons on a programmable nanophotonic chip. *Nature* 591:54
- Ballarini D, Gianfrate A, Panico R, Opala A, Ghosh S, Dominici L, Ardizzone V, Giorgi MD, Lerario G, Gigli G, Liew TCH, Matuszewski M, Sanvitto D (2020) *Nano Lett* 20:3506
- Banchi L, Quesada N, Arazola JM (2020) *Phys Rev A* 102:012417
- Barnett SM, Radmore PM (1997) *Methods in theoretical quantum optics*. Oxford University Press, New York
- Broome MA, Fedrizzi A, Rahimi-Keshari S, Dove J, Aaronson S, Ralph TC, White AG (2013) Photonic boson sampling in a tunable circuit. *Science* 339:794
- Broughton M, Verdon G, McCourt T, Martinez AJ, Yoo JH, Isakov SV, Massey P, Niu MY, Halavati R, Peters E, Leib M, Skolik A, Streif M, Dollen DV, McClean JR, Boixo S, Bacon D, Ho AK, Neven H, Mohseni M (2020) Tensorflow quantum: A software framework for quantum machine learning. arXiv:2003.02989
- Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, Zdeborová L. (2019) Machine learning and the physical sciences. *Rev Mod Phys* 91:045002
- Carolan J, Meinecke JDA, Shadbolt PJ, Russell NJ, Ismail N, Wörhoff K., Rudolph T, Thompson MG, O'Brien JL, Matthews JCF, Laing A (2014) On the experimental verification of quantum complexity in linear optics. *Nat Photonics* 8:621
- Fraltalocchi A, Fleming A, Conti C, Falco AD (2021) Nist-certified secure key generation via deep learning of physical unclonable functions in silica aerogels. *Nanophotonics* 10:457
- Gardiner CW, Zoller P (2004) *Quantum noise*, 3rd edn. Springer, Berlin
- Hamilton CS, Kruse R, Sansoni L, Barkhofen S, Silberhorn C, Jex I (2017) Gaussian boson sampling. *Phys Rev Lett* 119:170501
- Huang H-Y, Kueng R, Torlai G, Albert VV, Preskill J (2021) arXiv:2106.1267
- Hughes TW, Williamson IA, Minkov M, Fan S (2019) Wave physics as an analog recurrent neural network. *Sci. Adv.* 5 eaay6946
- Hoch F et al (2021) Boson sampling in a reconfigurable continuously-coupled 3d photonic circuit. arXiv:2106.08260
- Johansson J, Nation P, Nori F (2013) A python framework for the dynamics of open quantum systems. *Comput Phys Commun* 184:1234
- Kruse R, Hamilton CS, Sansoni L, Barkhofen S, Silberhorn C, Jex I (2019) A detailed study of gaussian boson sampling. *Phys Rev A* 100:032326
- Leedumrongwatthanakun S, Innocenti L, Defienne H, Juffmann T, Ferraro A, Paternostro M, Gigan S (2020) Programming linear quantum networks with a multimode fiber. *Nat Photonics* 14:139
- Li Y, Chen M, Chen Y, Lu H, Gan L, Lu C, Pan J, Fu H, Yang G (2020) Benchmarking 50-photon gaussian boson sampling on the sunway taihulight. arXiv:2009.01177
- Lumino A, Polino E, Rab AS, Milani G, Spagnolo N, Wiebe N, Sciarrino F (2018) Experimental phase estimation enhanced by machine learning. *Phys Rev Appl* 10:044033
- Marković D, Grollier J (2020) Quantum neuromorphic computing. *Appl Phys Lett* 117:150501
- Mangini S, Tacchino F, Gerace D, Bajoni D, Macchiavello C (2021) Quantum computing models for artificial neural networks. arXiv:2102.03879
- Marquardt F (2021) Machine learning and quantum devices. arXiv:2101.01759
- Marcucci G, Pierangeli D, Pinkse PWH, Malik M, Conti C (2020) Programming multi-level quantum gates in disordered computing reservoirs via machine learning. *Opt Express* 28:14018
- Marcucci G, Pierangeli D, Conti C (2020) Theory of neuromorphic computing by waves: machine learning by rogue waves, dispersive shocks, and solitons. *Phys Rev Lett* 125:093901
- Melnikov AA, Sekatski P, Sangouard N (2020) Setting up experimental bell tests with reinforcement learning. *Phys Rev Lett* 125:160401
- Nokkala J, Martínez-Peña R, Giorgi GL, Parigi V, Soriano MC, Zambrini R (2020) Gaussian states provide universal and versatile quantum reservoir computing. arXiv:2006.04821
- Pilozzi L, Farrelly FA, Marcucci G, Conti C (2021) Topological nanophotonics and artificial neural networks. *Nanotechnology* 32:142001
- Quesada N, Arazola JM, Killoran N (2018) Gaussian boson sampling using threshold detectors. *Phys Rev A* 98:062322
- Quesada N, Arazola JM (2020) Exact simulation of gaussian boson sampling in polynomial space and exponential time. *Phys Rev Res* 2:023005
- Silva NA, Ferreira TD, Guerreiro A (2021) Reservoir computing with solitons. *J New Phys* 23:023013
- Sgroi P, Palma GM, Paternostro M (2021) Reinforcement learning approach to nonequilibrium quantum thermodynamics. *Phys Rev Lett* 126:026601
- Spagnolo N, Vitelli C, Bentivegna M, Brod DJ, Crespi A, Flamini F, Giacomini S, Milani G, Ramponi R, Mataloni P, Osellame R, Galvão EF, Sciarrino F (2014) Experimental validation of photonic boson sampling. *Nat Photonics* 8:615
- Spring JB, Metcalf BJ, Humphreys PC, Kolthammer WS, Jin XM, Barbieri M, Datta A, Thomas-Peter N, Langford NK, Kundys D, Gates JC, Smith BJ, Smith PGR, Walmsley IA (2013) Boson sampling on a photonic chip. *Science* 339:798
- Taballione C, Wolterink TAW, Lugani J, Eckstein A, Bell BA, Grootjans R, Visscher I, Geskus D, Roeloffzen CGH, Renema JJ, Walmsley IA, Pinkse PWH, Boller KJ (2019) Reconfigurable quantum photonic processor based on silicon nitride waveguides. *Opt Express* 27:26842
- Tillmann M, Dakić B., Heilmann R, Nolte S, Szameit A, Walther P (2012) Experimental boson sampling. *Nat Photonics* 7:540
- Valencia NH, Goel S, McCutcheon W, Defienne H, Malik M (2020) Unscrambling entanglement through a complex medium. *Nat Phys* 16:1112

- Vicentini F, Biella A, Regnault N, Ciuti C (2019) Variational neural network ansatz for steady states in open quantum systems. *Phys Rev Lett* 122:250503
- Wang H, Qin J, Ding X, Chen MC, Chen S, You X, He YM, Jiang X, You L, Wang Z, Schneider C, Renema JJ, Höfling S, Lu CY, Pan JW (2019) Boson sampling with 20 input photons and a 60-mode interferometer in a  $10^{14}$ -dimensional hilbert space. *Phys Rev Lett* 123:250503
- Wang X, Hiroshima T, Tomita A, Hayashi M (2007) Quantum information with gaussian states. *Phys Rep* 448:1
- Zhao C, Gao X-S (2021) Qdnn: deep neural networks with quantum layers. *Quantum Mach Intell* 3:15
- Zhong HS, Wang H, Deng YH, Chen MC, Peng LC, Luo YH, Qin J, Wu D, Ding X, Hu Y, Hu P, Yang XY, Zhang WJ, Li H, Li Y, Jiang X, Gan L, Yang G, You L, Wang Z, Li L, Liu NL, Lu CY, Pan JW (2020) Quantum computational advantage using photons. *Science* 370:1460
- Zhong H-S et al (2021) arXiv:[2106.15534](https://arxiv.org/abs/2106.15534)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.