Neural Variational Boson Sampling

Shiv Shankar
College of Information and Computer Science
University of Massachusetts
USA

***Abstract*—Boson Samplers are near-term quantum devices based on photonic quantum technology, which can outperform classical computing systems. This paper takes a hybrid circuit learning approach to utilize boson samplers as a generative model called Neural-variational Boson Sampling (NBS). NBS introduces an optimizable parametric structure in the form of a neural network to characterize the characteristic function of a boson sampler and uses this model as a variational ansatz. To simulate working with real quantum devices, we use gradient free-optimization methods to optimize the resultant circuit. We experiment with this framework for problems in optimization and generative modeling.**

Index Terms— hybrid quantum-classical approach, quantum circuit learning, boson sampling, parameterized quantum circuit

I. Introduction

Quantum computing is a computational paradigm that utilizes properties of quantum systems such as superposition and entanglement for computational tasks [Nielsen and Chuang, 2002].

Research in quantum computing has led to the discovery of quantum algorithms that achieve polynomial-time speedups compared to classical methods on specific tasks. For example, [Grover, 1996] described a sub-linear time algorithm for search in an unordered database. Similarly, Shor's algorithm [Shor, 1999] is capable of integer factorization in polynomial-time. Quantum techniques have also been proposed for data-science tasks such as regression fitting [Yu et al., 2019] and association rules mining [Yu et al., 2016].

While the prospect of significant speedup over classical computing remains the most impactful aspect of quantum computing, the algorithms mentioned above remain largely out of the reach of current quantum devices. Hence researchers have explored options that use near-term devices and can achieve quantum supremacy [Arute et al., 2019], [Lund et al., 2017]. Boson sampling (BS) [Aaronson and Arkhipov, 2011] is a strong candidate for experimental demonstration of quantum algorithmic supremacy.

However, while boson sampling (BS) provides an exponential quantum advantage and has been used for applications such as combinatorial optimization [Arrazola and Bromley, 2018], there is some debate on its demonstrable advantage in 'realistic' applications [Bromley et al., 2021], [Oh et al., 2021]. Our goal in this work is to provide examples of how BS can be used in ML applications, specifically as a generative model.

One effective approach for capturing quantum advantage is the so-called hybrid quantum-classical (HQC) approach [McClean et al., 2016]. The HQC approach uses a combination of both quantum and classical resources. HQC based models have been getting recent traction and have been used for applications like supervised regression [Schuld et al., 2020], [Yu et al., 2016], clustering [Otterbach et al., 2017] and combinatorial optimization [Moll et al., 2018].

This work utilizes a similar hybrid circuit approach to use boson samplers for generative models. W use the technique of [Conti, 2021], to parameterize the characteristic funtion of Gaussian states propagating through an interferometer. This problem corresponds to the Gaussian Boson sampling [Hamilton et al., 2017], [Quesada et al., 2018], whose theory relies on phase-space methods [Kruse et al., 2019]. Since computing the exact sampling distribution of a BS is intractable [Aaronson and Arkhipov, 2011] we propose a specific parameterization for the characteristic function of the system using a neural networks. We then train networks following this design for a specific task by stochastic gradient-based minimization of the task loss. We conduct experiments with Ising models and image generation to evaluate the proposed scheme. The results show that this is a promising scheme to deploy boson sampling in practical tasks.

II. Preliminaries and Related Work

A. Quantum Circuit Born Machine

A quantum circuit born machine (QCBM) uses a variational quantum circuit (parameteried by parameters ϕ) to encode and sample from the probability distribution of a classical dataset. Given a set of D independent and identically distributed samples $\{x_1, ..., x_D\}$ from a target probability distribution, the QCBM can be optimized to generate samples that approximate the unknown target probability distribution. A QCBM takes the product state $|0\rangle^{\otimes n}$ and evolves it to the output state $|\psi_\phi\rangle$ by unitary transformation $U(\phi)$. Samples are by measuring the output state $|\psi_\phi\rangle$ in the computational basis which will produce them by the probability $|\langle x|\psi_\phi\rangle|^2$ To train a QCBM, the output samples of the quantum device are evaluated against the desired output (usually on the classical computer). The resultant discrepancy between desired and actual output is then used to adjust the parameters ϕ of the quantum device.

For machine learning applications such as classification, the output model distribution $p_\phi(x)$ is then optimized to minimize the negative log-likelihood of the observed data. The parameters are usually tuned via gradient descent though other methods can also be deployed [Wang et al., 2019].

B. Boson Sampling

The BosonSampling (BS) problem refers to sampling outcomes from a linear optical network. The seminal paper of [Aaronson and Arkhipov, 2011] demonstrates how the simulation of the probability distribution of indistinguishable photons evolving in such a circuit is classically intractable. More specifically [Aaronson and Arkhipov, 2011] define a model where N isolated photons are sent through a m ($m > 2N$) mode linear-optical circuit/interferometer.

In a standard version of this experiment, one considers a many-body squeezed vacuum state propagating in an Haar inteferometer, which distributes the photons in the output modes.

The interferometer is described by a matrix $U \in \mathbb{U}(m)$, which transforms m input modes into m output modes. An example circuit is depicted in Figure 1. Let $\bar{n} = |n_1, n_2, \dots n_m\rangle$ denote the output pattern with n_j photons in output j. The quantum state of the output photons is given by :

$$|\psi\rangle = \gamma_{\bar{n}} |n_1, n_2, \dots n_m\rangle$$
$$\gamma_{\hat{n}} = \frac{\text{Perm}(U_S)}{\sqrt{\hat{n}!}}$$

where $\hat{n}! = n_1! n_2! \dots n_m!$. U_S is the submatrix of U obtained by selecting the columns corresponding to input photons and rows corresponding to output photons. The probability of a certain output is given by

$$Pr(\bar{n}) = \frac{|\text{Perm}(U_S)|^2}{\bar{n}!}$$

Perm here refers to the permanent of the matrix. Computation of the permanent is #P-complete [Valiant, 1979], which makes exact sampling from such a circuit intractable. Further work [Morimae et al., 1998], [Gogolin et al., 2013], [Bremner et al., 2011] provided further connections between boson sampling and the polynomial hierarchy. Since a deterministic source of single photons as described in [Aaronson and Arkhipov, 2011] is physically challenging; variants such as Lattice walk sampling (LWS) [Muraleedharan et al., 2019] and Gaussian boson sampling (GBS) [Lund et al., 2014], [Hamilton et al., 2017] have been proposed. Recently, a quantum computing machine that uses Gaussian boson sampling was used to demonstrate quantum supremacy and was faster than the state-of-the-art classical supercomputers by a factor of 10^{14} [Zhong et al., 2020].

Recent research has also explored other applications of BS to solve diverse problems. [Guerreschi, 2015] demonstrated that molecular vibronic spectra could be efficiently generated using boson sampling. [Arrazola and Bromley, 2018] showed that boson sampling could be utilized for approximating the densest k-subgraph problem. Recently [Huang et al., 2019] developed a quantum symmetric encryption scheme built on boson sampling. A quantum signature protocol using BS-based unitary operation [Gao et al., 2018] has been demonstrated by [Feng et al., 2020]. Bosonic techniques have also been successfully used for graph similarity-based tasks in machine learning [Schuld et al., 2020], [Shankar and Towsley, 2020].

The works most related to our article are those of [Banchi et al., 2020] and [Shankar and Towsley, 2022]. [Banchi et al., 2020] show how under certain parameterizations, the unbiased estimates of the gradients of the parameters for a GBS device can be obtained directly via measurements on the same device. [Shankar and Towsley, 2022] show how parameterized boson samplers can be used for some non-combinatorial tasks in machine learning. Our approach is also similar in spirit to the variational learning of a quantum Born machine [Liu and Wang, 2018]. The major differences from [Banchi et al., 2020] is that a) we use a universal parameterization instead of a restricted one and b) we explore more generic applications. Our experiments focus on the tasks explored in [Shankar and Towsley, 2022] but our proposal is distinct in that instead of arbitrary parametric circuits used in their work, we use GBS [Lund et al., 2014] which is easier to realize experimentally. Furthermore our proposal used the neural phase space parameterization developed by [Conti, 2021] and finds it to be more effective at solving machine-learning tasks.

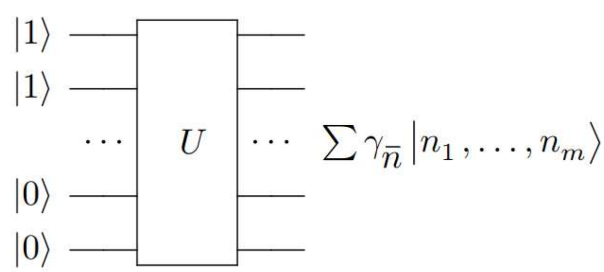


Fig. 1: Basic structure of a boson sampling scheme with evolution matrix U

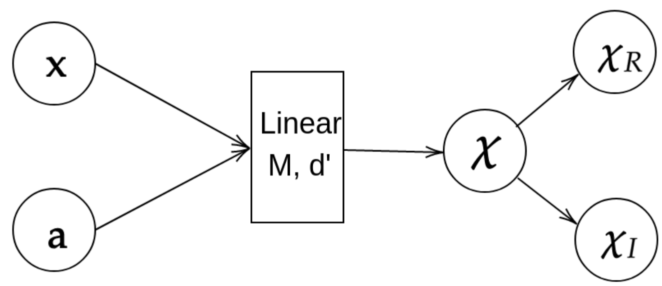


Fig. 2: A two layer neural network as representing a state with characteristic functionχ, subject to a unitary transformation. This is a pullback of a linear transform from the original state, which produces a new state with characteristic function $\tilde{\chi}$

Next, we briefly present the phase space based NBS parameterization that is both flexible and easily trainable. This design is based on the recent work of [Conti, 2021] that can be efficiently simulated via tensor networks.

III. Neural Boson Sampling Circuits

The state of an N-body system is represented in phase space by the characteristic function $\chi(\mathbf{x}) = \chi_R(\mathbf{x}) + \imath\chi_I(\mathbf{x})$ dependent on the vector $\mathbf{x}$ [Gardiner et al., 2004], [Barnett and Radmore, 2002]. For Gaussian states the characteristic function is given by [Wang et al., 2007]

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

with $\mathbf{g}$ the real covariance $N \times N$ matrix, and $\mathbf{d}$ the real displacement $N \times 1$ vector. These vectors are related to the canonical variables $\hat{q}_j = \hat{R}_{2j}$ and $\hat{p}_j = \hat{R}_{2j+1}$ of the system by the following relation [Wang et al., 2007]:

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

and

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

being $\mathbf{J} = \bigoplus_{j=0}^{n-1} \mathbf{J}_1$, $\mathbf{J}_1 = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ [Wang et al., 2007]. In Eq. (2), the canonical variables are assumed to be arranged linearly in an $N \times 1$ operator array $\hat{\mathbf{R}}$.

The vacuum state is characterized by $\mathbf{g} = \mathbf{1}$ and $\mathbf{d} = \mathbf{0}$. Addition and removal of bosons in the vacuum state is achieved by the displacement or squeezing operators which are both unitary. Under the action of these operators the canonical variables transform as $\tilde{\hat{\mathbf{R}}} = \mathbf{M}\,\hat{\mathbf{R}} + \mathbf{d}'$, where the symplectic matrix $\mathbf{M}$ and the vector $\mathbf{d}'$ depend on the specific operator (detailed, e.g., in [Wang et al., 2007]). The corresponding transformation of the characteristic function can be written as

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

[Conti, 2021] demonstrate how this characteristic function can be represented via a neural network with two inputs $\mathbf{x}$ and $\mathbf{a}$. An example is depicted in Figure 2c shows $\tilde{\chi}$. Note that the first layer incorporates the linear transformation of the state by a unitary operator corresponding to the symplectic matrix M and the displacement vector d'. With such a representation layers can be cascaded, and one can represent single mode squeezers, interferometers, and other unitary operators.

A. Observables

Observables in phase space are given by differential operators of the characteristic function. In the aforementioned scheme these correspond to the the derivatives of the corresponding parameterized neural network. For example, the mean photon number per mode is [Conti, 2021]:

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

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

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

B. Boson sampling with Neural Networks

For modelling boson sampling through characeteristic function neural networks, one needs to represent squeezing and interferometer layers in the neural network representation. The squeezing layers can be realized through symplectic matrices $\mathbf{M}$ with $\mathbf{d} = 0$; while interferometers can be implemented via Haar matrix operators. For more details on this schema we refer the readers to [Conti, 2021].

Let $\bar{n} = |n_1, n_2, \ldots n_m\rangle$ be a given photon pattern. Sampling this state, is equivalent to fining To simulate sampling, one can compute the probability $\Pr(\bar{n})$ of finding n_1 photons in mode 1, n_2 photons in mode 2, and so on. Letting $\hat{\rho}$ the density matrix, one has

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

with

$$|\bar{n}\rangle\langle\bar{n}| = \otimes_{j=1}^m |n_j\rangle\langle n_j|.$$

[Kruse et al., 2019], show that this probability is given by

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

where $\hat{n}! = \bar{n_0}!\bar{n}_1!\ldots\bar{n}_{n-1}!$ and

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

is the Q-representation of the density matrix [Gardiner et al., 2004], [Barnett and Radmore, 2002] with $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ as complex displacement vectors.

One can reparameterize the earlier expression by introducing the $\mathbf{k}$ vector which extracts the real and imaginary parts of the displacement vector as follows.

$$k_{2j} = \frac{\alpha_j^* + \alpha_j}{\sqrt{2}}$$
$$k_{2j+1} = \frac{\alpha_j^* - \alpha_j^*}{\sqrt{2}\imath}$$

Then the earlier expression can be rewritten as

$$\Pr(\bar{n}) = \frac{1}{\hat{n}!2^{\hat{n}_T}}\left.\left(\prod_j \tilde{\nabla}_j^{2n_j}\right) e^{\frac{\mathbf{k}^2}{2}} Q_\rho(\mathbf{k})\right|_{\mathbf{k}=0} \tag{8}$$

with $\tilde{\nabla}_j^2 = \partial^2/\partial k_{2j} + \partial^2/\partial k_{2j+1}$ and $\hat{n}_T = \sum_{j=0}^m n_j$. Q_ρ in Eq. (8) can be evaluated explicitly as a multidimensional Gaussian integral:

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

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

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

being $A_{pq} = \frac{1}{2}(g_{pq} + \delta_{pq})$. Eq. (9) and (10) can be implemented as further layers of a neural net, and the probability of a given pattern computed by running the model.

IV. EXPERIMENTS

Next, we try the aforementioned model for two tasks. The first task is to identify the ground states of an Ising Hamiltonian. Our experiments show that the NBS model can be trained to preferentially sample low-energy states from an Ising model. For the second task, we train a classically augmented NBS on the digits dataset [Alpaydin and Kaynak, 1998] to generate similar images. The results show that for comparable latent dimensions the NBS scheme is as expressive as Variational Autoencoders [Kingma and Welling, 2019].

A. Ising Model Optimization

An Ising model is essentially an energy model (or an unnormalized distribution) for which the score function (or the log-likelihood) is of the following form:

$$H(\bar{x}) = -\sum_i h_i x_i - \sum_{ij} J_{ij} x_i x_j, \qquad (11)$$

where $\bar{x} = (x_1, x_2, \ldots, x_m)$ and $x_k = 0, 1$ i.e. $\bar{x}$ is a binary vector. We are interested in finding a model distribution that samples the state with the lowest energy (also known as the ground state) with high probability. This is a challenging task, as finding such a state of a general $H(\bar{x})$ is NP-hard [Lucas, 2014]. While similar to sampling from an energy model, this task is closer to optimization as we wish to find the lowest energy state. In this experiment the NBS parameters (Ω) are updated so that it samples the minimum energy configuration with high probability.

We follow the procedure of [Banchi et al., 2020] and use our NBS model for predicting cliques. Since a boson sampling scheme produces an output with an integer number of bosons in different modes, the output of such a scheme can be thresholded to a vector $\bar{x}$ of binary variables for input to the Ising Hamiltonian. The training loss is given by:

$$\mathcal{L}(W) = \mathbb{E}_{\bar{x} \sim P_{\text{NBS}}(.|\Omega)} \left[H(\bar{x}) \right] \equiv \sum_{\bar{x}} H(\bar{x}) P_{\text{NBS}}(\bar{x}|\Omega) \qquad (12)$$

where $P_{\text{NBS}}(\bar{x}|\Omega)$ is the distribution of Equation (**??**).

Similar to [Banchi et al., 2020], we focus on the following Ising hamiltonian. Given a graph $G = (V, E)$ with vertex set V and edge set E and an integer K:

$$H_K(\bar{x}) = \lambda \left(K - \sum_{v \in V} x_v \right)^2 - \sum_{(u,v) \in E} x_u x_v \qquad (13)$$

where λ is a positive number and x_v are binary variables. It is easy to prove [Lucas, 2014] that for $\lambda > K$ the above Hamiltonian has ground state energy $E = -\frac{K(K-1)}{2}$ if and only if there is a clique of size K in the graph G. To see this note that the second term $x_u x_v$ computes the number of edges between the set of nodes corresponding to the binary vector x and for a clique of size K will contribute $\frac{K(K-1)}{2}$. On the other hand the first term tries to keep the number of selected nodes to K. With a large enough λ, If the number of selected nodes becomes more than K, then the increase in the first term is enough to compensate for the reduction in the second term.

Following the procedure of [Banchi et al., 2020] we use sampling to produce binary strings that corresponds to the ground state of the aforementioned Ising Hamiltonian. For each graph we set K as the size of the largest clique in the graph. Furthermore we set $\lambda = 2\Delta$ where Δ is the max degree of the graph. The success rate is estimated as the fraction of times that the correct bit pattern is sampled by the model in a 1000 samples, conditional on observing K output particles. Training is done using an estimation of the gradient using the REINFORCE algorithm [Williams, 1992], obtained with 200 samples per iteration.

We run NBS on the hamiltonian corresponding to the graphs experimented on by [Banchi et al., 2020]. These experiment are on the specific graphs depicted in 3 followed by a bunch of random graphs from the Erdos-Renyi and Barabasi Albert families (Figure 6). Figure 3(a) presents the training curve on a simple graph on 8 nodes with a clique size of 5. From the figure, it is clear that while the initial probability of sampling the ground state is low; it steadily increases as training progresses and is above 80% by the end. In Figure 3(b), a more challenging case with a degenerate ground model is presented. The underlying graph has ten nodes and two max-cliques of size $K = 5$. One can observe from the charts for both models NBS is able to outperform the trainable GBS approach of [Banchi et al., 2020]. The added variability is primarily due to sampling at each step. The figures also present the sampling variation (p=0.1) during different trials in the run. It is clear that the GBS curve is statistically better than the NBS curve.

Next, the experiment is repeated with the aforementioned families of random graphs. These results are presented in the Appendix (Figure 6). The first row presents results on instances of random Barabasi-Albert graphs. These graphs have many cliques of sizes three and four, leading to multiple local optimas. The second row illustrates the result of training a NBS model on random Erdos-Renyi graphs with ten vertices. We can observe that both GBS and NBS can with high probability ($> 80 - 90\%$) sample the energy minimum. However it is also clear that the NBS trained model can sample the configuration corresponding to largest clique in the graph with a higher success rate than the GBS approach. The performance curves also make it clear from these results that the NBS behaviour performance is fundamentally distinct from the GBS one (e.g. see subfigures 2,5).

B. Generative Modelling/Image Generation

Next we use the NBS scheme to learn a simple generative model. For this experiment, we used the test set of the UCI digits dataset [Alpaydin and Kaynak, 1998][1].

The UCI DIGITS dataset consists of 1797 data observations. Each observation is an 8×8 image of a handwritten digit. Since the simulation of a 64 photons BS scheme is not feasible on classical machines, we use a low dimensional embedding approach. The NBS model sample vectors from

[1] Available at https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+

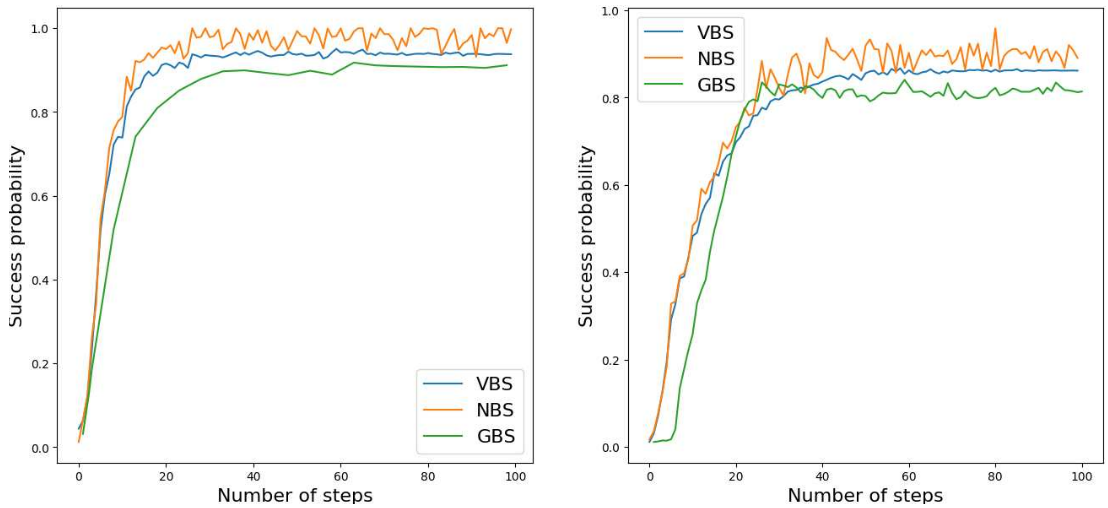


Fig. 3: Success rate of sampling the ground state of the Ising model over time for NBS and GBS. A max clique of size $K = 5$ is shown in red. In (a) there is a single max-clique, while in (b) there are two max-cliques. Training is done with 200 samples per iteration.

a 5-dimensional latent space which then constructs the images through a classical conditional generator. The complete generative model is depicted in Figure 5. In the figure, Y is a random variable that denotes the training images, Z is a latent variable, and P_Ω is the distribution of Z parameterized by Ω. The parameters Ψ are the parameters of the conditional generator. z and y are used to denote samples of the variables Z and Y respectively. We follow the same experiment design on 5000 randomly selected samples from MNIST.

In our experiments, we model the conditional distribution of the output Y given the latent z as a Gaussian variable. The distribution of the latent variable z is given by the output distribution of the Boson sampler. The combined log-likelihood of an observation y is then given by:

$$L(y; W, b, \Omega) = \int N(y; W\bar{z} + b, \Sigma) P_{\text{NBS}}(z; \Omega) dz$$

Here Ω refers to the parameters of the NBS scheme (i.e $\Omega = \{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}\}$ P_{NBS} is the induced distribution over z by Ω. $\bar{z}$ refers to the normalized value of z. Since z itself is a discrete distribution with the number of bosons in each mode being a non-negative integer, we scale the output by its norm to make it approximately continuous. This, in turn, determines the mean vector of the Gaussian distribution N. Σ is assumed to be a diagonal matrix, and W, b are parameters learned on a classical device. Hence in this case $\Psi = \{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma}\}$ We assume that all observed images are independent draws from the generating distribution. The total likelihood of the data is then just the product of likelihoods for each observation.

We train the model to maximize the log-likelihood of the data with a variant of the EM algorithm [Dempster et al., 1977]. In the EM algorithm, each iteration consists of repeated application of the E-step and the M-step. In the E-step, the data log-likelihood conditioned on the observed variables is computed. On the other hand, in the M-step, the likelihood obtained in the E-step is maximized with respect to the model parameters. The Monte-Carlo Expectation Maximization (MCEM) algorithm [Wei and Tanner, 1990] is a variant of the classic EM; often used for high-dimensional data or when the integral required in the E-step is intractable. The key difference between the two is that the MCEM uses a Monte-Carlo approximation to the conditional expectation during the E-step. Since computing the exact output distribution of the BS scheme is generally intractable, while quantum devices can sample from it easily, MCEM is a better choice for training such models. Specifically the Q function for the MCEM algorithm in our case is given by:

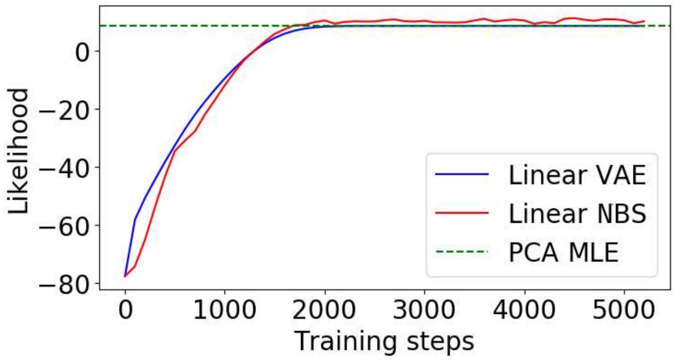
$$Q(\Omega_t|\Omega_{t-1}, y) = \mathbb{E}_{z\sim P_{\text{NBS}}(\Omega_{t-1})} \log N(y; W\bar{z} + b, \Sigma) \quad (14)$$

$$\approx \sum_{z_i \sim P_{\text{NBS}}(\Omega_{t-1})} \log N(y; W\bar{z}_i + b, \Sigma) \quad (15)$$

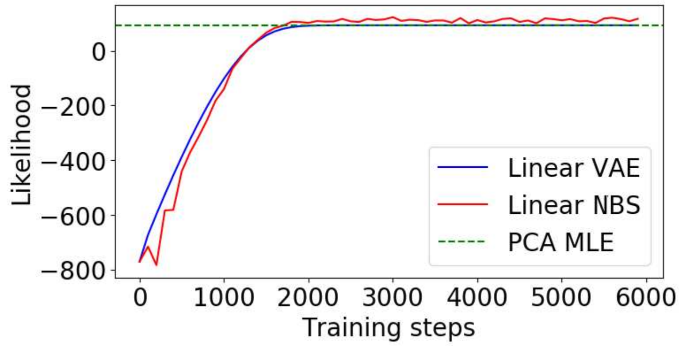
where the sum is over samples z_i drawn from P_{NBS}. This Q function is then optimized by gradient descent to estimate Ω. Similar to REINFORCE, monte-carlo EM adds extra variability and generally requires lower learning rate for smooth learning. However these problems are not as significant for these experiments as the likelihood loss is dominated by the decoder terms instead of the prior.

We also compare our results against a VAE [Kingma and Welling, 2019] with the same sized latent space and a linear decoder like in the NBS model. A linear decoder was chosen as such a model with Gaussian prior provides an exactly computable likelihood. One can use more complex decoder models for better sample quality, but our goal in this work is simply a working BS based generative model. Both models were trained with Adam optimizer with a learning rate of 5e-4.

Note that under a Gaussian prior the given generative model corresponds exactly to the PCA decomposition of the data, which can be analytically computed. We present this



(a)



(b)

Fig. 4: Test likelihood across different training iterations trained generative model described in Section 4.2 with 5 dimensional latent space. Left (a) presents the number on DIGITS dataset, while (b) shows results on a subset of MNIST

exact likelihood (pPCA MLE value) in Figure 4. Furthermore given the low latent dimensionality, a generic prior and the significantly higher contribution from the conditional model, the learnt NBS based distribution in this case is expected to be similar to the PCA decomposition. This can be seen from Figure 4, where we plot the data log-likelihood as learnt by the PCA, VAE and NBS models. We can also observe that the NBS model improves the likelihood by around 3%

The samples from both VAE and NBS runs are presented in Figure **??**. The upper rows present samples from the NBS model trained with MCEM. On the lower rows we have samples from the baseline linear VAE. It is evident from the image quality that a 5-dimensional latent space is too small. However this is primarily a function of the decoder. For example, MNIST has high quality reconstructions from a two dimensional latent space with a non-linear decoder [2]. On the other hand a linear decoder with (a gaussian conditional generator) is equivalent to PCA and produces similarly diffused images. However, even with the blurred images, the digit-like structure of these sampler is clear. A qualitative examination shows that both the NBS and the VAE models produces samples of a similar nature. Combined with the likelihood results, this experiment provides evidence that the NBS model is at least as powerful as a VAE in this setting.

V. Conclusion

In this paper, we demonstrate that boson sampling can be used to solve practical problems in machine learning and optimization. Towards this goal, we developed a hybrid quantum-classical variational scheme labeled Neural Boson Sampling (NBS). NBS introduces an optimizable parametric structure into the boson evolution operation and uses that as a variational ansatz. We then experiment with this framework for training NBS distributions for problems in ising optimization and generative modelling.

For optimization, we tried an algorithm where the NBS is used to generate samples that can be mapped to the states of an Ising model. We then use REINFORCE [Williams, 1992] to get stochastic gradients of the parameters device in order to maximize the probability of sampling the ground state of the Ising model. In generative modelling, we show that a NBS-based scheme trained using an EM algorithm is competitive with a VAE [Kingma and Welling, 2019] of similar capacity. While sampling based methods do add extra variability in learning, we do not believe this to be a major issue as physical implementation of a NBS is extremely time-efficient in producing samples.

Our results have shown that the NBS scheme can be used to implement algorithms for practical problems; we hope this sparks more research into variational boson sampling ansatz in the future. One future research direction is formalizing cases when a NBS-based scheme outperforms alternative algorithms such as VQE [Wang et al., 2019] or QAOA [Farhi et al., 2014] for standard qubit based devices. Another potential research direction is to develop schemes to approximate the gradients of the scheme using a quantum device. Finally recent work [Ostaszewski et al., 2021] has used deep Q-learning to optimize larger quantum circuits, and using such techniques can be of potential use in improving training for boson sampling as well.

[2] https://github.com/lttsh/VariationalAutoEncoder-MNIST

References

[Aaronson and Arkhipov, 2011] Aaronson, S. and Arkhipov, A. (2011). The computational complexity of linear optics. In *43rd Annual ACM Symp. Theory of Computing*, pages 333–342.

[Alpaydin and Kaynak, 1998] Alpaydin, E. and Kaynak, C. (1998). Cascaded classifiers. *Kybernetika*, 34:369–374.

[Arrazola and Bromley, 2018] Arrazola, J. M. and Bromley, T. R. (2018). Using gaussian boson sampling to find dense subgraphs. *Physical review letters*, 121(3):030503.

[Arute et al., 2019] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G., Buell, D. A., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.

[Banchi et al., 2020] Banchi, L., Quesada, N., and Arrazola, J. M. (2020). Training gaussian boson sampling distributions. *Physical Review A*, 102(1):012417.

[Barnett and Radmore, 2002] Barnett, S. and Radmore, P. M. (2002). *Methods in theoretical quantum optics*, volume 15. Oxford University Press.

[Bremner et al., 2011] Bremner, M. J., Jozsa, R., and Shepherd, D. J. (2011). Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472.

[Bromley et al., 2021] Bromley et al. (2021). Applications of near-term photonic quantum computers. *Quantum Science and Technology*.

[Conti, 2021] Conti, C. (2021). Training gaussian boson sampling by quantum machine learning. *Quantum Machine Intelligence*, 3(2):1–8.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Farhi et al., 2014] Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

[Feng et al., 2020] Feng, Y., Shi, R., Shi, J., Zhao, W., Lu, Y., and Tang, Y. (2020). Arbitrated quantum signature protocol with boson sampling-based random unitary encryption. *Journal of Physics A: Mathematical and Theoretical*, 53(13):135301.

[Gao et al., 2018] Gao, Z. J., Pansare, N., and Jermaine, C. (2018). Declarative parameterizations of user-defined functions for large-scale machine learning and optimization. *IEEE Transactions on Knowledge and Data Engineering*, 31(11):2079–2092.

[Gardiner et al., 2004] Gardiner, C., Zoller, P., and Zoller, P. (2004). *Quantum noise: a handbook of Markovian and non-Markovian quantum stochastic methods with applications to quantum optics*. Springer Science & Business Media.

[Gogolin et al., 2013] Gogolin, C., Kliesch, M., Aolita, L., and Eisert, J. (2013). Boson-sampling in the light of sample complexity. *arXiv preprint arXiv:1306.3995*.

[Grover, 1996] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219.

[Guerreschi, 2015] Guerreschi, G. G. (2015). Boson sampling for molecular vibronic spectra. *Bulletin of the American Physical Society*, 60.

[Hamilton et al., 2017] Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C., and Jex, I. (2017). Gaussian boson sampling. In *2017 Conference on Lasers and Electro-Optics (CLEO)*.

[Huang et al., 2019] Huang, Z., Rohde, P. P., Berry, D. W., Kok, P., Dowling, J. P., and Lupo, C. (2019). Boson sampling private-key quantum cryptography.

[Kingma and Welling, 2019] Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

[Kruse et al., 2019] Kruse, R., Hamilton, C. S., Sansoni, L., Barkhofen, S., Silberhorn, C., and Jex, I. (2019). Detailed study of gaussian boson sampling. *Physical Review A*, 100(3):032326.

[Liu and Wang, 2018] Liu, J.-G. and Wang, L. (2018). Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324.

[Lucas, 2014] Lucas, A. (2014). Ising formulations of many np problems. *Frontiers in Physics*, 2.

[Lund et al., 2017] Lund, A. P., Bremner, M. J., and Ralph, T. C. (2017). Quantum sampling problems, bosonsampling and quantum supremacy. *npj Quantum Information*, 3(1).

[Lund et al., 2014] Lund, A. P., Laing, A., Rahimi-Keshari, S., Rudolph, T., O'Brien, J. L., and Ralph, T. C. (2014). Boson sampling from a gaussian state. *Physical review letters*, 113(10):100502.

[McClean et al., 2016] McClean, J. R., Romero, J., Babbush, R., and Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023.

[Moll et al., 2018] Moll, N., Barkoutsos, P., Bishop, L. S., Chow, J. M., Cross, A., Egger, D. J., Filipp, S., Fuhrer, A., Gambetta, J. M., Ganzhorn, M., et al. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503.

[Morimae et al., 1998] Morimae, T., Nishimura, H., Fujii, K., and Tamate, S. (1998). Classical simulation of dqc12 or dqc21 implies collapse of the polynomial hierarchy. *Phys. Rev. Lett*, 81:5672.

[Muraleedharan et al., 2019] Muraleedharan, G., Miyake, A., and Deutsch, I. H. (2019). Quantum computational supremacy in the sampling of bosonic random walkers on a one-dimensional lattice. *New Journal of Physics*, 21(5):055003.

[Nielsen and Chuang, 2002] Nielsen, M. A. and Chuang, I. (2002). Quantum computation and quantum information.

[Oh et al., 2021] Oh, C. et al. (2021). Classical simulation of lossy boson sampling using matrix product operators. *Physical Review A*.

[Ostaszewski et al., 2021] Ostaszewski, M., Trenkwalder, L. M., Masarczyk, W., Scerri, E., and Dunjko, V. (2021). Reinforcement learning for optimization of variational quantum circuit architectures. *arXiv preprint arXiv:2103.16089*.

[Otterbach et al., 2017] Otterbach, J., Manenti, R., Alidoust, N., Bestwick, A., Block, M., Bloom, B., Caldwell, S., Didier, N., Fried, E. S., Hong, S., et al. (2017). Unsupervised machine learning on a hybrid quantum computer. *arXiv preprint arXiv:1712.05771*.

[Quesada et al., 2018] Quesada, N., Arrazola, J. M., and Killoran, N. (2018). Gaussian boson sampling using threshold detectors. *Physical Review A*, 98(6):062322.

[Schuld et al., 2020] Schuld, M., Brádler, K., Israel, R., Su, D., and Gupt, B. (2020). Measuring the similarity of graphs with a gaussian boson sampler. *Physical Review A*, 101(3).

[Shankar and Towsley, 2020] Shankar, S. and Towsley, D. (2020). Bosonic random walk networks for graph learning. *Quantum Tensor Networks in Machine Learning*.

[Shankar and Towsley, 2022] Shankar, S. and Towsley, D. (2022). Variational boson sampling. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.

[Shor, 1999] Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

[Valiant, 1979] Valiant, L. (1979). The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201.

[Wang et al., 2019] Wang, D., Higgott, O., and Brierley, S. (2019). Accelerated variational quantum eigensolver. *Physical review letters*, 122(14):140504.

[Wang et al., 2007] Wang, X.-B., Hiroshima, T., Tomita, A., and Hayashi, M. (2007). Quantum information with gaussian states. *Physics reports*, 448(1-4):1–111.

[Wei and Tanner, 1990] Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

[Williams, 1992] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

[Yu et al., 2016] Yu, C.-H., Gao, F., Wang, Q.-L., and Wen, Q.-Y. (2016). Quantum algorithm for association rules mining. *Physical Review A*, 94(4):042311.

[Yu et al., 2019] Yu, C.-H., Gao, F., and Wen, Q. (2019). An improved quantum algorithm for ridge regression. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhong et al., 2020] Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., et al. (2020). Quantum computational advantage using photons. *Science*, 370(6523):1460–1463.

APPENDIX

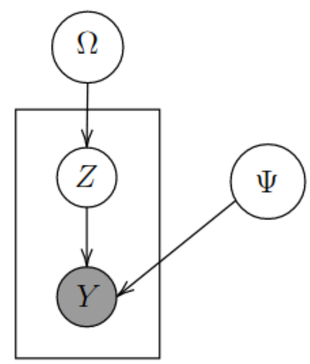


Fig. 5: Generative Model for Digit images

A. Clique Optimization

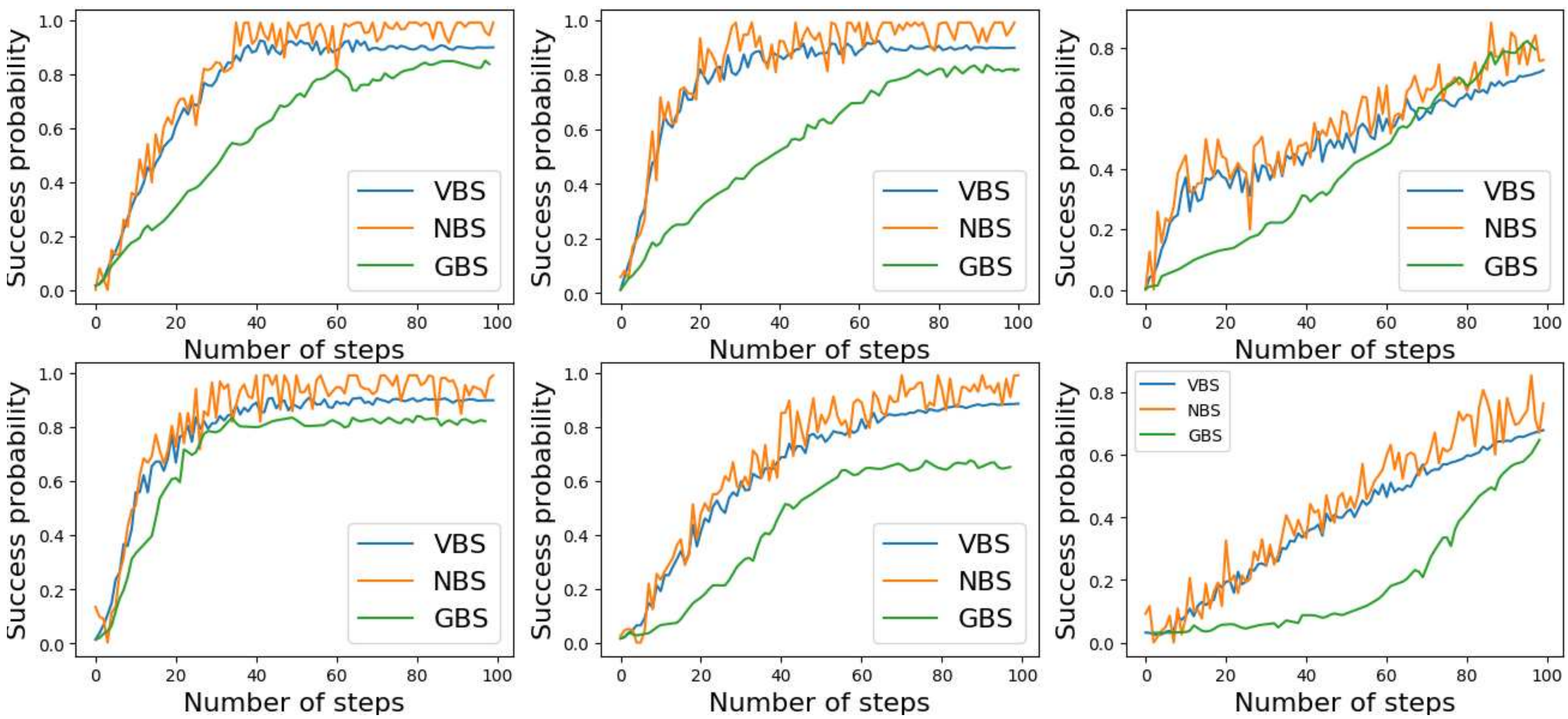


Fig. 6: Success rate over time as the training progresses, as in Figure 3, Graphs (a),(b),(c) are random Barabasi-Albert graphs with ten vertices, built starting from a clique of five vertices and attaching new vertices, each connected to three random nodes. Graphs (d),(e),(f) are random Erdos-Renyi graphs with ten vertices and probability $p = 0.5$ of adding an edge between pairs of vertices.