University of Barcelona
Master in Quantum Science and Technology

Master's thesis

Continuous Variables Quantum Neural Networks

Todor Krasimirov Ivanov
Supervisors: Antonio Acín, Federico Centrone
Course 2022–2023

Continuous variables in Quantum Neural Networks

Todor Krasimirov Ivanov

Supervised by: Antonio Acín, Federico Centrone

ICFO, 08860 Castelldefels, Barcelona

10 July 2023

Quantum mechanics in conjunction with continuous variables systems are promising for artificial neural networks in terms of information processing and real data domains. For classical simulation concerns, a brand-new continuous variables quantum neural network (QNN) was implemented, analyzed and tested keeping the affinity with quantum hardware. The simulated QNN approach is based on second statistical moments of the system's Gaussian stage jointly with expectation value expressions acting as the non-Gaussian evolution and measurement of the system. The required components, infrastructure and algorithms were described and developed along with its relative demonstrations giving a complete traceability of the proposed quantum neural network model supported with illustrative examples.

Keywords: Neural network, continuous variables, quantum optics, Gaussian states, non-Gaussianity, position, momentum, canonical basis, Fock basis, symplectic, orthogonal, squeezing, passive optics, covariance matrix, uncertainty principle

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor, Dr. Federico Centrone, for his invaluable guidance and support throughout the research and writing process. His extensive knowledge and expertise in the fields of quantum mechanics and quantum information theory have been instrumental in shaping the direction and outcomes of this thesis. I am truly grateful for his mentorship and the countless hours he dedicated to providing insightful feedback and suggestions.

I would also like to extend my appreciation to Dr. Antonio Acín, Dr. Paolo Stornati, Ulysse Chabaud and, generally, all ICFO Quantum Information group. Their expertise and willingness to share their knowledge have been instrumental in broadening my understanding of this complex and emerging field.

I would also like to thank my fellow Master's classmates for their support and assistance in navigating the intricacies of Quantum Physics concepts. Their willingness to collaborate, share resources, and engage in thoughtful discussions has been immensely beneficial for a stimulating academic environment.

I am truly humbled and honored to have had the opportunity to work with such talented people. Their guidance and contributions have undoubtedly shaped the outcome of this thesis, and for that, I am sincerely grateful.

Todor Krasimirov Ivanov: todorkriv@gmail.com

Contents

1 Introduction

Quantum neural networks are a cutting-edge field gaining more and more relevance due to its unstoppable recent progress and inspiring applications in different real-life areas [Flö23], [JC19]. This area keeps growing day by day with new interested members giving new approaches because it blends up two of the most groundbreaking realms of this century, artificial intelligence and quantum mechanics with respect to which many other fields arise like quantum computation or quantum machine learning.

2 Motivation

The high-performance computing capacity of quantum mechanics is a theoretical milestone. Hence, the principal motivation of this project is to tackle one of the most calculation-demanding processes in artificial intelligence: artificial neural networks training.

Another inspiration is to model a physically-realizable quantum neural network structure and procedure in a continuous-variable system as is quantum optics [SvdMA+21] aiming to speed up the learning process using an optical quantum computer and increase the accuracy of the QNN trained model.

Furthermore, this promising area as continuous variables on QNN open many doors to inventive paradigms of learning and new perspectives addressing practical tasks challenges.

3 Objective

The main goal of this project is to investigate and implement an innovative technique for classical simulation of continuous variables Quantum Neural Networks based on the system variables' covariances for their Gaussian states and . This involves the development of a classical program that mimics the physical functionality of a CV QNN and optimizes its free parameters by simulating a certain quantum system, the transformations modifying its state and the measurements over the system.

Through the simulation of the proposed model, the point is to experiment with the QNN learning capacity and research about the different parameters and components behavior, in terms of training and prediction performance.

Moreover, this method is hardware-inspired aiming to execute the QNN on continuous quantum systems like optical quantum computers where the components used in the simulation are easily translated and reproduced with real devices.

4 Foundations

The next two sections gather the basic notions for a complete understanding on the thesis topic "Continuous variables in quantum neural networks". The discussed fundamental concepts are the following: the structure, learning and prediction mechanisms of artificial neural networks, an introduction to quantum continuous variables, two crucial basis for the treatment of quantum continuous variables, Gaussianity and non-Gaussianity in quantum states, some mathematical classical groups and the statistical moments of quantum states.

5 Artificial Neural Networks

Nowadays, there are multiple approaches in the field of artificial intelligence and one of the most powerful is the artificial neural network used in machine learning aiming to computationally simulate a neural network able to acquire knowledge and make predictions.

Generally, there exist three main categories of machine learning [Meh21]. First is unsupervised learning, where artificial intelligence learns patterns without requiring feedback on question outcomes. Second is supervised learning, which builds knowledge and generalizes based on data sets that record results under specific conditions. Lastly, there is reinforcement learning, which generates knowledge by observing the environment or situation of a problem. It takes actions that modify its state and assesses whether the new state brings it closer to the desired solution, thereby assigning a rating to each action in each environment state.

Artificial neural networks are made of artificial neurons that are interconnected through edges or dendrites. The neurons are typically clustered in layers where the input data of one layer is the output of the previous one. In each neuron, some kind of parameterized function is applied to the input data to produce an output that flows towards the neural network in order to produce a prediction. When there are more than one layer in a neural network, i.e. a neural network with at least one hidden layer of neurons, the artificial neural network are said to represent deep learning.

For the purpose of the project, the main feature to keep in mind is the mathematical transformation that actually happens when giving an input to a single-layer artificial neural network, expressed by

$$f(\vec{x}) = \Phi(W\vec{x} + \vec{b}), \tag{1}$$

where $f(\vec{x})$ would represent the output of the neural network, W is the weight matrix that transforms the input vector $\vec{x}$ in a linear way, $\vec{b}$ is the real vector of the bias and Φ is a non-linear activation function allowing to learn non-linearities of the target function.

Having said that, the purpose of an artificial neural network is to tune the parameters of the weight matrix W and $\vec{b}$ components in order to approximate its output $f(x)$ to a given target function represented.

The project is centered in quantum neural networks with continuous variables, whose mechanisms are based on those of classical neural networks. Therefore, the study of classical neural network features is the key to properly replicate this artificial learning model taking advantage of quantum mechanics [HSM+22].

6 Quantum continuous variables

The degrees of freedom of a physical system can be continuous or discrete. While discrete variables are those referred to physical quantities whose possible values are restricted by a numerable set of fixed quantities, i.e. the set is finite or countably infinite, continuous variables can take any possible value existing in a continuous range.

In quantum mechanics, all observable physical quantities are mapped to an operator and one can find both types: discrete, like the spin of a particle that can take a finite number of classical values, and continuous, like the position or momentum of a particle that can take any possible classical value in a continuous range of real numbers.

6.1 Canonical operators

In the CV formalism, information is encoded in the quadratures, denoted as $\hat{x}$ and $\hat{p}$, which are referred to as position and momentum in relation to the harmonic oscillator terminology. Throughout the thesis, the focus is centered into these operators which are a pair of canonical operators and they exhibit some properties [Ser17] like the canonical commutation relation (CCR)

$$[\hat{x}, \hat{p}] = i\hbar, \tag{2}$$

where i is the complex unit and $\hbar$ is the reduced Planck constant or the quantum elementary action, often taken as 1.Specifically, when treating with more than one canonical degree of freedom or mode, this is, more than one particle in the considered system, the CCR is generalized as

$$[\hat{x}_j, \hat{p}_k] = i\hbar\delta_{j,k}, \tag{3}$$

where j and k point out the mode or degree of freedom each operator refers to, which means that position and momentum operators commute whenever they act over a different mode due to the delta function $\delta_{j,k}$.

To simplify the joint representation of N modes with their canonical operators, the so-called xxpp representation is adopted, which orders all modes' position operators at first place followed by their respective momentum operators. Then, the vector $\hat{\mathbf{s}}$ related to this representation is defined as

$$\hat{\mathbf{s}} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_N, \hat{p}_1, \hat{p}_2, ..., \hat{p}_N)^T \tag{4}$$

Now, recasting the canonical commutation relation to the N-mode vector $\hat{\mathbf{s}}$, the generalized CCR has the form

$$[\hat{\mathbf{s}}, \hat{\mathbf{s}}^T] = iJ \text{ with } J = \begin{pmatrix} \mathbb{0}_N & \mathbb{1}_N \\ -\mathbb{1}_N & \mathbb{0}_N \end{pmatrix} \tag{5}$$

where the commutator row-column products shall be performed as outer products and the matrix J is the real canonical anti-symmetric form, also called the symplectic form of N modes with xxpp order.

6.2 Fock basis

In second quantization, the Fock basis is used to describe a quantum many-body system with an unknown number of particles. The canonical operators of one single mode have their equivalence in the Fock space with the form of bosonic creation and annihilation operators like

$$\hat{a} = \hat{x} + i\hat{p}, \ \hat{a}^\dagger = \hat{x} - i\hat{p}, \tag{6}$$

also called ladder operators or, when it comes to photons, photon addition and subtraction operators.

As a consequence, the unitary transformation related to the change from canonical to ladder operators for an N-mode quantum system is given by

$$\bar{U} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbb{1}_N & i\mathbb{1}_N \\ \mathbb{1}_N & -i\mathbb{1}_N \end{pmatrix}. \tag{7}$$

Therefore, the previously defined vector of canonical operators $\hat{\mathbf{s}}$ can also describe the modes in terms of creation and annihilation operators using the Eq. (7) unitary matrix resulting into a new vector of N modes in Fock basis

$$\hat{b} = (\hat{a}_1, \hat{a}_2, ..., \hat{a}_N, \hat{a}_1^\dagger, \hat{a}_2^\dagger, ..., \hat{a}_N^\dagger)^T. \tag{8}$$

Likewise, the canonical commutation relation from Eq. (5) turns into

$$[\hat{b}, \hat{b}^{\dagger}] = [\bar{U}\hat{\mathbf{s}}, \hat{\mathbf{s}}^{T}\bar{U}^{\dagger}] = \bar{U}[\hat{\mathbf{s}}, \hat{\mathbf{s}}^{T}]\bar{U}^{\dagger} = i\bar{U}J\bar{U}^{\dagger} = \begin{pmatrix} \mathbb{1}_N & \mathbb{0}_N \\ \mathbb{0}_N & -\mathbb{1}_N \end{pmatrix} \tag{9}$$

in Fock space picture using $\bar{U}$ as well which, in turn, comes from the commutation relation of both ladder operators

$$[\hat{a}_j, \hat{a}_k^{\dagger}] = -[\hat{a}_j^{\dagger}, \hat{a}_k] = \delta_{j,k}. \tag{10}$$

6.3 Symplectic, orthogonal and unitary groups

For the sake of further understanding, three types of algebraic groups that play an important role in the thesis need are introduced here and whose derivations are shown in Appendix A [Ser17].

The unitary group $U(N)$ is the group of $N \times N$ unitary matrices such that

$$UU^{\dagger} = \mathbb{1}_N, \, \forall U \in U(N). \tag{11}$$

The orthogonal group $O(N)$ is the group of $N \times N$ orthogonal matrices satisfying

$$OO^{T} = O^{T}O = \mathbb{1}, \, \forall O \in O(N). \tag{12}$$

The real symplectic group $Sp(2N, \mathbb{R})$ is the group of $2N \times 2N$ real symplectic matrices that preserve the real symplectic form J as

$$SJS^{T} = S^{T}JS = J \text{ with } J = \begin{pmatrix} \mathbb{0}_N & \mathbb{1}_N \\ -\mathbb{1}_N & \mathbb{0}_N \end{pmatrix}, S \in Sp(2N, \mathbb{R}). \tag{13}$$

Following the Appendix A deductions, the intersection of both symplectic and orthogonal groups denoted by $K(N) = Sp(2N, \mathbb{R}) \cap O(2N)$, so a given matrix $Q \in K(N)$ fulfills

$$Q = \begin{pmatrix} X & Y \\ -Y & X \end{pmatrix} \text{ with } \begin{cases} XY^{T} - YX^{T} = \mathbb{0}_N \\ XX^{T} + YY^{T} = \mathbb{1}_N \end{cases}. \tag{14}$$

This drives to an isomorphism between $K(N)$ and $U(N)$, thus allowing the representation of Q as a unitary matrix U_Q by applying (7) like

$$\bar{U}Q\bar{U}^{\dagger} = \begin{pmatrix} X - iY & \mathbb{0}_N \\ \mathbb{0}_N & X + iY \end{pmatrix} = \begin{pmatrix} U_Q^{*} & \mathbb{0}_N \\ \mathbb{0}_N & U_Q \end{pmatrix}. \tag{15}$$

The relation of both representations through the matrix $\bar{U}$ provide the tools to build a symplectic and orthogonal matrix $Q \in K(N)$ in a simple way out of a unitary one.

6.4 Gaussian quantum states

According to [WPGP$^+$12], Gaussian states are referred to continuous-variable systems whose states can be represented via Gaussian probability distributions in the phase space of their quadrature, also called normal distributions. Thereby, as stated in [Ser17], a Gaussian quantum state is only possible in those quantum systems whose dynamics are governed by a Hamiltonian at most quadratic in the quadratures.

In general, a quantum system characterized by a more-than-quadratic Hamiltonian carries the system to evolve into a non-Gaussian state which, in general, is not easy to simulate classically while Gaussian states are always efficiently simulable [BSBN02].

The set of Gaussian quantum states are composed by all the thermal and ground states of second-order Hamiltonians with a positive definite matrix $H > 0$ modelling the dynamics of a system and whose expectation value is the energy [Ser17].

Any Gaussian quantum state of N modes can be described by

$$\rho_G = \frac{e^{-\beta \hat{H}}}{Tr[e^{-\beta \hat{H}}]} \text{ with } \beta \in \mathbb{R}^+, \tag{16}$$

where β is the inverse temperature of the Boltzmann constant and $\hat{H}$ is the most general second-order Hamiltonian with the form

$$\hat{H} = \frac{1}{2}\hat{\mathbf{s}}^T H \hat{\mathbf{s}} + \hat{\mathbf{s}}^T \hat{\mathbf{s}} \text{ with } \begin{cases} \hat{\mathbf{s}} = (\hat{x}_1, ..., \hat{x}_N, \hat{p}_1, ..., \hat{p}_N)^T \\ H \in > 0 \end{cases} . \tag{17}$$

The Gaussian quantum state in (16) represents a mixed state for β finite. The equation would represent a pure state when taking $\beta \to \infty$ [Ser17].

Once the set of N-mode Gaussian quantum states is defined in terms of Hamiltonian matrix H and their corresponding vector $\hat{\mathbf{s}}$ containing the canonical operators $\hat{x}$ and $\hat{p}$ for each mode, the symplectic diagonalization of the Hamiltonian matrix

$$S_H H S_H^T = D \text{ with } \begin{cases} S_H = e^{JH} \in Sp(2N, \mathbb{R}) \\ D = diag(d_1, ..., d_N, d1, ...d_N), \; d_j \in \mathbb{R}^+ \; \forall j \end{cases} \tag{18}$$

can be performed.

This method allows the normal mode decomposition of H that yields a symplectic transformation S_H defined as

$$H = S_H^{-1} D (S_H^T)^{-1} = S_H^T D S_H \tag{19}$$

$$H = S_H^T \left[\left(\bigoplus_{j=1}^{N} \omega_j \right) \oplus \left(\bigoplus_{j=1}^{N} \omega_j \right) \right] S_H = S_H^T \left[\bigoplus_{l=1}^{2} \left(\bigoplus_{j=1}^{N} \omega_j \right) \right] S_H \text{ with } j \equiv mode \tag{20}$$

and the symplectic eigenvalues ω_j of H which are equivalent to the frequencies of the quantum state normal modes.

Additionally, by the fact that position and momentum of each mode are related, the symplectic eigenvalues are degenerated in pairs, one degenerated pair per mode, as Eq. (20) dictates.

Therefore, plugging the new expression of the Hamiltonian matrix (20) into the first term of the Hamiltonian (17) and using the action by congruence of the quadratic Hamiltonian on canonical operators inside $\hat{r}$ defined as

$$S_H \hat{r} = \hat{S}_H \hat{r} \hat{S}_H^\dagger \text{ where } \begin{cases} S_H = e^{JH} \in Sp(2N, \mathbb{R}) \\ \hat{S}_H = e^{i\frac{1}{2}\hat{r}^T H \hat{r}}, \end{cases} \tag{21}$$

the canonical operators of each mode can be joined with its frequency ω_j and simplified like

$$\hat{r}^T H \hat{r} = \hat{r}^T S_H^T \left[\bigoplus_{l=1}^{2} \left(\bigoplus_{j=1}^{N} \omega_j \right) \right] S_H \hat{r} = \hat{S}_H \left(\sum_{j=1}^{N} \omega_j (\hat{x}_j^2 + \hat{p}_j^2) \right) \hat{S}_H^\dagger. \tag{22}$$

By expression (22), it turns out that every second-order Hamiltonian with no displacement and $H > 0$ is equivalent to the Hamiltonian of N free and non-interacting harmonic oscillators.

6.5 Statistical moments of Gaussian states. The covariance matrix

Quantum states can be fully described by their of statistical moments. In the case of Gaussian quantum states, the first two statistical moments are enough to completely define the spectrum of any Gaussian state [Ser17].

The covariance matrix σ is reachable in reality through the second statistical moments of the canonical operators' [Ser17]. Formally, they are defined as

$$\bar{s} = Tr[\rho_G \hat{\mathbf{s}}] \text{ and } \sigma = Tr[\rho_G\{(\hat{\mathbf{s}}-\bar{s}),(\hat{\mathbf{s}}-\bar{s})^T\}], \text{ where } \{\hat{\mathrm{a}},\hat{\mathrm{a}}^T\} = \hat{\mathrm{a}}\hat{\mathrm{a}}^T + (\hat{\mathrm{a}}\hat{\mathrm{a}}^T)^T. \quad (23)$$

For simplicity concerns, the main object used to describe and treat with a Gaussian state throughout the thesis is the covariance matrix σ of the state representing its second statistical moments. Thus, the mean values or first moments vector of the Gaussian state is always be neglected in this project.

The spectrum of a Gaussian state is determined by its covariance matrix σ which is composed by a symplectic transformation S and symplectic eigenvalues ν_j of the state that diagonalizes the system's Hamiltonian, as seen in Eq. (20). The expression

$$\sigma = S\left[\bigoplus_{l=1}^{2}\left(\bigoplus_{j=1}^{N}\nu_j\right)\right]S^T, \quad \text{where} \begin{cases} \nu_j = \frac{1+e^{-\beta\omega_j}}{1-e^{-\beta\omega_j}} \geq 1, \text{ eigenvalues of } iJ\sigma \\ S \in Sp(2N,\mathbb{R}) \end{cases} \quad (24)$$

shows the general form of the covariance matrix σ for any Gaussian state and, provided its form, it is always real and symmetric.

To conclude this section, the purity μ of a quantum state ρ is commonly defined as the trace of its squared density operator which measures the level of mixture of the state. When it comes to a Gaussian quantum state ρ_G, its purity can be defined in terms of its covariance matrix σ whose expression is simplified by [Ser17] as

$$\mu(\rho_G) = Tr[\rho_G^2] = \frac{1}{\sqrt{Det(\sigma)}}. \quad (25)$$

Due to the definition and properties of the covariance matrix σ of a Gaussian state ρ_G, there are multiple metrics for the purity measurement of the state [Ser17], all collected in

$$\rho_G \text{ is pure } \iff \begin{cases} \bullet\ \mu(\rho_G) = 1 \\ \bullet\ Det(\sigma) = 1 \\ \bullet \text{ All } \sigma \text{ symplectic eigenvalues } (iJ\sigma) = 1 \\ \bullet\ \sigma = SS^T \quad \forall S \in Sp(2N,\mathbb{R}) \\ \bullet\ -JSJS = \mathbb{1} \end{cases}, \quad (26)$$

where all metrics from can be derived from one another, i.e. they are equivalent between them. If only one of the conditions is fulfilled, the rest are automatically true as well.

6.6 The uncertainty principle

Although a Gaussian quantum state is represented by its covariance matrix which has to be real and symmetric, not all matrices fulfilling these conditions are related to a possible quantum state. As the canonical operators $\hat{x}$ and $\hat{p}$ do not commute, i.e. $[\hat{x},\hat{p}] \neq 0$, there are some constraints over the covariance of said operators that are the outcome of the Heisenberg uncertainty principle formulated as

$$\sigma_x\sigma_p \geq \frac{\hbar}{2}. \quad (27)$$

From [Ser17] derivations, the uncertainty principle can be expressed in terms of the covariance matrix and the canonical commutation relation as

$$Tr[\rho(\hat{r}-\bar{r})(\hat{r}-\bar{r})^T] = \frac{1}{2}Tr\left[\rho\{(\hat{r}-\bar{r}),(\hat{r}-\bar{r})^T\} + [\hat{r},\hat{r}^T]\right] = \sigma + iJ, \tag{28}$$

where an inequality can be induced by the positiveness of the final state's trace reaching the so-called Robertson-Schrödinger uncertainty relation [Ser17] described as

$$\sigma + iJ \geq 0, \tag{29}$$

from which it is inferred that $\sigma > 0$.

In conclusion, σ is the covariance matrix of a valid quantum state if and only if all eigenvalues of $\sigma + iJ$ are positive. In addition, pure Gaussian states always minimize the uncertainty principle.

6.7 Gaussian operators

A Gaussian operator is a transformation that maps a Gaussian state to another Gaussian state, which means that Gaussian operators preserve the Gaussianity of the quantum state when acting upon it [Ser17].

In general, there are plenty of these Gaussian operators but for this project the most important ones are the squeezing operator and the symplectic-orthogonal operators, which altogether form a symplectic operator, by the fact that all pure Gaussian operations can be decomposed in the product of these two operations.

These operators can be applied in a specific way to any Gaussian state represented by its density matrix ρ_G. However, as ρ_G is characterized by its covariance matrix σ, the focus is set on how these Gaussian operators change the variances of the state quadratures when they act over it.

The displacement operator is a linear displacement of the Gaussian quantum state ρ_G but its covariance matrix remains unchanged, this is

$$D_r: \ \sigma \rightarrow \sigma, \tag{30}$$

then it is completely neglected in the project.

Besides that, as seen in (21), the unitary action of a Gaussian quadratic operator $\hat{S}$ related to the symplectic transformation S causes a transformation of the state ρ_G equivalent to $\hat{S}\rho_G\hat{S}^\dagger$ which maps, in turn, the covariance matrix of ρ_G the same way as seen in (24) [Ser17], what can be expressed as

$$S: \ \sigma \rightarrow S\sigma S^T, \ \text{where } S = e^{JH}. \tag{31}$$

Last but not least, the squeezing operator act on quantum state quadratures reducing the variance of one while increasing the variance of the other. It is represented by a diagonal matrix Z and acts over σ in the same way as Eq. (31):

$$Z: \ \sigma \rightarrow Z\sigma Z^T, \ \text{where } Z = diag(z_1, ..., z_N, z_1^{-1}, ..., z_N^{-1}). \tag{32}$$

It is worth mentioning that any symplectic matrix S can be decomposed as

$$S = Q_2 Z Q_1, \ \text{where} \begin{cases} Q_1, Q_2 \in Sp(2N, \mathbb{R}) \cap O(2N) \\ Z = diag(z_1, ..., z_N, z_1^{-1}, ..., z_N^{-1}) \end{cases} \tag{33}$$

what's named after the Bloch-Messiah or Euler decomposition [Ser17], or, the other way around, any symplectic matrix S can be constructed with two symplectic-orthogonal matrices and one diagonal matrix related to the squeezing operator.

This is important for the QNN construction because any passive-optics operator is represented by a symplectic-orthogonal matrix $Q \in Sp(2N, \mathbb{R}) \cap O(2N)$.

6.8 Non-Gaussianity

As aforementioned, a Gaussian quantum state is fully represented by its first and second statistical moments, i.e. its mean vector and covariance matrix. When a quantum state is non-Gaussian it requires more statistical moments in order to completely characterize the quantum state.

Non-Gaussian operators introduce non-linearity when applied to Gaussian quantum states which is an indispensable piece to implement non-linear transformations on artificial neural networks such as activation functions and feature extraction [KBA+19].

Moreover, [WST+23] shows that non-Gaussianity is a necessary element to gain quantum computational advantage against classical algorithms and simulations.

The simplest non-Gaussian operators and readily available in many well-equipped photonics labs are the photon addition or subtraction operators. They are the main non-Gaussian operators used in this project but, in addition, the Kerr operator $\hat{K}(\kappa) = e^{i\kappa\hat{n}^2}$, where $\hat{n}$ is the number operator, and the cubic phase operator $\hat{V}(\gamma) = e^{i\gamma\hat{x}^3/3}$, where $\hat{x}$ is the position operator, are other valid non-linear operators [KBA+19].

A crucial aspect is that, unlike the Kerr and cubic phase operators, ladder operators are not unitary so, when they act over a quantum state, the resulting state must be normalized.

6.9 Gaussian measurements

An essential requirement for a neural network is to introduce non-linear transformations as a way to learn non-linear relationships between the input variables and the outputs. What's more, the Universal Approximation theorem holds that an artificial neural network with one hidden layer and a non-linear activation function can approximate any continuous function with an arbitrary accuracy depending on the number of neurons in the hidden layer [PSCLGFL20].

As previously stated, the ladder operators are the main non-linear transformations employed for this quantum neural network model. This would imply in some cases to consider more than the first and second statistical moments to work out the non-Gaussianity of the state and it would still not be classically simulable. However, there are direct relations between the action of creation and annihilation operators over a Gaussian state ρ_G and its covariance matrix σ [**?**].

In order to obtain the expectation value of a Gaussian quantum state of N modes after a non-Gaussian operator like $\hat{a}_j$ or $\hat{a}_j^\dagger$, where j stands for the mode the operators acts on, the covariance matrix σ representing the Gaussian quantum state ρ_G may be used according to [**?**]. There, it is shown how the expectation value of the energy is obtained through σ when ladder operators are applied by congruence over any ρ_G.

Each expectation value of a pair of ladder operators acting on a specific mode of ρ_G has its correspondence with one of the four identities in

$$I_1 = Tr[\hat{a}_j\hat{a}_k\rho_G] = \frac{1}{4}[\sigma_{j,k} - \sigma_{N+j,N+k} + i(\sigma_{j,N+k} + \sigma_{N+j,k})], \tag{34}$$

$$I_2 = Tr[\hat{a}_j^\dagger\hat{a}_k\rho_G] = \frac{1}{4}[\sigma_{j,k} + \sigma_{N+j,N+k} + i(\sigma_{j,N+k} + \sigma_{N+j,k})] - 2\delta_{j,k}, \tag{35}$$

$$I_3 = Tr[\hat{a}_j\hat{a}_k^\dagger\rho_G] = I_2 + \delta_{j,k}Tr[\rho_G], \tag{36}$$

$$I_4 = Tr[\hat{a}_j^\dagger\hat{a}_k^\dagger\rho_G] = \frac{1}{4}[\sigma_{j,k} - \sigma_{N+j,N+k} - i(\sigma_{j,N+k} + \sigma_{N+j,k})], \tag{37}$$

where $\sigma_{j,k}$ is the covariance matrix element of row j and column k and $\delta_{j,k}$ is the Kronecker delta function which is always zero except for $j = k$.

Moreover, if there are multiple ladder operators transforming a Gaussian state ρ_G, the final expectation value can also be calculated by computing all possible perfect matchings $\mathcal{P}$ of the operators inside the trace representing the expectation value [**?**], which is given by the formulas

$$Tr[\hat{a}^\dagger_{S_N}...\hat{a}^\dagger_{S_1}\hat{a}^\dagger_{C_M}...\hat{a}^\dagger_{C_1}\hat{a}_{C_1}...\hat{a}_{C_M}\hat{a}_{S_1}...\hat{a}_{S_N}\rho_G] = \frac{1}{K}\sum_{\mathcal{P}}\prod_{\{p_1,p_2\}\in\mathcal{P}} Tr[\hat{a}^{\#}_{p1}\hat{a}^{\#}_{p2}\rho_G], \tag{38}$$

$$K = Tr[\hat{a}^\dagger_{S_N}...\hat{a}^\dagger_{S_1}\hat{a}_{S_1}...\hat{a}_{S_N}\rho_G], \tag{39}$$

where K is the normalization factor after the photon subtractions $\hat{a}_{S_i}$, $\hat{a}_{C_i}$ correspond to the observable quantity whose expectation value is to be found, and $\hat{a}^{\#}_j$ means that the operator of mode j may be whether a creation or annihilation operator.

It is crucial to mention that there are no real difference between $\hat{a}^{\#}_{C_i}$ and $\hat{a}^{\#}_{S_i}$. It is only a way to distinguish the operators that create or annihilate photons and the operators related to the observable.

Note that the number of total ladder operators must be even to be able to use the expression.

Regarding the complexity, defining the Eq. (38) with a total of n ladder operators, the number of different perfect matchings is $(n-1)!!$, which translates into $(n-1)!!$ different trace expressions and makes it a hard task for classical computation.

In terms of simulation of the quantum state evolution for the neural network training using the covariance matrix of a Gaussian state, the measurements can be performed by homodyne or heterodyne detection which are Gaussian measurements based on the observation of quadrature operators or ladder operators that agree with the expectation values expressions obtained through the covariance matrix.

From the structure of Eq. (38), one feasible observable could be the number operator $\hat{N}$, defined as

$$\hat{N}_i = \hat{a}^\dagger_{C_i}\hat{a}_{C_i},\ i \in \{1, ..., N\}, \tag{40}$$

which counts the number of photons of a certain mode.

7 Quantum Neural Network with Quantum Optics

As mentioned at the beginning of the project, the purpose is to develop a method such that, for a given objective function $f(\vec{x})$ of M inputs $\vec{x} = (x_1, ..., x_M)$, the neural network is capable of approximating $f(x)$ as for any input like

$$\tilde{f}(\vec{x}) = \Phi(W\vec{x} + \vec{b}), \tag{41}$$

where W is the weights matrix that linearly transforms the input vector $\vec{x}$, $\vec{b}$ is a real vector related to the bias and Φ is a non-linear activation function allowing the learning of non-linear patterns.

The approach used in [KBA$^+$19] is based on a general quantum neural network in a continuous variable framework like quantum optics, the data is encoded in $|x_i\rangle$, corresponding to the real part of the electromagnetic complex field of the photon i, and processed with a sequence of l layers. Each layer $\mathcal{L}_l$ has an independent composition of operators

$$\mathcal{L}_l := \hat{\Phi}^{(l)} \circ \hat{\mathcal{D}}^{(l)} \circ \hat{\mathcal{U}_2}^{(l)} \circ \hat{\mathcal{S}}^{(l)} \circ \hat{\mathcal{U}_1}^{(l)}, \tag{42}$$

where $\hat{\mathcal{U}_1}$,$\hat{\mathcal{U}_2}$ are passive optics operators acting on all modes of the system, $\hat{\mathcal{S}}$ is a joint squeezing operator with one squeezing factor for each mode, $\hat{\mathcal{D}}$ is also a joint displacement with one displacement parameter per mode and $\hat{\Phi}$ is a non-Gaussian transformation. The direct mapping between these components and Eq. (41) are

$$W = \hat{\mathcal{U}}_2 \hat{\mathcal{S}} \hat{\mathcal{U}}_1, \;\; \vec{b} = \hat{\mathcal{D}} \text{ and } \Phi = \hat{\Phi}. \tag{43}$$

However, the innovative QNN simulation approach of this project focuses on the covariance matrix of the quantum state as the unique entity of information representation, which means that the information has to be encoded in the variances of the particles positions instead of positions themselves as in [KBA+19]. Therefore, the layer components from (42) need to be adjusted for the covariance matrix to represent the information. Furthermore, this implementation goes beyond [KBA+19] because it employs a hardware inspired ansatz for the QNN and the non-Gaussian operations used are actually available with current technology. In the experiment carried out in [SvdMA+21] the architecture would be the same as the one described here.

Up to now, all required components to build the proposed QNN mechanism in a continuous-variable system were outlined throughout the previous sections. Next, in this section, the components, parameters, data preparation, architecture for data processing and output obtaining of covariance-based QNN method are thoroughly described for classical simulation and optical quantum computers implementation.

7.1 Components

Bringing up the studied foundations of earlier sections in order to adapt the components of (42) to the presented QNN approach, the elements of the suggested neural network mimicking the artificial neural network composition are listed in the following table.

NN COMPONENT	QNN COMPONENT	QNN OBJECT
Encoding structure	Quadratures covariance matrix	σ
Inputs and neurons	Number of modes	N
Weight matrix	Symplectic matrix	$S = Q_2 Z Q_1$
Non-linear function	Photon creation/annihilation	$\hat{\Phi} = \;\hat{a}^{\#}$
Layers	Previous components stack	$\mathcal{L}$
Outputs	Homodyne measurement	$\prod \hat{a}^{\#}\hat{a}^{\#}$

Table 1: Quantum optics components relative to artificial neural networks components

As seen in Eq. (30), the displacement transformation does not affect the covariance matrix of the quantum state, therefore all kinds of displacements are entirely neglected for this QNN methodology.

Given that the number of modes N plays both number of inputs and number of neurons roles, one remark that must be underlined is the fact that N shall be at least as many as the number of inputs of the QNN so the encoding structure σ has sufficient degrees of freedom for all inputs to be considered in the learning and predicting phases.

Nonetheless, the number of modes N can exceed the number of inputs as a way to implement more neurons in the layers where the extra modes are not encoding any input but remain encoded by default, which translates into a squeezing factor of 1 for said extra modes. So, in order to decrease the number of neurons for each layer, the spare modes can simply be traced out or measured [KBA+19].

7.2 Data preparation

As just stated, the data encoding element for this QNN model is the covariance matrix of the quadratures of each mode, meaning the quadratures variances store the information, specifically, only positions variances because position and momentum variances are inversely related which would result into less degrees of freedom if encoding in both of them.

This forces the input data encoding to be carried out by some operator that modifies the variances of each mode independently, what is achieved through the squeezing operator. Consequently, the input data encoding take place on the squeezing factor r_i of each mode's position.

At the beginning stage, the initial Gaussian state represented by its covariance matrix is initialized with squeezing operators to encode the input data for the QNN. Due to some limitations described in [PSCLGFL20], the adopted encoding strategy is the one suggested there: data re-uploading.

The data re-uploading technique consists of encoding multiple times the data along the neural network with the objective of allowing complex patterns learning. The main goal with this technique is to induce non-linearity needed to apply the universal approximation theorem [PSCLGFL20].

As each layer $\mathcal{L}_l$ has an independent squeezing operator per mode, the data re-uploading at each layer is feasible and easily achievable in this QNN model.

7.3 Parameters

The parameters to be optimized may be kind of flexible. In the implemented QNN model, the ones chosen to be tuned are the squeezing factors from Z of the first layer only because of the data re-uploading strategy and the passive optics parameters corresponding to the symplectic-orthogonal matrices $Q1$, $Q2$. This results into N parameters for the first squeezing operator and N^2 free parameters for each passive optics operator $Q1, Q2$. If the number of layers is l, this sums up to $l(2N^2) + N$ tunable parameters.

The squeezing operator is a diagonal matrix with the squeezing values for the position variances and the inverse of these values for the momentum variances.

The passive optics operator Q_j is a $2N \times 2N$ symplectic-orthogonal matrix whose construction is based on Eq. (15), which uses a unitary matrix U obtained by

$$U = e^{iH} \text{ with } H \in \mathbb{R}^N \times \mathbb{R}^N. \tag{44}$$

Nevertheless, the non-linear activation function $\hat{\Phi}$ could be also added to the optimization parameters in terms of the number of photon addition/subtraction operators applied at each layer. Likewise, the observable yielding the QNN outputs could be another tunable object.

It is worth noting that, in general, the ladder operators may be restricted to act over one mode only because swapping two modes is a Gaussian operator [CFGM21] which is going to be optimized. Thereby, the application of the non-Gaussian operators is usually seen just on the first mode throughout the project and the optimization would eventually reach to mode swapping if needed.

7.4 Architecture

Since the data structure used for information encoding and processing is the covariance matrix σ of the quantum system, the action of each operator from $\mathcal{L}_l$ on σ shall be consistently translated using the relations from Table 1.

It's worth highlighting that all operators from (42) are Gaussian except for the nonlinear transformation $\hat{\Phi}$ whose effect, performed in quantum optics via photon creation and annihilation operators ($\hat{a}$ and $\hat{a}^\dagger$), cannot be entirely described by the covariance matrix of the quantum state but requires more statistical moments.

To avoid the computation of additional statistical moments of the state and just stick to the covariance matrix evolution, Eq. (38) is employed in order to calculate the expectation value of some quadratic ladder observable after the action of any number of photon subtractions or additions over the Gaussian quantum state.

Therefore, by virtue of simplicity, it is convenient to split the Gaussian and the non-Gaussian parts which is even required for a later stage.

$$\begin{aligned} S^{(l)} &= Q_2^{(l)} Z^{(l)} Q_1^{(l)} \\ \hat{\Phi}^{(l)} &= \hat{a}_j^{\#} \end{aligned} \tag{45}$$

Hence, the action of the Gaussian part of the layer would modify the covariance matrix σ of the quantum state as

$$\hat{G}: \ \sigma \rightarrow Q_2 Z Q_1 \sigma Q_1^T Z Q_2^T \tag{46}$$

and, finally, the non-Gaussian transformation $\hat{\Phi}$ effect over this Gaussian quantum state is obtained together with some Fock space quadratic observable by defining its corresponding expectation value expression form analogous to template Eq. (38).

Summing up, if the chosen non-Gaussian operator is a photon creation on mode 1 and the initial quantum state ρ_{G_0} is a Gaussian state of N modes $|G_0\rangle\langle G_0|^{\otimes N}$, which in this QNN model is a squeezed Gaussian state related to the initial input encoding, the quantum state evolution after the complete layer would be characterized by

$$\hat{a}_1^\dagger \hat{G} \ \rho_{G_0} \ \hat{G}^\dagger \hat{a}_1. \tag{47}$$

Clearly, this method is only valid when the QNN is composed just by one single layer, i.e. a quantum perceptron, due to the lack of expressiveness of the covariance matrix after the non-Gaussian transformation of the first layer. So, it would be impossible to track the evolution of multiple layers as

$$\hat{a}_1^\dagger \hat{G}^{(l)} \hat{a}_1^\dagger \hat{G}^{(l-1)} ... \hat{a}_1^\dagger \hat{G}^{(1)} \ \ |G_0\rangle\langle G_0|^{\otimes N} \ \hat{G}^{\dagger(1)} \hat{a}_1 ... \hat{G}^{\dagger(l-1)} \hat{a}_1 \hat{G}^{\dagger(l)} \hat{a}_1 \tag{48}$$

solely taking into account the covariance matrix of the quantum state.

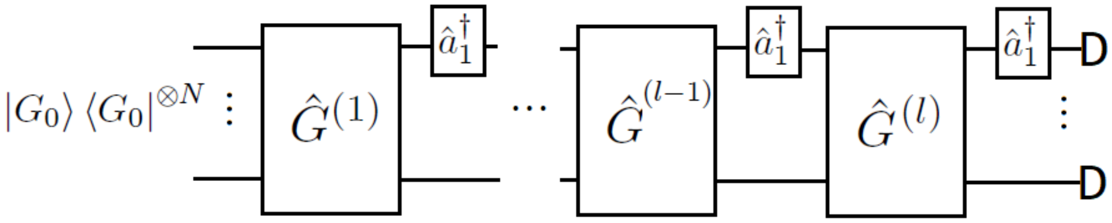

Figure 1: Multi-layer Quantum Neural Network circuit of Eq. (48).

To be able to implement a multi-layer quantum neural network, as the one shown in Figure 1, with the proposed method using the expression of Eq. (38), it is essential to have all non-Gaussian transformations at the latest stage and all Gaussianity at the beginning.

Fortunately, this can be fixed by looking at how the action of a Gaussian unitary operator $\hat{G}$ influences the ladder operator. This way, the commutation between a Gaussian operator and the photon addition or subtraction non-Gaussian operators can be performed.

Recalling from Gaussian operators section, the unitary action of a Gaussian operator $\hat{G}$ over a Gaussian quantum state ρ_G modifies its covariance matrix through its symplectic representation S as Eq. (31).

Similarly, the unitary application of a Gaussian operator $\hat{G}$ over the photon addition operator in mode k is achieved with its symplectic matrix representation S of $\hat{G}$ [CFGM21], [WPGP+12] like

$$\hat{G}\hat{a}_k^\dagger\hat{G}^\dagger = \sum_{j=1}^{N} s_{k,j}\hat{a}_j^\dagger + s_{k,N+j}\hat{a}_j, \tag{49}$$

where $s_{j,k}$ represent the element from row j and column k of the symplectic matrix S.

Working out Eq. (49) to drag the non-Gaussianity to the left side gives as result

$$\hat{G}\hat{a}_k^\dagger = \left(\sum_{j=1}^{N} s_{k,j}\hat{a}_j^\dagger + s_{k,N+j}\hat{a}_j\right)\hat{G} = \mathbb{S}_k\hat{G} \tag{50}$$

where $\mathbb{S}_k$ is a superposition of photon addition and subtraction operators for each mode $j \in 1, ..., N$ with coefficients $s_{j,k} \in \mathbb{R}$ corresponding to elements in the symplectic matrix of $\hat{G}$.

Now, plugging Eq. (50) into Eq. (48) multiple times starting from the edges of the expression gives

$$\hat{a}_1^\dagger\mathbb{S}^{(l)}...\ \mathbb{S}^{(2)}\hat{G}^{(l)}...\ \hat{G}^{(1)}\ \left|G_0\right\rangle\left\langle G_0\right|^{\otimes N}\hat{G}^{\dagger(1)}...\ \hat{G}^{\dagger(l)}\mathbb{S}^{\dagger(2)}...\ \mathbb{S}^{\dagger(l)}\hat{a}_1. \tag{51}$$

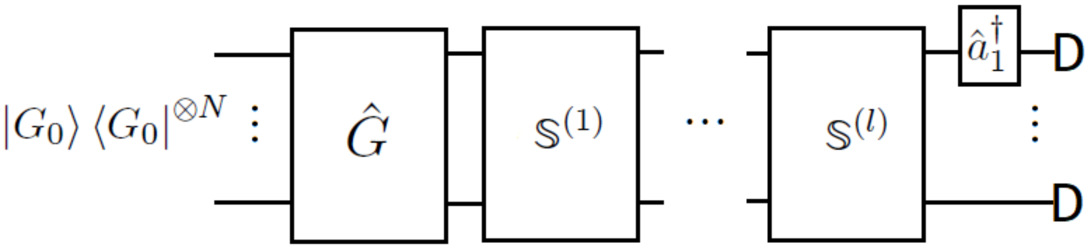

Figure 2: Non-Gaussianity swapped multi-layer Quantum Neural Network circuit of Eq. (51).

Note that the symplectic coefficients of ladder operators superposition $\mathbb{S}^{(k)}$ are directly related to the Gaussian unitary $\hat{G}^{(k)}$ configuration. In spite of that, all Gaussian operators can be gathered into a single one $\hat{G}$ by two facts: 1) the product of Gaussian transformations is also a Gaussian transformation and 2) the first Gaussian operator $\hat{G}^{(1)}$ is of free choice because it does not need to be permuted with any non-Gaussian operator.

As Gaussian operators are to be optimized during the quantum neural network training process, $\hat{G}^{(1)}$ would have an independent shape and it would completely transform all other $\hat{G}^{(k)}$ decoupling all of them from their symplectic representation needed for $\mathbb{S}^{(k)}$.

At last, the desired general form for an l-layer quantum neural network is achieved and it looks like

$$\hat{a}_1^\dagger\mathbb{S}^{(l)}...\ \mathbb{S}^{(1)}\hat{G}\ \left|G_0\right\rangle\left\langle G_0\right|^{\otimes N}\hat{G}^\dagger\mathbb{S}^{\dagger(1)}...\ \mathbb{S}^{\dagger(l)}\hat{a}_1. \tag{52}$$

7.5 Output measurement

After the Gaussian operator $\hat{G}$ is applied to the initial Gaussian state $|G_0\rangle\langle G_0|^{\otimes N}$ at Eq. (52) giving a new Gaussian state ρ_G, the application of the non-Gaussian part together with the desired observable to be measured would result into an expression arranged as Eq. (38).

Therefore, this expression's expectation value representing the QNN output is obtained by performing all perfect matchings of the ladder operators in the expression, up to a normalization constant given by (39).

Thanks to the linearity of the trace, the summation of the different terms in the non-Gaussian ladder operator superposition can be split into the sum of traces of each of these terms.

In practice, the outcomes of the QNN with quantum optics are obtained under homodyne or heterodyne measurement of the final non-Gaussian quantum state. The form of the homodyne or heterodyne detection depends on the observable to be measured. This way, the expectation value of the observable can be statistically obtained by repeating the detection process.

7.6 Training and evaluation

Once defined all hyperparameters and the topology of the quantum neural network based on the objective function or the chosen dataset, the optimization algorithm has to be selected along with the loss function that is going to be minimized for the dataset evaluation and the QNN training stage can start.

Typically, the loss function is the mean squared error defined as

$$\text{MSE}\ (\tilde{f}(\vec{x})) = \frac{1}{M}\sum_{j=1}^{M}(f(\vec{x}_j) - \tilde{f}(\vec{x}_j))^2, \tag{53}$$

where $f(\vec{x})$ is the desired output value for inputs in $\vec{x}$, $\tilde{f}(\vec{x})$ is the predicted value of the neural network and M is the number of samples of one dataset batch.

First, all $l(2N^2) + N$ parameters to be tuned are initialized to a random value and all possible perfect matchings of the expectation value expression are calculated for both the observable and the non-Gaussianity normalization factor.

Then, the inputs and outputs of the dataset are loaded and normalized. Although it is optional, the dataset may also be split into minibatches for the training.

After that, the parameter optimization starts and repeats the Algorithm 1 calculating the loss function value of Eq. (53) and using the optimization method to update the parameters.

When the optimization has converged, meaning that the difference between consecutive loss function values is lower than a certain threshold defined in the optimizer, the training comes to an end.

Finally, the optimal parameters, i.e. the ones that minimizes the loss function, are retrieved and stand for the trained parameters of the neural network.

For evaluation, the optimal parameters obtained after training the QNN are used to build the passive optics and squeezing operators in order to reproduce the trained QNN and start making predictions by inserting inputs and measuring at the end of the quantum circuit. This is also reflected in Algorithm 1 but using the optimized parameters instead of random values.

7.7 Two-layer QNN example

As an illustration example, let's define a quantum neural network of 2 inputs, 2 outputs and 2 layers. This implies the constraint of $N \geq 2$ but, for simplicity, let's consider only 2 neurons per layer, so $N = 2$. On the other hand, the non-Gaussian part consists of one photon addition on first mode for each layer.

Then, considering that the initial Gaussian quantum state $|G_0\rangle \langle G_0|^{\otimes 2}$ is a 2-mode squeezed vacuum state with position squeezing factors equal to the inputs' values, the standard expression for the quantum system would be

$$\hat{a}_1^\dagger \hat{G}^{(2)} \hat{a}_1^\dagger \hat{G}^{(1)} \ |G_0\rangle \langle G_0|^{\otimes 2} \, \hat{G}^{\dagger(1)} \hat{a}_1 \hat{G}^{\dagger(2)} \hat{a}_1, \tag{54}$$

which, after applying the commutation relation, turns into

$$\hat{a}_1^\dagger \mathbb{S}^{(1)} \hat{G} \ |G_0\rangle \langle G_0|^{\otimes 2} \, \hat{G}^\dagger \mathbb{S}^{\dagger(1)} \hat{a}_1. \tag{55}$$

Now, establishing the evolution in terms of the covariance matrix of the system, the vacuum state covariance matrix σ_v is described by the identity $2N \times 2N = 4 \times 4$ where the initial squeezing operator related to the input encoding acts like

$$\sigma_0 = Z_\text{IN} \sigma_\text{v} Z_\text{IN}. \tag{56}$$

Next, recalling from Eq. (46) that the Gaussian operator $\hat{G}$ acts over the covariances with its symplectic representation $S = Q_2 Z Q_1$ applied by congruence, the covariance matrix σ_G of the final Gaussian state ρ_G ends up as

$$\sigma_G = Q_2 Z Q_1 \ \sigma_0 \ Q_1^T Z Q_2^T = Q_2 Z Q_1 Z_\text{IN} \ \sigma_\text{v} \ Z_\text{IN} Q_1^T Z Q_2^T. \tag{57}$$

Since all Gaussianity has been computed, the only step left is the non-Gaussianity and the measurement whose procedures are performed in a single shot when it comes to classical simulation.

On the one hand, the non-linear transformation is carried out by one photon addition operator on the first mode together with a superposition of all ladder operators of all modes whose coefficients come from the symplectic matrix representing $\hat{G}^{(1)}$ as seen in Eq. (50). They act by congruence over ρ_G resulting into the non-Gaussian state ρ_{NG} with the appearance

$$\rho_{NG} = \hat{a}_1^\dagger \mathbb{S}^{(1)} \ \rho_G \ \mathbb{S}^{\dagger(1)} \hat{a}_1, \tag{58}$$

which expanded has the form

$$\rho_{NG} = \hat{a}_1^\dagger \left(s_{1,1}^{(1)} \hat{a}_1^\dagger + s_{1,2}^{(1)} \hat{a}_2^\dagger + s_{1,3}^{(1)} \hat{a}_1 + s_{1,4}^{(1)} \hat{a}_2 \right) \rho_G \left(s_{1,1}^{(1)} \hat{a}_1 + s_{1,2}^{(1)} \hat{a}_2 + s_{1,3}^{(1)} \hat{a}_1^\dagger + s_{1,4}^{(1)} \hat{a}_2^\dagger \right) \hat{a}_1. \tag{59}$$

On the other hand, let's say that the observable to be measured at each mode is the number operator $\hat{N}_j = \hat{a}_j^\dagger \hat{a}_j$ where j stands for the measured mode. The normalized expectation values would be

$$\langle \rho_{NG} | \hat{N}_j | \rho_{NG} \rangle = \frac{Tr\left[\hat{N}_j \rho_{NG}\right]}{K} = \frac{Tr\left[a_j^\dagger \hat{a}_j \rho_{NG}\right]}{K}, \tag{60}$$

where the normalization factor K is equal to

$$K = Tr\left[\hat{a}_1^\dagger \left(s_{1,1}^{(1)} \hat{a}_1^\dagger + s_{1,2}^{(1)} \hat{a}_2^\dagger + s_{1,3}^{(1)} \hat{a}_1 + s_{1,4}^{(1)} \hat{a}_2 \right) \rho_G \left(s_{1,1}^{(1)} \hat{a}_1 + s_{1,2}^{(1)} \hat{a}_2 + s_{1,3}^{(1)} \hat{a}_1^\dagger + s_{1,4}^{(1)} \hat{a}_2^\dagger \right) \hat{a}_1 \right]. \tag{61}$$

By plugging Eq. (59) into (60), the outcomes when measuring the two modes observables look like

$$\langle \hat{N}_j \rangle = \frac{Tr\left[\hat{N}_j \hat{a}_1^\dagger \left(s_{1,1}^{(1)}\hat{a}_1^\dagger + s_{1,2}^{(1)}\hat{a}_2^\dagger + s_{1,3}^{(1)}\hat{a}_1 + s_{1,4}^{(1)}\hat{a}_2\right)\rho_G \left(s_{1,1}^{(1)}\hat{a}_1 + s_{1,2}^{(1)}\hat{a}_2 + s_{1,3}^{(1)}\hat{a}_1^\dagger + s_{1,4}^{(1)}\hat{a}_2^\dagger\right)\hat{a}_1\right]}{Tr\left[\hat{a}_1^\dagger \left(s_{1,1}^{(1)}\hat{a}_1^\dagger + s_{1,2}^{(1)}\hat{a}_2^\dagger + s_{1,3}^{(1)}\hat{a}_1 + s_{1,4}^{(1)}\hat{a}_2\right)\rho_G \left(s_{1,1}^{(1)}\hat{a}_1 + s_{1,2}^{(1)}\hat{a}_2 + s_{1,3}^{(1)}\hat{a}_1^\dagger + s_{1,4}^{(1)}\hat{a}_2^\dagger\right)\hat{a}_1\right]}. \tag{62}$$

Finally, expanding the terms and using the trace cyclic and linear properties, the expression (62) ends up as

$$\begin{aligned} \langle \hat{N}_j \rangle = \frac{1}{K}\left(s_{1,1}^{(1)2}\ Tr\left[\hat{N}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\hat{a}_1\hat{a}_1\right] + s_{1,1}^{(1)}s_{1,2}^{(1)}\ Tr\left[\hat{N}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\hat{a}_2\hat{a}_1\right] + \ ...\right) = \\ \frac{1}{K}\left(s_{1,1}^{(1)2}\ Tr\left[\hat{a}_1\hat{a}_1\hat{N}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\right] + s_{1,1}^{(1)}s_{1,2}^{(1)}\ Tr\left[\hat{a}_2\hat{a}_1\hat{N}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\right] + \ ...\right) = \\ \frac{1}{K}\left(s_{1,1}^{(1)2}\ Tr\left[\hat{a}_1\hat{a}_1\hat{a}_j^\dagger\hat{a}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\right] + s_{1,1}^{(1)}s_{1,2}^{(1)}\ Tr\left[\hat{a}_2\hat{a}_1\hat{a}_j^\dagger\hat{a}_j\hat{a}_1^\dagger\hat{a}_1^\dagger\rho_G\right] + \ ...\right), \end{aligned} \tag{63}$$

where the form of Eq. (38) becomes visible.

The last stage left for the calculation of the two QNN outputs $\langle \hat{N}_1 \rangle$ and $\langle \hat{N}_2 \rangle$ would be to compute all the perfect matchings for each term in Eq. (63) as well as for the normalization factor K using the right-hand-side formula from Eq. (38).

One significant remark is that the total number of terms in the expectation value expression scales in function of the number of modes N and the number of layers l like

$$(2N)^{2(l-1)}, \tag{64}$$

hence, for this case with 2 modes and 2 layers, the total number of terms sums up to 16.

This leads to the truth that any physical observable represented by ladder operators would be classically simulable with the presented method if it were not for the classical intractability of the double factorial together with the scaling factors of the number of modes N and layers l.

8 Results

First, the general resources used for the simulations are

- L-BFGS-B optimization algorithm from SciPy Python library
- MSE loss function from Eq. (53)
- Observable $\hat{N}_1 = \hat{a}_1^\dagger \hat{a}_1$.

The obtained results were made by using continuous monotonic-increasing polynomial functions as target functions, hence, implying only one input and one output of the QNNs.

To create the datasets for the polynomial functions, normalized random inputs between a range were created to represent the features of the problem along with their respective normalized function evaluations constituting the expected output of the QNN.

The results are composed by: 1) multiple modes and layers 2) the loss function values obtained throughout each iteration of the training phase of the QNN simulation for each mode and layers and 3) the evaluation of the testing set generated by randomly sampling the continuous functions and subsequently sorted by virtue of predictions visualization.

The first result comes from a cubic polynomial target function:

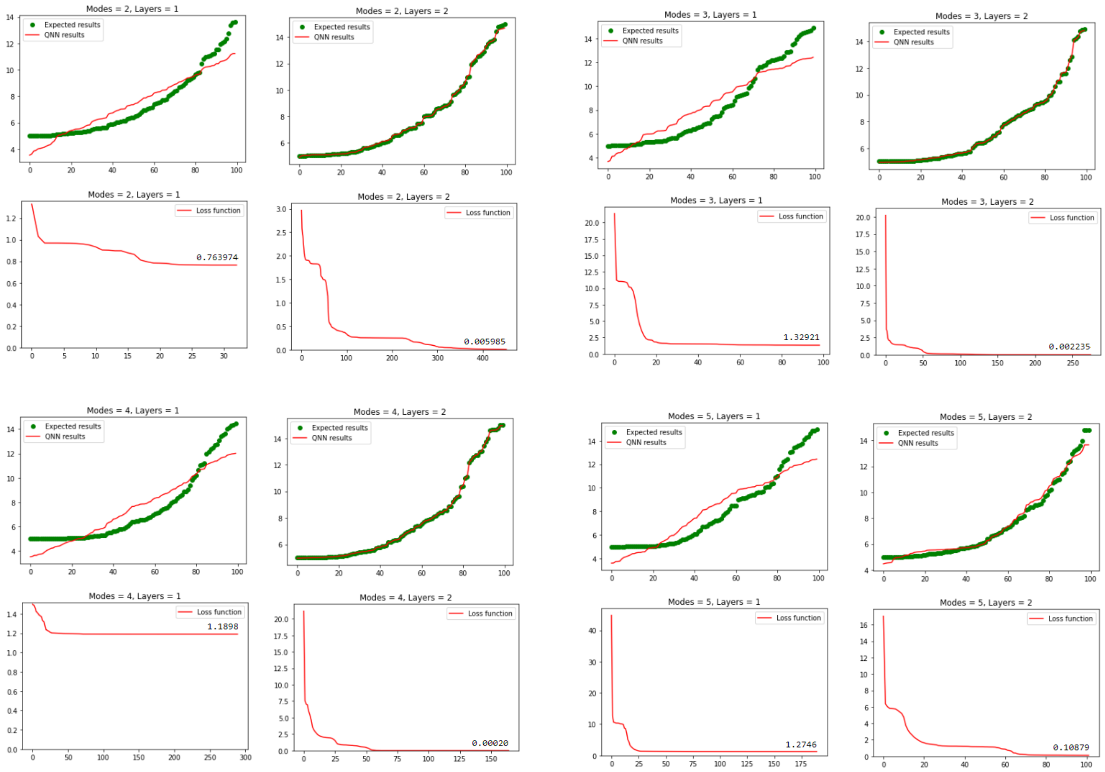

Figure 3: QNN training of target function $f(x) = 0.7x^3 + 1.7x^2 + 2.4x + 5$ for different modes and layers where the testing set is evaluated in the first and third row, with horizontal axis representing the evaluation sample, and the loss convergence of the training is shown in the second and fourth row, with horizontal axis representing the iteration of the optimization.

To increase the complexity, the following result is from a 5th-grade polynomial function:

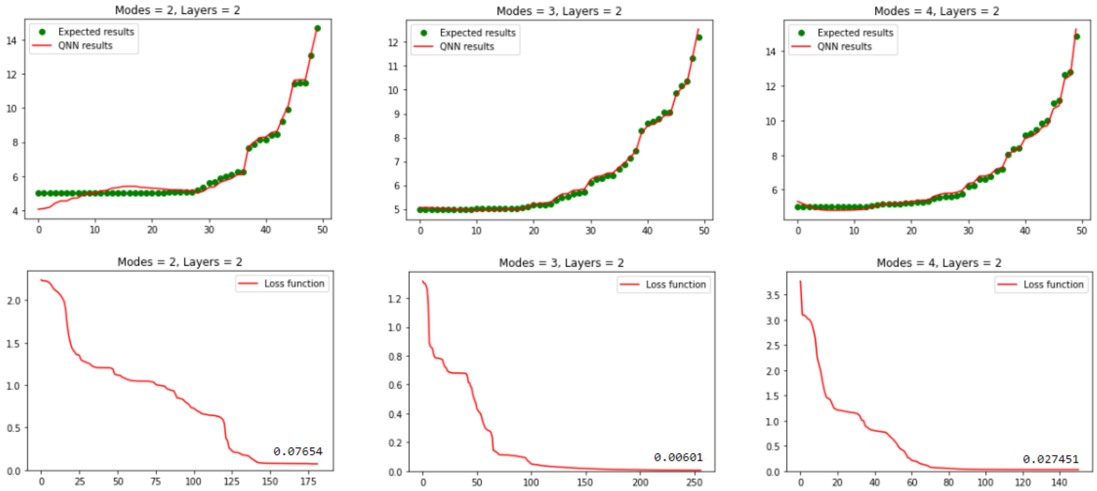

Figure 4: QNN training of target function $f(x) = 0.2x^5 + 1.3x^4 + 0.7x^3 + 1.7x^2 + 2.4x + 5$ for different modes and layers where the testing set is evaluated in the first row, with horizontal axis representing the evaluation sample, and the logarithmic loss convergence of the training in the second row, with horizontal axis representing the iteration of the optimization.

On the other hand, another outstanding result is the Gaussian quantum perceptron where only the Gaussian operator is considered without any non-Gaussianity.

It turns out that, given an initial Gaussian state, there exists a symplectic transformation that is capable of transforming the initial Gaussian state into any other Gaussian state. Then, after the QNN optimizes the Gaussian transformation represented by the symplectic matrix, the preparation of any Gaussian state is achieved.

9 Conclusions

From the results it is deduced that the simulated quantum neural network is perfectly designed to be able to learn, at least, any degree polynomial function with a very small error whenever it has more than one layer. With a single layer, the QNN is not able to learn and generalize the target function.

Also, appears like two layers are enough to learn any-degree polynomial function and that a third layer does not offer any improvement in the loss function. However, there is an improvement in the loss function convergence value when increasing the number of modes until having 5 modes.

Moreover, the QNN is capable of generalizing the objective function with a reduced number of samples, as demonstrated by the testing set results.

Given the double factorial complexity of Eq. (38) and the number of terms hugely scaling with the number of modes and layers as Eq. (64) holds, shows that non-Gaussianity is classically an intractable task.

Despite of that, the complex computational part would ideally be executed on a real quantum hardware device avoiding both the matrix exponentiations and the expectation value expression calculations.

Regarding the data re-uploading encoding strategy, the obtained results may have improved the learning in contrast with just a single input encoding, but more tasks and tests would be needed in order to affirm its real effectiveness.

10 Further work

10.1 Real tasks

The proposed QNN was classically trained and tested for supervised learning with polynomial functions of different degrees, by generating random datasets of the target functions.

One interesting application would be to train the QNN with a real task in order to test its effectiveness and study how the model could be improved. For this, different real datasets could be investigated and used to test the QNN.

10.2 Physical realization

When it comes to do the physical implementation of the optical quantum neural network, all operators used in the simulation, found in Table 1, have their physical correspondence in terms of phase shifters, beam splitters, squeezers, photon addition or subtractions and homodyne and heterodyne detectors.

Then, in order to physically implement the proposed QNN, the only thing to do is replace the classical computation of the operators by its corresponding quantum optics components that were just mentioned.

A future task would be the implementation of the QNN in a real optical quantum computer which would require a software and hardware infrastructures that are able to translate the shape of the operators obtained in the classical optimization process into the different physical components such as interferometers, squeezers, polarizers, etc.

10.3 Classical optimization improvement

As aforementioned, the number of perfect matchings scales with the double factorial of the number of ladder operators forming the observable expectation value and normalization factor expressions. In addition, the operation of matrix exponentiation is a heavy task repeated twice per layer of the QNN when building the unitary matrix which, in turn, increases the matrix dimension with the number of modes. As this process is repeated many times in the QNN training stage, the simulation slows down considerably/notably as the number of modes and/or layers of the QNN increase.

Some future challenge to tackle could be to reduce the number of operations with some strategy, to avoid matrix exponentiation or to execute the simulation in a high-performance environments like distributed systems or supercomputers in order to speed up the training process.

10.4 Quantum natural gradient descent

With the goal of designing a QNN where all the process is carried out with quantum mechanics, one has to get rid of the classical part which is the optimization of the training.

The point would be to develop a natural quantum gradient descent aiming to replace classical optimization algorithms by a quantum optimizer and its feasibility relies on the similarity between the Stochastic Gradient Descent formula,

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\vec{\theta}), \tag{65}$$

where each $\theta_j \in \vec{\theta}$ is one parameter to be tuned and $J(\vec{\theta})$ is the loss function to be minimized, and the definition of momentum operator

$$\hat{p}\Psi = -i\hbar \frac{\partial}{\partial x}\Psi \tag{66}$$

where x stands for the the position.

Some articles like [SIKC20] and [Yam19] already study and implement different techniques for quantum natural gradient descent.

Bibliography

[BSBN02] Stephen D Bartlett, Barry C Sanders, Samuel L Braunstein, and Kae Nemoto. Efficient classical simulation of continuous variable quantum information processes. *Physical Review Letters*, 88(9):097904, 2002.

[CFGM21] Ulysse Chabaud, Giulia Ferrini, Frédéric Grosshans, and Damian Markham. Classical simulation of gaussian quantum circuits with non-gaussian input states. *Physical Review Research*, 3(3):033018, 2021.

[Flö23] Frederik F Flöther. The state of quantum computing applications in health and medicine. *arXiv preprint arXiv:2301.09106*, 2023.

[HSM$^+$22] Chaoran Huang, Volker J Sorger, Mario Miscuglio, Mohammed Al-Qadasi, Avilash Mukherjee, Lutz Lampe, Mitchell Nichols, Alexander N Tait, Thomas Ferreira de Lima, Bicky A Marquez, et al. Prospects and applications of photonic neural networks. *Advances in Physics: X*, 7(1):1981155, 2022.

[JC19] SK Jeswal and S Chakraverty. Recent developments and applications in quantum neural network: A review. *Archives of Computational Methods in Engineering*, 26:793–807, 2019.

[KBA$^+$19] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[Meh21] Bernhard Mehlig. *Machine learning with neural networks: an introduction for scientists and engineers*. Cambridge University Press, 2021.

[PSCLGFL20] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

[Ser17] A. Serafini. *Quantum Continuous Variables: A Primer of Theoretical Methods*. CRC Press, 2017.

[SIKC20] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[SvdMA$^+$21] Frank HB Somhorst, Reinier van der Meer, Malaquias Correa Anguita, Riko Schadow, Henk J Snijders, Michiel de Goede, Ben Kassenberg, Pim Venderbosch, Caterina Taballione, Jörn Epping, et al. Quantum simulation of thermodynamics in an integrated quantum photonic processor. *arXiv preprint arXiv:2201.00049*, 2021.

[WPGP$^+$12] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2):621, 2012.

[WST$^+$23] Mattia Walschaers, Bhuvanesh Sundar, Nicolas Treps, Lincoln D Carr, and Valentina Parigi. Emergent complex quantum networks in continuous-variables non-gaussian states. *Quantum Science and Technology*, 8(3):035009, 2023.

[Yam19] Naoki Yamamoto. On the natural gradient for variational quantum eigensolver. *arXiv preprint arXiv:1909.05074*, 2019.

A Symplectic-orthogonal and unitary isomorphism

First, let's start with the unitary group denoted by $U(N)$ which is the group of $N \times N$ unitary matrices. Among other properties, any unitary matrix obeys

$$UU^\dagger = \mathbb{1}_N, \forall U \in U(N). \tag{67}$$

If the unitary matrix is represented instead as the sum of its real and imaginary parts $U = X + iY$ with $X, Y \in \mathbb{R}^{N\times N}$, the aforementioned expression (67) turns into the following one

$$UU^\dagger = (X+iY)(X^T - iY^T) = (XX^T + YY^T) - i(XY^T - YX^T) = \mathbb{1}_N, \tag{68}$$

where the employed fact is that the Hermitian transpose of a real matrix is the same as its transpose.

Next, the orthogonal group represented by $O(N)$ is the group of $N \times N$ orthogonal matrices fulfilling the property

$$OO^T = O^TO = \mathbb{1}, \forall O \in O(N) \tag{69}$$

for any orthogonal matrix O.

Finally, the real symplectic group symbolized with $Sp(2N, \mathbb{R})$ is the group of $2N \times 2N$ real symplectic matrices. A real symplectic matrix S is a real matrix that preserves the real anti-symmetric symplectic form J like

$$SJS^T = S^TJS = J \text{ with } J = \begin{pmatrix} \mathbb{0}_N & \mathbb{1}_N \\ -\mathbb{1}_N & \mathbb{0}_N \end{pmatrix}. \tag{70}$$

As Appendix B from [Ser17] shows, any symplectic matrix S can be written as a matrix exponential of the form $S = e^{JHt}$ where J is the symplectic form, H is a real and symmetric matrix and t is a real parameter.

Now, Imposing together the conditions of the symplectic group $Sp(2N, \mathbb{R})$ and the orthogonal group $O(2N)$, both for matrices $2N \times 2N$, gives the orthogonal compact subgroup which is represented as $K(N) = Sp(2N, \mathbb{R}) \cap O(2N)$ [Ser17].

To do this, let's write a general $2N \times 2N$ matrix Q divided in blocks of four $N \times N$ matrices with the form

$$Q = \begin{pmatrix} X & Y \\ W & Z \end{pmatrix} \tag{71}$$

and let's apply the symplectic and orthogonal conditions to Q.

On the one hand, enforcing the condition of the symplectic form preservation to Q seen in Eq. (70) results into the following restrictions

$$QJQ^T = J \iff \begin{pmatrix} XY^T - YX^T & XZ^T - YW^T \\ WY^T - ZX^T & WZ^T - ZW^T \end{pmatrix} = \begin{pmatrix} \mathbb{0}_N & \mathbb{1}_N \\ -\mathbb{1}_N & \mathbb{0}_N \end{pmatrix}, \tag{72}$$

$$Q^TJQ = J \iff \begin{pmatrix} X^TW - W^TX & X^TZ - W^TY \\ Y^TW - Z^TX & Y^TZ - Z^TY \end{pmatrix} = \begin{pmatrix} \mathbb{0}_N & \mathbb{1}_N \\ -\mathbb{1}_N & \mathbb{0}_N \end{pmatrix} \tag{73}$$

On the other hand, when imposing the orthogonality condition (69) to Q, the derived constraints turn to be

$$Q^TQ = \mathbb{1}_{2N} \iff \begin{pmatrix} X^TX + W^TW & X^TY + W^TZ \\ Y^TX + Z^TW & Y^TY + Z^TZ \end{pmatrix} = \begin{pmatrix} \mathbb{1}_N & \mathbb{0}_N \\ \mathbb{0}_N & \mathbb{1}_N \end{pmatrix}, \tag{74}$$

$$QQ^T = \mathbb{1}_{2N} \iff \begin{pmatrix} XX^T + YY^T & XW^T + YZ^T \\ WX^T + ZY^T & WW^T + ZZ^T \end{pmatrix} = \begin{pmatrix} \mathbb{1}_N & \mathbb{0}_N \\ \mathbb{0}_N & \mathbb{1}_N \end{pmatrix}. \tag{75}$$

Putting all these constraints together in order to make $Q \in K(N) = Sp(2N, \mathbb{R}) \cap O(2N)$, i.e. Q to be symplectic and orthogonal, it is known that $X = Z$ and $W = -Y$ and the most generic form of Q become

$$Q = \begin{pmatrix} X & Y \\ -Y & X \end{pmatrix} \text{ with } \begin{cases} XY^T - YX^T = \mathbb{0}_N \\ XX^T + YY^T = \mathbb{1}_N \end{cases}. \tag{76}$$

In the end, recalling the unitary expression in terms of its real and imaginary parts derived in (68), one may notice that it matches exactly with the conditions from (76). This important equivalence affirms an isomorphism between $K(N)$ and $U(N)$, thus allowing the representation of Q as a unitary matrix U_Q by applying (7) like

$$\bar{U} Q \bar{U}^\dagger = \begin{pmatrix} X - iY & \mathbb{0}_N \\ \mathbb{0}_N & X + iY \end{pmatrix} = \begin{pmatrix} U_Q^* & \mathbb{0}_N \\ \mathbb{0}_N & U_Q \end{pmatrix}. \tag{77}$$

B QNN algorithm

The initialization block is only executed once for all iterations, i.e. N and the final expression of the target expectation value are fixed values for one given a QNN.

Algorithm 1 One iteration of QNN

```
init:
    $N \leftarrow$ Num inputs
    ExpValExpression $\leftarrow$ 38 Form-like expression
$Z_{\text{IN}} = \left(\bigoplus_{j=1}^{N} x_j\right) \oplus \left(\bigoplus_{j=1}^{N} x_j^{-1}\right)$
$\sigma \leftarrow Z_{\text{IN}} \mathbb{1} Z_{\text{IN}}$
$Q_1 \leftarrow$ RandomPassiveOptics$(N)$
$Q_2 \leftarrow$ RandomPassiveOptics$(N)$
$Z \leftarrow$ Random $(N)$
$\sigma \leftarrow Q_2 Z Q_1 \sigma Q_1^T Z Q_2^T$
for i in layer [2:N] do
    $Q_1 \leftarrow$ RandomPassiveOptics$(N)$
    $Q_2 \leftarrow$ RandomPassiveOptics$(N)$
    $Z \leftarrow Z_{\text{IN}}$
    $S_i \leftarrow Q_2 Z Q_1$
ExpVal $\leftarrow 0$
for (i in All perfect matchings of ExpValExpression) do
    ExpVal $+= S_i * i$ as held in Eq. (50)
procedure RandomPassiveOptics$(N)$
    $M \leftarrow$ Random $(N \times N)$
    $H \leftarrow M + M^T$
    $U \leftarrow e^{iH}$
    $Q \leftarrow \bar{U}^\dagger \begin{pmatrix} U^* & \mathbb{0}_N \\ \mathbb{0}_N & U \end{pmatrix} \bar{U}$
    return $Q$
```