Quantum machine learning with bosonic modes

Julien Dudas

To cite this version:

Julien Dudas. Quantum machine learning with bosonic modes. Quantum Physics [quant-ph]. Université Paris-Saclay, 2025. English. NNT : 2025UPASP083 . tel-05383369

HAL Id: tel-05383369
https://theses.hal.science/tel-05383369v1

Submitted on 26 Nov 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Quantum machine learning with bosonic modes

Apprentissage automatique quantique avec des modes bosoniques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°564 Physique en Île-de-France (PIF)
Spécialité de doctorat: Physique
Graduate School : Physique. Référent : faculté de sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire Albert Fert, (CNRS, Thales, Université Paris-Saclay)**, sous la direction de **Julie GROLLIER**, Directrice de recherche, Laboratoire Albert Fert, et le co-encadrement de **Danijela MARKOVIC**, Chargée de recherche, Laboratoire Albert Fert.

Thèse soutenue à Paris-Saclay, le 30 octobre 2025, par

Julien DUDAS

Composition du jury

Membres du jury avec voix délibérative

Xavier WAINTAL Professeur, CEA Grenoble	Président
Mauro PATERNOSTRO Professeur, Universita di Palermo	Rapporteur & Examinateur
Valentina PARIGI Professeure, Sorbonne Université	Rapportrice & Examinatrice
Géraldine HAACK Maître enseignant recherche, Université de Genève	Examinatrice

THESE DE DOCTORAT

NNT : 2025UPASP083

université PARIS-SACLAY

ÉCOLE DOCTORALE
Physique en
Île-de-France (PIF)

Titre: APPRENTISSAGE AUTOMATIQUE QUANTIQUE AVEC DES MODES BOSONIQUES

Mots clés: Apprentissage, Quantique, Bosonique

Résumé: Les systèmes quantiques ont le potentiel d'améliorer les techniques informatiques classiques de pointe, grâce à des propriétés telles que la superposition et l'intrication. Leur combinaison avec des cadres d'intelligence artificielle pourrait améliorer de nombreux domaines du traitement des données quantiques, tels que la reconnaissance automatique des états quantiques et le contrôle des dispositifs quantiques. Malgré des résultats prometteurs obtenus avec les réseaux de neurones quantiques, des défis majeurs subsistent : (i) la mise en oeuvre d'architectures comportant un grand nombre de neurones et de paramètres entraînables sur le matériel quantique actuel, et (ii) l'entraînement fiable de ces modèles, où des problèmes tels que les 'barren plateau' dans les fonction de coût peuvent supprimer les gradients et entraver l'apprentissage. En effet, la résolution de tâches complexes nécessite un grand nombre de neurones densément connectés par de nombreux paramètres entraînables, ce qui est difficile à réaliser sur du matériel quantique.

Dans cette thèse, nous envisageons la mise en oeuvre de réseaux de neurones quantiques avec des modes bosoniques couplés. Nous obtenons un grand nombre de neurones en considérant les probabilités d'état de Fock comme caractéristiques de sortie, et des paramètres entraînables en couplant paramétriquement les modes, grâce à différents processus de mélange à trois ondes tels que la conversion cohérente de photons et le squeezing entre deux modes.

Afin d'entraîner le système quantique, nous explorons deux approches : (i) le calcul par réservoir, qui contourne l'entraînement des paramètres physiques, et (ii) l'entraînement direct des couplages paramétriques par descente de gradient. La dynamique intrinsèque des modes bosoniques étant linéaire, la non-linéarité ne provient que des mesures d'états de Fock. Nous constatons que cette non-linéarité induite par la mesure est remarquablement expressive : en utilisant seulement deux oscillateurs quantiques, nous pouvons extraire un ensemble suffisant de caractéristiques non linéaires pour apprendre des tâches qui nécessitent généralement environ 20 neurones dans un réseau classique, notamment la classification de signaux sinus/carré et la prédiction de séries temporelles Mackey-Glass.

Cependant, le calcul quantique par réservoir se heurte à deux limites principales : comme tous les poids entraînables résident uniquement dans le post-traitement classique des résultats de mesure, obtenir de bonnes performances nécessite de mesurer un grand nombre de caractéristiques, et son expressivité est limitée par son architecture monocouche. Pour remédier à cela, nous entraînons directement les paramètres de mélange à trois ondes; pour une simulation efficace, nous limitons la dynamique interne aux modes gaussiens et utilisons la rétro-propagation pour l'optimisation. Nous montrons que cette approche apprend efficacement : bien que le nombre de paramètres entraînables évolue linéairement avec le nombre de modes, le nombre de caractéristiques mesurées requis baisse par rapport au calcul par réservoir, l'expressivité augmente et, avec le même nombre de modes, nous résolvons des tâches plus difficiles que la base de référence du réservoir.

Cette thèse a démontré que les réseaux de neurones quantiques bosoniques constituent une voie prometteuse vers un apprentissage efficace sur le plan matériel et a fourni des outils théoriques permettant de comprendre comment ils apprennent, et améliorer leurs performances. À l'avenir, l'addition dans l'Hamiltonien de non-linéarités d'ordre supérieur telles que les processus Kerr permettrait de pousser le système au-delà du régime gaussien, libérant ainsi des ressources non gaussiennes et un espace de caractéristiques effectif plus vaste. Dans une perspective plus large, il sera important d'étudier leur susceptibilité aux barren plateaus et d'identifier les conditions dans lesquelles le dimensionnement du gradient reste favorable.

Title: QUANTUM MACHINE LEARNING WITH BOSONIC MODES

Keywords: Learning, quantum, bosonic

Abstract: Quantum systems have the potential to improve state of the art classical computing techniques, thanks to properties such as superposition and entanglement. Combining them with machine learning frameworks could enhance many domains of quantum data processing such as automatic recognition of quantum states and the control of quantum devices. Despite early promising results with quantum neural networks (QNNs), key challenges remain: (i) implementing architectures with a large number of neurons and trainable parameters on current quantum hardware, and (ii) reliably training such models, where issues like barren plateaus in the loss landscape can suppress gradients and hinder learning. Indeed solving complex tasks requires a large number of neurons densely connected by many trainable parameters, which is hard to achieve on quantum hardware.

In this thesis, we consider the implementation of quantum neural networks with coupled bosonic modes. We obtain a large number of neurons by considering the Fock state probabilities as output features, and trainable parameters by coupling the modes parametrically, through simultaneous three-wave mixing processes such as coherent photon conversion and two-mode squeezing.

In order to train the quantum system, we explore two approaches: (i) quantum reservoir computing, which sidesteps training of the physical parameters, and (ii) direct training of the parametric couplings via gradient descent. Because the intrinsic dynamics of coupled bosonic modes are linear, nonlinearity arises solely through Fock basis measurements. We find that this measurement-induced nonlinearity is remarkably expressive: using only two quantum oscillators, we can extract a sufficient set of nonlinear features to learn tasks that typically require on the order of 20 neurons in a classical network, including sine/square classification and Mackey-Glass time series prediction.

However, quantum reservoir computing faces two main limitations: because all trainable weights reside solely in the classical post-processing of measurement outcomes, achieving good performance typically requires measuring a large number of features, and its expressivity is constrained by its essentially single-layer architecture. To address this, we train the three-wave mixing parameters directly; for simulation efficiency we restrict the internal dynamics to Gaussian modes and use backpropagation for end-to-end optimization. We show that this approach learns effectively: although the number of trainable parameters scales only linearly with the number of modes, the number of measured features needed is reduced compared to reservoir computing, the expressivity is increased, and with the same number of modes we solve harder tasks than the reservoir baseline.

This thesis has shown that bosonic quantum neural networks are a promising route to hardware-efficient learning and has provided theoretical tools to understand how they learn and how to improve their performance. Looking ahead, enriching the Hamiltonian with higher-order nonlinearities such as Kerr, cross-Kerr or engineered multi-photon processes, would push the system beyond the Gaussian regime, unlocking non-Gaussian resources and a larger effective feature space. As a further perspective, it will be important to probe their susceptibility to barren plateaus and identify conditions under which gradient scaling remains favorable.

Acknowledgments

Durant cette thèse, j'ai eu l'opportunité inestimable d'être accueilli dans un environnement très bienveillant au Laboratoire Albert Fert, qui cultive à la fois la curiosité scientifique et l'entraide entre les équipes de recherche et leurs membres. Je remercie naturellement mon encadrante Danijela Marković, qui a été d'une patience plus qu'exemplaire avec l'étourdi que je peux être, et qui a toujours sû bien me guider dans le monde vaste et parfois intimidant de la recherche académique. Le sujet de notre recherche étant au carrefour de multiples disciplines différentes, j'ai toujours été impressionné par ta culture inépuisable et ta curiosité personnelle qui s'étend largement au delà de tes occupations professionnelles. Il serait un crime de ne pas aussi remercier mon collègue thésard Baptiste Carles. Je ne saurais compter le nombre de problèmes que nous avons résolu dans nos recherches respectives, grâce à nos discussions lors de tes visites spontanées de mon bureau. J'ai beaucoup aimé la dynamique que nous avons dans notre trio de recherche, et je pouvais toujours compter sur vous pour remettre en question mes idées préconçues sur la physique. Je remercie également Elie Gouzien, avec lequel les discussions scientifiques ont été très importantes pour la direction de ma recherche. Je tiens également à remercier les membres du jury, pour avoir consacré une part de leur temps à mon travail de thèse.

Durant l'intégralité de ma thèse je n'ai jamais été délogé de mon bureau, anciennement un placard à balais à côté de la salle café, mais qui est devenu mon QG bien aimé. Je remercie chaleureusement mes collègues de bureau Julien Berthomier et Clara Zederman, qui ont supporté mes musiques bizzares durant plus d'un an. Je remercie également mes anciens collègues de bureau Hadi Hassan et Gabriel Lazrak qui mettaient la bonne humeur. Merci à tout le groupe neuromorphique dont Julie Grollier, Frank Mizrahi, Dedalo Sanz Hernandez, Arnaud de Riz, Xing Chen, Dongshu Liu, Théophile Rageau, Mohamed Menshawy, Bibekananda Paikaray, Filip Osana, Yiming Wei, David Zehner, Erwan Plouet et Olivier Faure. Nos réunions de groupe étaient riches en réflexion et échange, et vous étiez toujours disponibles pour mes questionnements scientifiques. Je remercie l'équipe administrative du laboratoire, qui a facilité mes missions: Anne Dussart, Florence Hamet, Zelia de Figuereido et Ylliam Auerbach.

Je remercie les professeurs qui durant ma scolarité ont sû m'insuffler la passion des sciences. Pour leur encadrement exemplaire durant mes stages de L3 je remercie Kevin Vynck et Philippe Lalanne, ainsi que Vincent Jeudy et André Thiaville pour mon stage de L1 pendant la période Covid.

Depuis le début de mes études supérieures, je réalise ma chance immense d'être aussi bien entouré, que ce soit au niveau familial ou amical. Je ne serais sans doute pas la même personne sans mes amis de longue date de la ville d'Antony. Je remercie mes amis du collège (et même primaire) Pierre Pakhomov, Yukio Chapuis, Damian Contant, et Geoffrey Gobert d'être toujours aussi géniaux, chacun à leur manière. Rencontrés durant séjour écourté au lycée franco-allemand, je remercie Bogdan Mihailescu et Mathilde Mihailescu. Puis au lycée Descartes j'ai eu la chance de tomber dans le groupe de Léon-Gabriel Edouard-Rambaud, Désirée Varella, Matthias Dhellin, Julia Wagner, Jérémie Besson et Wenfan Xia, que je remercie pour m'avoir accepté malgré la personne souvent bizzare que je suis. Merci aussi à Jérôme Ruiz qui est un gars très cool.

Je n'ai malheureusement pas gardé contact avec mes camarades de prépa, mais je me rappellerais toujours leur bienveillance et la bonne ambiance qui régnait malgré la difficulté de ces deux années. Je remercie ma mère qui veillait encore sur moi malgré mon âge majeur et mon père qui a toujours éveillé ma curiosité en sciences. Grâce à eux, j'ai eu un vécu meilleur de la prépa que bien d'autres gens. Dédicace particulière aux soirées passées avec Jérémie Besson à regarder des animes en mangeant le dîner. Une fois entré à l'ENS Cachan, j'ai eu le plaisir de rencontrer le club jeux vidéo qui m'a permis de finalement réaliser un de mes rêves, de participer à des compétitions de jeux vidéos. Merci à Abd, Random, Nounours, Mammouth, Kyussey, le Shibateur, Meracx et les autres membres, avec qui j'ai vécu des années d'études absolument stellaires. Bien que j'ai perdu contact avec la totalité d'entre eux, je remercie la communauté de Smash Bros. de Bordeaux, qui m'ont aidé à développer mon aisance sociale. En organisant un tournoi entre écoles, j'ai aussi fait connaissance avec un de mes meilleurs amis actuels auquel je suis très reconnaissant, Thomas von Ascheberg. Dans ma promo de l'ENS, je remercie mes amis Raphaël Dubourg, Léon Carde, Guillaume Brouillon, Lucas Bugaud, Léonard Lehoucq et Léo Noirot qui ont partagés avec moi de bons moments en amphi, et de pires moments en partiels. Je remercie Valentine Fresse et Lucia Xu, avec qui j'espère faire encore plein de voyages en tout genre. Rencontrés durant mon année de césure en licence d'informatique, je remercie Kilian Perlin, Théo Manea, Léo Porte, Fred, Trevis Morvany, Fadma Amiche, Nao Hanus, Brice, Victor Tran, Louis Haurie et Raphaelle Couret qui m'ont appris à plus lâcher prise.

Merci à Diane et Maylis, ma soeur et mon frère adorés. Merci à ma famille étendue très nombreuse, qui ont toujours crû en moi. Finalement, merci à Tiffany Bourdier, que j'ai rencontré quelques mois avant le début de ma thèse, et que j'aime de tout mon coeur.

Contents

Chapter 1

Introduction

The brain is incredibly efficient at processing information and performing different tasks. Aside from logical thought, it handles many different kinds of sensory inputs, and can solve tasks without being explicitly programmed to do so. Desiring to replicate and even surpass the brain's computing power, scientists have developed learning models inspired from biological neurological systems: this was the birth of machine learning.

A biological neuron is a non-linear oscillator, receiving electrical spike signals from other neurons through synapses. Each received spike contributes to charging its membrane, and to its own voltage. When this voltage exceeds a threshold, the neuron discharges by emitting a spike signal itself. Synapses can excite or inhibit the signal they transmit, and learning is associated with a reinforcement or weakening of their strengths. Furthermore, imaging brains with electro-encephalograms and magnetic fields has revealed that phenomena such as synchronization of oscillations of different neurons in the brain are associated to pattern recognition [1, 2, 3, 4].

In order to both understand biological information processing and find improvements to current machine learning methods, dynamic models of neural networks are being investigated [5]. Furthermore, early developments in machine learning were rooted in physics-inspired models such as Hopfield networks [6] and Boltzmann machines [7].

However, the success of deep learning [8] on various image recognition or language processing tasks has led to widespread adoption of software-based machine learning models such as artificial neural networks (ANN's) [9] which implement neurons as non-linear functions, and synapses as static, scalar values. Many algorithmic advances contributed to this shift, such as the adoption of deep neuron layers to increase the expressivity (i.e the ability to fit an arbitrary function) of learning models, and the back-propagation algorithm [10], which allows efficient analytical computations of gradients in software. These computed gradients enable model training through gradient descent rules.

Despite their ever-growing influence on technology, software neural network models exhibit some limitations. Contrary to brains which both store memory in the synapses and process information, the von Neumann architecture of conventional computers separates their memory and computing units. The resulting back-and-forth travel of data between these units when training ANN creates significant energy and latency overheads: this phenomenon is known as the von Neumann bottleneck [11]. This bottleneck combined with the use of ANN to process ever larger datasets has resulted in increasingly energy-intensive training of modern learning models [12], so much that Amazon and Google are now investing in nuclear power plants to this end alone [13]. In stark contrast, the human brain operates with remarkable energy efficiency, consuming merely around $20\,\mathrm{W}$. These considerations have motivated the exploration of energy-efficient neuromorphic systems implementing ANN on dedicated physical hardware: this field is called neuromorphic computing. Different platforms have been studied for this use, such as memristor crossbar arrays in electronics [14], spatial light modulators in photonics [15], and spintronic devices [16] to name just a few.

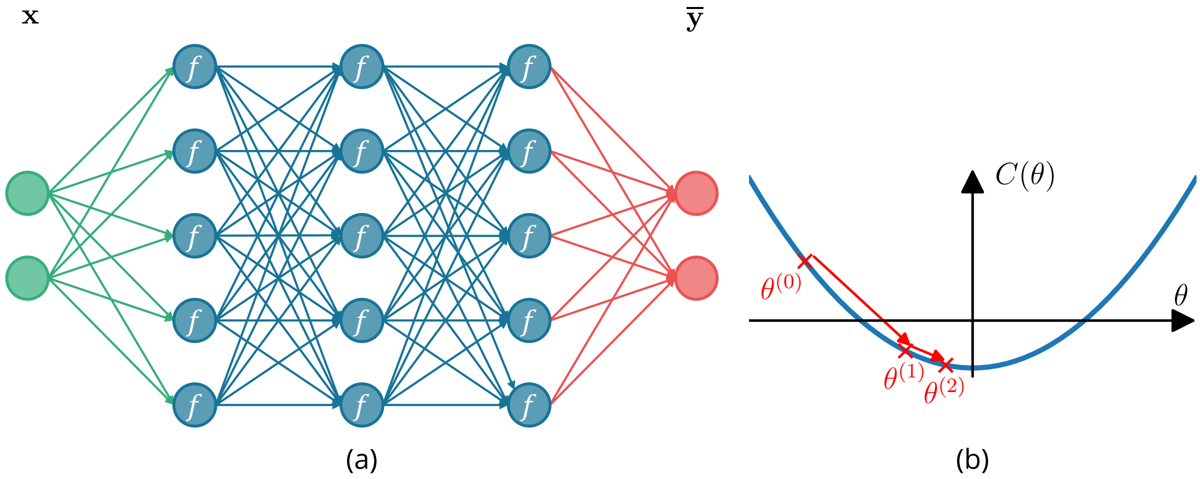


Figure 1.1: (a) Sketch of a Multi-Layer Perceptron (MLP) with three hidden layers. The 2-dimensional inputs $\mathbf{x}$ (green circles) are fed forward into the hidden layers (blue circles) with linear synaptic connections (arrows), up to the two-dimensional prediction $\bar{\mathbf{y}}$ (red circles). Each hidden layer contains 5 neurons, which apply a non-linear activation function f. The weights of synaptic connections make up the model parameters. (b) Illustration of the gradient descent algorithm, when optimizing a cost function $C(\theta)$ with respect to a one-dimensional parameter θ. Starting from $\theta^{(0)}$, the parameter is iteratively updated (red crosses) with the rule $\theta^{(n+1)} = \theta^{(n)} - \gamma\nabla_\theta C(\theta^{(n)})$, where γ is a learning rate. The algorithm converges to the minimum of this function, as the closer it gets, the smaller the gradient $\nabla_\theta C$ becomes.

We illustrate how artificial neural networks are trained with forward passes to compute predictions and gradient-based optimization, on the example of a Multi-Layer Perceptron (MLP) depicted in Fig. 1.1a. Neurons applying a non-linear activation function

f are grouped by layers, which are fully connected to each other through linear synaptic connections. In a forward pass, the first layer encodes a vector input $\mathbf{x}$, then passes through multiple hidden layers, and the last layer yields the network prediction $\bar{\mathbf{y}}$, which is used to compute a cost function C. The cost function continuously quantifies the difference between the target and the prediction: the goal of a learning algorithm is to minimize it, with respect to the model parameters.

In artificial neural networks, the linear weights of synaptic connections are the model parameters. The cost function $C(\boldsymbol{\theta}, \mathbf{x})$ depends on a set of network parameters $\boldsymbol{\theta}$ and inputs $\mathbf{x}$. Gradient descent optimization is based on the fact that extrema of a function have zero gradient. It has proven its efficiency in training large deep learning models such as GPT-4 [17] or ResNet [18]. To minimize a cost function $C(\theta, \mathbf{X})$ with respect to a dataset $\mathbf{X} = \{\mathbf{x}\}$, an instance of gradient descent algorithms called stochastic gradient descent (SGD) iteratively updates the initial set of parameters $\boldsymbol{\theta}^{(0)}$ according to

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \gamma \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(n)}, \mathbf{x}), \tag{1.1}$$

with γ a learning rate defined between 0 and 1, and the input $\mathbf{x}$ is chosen randomly from the dataset. This update rule is shown in Fig. 1.1b with a cost function depending on a one-dimensional parameter (for visual clarity), and no input. Gradients $\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(n)}, \mathbf{x})$ are computed with the back-propagation algorithm. It is the most common algorithm in machine learning, and will be explained in Section 2.1.3.

On the other hand, it seems like quantum devices have the potential to shape the future in many fields of computing. The notion of using finely controlled quantum systems for computation [19, 20] was conceived during the second quantum revolution in the 1980s, when experiments demonstrated measurements and manipulation of individual quantum systems [21]. Although the first applications considered for quantum computers focused on simulating physical systems, landmark results by Peter W. Shor and Lov K. Grover demonstrated that quantum computers could also outperform classical computers in certain algorithmic tasks, respectively solving integer factorization [22] and unstructured search [23]. This diversity of potential applications combined with the gradual mastery of experimental quantum physics in the past decades has driven intense interest in building useful quantum computers, in academic and industrial sectors alike.

Consequently, researchers have wondered if combining quantum devices and machine learning could produce interesting results. This thesis falls within the context of quantum machine learning (QML), a research field that has been quickly developing for the past 10 years.

1.1 Quantum machine learning

The prospect of quantum machine learning lies in the exponential growth of the state space (Hilbert space) of quantum systems with the number of its constituents, as illustrated in Fig. 1.2 for the case of a qubit ensemble. In contrast to classical neuromorphic computing where a single device corresponds to a neuron and the coupling between two devices to a synapse, in the quantum realm, basis states can be considered as neurons, and the hopping probabilities between states as synapses. This change of paradigm provides a feature space (a feature space is a conceptual environment where each dimension represents a specific feature of the data being analyzed or used in machine learning models) that is exponential in the size of the physical system. Due to this property, it is hoped that learning on quantum systems may produce learning models with higher expressivity.

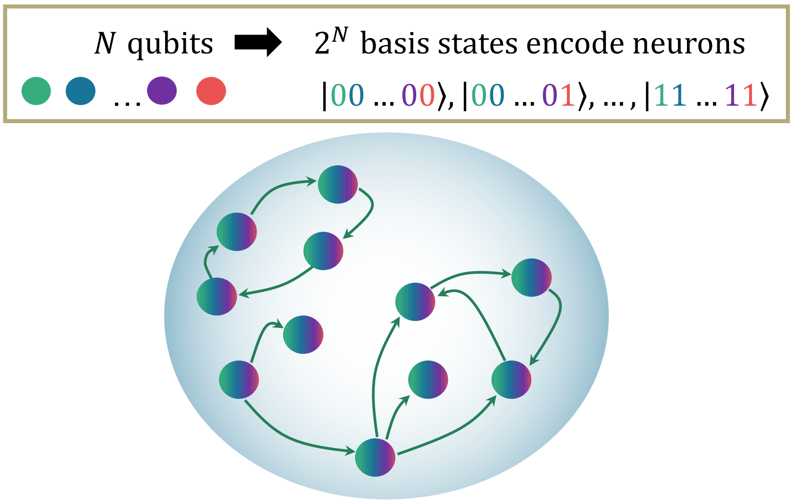
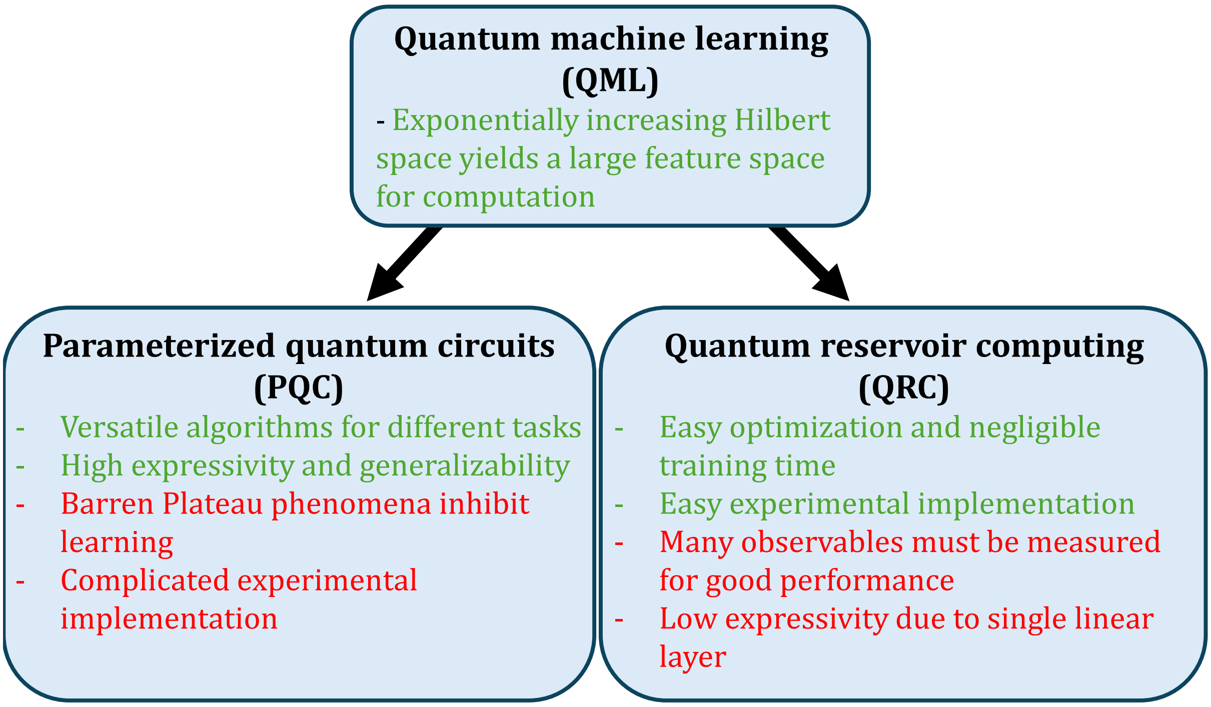

N qubits
2^N basis states encode neurons
$|00\ldots00\rangle, |00\ldots01\rangle, \ldots, |11\ldots11\rangle$

Figure 1.2: Sketch of the exponential growth of the Hilbert space. The Hilbert space is represented by the big light blue circle, and small multi-colored circles correspond to joint qubit states. Arrows linking them symbolize hopping probability amplitudes. The number of basis states scales as 2^N, where N is the number of qubits.

Moreover, multiple empirical and theoretical studies have shown that quantum learning models tend to have better generalization capacity (i.e the ability to perform well on unseen data) than classical models [24, 25, 26]. Improving the generalization capacity is important in applications where large datasets corresponding to real-life data are difficult to obtain.

Finally, an interesting perspective for quantum neuromorphic computing is learning on quantum data. Indeed, providing quantum data to a classical neural network requires a measurement in order to obtain a classical representation of the quantum state. With quantum neural networks, this would be simplified as directly processing the quantum data with a quantum neuromorphic device could be less resource-intensive [27, 28, 29, 30]. This is particularly useful in the context of rapidly evolving quantum technologies, where a large amount of quantum data needs to be processed efficiently [31].



Quantum machine learning (QML)

- Exponentially increasing Hilbert space yields a large feature space for computation

Parameterized quantum circuits (PQC)

- Versatile algorithms for different tasks
- High expressivity and generalizability
- Barren Plateau phenomena inhibit learning
- Complicated experimental implementation

Quantum reservoir computing (QRC)

- Easy optimization and negligible training time
- Easy experimental implementation
- Many observables must be measured for good performance
- Low expressivity due to single linear layer

Figure 1.3: Diagram comparing advantages (green) and disadvantages (red) in two different approaches of quantum machine learning. Parameterized quantum circuits implement learning by training rotation gate angles, while quantum reservoirs use an untrained quantum system to project the input data into a higher dimensional space, which is then classified by linear regression on a classical computer.

Two main approaches have emerged to quantum machine learning, with different optimization methods: in parameterized quantum circuits (PQC) the physical parameters are trained, while in quantum reservoir computing (QRC) an untrained quantum system projects input data into a higher dimensional space, which is then classified with linear regression executed on a classical computer.

1.1.1 Parameterized quantum circuits

Optimizing a hardware device for a machine learning task involves executing forward passes (where the device issues a prediction given an input) whose predictions are used to compute the cost function $C(\boldsymbol{\theta})$. The cost function is minimized with respect to device parameters $\boldsymbol{\theta}$, which correspond to the model's trainable parameters. While in classical software neural networks the first neuron layer encodes inputs, and synaptic weights

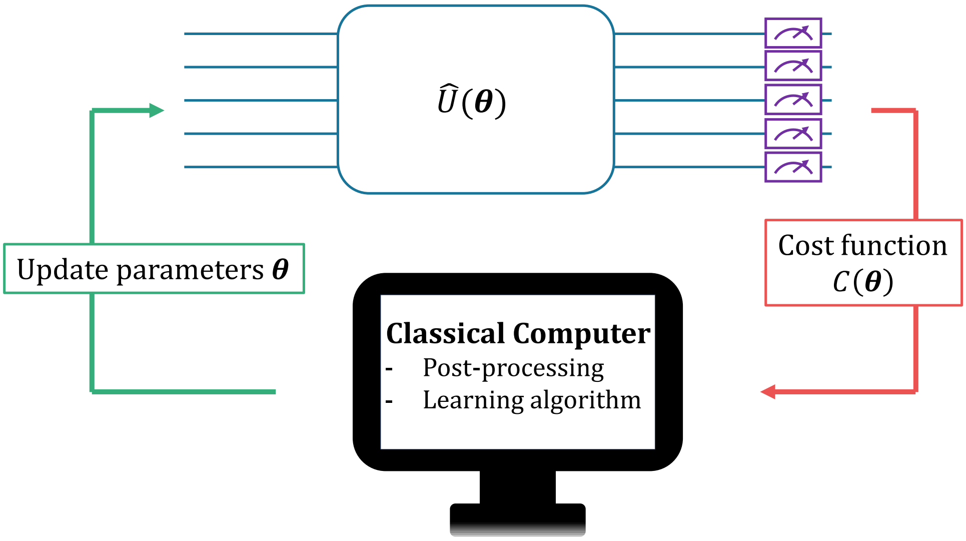


Figure 1.4: Schematic of the hybrid optimization of a parameterized quantum circuit (PQC). Each blue line corresponds to a qubit. To evaluate a cost function $C(\boldsymbol{\theta})$, gates of parameters $\boldsymbol{\theta}$ transform the circuit state, with unitary operation $\hat{U}(\boldsymbol{\theta})$. Then $C(\boldsymbol{\theta})$ is computed from the observables measured on the circuit (purple boxes). Parameter updates are computed using classical optimization tools on a conventional classical computer, which are re-injected into the gates for the next forward pass.

are learnable parameters, in quantum circuits, the unitary gate parameters are used to both encode inputs and provide learnable parameters. A prediction is obtained from measured observables, and a cost function is inferred from it. Using a conventional computer, classical optimization tools like gradient descent are used to update the gate parameters. Such hybrid optimization in parameterized quantum circuits (PQC) [32] is illustrated in Fig. 1.4.

In the first implementations of PQC, gradients were estimated using their first degree variational approximation

$$\nabla_\theta C \approx \frac{C(\theta + d\theta) - C(\theta)}{d\theta}. \tag{1.2}$$

This method was imprecise and consequently made it difficult for optimization algorithms to converge to a solution. To work around this issue, parameter shift rules have been found, allowing to compute analytical gradients via two PQC forward passes [33, 34]. For instance, the gradient with respect to the parameter θ of a rotation gate is [35]

$$\nabla_\theta C(\theta) = \frac{1}{2}\left(C\left(\theta + \frac{\pi}{2}\right) - C\left(\theta - \frac{\pi}{2}\right)\right). \tag{1.3}$$

Parameter shift rules have improved the precision of estimated gradients in PQC, which in turn has enhanced their fine-tuning capabilities [36].

However, the large Hilbert space promising high expressivity of parameterized quantum circuits is also at the source of problems in their optimization. Indeed when scaling up the size of highly expressive PQC, measured gradients through parameter shift rule vanish exponentially with the number of quantum devices [37]. This phenomenon is known as barren plateaus (BP), and makes training PQC infeasible at scale for the moment. Research is very active in looking for PQC frameworks which avoid vanishing gradients at scale [38, 39, 40, 41].

1.1.2 Quantum reservoir computing

In this context, quantum reservoir computing was proposed by Ref. [42] for learning with quantum systems. It is directly inspired by reservoir computing (RC), a framework developed to address challenges in training classical artificial neural networks when tackling time-dependent tasks [43, 44]. Reservoir computing is applied in the supervised learning scenario, where the input dataset $\mathbf{X}$ is associated to a target dataset $\mathbf{Y}$. The goal of training is to obtain a prediction $\bar{\mathbf{Y}}$ that matches the target as close as possible. In essence, a set of neurons with random and untrained synaptic connections (called the 'reservoir') are used to project $\mathbf{X}$ onto a linearly separable feature space during the forward pass, which is then classified with a linear regression.

Interestingly, the reservoir can be replaced by a physical system, in which the inputs are encoded through some drives and the outputs are obtained by measuring some observables. It has first been implemented on diverse classical physical systems, ranging from electronics [45, 46, 47, 48], optics [49, 50] and spintronics [51, 52]. In the field of neuromorphic computing with physical systems, this is particularly interesting because the back-propagation algorithm is not compatible with learning in the brain or physical devices. Indeed back-propagation requires storing the output of every neuron layer, instead of just the final one. In a reservoir computing framework, the only learned parameters are stored in a classical computer, making them trainable by any classical optimization scheme.

In QRC the physical reservoir is in the quantum regime, and its hyper-parameters are fixed and random. For instance if the reservoir is a spin ensemble, then its hyper-parameters can be spin to spin coupling strengths. During a forward pass we let its dynamics freely evolve for a set amount of time, and measuring some observables yields the reservoir's output features $\mathbf{F}(\mathbf{X})$. The prediction $\bar{\mathbf{Y}} = \mathbf{W}\mathbf{F}(\mathbf{X})$ is then learned using a linear regression, where $\mathbf{W}$ is the output weight matrix. This algorithm is illustrated in Fig. 1.5.

Outsourcing the optimization onto a software linear regression after the quantum reservoir avoids learning the quantum system parameters, and thus avoids barren plateaus. However a reservoir is in essence a single layer network. Since the expressivity of neural networks scales with their number of layers [53], the expressivity of

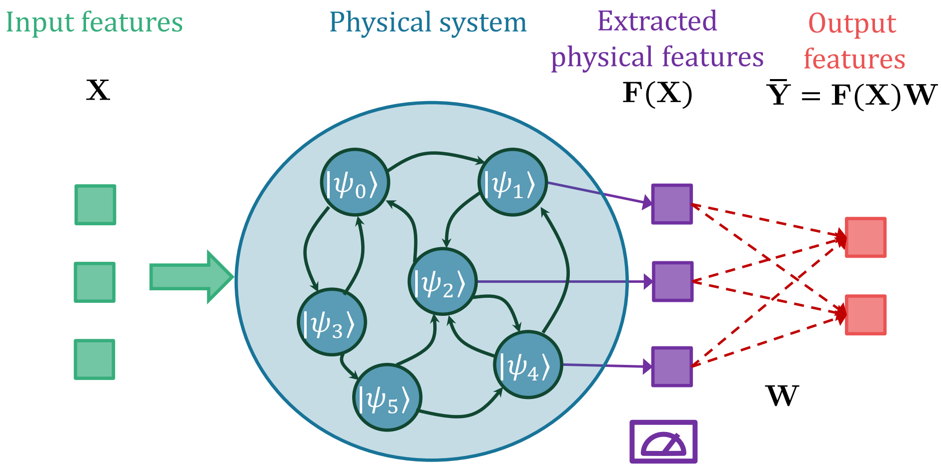


Figure 1.5: Sketch of the quantum reservoir computing algorithm. The input dataset $\mathbf{X}$ is encoded into the quantum system dynamics. The system now called reservoir freely evolves, until some observables are measured to obtain a physical feature vector $\mathbf{F}(\mathbf{X})$. Recurrent connections between the reservoir neurons (the quantum system's basis states $|\psi_k\rangle$) symbolize the memory property of the reservoir, stemming from its dynamics. Multiplication of the feature vector $\mathbf{F}(\mathbf{X})$ by a matrix $\mathbf{W}$ yields the prediction $\bar{\mathbf{Y}}$. $\mathbf{W}$ is learned through linear regression, while the reservoir's hyper-parameters are fixed and random.

quantum reservoirs is thus limited compared to parameterized quantum circuits [54].

Thus, current QRC proposals and implementations do not aim to solve all of the same tasks for which PQC were conceived. Namely, their main practical implementation proposals are for enhancing quantum sensing and tomography [28, 27, 55, 56].

1.2 Quantum reservoir computing with Fock state occupations of bosonic modes

While a part of the community focuses on learning with qubits and their gate operations, in my PhD work I have focused on bosonic modes. A bosonic mode is a quantum harmonic oscillator with bosonic statistics. Its Hamiltonian is

$$\hat{H} = \omega \hat{a}^\dagger \hat{a}, \tag{1.4}$$

where ω is its resonance frequency, and $\hat{a}$ is the field operator. The eigenstates of these quantum oscillators are Fock states denoted $|n\rangle$, where n corresponds to a number of photons. Our motivation for using bosonic modes in neuromorphic computing stems from their higher number of basis states per physical device: while qubits are two-level systems, bosonic modes have an infinite number of energy levels. In practice, we will use of the order of 10 levels per bosonic modes.

Bosonic modes can be experimentally implemented in different physical systems, such as optomechanics [57], optical ring resonators [58] and circuit quantum electrodynamics (cQED) [59]. In this thesis we will consider electromagnetic modes of superconducting resonators [60], but the results are applicable to any hardware implementing bosonic modes.

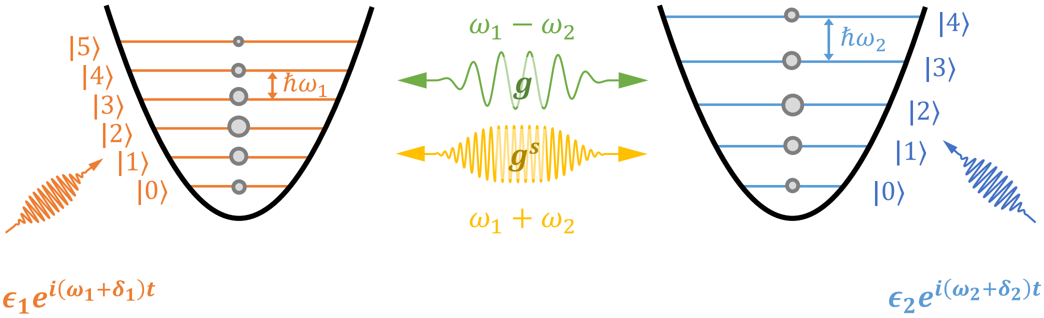


Figure 1.6: Sketch of energy levels of two bosonic modes, with respective resonance frequencies ω_1 and ω_2. We choose to apply to each mode nearly resonant drives of complex amplitudes $\epsilon_{1,2}$ and frequency detunings $\delta_{1,2}$ in order to create coherent states (gray circles). The modes can be coupled parametrically by introducing a nonlinear element, such as the Josephson junction in superconducting circuits [61, 62]. We consider a three-wave mixing that yields two coupling processes depending on the pump frequency: coherent photon conversion and two-mode squeezing, represented respectively by the green and yellow arrow. A pump tone applied at the difference of their frequencies creates photon conversion at a rate g (green arrow), while applying it to the sum of their frequencies creates two-mode squeezing at a rate g^s (yellow arrow).

We want to implement a tunable coupling between bosonic modes, in order to control their dynamic response and potentially train them as parameters of a learning model. To this end, they must first be connected by a non-linear element: in the case of cQED this can be done with a Josephson junction, but in optics non-linear crystals can also fill this role [63]. We chose in this thesis to use nearly resonant drives in order to inject photons in each individual mode, and pump tones to couple different modes at rates depending on the pump tone amplitudes and phases. Figure 1.6 illustrates the case of two coupled modes, where applying drives at the resonator frequencies ω_1 and ω_2 enables creation of coherent states, and applying three-wave mixing pump tones at frequencies $\omega_1 - \omega_2$ or $\omega_1 + \omega_2$ respectively creates photon conversion and two-mode squeezing. The combination of these phenomena results in tunable dynamics, which we leverage for neuromorphic computation.

However, the dynamics of these oscillators are linear. This poses an obstacle to their function as learning models, which require non-linearity to separate data. To deal with this issue, prior works to my PhD have proposed introducing Kerr effect [64, 65], or encoding inputs into the phase of parametric pump tones [66]. In my PhD work, we were interested in the measurement nonlinearity. We have considered measurements in the Fock state basis, which allows us to extract a large number of features. This is particularly interesting for reservoir computing where all the learning is happening on the linear layer applied to these features. Other possible nonlinear measurements include state parity measurements [67], that have the advantage of leaving the system in a superposition of states, but yield less interpretable measurements. The first part of my PhD was dedicated to determining whether measuring Fock state occupations in bosonic modes would lead to possible hardware improvements for quantum neuromorphic computers. The quantum machine learning framework I chose was QRC, because of its simplicity and ease of implementation.

In Section 4.3, we will show that nonlinearity from the Fock state occupation measurement is sufficient to solve benchmark tasks requiring non-linearity such as sine/square classification. Better yet, by comparing to the performance of classical reservoirs on the same tasks, we have shown that for the same data processing power, a smaller number of individual physical devices is required in the quantum reservoir, and a smaller number of physical quantities needs to be measured. Finally, we observe that losing quantum coherence deteriorates the reservoir performance, suggesting that it plays a key role in the exploration of the full feature space.

However, the QRC approach runs into some limitations. First, many observables still need to be measured in order to have enough output features to learn on. This poses an obstacle to experimental implementation, where measuring even a single observable is time consuming because of the stochastic nature of quantum measurements. Second, learning a single linear layer results in a limited expressivity of the reservoir, and such constraints have also been identified in classical reservoir computing [68]. While in the asymptotic case, the universal approximation theorem [69] guarantees that sufficiently large layer depth or width enables a neural network to approximate any target function, it has been found that depth generally plays a more significant role than the width [70, 71]. To illustrate this, we note that modern learning models such as GPT-4 employ around a hundred neuron layers [72].

1.3 Analog quantum neural network with Gaussian bosonic modes

In this context, I spent the second part of my PhD making a software model to train the quantum system parameters through back-propagation and gradient descent. Before starting this work, the question we aimed to answer was whether training the amplitudes, phases, and detunings of parametric couplings would be sufficient to learn on tasks, and improve on the QRC performance.

As gradient descent learning involves executing many forward passes, the simulation method for the physical system should be fast and scalable. This can be challenging for the quantum bosonic modes we want to simulate, because the exponentially large Hilbert space of quantum systems makes their classical simulation challenging. The state of an open quantum system weakly interacting with a Markovian environment [73] is modeled with a density matrix $\hat{\rho}$, obeying the Lindblad master equation

$$\begin{aligned}\frac{d\hat{\rho}}{dt} &= -\frac{i}{\hbar}[\hat{H}(t), \hat{\rho}(t)] + \sum_k \frac{1}{2}[2\hat{C}_k\hat{\rho}(t)\hat{C}_k^\dagger - \hat{\rho}(t)\hat{C}_k^\dagger\hat{C}_k - \hat{C}_k^\dagger\hat{C}_k\hat{\rho}(t)] \\ &= \mathcal{L}(\hat{\rho}),\end{aligned} \tag{1.5}$$

where $\hat{H}$ is the hermitian Hamiltonian, and $\hat{C}_k$ are jump operators describing different coupling phenomena to the environment. $\mathcal{L}$ is a superoperator describing the dynamics.

Although there exists libraries such as `Dynamiqs` [74] and `torchdiffeq` [75] that implement back-propagation into differential equation solvers, the computation time of this method scales exponentially with the number of modes and occupied basis states. This is due to the density matrix of M modes having an exponentially scaling size $n_{\mathcal{H}}$, with $n_{\mathcal{H}}$ being the Hilbert space cutoff. As a result, no more than 2-3 modes can be simulated efficiently without approximation techniques like truncated cumulants [55, 76] or the positive-P representation [77], both of which are unreliable for high non-linearities or long evolution times. One of our aims will be to process time-series datasets, meaning the quantum system will undergo long time evolution in order to remember past inputs, so we have not applied these approximation techniques. We have not applied tensor network [78, 79] methods either, as in the scope of my thesis, we were interested in Gaussian states which can be efficiently simulated for a large number of modes.

A Gaussian state has a Gaussian distribution in the phase space, as depicted in Fig. 1.7. This distribution can be fully characterized by the displacement vector $\boldsymbol{\alpha}^R \in \mathbb{C}^{2M}$ and covariance matrix $\boldsymbol{\sigma}^R \in \mathbb{C}^{2M\times 2M}$. This means that a polynomial amount of variables defines the state, making its simulation classically feasible. Inspiring ourselves from the work of Ref. [80] for efficiently simulating Gaussian state dynamics, and Ref. [81] to analytically compute Fock state occupations from Gaussian distributions, I developed a Python library enabling economical fully differentiable simulation of dynamics and Fock measurements on Gaussian states with the `PyTorch` package.

One drawback of this framework is that it cannot simulate non-linear (i.e non-quadratic) terms in the system Hamiltonian, as they create non-Gaussian states. This restricts the number of pump tones we would want to introduce in order to add more trainable parametric couplings: the only allowed ones will be coherent photon conversion and

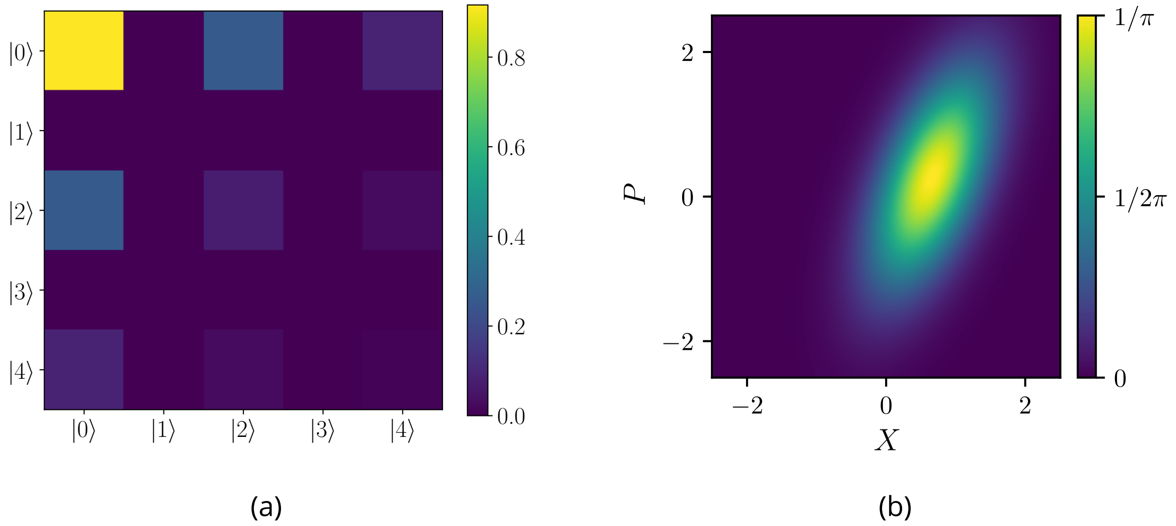


Figure 1.7: Two different representations of a single mode squeezed Gaussian state, of squeezing parameter $r = 0.3(1+i)$ (a) Density matrix representation. Density matrices are complex valued, but we display its absolute values $|\langle k| \hat{\rho} |l\rangle|$ for simplicity. The Hilbert space is cut off at Fock state $|4\rangle$, due to higher states having negligible probability amplitude. (b) Wigner function in the phase space. We observe its distribution is Gaussian, meaning in this case that it is completely characterized by its displacement $\boldsymbol{\alpha}^R \in \mathbb{C}^2$ and covariance matrix $\boldsymbol{\sigma}^R \in \mathbb{C}^{2\times 2}$.

two-mode squeezing. So the goal of the second part of my PhD will be to establish whether there is an advantage in terms of the number of measured observables and expressivity when learning with bosonic modes compared to QRC, even with a quadratic number of trained parameters $\boldsymbol{\theta}$.

To this end we use an analog model, meaning that we do not use multiple gate-based operations. Instead of applying sequential Lindbladian evolutions $\mathcal{L}_k$ each with a different set of parameters $\boldsymbol{\theta}_k$, we apply a single one and harness complex behaviors in the dynamics. The two different approaches are illustrated in Fig. 1.8. Our choice of analog computation stems from its relative simplicity: to highlight advantages of learning in bosonic modes, we do not require the use of multiple parametric gate operations.

We explain the benefits of learning the physical parameters in Chapter 5. We will observe that learning indeed answers limitations of QRC, in that less observables need to be measured for task solving, and the enhanced expressivity allows the bosonic quantum neural network to tackle much harder tasks than previously possible such as DIGITS classification [82]. An additional study of the encoding parameter choice is presented, where we show that encoding into two-mode squeezing rates improves performance on highly non-linear tasks.

These results demonstrate that learning with bosonic modes is a promising perspective in quantum neuromorphic computing. Thanks to their large Hilbert space and

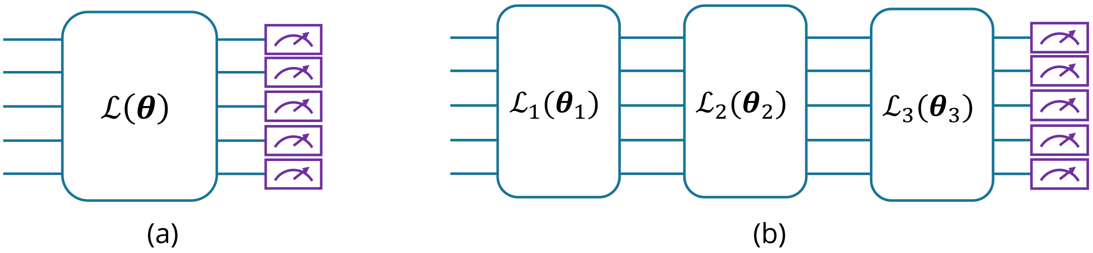

$\mathcal{L}(\boldsymbol{\theta})$
(a)
$\mathcal{L}_1(\boldsymbol{\theta}_1)$
$\mathcal{L}_2(\boldsymbol{\theta}_2)$
$\mathcal{L}_3(\boldsymbol{\theta}_3)$
(b)

Figure 1.8: Two different circuit approaches to parametrically control bosonic modes. Each blue line corresponds to a singular mode. (a) Analog computing, in which all parameters $\boldsymbol{\theta}$ of the Lindbladian evolution described by the superoperator $\mathcal{L}$ are fixed. (b) Gate-based computing with three gates in this example, where sequential Lindbladian evolutions described by superoperators $\mathcal{L}_k$ each have their own set of parameters $\boldsymbol{\theta}_k$. The total set of parameters is then $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k\in[1,3]}$.

parametric couplings, their complex dynamics can be leveraged to fit highly non-trivial data, and act as a memory. In future proposals, using qubits in conjunction with bosonic modes could be useful to create complex non-Gaussian states through measurement protocols. Moving from analog learning to gate-based algorithms will also widen the diversity of tasks which can be solved.

In this work we have trained software simulations of coupled bosonic modes using back-propagation, but in hardware implementations this algorithm is not accessible. Gradient computation with parameter shift rules is not an option either, as none have been established for bosonic modes with Fock state measurements. We could consider offline training, in which all of the training is performed in a software model of the physical system. Parameters learned this way can be injected into the physical system for inference. Two main problems arise when applying this approach: (i) scaling up is difficult because simulating large quantum systems is challenging, and (ii) unavoidable deviations from simulation to experiment decrease its effectiveness. The latter can be mitigated using the hybrid training algorithm developed by Ref. [83], where forward passes happen in the physical system (even during training), but its simulation is used to compute gradients. This method is more precise than offline training.

Nevertheless, computing gradients with back-propagation is not scalable, as it requires to classically simulate the system, if the states are non-Gaussian. When training an experimental bosonic QNN, it will therefore be crucial to explore physics-compatible training methods, on which research is very active [84, 85, 86]. Exploring non-Gaussian states for quantum machine learning will allow for more complex classically intractable dynamics, which may translate to a quantum advantage over classical learning models [87]. Adding higher-order nonlinearities will also provide additional learning parameters.

Finally, further research on the susceptibility of bosonic quantum neural networks to barren plateau is yet to be conducted. Because of the structured way in which the Hilbert space is explored during their dynamics, it is possible that gradient information is not lost as the system size is scaled up.

Chapter 2

Quantum machine learning

The field of quantum computing initially grew in parallel with machine learning, until Bang *et al* proposed a method where classical computers trained the unitary operations on a quantum device to learn quantum algorithms [88]. Such hybrid approaches where classical optimization tools are used on parameterized quantum circuits (PQC) are called variational quantum algorithms (VQA).

A specific case of VQA are quantum neural networks (QNN), which are applied to machine learning tasks. To provide an analogous description to artificial neural networks, in QNN the quantum system's basis states can be considered as neurons, with synaptic weights corresponding to hopping probabilities between them. Consequently, increasing the number of quantum devices results in an exponential increase in the number of neurons. Thanks to this property, it is hoped that learning on quantum systems may produce learning models with higher expressivity than classical ones [89].

Another promising characteristic of QNN's is their generalization capacity (i.e the ability to perform well on unseen data). In multiple empirical studies, it has been observed that QNN's generalize better than classical ANN's. Furthermore, theoretical work has proven that there exists a minimal difference (called "generalization bound") between the performance of a QNN's on seen, and unseen data [25, 26]. This could make them easier to train than their classical counterparts, in big data applications [90].

However, an important obstacle to variational quantum algorithms is the noise and reduced scale of quantum devices defining this current noisy-intermediate scale quantum (NISQ) era. Noise has already proven to hinder the use of NISQ devices in most quantum algorithms. For example in non-learning quantum algorithms such as Shor's, any kind of error falsifies the output result, incurring the use of exponentially time-consuming averaging methods [91, 92]. The error-correcting schemes preventing these issues require multiple physical qubits to simulate a single logical qubit [93, 94], further reducing the feature space available for computation.

In comparison, noise in neuromorphic computing does not seem to make learning

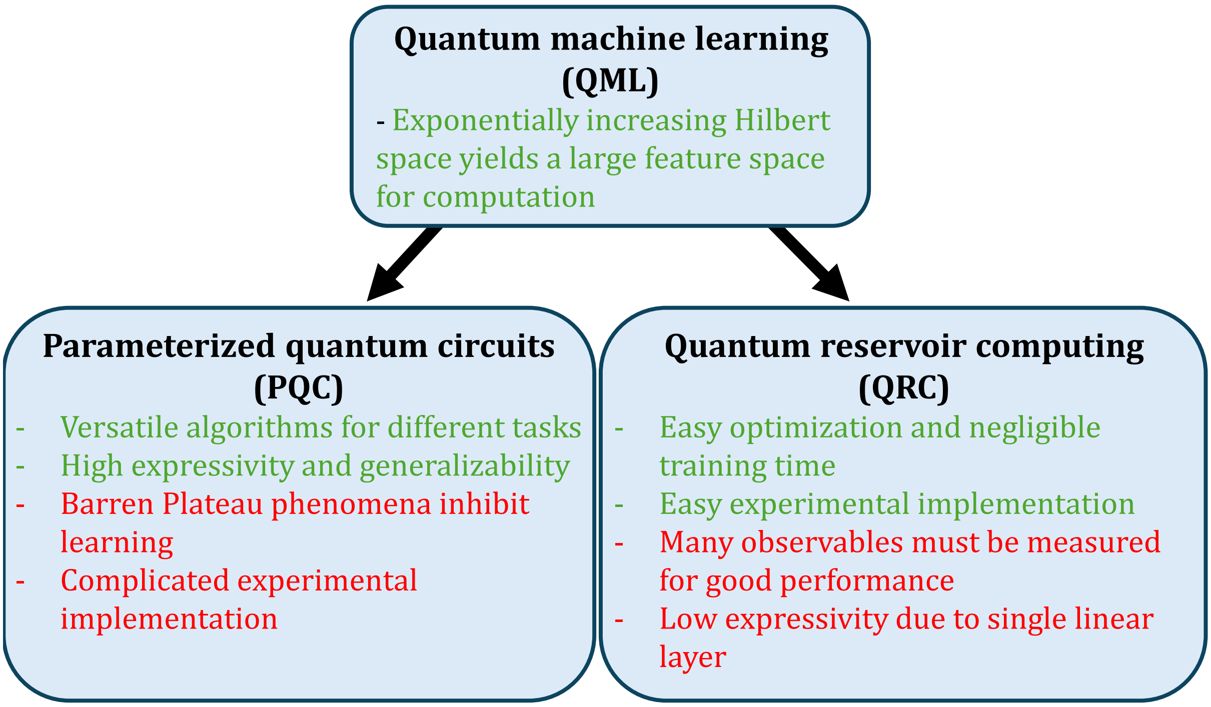

Quantum machine learning (QML)
- Exponentially increasing Hilbert space yields a large feature space for computation
Parameterized quantum circuits (PQC)
- Versatile algorithms for different tasks
- High expressivity and generalizability
- Barren Plateau phenomena inhibit learning
- Complicated experimental implementation
Quantum reservoir computing (QRC)
- Easy optimization and negligible training time
- Easy experimental implementation
- Many observables must be measured for good performance
- Low expressivity due to single linear layer

Figure 2.1: Diagram comparing advantages (green text) and disadvantages (red text) in two different approaches of quantum machine learning. Parameterized quantum circuits implement learning by training gate operation parameters, while quantum reservoirs use an untrained quantum system to project input data into a higher dimensional space, which is then classified with linear regression executed on a classical computer.

models useless. Studies on artificial neural networks show that moderate (around $10\,\%$) amount of noise in input signals or synaptic weights does not have a negative effect on network predictions [95, 96], and is even used to regularize deep neural networks [97]. In the specific case of parameterized quantum circuits, although noise can exacerbate challenges in learning parameters [39], it has been shown that small noise levels do not degrade learning ability [98]. This tolerance to noise has positioned PQC as a promising candidate for the practical use of NISQ hardware.

Like their classical counterparts, quantum learning algorithms have diverse use cases. By default, most proposals implement learning in arrays of qubits, but there have also been both proposals and experimental implementations of VQA in circuits composed of bosonic modes [99, 100, 101]. Compared to qubits, they offer the possibility of using continuous variables.

The first proposed instances of VQA include the variational quantum eigensolver (VQE) [102], the quantum support vector machine (QSVM) [103], and quantum approximate optimization algorithm (QAOA) [104]. Their distinct uses illustrate the versatility with which PQC can be employed. For example, the aim of the VQE is to search ground states of Hamiltonians, and has been applied in finding those of molecules [105] or condensed matter systems [106]. In contrast, the QSVM classifies data by finding an optimal hyperplane that maximizes the distance between each class in the feature space [107]. The QAOA and its variations once again serve a different use, which is combinatorial optimization [108].

In the following sections, we will explain how artificial neural networks and PQC are trained, to provide the necessary tools to understand the context behind quantum reservoir computing, and formulate their functioning. In Section 2.1 I explain how software learning models are trained with gradient descent, since this classical optimization scheme will be re-used in the hybrid optimization of PQC. Finally, Section 2.2 presents a generic PQC framework to illustrate how they are trained, and the challenges arising from their training.

2.1 Machine learning and neural networks

The field of machine learning takes inspiration from the brain's architecture to set its two main building blocks: neurons and synapses. Popular nonlinear activation functions include rectified linear unit (ReLU) defined as

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0, \end{cases} \tag{2.1}$$

and hyperbolic tangent (tanh) defined as

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \tag{2.2}$$

In this section, we establish the basics of learning with neural networks to lay the groundwork to explaining more unconventional quantum machine learning next in this chapter. To classify time-independent data, an instance of ANN architecture is the Multi-Layer Perceptron (MLP), illustrated Fig. 2.2. It will serve as our example to explain how neural networks are trained.

2.1.1 Example of a learning model: the Multi-Layer Perceptron

We describe a supervised learning scenario, where the training dataset $\{\mathbf{X}^{\text{train}}, \mathbf{Y}^{\text{train}}\}$ associates to each input feature vector $\mathbf{x}^{\text{train}} \in \mathbf{X}^{\text{train}}$ a target label $\mathbf{y}^{\text{train}} \in \mathbf{Y}^{\text{train}}$. After training when given the input $\mathbf{x}^{\text{test}}$, the model aims to issue a prediction $\bar{\mathbf{y}}^{\text{test}}$ closest possible to the target label $\mathbf{y}^{\text{test}}$. This step is called an inference.

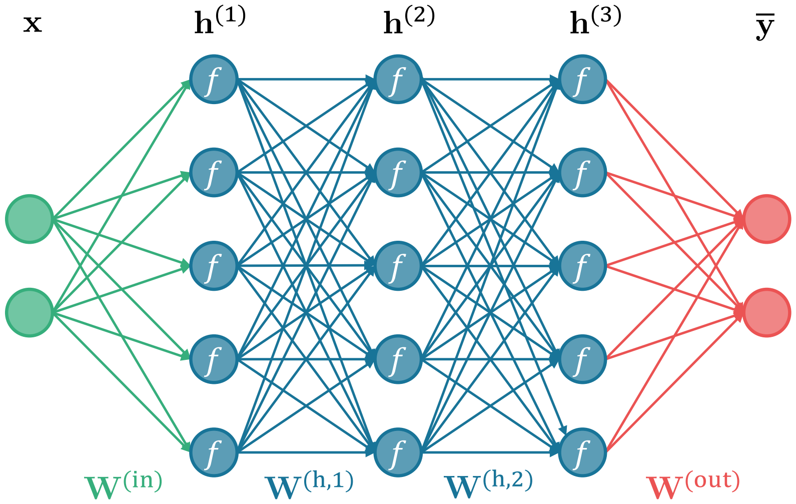

$\mathbf{x}$
$\mathbf{h}^{(1)}$
$\mathbf{h}^{(2)}$
$\mathbf{h}^{(3)}$
$\bar{\mathbf{y}}$
f
$\mathbf{W}^{(\text{in})}$
$\mathbf{W}^{(\text{h},1)}$
$\mathbf{W}^{(\text{h},2)}$
$\mathbf{W}^{(\text{out})}$

Figure 2.2: Sketch of a Multi-Layer Perceptron (MLP) with three hidden layers. The two-dimensional inputs $\mathbf{x}$ (green circles) are fed forward into the hidden layers $\{\mathbf{h}^{(n)}\}_{n\in[1,3]}$ (blue circles) with the linear weight matrices $\{\mathbf{W}^{(\text{in})}, \mathbf{W}^{(\text{h},1)}, \mathbf{W}^{(\text{h},2)}, \mathbf{W}^{(\text{out})}\}$ (arrows), up to the two-dimensional prediction $\bar{\mathbf{y}}$ (red circles). Each hidden layer contains 5 neurons, which apply a nonlinear activation function f.

MLP's are feed-forward neural networks, meaning that to compute a prediction $\bar{\mathbf{y}}$ from an input $\mathbf{x}$, data flows sequentially through the different neuron layers. Let f be the neuronal activation function, N_l the number of hidden neuron layers, and N_h the number of neurons per hidden layer. The input features $\mathbf{x}_k$ are encoded into the first hidden layer $\mathbf{h}^{(1)}$ according to

$$\mathbf{h}_j^{(1)} = f\left(\mathbf{b}_j^{(\text{in})} + \sum_k \mathbf{W}_{jk}^{(\text{in})}\mathbf{x}_k\right), \tag{2.3}$$

where $\mathbf{W}^{(\text{in})}$ is the input weight matrix, and $\mathbf{b}^{(\text{in})}$ are the input biases. Then the hidden layers feed the signal forward according to

$$\mathbf{h}_j^{(n+1)} = f\left(\mathbf{b}_j^{(n)} + \sum_k \mathbf{W}_{jk}^{(\text{h},n)}\mathbf{h}_k^{(n)}\right), \tag{2.4}$$

where $\mathbf{W}^{(\text{h},n)}, \mathbf{b}^{(n)}$ are the hidden weight matrix and biases of the layer n, respectively. The prediction $\bar{\mathbf{y}}$ is obtained with the final output matrix $\mathbf{W}^{(\text{out})}$ and biases $\mathbf{b}^{(\text{out})}$:

$$\bar{\mathbf{y}}_j = \mathbf{b}_j^{(\text{out})} + \sum_k \mathbf{W}_{jk}^{(\text{out})}\mathbf{h}_k^{(n+1)}. \tag{2.5}$$

The set $\boldsymbol{\theta}$ of the MLP's parameters are the weight matrices $\mathbf{W}^{(\text{in})}$, $\{\mathbf{W}^{(\text{h},n)}\}_{n\in[1,N_l-1]}$, $\mathbf{W}^{(\text{out})}$ and the biases $\mathbf{b}^{(\text{in})}$, $\{\mathbf{b}^{(n)}\}_{n\in[1,N_l-1]}$, $\mathbf{b}^{(\text{out})}$. To quantify how far the prediction $\bar{\mathbf{y}}$ is from the target $\mathbf{y}$, we compute a cost function $C(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$: the lower its value, the better the prediction. Evaluations of this cost function for each input $\mathbf{x}$, target $\mathbf{y}$, and parameter set $\boldsymbol{\theta}$, are called 'losses'. Therefore, we want to find the parameters $\boldsymbol{\theta}$ minimizing the loss.

2.1.2 Learning with gradient descent

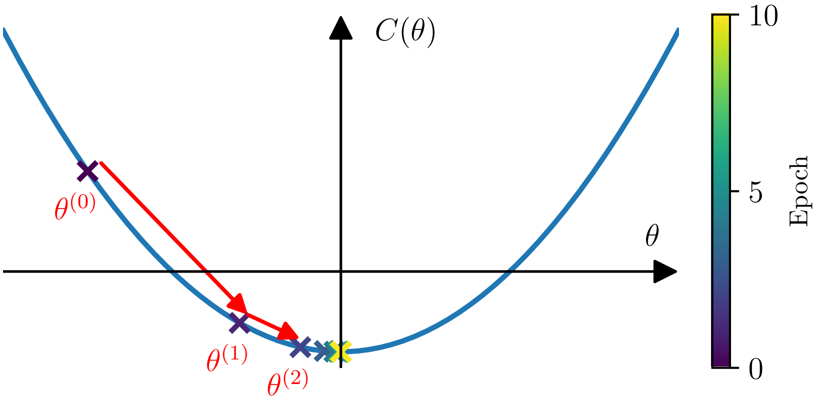


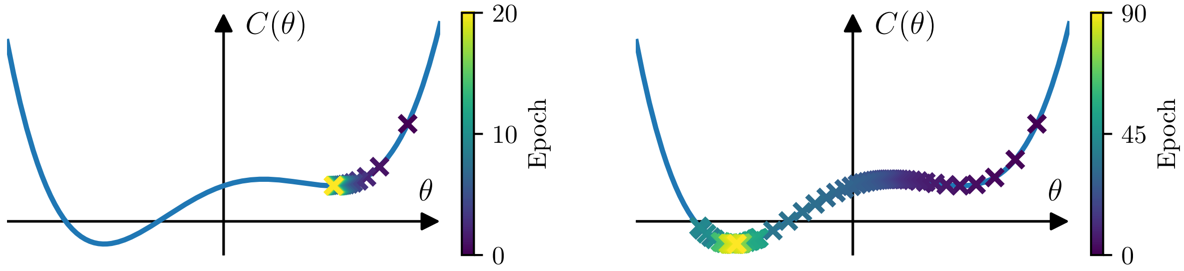
Figure 2.3: Illustration of the stochastic gradient descent algorithm for a convex one dimensional cost function $C(\theta)$. The cost function is optimized with respect to its parameter θ. Starting from $\theta^{(0)}$ (dark blue cross), the parameter is iteratively updated with the rule $\theta^{(n+1)} = \theta^{(n)} - \gamma \nabla_\theta C(\theta^{(n)})$, where γ is a learning rate. The algorithm converges (yellow cross) to the minimum of this function, as the closer it gets, the smaller the gradient $\nabla_\theta C$ becomes.

The most popular optimization algorithm is gradient descent, thanks to its proven efficiency in training large deep learning models such as GPT-4 or ResNet. An optimization algorithm or method used to minimize a cost function is called an 'optimizer'. In its simplest formulation, called the stochastic gradient descent (SGD), starting from an initial set of parameters $\boldsymbol{\theta}^{(0)}$, the parameters $\boldsymbol{\theta}^{(n)}$ at step n are iteratively updated according to

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \gamma \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(n)}, \mathbf{x}, \mathbf{y}), \tag{2.6}$$

where is γ a learning rate defined between 0 and 1, and the input and target label $\{\mathbf{x}, \mathbf{y}\}$ are randomly chosen from the training dataset. A process of evaluating a cost function gradient $\nabla_\theta C$ and updating multiple parameters is called a training epoch. Gradient descent is illustrated in the 1-dimensional case in Fig. 2.3. If the optimizer remains around a minimum of the cost function and iterations do not update parameters anymore, it is said to have converged.

However, the convex case shown Fig. 2.3 does not represent most cost function landscapes: the harder a problem, the more complex its cost landscape will be. It may contain many local minima, and flat areas. In these cases, stochastic gradient descent as formulated in Eq. (2.6) is usually not the most efficient choice because it is prone to getting trapped in local minima, as illustrated in Fig. 2.4a. Since gradients vanish around landscape minima, parameter value updates will eventually stop, and the optimizer will converge to the first minimum it encounters.



(a)

(b)

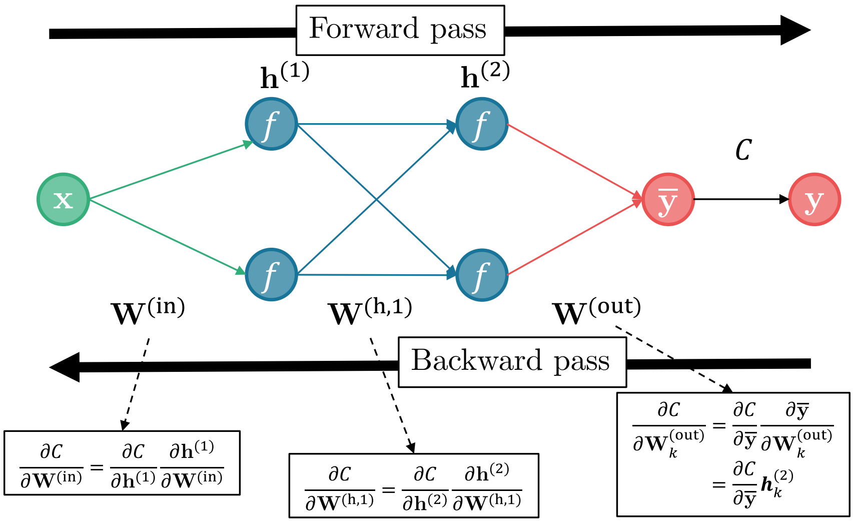
Figure 2.4: Optimization of the cost function $C(\theta) = \theta(\theta + 0.82)(\theta - 0.5)^2 + 0.1$, which possesses 2 local minima. The optimizer iteratively updates its estimation starting at $\theta^{(0)} = 0.85$ (dark blue cross) until convergence (yellow cross). (a) The SGD optimizer is trapped in a local minimum. (b) The Adam optimizer uses momentum to go beyond the local minimum, and find the global minimum.

Improving on the gradient descent method, modern optimizers take into account first and second order momenta of the gradients to get out of local minima [109], and adaptive learning rates help training parameters of different magnitudes [110]. The Adam optimizer combines these two concepts to provide a more balanced and efficient optimization process [111]. As illustrated in Fig. 2.4b, it converges to the global minimum in a cost landscape where SGD cannot, given the same starting point. This is why, in all the work I have performed during my PhD, reported in this thesis, I have used the Adam optimizer.

To check whether the trained model generalizes to unseen data, we compare its performance on the training dataset $\{\mathbf{X}^{\text{train}}, \mathbf{Y}^{\text{train}}\}$ it has been optimized on, with its performance on a test dataset $\{\mathbf{X}^{\text{test}}, \mathbf{Y}^{\text{test}}\}$. If the model performs better on the training dataset, it has trouble processing new data: this is called an overfitting issue, which we want to avoid. Overfitting can be caused by the training dataset not accurately representing the data, or the model fitting too closely the training dataset due to being too complex (e.g having too many learned parameters) for the task.

Furthermore, training can be accelerated by dividing the training dataset into several batches, which can be processed in parallel. Their losses are averaged, and the average is used to compute the cost function gradient.

2.1.3 Computing gradients with the back-propagation algorithm



Forward pass
$\mathbf{h}^{(1)}$
$\mathbf{h}^{(2)}$
f
f
f
f
x
C
$\bar{\mathbf{y}}$
y
$\mathbf{W}^{(\text{in})}$
$\mathbf{W}^{(\text{h},1)}$
$\mathbf{W}^{(\text{out})}$
Backward pass
$\frac{\partial C}{\partial \mathbf{W}^{(\text{in})}} = \frac{\partial C}{\partial \mathbf{h}^{(1)}} \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{W}^{(\text{in})}}$
$\frac{\partial C}{\partial \mathbf{W}^{(\text{h},1)}} = \frac{\partial C}{\partial \mathbf{h}^{(2)}} \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{W}^{(\text{h},1)}}$
$\frac{\partial C}{\partial \mathbf{W}_k^{(\text{out})}} = \frac{\partial C}{\partial \bar{\mathbf{y}}} \frac{\partial \bar{\mathbf{y}}}{\partial \mathbf{W}_k^{(\text{out})}} = \frac{\partial C}{\partial \bar{\mathbf{y}}} \boldsymbol{h}_k^{(2)}$

Figure 2.5: Backward pass in a 2-layer MLP with 2 hidden neurons each, and no biases. After computing the cost function, its derivative is computed with respect to each trainable parameter via the chain rule. Gradient-based optimization algorithms then update the weights.

Performing gradient-based optimization requires knowledge of the cost function gradient with respect to the model's parameters. In software, this is enabled by the back-propagation algorithm [10]. Back-propagation is an efficient version of the chain rule to calculate the derivatives, thanks to its computation of the gradient layer by layer, iterating backward from the final layer to avoid redundant calculations of intermediate terms in the chain rule. It is automatically implemented in multiple Python libraries such as `PyTorch` and `JAX`. This functionality is called 'automatic differentiation'.

As long as the neuron activation and cost functions are differentiable, back-propagation computes gradients with respect to any parameter θ with a single inference. This is possible thanks to the algorithm storing the output of each neuron during inference. We illustrate this process in Fig. 2.5 for a 2-layer MLP with 2 hidden neurons each, and no biases.

This small-scale example is useful to illustrate how back-propagation works. Let f' be the derivative of the neuron activation function. The chain rule allows to compute the gradients:

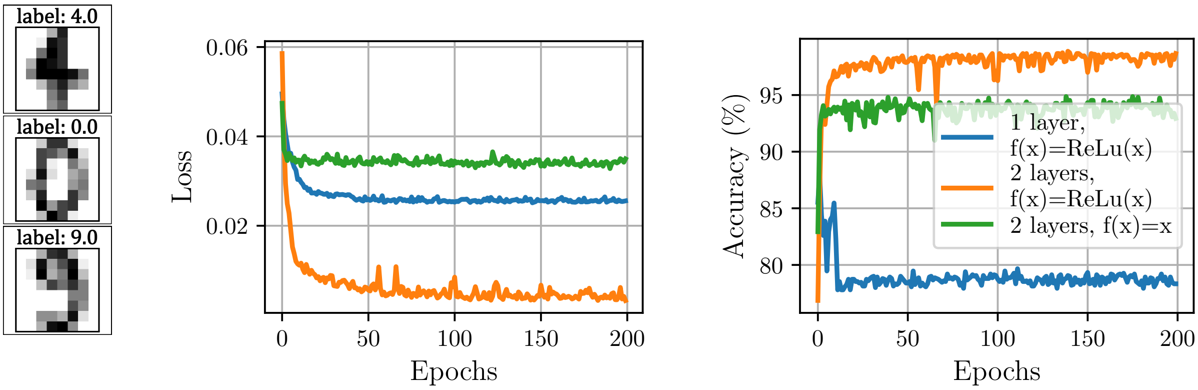
$$\begin{aligned} \frac{\partial C}{\partial \mathbf{W}^{(\text{out})}} &= \frac{\partial C}{\partial \bar{\mathbf{y}}} \frac{\partial \bar{\mathbf{y}}}{\partial \mathbf{W}^{(\text{out})}} \\ &= \frac{\partial C}{\partial \bar{\mathbf{y}}} \times \mathbf{h}^{(2)} \end{aligned} \tag{2.7}$$

$$\begin{aligned}
\frac{\partial C}{\partial \mathbf{W}^{(\mathsf{h},1)}} &= \frac{\partial C}{\partial \mathbf{h}^{(2)}} \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{W}^{(\mathsf{h},1)}} \\
&= \frac{\partial C}{\partial \bar{\mathbf{y}}} \frac{\partial \bar{\mathbf{y}}}{\partial \mathbf{h}^{(2)}} \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{W}^{(\mathsf{h},1)}} \\
&= \frac{\partial C}{\partial \bar{\mathbf{y}}} \times \mathbf{W}^{(\mathsf{out})} \times \mathbf{h}^{(1)} \times f'\left(\mathbf{W}^{(\mathsf{h},1)} \times \mathbf{h}^{(1)}\right)
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
\frac{\partial C}{\partial \mathbf{W}^{(\mathsf{in})}} &= \frac{\partial C}{\partial \mathbf{h}^{(1)}} \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{W}^{(\mathsf{in})}} \\
&= \frac{\partial C}{\partial \mathbf{h}^{(2)}} \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{h}^{(1)}} \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{W}^{(\mathsf{in})}} \\
&= \frac{\partial C}{\partial \bar{\mathbf{y}}} \times \mathbf{W}^{(\mathsf{out})} \times \mathbf{W}^{(\mathsf{h},1)} \times f'\left(\mathbf{W}^{(\mathsf{h},1)} \times \mathbf{h}^{(1)}\right) \times \mathbf{x} \times f'\left(\mathbf{W}^{(\mathsf{in})} \times \mathbf{x}\right).
\end{aligned} \tag{2.9}$$

To confirm our earlier statement, we observe that computing these expressions only requires prior knowledge of derivatives $\frac{\partial C}{\partial \bar{\mathbf{y}}}, f'$, and storing the features $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}$ during inference.

2.1.4 Example of a learning task: DIGITS classification



(a) (b) (c)

Figure 2.6: (a) Three examples from three different classes of the DIGITS dataset, with corresponding labels. The inputs are 8×8 size images, with targets corresponding to the handwritten digit. (b) Loss and (c) accuracy as a function of the training epoch on the DIGITS dataset of a multi layer perceptron (MLP) with ReLU activation functions, 12 neurons per hidden layer, and 1 or 2 layers. In the green curve, the activation function is the identity.

To give a concrete example of supervised learning, we introduce the DIGITS dataset.

Three samples from this dataset are shown in Fig. 2.6a. The inputs are 8×8 pixels images, with 10 different class labels, corresponding to the handwritten digit. After flattening the images, input feature vectors $\mathbf{x}$ are 64-dimensional, and the target feature vectors $\mathbf{y}$ are 10-dimensional. To check for overfitting, the dataset is divided into training and test sections, and we check if performance is the same on both sets after training. They are equally sized, containing 1500 input-label samples.

In Figs. 2.6b and 2.6c, we show the performance of a MLP with ReLU activation functions and 12 hidden neurons per hidden layers on the 'test' dataset, after having optimizing it on the 'training' dataset with the mean-square error (MSE) cost function defined as

$$\text{MSE}(\bar{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2N_{\text{samples}}} \sum_{k=1}^{N_{\text{samples}}} (\bar{\mathbf{Y}}_k - \mathbf{Y}_k)^2. \tag{2.10}$$

$\mathbf{Y}, \bar{\mathbf{Y}}$ are respectively the target dataset and prediction by the model, both containing N_{samples} samples. I could have considered other cost functions such as cross-entropy or mean-absolute error, but I choose the mean-square error because of its simplicity, and its natural implementation of a distance function in vector space.

To study the influence of the neural network's feature space size, we increase the number of hidden layers from 1 to 2, and observe in Fig. 2.6c that the test accuracy raises from $80\,\%$ to $98\,\%$. This illustrates the advantage in classification capacity when increasing the network feature space size.

Moreover, by changing the neuron activation function to the identity, classification performance is degraded, highlighting the importance of the nonlinearity in neural networks.

2.2 Overview of parameterized quantum circuits

2.2.1 Basic principles of quantum circuits

Now that classical optimization tools and concepts have been introduced, we can begin explaining the particularities of optimizing quantum learning models.

As an example, we describe a parameterized quantum circuit composed of qubits, where qubit rotation angles are controllable parameters. This section will introduce the basic tools required to understand their operation. We first present in Section 2.2.1.1 the Bloch sphere representation, in order to help visualize the effect of parametric operations described next in Section 2.2.1.2. Because of quantum mechanics, special attention must be given to measurements in the circuits. The measurement protocol for the estimation of the expectation values with projective measurements is detailed in Section 2.2.1.3.

2.2.1.1 Bloch sphere representation of a qubit

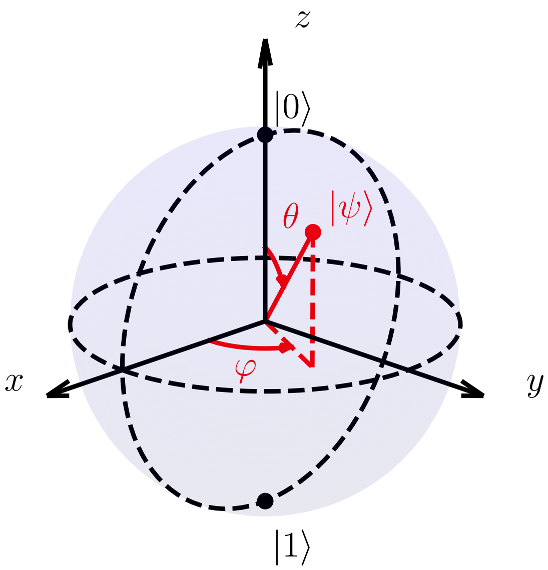


Figure 2.7: Bloch sphere representation of a qubit state $|\psi\rangle$. The polar and azimuthal angles θ, φ define the state on the unit sphere via spherical coordinates.

The qubit Hamiltonian is given by

$$\hat{H}_q = -\frac{\hbar\omega_q}{2}\hat{\sigma}_z, \tag{2.11}$$

where $\hbar\omega_q$ is the energy difference between the ground state $|0\rangle$ and the excited state $|1\rangle$. Here, $\hat{\sigma}_z$ is one of the three Pauli operators, which, together with $\hat{\sigma}_x$, $\hat{\sigma}_y$, and the identity $\hat{I}$, forms a complete basis for single-qubit operations. In the $\{|0\rangle, |1\rangle\}$ basis, the Pauli operators are represented as matrices

$$\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.12}$$

The eigenstates of $\hat{H}_q$ are $|0\rangle$ and $|1\rangle$ such that

$$\begin{aligned} \hat{H}_q|0\rangle &= -\frac{\hbar\omega_q}{2}|0\rangle \\ \hat{H}_q|1\rangle &= \frac{\hbar\omega_q}{2}|1\rangle. \end{aligned} \tag{2.13}$$

So we can express the general pure qubit state in the $\{|0\rangle, |1\rangle\}$ eigenbasis of $\hat{H}_q$ as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.14}$$

with complex coefficients $\alpha, \beta \in \mathbb{C}$ satisfying $|\alpha|^2+|\beta|^2 = 1$. Quantum states are defined up to an arbitrary global phase, so this state can be parameterized as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{2.15}$$

with $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$. This form is known as the Bloch representation. Its geometrical depiction in Fig. 2.7 is useful to visualize operations on qubits. A pure qubit state $|\psi\rangle$ corresponds to a point on the unit sphere with cartesian coordinates $(\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$.

2.2.1.2 Gate operations on qubits

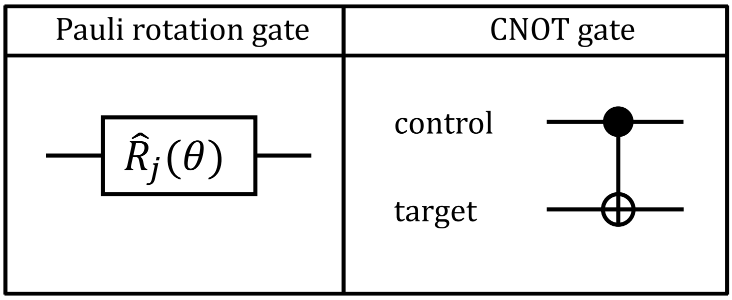


Figure 2.8: Circuit representations of a Pauli rotation gate $\hat{R}_j(\theta)$ (left), and a CNOT gate (right). Horizontal lines represent qubits.

Any unitary single-qubit gate can be expressed as a rotation around an axis of the Bloch sphere. Such rotations are generated by the Pauli operators

$$\hat{R}_j(s) = \exp\left(-\frac{is\hat{\sigma}_j}{2}\right), \quad j \in \{x, y, z\}. \tag{2.16}$$

The operator $\hat{R}_j(s)$ performs a counterclockwise rotation by an angle s around the axis j on the Bloch sphere. Arbitrary single-qubit unitaries can thus be constructed from a sequence of such rotations.

Entanglement between qubits is achieved via multi-qubit gates. The Controlled-NOT (CNOT) gate is a standard two-qubit gate acting on a control and a target qubit. Its expression in terms of Pauli operators is

$$\hat{\text{CNOT}} = \exp\Big(i\frac{\pi}{4}(\hat{I} - \hat{\sigma}_z)_c \otimes (\hat{I} - \hat{\sigma}_x)_t\Big), \tag{2.17}$$

where the c, t subscripts denote an operator acting on the control or target qubit. If the initial qubit states are only allowed to be in $\{|0\rangle, |1\rangle\}$, the CNOT gate applies the classical XOR operation:

$$\begin{aligned}
\widehat{\text{CNOT}}(|0\rangle_c \otimes |0\rangle_t) &= |0\rangle_c \otimes |0\rangle_t \\
\widehat{\text{CNOT}}(|0\rangle_c \otimes |1\rangle_t) &= |0\rangle_c \otimes |1\rangle_t \\
\widehat{\text{CNOT}}(|1\rangle_c \otimes |0\rangle_t) &= |0\rangle_c \otimes |1\rangle_t \\
\widehat{\text{CNOT}}(|1\rangle_c \otimes |1\rangle_t) &= |1\rangle_c \otimes |0\rangle_t .
\end{aligned} \tag{2.18}$$

The circuit representation of the Pauli rotation and CNOT operations are depicted in Fig. 2.8.

Although we describe here Pauli rotations and CNOT in the context of parameterized quantum circuits, they also play an important role in quantum algorithms. This stems from the fact that CNOT combined with qubit rotations form a universal gate set for quantum computation. This means that any operation possible on a quantum computer can be expressed as a finite sequence of such gates. An example of unitary parametric gate decomposed this way is illustrated in Fig. 2.9.

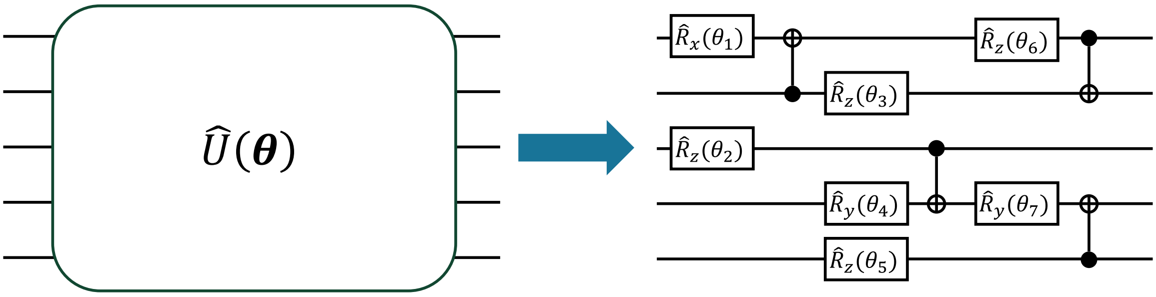


Figure 2.9: Circuit representation of a parametric gate $\hat{U}(\boldsymbol{\theta})$, that can be decomposed into a sequence of Pauli rotations and CNOT gates. Operations are applied sequentially from left to right.

2.2.1.3 Estimating the expectation value of a projection operator with multiple measurement shots

After applying multiple gate operations on the circuit, observables are measured to obtain the output from the PQC. In this section, we describe a setting where these measurements are projective, as is the case in the vast majority of PQC proposals. As of my knowledge, weak measurement have only been considered in quantum reservoir computing frameworks [112, 113, 114] because of their interest for the time-series tasks.

Projective quantum measurements are subject to Born's rule, which states that for any Hermitian operator $\hat{B}$, i.e $\hat{B}^\dagger = \hat{B}$, corresponding to an observable measured in a quantum system in a pure state $|\psi\rangle$, if the spectrum of $\hat{B}$ is discrete and degenerate with real eigenvalues λ_i and associated eigenstates $|\lambda_i\rangle$, then the measurement outcome is

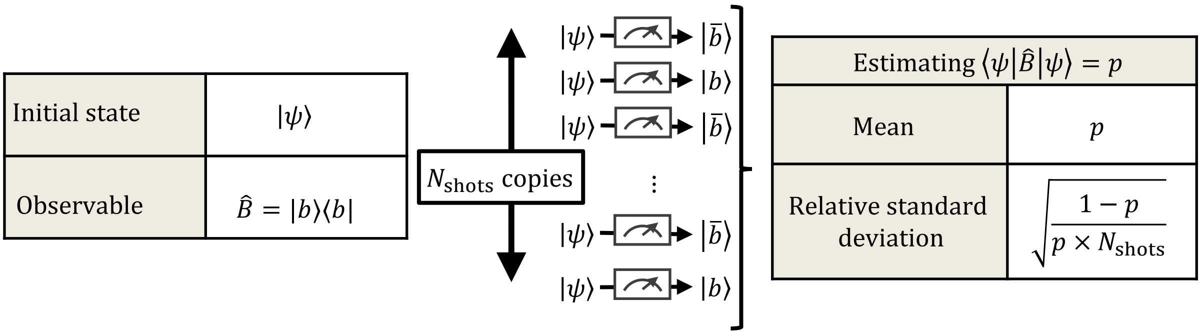

Initial state
$|\psi\rangle$
Observable
$\hat{B} = |b\rangle\langle b|$
N_{shots} copies
$|\psi\rangle$
$|\bar{b}\rangle$
$|b\rangle$
$\vdots$
Estimating $\langle\psi|\hat{B}|\psi\rangle = p$
Mean
p
Relative standard deviation
$\sqrt{\frac{1-p}{p \times N_{\text{shots}}}}$

Figure 2.10: Measurement scheme to estimate the expectation value of the projector observable $\hat{B} = |b\rangle\langle b|$ in a system state $|\psi\rangle$. Here we define $|\bar{b}\rangle$ such that $|b\rangle\langle b| + |\bar{b}\rangle\langle\bar{b}| = \hat{I}$. N_{shots} identical copies of $|\psi\rangle$ are measured, and extracted statistics allow for an estimation of $p \overset{\text{def}}{=} \langle\psi|\hat{B}|\psi\rangle$, with a relative standard deviation $\sqrt{\frac{(1-p)}{pN_{\text{shots}}}}$.

an eigenvalue λ_i. The probability of this outcome is $|\langle\psi|\lambda_i\rangle|^2$, and upon measurement the system collapses to the eigenstate $|\lambda_i\rangle$. This action of measurement on the quantum state is commonly referred to as "measurement back-action". The expectation value of $\hat{B}$ with respect to $|\psi\rangle$ is defined as

$$\langle B\rangle_{|\psi\rangle} \overset{\text{def}}{=} \langle\psi|\hat{B}|\psi\rangle = \sum_i \lambda_i |\langle\psi|\lambda_i\rangle|^2. \tag{2.19}$$

If instead the quantum state were to be mixed, i.e defined by a density matrix $\hat{\rho}$, the expectation value is defined as

$$\langle B\rangle_{\hat{\rho}} \overset{\text{def}}{=} \mathrm{Tr}\Big(\hat{\rho}\hat{B}\Big). \tag{2.20}$$

A simple illustrative example of Born's rule is that of a qubit in a general superposition of states as defined in Eq. (2.14). Measuring the qubit energy assigned in Eq. (2.11) results in observing the state $|0\rangle$ with probability $|\alpha|^2$, or the state $|1\rangle$ with probability $|\beta|^2$.

In experiment, estimating the expectation value of $\hat{B}$ with respect to the pure state $|\psi\rangle$ requires an experimenter to produce N_{shots} independent samples of $|\psi\rangle$, due to the probabilistic nature of Born's rule. To show the error when estimating $\langle\psi|\hat{B}|\psi\rangle$ caused by quantum uncertainty, we consider the case illustrated in Fig. 2.10 where the observable $\hat{B} = |b\rangle\langle b|$ is a projector on the state $|b\rangle$.

Defining $|\bar{b}\rangle$ such that $|b\rangle\langle b| + |\bar{b}\rangle\langle\bar{b}| = \hat{I}$, $\hat{B}$ has two eigenstates $|b\rangle$ and $|\bar{b}\rangle$, with respective associated eigenvalues 1 and 0. We define $p \overset{\text{def}}{=} \langle\psi|\hat{B}|\psi\rangle = |\langle\psi|b\rangle|^2$. Let $\{Z_i\}_{i\in[1,N_{\text{shots}}]}$ be N_{shots} independent and identically distributed Bernoulli random variables with $P(Z_i = 1) = p$. Computing $\langle\psi|\hat{B}|\psi\rangle$ from N_{shots} measurement outcomes is equivalent to computing the expectation value of the random variable

$$Z = \frac{\sum_{i=1}^{N_{\text{shots}}} Z_i}{N_{\text{shots}}}. \tag{2.21}$$

We note $P(Z)$ is a binomial probability distribution with parameters N_{shots} and p, with a $1/N_{\text{shots}}$ prefactor. Its mean and variance are respectively

$$\begin{aligned} &\mathbb{E}(Z) = p \\ &\sigma(Z) \stackrel{\text{def}}{=} \mathbb{E}(Z^2) - (\mathbb{E}(Z))^2 = \frac{p(1-p)}{N_{\text{shots}}}. \end{aligned} \tag{2.22}$$

The relative standard deviation is a more relevant metric than the variance to assert how well the expectation value $\langle\psi|\hat{B}|\psi\rangle$ is estimated, because it shows the extent of variability in relation to the mean measured result. Its calculation yields

$$\frac{\sqrt{\sigma(Z)}}{\mathbb{E}(Z)} = \sqrt{\frac{(1-p)}{pN_{\text{shots}}}}. \tag{2.23}$$

From this expression, we observe that for $p \ll 1$, the relative standard deviation scales as $\propto \frac{1}{\sqrt{pN_{\text{shots}}}}$. This is an important result to remember for the remaining of this manuscript, because it means that to estimate an expectation value which vanishes exponentially with a given fidelity, the number of necessary measurement samples increases exponentially.

2.2.2 General formulation of parameterized quantum circuits

We have now introduced all the necessary elements to explain how parameterized quantum circuits work. We describe a setting where the PQC is a sequence of L parameterized unitary gates

$$\hat{U}(\boldsymbol{\theta}) = \prod_{i=1}^{L} \hat{U}_i(\boldsymbol{\theta}_i), \tag{2.24}$$

applied to an array of qubits, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$ is a set of trainable parameters. The cost function is obtained by estimating the expectation value of an observable $\hat{O}$

$$C(\boldsymbol{\theta}) = \mathrm{Tr}\Big(\hat{\rho}\hat{U}(\boldsymbol{\theta})^\dagger \hat{O}\hat{U}(\boldsymbol{\theta})\Big), \tag{2.25}$$

where $\hat{\rho}$ is the initial state of the PQC in density matrix form. If it is a pure state, it can be expressed as $\hat{\rho} = |\psi\rangle\langle\psi|$. Eq. (2.25) expresses a Hilbert-Schmidt inner product between two operators $\hat{\rho}$ and $\hat{U}(\boldsymbol{\theta})^\dagger\hat{O}\hat{U}(\boldsymbol{\theta})$. The choice of the initial state $\hat{\rho}$, unitary gates $\hat{U}_i(\boldsymbol{\theta}_i)$ and measured observable $\hat{O}$ fully characterizes the PQC, and is called its ’ansatz’.

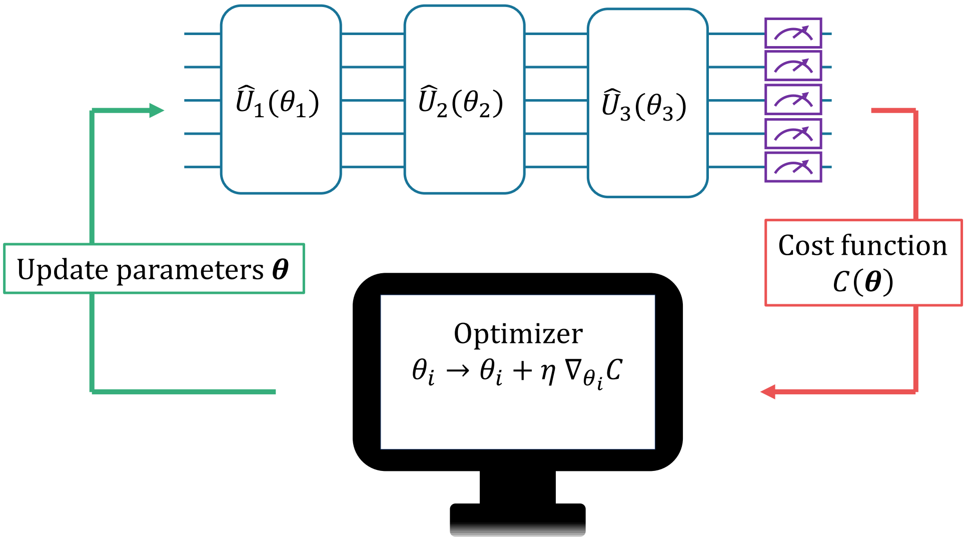

$\hat{U}_1(\theta_1)$
$\hat{U}_2(\theta_2)$
$\hat{U}_3(\theta_3)$
Update parameters $\boldsymbol{\theta}$
Cost function $C(\boldsymbol{\theta})$
Optimizer
$\theta_i \to \theta_i + \eta \, \nabla_{\theta_i} C$

Figure 2.11: Sketch of an example parameterized quantum circuit optimization. During an inference three $\hat{U}_j$ gates are sequentially applied, with respective parameters $\theta_1, \theta_2, \theta_3$. A cost function $C(\boldsymbol{\theta})$ is computed by measuring an observable. A classical optimizer (here gradient descent) updates the parameters, which are re-injected into the gates for the next forward pass.

A difficulty that arises when optimizing $C(\boldsymbol{\theta})$ is that contrary to software implementations portrayed in Section 2.1.3, gradients of the cost function with respect to parameters cannot be analytically computed with back-propagation. One simple solution is to train a software simulation of the physical model, into which these trained weights are imported. The major shortcoming to this workaround is the difference between an experiment and its simulation due to imperfect knowledge of it. A second shortcoming specific to optimizing quantum systems in general is that simulating large quantum systems is infeasible. Furthermore, many PQC applications aim for an advantage in resources used compared to conventional computers, so their classical simulability would nullify this claim.

2.2.3 Parameter shift rules

The first method to obtain gradients in PQC is variational. It relies on a numerical approximation of a gradient. To compute the gradient of the cost function with respect to a single parameter θ, two inferences would be run with the PQC: one with parameter θ, and another with $\theta + d\theta$, where $d\theta$ is a small increment. Then a first degree approximation of the gradient could be estimated:

$$\nabla_\theta C \approx \frac{C(\theta + d\theta) - C(\theta)}{d\theta}. \tag{2.26}$$

This procedure lacks precision, restricting the fine tuning possibilities of PQC. In 2018, the identification of "parameter shift rules" allowing to compute exactly analytical gradients via two PQC forward passes by Ref. [33] for Pauli rotation gates paved the way towards fixing this limitation. Parameter shift rules were generalized to broader gate sets by Ref. [34].

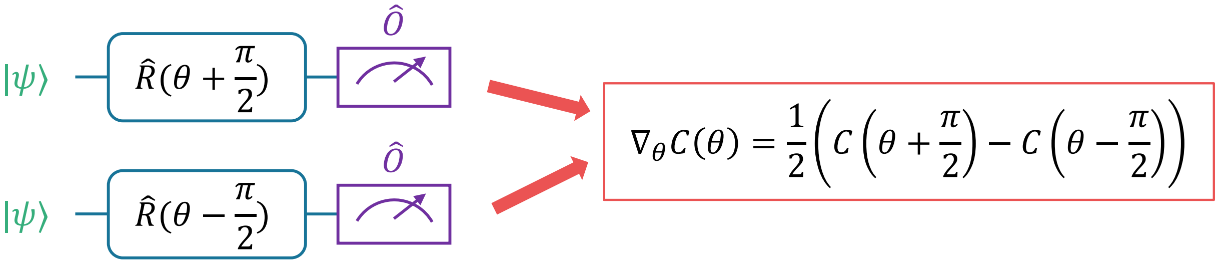

$|\psi\rangle$
$\hat{R}(\theta + \frac{\pi}{2})$
$\hat{O}$
$|\psi\rangle$
$\hat{R}(\theta - \frac{\pi}{2})$
$\hat{O}$
$\nabla_\theta C(\theta) = \frac{1}{2}\left(C\left(\theta + \frac{\pi}{2}\right) - C\left(\theta - \frac{\pi}{2}\right)\right)$

Figure 2.12: Circuit used to illustrate a parameter shift rule. Starting in state $|\psi\rangle$, the qubit undergoes a Pauli rotation $\hat{R}(\theta)$. The cost function is then obtained by projectively measuring the expectation value of $\hat{O}$. The parameter shift rule allows the computation of $\nabla_\theta C$ by evaluating $C(\theta + \frac{\pi}{2})$ and $C(\theta - \frac{\pi}{2})$.

Let us give here a concrete and commonly found example of a parameter shift rule, following the demonstration presented in Ref. [35]. The circuit depicted in Fig. 2.12 is that of a Pauli rotation $\hat{R}(\theta)$ applied to an initial pure qubit state $|\psi\rangle$. Measuring an observable $\hat{O}$ expectation value yields a cost function

$$C(\theta) = \langle\psi|\hat{R}(\theta)^\dagger\hat{O}\hat{R}(\theta)|\psi\rangle\,. \tag{2.27}$$

We calculate its gradient using the chain rule, and write

$$\nabla_\theta C(\theta) = \langle\psi|\nabla_\theta\hat{R}(\theta)^\dagger\hat{O}\hat{R}(\theta) + \hat{R}(\theta)^\dagger\hat{O}\nabla_\theta\hat{R}(\theta)|\psi\rangle\,. \tag{2.28}$$

The gradient of Pauli rotation gates defined in Eq. (2.16) is

$$\nabla_\theta\hat{R}(\theta) = -\frac{i}{2}\hat{\sigma}_j\hat{R}(\theta), \tag{2.29}$$

where $\hat{\sigma}_j$ is the Pauli operator generating $\hat{R}(\theta)$. Substituting this formula into Eq. (2.28) gives

$$\nabla_\theta C(\theta) = \frac{i}{2}\langle\psi|\hat{R}(\theta)^\dagger\left[\hat{\sigma}_j, \hat{O}\right]\hat{R}(\theta)|\psi\rangle\,, \tag{2.30}$$

where $[X, Y] \overset{\text{def}}{=} XY - YX$ denotes the commutator. Using the mathematical identity for commutators including Pauli operators [33]

$$\left[\hat{\sigma}_j, \hat{O}\right] = -i\left(\hat{R}\left(\frac{\pi}{2}\right)^\dagger\hat{O}\hat{R}\left(\frac{\pi}{2}\right) - \hat{R}\left(-\frac{\pi}{2}\right)^\dagger\hat{O}\hat{R}\left(-\frac{\pi}{2}\right)\right), \tag{2.31}$$

we develop Eq. (2.30):

$$\begin{aligned}\nabla_\theta C(\theta) = &\frac{i}{2}\langle\psi|\hat{R}\left(\theta+\frac{\pi}{2}\right)^\dagger \hat{O}\hat{R}\left(\theta+\frac{\pi}{2}\right)|\psi\rangle \\ &-\frac{i}{2}\langle\psi|\hat{R}\left(\theta-\frac{\pi}{2}\right)^\dagger \hat{O}\hat{R}\left(\theta-\frac{\pi}{2}\right)|\psi\rangle\,.\end{aligned} \tag{2.32}$$

Using the cost function's definition, we can finally express the gradient with a parameter shift rule

$$\nabla_\theta C(\theta) = \frac{1}{2}\left(C\left(\theta+\frac{\pi}{2}\right)-C\left(\theta-\frac{\pi}{2}\right)\right)\,. \tag{2.33}$$

The gradients can thus be exactly expressed as a difference of two cost function evaluations for two values of parameters shifted by $\pi/2$. We note that this is a specific case, and this shift is different in other parameter shift rules, depending on the parametric operation to differentiate.

2.2.4 PQC expressivity

While parameter shift rules have enabled faster convergence to local minima of PQC [36], many challenges remain in their optimization. The main open question remains the ansatz design. While classical optimization algorithms determine gate parameters, a conscious choice is made to determine the initial state, sequence of gates and measured observables. Similar to choosing appropriate classical ANN architectures for different tasks, we want the PQC expressivity to be high enough in order to fit the dataset and generalize to unseen data, but not so high that training becomes difficult, i.e resource intensive.

In the specific context of PQC, ansatz expressivity relates to the distribution of explored states $\hat{U}(\boldsymbol{\theta})^\dagger\hat{\rho}\hat{U}(\boldsymbol{\theta})$ over random $\boldsymbol{\theta}$ parameters. The closer it gets to the uniform distribution over the Hilbert space, called the Haar distribution, the more expressible the ansatz. Reaching a Haar distribution means the PQC can reach any state in its Hilbert space.

Many ansatze maximizing this expressivity have been proposed [115, 116]. However, high expressivity is also a source of problems in PQC optimization, due to the Barren Plateau (BP) phenomenon.

2.2.5 Barren plateaus

Barren Plateaus occur when gradient magnitudes vanish exponentially with the number of qubits, posing a problem when scaling up a model. They are caused by the high expressivity of PQC, whose Hilbert feature space scales exponentially with model size. Rescaling the measured observable does not bypass the problem: since gradients are computed by subtracting two expectation values, if these expectation values become exponentially close then an exponentially large amount of measurement shots are needed to estimate their difference correctly (see Section 2.2.1.3).

More specifically, barren plateaus are characterized by exponentially vanishing gradients in the cost landscape. For instance we say a cost function exhibits a probabilistic BP [37] if for some (but not necessarily all) $\theta_j \in \boldsymbol{\theta}$,

$$\mathsf{Var}(\nabla_{\theta_j} C) \in \mathcal{O}\left(\frac{1}{b^{n_\mathsf{q}}}\right), \tag{2.34}$$

where n_q is the number of qubits, $b > 1$, and $\mathsf{Var}(f)$ is the variance of a function f with respect to its parameters: $\mathsf{Var}(f(\boldsymbol{\theta})) \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{\theta}}(f^2) - (\mathbb{E}_{\boldsymbol{\theta}}(f))^2$. This barren plateau is called probabilistic because it is an average property of the cost landscape, but some parts of the landscape may be less flat.

We introduce toy examples taken from Ref. [38] to illustrate ansatze featuring barren plateaus. The task is state preparation of n_q qubits, with the target state being vacuum $|\mathbf{0}\rangle \overset{\text{def}}{=} |0\rangle_1 \otimes ... \otimes |0\rangle_{n_\mathsf{q}}$. The qubits are initialized in the vacuum state, and they all undergo a Pauli rotation $\hat{R}_x(\theta_j)$:, such that the state of the whole circuit is transformed by a unitary

$$\hat{U}(\boldsymbol{\theta}) = \bigotimes_{j=1}^{n_\mathsf{q}} \hat{R}_x(\theta_j). \tag{2.35}$$

A cost function that is minimal when reaching the vacuum state can be evaluated by measuring $\hat{O}_\mathsf{G} = \hat{I} - |\mathbf{0}\rangle\langle\mathbf{0}|$

$$\begin{aligned} C_1(\boldsymbol{\theta}) &= 1 - \langle\mathbf{0}|\hat{U}(\boldsymbol{\theta})^\dagger|\mathbf{0}\rangle\langle\mathbf{0}|\hat{U}(\boldsymbol{\theta})|\mathbf{0}\rangle \\ &= 1 - \prod_{j=1}^{n_\mathsf{q}} \cos^2\left(\frac{\theta_j}{2}\right). \end{aligned} \tag{2.36}$$

By instead measuring the local observable

$$\hat{O}_\mathsf{L} = \hat{I} - \frac{1}{n_\mathsf{q}} \sum_{j=1}^{n_\mathsf{q}} |0\rangle_j \langle 0|_j \otimes \hat{I}_{\bar{j}}, \tag{2.37}$$

where $\hat{I}_{\bar{j}}$ is the identity on all qubits except qubit j, another cost function can be defined for this task

$$\begin{aligned} C_2(\boldsymbol{\theta}) &= \langle\mathbf{0}|\hat{U}(\boldsymbol{\theta})^\dagger\hat{O}_\mathsf{L}\hat{U}(\boldsymbol{\theta})|\mathbf{0}\rangle \\ &= 1 - \frac{1}{n_\mathsf{q}} \sum_{j=1}^{n_\mathsf{q}} \cos^2\left(\frac{\theta_j}{2}\right). \end{aligned} \tag{2.38}$$

We want to calculate the variances of the gradients of the cost functions C_1 and C_2 to reveal the concentration of their landscapes. The respective gradients are

$$\nabla_{\theta_j} C_1 = -\frac{1}{2}\sin(\theta_j)\prod_{\substack{k=1\\k\neq j}}^{n_\mathsf{q}}\cos^2\left(\frac{\theta_k}{2}\right) \tag{2.39}$$

$$\nabla_{\theta_j} C_2 = -\frac{1}{2n_\mathsf{q}}\sin(\theta_j). \tag{2.40}$$

Since $\nabla_{\theta_j} C_1$ and $\nabla_{\theta_j} C_2$ are both anti-symmetric, their means are zero, i.e $\mathbb{E}_{\boldsymbol{\theta}}(\nabla_{\theta_j} C_1) = \mathbb{E}_{\boldsymbol{\theta}}(\nabla_{\theta_j} C_2) = 0$. Using the integrals

$$\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sin^2(\theta) = \frac{1}{2} \tag{2.41}$$

$$\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\cos^4\left(\frac{\theta}{2}\right) = \frac{3}{8}, \tag{2.42}$$

we compute the mean of the squared gradient for each cost function

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\theta}}\left(\left(\nabla_{\theta_j} C_1\right)^2\right) &= \frac{1}{4}\int_{-\pi}^{\pi}\prod_{k=1}^{n_\mathsf{q}}\frac{d\theta_k}{2\pi}\sin^2(\theta_j)\prod_{\substack{k'=1\\k'\neq j}}^{n_\mathsf{q}}\cos^4\left(\frac{\theta_{k'}}{2}\right)\\ &= \frac{1}{4}\left(\int_{-\pi}^{\pi}\sin^2(\theta)\frac{d\theta}{2\pi}\right)\left(\int_{-\pi}^{\pi}\cos^4\left(\frac{\theta}{2}\right)\frac{d\theta}{2\pi}\right)^{n_\mathsf{q}-1}\\ &= \frac{1}{8}\left(\frac{3}{8}\right)^{n_\mathsf{q}-1}\end{aligned} \tag{2.43}$$

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\theta}}\left(\left(\nabla_{\theta_j} C_2\right)^2\right) &= \frac{1}{4}\frac{1}{{n_\mathsf{q}}^2}\int_{-\pi}^{\pi}\prod_{k=1}^{n_\mathsf{q}}\frac{d\theta_k}{2\pi}\sin^2(\theta_j)\\ &= \frac{1}{4}\frac{1}{{n_\mathsf{q}}^2}\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sin^2(\theta)\\ &= \frac{1}{8}\frac{1}{{n_\mathsf{q}}^2}.\end{aligned} \tag{2.44}$$

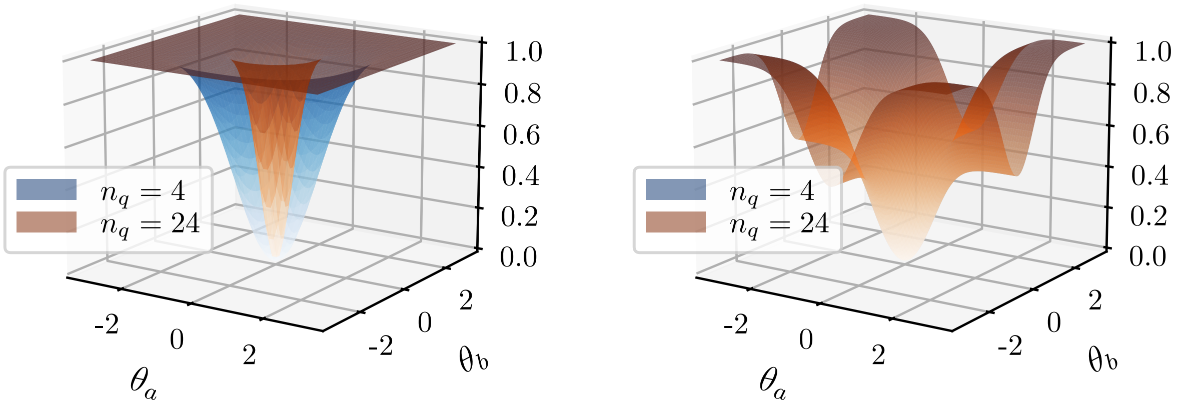
Finally, we deduce the variances of the gradients

$$\mathsf{Var}(\nabla_{\theta_j} C_1) = \frac{1}{8}\left(\frac{3}{8}\right)^{n_\mathsf{q}-1} \tag{2.45}$$

$$\mathsf{Var}(\nabla_{\theta_j} C_2) = \frac{1}{8{n_\mathsf{q}}^2}. \tag{2.46}$$

We observe that C_1 exhibits barren plateaus, while C_2 does not. This is illustrated in Fig. 2.13 when plotting the two-dimensional cross-section of their cost landscapes with

$n_\mathrm{q} = 4$ or 24 qubits. This cross-section is shown for its visual clarity; half of rotation parameters are set equal to $\theta_a \in [-\pi, \pi]$ and the others to $\theta_b \in [-\pi, \pi]$. We observe for C_1 that the two-dimensional cross-section narrows exponentially when increasing n_q, while it remains unchanged for C_2, reflecting their loss concentration behaviors. In this example, the barren plateau comes from the locality of the measured cost function, but barren plateaus can arise in different ansatze through other means such as hardware noise [39] and excessively high entanglement [117].



(a)

(b)

Figure 2.13: Two dimensional cross-sections of the cost functions (a) $C_1(\boldsymbol{\theta}) = 1 - \prod_{j=1}^{n_\mathrm{q}} \cos^2\left(\frac{\theta_j}{2}\right)$ and (b) $C_2(\boldsymbol{\theta}) = 1 - \frac{1}{n_\mathrm{q}} \sum_{j=1}^{n_\mathrm{q}} \cos^2\left(\frac{\theta_j}{2}\right)$ as a function of two parameters θ_a and θ_b. These landscapes are obtained by applying Pauli rotations $\hat{U}(\boldsymbol{\theta}) = \bigotimes_{j=1}^{n_\mathrm{q}} \hat{R}_x(\theta_j)$ on n_q qubits followed by either measuring $\hat{O}_\mathrm{G} = \hat{I} - |\mathbf{0}\rangle \langle \mathbf{0}|$ or $\hat{O}_\mathrm{L} = \hat{I} - \frac{1}{n_\mathrm{q}} \sum_{j=1}^{n_\mathrm{q}} |0\rangle_j \langle 0|_j \otimes \hat{I}_{\bar{j}}$. Half of rotation parameters are set equal to $\theta_a \in [-\pi, \pi]$ and the other half to $\theta_b \in [-\pi, \pi]$. We plot the landscapes for $n_\mathrm{q} = 4$ or $n_\mathrm{q} = 24$ qubits. In (a) the landscape becomes concentrated around the origin, revealing a barren plateau. In the case of (b) the cross-section is unchanged by the qubit number n_q: surfaces overlap over the whole space, meaning increasing n_q does not make training more difficult.

The origin of BP is not the same as vanishing gradient problems in software neural networks. In software ANN, the multiplication of small gradients together in the back-propagation to the first layers make their magnitudes exponentially smaller than gradients in the later layers [118]. In PQC, barren plateaus are caused by the exponentially large Hilbert space, because the inner product between two exponentially large parameterized operators as described in Eq. (2.25) will be on average exponentially small and concentrated [37]. In particular, it has been demonstrated that BP ensue if an ansatz forms a 2-design, i.e it matches the Haar distribution up to the second moment. The Hilbert space then seems like a double-edged sword, as its exponential size is both appealing due to its expressivity, but prohibitive due to its BP phenomena.

As the source of BP is different than that of vanishing gradients in software ANN, mitigation strategies such as batch normalization [119], layer normalization [120], or using ReLU activation functions [121] that are used to regulate the vanishing gradients cannot be transposed to parameterized quantum circuits. Although strategies to circumvent BP have been suggested, such as limiting ansatz expressivity [116], measuring local observables [38] or lowering entanglement of the input state [122], it has been observed that many of them make the PQC classically simulable [41].

A recent but powerful method to understanding BP introduced by Ref. [40] is Lie algebraic theory, where studying the dynamical Lie algebra of the ansatz with respect to the unitary gate $\hat{U}_i(\boldsymbol{\theta}_i)$ generators, the input state $\hat{\rho}$ and the measured observable $\hat{O}$ allows to analytically compute the cost function gradient variance Var$(\nabla_{\boldsymbol{\theta}} C)$. This proves to be a rigorous systematic approach to finding out if an ansatz will lead to BP.

The future of learning with PQC will depend on whether PQC optimization schemes avoiding BP still retain the attractive properties of non-simulability, expressivity and generalization capability which aroused interest from the quantum community in the first place.

Chapter 3

Quantum reservoir computing

In the previous chapter, we saw that learning the parameters of a parameterized quantum circuit is challenging due to barren plateau phenomena, with no unequivocal solution as of today. In this context, the question is if the information processing capabilities of quantum devices can still be harnessed for learning problems.

Quantum reservoir computing (QRC) as a quantum machine learning framework was introduced in an attempt to answer this question. It was inspired by classical reservoir computing, a framework used in machine learning using physical systems (i.e neuromorphic computing), in which the parameters of the physical system, called reservoir in this context, are not trained. Because it does not require computation of gradients in the reservoir, classical reservoir computing is well adapted to establish whether a certain hardware has potential for neuromorphic computing, independently of the physics-compatible learning method to adopt in future perspectives. This had led to many hardware neural network architectures first being proposed with RC, ranging from electronics [45, 46, 47, 48], optics [49, 50] and spintronics [51, 52].

Ideas of reservoir computing emerged independently in software neural networks with echo state networks (ESN) [43] and liquid state machines (LSM) [44], where concerns of vanishing gradients in recurrent neural networks (RNN) [10, 123, 124] motivated scientists to find ways of exploiting dynamical neural network's processing power without training their synaptic weights.

We note that the reservoir computing framework can be applied to time independent tasks, in which case it is called an extreme learning machine (ELM). ELM have been implemented with optical [50] and memristor [125] reservoirs.

The role of this chapter is to position the proposed frameworks of QRC with bosonic modes in the scientific literature leading up to it. In the first section we will introduce recurrent neural networks, and why they have vanishing gradient problems. The second subsection will cover the basics of RC with software and dynamical systems, and how it solves the problems of RNN. Then a state of the art in QRC with qubits arrays will be described. The stochastic nature of quantum measurements requires specific protocols for experimental QRC implementations, which we discuss in Section 3.4. Finally, we consider the different real-life use cases of QRC, to argue how the implementation of bosonic modes in the reservoir computing framework could be advantageous.

3.1 Recurrent neural networks

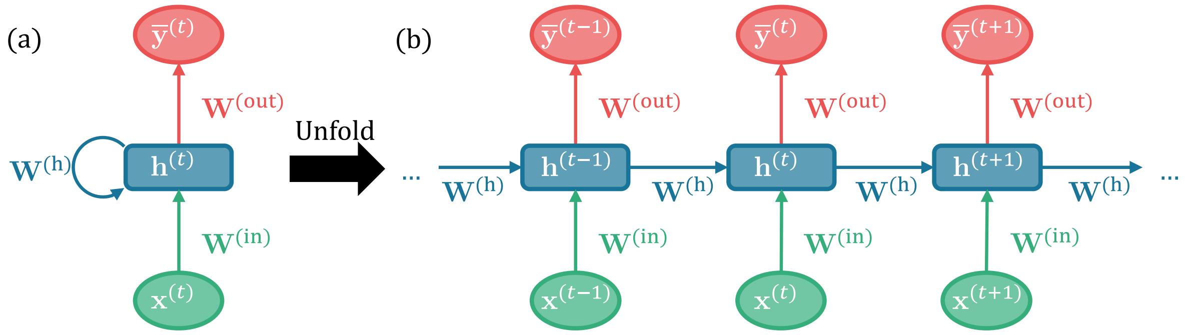


Figure 3.1: (a) Folded and (b) unfolded representation of a fully connected recurrent neural network (RNN). Data flows vertically from bottom towards the top. The data is temporal, so horizontal lines correspond to following data inputs. At each discrete time increment ν, the inputs $\mathbf{x}^{(\nu)}$ are sequentially passed through a linear layer $\mathbf{W}^{(\text{in})}$, and the hidden features $\mathbf{h}^{(\nu)}$ remember past inputs using linear recurrent connections $\mathbf{W}^{(\text{h})}$. The predictions $\bar{\mathbf{y}}^{(\nu)}$ are obtained from the hidden features with the linear layer $\mathbf{W}^{(\text{out})}$.

The precursor to reservoir computers were recurrent neural networks (RNN), a class of artificial neural networks conceived for processing sequential datasets such as speech, text and time series. To this end, they require properties of nonlinearity and memory (remembering previous inputs). The first property is obtained with the nonlinear activation functions of neurons, as is commonly found in software ANN. The latter is achieved with recurrent connections between neurons, whose outputs are fed back into the input during the next time step, as seen in its folded representation in Fig. 3.1a.

Of all the different RNN architectures such as the long short term memory (LSTM) [126] and Hopfield networks [6], we choose to describe a fully connected RNN as it is the most general topology, and the closest to echo state networks, the reservoir architecture we will detail next as an example. For simplicity, we omit bias terms.

Learning is performed in a supervised setting, where the inputs $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_\nu$ and targets $\mathbf{Y} = \left\{\mathbf{y}^{(\nu)}\right\}_\nu$ are time series with discretized time increments ν. The first output is obtained by forward pass through a single hidden neuron layer

$$\begin{cases} \mathbf{h}^{(0)} & = f\left(\mathbf{W}^{(\text{in})}\mathbf{x}^{(0)}\right) \\ \mathbf{y}^{(0)} & = \mathbf{W}^{(\text{out})}\mathbf{h}^{(0)}, \end{cases} \tag{3.1}$$

where f is the neuron activation function, and $\mathbf{W}^{(\text{in})}, \mathbf{W}^{(\text{out})}$ are respectively the input and output weight matrices. Memory is achieved with recurrent connections $\mathbf{W}^{(\text{h})}$ in the hidden neurons $\mathbf{h}^{(\nu)}$, meaning that for all the other inputs, the inference relation is recurrently defined as

$$\begin{cases} \mathbf{h}^{(\nu)} & = f\left(\mathbf{W}^{(\text{in})}\mathbf{x}^{(\nu)} + \mathbf{W}^{(\text{h})}\mathbf{h}^{(\nu-1)}\right) \\ \mathbf{y}^{(\nu)} & = \mathbf{W}^{(\text{out})}\mathbf{h}^{(\nu)}. \end{cases} \tag{3.2}$$

The unfolded representation of the fully connected RNN in Fig. 3.1b illustrates the recurrence relation. To account for this recurrence, the back-propagation algorithm needed to be adapted, hence the conception of the back-propagation through time (BPTT) algorithm. As can be observed in the unfolded representation, to compute a gradient at discrete time ν, the algorithm must carry out the backward pass through all the previous hidden layers $\mathbf{h}^{(\nu' \leq \nu)}$.

As a consequence, the backward pass goes through a number of layers as large as the time series dataset length. However in deep learning, computing gradients through a large number of layers is often the source of vanishing or exploding gradients, where gradient magnitudes increase or decrease exponentially from layer to layer, making deep networks hard to train [127]. Modern RNN architectures such as long short term memory (LSTM) aim to avoid this problem.

To adress this issue, two different frameworks later encompassed in the 'reservoir computing' paradigm were developed independently. Contrary to liquid state machines, which were proposed as a means of explaining how the brain operated with spiking neural networks, echo state networks suggested an alternative training method to avoid BPTT problems. In the next subsection, we will formulate the structure of ESN as it illustrates the RC paradigm well.

3.2 Example of a software reservoir computer: Echo state networks

Echo state networks are an instance of reservoir computers, which we present here to introduce the reader to important concepts in reservoir computing, that are nonlinearity, and memory.

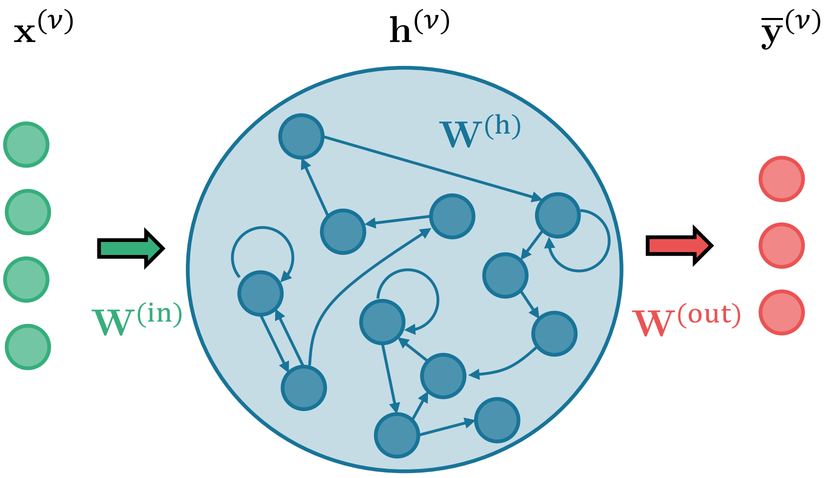

$\mathbf{x}^{(\nu)}$
$\mathbf{h}^{(\nu)}$
$\bar{\mathbf{y}}^{(\nu)}$
$\mathbf{W}^{(\text{h})}$
$\mathbf{W}^{(\text{in})}$
$\mathbf{W}^{(\text{out})}$

Figure 3.2: Sketch of an echo state network (ESN), a static software reservoir architecture. Like in recurrent neural networks, the input time series $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_{\nu}$ is encoded into the ESN hidden layer $\mathbf{h}^{(\nu)}$ with the linear weights $\mathbf{W}^{(\text{in})}$. The neurons are interconnected with fixed and random weights $\mathbf{W}^{(\text{h})}$. Then the prediction $\bar{\mathbf{Y}} = \left\{\bar{\mathbf{y}}^{(\nu)}\right\}_{\nu}$ is obtained by multiplying the output $\mathbf{f}^{(\nu)} = \mathbf{h}^{(\nu)}$ with a linear layer $\mathbf{W}^{(\text{out})}$. It is the only learned parameter, enabling its learning through linear regression.

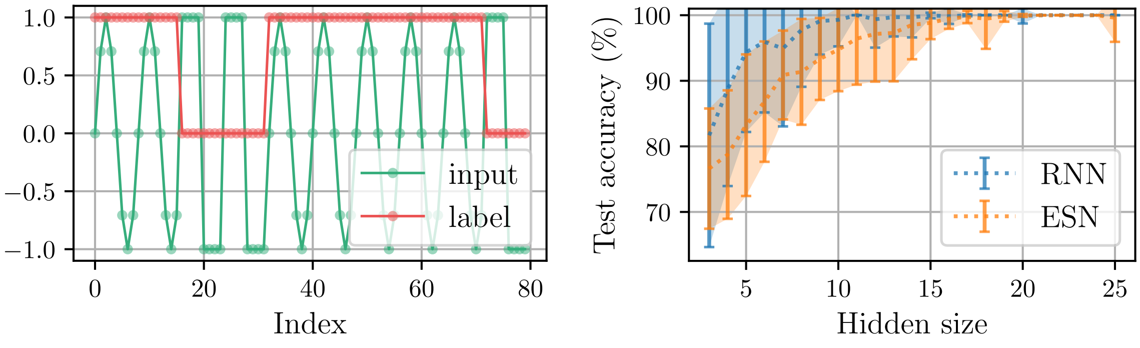
The architecture of an ESN we illustrate in Fig. 3.2 is identical to the fully connected RNN detailed in the previous section, with two key differences: (i) the connectivity and weights of $\mathbf{W}^{(\text{h})}$ and $\mathbf{W}^{(\text{in})}$ are fixed and random (ii) only the output weights $\mathbf{W}^{(\text{out})}$ are learned. In the reservoir computing community, hidden neurons belonging to the hidden layer $\mathbf{h}^{(\nu)}$ are commonly called the reservoir nodes. We present a setting where the echo state network output $\mathbf{f}^{(\nu)}$ is the full hidden layer $\mathbf{h}^{(\nu)}$. In the case where the cost function C is mean squared error as defined in Eq. (2.10), the optimization can be performed more efficiently than with gradient descent algorithms. Indeed if we want to find the output weight matrix $\mathbf{W}^{(\text{out})}$ which minimizes the mean-square error, then the analytical estimator is [128]

$$\mathbf{W} = \left(\bar{\mathbf{Y}}^{\text{train}}\right)^{+}\mathbf{F}, \tag{3.3}$$

where $\mathbf{F}$ is the matrix whose rows are the reservoir outputs $\mathbf{f}^{(\nu)}$, and A^{+} denotes the Moore-Penrose pseudo-inverse of a matrix A, a generalization of inversion to non-square matrices.

Although Eq. (3.3) only operates a linear regression, the ESN nodes performs a nonlinear transformation on the input, meaning ESN can still fit linearly inseparable datasets. Interestingly, comparing RC models to RNN on chaotic time series tasks shows competitive performance [129, 130, 131] despite the simpler training method.

We compare the performance of ESN and RNN for the common benchmark task of sin/square waveform classification in Fig. 3.3. It is a time series dataset shown in



(a)

(b)

Figure 3.3: Performance of an echo state network with ReLU activation functions on the sin/square waveform classification task. (a) The input data is a time series of points belonging to a sine or a square discretized in 8 points. Labels are $+1$ or 0, if the input belongs to a sine or square waveform respectively. (b) Average accuracy of the ESN and fully connected RNN on the test dataset as a function of the hidden layer size $\mathbf{h}^{(t)}$, over 100 different initializations. Error bars represent $\pm$ standard deviation.

Fig. 3.3a, whose input points belong to a sine or a square discretized in 8 points: a sine waveform is $[0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071]$, while a square is $[1, 1, 1, 1, 0, 0, 0, 0]$. The label associated to an input is $+1$ if it belongs to a sine, and 0 if it belongs to a square. The sin/square time series dataset is useful to probe nonlinearity of a reservoir, as a single threshold value cannot separate points from the sine or square waveforms. Furthermore, memory is required to solve this task, as ± 1 points can belong to both a sine or square, and are distinguished using their precedent inputs. These characteristics have made this time series a popular benchmark task in the reservoir computing community, to probe nonlinearity and memory properties of a reservoir.

To solve this task, we use ESN's of increasing hidden layer size. Since reservoir outputs $\mathbf{f}^{(\nu)}$ depend on the randomly initialized $\mathbf{W}^{(\mathrm{in})}$ and $\mathbf{W}^{(\mathrm{h})}$, the performance of the network relative to its hidden layer size is displayed in Fig. 3.3b after averaging over 100 random initializations. The figure shows that around 17 hidden neurons are required to solve the sin/square classification task with $100\,\%$ test accuracy.

We also plot the performance of the RNN architecture presented in the previous section, and find that it needs 11 hidden neurons to solve the task. Comparing the two models reveals that although the RNN can classify the time series with 6 less hidden neurons, its optimization with gradient descent is more time consuming than the pseudo-inverse required to optimize the ESN, and the number of hidden neurons is nonetheless of the same order of magnitude.

3.3 Quantum reservoir computing with qubits

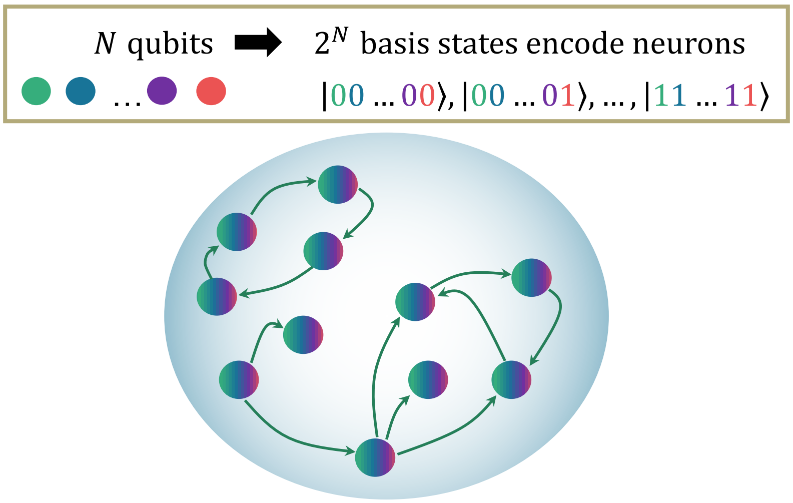

N qubits
2^N basis states encode neurons
$|00\ldots00\rangle, |00\ldots01\rangle, \ldots, |11\ldots11\rangle$

Figure 3.4: Sketch of the exponential growth of the Hilbert space. The Hilbert space is represented by the big light blue circle, and small multi-colored circles correspond to joint qubit states. Arrows linking them symbolize hopping probability amplitudes. The number of basis states scales as 2^N, where N is the number of qubits.

The first proposal of QRC was in the simulation article Ref. [42], with a set of n_q qubits arranged in a two-dimensional lattice with random nearest-neighbor hopping. Because the Hilbert space size is 2^{n_q} as illustrated in Fig. 3.4, the reservoir size scales exponentially with the qubit number. Another advantage of using qubits is their omnipresence in the field of quantum computing. From an experimental standpoint, quantum computers containing $O(50-100)$ noisy qubits are now commercially available, for instance those produced by IBM [132] and Google [93]. Moderate noise can even be taken into account in neuromorphic computing, making these NISQ qubits readily available in QRC.

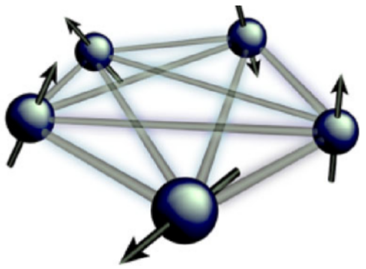


Figure 3.5: Illustration of 5 qubits, with two by two coupling described by the transverse-field Ising model. Here they are represented as spins (arrows), which are natural two-level systems defined by their up and down states. Lines connecting the spins represent spin to spin coupling. Figure taken from Ref. [42].

An extensively studied and very general qubit array architecture is the transverse-field Ising model. Here spins are considered as qubits, because they are natural two-level systems. The configuration generating this model is illustrated Fig. 3.5. Let $(\hat{\sigma}_x)_j$, $(\hat{\sigma}_y)_j$, $(\hat{\sigma}_z)_j$ respectively be the X, Y and Z direction Pauli operators acting on qubit number j, then its Hamiltonian is

$$\hat{H} = \sum_{i,j=1}^{n_\mathrm{q}} J_{ij} (\hat{\sigma}_x)_i (\hat{\sigma}_x)_j + h \sum_{j=1}^{n_\mathrm{q}} (\hat{\sigma}_z)_j, \tag{3.4}$$

where J_{ij} is the spin-spin coupling energy, and h is the magnetic field value [113, 112]. $\{J_{ij}\}_{i,j \in [1,n_\mathrm{q}]}$ and $h \in \mathbb{R}$ constitute the set of reservoir hyper-parameters $\boldsymbol{\theta}$. Initially studied due to its application in the understanding of ferromagnetism [133], this model has well known dynamical regimes [134] and can be implemented in liquid and solid-state NMR systems with nuclear and electron spin ensembles [135, 136]. Ising models implemented in superconducting qubits have also been used for quantum annealing applications [137].

A possible way of encoding a scalar input $\mathbf{x}$ would be initially setting the first qubit's state to be $\hat{\rho}_\mathbf{x} = |\psi\rangle_\mathbf{x} \langle\psi|_\mathbf{x}$ [42, 112, 134], where

$$|\psi\rangle_\mathbf{x} \coloneqq \sqrt{1-\mathbf{x}}|0\rangle + \sqrt{\mathbf{x}}|1\rangle. \tag{3.5}$$

This encoding method transforms the initial reservoir state $\hat{\rho}_0$ with a map $\mathcal{E}[\mathbf{x}]$:

$$\hat{\rho} \to \mathcal{E}[\mathbf{x}](\hat{\rho}) \overset{\text{def}}{=} \hat{\rho}_\mathbf{x} \otimes \mathrm{Tr}_1[\hat{\rho}], \tag{3.6}$$

where Tr_1 represents the partial trace with respect to the first qubit.

A forward pass is then executed by letting the reservoir state evolve for a set amount of time Δt. Taking into account the qubit decoherence, this time evolution is modeled by a Lindblad master equation

$$\begin{aligned} \frac{d\hat{\rho}}{dt} &= \mathcal{L}_{\boldsymbol{\theta}}(\hat{\rho}) \\ &= -\frac{i}{\hbar}[\hat{H}, \hat{\rho}] + \sum_j \mathcal{D}[\hat{C}_j](\hat{\rho}), \end{aligned} \tag{3.7}$$

where $\mathcal{L}_{\boldsymbol{\theta}}$ is the superoperator describing the dynamics with the $\boldsymbol{\theta}$ subscript indicating the influence of the reservoir hyper-parameters, and

$$\mathcal{D}[\hat{C}](\hat{\rho}) \overset{\text{def}}{=} \hat{C}\hat{\rho}\hat{C}^\dagger - \frac{1}{2}\left\{\hat{C}^\dagger\hat{C}, \hat{\rho}\right\} \tag{3.8}$$

is the Lindblad dissipation superoperator for any jump operator $\hat{C}$, $\{a, b\} \overset{\text{def}}{=} ab + ba$ is the anticommutator, and $\left\{\hat{C}_j\right\}_j$ is a set of jump operators describing the dissipative part of the dynamics. For example the jump operator $\hat{C}_j = \sqrt{\kappa_j}(\hat{\sigma}^-)_j$ with $(\hat{\sigma}^-)_j = \frac{1}{2}((\hat{\sigma}_x)_j - i(\hat{\sigma}_y)_j)$ models relaxation in qubit j, at a rate κ_j. This dynamical evolution applies a map

$$\hat{\rho} \to \exp(\Delta t \mathcal{L}_{\boldsymbol{\theta}})\hat{\rho} \tag{3.9}$$

to the qubit state $\mathcal{E}[\mathbf{x}](\hat{\rho}_0)$ encoding the input $\mathbf{x}$. For practical discussion around quantum reservoir computing, we define the map combining the input encoding and dynamical evolution for the forward pass in the reservoir

$$\hat{\rho} \to \mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}](\hat{\rho}) \overset{\text{def}}{=} (\exp(\Delta t \mathcal{L}_{\boldsymbol{\theta}}) \circ \mathcal{E}[\mathbf{x}])\, \hat{\rho}. \tag{3.10}$$

This parameterization of the quantum system evolution is analog and accepts dissipation, whereas the unitary gate operations for parameterized quantum circuits in Eq. (2.24) were non-dissipative. As dissipation occurs naturally in physical systems, there are less experimental requirements to implement such analog dissipative forward passes in QRC or PQC [138], compared to implementing a sequence of unitary gates as in quantum algorithms. Indeed, state of the art qubits have high dissipation and decoherence rates. Moreover, quantum reservoir computers actually benefit from dissipation as it implements a memory timescale for the reservoir, and allows classification of time series [139]

After the time evolution, the reservoir output $\mathbf{f}$ is obtained by measuring some observables $\hat{O}_j$, and computing their expectation values

$$\mathbf{f}_j = \mathrm{Tr}\Big(\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}](\hat{\rho})\hat{O}_j\Big). \tag{3.11}$$

These measurements can be projective, where the post-measurement state is an eigenstate of $\hat{O}_j$ as seen in Section 2.2.1.3, or they can be weak, meaning after the measurement we are not completely sure if the state has been projected or not. Weak measurements will be discussed in Section 3.4.3. Different observables have been proposed for measurements, such as Pauli operators [42, 112, 113] or qubit occupations [28] $\frac{1-(\hat{\sigma}_z)_j}{2}$.

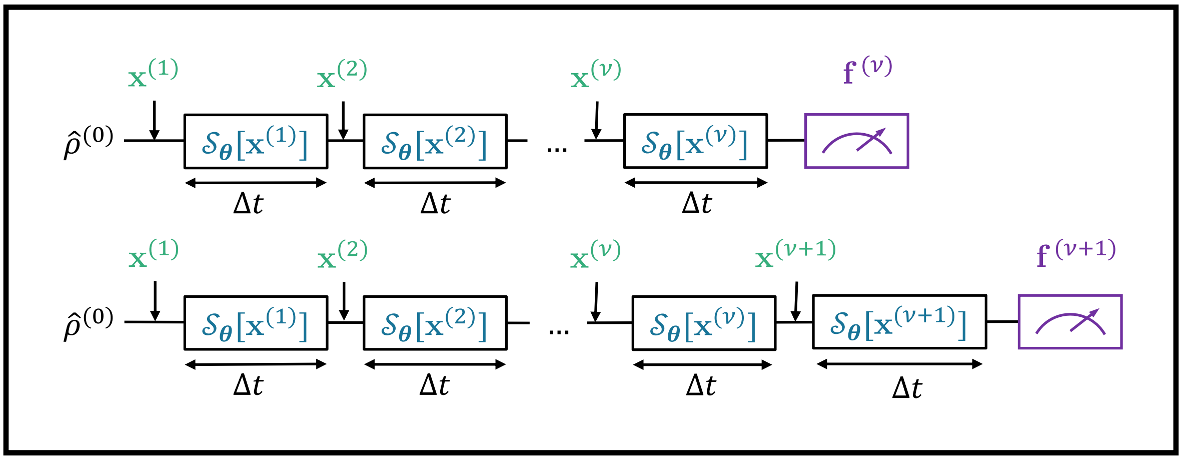
Caution must be taken when considering these reservoir outputs because as illustrated in Section 2.2.1.3, many measurements are required to properly estimate an expectation value.

3.4 Measurement protocols in quantum reservoir computing

Regardless of the measured observables for reservoir outputs, measurements perturb the evolution equations Eq. (1.5) or Eq. (5.48). This stochastic state projection makes the subsequent dynamics depend on the measurement outcome. Measurement back-action plays an important role when processing time-series with quantum reservoirs, because measurements are performed after each processing time step. In the simulations carried out for this thesis I did not take into account back-action for the sake of simplicity, but in experimental work [140] this issue cannot be ignored.

To solve the issue of measurement back-action perturbing system dynamics when processing time series, and the need to measure many copies of a state to estimate observable expectation values, different measurement protocols have been proposed. Most of them are explained and compared in Ref. [112], except continuous weak measurements as implemented in Refs. [67, 114]. In this section, we will introduce these different protocols, adopting the labels they have been given in Ref. [112].

3.4.1 Restarting protocol



$\mathbf{x}^{(1)}$
$\mathbf{x}^{(2)}$
$\mathbf{x}^{(\nu)}$
$\mathbf{f}^{(\nu)}$
$\hat{\rho}^{(0)}$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(1)}]$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(2)}]$
...
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$
Δt
Δt
Δt
$\mathbf{x}^{(1)}$
$\mathbf{x}^{(2)}$
$\mathbf{x}^{(\nu)}$
$\mathbf{x}^{(\nu+1)}$
$\mathbf{f}^{(\nu+1)}$
$\hat{\rho}^{(0)}$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(1)}]$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(2)}]$
...
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu+1)}]$
Δt
Δt
Δt
Δt

Figure 3.6: Restarting protocol. To estimate an element of the reservoir output $\mathbf{f}_j^{(\nu)} = \mathrm{Tr}\left(\hat{\rho}^{(\nu)}\hat{O}_j\right)$ which is the expectation value of the observable $\hat{O}_j$, all previous inputs $\left\{\mathbf{x}^{(\nu')}\right\}_{\nu'\in[1,\nu]}$ are encoded in the reservoir with successive forward pass maps $\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu')}]$, and the dynamics are re-initialized after projectively measuring $\hat{O}_j$.

We describe the restarting protocol, illustrated in Fig. 3.6. Let $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_{\nu\in[1,N_{\text{samples}}]}$ be the input time series dataset of length N_{samples}. The encoding of an input $\mathbf{x}^{(\nu)}$ into the quantum reservoir and evolution of its state during a time Δt is modeled by a forward pass map

$$\hat{\rho} \longrightarrow \mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}](\hat{\rho}), \tag{3.12}$$

where $\boldsymbol{\theta}$ denotes the quantum reservoir hyper-parameters conditioning the reservoir dynamics. We label $\hat{\rho}^{(0)}$ the initial quantum reservoir state. Let $\hat{\rho}^{(\nu)}$ be the density matrix at time $\nu \times \Delta t$, with $\nu \in \mathbb{N}^*$, then after each sequential input and time evolution, it verifies the recurrence relation

$$\hat{\rho}^{(\nu)} = \mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}](\hat{\rho}^{(\nu-1)}). \tag{3.13}$$

We describe a framework where the reservoir outputs $\mathbf{f}_j^{(\nu)}$ are the expectation values of N_{obs} observables $\left\{\hat{O}_j\right\}_{j\in[1,N_{\text{obs}}]}$, i.e

$$\mathbf{f}_j^{(\nu)} = \mathrm{Tr}\left(\hat{\rho}^{(\nu)}\hat{O}_j\right). \tag{3.14}$$

To estimate $\mathbf{f}_j^{(\nu)}$ using projective measurements, the state $\hat{\rho}^{(\nu)}$ is prepared, measured and re-initialized N_{shots} times. However, to prepare $\hat{\rho}^{(\nu)}$, starting from the initial state $\hat{\rho}^{(0)}$ and without taking into account the measurement back-action, the recurrence relation of Eq. (3.13) must be applied ν times. Therefore the number of forward passes which needs to be applied to estimate $\mathbf{f}_j^{(\nu)}$ is $\nu \times N_{\text{shots}}$. Finally, estimating all of the reservoir outputs requires $\frac{N_{\text{samples}}(N_{\text{samples}}-1)}{2} \times N_{\text{shots}} \times N_{\text{obs}}$ forward pass map executions.

The experiment time of this protocol scales quadratically with the input time series length N_{samples}.

3.4.2 Rewinding protocol

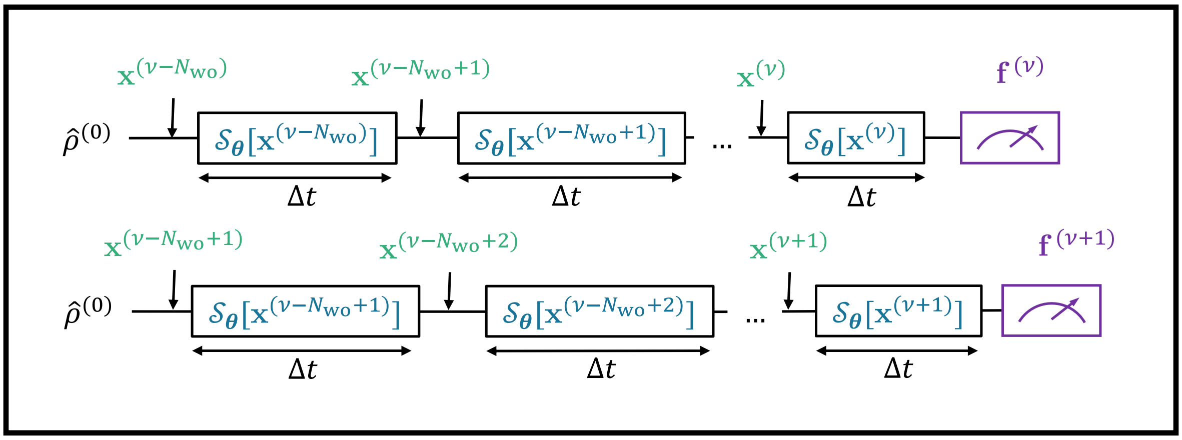

$\mathbf{x}^{(\nu-N_{\text{wo}})}$
$\mathbf{x}^{(\nu-N_{\text{wo}}+1)}$
$\mathbf{x}^{(\nu)}$
$\mathbf{f}^{(\nu)}$
$\hat{\rho}^{(0)}$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu-N_{\text{wo}})}]$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu-N_{\text{wo}}+1)}]$
...
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$
Δt
Δt
Δt
$\mathbf{x}^{(\nu-N_{\text{wo}}+1)}$
$\mathbf{x}^{(\nu-N_{\text{wo}}+2)}$
$\mathbf{x}^{(\nu+1)}$
$\mathbf{f}^{(\nu+1)}$
$\hat{\rho}^{(0)}$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu-N_{\text{wo}}+1)}]$
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu-N_{\text{wo}}+2)}]$
...
$\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu+1)}]$
Δt
Δt
Δt

Figure 3.7: Rewinding protocol. To estimate an element of the reservoir output $\mathbf{f}_j^{(\nu)} = \mathrm{Tr}\left(\hat{\rho}^{(\nu)}\hat{O}_j\right)$, the N_{wo} previous inputs $\{\mathbf{x}^{(\nu')}\}_{\nu'\in[\nu-N_{\text{wo}},\nu]}$ are encoded in the reservoir with successive forward pass maps $\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu')}]$, and the dynamics are re-initialized after projectively measuring $\hat{O}_j$.

A more practical solution can be envisioned, by considering the echo state property [43] of the reservoir. This property states that the reservoir can process time series independently of its initial state $\hat{\rho}^{(0)}$. This implies the existence of a washout time τ_{wo}

after which the dynamical system loses any information on its initial condition. From the washout time and the evolution time between inputs Δt, we define $N_{\mathrm{wo}} = \tau_{\mathrm{wo}}/\Delta t$, the number of inputted data points after which information on the initial condition is lost.

Using this property, the rewinding protocol illustrated in Fig. 3.7 aims to use less resources to overcome measurement back-action than the restarting protocol. The recurrence relation described by Eq. (3.13) is unchanged, but now to prepare $\hat{\rho}^{(\nu)}$ and compute $\mathbf{f}^{(\nu)}$, N_{wo} inputs $\left\{\mathbf{x}^{(\nu')}\right\}_{\nu' \in [\nu - N_{\mathrm{wo}}, \nu]}$ are injected before measurement. Still using projective measurements to estimate reservoir outputs, this means the number of executed forward pass maps is reduced to $N_{\mathrm{wo}} \times N_{\mathrm{samples}} \times N_{\mathrm{shots}} \times N_{\mathrm{obs}}$. Thanks to this number only scaling linearly in the time series length N_{samples}, this protocol is more feasible experimentally than the restarting protocol. Indeed, N_{samples} commonly exceeds 1000.

In the simulations that I have performed in my PhD work, I did not consider the projection of the quantum state by the measurement. However, my simulations have corroborated the experimental work done in out group, lead by Baptiste Carles [140]. For this experimental quantum reservoir computing, implemented using a single bosonic mode of a superconducting resonator coupled to a transmon qubit, the rewinding protocol was used.

3.4.3 Weak measurement protocols

Although the restarting and rewinding protocols work around the effects of measurement back-action, they still require a large amount of dynamical re-initializations and external storage of previous input data. This limitation makes them unsuitable for online learning, in which the reservoir is still trained with sequential data during its practical deployment, after its initial training. An attractive alternative to projective measurement protocols is to instead weakly measure the observables, in order to partially preserve quantum coherences post-measurement.

Indeed, a weak measurement allows for extraction of a small amount of information from a quantum system, in exchange for a weak perturbation of its state. So in quantum reservoir computing they enable information extraction from the quantum system, while keeping a greater part of its fading memory. Two main approaches to weak measurements in QRC have been proposed: either performing weak measurements at discrete time intervals [112], or continuously monitoring the reservoir throughout its dynamical evolution [114]. I will briefly describe their differences and applications, but will not delve into details as I did not explore weak measurements in my work.

3.4.3.1 Online protocol.

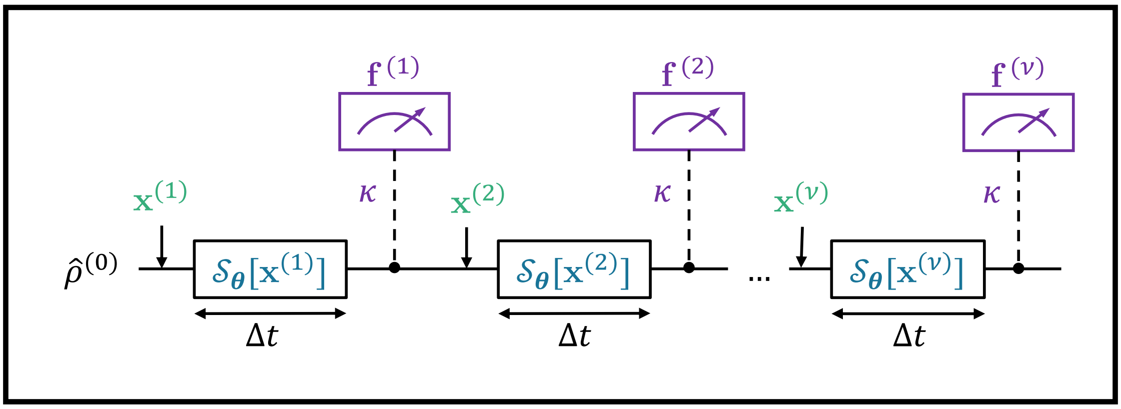


Figure 3.8: Online protocol. To estimate the full reservoir output $\mathbf{F} = \left\{\mathbf{f}^{(\nu)}\right\}_\nu$, the full dataset $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_\nu$ is encoded in the reservoir with successive forward pass maps $\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$, in between which weak measurements of strength κ on observables $\hat{O}_j$ weakly perturb the reservoir state.

A discrete weak measurement parameterized by its strength κ is described by a measurement superoperator $\mathcal{M}\left[\kappa\right]$, whose action on the quantum state $\hat{\rho}$ is expressed as

$$\mathcal{M}\left[\kappa\right](\hat{\rho}) = \sum_j \hat{\Omega}_j\left[\kappa\right]\hat{\rho}\hat{\Omega}_j^\dagger\left[\kappa\right], \tag{3.15}$$

where $\left\{\hat{\Omega}_j\left[\kappa\right]\right\}_j$ is a set of operators verifying

$$\sum_j \hat{\Omega}_j^\dagger\left[\kappa\right]\hat{\Omega}_j\left[\kappa\right] \leq \hat{I}. \tag{3.16}$$

These operators are called 'Kraus operators', and they are generally used to describe physical processes such as measurements or dynamical evolutions [141]. If the measurement strength κ is small, most of the Kraus operators will be small with one being close to the identity $\hat{I}$. On the contrary, if the measurement strength is high, the Kraus operators will become a set of orthogonal projectors.

The online protocol uses such weak measurements after each forward pass map $\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$ to estimate reservoir outputs, as illustrated in Fig. 3.8. The main difference with the restarting and rewinding protocols is that the full dataset is injected into the reservoir at each inference, instead of only injecting parts of it. Because of the measurement back-action, the reservoir state follows a quantum trajectory that depends on the measurement outcomes. We note that the expectation values are still estimated with a finite amount of measurements N_{shots}, meaning the full dataset still needs to be injected N_{shots} times to estimate an expectation value.

Moreover, it is possible to weakly measure different observables simultaneously [142]. In this case, only $N_{\text{shots}} \times N_{\text{samples}}$ forward pass maps are needed to compute the full reservoir output. We observe that for an equal number of measurements N_{shots}, $N_{\text{shots}} \times N_{\text{samples}}$ is lower compared to the $N_{\text{shots}} \times N_{\text{samples}} \times N_{\text{wo}} \times N_{\text{obs}}$ forward pass maps required in the rewinding protocol. However, in practice more copies of a state are needed to estimate an observable's expectation value using weak measurements compared to projective measurements, because less information is extracted. Nevertheless, a condition for online protocol to be advantageous with respect to the rewinding protocol when measuring single-qubit observables in qubit reservoirs has been found in Ref. [112]. With the definition of κ with respect to the set of Kraus operators $\left\{\hat{\Omega}_j\left[\kappa\right]\right\}$ in this reference, this condition is

$$\kappa \geq \sqrt{\frac{1}{N_{\text{wo}} - 1}}. \tag{3.17}$$

We observe that for low N_{wo}, the measurement strength must be higher to be advantageous. So for tasks with a low N_{wo}, it may be advantageous to use a projective measurement scheme.

3.4.3.2 Online continuous protocol.

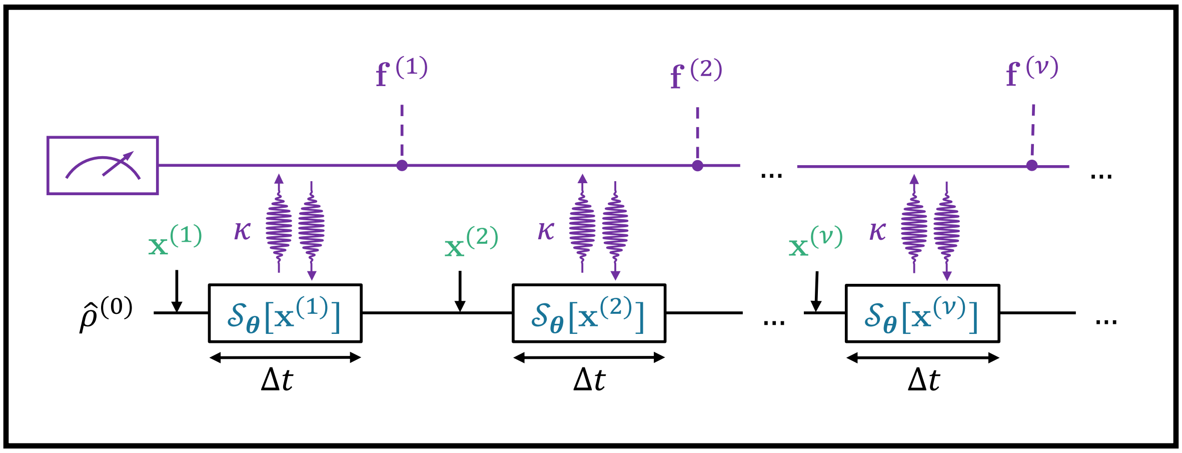


Figure 3.9: Online continuous protocol. To estimate the full reservoir output $\mathbf{F} = \left\{\mathbf{f}^{(\nu)}\right\}_\nu$, the full dataset $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_\nu$ is encoded in the reservoir with successive forward pass maps $\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{x}^{(\nu)}]$. During the dynamical evolutions, the reservoir is continously monitored using a probe, which is coupled to the reservoir with measurement strength κ. Measurement records are sampled after each forward pass map.

A different approach to weak measurements is to instead continuously measure the quantum system, using a probe which is weakly coupled to it. The probe is measured continuously. This can be modeled by a quantum reservoir whose observables $\left\{\hat{O}_k\right\}_{k\in[1,N_{\text{obs}}]}$

are each continuously monitored through homodyne detection with respective measurement strengths κ_k. When the measurement records $\boldsymbol{J}$ are known, the evolution of the density matrix $\hat{\rho}_J(t)$ conditioned on the measurement records, can be described by the stochastic master equation [141]

$$d\hat{\rho}_J = \mathcal{L}_{\boldsymbol{\theta}}(\hat{\rho}_J)dt + \sum_{k=1}^{N_{\text{obs}}} \mathcal{D}[\sqrt{\kappa_k}\hat{O}_k](\hat{\rho}_J)dt + \sum_{k=1}^{N_{\text{obs}}} \mathcal{H}[\sqrt{\kappa_k}\hat{O}_k](\hat{\rho}_J)dW_k, \tag{3.18}$$

where $\mathcal{L}_{\boldsymbol{\theta}}$ describes the hermitian reservoir dynamics, and $\mathcal{D}[\sqrt{\kappa_k}\hat{O}_k]$ are Lindblad dissipation superoperators describing coupling of the system to the probe. We define the stochastic superoperator for the jump operator $\hat{C}$

$$\mathcal{H}[\hat{C}](\hat{\rho}_J) \overset{\text{def}}{=} \hat{C}\hat{\rho}_J + \hat{\rho}_J\hat{C}^\dagger - \operatorname{Tr}\left(\hat{C}\hat{\rho}_J + \hat{\rho}_J\hat{C}^\dagger\right)\hat{\rho}_J, \tag{3.19}$$

which describes information gained from the measurement. The stochasticity of measurement is modeled by a Wiener process, because continuous measurement can be considered as the continuous limit of infinitesimally short measurements: the Wiener increment dW_j is drawn from a Gaussian distribution of zero mean and $\sqrt{dt}$ standard deviation. The measurement records are then

$$\boldsymbol{J}_k = \left\langle \hat{O}_k \right\rangle_J + dW_k/dt, \tag{3.20}$$

where $\left\langle \hat{O}_k \right\rangle_J \overset{\text{def}}{=} \operatorname{Tr}\left(\hat{\rho}_J\hat{O}_k\right)$. When averaging the stochastic quantum trajectories over an infinite number N_{shots} of measured conditional states $\hat{\rho}_J$, the effects of measurement back-actions dW_k average out. In this case, the stochastic terms in Eq. (3.18) disappear and the master equation becomes fully deterministic

$$\frac{d\hat{\rho}}{dt} = \mathcal{L}_{\boldsymbol{\theta}}(\hat{\rho}) + \sum_{k=1}^{N_{\text{obs}}} \mathcal{D}[\sqrt{\kappa_k}\hat{O}_k](\hat{\rho}). \tag{3.21}$$

In the deterministic limit, the measurement records are $\boldsymbol{J}_k = \left\langle \hat{O}_k \right\rangle = \operatorname{Tr}\left(\hat{\rho}\hat{O}_k\right)$.

A continuous weak measurement framework for quantum reservoir computing has been proposed in Ref. [114]. Because it has not been given a name in the article, we label it the online continuous protocol in this manuscript. Its operation is schematized in Fig. 3.9. While measurement records are sampled after the forward pass maps like in previously mentioned protocols, the continuous monitoring introduces additional dissipation during the dynamics. However, the continuity of measurement removes the question of measurement duration present in all the previous protocols: in theoretical proposals of quantum reservoir computing, this parameter is often ignored, while it may be an important resource in experimental implementations.

The interests of the online continuous protocol are similar to that of the online protocol, and a balance must be adjusted between measurement strength κ and number of measurements N_{shots}. A higher measurement strength implies that a smaller number of samples is required to estimate $\left\langle \hat{O}_k \right\rangle_J$, but it also increases the dissipation, thus limiting the reservoir memory. We note that Ref. [114] considers an infinite amount of measurements, so no optimization of resources on this subject has been researched.

3.5 Applications of quantum reservoir computers

The QRC framework described in Section 3.3 enables harnessing the large feature space offered by quantum devices, while avoiding barren plateau phenomena which pose an obstacle to training parametric quantum circuits for their use in quantum neural networks, or other variational quantum algorithms. In this section, we present different applications for which they have been considered, and show their advantages and flaws.

Classical data processing. On classical time prediction tasks, arrays of 5 to 7 qubits have been shown to perform as well as echo state networks of 100 nodes [42]. This can be understood because 7 qubits translates to $2^7 = 128$ basis state neurons in the quantum reservoir. Large scale implementations of QRC have shown that its performance scales better in size than its classical counterpart, and the improvement of their performance with size has not yet been found to plateau [143]. However, scaling up physical reservoir computers is difficult because of the innate difficulty in connecting a large amount of devices together [144]. For this reason, as of now quantum reservoirs do not outperform state of the art classical recurrent neural networks in classical time prediction tasks.

Quantum data processing. However, if the input is not a classical time series but rather a quantum state, it was shown in Ref. [28] than QRC can be used to estimate multiple properties from a single measurement or even perform tomography. This approach hopes to reduce the number of observables to measure for the estimation of quantum properties such as entanglement and polarization [28, 29, 30]. QRC could also increase the fidelity of tomography compared to conventional approaches such as optimal filters [27].

More generally, if the quantum reservoir is utilized to process quantum data, then near-term applications become more feasible. A different advantage than a purely computational one can be achieved: by directly feeding data to a quantum reservoir from a quantum sensor without a digitization process, it is possible to use fewer data samples compared to classical post-processing methods [145].

For quantum data processing applications, quantum reservoirs composed only of qubits run into the problem of needing to couple many quantum devices for good performance, resulting in complex fabrication processes and large system size. Ideally, QRC implementations in quantum sensing should be lightweight and fit into a quantum sensing setup. An attractive addition to these reservoirs would be quantum systems with larger individual Hilbert spaces: bosonic modes are a good candidate to this end.

Chapter 4

Quantum reservoir computing with bosonic modes

Thanks to their larger Hilbert space, bosonic modes hold the promise of increasing the performance of quantum reservoir computers with a lower number of coupled devices, compared to qubits. In this chapter, I will first explain in Section 4.1 the characteristics of the Hilbert space in a bosonic mode. Because a bosonic mode has linear dynamics, I will show in Section 4.2 how quantum reservoir computing frameworks with bosonic modes proposed in the scientific literature have introduced nonlinearity, using non-quadratic Hamiltonians and the nonlinearity of some encoding schemes. In my PhD, we have chosen to harness the nonlinearity introduced by measurements in the Fock basis. In Section 4.3, we benchmark the performance of this bosonic quantum reservoir on the sin/square waveform classification task and Mackey-Glass time series task.

4.1 Hilbert space of bosonic modes

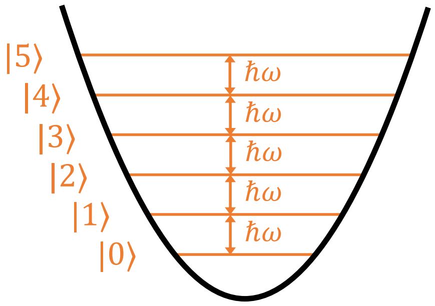


Figure 4.1: Illustration of the energy levels in a single bosonic mode. Each Fock state $|n\rangle$ is an eigenstate containing n photons, with an energy $n\hbar\omega$.

We remind the Hamiltonian of a bosonic mode is that of a quantum oscillator

$$\hat{H} = \hbar\omega\hat{a}^\dagger\hat{a}, \tag{4.1}$$

where ω is the resonance frequency, and $\hat{a}$ is labeled either the ladder annihilation operator, or field operator. It has bosonic statistics, meaning that $\hat{a}$ verifies the commutation relations

$$\left[\hat{a}, \hat{a}^\dagger\right] = \hat{I}, \quad [\hat{a}, \hat{a}] = 0. \tag{4.2}$$

The Hamiltonian of Eq. (4.1) has a discrete eigenspectrum, whose eigenstates are called Fock states $|n\rangle$. Each Fock state corresponds to a well-defined integer number of bosons n, with associated energy $\hbar\omega n$,

$$\hat{H}\,|n\rangle = \hbar\omega n\,|n\rangle\,. \tag{4.3}$$

Bosonic field operators verify the relations

$$\begin{cases} \hat{a}\,|n\rangle & = \sqrt{n}\,|n-1\rangle \\ \hat{a}^\dagger\,|n\rangle & = \sqrt{n+1}\,|n+1\rangle\,. \end{cases} \tag{4.4}$$

We observe that contrary to qubits, a single bosonic mode possesses an infinitely large Hilbert space, i.e their basis is composed of an infinite number of states. This makes them theoretically impossible to simulate by classical means. In practice, when weakly driving a bosonic mode, we create coherent states

$$|\alpha\rangle \overset{\text{def}}{=} e^{-\frac{|\alpha|^2}{2}} \sum_{k=0}^{\infty} \frac{\alpha^k}{\sqrt{k!}}\,|k\rangle \tag{4.5}$$

defined by their displacement α. The probability amplitude of measuring $|n\rangle\langle n|$ converges to 0 for high n,

$$|\langle n|\alpha\rangle|^2 = e^{-|\alpha|^2}\frac{|\alpha|^{2n}}{n!} \underset{n\to\infty}{=} 0. \tag{4.6}$$

So we observe that for low displacement α, only a limited amount of Fock states is significantly populated. The vanishing of their Fock state occupation probability for high photon number is observed in Fig. 4.2, where these probabilities are shown for two different coherent states $|\alpha = 0.9\rangle$ and $|\alpha = 2.0\rangle$. Respectively, they have negligible occupation probabilities for $n > 4$ and $n > 11$. This means that the dynamics of these single-mode coherent states can be accurately modeled in a Hilbert space of finite dimension 12.

The dimension of the Hilbert space of M coupled bosonic modes, whose individual dynamics could be captured with a cutoff $n_\mathcal{H}$, is $(n_\mathcal{H}+1)^M$. As Hilbert space size plays a major role in the processing power of QRC, bosonic mode reservoirs have the potential

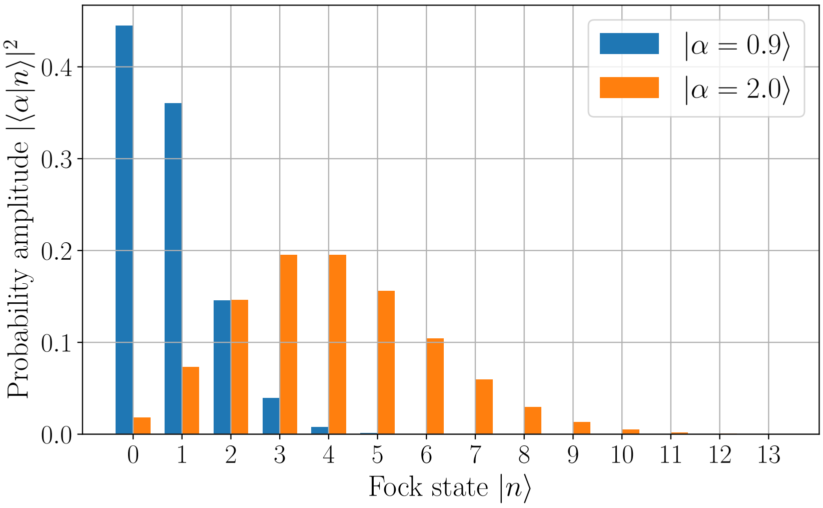


Figure 4.2: Histogram of Fock state $|n\rangle$ occupation probability amplitudes in single mode coherent states $|\alpha = 0.9\rangle$ (blue bars) or $|\alpha = 2.0\rangle$ (orange bars).

to be exponentially more powerful than their qubit counterparts, whose Hilbert space size is 2^M.

Bosonic modes can be experimentally implemented in different physical media, such as optomechanics [57], optical ring resonators [58] and circuit quantum electrodynamics (cQED) [59], in which photons are the bosons of interest. In this thesis we will consider electromagnetic modes of superconducting resonators [60], but the results are applicable to any hardware realizing bosonic modes.

4.2 Introducing nonlinearity in bosonic quantum reservoirs

Despite their large Hilbert space, a major obstacle to using bosonic modes for QRC is the linearity of their dynamics [80]. For instance with a single mode as defined in Eq. (4.1), writing the Heisenberg equation of motion of its field operator in the Heisenberg representation

$$\hat{a}(t) \overset{\text{def}}{=} \exp\left(\frac{i}{\hbar}\hat{H}t\right)\hat{a}\exp\left(-\frac{i}{\hbar}\hat{H}t\right) \tag{4.7}$$

reveals a first order linear differential equation

$$\begin{aligned}\frac{d\hat{a}}{dt}(t) &= -\frac{i}{\hbar}\left[\hat{a}(t), \hat{H}\right] \\ &= i\omega\hat{a}(t).\end{aligned} \tag{4.8}$$

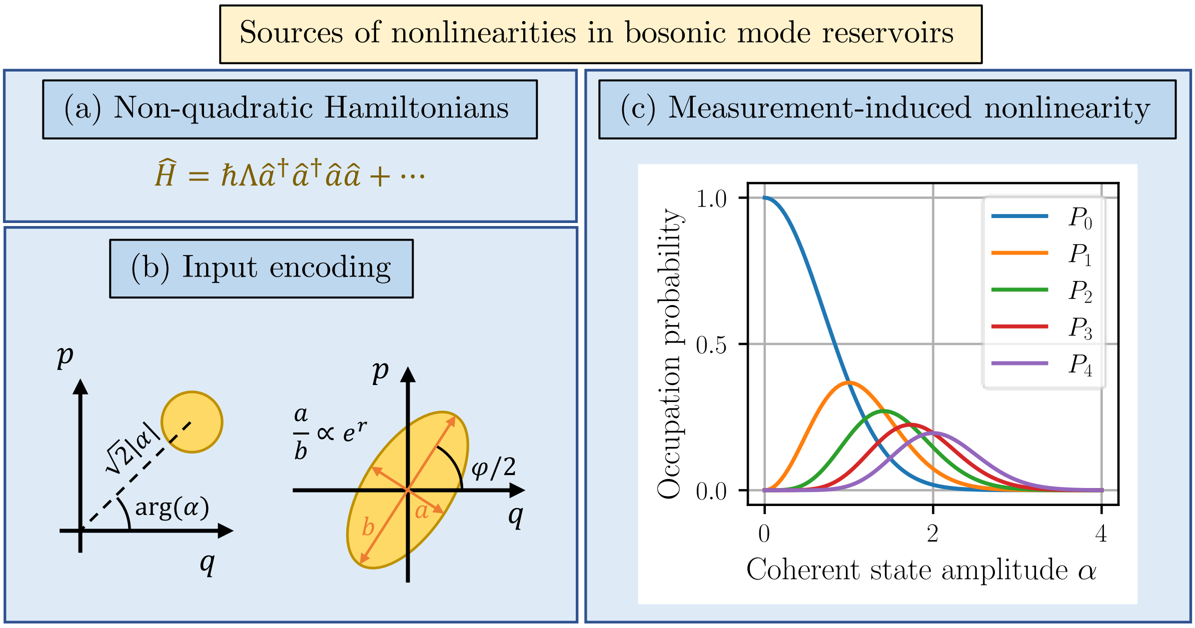

Sources of nonlinearities in bosonic mode reservoirs
(a) Non-quadratic Hamiltonians
$\widehat{H} = \hbar\Lambda\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a} + \cdots$
(b) Input encoding
p
q
$\sqrt{2}|\alpha|$
arg(α)
p
q
$\frac{a}{b} \propto e^r$
$\varphi/2$
a
b
(c) Measurement-induced nonlinearity
Occupation probability
Coherent state amplitude α
1.0
0.5
0.0
0
2
4
P_0
P_1
P_2
P_3
P_4

Figure 4.3: Different methods to introducing nonlinearity in bosonic mode reservoirs. (a) Example of a non-quadratic Hamiltonian (b) Wigner quasi-probability distribution of encoding states in phase space of position and momentum operators p and q. Dark yellow lines indicate the contours of the distribution. A coherent state of displacement $\alpha \in \mathbb{C}$ can be used to encode an input into its amplitude $|\alpha|$ or phase $\arg(\alpha)$. Similarly, an input may be encoded in squeezed states using the squeezing parameter r or the phase of squeezing φ. (c) Occupation probability P_n of different Fock states $|n\rangle$ in a coherent state $|\alpha\rangle$, as a function of its complex amplitude α.

In general, if a system of bosonic modes has a quadratic Hamiltonian, i.e it is at most a second order polynomial of the field operators $\hat{a}$ and $\hat{a}^\dagger$, then its dynamics will be linear [80].

To work around this issue, three main approaches have stood out to apply bosonic modes in quantum reservoir computing: introducing non-quadratic terms through Kerr effect [64, 27, 56], encoding inputs in a nonlinear manner [66] and measurement-induced nonlinearity [146, 67]. They are illustrated in Fig. 4.3.

In this section, we explain how these methods introduce nonlinearity in the quantum reservoirs, and their advantages as well as disadvantages. Measurement-induced nonlinearity will be explained in the next section, as it is our chosen method to introducing nonlinearity.

Non-quadratic Hamiltonians In Refs. [64, 27, 56], Kerr nonlinearities are leveraged to add non-quadratic terms in the Hamiltonian. Single-mode Kerr nonlinearities take the form

$$\hat{H}_{\text{kerr}} = \hbar \sum_{k=1}^{M} \Lambda_k (\hat{a}_k{}^{\dagger})^2 (\hat{a}_k)^2, \tag{4.9}$$

where Λ_k is the Kerr nonlinearity in each mode k. To illustrate why the dynamics are now nonlinear, we write the evolution equation of the field operator in the Heisenberg picture

$$\begin{aligned} \frac{d\hat{a}}{dt}(t) &= -\frac{i}{\hbar}\left[\hat{a}(t), \hat{H}_{\text{kerr}}\right] \\ &= 2\Lambda \hat{a}^{\dagger}(t)\hat{a}^2(t). \end{aligned} \tag{4.10}$$

This differential equation is indeed nonlinear.

In the semi-classical regime where field operators $\hat{a}(t)$ can be considered as scalar values $a(t)$, the use of bosonic modes has been proposed in Ref. [27] for quantum information tasks. While in this regime there are no advantages of processing capacity over classical reservoirs, the lack of measurement back-action and the deterministic measurement outcomes (as opposed to the probabilistic quantum measurements described in Section 2.2.1.3) means that reservoir outputs can be correctly estimated with less measurements than in the fully quantum case. Indeed their reservoir outputs are not expectation values of quantum operators, but rather deterministic evaluations of the classical quadratures $\text{Re}(a(t))$ and $\text{Im}(a(t))$. In their article, Angelatos *et al* [27] directly coupled the target quantum system with their bosonic reservoir in order to process its data, instead of amplifying the signal and processing it with classical post-processing techniques, which has the drawback of introducing additional amplification noise. They demonstrate that for the joint dispersive measurement of two qubits, the fidelity of their 5-mode semi-classical Kerr oscillator reservoir matches that of the conventional optimized readout system using a matched filter, while needing a ≈ 10 times smaller training set.

Instead, operating Kerr nonlinear bosonic modes in the quantum regime as proposed in Refs. [64, 56, 147, 55, 148, 87, 140] could allow for a improved performance of reservoir computing with quantum oscillators over classical equivalents. For instance Govia *et al* show in Ref. [64] that a single quantum nonlinear oscillator reservoir has better performance on average than its classical equivalent on the sine wave phase estimation task, for small training set sizes ($< 10^3$). Also on a single quantum nonlinear oscillator, Motamedi *et al* find in Ref. [87] that high quantumness (quantified by Lee-Jeong's measure [149]) provides a broader set of outcomes (depending on the random reservoir hyper-parameters), in which a better performance than the classical equivalent is achieved. However, they highlight that quantumness alone does not guarantee good performance.

As expected, studies in the classical or quantum regime both observe that Kerr nonlinearity plays a key role for reservoir performance. In optics, large nonlinearities can be challenging to implement experimentally [150, 151], and high photon numbers are required to observe nonlinear interactions. But in superconducting circuits, large Kerr nonlinearities can be engineered using a Josephson Ring [61].

Input encoding An alternative approach is to obtain the nonlinearity through the drive which encodes the data. Whereas for qubit reservoirs we have described a setting where the input is encoded in the initial state, here we describe a setting where a drive applied at the resonance frequency of the bosonic mode encodes the input data. The application of such a drive results in the Hamiltonian

$$\hat{H}_{\text{drive}} = i\hbar\sqrt{\kappa}\left(\epsilon^* \hat{a} - \epsilon \hat{a}^\dagger\right), \tag{4.11}$$

where κ is the coupling strength to the input device, and ϵ the complex drive amplitude, corresponding to the displacement of the coherent state in the input device .

As demonstrated in Ref. [66], encoding the data in the phase of the drive $\arg(\epsilon)$, instead of its amplitude $|\epsilon|$, is sufficient to process nonlinear data using a reservoir of coupled bosonic modes with quadratic Hamiltonians. They also explore encoding into the single-mode squeezing parameters of the input device state, and observe that this increases the performance with respect to encoding into the drive displacements. Other forms of encoding have also been proposed to introduce nonlinearity, such as quantum teleportation [152].

The downside of this approach is that it is not applicable to quantum input data.

4.3 Quantum reservoir computing with coupled bosonic modes measured in the Fock basis

The approach we have chosen during my PhD to introduce nonlinearity in the bosonic reservoir is through quantum non-Gaussian measurements. I focused on measuring Fock state occupations $|n\rangle\langle n|$, because of their availability in superconducting circuits through dispersive coupling to a qubit. This measurement nonlinearity is apparent when showing in Fig. 4.3c the Poissonian distribution obtained when measuring $|n\rangle\langle n|$ in a single mode coherent state $|\alpha\rangle$

$$|\langle n|\alpha\rangle|^2 = e^{-|\alpha|^2}\frac{|\alpha|^2}{n!}. \tag{4.12}$$

In order to prove whether Fock basis measurements introduce sufficient nonlinearity to solve nonlinear tasks, we consider in this section a minimal system of two coherently coupled bosonic modes, in the reservoir computing framework. This system is illustrated in Fig. 4.4.

Although I have considered projective measurements of Fock state occupations, other measurement schemes have different advantages or disadvantages. For instance Ref. [67] implements measurements of the parity operator $\exp\left(i\pi\hat{a}^{\dagger}\hat{a}\right)$ in experiment. They extract the measurement result, and the bosonic mode state is projected into superpositions of either even or odd Fock states, to introduce nonlinearity. The reservoir output is then obtained by dispersively coupling a qubit to the bosonic mode, applying a sequence of unitary gates to it, then projectively measuring it. Both the Fock state measurement that we perform and the parity measurement are quantum non demolition measurements. However, the parity measurement preserves the mode in a superposition of states which might lead to some quantum advantage. The downside might be that parity measurements are harder to interpret than Fock state measurements.

4.3.1 Dynamical description of the two-mode quantum reservoir with coherent coupling

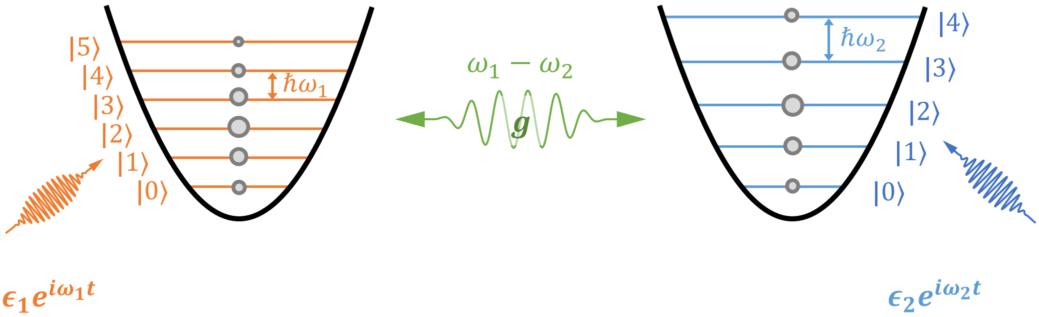


Figure 4.4: Sketch of energy levels of two bosonic modes, with respective resonance frequencies ω_1 and ω_2. We choose to apply to each mode resonant drives of amplitude $\epsilon_{1,2}$ in order to create coherent states (gray circles). The modes can be coupled parametrically by introducing a nonlinear element, such as the Josephson junction in superconducting circuits [61, 62]. A coherent coupling can be engineered with a three-wave mixing interaction, in which a pump tone applied at the difference of the resonance frequencies creates photon conversion at a rate g (green arrow).

Coherent coupling between the two modes, labeled by their respective field operators $\hat{a}_1$ and $\hat{a}_2$, is modeled by the Hamiltonian

$$\hat{H} = g\hat{a}_1{}^{\dagger}\hat{a}_2 + g^{*}\hat{a}_1\hat{a}_2{}^{\dagger}, \tag{4.13}$$

where g is the photon conversion rate between the two modes. We note that this is a rotating frame Hamiltonian, so the harmonic oscillator term $\omega_1 \hat{a}_1{}^\dagger \hat{a}_1 + \omega_2 \hat{a}_2{}^\dagger \hat{a}_2$ is not present. As we will explain in the next chapter, such a coupling can be engineered by applying a three-wave mixing pump tone at the difference of the mode resonance frequencies. The two coupled modes are illustrated in Fig. 4.4.

To simulate driving by an input source without the system acting back onto it, we use the cascaded formalism [153]. The system Hamiltonian now features an additional drive term

$$\hat{H} = g\hat{a}_1{}^\dagger \hat{a}_2 + g^* \hat{a}_1 \hat{a}_2{}^\dagger + i\hbar \sum_{k=1}^{2} \sqrt{\kappa_k} \left(\epsilon_k{}^* \hat{a}_k - \epsilon_k \hat{a}_k{}^\dagger \right), \tag{4.14}$$

where $\epsilon_k \in \mathbb{C}$ is the complex drive amplitude applied to mode k. This drive term creates coherent states in the bosonic modes, allowing the population of higher level Fock states.

The density matrix of the system $\hat{\rho}$ then evolves according to the Lindblad master equation

$$\frac{d\hat{\rho}}{dt} = -\frac{i}{\hbar}\left[\hat{H}, \hat{\rho}\right] + \sum_{j=1}^{M} \mathcal{D}[\hat{C}_j](\hat{\rho}), \tag{4.15}$$

where the jump operator

$$\hat{C}_j = \sqrt{\kappa_j} \hat{a}_j \tag{4.16}$$

models single photon dissipation at a rate κ_j.

4.3.2 Execution of the forward pass in the quantum reservoir

In this section, we describe how the input data of a time series is injected into the bosonic reservoir, and how we extract information from the reservoir dynamics. Anticipating our benchmarking of the reservoir performance on the sin/square waveform classification task (see Section 3.2 for details on the task) in the next section, we will use a sine waveform $\left\{\mathbf{x}^{(\nu)}\right\}_\nu = [0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071]$ as an example input sequence for the remainder of this section.

Let $\mathbf{X} = \left\{\mathbf{x}^{(\nu)}\right\}_\nu$ be a time series, where inputs $\mathbf{x}^{(\nu)} \in \mathbb{R}$ are 1-dimensional. The bosonic system starts in the vacuum state, and its evolution follows the Lindblad master equation Eq. (4.15). We decide to use the complex drive amplitudes ϵ_1, ϵ_2 to encode the input data, as it is a simple first test. This encoding scheme is inspired by the work of Fujii and Nakajima on qubit reservoirs [42].

In all of the simulations shown in the remainder of this chapter, we set the physical parameters to be

$$\begin{cases} g &= 35\kappa \\ \kappa_1 &= (17/20)\kappa \\ \kappa_2 &= (21/20)\kappa, \end{cases} \tag{4.17}$$

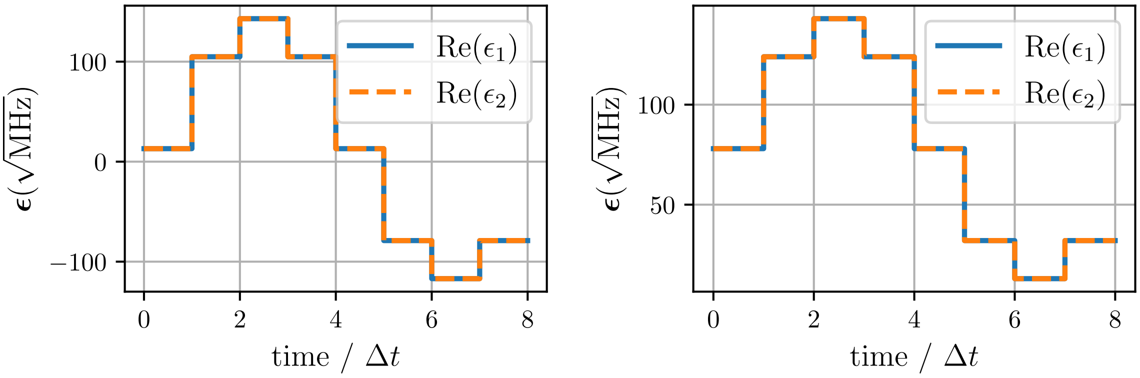
where $\kappa = 20\,\mathrm{MHz}$ is a scaling parameter, close to the value of the dissipation rates.

4.3.2.1 Drive encoding

Let $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2)^T$ be the vector containing the complex drive amplitudes. We choose to encode inputs $\mathbf{x}^{(\nu)}$ into the drive amplitudes $|\boldsymbol{\epsilon}|$, i.e set

$$\boldsymbol{\epsilon}(\mathbf{x}^{(\nu)}) = \boldsymbol{\theta}_0 \cdot \mathbf{x}^{(\nu)} + \boldsymbol{\theta}_{\text{bias}}, \tag{4.18}$$

where the linear encoding weights $\boldsymbol{\theta}_0 \in \mathbb{C}^2$ and the biases $\boldsymbol{\theta}_{\text{bias}} \in \mathbb{C}^2$ are reservoir hyper-parameters. The operator "$\cdot$" denotes the scalar multiplication. This encoding method has also been considered in Refs. [27, 66]. The inputs are sequentially encoded with time intervals $\Delta t = 0.8\kappa^{-1}$, meaning $\mathbf{x}^{(\nu)}$ is encoded at time $\Delta t \times \nu$. For instance in Fig. 4.5a, a sine waveform as defined in Section 3.2 is encoded into the drives, with encoding parameters $\boldsymbol{\theta}_0 = (\bar{\epsilon}, \bar{\epsilon})^T$ and $\boldsymbol{\theta}_{\text{bias}} = 0.1\boldsymbol{\theta}_0$, where $\bar{\epsilon} = 130\sqrt{\kappa/20}$.



(a)

(b)

Figure 4.5: (a) Encoding scheme for a one-dimensional sine input sequence $\{\mathbf{x}^{(\nu)}\}_\nu = [0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071]$ into the drive amplitude $\boldsymbol{\epsilon} \in \mathbb{C}^2$. At time intervals $\Delta t = 0.8\kappa^{-1}$, both drives are updated with the same next input , according to $\boldsymbol{\epsilon}(\mathbf{x}^{(\nu)}) = \boldsymbol{\theta}_0 \cdot \mathbf{x}^{(\nu)} + \boldsymbol{\theta}_{\text{bias}}$. Here the encoding parameters are $\boldsymbol{\theta}_0 = (\bar{\epsilon}, \bar{\epsilon})^T$ and $\boldsymbol{\theta}_{\text{bias}} = 0.1\boldsymbol{\theta}_0$, where $\bar{\epsilon} = 130\sqrt{\kappa/20}$. For visual clarity, Im$(\epsilon_1)$ and Im(ϵ_2) are not plotted since their values are zero. (b) Same encoding scheme, but after renormalizing the inputs to remain in the interval $[0, 1]$.

In order to precisely control the range of values taken by the drive amplitudes, we renormalize the inputs before their encoding into the drive amplitudes, so that their values are always contained between 0 and 1. This is realized by determining the dataset minimum and maximal values $\min(\mathbf{X}), \max(\mathbf{X})$, and applying the transformation

$$\bar{\mathbf{x}}^{(\nu)} = \frac{\mathbf{x}^{(\nu)} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})}. \tag{4.19}$$

As shown in Fig. 4.5b, this renormalization enables the encoded drive absolute values $|\boldsymbol{\epsilon}|$ to remain in the intervals

$$|\epsilon_k| \in \begin{cases} \Big[|(\boldsymbol{\theta}_{\text{bias}})_k|, |(\boldsymbol{\theta}_0)_k|\Big] & \text{if } |(\boldsymbol{\theta}_{\text{bias}})_k| > |(\boldsymbol{\theta}_0)_k| \\ \Big[|(\boldsymbol{\theta}_0)_k|, |(\boldsymbol{\theta}_{\text{bias}})_k|\Big] & \text{if } |(\boldsymbol{\theta}_0)_k| > |(\boldsymbol{\theta}_{\text{bias}})_k|. \end{cases} \tag{4.20}$$

4.3.2.2 Fock state measurement

After encoding the input $\mathbf{x}^{(\nu)}$ at time $\nu \times \Delta t$, and letting the dynamics evolve during a time Δt according to Eq. (4.15), at time $(\nu + 1) \times \Delta t$ we obtain the reservoir outputs $\mathbf{f}^{(\nu)}$ by measuring the occupations in a set of joint Fock states $\{|\vec{n}\rangle\}_{\vec{n}}$ where

$$|\vec{n}\rangle \overset{\text{def}}{=} |n_1\rangle \otimes |n_2\rangle\,, \tag{4.21}$$

and

$$\vec{n} \overset{\text{def}}{=} (n_1, n_2) \tag{4.22}$$

is the vector containing the number of photons n_k in each mode k. The measurement probability $P(\vec{n})$ of the Fock state $|\vec{n}\rangle$ is the expectation value of the projector $|\vec{n}\rangle\langle\vec{n}|$, and can be computed from the Schrödinger representation density matrix $\hat{\rho}$ with

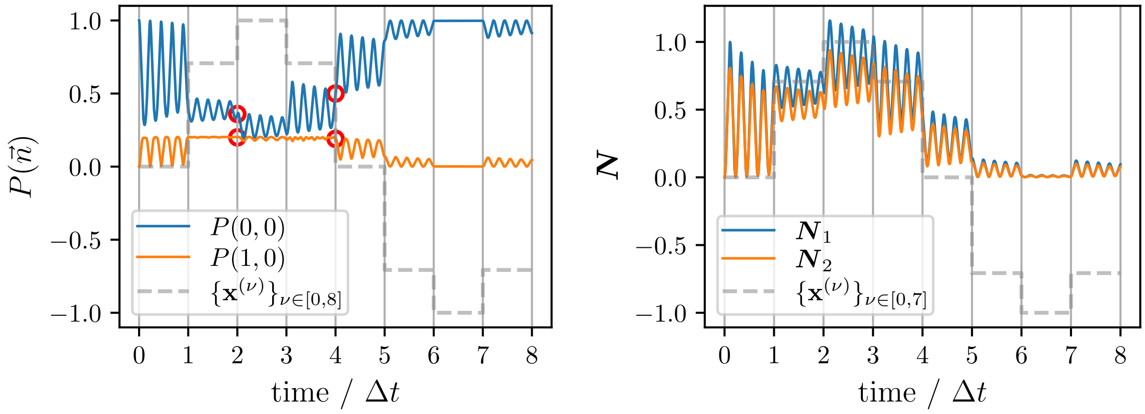
$$P(\vec{n}) \overset{\text{def}}{=} \text{Tr}(\hat{\rho}\,|\vec{n}\rangle\langle\vec{n}|). \tag{4.23}$$

To each reservoir output $\mathbf{f}_k^{(\nu)}$ we attribute a Fock state probability $P(\vec{n})$. In order to learn a bias term in the reservoir output matrix, we append a scalar 1 value to the full reservoir output $\mathbf{f}^{(\nu)}$. From $\mathbf{F}$ the matrix whose rows are the reservoir outputs $\mathbf{f}^{(\nu)}$, we calculate the output weight matrix $\mathbf{W}$ using Moore-Penrose matrix inversion as described in Eq. (3.3).

4.3.2.3 Reservoir dynamics during the forward pass

In order to observe the effect of the input encoding onto the reservoir, it is useful to not only probe the dynamics of Fock state occupation probabilities $P(\vec{n})$ under the drive encoding as described in Section 4.3.2.1, but also that of the mean photon number $\boldsymbol{N}$.

In Fig. 4.6, we present the dynamics of joint Fock state probabilities $P(0,0), P(0,1)$



(a)

(b)

Figure 4.6: Dynamics in the 2-mode bosonic reservoir of (a) two Fock state probability amplitudes $P(0,0), P(0,1)$ and (b) the mean photon numbers $\boldsymbol{N}_1, \boldsymbol{N}_2$ under drive encoding of a sine waveform $\{\mathbf{x}^{(\nu)}\}_{\nu\in[0,7]} = [0, 0.7071, 1, 0.7071, 0, -0.7071, -1, -0.7071]$ renormalized to take values in $[0,1]$ as illustrated in Fig. 4.5b, where new data points are encoded at Δt time intervals. The physical parameters are specified in Eq. (4.17). The measured Fock state probabilities after encoded input values $\mathbf{x}^{(\nu)} = 0.7071$ are highlighted by red circles. We observe that depending on the previous input, the measured probability can be different.

and photon number $\boldsymbol{N}$ resulting from the encoding of a sine waveform into the drive amplitudes following the procedure described in Fig. 4.5b. We observe in Fig. 4.6a that for the same $\mathbf{x}^{(\nu)} = 0.7071$ input value, the Fock state probabilities will have different responses (highlighted by the red circles), depending on the previous inputs $\mathbf{x}^{(\nu'<\nu)}$. This illustrates the memory property of the bosonic modes.

Then studying the evolution of $\boldsymbol{N}$ shows that the chosen encoding parameters $\boldsymbol{\theta}_0 = (\bar{\epsilon}, \bar{\epsilon})^T$ and $\boldsymbol{\theta}_{\text{bias}} = 0.1\boldsymbol{\theta}_0$, where $\bar{\epsilon} = 130\sqrt{\kappa/20}$, generally tend to increase the photon number for higher value inputs, but do not make it go above 10, meaning we are indeed still in the quantum regime. We can thus conclude that the physical parameters we have chosen in Eq. (4.17) and the drive encoding parameters chosen in Fig. 4.5 are good hyper-parameters for the reservoir.

4.3.3 Reservoir performance benchmark on the sin/square waveform classification task

4.3.3.1 Classification of the sin/square dataset

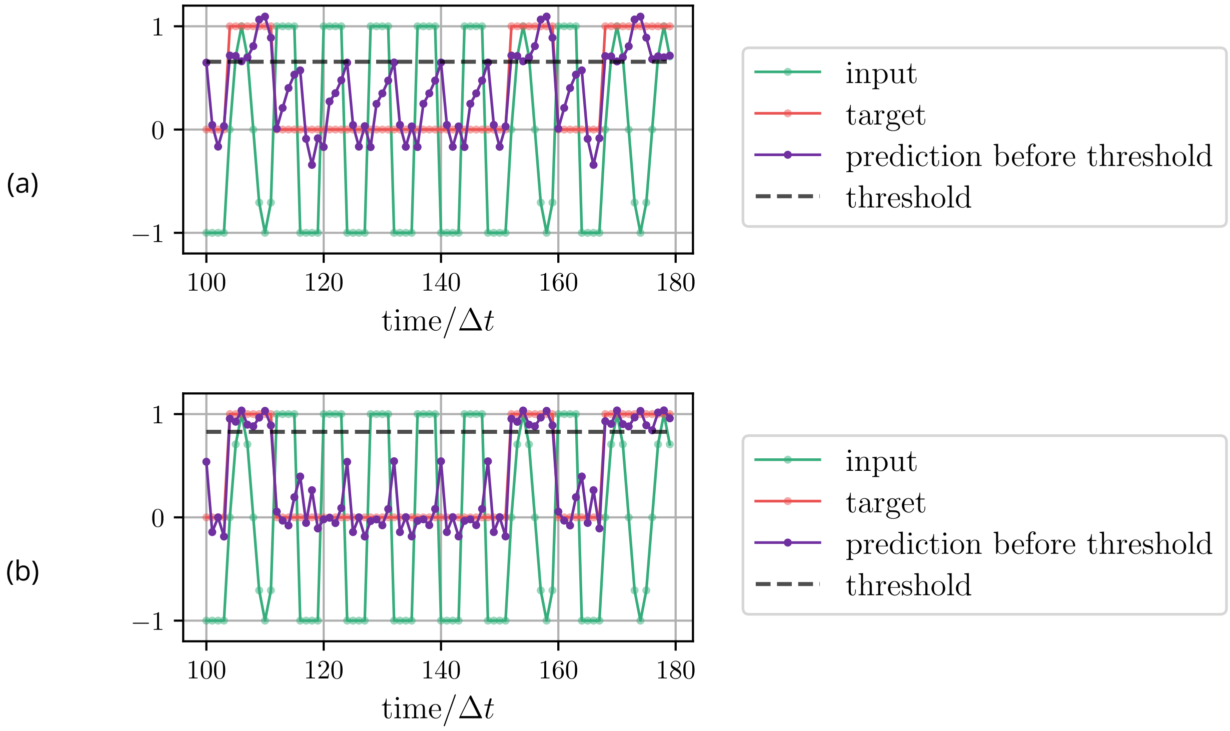


Figure 4.7: Reservoir performance on the sin/square waveform classification task, for (a) 4 or (b) 9 measured Fock probability amplitudes. The input data (green dots) is a time series of points belonging to a sine or square discretized in 8 points. The target label is 1 if the point belongs to a sine, or 0 if it belongs to a square. The prediction (purple dots) is separated by a threshold (black dashed line), for which the prediction is 0 if the output is below the threshold, or 1 if it is above.

In order to evaluate the capacity of the 2-mode bosonic reservoir, we address the sin/square classification task, a standard benchmark already introduced in Section 3.2. As the input data points cannot be linearly separated, this task is chosen to probe the nonlinearity introduced by the Fock state basis measurements. Moreover, memory is required as points equal to ± 1 can both belong to a sine or a square. The forward pass is executed as described in Section 4.3.2, with data points sent at $\Delta t = 0.8\kappa^{-1}$ time intervals.

We separate the dataset into a training and testing set, each containing 200 random sine or square waveforms with 8 points. Therefore, the reservoir is trained on $N_{\text{samples}} = 8 \times 200$ points. We investigate the performance of the quantum reservoir as a function of the number of measured Fock states. First, we measure the states $|0,0\rangle$

to $|1,1\rangle$, yielding 4 output neurons. The one-dimensional reservoir prediction is shown in Fig. 4.7a, and a threshold that separates it into 1 or 0 labels. We set the threshold by selecting a threshold between 0 and 1 which yields the best classification accuracy. The reservoir prediction matches the target with $99.93\,\%$ accuracy. To improve even further the performance, we measure the states $|0,0\rangle$ to $|2,2\rangle$, increasing the number of output neurons to 9. We see the reservoir prediction points are further away from the separating threshold, illustrating the better separation of data thanks to the larger number of neurons. The test accuracy is now $100\,\%$.

4.3.3.2 Comparison of the bosonic reservoir with a static classical reservoir

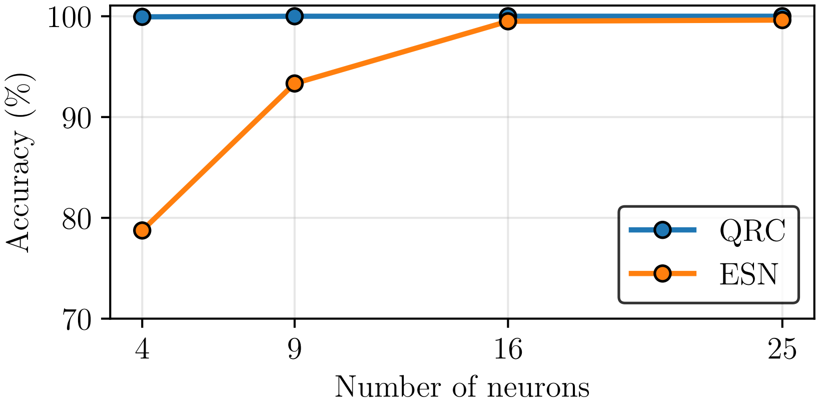


Figure 4.8: Performance comparison of the quantum, and classical reservoirs. Test accuracy on the sin/square waveform classification tasks as a function of the number of reservoir neurons, for a classical reservoir (ESN), and for the bosonic QRC. The classical reservoir accuracy is averaged over 100 random initializations. For the bosonic QRC, the number of measured joint Fock states corresponds to the number of reservoir neurons.

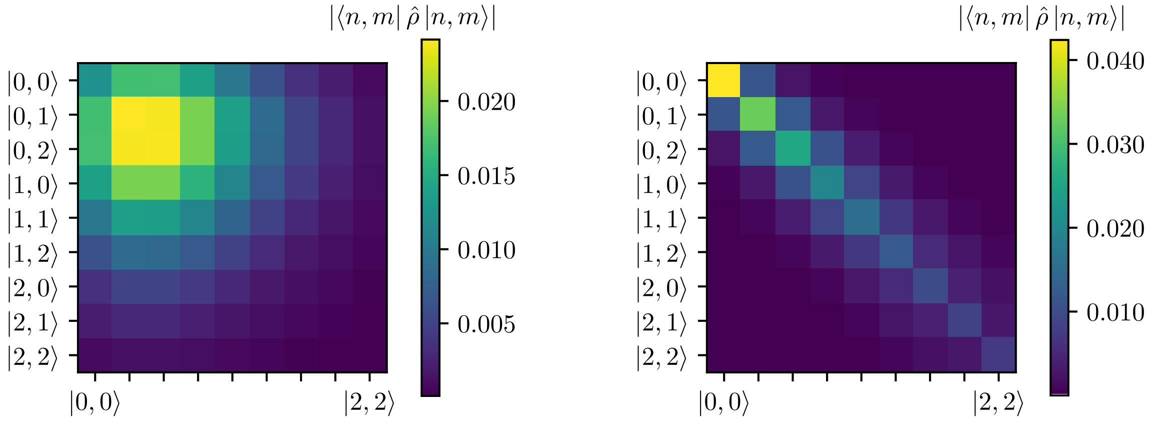
By comparing its performance with that of a classical reservoir on the same task, we will assess whether the quantum reservoir displays any advantages compared to classical models. For the classical reservoir we use the echo state network described in Section 3.2 that applies a nonlinear ReLU function, for which we have already established that 17 neurons are needed to solve the task with $100\,\%$ accuracy. In Fig. 4.8 we show the classification accuracy on the test dataset of the 2-mode bosonic reservoir, with 4, 9, 16 and 25 neurons. These numbers of neurons correspond to the measured joint Fock states

$$\begin{cases} 4 \text{ neurons} & \longrightarrow |0,0\rangle\,, |0,1\rangle\,, |1,0\rangle\,, |1,1\rangle \\ 9 \text{ neurons} & \longrightarrow |0,0\rangle\,, \dots, |2,2\rangle \\ 16 \text{ neurons} & \longrightarrow |0,0\rangle\,, \dots, |3,3\rangle \\ 25 \text{ neurons} & \longrightarrow |0,0\rangle\,, \dots, |4,4\rangle\,, \end{cases} \tag{4.24}$$

for the reservoir output. We saw in Section 4.3.3.1 that an accuracy of $> 99\,\%$ is obtained with the bosonic QRC when measuring only 4 neurons, thus outperforming the classical reservoir.

To better understand where this advantage comes from, we also learned this task using the bosonic QRC in its classical limit, by setting a large dephasing rate. First, we briefly explain the influence of dephasing on the reservoir dynamics, to understand the dynamics of our QRC in its classical limit.

4.3.4 Classical limit of the quantum reservoir: dephasing



(a)

(b)

Figure 4.9: Density matrix of the 2-mode bosonic reservoir, after encoding an input $\mathbf{x} = 1$ into the complex drives for a time $\Delta t = 0.8\kappa^{-1}$. The encoding parameters are $\boldsymbol{\theta}_0 = [\bar\epsilon, \bar\epsilon]$ and $\boldsymbol{\theta}_{\text{bias}} = 0.1\boldsymbol{\theta}_0$, where $\bar\epsilon = 130\sqrt{\kappa/20}$. In (a), there is no dephasing, whereas in (b) the dephasing rate is $\kappa_\varphi = 5\kappa$. The colorbar indicates the absolute value of the density matrix element $|\langle n,m|\,\hat\rho\,|n,m\rangle|$. The Hilbert space is shown up to the joint Fock state $|2,2\rangle$.

Quantum coherences give rise to off-diagonal elements in the density matrix of the system. Indeed for instance, the single-mode pure state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4.25}$$

translates to the density matrix

$$\begin{aligned}\hat\rho &= |\psi\rangle\,\langle\psi| \\ &= |\alpha|^2\,|0\rangle\langle 0| + |\beta|^2\,|1\rangle\langle 1| + \alpha^*\beta\,|0\rangle\langle 1| + \alpha\beta^*\,|1\rangle\langle 0|\,,\end{aligned} \tag{4.26}$$

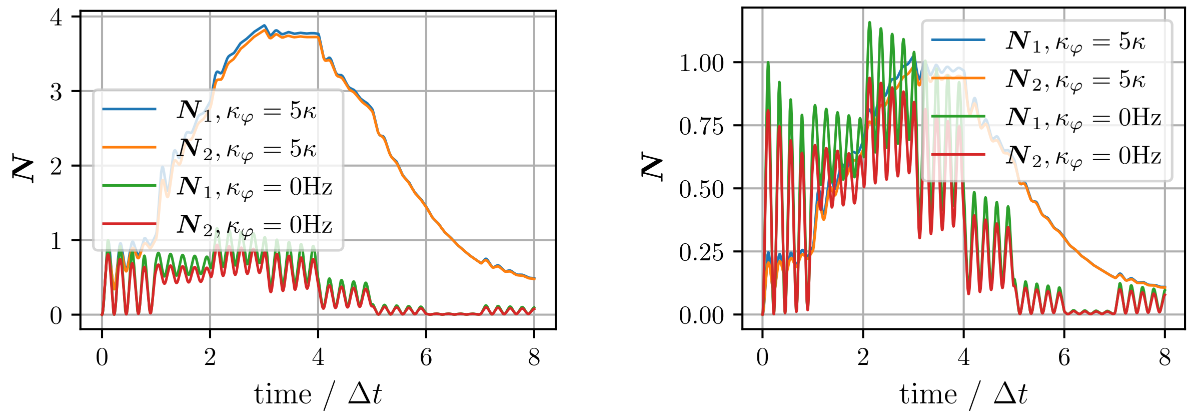
where the last two terms are its off-diagonal components. We observe that the superposition of states is at the origin of quantum coherences. If the density matrix of a state has no off-diagonal components, then the probabilistic outcome resulting from measuring an observable is caused by incomplete knowledge of the state, and not by the quantum superposition.

To take the classical limit and attenuate the quantum coherences, we add a dephasing term in the Lindblad master equation in Eq. (4.15). It is modeled by jump operators

$$\hat{C}_{\varphi,k} = \kappa_\varphi \hat{a}_k{}^\dagger \hat{a}_k, \tag{4.27}$$

where κ_φ is the dephasing rate, set to be equal in the two modes.

We show the effect of this dephasing rate on the quantum coherences in Fig. 4.9, where we plot the density matrix of the 2-mode bosonic reservoir after encoding an input $\mathbf{x} = 1$ into the complex drives for a time $\Delta t = 0.8\kappa^{-1}$, according to Eq. (4.18). We see in Fig. 4.9a that without dephasing, off-diagonal terms $|0,1\rangle\,, |0,2\rangle\,, |1,0\rangle$ have nonzero values. When applying a dephasing $\kappa_\varphi = 5\kappa$, we see in Fig. 4.9b that off-diagonal terms are suppressed, as expected.



(a)

(b)

Figure 4.10: Dynamics comparison of the 2-mode quantum bosonic reservoir (blue and orange lines) and its classical limit (red and green lines) with dephasing rate $\kappa_\varphi = 5\kappa$. A sine input sequence is encoded into the drive amplitude according to $\boldsymbol{\epsilon}(\mathbf{x}^{(\nu)}) = \boldsymbol{\theta}_0 \cdot \mathbf{x}^{(\nu)} + \boldsymbol{\theta}_{\text{bias}}$. The encoding parameters are $\boldsymbol{\theta}_0 = (\bar{\epsilon}, \bar{\epsilon})^T$ and $\boldsymbol{\theta}_{\text{bias}} = 0.1\boldsymbol{\theta}_0$, where (a) $\bar{\epsilon} = 130\sqrt{\kappa/20}$ or (b) $\bar{\epsilon} = \frac{130}{2}\sqrt{\kappa/20}$. The coupling parameters are as defined in Eq. (4.17).

Next, we observe the effect of dephasing on the dynamics of the mean photon number $\boldsymbol{N}$ in Fig. 4.10a, where a sine waveform is encoded into the drives, as described in Section 4.3.2.1. We see that for $\kappa_\varphi = 5\kappa$, the oscillations due to photon conversion g are suppressed, and the number of photons becomes 3 times higher on average.

For the comparison to be fair between the bosonic reservoir and its classical limit, we want the mean photon number N to be equal on average for both of them. So in Fig. 4.10b we show that halving the amplitudes of the drive encoding parameters allows us to recover approximately the same average N as without dephasing. The dynamics of this system are simulated with a Hilbert space cutoff $n_{\mathcal{H}} = 11$, high enough to take into account all of the basis states which have a non-negligible occupation. With these parameters, we now benchmark the performance of the QRC classical limit in Fig. 4.11.

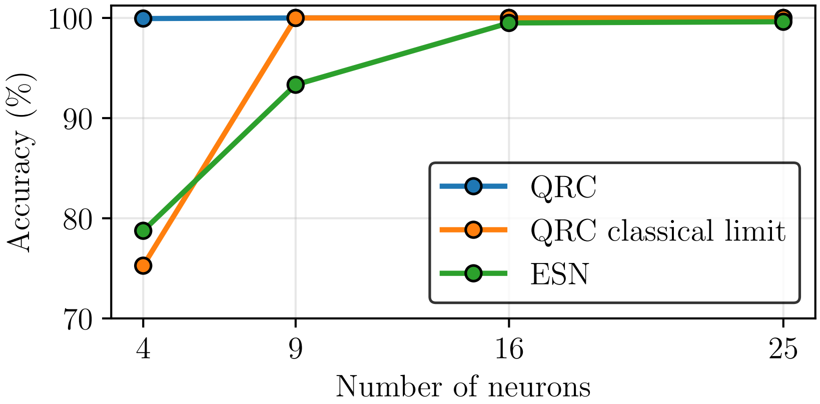


Figure 4.11: Performance comparison of the quantum, quantum in the classical limit, and static classical reservoirs on the sin/square waveform classification task. Test accuracy as a function of the number of reservoir neurons, for a classical reservoir (ESN), and for the bosonic QRC (including the classical limit obtained at dephasing $\kappa_\varphi = 5\kappa$). The static reservoir accuracy is averaged over 100 random initializations. For the bosonic QRC, the number of measured joint Fock states corresponds to the number of reservoir neurons.

Our observation shows that for 4 neurons, the quantum reservoir performs better than in the classical limit, indicating that quantum coherences play a role in learning. Moreover, we observe that the bosonic reservoir in the classical limit slightly outperforms the classical static reservoir by achieving $100\,\%$ accuracy with 9 neurons, while the ESN achieves only $93.33\,\%$ accuracy. This can be attributed to the fact that unmeasured basis states still participate in the transformation of input data. From an experimental perspective, this aspect is interesting because despite the fact that quantum measurements need to be repeated multiple times to reconstruct the Fock state probability amplitudes (see Section 2.2.1.3), a much smaller number of states need to be measured in comparison to the classical case.

4.3.5 Influence of shot noise on performance

In Sections 4.3.3.2 and 4.3.4 the Fock state probabilities $P(\vec{n})$ are estimated exactly in simulations, with no shot noise due to a finite number of measurements. To investigate the experimental feasibility of the QRC, we study the number of measurements

N_{shots} that are needed to obtain sufficiently precise Fock state occupation probabilities in order to perform learning with the same accuracy as with the exact values.

4.3.5.1 Simulating a large number of measurements with the central limit theorem

When simulating shot noise from measuring a Fock state occupation $|\vec{n}\rangle\langle\vec{n}|$, a more efficient method is preferred to what is described in Fig. 2.10, where N_{shots} copies of a state are required to estimate its expectation value. We note that $|\vec{n}\rangle\langle\vec{n}|$ is a projector, so the calculations of Section 2.2.1.3 are applicable.

Using the Central Limit theorem [154] when N_{shots} is large, the estimator of the Fock state probability Z defined in Eq. (2.21) approximates a normal law with its same mean and variance

$$\begin{aligned} &\mathbb{E}(Z) = p \\ &\sigma(Z) = \mathbb{E}(Z^2) - (\mathbb{E}(Z))^2 = \frac{p(1-p)}{N_{\text{shots}}}, \end{aligned} \tag{4.28}$$

where $p \overset{\text{def}}{=} P(\vec{n})$ is the Fock state probability to estimate. With N_{shots} measurements, using the linear stability property of the normal law, we can approximate a value of the probability density

$$Z(x) = p + \sqrt{\frac{p \times (1-p)}{N_{\text{shots}}}} \times N(x), \tag{4.29}$$

with N a probability density following the normal law $\mathcal{N}(0,1)$, of mean 0 and standard deviation 1. I chose this method to simulate shot noise caused by the finite number of measurements.

4.3.5.2 Performance comparison

For three different values of the number of measurement samples N_{shots}, we show the accuracy of the bosonic QRC on the sin/square task in Fig. 4.12. For the drives that we apply in our simulations, lower energy levels in each mode have higher probabilities to be occupied, which means that their measurements induce smaller errors. We find that for the states containing either 0 or 1 photons in each modes, corresponding to the quantum reservoir with 4 neurons in the Figure 4.12, with 10^9 shots we obtain sufficiently precise probability estimations to achieve the same accuracy on the sin/square classification task as with the exact probability amplitude values. For the states containing 2, 3 or 4 photons in either one of both modes, corresponding to quantum reservoirs with 9, 16 and 25 neurons in the Figure 4.12, we need 10^7 shots. Finally, in the case where Fock occupation probabilities are measured with 10^6 shots, we observe that the

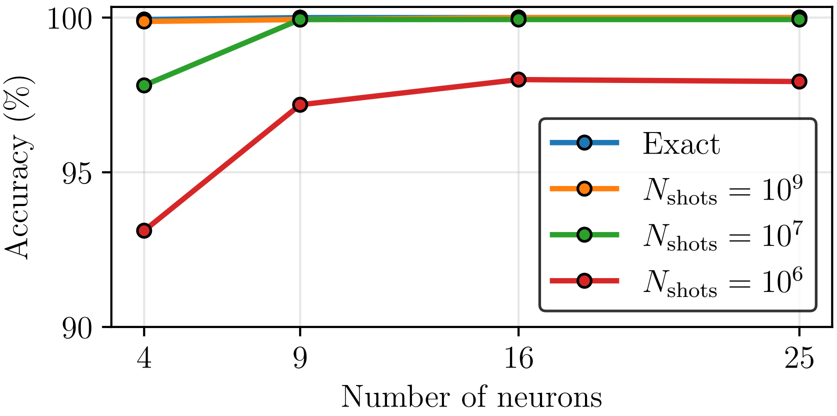


Figure 4.12: Impact of the finite number of measurements on the QRC performance solving the sin/square waveform classification task. Test accuracy is plotted as a function of the number of neurons, with a finite number of measurement samples $N_{\text{shots}} = 10^6$, $N_{\text{shots}} = 10^7$, and $N_{\text{shots}} = 10^9$. Performance with ideal measurements of the Fock states is shown as a reference.

reservoir does not achieve $100\,\%$ accuracy even with 25 neurons. We conclude that to solve the benchmark task with $100\,\%$ accuracy using 4 neurons we need 10^9 shots, and if we increase the number of neurons we can lower the number of shots to 10^7.

To optimize the number of shots used for classification, adaptive measurement schemes can be considered. A possible method would be to use less measurement samples for highly occupied states, than for scarcely occupied ones.

4.3.6 Reservoir performance benchmark on Mackey-Glass time series

The second benchmark that we address is the prediction of Mackey-Glass chaotic time series. Compared to the sin/square task, predicting time series data assesses the memory of the reservoir. This task will serve as a second benchmark of the bosonic QRC performance, which we will compare with the performance of a classical spintronic reservoir composed of 4 skyrmions simulated using a software neural network [155], and a simulation of a quantum reservoir composed of 3 atoms inside an optical cavity [114] as references. To make full use of the reservoir's dynamical complexity, we sample the Fock state occupations at several distinct times for each input data point.

The input data is obtained by solving the first order differential the equation

$$\frac{dx(t)}{dt} = \frac{\beta x(t-\tau_m)}{1 + x^{10}(t-\tau_m)} - \gamma x(t), \tag{4.30}$$

where β, γ and τ_m are dynamical parameters. Chaotic regime is achieved for parameters $\beta = 0.2$, $\gamma = 0.1$ and $\tau_m = 17$ [155]. The dataset is composed of discrete values of the solution to Eq. (4.30), as illustrated in Fig. 4.13a. The task consists in predicting a point with a certain delay in the future. For example, a target dataset associated with the inputs in Fig. 4.13a for a delay $= 20$ is shown in Fig. 4.13b.

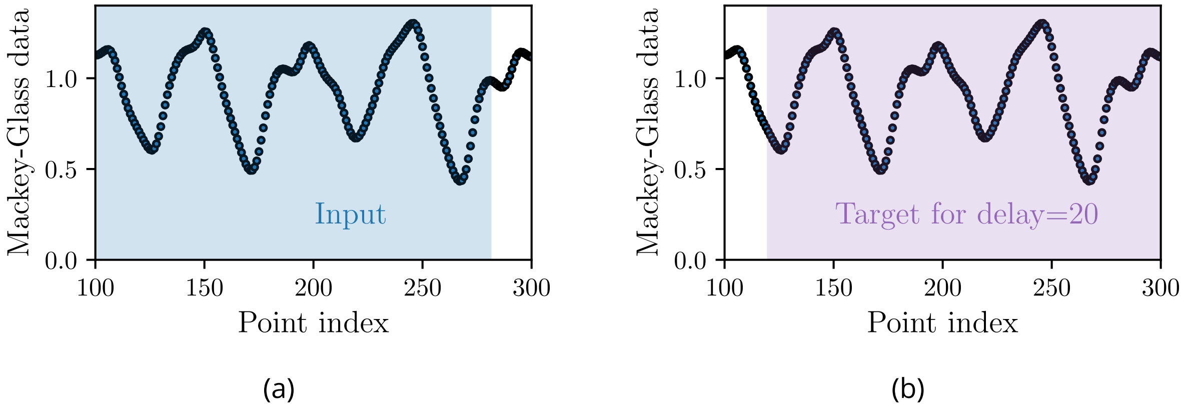


Figure 4.13: (a) Mackey-Glass chaotic time-series data. The data points are taken at the discrete values of time. (b) For each input point, the target is the point with a certain delay in the future, here illustrated for a delay $= 20$.

We train the reservoir for different delays ranging from 1 to 100. Each point is sent for $\Delta t = 1\kappa^{-1}$. In all of the simulations we measure 16 Fock basis states, from $|0,0\rangle$ to $|3,3\rangle$, and we sample the reservoir 3 times for each input, corresponding to a measurement every $\Delta t/3$. With this procedure, we obtain 48 reservoir features. In comparison, the spintronic reservoir measures 50 features, and the three-atom quantum reservoir measures 37 features. The size of both the training and test dataset is of 1000 points, and we use the normalized root mean square error (NRMSE)

$$\text{NRMSE}(\bar{\mathbf{Y}}, \mathbf{Y}) \overset{\text{def}}{=} \frac{\sqrt{\text{MSE}(\bar{\mathbf{Y}}, \mathbf{Y})}}{\max(\mathbf{Y}) - \min(\mathbf{Y})}, \tag{4.31}$$

where $\mathbf{Y}$ is the target dataset, and $\bar{\mathbf{Y}}$ is the reservoir prediction. The results are shown in Fig. 4.14. We plot the logarithmic NRMSE on the test dataset as a function of the delay for the three different reservoir computers, from delay $= 0$ up to delay $= 80$. For the quantum reservoir from Ref. [114], the performances have only been computed for delays with values in $[0, 2, 20, 80]$.

In all of the reservoir performances, we observe a logarithmic increase of the error as a function of the delay. This is expected, as points further in the future are harder to predict because the reservoir memory is lost. For large delays, the error saturates because the reservoir learns the range in which the points are situated. In particular, the error is lower around the regions of the minima and the maxima of the time series, where the data is concentrated. These extrema occur periodically, which is reflected in

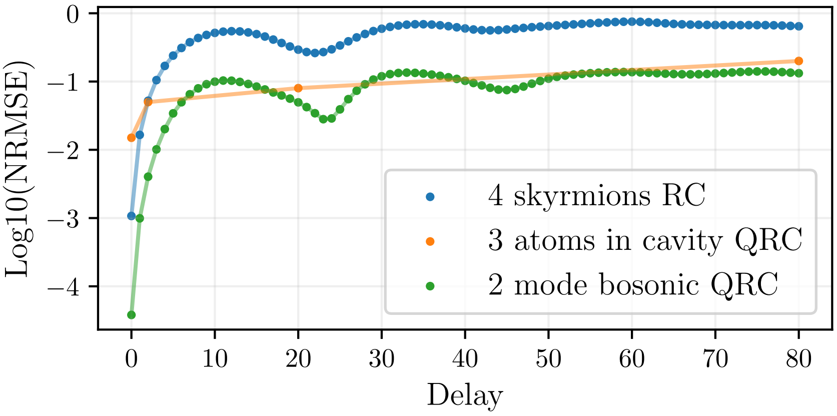


Figure 4.14: Logarithmic normalized root mean square error for the Mackey-Glass task as a function of the number of points in future that the reservoir attempts to predict for (blue dots) a classical reservoir composed of 4 skyrmions with 50 measured features [155], (orange dots) a quantum reservoir composed of 3 atoms inside an optical cavity with 37 measured features [114] and (green dots) the bosonic quantum reservoir with 2 coupled modes and 48 measured features.

the oscillation of the error as a function of delay.

When examining the performances of the three reservoir computers in contrast with each other, we observe that the bosonic QRC performs better than the other two with a comparable number of neurons. This result shows that the bosonic QRC can achieve competitive performance compared to other physical reservoir computers, with a similar number of measured features.

4.4 Discussion

We have shown in this chapter that bosonic modes can be successfully implemented in the quantum reservoir computing framework in order to solve non-trivial tasks such as the sin/square waveform classification and Mackey-Glass time series prediction. Solving such a benchmark task proves the measurement outcome statistics of the bosonic modes in the Fock basis are sufficiently nonlinear to process nonlinear datasets. The chosen encoding scheme in input drive amplitudes, as well as the system dynamics, are linear. Thus, Fock basis measurements were the only source of nonlinearity in our work. We remind however, that other sources of nonlinearities can be considered, as we have discussed in Section 4.2.

The comparison of the bosonic QRC with the echo state network in Section 4.3.3.2 has shown the quantum reservoir can outmatch classical reservoirs, when using the same number of neurons. To understand the origin of this advantage, we have simulated the reservoir in its classical limit by adding a strong dephasing term in Section 4.3.4. We have found that in this limit, the reservoir performance decreases and becomes more similar to that of the ESN. This result shows that quantum coherences play an important role in reservoir performance. Inspired by Fujii and Nakajima [42], and their concept of hidden neurons, we interpret this is as an increase of the number of accessible states for computation in the Hilbert space. While 4 joint Fock states are measured to solve the task, states not measured also participate in the reservoir dynamics.

Compared to classical reservoir computing, we therefore expect QRC to scale better with the number of devices, which simplifies experimental implementation. Less physical observables will need to monitored, than in classical reservoir computing implementations on physical systems.

However, the stochastic nature of quantum measurements makes them require more measurements per observables, compared to classical systems. As we saw in Section 4.3.5, around 10^7 measurement samples are needed to solve the benchmark task with $100\,\%$ test accuracy, and this doesn't take into account the measurement noise, only the quantum noise. In order to decrease this number, we would need our measured Fock state probabilities $P(\vec{n})$ to be larger, as their relative standard deviation scales as $\sqrt{\frac{1-P(\vec{n})}{N_{\text{shots}}P(\vec{n})}}$. Decreasing the number of measured observables for classification is also an objective to reduce the number of measurement samples required, and to simplify experimental setups. Finally, the fact that in reservoir computing only a single layer of linear weights is learned results in a limited expressivity of the learning model.

To improve on the shortcomings of the quantum bosonic reservoir, in the second part of my thesis I have introduced parametric coupling between the bosonic modes, in order to obtain tunable parameters in the physical system, and I have attempted to train them as parameters in a neural network.

Chapter 5

Training the parametric interactions in an analog bosonic quantum neural network

In order to improve the learning performance of the bosonic system, we choose to move away from the reservoir computing paradigm and train the physical parameters. To increase the number of available physical parameters, we describe in Section 5.1 how three-wave mixing can be leveraged to tune photon conversion and two-mode squeezing rates, and drive detunings for an arbitrary number of bosonic modes.

In the framework we will detail in Section 5.5, we propose to treat the amplitudes, phases, and frequency detunings of the driving and parametric mixing processes as trainable parameters, and use some of them for data encoding. Inspired by chip-in-the-loop training [83], we propose to use a model of the circuit in which gradients are accessible to train the actual physical system through gradient descent.

We could simulate the quantum model in the Schrödinger picture with the Lindblad master equation and use libraries implementing back-propagation into differential equation solvers [74, 75], but the computation time of this method scales exponentially with the number of modes and occupied basis states. Because gradient descent training requires executing many forward passes, we need a more efficient method to simulating the quantum system.

Many methods to efficiently simulate quantum systems exist such as truncated cumulants [55, 76], the positive-P representation [77] or tensor networks [78, 79], but they require approximations. We choose to restrict ourselves to using Gaussian modes, which can be simulated efficiently [80]. While Gaussian modes do not explore all the degrees of freedom in the Hilbert space, all of the conclusions of this work will also be applicable to more complex non-Gaussian states.

To compute Fock state occupation probabilities, we use the Gaussian boson sampling formula for which an efficient algorithm has been devised in Ref. [81]. During the second and third years of my PhD, I developed a Python library enabling economical fully differentiable simulations of dynamics and Fock measurements on Gaussian states with the `PyTorch` package.

5.1 Coupling bosonic modes together

In this section we show how three wave mixing with a pump tone can be applied on two bosonic modes denoted by their respective field operators $\hat{a}, \hat{b}$, to convert bosons and perform two-mode squeezing. These results will then be generalized to an arbitrary number of modes.

The end goal of the pump tones that we will introduce in the next two subsections, is to produce the Hamiltonian

$$\frac{\hat{H}}{\hbar} = \delta_a \hat{a}^\dagger \hat{a} + \delta_b \hat{b}^\dagger \hat{b} + g(\hat{a}\hat{b}^\dagger + \hat{a}^\dagger \hat{b}) + g^s(\hat{a}\hat{b} + \hat{a}^\dagger \hat{b}^\dagger), \tag{5.1}$$

where δ_a, δ_b are the drive detunings from the bosonic mode frequencies, g is the photon conversion rate, and g^s is the two-mode squeezing rate.

5.1.1 Three wave mixing

We introduce a pump tone applied to two bosonic modes denoted by their respective field operators $\hat{a}, \hat{b}$, to convert cavity photons and perform two-mode squeezing. The pump tone is itself modeled by a bosonic mode with the field operator $\hat{p}$, of frequency ω_p. We show in this section that appropriately choosing the frequency ω_p enables tunable coupling between the bosonic modes. This type of interaction can be obtained in circuit quantum electrodynamics (cQED) using parametric couplers based on Josephson junctions [156], or in optics using non-linear crystals [157]. In the laboratory frame, the Hamiltonian of two modes $\hat{a}$ and $\hat{b}$, of resonance frequencies ω_a and ω_b, coupled by a three-wave mixing interaction is

$$\begin{aligned} \frac{\hat{H}}{\hbar} &= \omega_a \hat{a}^\dagger \hat{a} + \omega_b \hat{b}^\dagger \hat{b} + \omega_p \hat{p}^\dagger \hat{p} \\ &+ \chi(\hat{p} + \hat{p}^\dagger)(\hat{a} + \hat{a}^\dagger)(\hat{b} + \hat{b}^\dagger), \end{aligned} \tag{5.2}$$

where χ is the three wave mixing coupling strength. Now we use the operator

$$\hat{\mathcal{R}}(t) = \exp\Big(it \left(\omega_a \hat{a}^\dagger \hat{a} + \omega_b \hat{b}^\dagger \hat{b} + \omega_p \hat{p}^\dagger \hat{p}\right)\Big),$$

to apply the rotating frame transformation

$$(\hat{H})_{\mathcal{R}} \overset{\text{def}}{=} i\hbar\frac{d\hat{\mathcal{R}}}{dt}(t)\hat{\mathcal{R}}^\dagger(t) + \hat{\mathcal{R}}(t)\hat{H}\hat{\mathcal{R}}^\dagger(t), \tag{5.3}$$

where the $\mathcal{R}$ subscript denotes rotating frame operators. To apply this transformation on the Hamiltonian Eq. (5.2), we first verify by recurrence that

$$\forall n \in \mathbb{N}, \begin{cases} (\hat{a}^\dagger\hat{a})^n\hat{a} & = \hat{a}(\hat{a}^\dagger\hat{a} - 1)^n \\ (\hat{a}^\dagger\hat{a})^n\hat{a}^\dagger & = \hat{a}^\dagger(\hat{a}^\dagger\hat{a} + 1)^n. \end{cases} \tag{5.4}$$

Using the definition of an operator exponential $e^{\hat{B}} \overset{\text{def}}{=} \sum_{j=0}^{\infty}\frac{\hat{B}^n}{n!}$, we derive from Eq. (5.4)

$$\begin{cases} \exp\left(it\omega_a\hat{a}^\dagger\hat{a}\right)\hat{a} & = \hat{a}\exp(-it\omega_a)\exp\left(it\omega_a\hat{a}^\dagger\hat{a}\right) \\ \exp\left(it\omega_a\hat{a}^\dagger\hat{a}\right)\hat{a}^\dagger & = \hat{a}^\dagger\exp(+it\omega_a)\exp\left(it\omega_a\hat{a}^\dagger\hat{a}\right). \end{cases} \tag{5.5}$$

From Equation (5.5) we calculate $\hat{a}$ and $\hat{a}^\dagger$ in the rotating frame $\mathcal{R}$

$$\begin{cases} (\hat{a})_{\mathcal{R}} & \overset{\text{def}}{=} \hat{\mathcal{R}}\hat{a}\hat{\mathcal{R}}^\dagger = \hat{a}\exp(-it\omega_a) \\ (\hat{a}^\dagger)_{\mathcal{R}} & \overset{\text{def}}{=} \hat{\mathcal{R}}\hat{a}^\dagger\hat{\mathcal{R}}^\dagger = \hat{a}^\dagger\exp(+it\omega_a). \end{cases} \tag{5.6}$$

As these relations hold for any bosonic field operator, they also apply to $\hat{b}$ and $\hat{p}$. We note that $\hat{a}^\dagger\hat{a}$ is unchanged by the rotating frame. By replacing the expression Eq. (5.6) in the Eq. (5.3) we find

$$\begin{aligned} (\hat{H})_{\mathcal{R}} = {} & i\frac{d\hat{\mathcal{R}}}{dt}(t)\hat{\mathcal{R}}^\dagger(t) + \omega_a\hat{a}^\dagger\hat{a} + \omega_b\hat{b}^\dagger\hat{b} \\ & + \hbar\chi\left(\hat{p}e^{-it\omega_p} + \hat{p}^\dagger e^{it\omega_p}\right)\left(\hat{a}e^{-it\omega_a} + \hat{a}^\dagger e^{it\omega_a}\right)\left(\hat{b}e^{-it\omega_b} + \hat{b}^\dagger e^{it\omega_b}\right). \end{aligned} \tag{5.7}$$

The first line cancels out, and we develop the second one which yields 8 terms

$$\begin{aligned} (\hat{H})_{\mathcal{R}}/\hbar\chi = {} & \hat{p}\hat{a}\hat{b}e^{it(-\omega_a-\omega_b-\omega_p)} + \hat{p}^\dagger\hat{a}\hat{b}e^{it(-\omega_a-\omega_b+\omega_p)} \\ & + \hat{p}\hat{a}\hat{b}^\dagger e^{it(-\omega_a+\omega_b-\omega_p)} + \hat{p}^\dagger\hat{a}\hat{b}^\dagger e^{it(-\omega_a+\omega_b+\omega_p)} \\ & + \hat{p}\hat{a}^\dagger\hat{b}e^{it(\omega_a-\omega_b-\omega_p)} + \hat{p}^\dagger\hat{a}^\dagger\hat{b}e^{it(\omega_a-\omega_b+\omega_p)} \\ & + \hat{p}\hat{a}^\dagger\hat{b}^\dagger e^{it(\omega_a+\omega_b-\omega_p)} + \hat{p}^\dagger\hat{a}^\dagger\hat{b}^\dagger e^{it(\omega_a+\omega_b+\omega_p)}. \end{aligned} \tag{5.8}$$

Finally, by setting $\omega_p = \omega_a - \omega_b$, we use the rotating wave approximation (RWA) to cancel out the fast oscillating terms, and we obtain the photon conversion interaction Hamiltonian

$$(\hat{H})_{\mathcal{R}} = \hbar\chi\left(\hat{p}^\dagger\hat{a}\hat{b}^\dagger + \hat{p}\hat{a}^\dagger\hat{b}\right). \tag{5.9}$$

If we instead set $\omega_p = \omega_a + \omega_b$, we achieve two-mode squeezing, described by the Hamiltonian

$$(\hat{H})_{\mathcal{R}} = \hbar\chi \left(\hat{p}^\dagger \hat{a}\hat{b} + \hat{p}\hat{a}^\dagger\hat{b}^\dagger\right). \tag{5.10}$$

This RWA is only valid if

$$|\chi| \ll \omega_a, \omega_b, |\omega_a - \omega_b|. \tag{5.11}$$

We can apply these two couplings simultaneously with two different pump tones $\hat{p}_{(g)}$ and $\hat{p}_{(g^s)}$, at respective frequencies $\omega_{(g)} = \omega_a - \omega_b$ and $\omega_{(g^s)} = \omega_a + \omega_b$. In the classical limit where these pump tones are coherent states with large displacement values, they can be considered as scalar: $\hat{p}_{(g)} \to p_{(g)} \in \mathbb{C}$ and $\hat{p}_{(g^s)} \to p_{(g^s)} \in \mathbb{C}$. We finally get the Hamiltonian

$$(\hat{H})_{\mathcal{R}}/\hbar = g(\hat{a}\hat{b}^\dagger + \hat{a}^\dagger\hat{b}) + g^s(\hat{a}\hat{b} + \hat{a}^\dagger\hat{b}^\dagger), \tag{5.12}$$

where $g \overset{\text{def}}{=} \chi p_{(g)}$ is the photon conversion rate and $g^s \overset{\text{def}}{=} \chi p_{(g^s)}$ the two-mode squeezing rate.

5.1.2 Introducing drive detuning

Detuning the frequencies of the pump tones adds another controllable parameter to influence and complexify the dynamics. The more complex the dynamics of the quantum system, the more complex functions and time series the learning model will be able to fit. We apply three wave mixing with two pump tones $\hat{p}_{(g)}$ and $\hat{p}_{(g^s)}$. We set the pump tone frequencies to

$$\begin{cases} \omega_{(g)} &= \omega_a - \omega_b + 2\delta_{(g)} \\ \omega_{(g^s)} &= \omega_a - \omega_b + 2\delta_{(g^s)}, \end{cases} \tag{5.13}$$

where $2\delta_{(g)}$ and $2\delta_{(g^s)}$ are respectively the photon conversion pump detuning and the two-mode squeezing pump detuning. In this case, the rotating frame

$$\hat{\mathcal{R}}(t) = \exp\Big(it\left(\omega_a\hat{a}^\dagger\hat{a} + \omega_b\hat{b}^\dagger\hat{b} + \omega_{(g)}\hat{p}_{(g)}{}^\dagger\hat{p}_{(g)} + \omega_{(g^s)}\hat{p}_{(g^s)}{}^\dagger\hat{p}_{(g^s)}\right)\Big),$$

combined with the RWA allows us to model a time-independent Hamiltonian

$$\begin{aligned}(\hat{H})_{\mathcal{R}}/\hbar = &\left(-\delta_{(g)} - \delta_{(g^s)}\right)\hat{a}^\dagger\hat{a} + \left(\delta_{(g)} - \delta_{(g^s)}\right)\hat{b}^\dagger\hat{b} \\ &+ \hbar\chi\left(\hat{p}^\dagger_{(g)}\hat{a}\hat{b}^\dagger + \hat{p}_{(g)}\hat{a}^\dagger\hat{b}\right) + \hbar\chi\left(\hat{p}^\dagger_{(g^s)}\hat{a}\hat{b} + \hat{p}_{(g^s)}\hat{a}^\dagger\hat{b}^\dagger\right),\end{aligned} \tag{5.14}$$

if the RWA requirements are met

$$\left|\delta_{(g)}\right|, \left|\delta_{(g^s)}\right|, \chi \ll \omega_a, \omega_b, |\omega_a - \omega_b| \tag{5.15}$$

The detunings δ_a, δ_b to each bosonic mode can be chosen as free parameters using the relation

$$\begin{cases} \delta_{(g)} & = \left(\delta_a - \delta_b\right)/2 \\ \delta_{(g^s)} & = \left(\delta_a + \delta_b\right)/2. \end{cases} \tag{5.16}$$

Finally, considering the pumps $\hat{p}_{(g)}, \hat{p}_{(g^s)}$ in their classical limit allows us to recover Eq. (5.1).

We note that if the detunings are higher than the dissipation rates, the dynamical regime changes, which we want to avoid [61]. Therefore, for all of the simulations in this chapter, we always set the detuning absolute values to be lower than five times the average dissipation rate.

5.1.3 M coupled bosonic modes

The calculations of Section 5.1.2 can be generalized to M coupled bosonic modes $\hat{a}_k$ verifying the commutation relation

$$\left[\hat{a}_j, \hat{a}_k{}^\dagger\right] = \delta^{jk} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}, \tag{5.17}$$

where δ^{jk} is the Kronecker symbol. Three wave mixing allows us to couple the bosonic modes pairwise, with photon conversion at a rate g_{kl} and two-mode squeezing at a rate g^s_{kl} for modes k and l. In the rotating frame, the Hamiltonian of this system writes

$$\boxed{\frac{\hat{H}_{\text{sys}}}{\hbar} = -\sum_{k=1}^{M} \delta_k \hat{a}_k{}^\dagger \hat{a}_k + \sum_{\substack{k,l=1 \\ k<l}}^{M} \left(g_{kl} \hat{a}_k{}^\dagger \hat{a}_l + g^s_{kl} \hat{a}_k{}^\dagger \hat{a}_l{}^\dagger + \text{h.c}\right)} \tag{5.18}$$

where δ_k is the drive detuning of mode k from its resonance frequency. The notation h.c denotes the hermitian conjugate of an expression.

5.2 Driving the bosonic modes with the input-output formalism in the Heisenberg representation

In order to populate the Fock states of the modes, we drive them with nearly resonant drives in the input-output formalism. We will model the evolutions of the operators $\hat{a}_k(t)$ in the Heisenberg picture when the system interacts with a bath of modes, using the quantum Langevin equation. The demonstrations of this section are inspired from

Ref. [158].

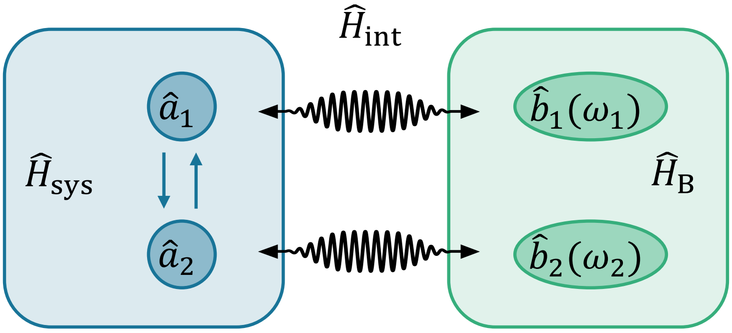


Figure 5.1: Illustration of 2 bosonic modes $\hat{a}_1, \hat{a}_2$ coupled to their respective bath of modes $\hat{b}_1(\omega_1), \hat{b}_2(\omega_2)$. The system and bath are described by their corresponding Hamiltonians $\hat{H}_{\text{sys}}$, $\hat{H}_{\text{B}}$, and their coupling is modeled by the interaction Hamiltonian $\hat{H}_{\text{int}}$.

We suppose a setting illustrated in Fig. 5.1, where each mode in our quantum system is considered to be interacting with a different heat bath k, in the form

$$\begin{aligned}\hat{H} &= \hat{H}_{\text{sys}} + \hat{H}_{\text{B}} + \hat{H}_{\text{int}}\\ \hat{H}_{\text{B}} &= \hbar \sum_{k=1}^{M} \int_{-\infty}^{\infty} d\omega_k \omega_k \hat{b}_k^\dagger(\omega_k) \hat{b}_k(\omega_k)\\ \hat{H}_{\text{int}} &= i\hbar \sum_{k=1}^{M} \int_{-\infty}^{\infty} d\omega_k \gamma_k(\omega_k) \left(\hat{b}_k^\dagger(\omega_k) \hat{c}_k - \hat{c}_k^\dagger \hat{b}_k(\omega_k) \right)\end{aligned} \tag{5.19}$$

where $\hat{H}_{\text{sys}}$ and $\hat{H}_{\text{B}}$ are respectively the system and bath Hamiltonian, and $\hat{b}_k$ are field operators for the bath k of modes. They verify the commutation relation

$$\left[\hat{b}_k(\omega_k), \hat{b}_k^\dagger(\omega_k') \right] = \delta(\omega_k - \omega_k'), \tag{5.20}$$

where $\delta(t)$ is the Dirac delta function [159]. The operator $\hat{c}_k$ acts on system mode k. The bosonic modes are coupled with strengths $\gamma_k(\omega_k)$ to the bath modes through the interaction Hamiltonian $\hat{H}_{\text{int}}$, which is assumed to be linear in $\hat{b}_k(\omega_k)$ and $\hat{b}_k^\dagger(\omega_k)$. This is a general formulation, so the specific Hamiltonian of our bosonic modes $\hat{H}_{\text{sys}}$ will not affect the following calculations.

We move from the Schrödinger picture where quantum states depend on time to the Heisenberg picture, where quantum states are static and the operators depend on time. The Heisenberg picture operators $\hat{O}(t)$ are related to the Schrödinger picture operators $\hat{O}$ through the transformation

$$\hat{O}(t) = \exp\left(\frac{i}{\hbar}\hat{H}t\right) \hat{O} \exp\left(-\frac{i}{\hbar}\hat{H}t\right). \tag{5.21}$$

Evaluating expectation values of the operators in the Heisenberg picture with respect to the initial quantum state $\hat{\rho}(t=0)$ yields the same result as its expectation value in the Schrödinger picture:

$$\left\langle \hat{O}(t) \right\rangle = \mathrm{Tr}\Big(\hat{\rho}(t=0)\hat{O}(t)\Big) = \mathrm{Tr}\Big(\hat{\rho}(t)\hat{O}\Big). \tag{5.22}$$

As the operators now depend on time, they evolve according to the Heisenberg equations of motion

$$\frac{d\hat{O}}{dt} = -\frac{i}{\hbar}\Big[\hat{O}(t), \hat{H}(t)\Big], \tag{5.23}$$

where the Hamiltonian is unchanged by the Heisenberg picture.

We develop the equations of motion in the Heisenberg picture for the bath of field operators $\hat{b}_k(\omega_k)$ and an arbitrary system operator $\hat{O}$ using Eq. (5.23), and find

$$\frac{d\hat{b}_k(\omega_k)}{dt} = -i\omega_k\hat{b}_k(\omega_k) + \gamma_k(\omega_k)\hat{c}_k \tag{5.24}$$

$$\frac{d\hat{O}}{dt} = -\frac{i}{\hbar}\Big[\hat{O}, \hat{H}_{\textsf{sys}}\Big] + \sum_{k=1}^{M}\int_{-\infty}^{\infty} d\omega_k\gamma_k(\omega_k)\left(\hat{b}_k^\dagger(\omega_k)\Big[\hat{O}, \hat{c}_k\Big] - \Big[\hat{O}, \hat{c}_k^\dagger\Big]\hat{b}_k(\omega_k)\right). \tag{5.25}$$

Solving Eq. (5.24) gives

$$\hat{b}_k(\omega_k) = e^{-i\omega_k t}\hat{b}_{k,0}(\omega_k) + \gamma_k(\omega_k)\int_0^t e^{-i\omega_k(t-t')}\hat{c}_k(t')dt', \tag{5.26}$$

where $\hat{b}_{k,0}(\omega_k)$ is the initial value of $\hat{b}_k(\omega_k)$ at $t=0$. Substituting Eq. (5.26) into Eq. (5.25) returns the expression

$$\begin{aligned}\frac{d\hat{O}}{dt} &= -\frac{i}{\hbar}\Big[\hat{O}, \hat{H}_{\textsf{sys}}\Big] \\ &\quad + \sum_{k=1}^{M}\int_{-\infty}^{\infty} d\omega_k\gamma_k(\omega_k)\left(e^{i\omega_k t}\hat{b}_{k,0}^\dagger(\omega_k)\Big[\hat{O}, \hat{c}_k\Big] - \Big[\hat{O}, \hat{c}_k^\dagger\Big]e^{-i\omega_k t}\hat{b}_{k,0}(\omega_k)\right) \\ &\quad + \sum_{k=1}^{M}\int_{-\infty}^{\infty} d\omega_k(\gamma_k(\omega_k))^2\int_0^t dt'\left(e^{i\omega_k(t-t')}\hat{c}_k^\dagger(t')\Big[\hat{O}, \hat{c}_k\Big] - \Big[\hat{O}, \hat{c}_k^\dagger\Big]e^{-i\omega_k(t-t')}\hat{c}_k(t')\right).\end{aligned} \tag{5.27}$$

For visual clarity, we omitted the time argument of operators depending on t, but kept it explicitly for those depending on t'. So far up to Eq. (5.27), the equations are exact. To simplify the integrals in the third line, we introduce the first Markov approximation, which states that the coupling constants are independent of frequency

$$\gamma_k(\omega_k) = \sqrt{\kappa_k/2\pi}, \tag{5.28}$$

where κ_k is the coupling rate of the mode k to its input. So the integrals in ω_k will give Dirac deltas:

$$\int_{-\infty}^{\infty} d\omega e^{-i\omega(t-t')} = 2\pi\delta(t-t'), \tag{5.29}$$

verifying

$$\int_{0}^{t} \hat{c}_k(t')\delta(t-t')dt' = \frac{1}{2}\hat{c}_k(t) \tag{5.30}$$

when Eq. (5.29) is obtained as the limit of an integral over a function going smoothly to zero at $\pm\infty$, which happens in our case. We also define the input field operators as

$$\hat{a}_{k,\text{in}}(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\omega_k e^{-i\omega_k t}\hat{b}_{k,0}(\omega_k), \tag{5.31}$$

which satisfy the commutation relation

$$\left[\hat{a}_{j,\text{in}}(t), {\hat{a}_{k,\text{in}}}^{\dagger}(t')\right] = \delta^{jk}\delta(t-t'). \tag{5.32}$$

Using Eqs. (5.28) to (5.32), we derive the quantum Langevin equation governing the time evolution of the operators

$$\begin{aligned} \frac{d\hat{O}}{dt} &= -\frac{i}{\hbar}\left[\hat{O}, \hat{H}_{\text{sys}}\right] \\ &- \sum_{k=1}^{M} \left[\hat{O}, \hat{c}_k^{\dagger}\right] \left(\frac{\kappa_k}{2}\hat{c}_k + \sqrt{\kappa_k}\hat{a}_{k,\text{in}}(t)\right) \\ &- \sum_{k=1}^{M} \left(\frac{\kappa_k}{2}\hat{c}_k^{\dagger} + \sqrt{\kappa_k}{\hat{a}_{k,\text{in}}}^{\dagger}(t)\right) \left[\hat{O}, \hat{c}_k\right]. \end{aligned} \tag{5.33}$$

We want to model the dynamics of the field operators in the presence of single-photon dissipation. Single-photon dissipation is modeled by setting $\hat{c}_j \to \hat{a}_j$. By applying the quantum Langevin equation Eq. (5.33) to the field operators $\hat{a}_j$ that describe the single photon loss, we find

$$\boxed{\frac{d\hat{a}_k}{dt} = -\frac{i}{\hbar}\left[\hat{a}_k, \hat{H}_{\text{sys}}\right] - \frac{\kappa_k}{2}\hat{a}_k - \sqrt{\kappa_k}\hat{a}_{k,\text{in}}.} \tag{5.34}$$

We note that the $-\sqrt{\kappa_k}\hat{a}_{k,\text{in}}$ term models nearly resonant driving, and the $-\frac{\kappa_k}{2}\hat{a}_k$ term models single-photon loss. For an isolated system, only the first term of Eq. (5.34) would remain.

In this work, the input modes $\hat{a}_{k,\text{in}}$ will always be coherent states of amplitude $\langle\hat{a}_{k,\text{in}}\rangle \overset{\text{def}}{=} \epsilon_k$. We note that this driving formulation is equivalent to the cascaded formalism introduced in Section 4.3.1, with complex drive amplitudes ϵ_k.

5.3 Modeling the quantum state with its Gaussian moments

The coupling Hamiltonian Eq. (5.18) is quadratic in the field operators, and we only model single photon loss, signifying the state of this system will always be Gaussian, if it starts in a Gaussian state [160]. In this section, we will first introduce the tools to define what a Gaussian state is, and how they can be efficiently simulated in polynomial time.

5.3.1 Gaussian states

A Gaussian state is defined by its Gaussian distribution in the phase space. To express the two moments of this distribution in systems of M bosonic modes, we first introduce the quadrature operators

$$\hat{x}_k \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}(\hat{a}_k + \hat{a}_k{}^\dagger), \qquad \hat{p}_k \stackrel{\text{def}}{=} \frac{1}{i\sqrt{2}}(\hat{a}_k - \hat{a}_k{}^\dagger). \tag{5.35}$$

Let $\hat{R} \stackrel{\text{def}}{=} (\hat{x}_1, \ldots, \hat{x}_M, \hat{p}_1, \ldots, \hat{p}_M)^T$ be the column vector containing the quadratures of the modes, we can define the first and second moments of the quadratures as

$$\boldsymbol{\alpha}^R \stackrel{\text{def}}{=} \left\langle \hat{R} \right\rangle \tag{5.36a}$$

$$\boldsymbol{\sigma}^R_{kl} \stackrel{\text{def}}{=} \frac{1}{2} \left\langle \hat{R}_k \hat{R}_l^\dagger + \hat{R}_l^\dagger \hat{R}_k \right\rangle - \boldsymbol{\alpha}^R_k (\boldsymbol{\alpha}^R_l)^*. \tag{5.36b}$$

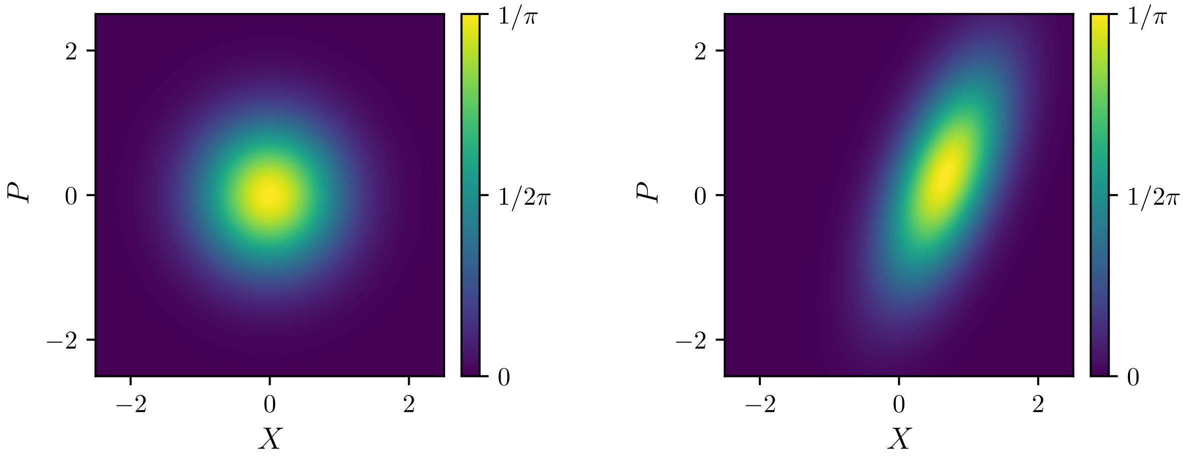
We label $\boldsymbol{\alpha}^R \in \mathbb{C}^{2M\times 1}$ the quadrature displacement vector and $\boldsymbol{\sigma}^R \in \mathbb{C}^{2M\times 2M}$ the quadrature covariance matrix.

The quantum quasi-probability distribution of a M mode mixed state $\hat{\rho}$ in the phase space can be represented using the Wigner function [161]

$$W(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{(2\pi)^M} \int_{\mathbb{R}^M} \left\langle \boldsymbol{x} + \frac{1}{2}\boldsymbol{q} \middle| \hat{\rho} \middle| \boldsymbol{x} - \frac{1}{2}\boldsymbol{q} \right\rangle e^{i\boldsymbol{p}\cdot\boldsymbol{q}} d^M \boldsymbol{q}, \tag{5.37}$$

where $\boldsymbol{x}$ and $\boldsymbol{p}$ denote M-dimensional real vectors. We denote $|x_k\rangle$ an eigenstate of the position quadrature operator $\hat{x}_k$ with the eigenvalue x_k, and $|\boldsymbol{x}\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_M\rangle$.

The Wigner function can be reformulated as $W(\boldsymbol{r})$, for $\boldsymbol{r} = (x_1, \ldots, x_M, p_1, \ldots, p_M)$, such that $W(\boldsymbol{r}) = W(\boldsymbol{x}, \boldsymbol{p})$. For Gaussian states, the Wigner function is a Gaussian



(a) (b)

Figure 5.2: Wigner function distribution of two different single mode Gaussian states in the phase space. (a) Vacuum state. (b) Gaussian state with $\boldsymbol{\alpha}^R = (0.66, 0.26)^T$ and $\boldsymbol{\sigma}^R = \begin{pmatrix} 0.35 & 0.34 \\ 0.34 & 1.03 \end{pmatrix}$.

distribution defined by the quadrature displacement vector and covariance matrix

$$W(\boldsymbol{r}) = \frac{1}{(2\pi)^M \sqrt{\det(\boldsymbol{\sigma}^R)}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\alpha}^R - \boldsymbol{r}\right)^T \left(\boldsymbol{\sigma}^R\right)^{-1} \left(\boldsymbol{\alpha}^R - \boldsymbol{r}\right)\right). \tag{5.38}$$

A notable Gaussian state is the vacuum $|\mathbf{0}\rangle \overset{\text{def}}{=} |0\rangle_1 \otimes ... \otimes |0\rangle_M$, with moments

$$\begin{cases} \boldsymbol{\alpha}^R & = (0, \ldots, 0) \\ \boldsymbol{\sigma}^R & = \boldsymbol{\sigma}_0 \overset{\text{def}}{=} \begin{pmatrix} \frac{\mathbb{1}_M}{2} & \mathbb{0}_M \\ \mathbb{0}_M & \frac{\mathbb{1}_M}{2} \end{pmatrix} . \end{cases} \tag{5.39}$$

$\mathbb{1}_M$ and $\mathbb{0}_M$ denote the unitary and zero matrices of size $M \times M$. The Wigner function of a single mode in a vacuum state is shown in Fig. 5.2a. The Wigner function distribution of a single mode Gaussian state with $\boldsymbol{\alpha}^R = (0.66, 0.26)^T$ and $\boldsymbol{\sigma}^R = \begin{pmatrix} 0.35 & 0.34 \\ 0.34 & 1.03 \end{pmatrix}$ is depicted in the phase space in Fig. 5.2b, to illustrate its Gaussian shape.

For the remainder of this manuscript, it will be more convenient to represent Gaussian states using their field operator displacement and covariance matrix, instead of those of the field quadratures. Let $\hat{A} \overset{\text{def}}{=} \left(\hat{a}_1, \ldots, \hat{a}_M, \hat{a}_1{}^\dagger, \ldots, \hat{a}_M{}^\dagger\right)^T$ be the column vector containing the field operators of M bosonic modes. The displacement vector and covariance matrix are

$$\boldsymbol{\alpha} \overset{\text{def}}{=} \left\langle \hat{A} \right\rangle \tag{5.40a}$$

$$\boldsymbol{\sigma}_{kl} \overset{\text{def}}{=} \frac{1}{2} \left\langle \hat{A}_k \hat{A}_l^{\dagger} + \hat{A}_l^{\dagger} \hat{A}_k \right\rangle - \boldsymbol{\alpha}_k \boldsymbol{\alpha}_l^{*}. \tag{5.40b}$$

The field operator moments can be derived from the quadrature moments

$$\boldsymbol{\alpha} = \gamma \boldsymbol{\alpha}^R \tag{5.41a}$$

$$\boldsymbol{\sigma} = \gamma \boldsymbol{\sigma}^R \gamma^{\dagger}, \tag{5.41b}$$

where γ is the change of frame matrix

$$\gamma \overset{\text{def}}{=} \frac{1}{\sqrt{2}} \left(\begin{array}{c|c} \mathbb{1}_M & i\mathbb{1}_M \\ \hline \mathbb{1}_M & -i\mathbb{1}_M \end{array} \right), \quad \gamma^{-1} = \gamma^{\dagger} = \frac{1}{\sqrt{2}} \left(\begin{array}{c|c} \mathbb{1}_M & \mathbb{1}_M \\ \hline -i\mathbb{1}_M & i\mathbb{1}_M \end{array} \right). \tag{5.42}$$

Moving to the Heisenberg picture allows to define the field operator moments with respect to time

$$\boldsymbol{\alpha}(t) \overset{\text{def}}{=} \left\langle \hat{A}(t) \right\rangle \tag{5.43a}$$

$$\boldsymbol{\sigma}_{kl}(t, t') \overset{\text{def}}{=} \frac{1}{2} \left\langle \hat{A}_k(t) \hat{A}_l^{\dagger}(t') + \hat{A}_l^{\dagger}(t') \hat{A}_k(t) \right\rangle - \boldsymbol{\alpha}_k(t) \boldsymbol{\alpha}_l^{*}(t'). \tag{5.43b}$$

For visual clarity, we will simplify the notation of $\boldsymbol{\sigma}(t, t')$ to $\boldsymbol{\sigma}(t)$ if $t = t'$.

5.3.2 Evolution of the Gaussian moments

Since Gaussian states are fully described by their polynomially-sized displacement and covariance matrix, it is more computationally efficient to compute their dynamics using the Gaussian variables in the Heisenberg picture, rather than simulating the exponentially-sized density matrix in the Schrödinger picture. In this section we will use the quantum Langevin equation Eq. (5.34) to derive the time evolution equations for the displacement and covariance matrix of a Gaussian mode.

We start by developing the unitary evolution term induced by the coupling Hamiltonian $\hat{H}_{\text{sys}}$ using the commutation property $[a, bc] = [a, b]c + b[a, c]$ and Eq. (5.17), which gives

$$\begin{aligned}
-\frac{i}{\hbar}\left[\hat{a}_k, \hat{H}_{\text{sys}}\right] &= i\sum_{j=1}^{M}\delta_j\left[\hat{a}_k, {\hat{a}_j}^\dagger\hat{a}_j\right] \\
&\quad - i\sum_{\substack{i,j=1\\ i<j}}^{M}\left(g_{ij}\left[\hat{a}_k, {\hat{a}_i}^\dagger\hat{a}_j\right] + {g_{ij}}^*\left[\hat{a}_k, \hat{a}_i{\hat{a}_j}^\dagger\right] + g^s_{ij}\left[\hat{a}_k, {\hat{a}_i}^\dagger{\hat{a}_j}^\dagger\right] + {g^s_{ij}}^*[\hat{a}_k, \hat{a}_i\hat{a}_j]\right) \\
&= i\delta_k\hat{a}_k - i\sum_{j=1}^{M}\left(\gamma_k(g_{kj} + {g_{jk}}^*)\hat{a}_j + (g^s_{kj} + g^s_{jk}){\hat{a}_j}^\dagger\right).
\end{aligned} \tag{5.44}$$

For the quadratic system Hamiltonian $\hat{H}_{\text{sys}}$ of Eq. (5.18), the Langevin equation becomes

$$\frac{d\hat{a}_k}{dt} = i\delta_k\hat{a}_k - i\sum_{j=1}^{M}\left((g_{kj} + {g_{jk}}^*)\hat{a}_j + (g^s_{kj} + g^s_{jk}){\hat{a}_j}^\dagger\right) - \frac{\kappa_k}{2}\hat{a}_k - \sqrt{\kappa_k}\hat{a}_{k,\text{in}}. \tag{5.45}$$

Since $\hat{a}_k$ and $\hat{a}_{k,\text{in}}$ only evolve linearly with this equation, we notice that the system of differential equations defined by Eq. (5.45) is closed, and thus they are exactly solvable. We describe the evolution of a Gaussian state using the coupling matrix [80]

$$L \overset{\text{def}}{=} \frac{1}{i\hbar}\begin{pmatrix} G & G^s \\ -G^{s\dagger} & -G^T \end{pmatrix}, \tag{5.46}$$

where the matrix elements are

$$(G)_{k,l} \overset{\text{def}}{=} \hbar \times \begin{cases} -\delta_k & \text{if } k = l \\ g_{kl} & \text{if } k < l \\ {g_{kl}}^* & \text{if } k > l \end{cases}, \quad (G^s)_{k,l} \overset{\text{def}}{=} \hbar \times \begin{cases} 0 & \text{if } k = l \\ g^s_{kl} & \text{otherwise.} \end{cases} \tag{5.47}$$

The vectorized Langevin equation for the entire system becomes

$$\frac{d\hat{A}}{dt} = L\hat{A} - \frac{K}{2}\hat{A} - \sqrt{K}\hat{A}_{\text{in}}, \tag{5.48}$$

where

$$\begin{cases} K & \overset{\text{def}}{=} \text{diag}(\kappa_1, \ldots, \kappa_M, \kappa_1, \ldots, \kappa_M) \\ \hat{A}_{\text{in}} & \overset{\text{def}}{=} \left(\hat{a}_{1,\text{in}}, \ldots, \hat{a}_{M,\text{in}}, {\hat{a}_{1,\text{in}}}^\dagger, \ldots, {\hat{a}_{M,\text{in}}}^\dagger\right)^T. \end{cases} \tag{5.49}$$

The general solution to this first-order linear differential equation is

$$\hat{A}(t) = F(t)\hat{A}(t=0) - \int_0^t F(t-\tau)\sqrt{K}\hat{A}_{\mathsf{in}}(\tau)d\tau, \tag{5.50}$$

where we have introduced the propagator matrix

$$F(t) \overset{\text{def}}{=} \exp(F't), \tag{5.51}$$

with

$$F' \overset{\text{def}}{=} L - \frac{K}{2}. \tag{5.52}$$

Time evolution of the displacement

Using Eq. (5.48) and the definition of the field operator displacement $\boldsymbol{\alpha}(t)$ from Eq. (5.43a), we obtain

$$\boxed{\boldsymbol{\alpha}(t) = F(t)\boldsymbol{\alpha}(t=0) - \int_0^t F(t-\tau)\sqrt{K}\boldsymbol{\alpha}_{\mathsf{in}}(\tau)d\tau,} \tag{5.53}$$

where $\boldsymbol{\alpha}_{\mathsf{in}}(t) \overset{\text{def}}{=} \left\langle \hat{A}_{\mathsf{in}}(t) \right\rangle = (\epsilon_1(t), \dots, \epsilon_M(t), {\epsilon_1}^*(t), \dots, {\epsilon_M}^*(t))^T$ is the displacement vector of the input modes.

Time evolution of the covariance matrix

The calculation of $\boldsymbol{\sigma}(t)$ is lengthier. First we develop the different terms of Eq. (5.43b)

$$\begin{aligned}
\boldsymbol{\alpha}_i(t)\boldsymbol{\alpha}_j(t) = &\sum_{k,l=1}^{2M} F_{ik}(t){F_{jl}}^*(t)\boldsymbol{\alpha}_k(t=0)\boldsymbol{\alpha}_l(t=0)\\
&+\sum_{k,l=1}^{2M}\int_0^t\int_0^t F_{ik}(t-\tau){F_{jl}}^*(t-\tau')\sqrt{K_kK_l}\boldsymbol{\alpha}_{\mathsf{in},k}(\tau){\boldsymbol{\alpha}_{\mathsf{in},l}}^*(\tau')d\tau d\tau'\\
&-\sum_{k,l=1}^{2M} F_{ik}(t)\boldsymbol{\alpha}_k(t=0)\int_0^t {F_{jl}}^*(t-\tau)\sqrt{K_l}{\boldsymbol{\alpha}_{\mathsf{in},l}}^*(\tau)d\tau\\
&-\sum_{k,l=1}^{2M} {F_{jl}}^*(t){\boldsymbol{\alpha}_l}^*(t=0)\int_0^t F_{ik}(t-\tau)\sqrt{K_k}\boldsymbol{\alpha}_{\mathsf{in},k}(\tau)d\tau
\end{aligned} \tag{5.54}$$

$$
\begin{aligned}
\left\langle \hat{A}_i(t)\hat{A}_j(t)\right\rangle = &\sum_{k,l=1}^{2M} F_{ik}(t)F_{jl}{}^*(t)\left\langle \hat{A}_k(t=0)\hat{A}_l{}^\dagger(t=0)\right\rangle \\
&+\sum_{k,l=1}^{2M}\int_0^t\int_0^t F_{ik}(t-\tau)F_{jl}{}^*(t-\tau')\sqrt{K_kK_l}\left\langle \hat{A}_{\mathsf{in},k}(\tau)\hat{A}_{\mathsf{in},l}{}^\dagger(\tau')\right\rangle d\tau d\tau' \\
&-\sum_{k,l=1}^{2M} F_{ik}(t)\boldsymbol{\alpha}_k(t=0)\int_0^t F_{jl}{}^*(t-\tau)\sqrt{K_l}\boldsymbol{\alpha}_{\mathsf{in},l}{}^*(\tau)d\tau \\
&-\sum_{k,l=1}^{2M} F_{jl}{}^*(t)\boldsymbol{\alpha}_l{}^*(t=0)\int_0^t F_{ik}(t-\tau)\sqrt{K_k}\boldsymbol{\alpha}_{\mathsf{in},k}(\tau)d\tau.
\end{aligned}
\tag{5.55}
$$

We observe that the last two terms in Eq. (5.54) and Eq. (5.55) will cancel out. So the expression for the covariance matrix is

$$
\boldsymbol{\sigma}_{ij}(t)=\sum_{k,l=1}^{2M} F_{ik}(t)F_{jl}{}^*(t)\boldsymbol{\sigma}_{kl}(t=0)+\sum_{k,l=1}^{2M}\sqrt{K_kK_l}\int_0^t\int_0^t F_{ik}(t-\tau)F_{jl}{}^*(t-\tau')\boldsymbol{\sigma}_{\mathsf{in},kl}(\tau,\tau')d\tau d\tau',
\tag{5.56}
$$

where $\boldsymbol{\sigma}_{\mathsf{in}}$ is the covariance matrix of the input modes. In the setting where input modes are in coherent states without temporal correlations, their covariance matrix verifies

$$
\boldsymbol{\sigma}_{\mathsf{in}}(t,t')=\delta(t-t')\boldsymbol{\sigma}_0.
\tag{5.57}
$$

In this regime, the Eq. (5.56) simplifies to

$$
\begin{aligned}
\boldsymbol{\sigma}_{ij}(t) &= \sum_{k,l=1}^{2M} F_{ik}(t)F_{jl}{}^*(t)\boldsymbol{\sigma}_{kl}(t=0)+\sum_{k,l=1}^{2M}\sqrt{K_kK_l}\int_0^t F_{ik}(t-\tau)F_{jl}{}^*(t-\tau)\boldsymbol{\sigma}_{0,kl}d\tau \\
&= \sum_{k,l=1}^{2M} F_{ik}(t)F_{jl}{}^*(t)\boldsymbol{\sigma}_{kl}(t=0)+\frac{1}{2}\sum_{k=1}^{2M}K_k\int_0^t F_{ik}(t-\tau)F_{jk}{}^*(t-\tau)d\tau.
\end{aligned}
\tag{5.58}
$$

Finally, we get the expression

$$
\boxed{\boldsymbol{\sigma}(t)=F(t)\boldsymbol{\sigma}(t=0)F^\dagger(t)+\boldsymbol{\sigma}_0\int_0^t F(t-\tau)KF^\dagger(t-\tau)d\tau.}
\tag{5.59}
$$

5.3.3 Computation of the displacement and covariance matrix via diagonalization

To simulate the bosonic modes when they are in Gaussian states, we can compute the displacement $\boldsymbol{\alpha}(t)$ and covariance matrix $\boldsymbol{\sigma}(t)$. Their computation is possible by integrating Eqs. (5.53) and (5.59), but this method is time-consuming. Instead, diagonalizing the propagator matrix F allows for their efficient computation, as we will detail below.

The input modes are set to be in coherent states of constant values $\boldsymbol{\alpha}_{\text{in}}$, and we assume the propagator matrix generator F' is diagonalizable as

$$\Lambda = U^{-1}F'U = \text{diag}\left(\lambda_1, \ldots, \lambda_{2M}\right), \tag{5.60}$$

where U is an invertible matrix, and λ_k are eigenvalues of F'. The propagator matrix can be diagonalized using the exponential of a diagonal matrix

$$\begin{aligned} F(t) &= \exp(F't) \\ &= Ue^{t\Lambda}U^{-1}. \end{aligned} \tag{5.61}$$

The integral of $\boldsymbol{\alpha}(t)$ in Eq. (5.53) becomes

$$\begin{aligned} \boldsymbol{\alpha}(t) &= F(t)\boldsymbol{\alpha}(t=0) + \sqrt{K}U\left(\int_0^t e^{\Lambda(t-\tau)}d\tau\right)U^{-1}\boldsymbol{\alpha}_{\text{in}} \\ &= F(t)\boldsymbol{\alpha}(t=0) + \sqrt{K}UI_1U^{-1}\boldsymbol{\alpha}_{\text{in}}, \end{aligned} \tag{5.62}$$

where $I_1 = \Lambda^{-1}\left(e^{\Lambda t} - \mathbb{1}_{2M}\right)$. To compute the covariance matrix, we introduce the matrices P and I_2 such that

$$P = U^{-1}K(U^{-1})^\dagger \tag{5.63}$$

$$(I_2)_{i,j} = (P)_{i,j}\frac{\exp((\lambda_i + \lambda_j{}^*)\,t) - 1}{\lambda_i + \lambda_j{}^*}. \tag{5.64}$$

Finally, we find

$$\boldsymbol{\sigma}(t) = F(t)\boldsymbol{\sigma}(t=0)F(t)^\dagger + \boldsymbol{\sigma}_0UI_2U^\dagger. \tag{5.65}$$

Simulating the evolution of Gaussian states using Eqs. (5.62) and (5.65) in the Heisenberg picture is much more efficient than solving the Lindblad master equation of their density matrices in the Schrödinger representation, because their size scales exponentially with the number of modes. In the calculations considered above, the most resource intensive operation is matrix diagonalization, which is of time complexity $\mathcal{O}(M^3)$ using the QR algorithm [162]. In my simulations, this was my method for computing the displacement and covariance matrix.

5.4 Effects of coupling parameters

Before training the physical parameters for learning, it is important to understand their effects on the system dynamics. Moreover, if the mean number of photons in the mode k

$$\boldsymbol{N}_k \overset{\text{def}}{=} \mathrm{Tr}\big(\hat{\rho}\hat{a}_k{}^\dagger\hat{a}_k\big) = \Big\langle \hat{A}_{k+M}\hat{A}_k \Big\rangle \tag{5.66}$$

becomes very high, the system will exit the quantum regime because of thermal processes, which we want to avoid. In this section we study the dynamical behaviors in two modes induced by constant drives applied to each mode with detunings δ_1, δ_2, with modes coherently coupled at a rate g_{12} and two-mode squeezing rate g^s_{12}. To this end, we will use the formulas Eqs. (5.62) and (5.65) derived in Section 5.3.3.

To simplify the calculations of this section, we set equal dissipation rates and drive amplitudes

$$\forall k, \begin{cases} \kappa_k &= \kappa \\ \epsilon_k &= \epsilon, \end{cases} \tag{5.67}$$

and the system always starts in vacuum. The mean photon number $\boldsymbol{N} = \{\boldsymbol{N}_1, \boldsymbol{N}_2\}$ will be computed from the Gaussian moments using the relation

$$\boldsymbol{N}_k = |\boldsymbol{\alpha}_k|^2 + \boldsymbol{\sigma}_{kk} - \frac{1}{2}, \tag{5.68}$$

which can be recovered using the definition of the covariance matrix. In particular, if the covariance matrix is that of vacuum, then the diagonalization of F' in Section 5.3.3 yields

$$\begin{aligned} \boldsymbol{N}_k(t) &= |\boldsymbol{\alpha}_k(t)|^2 \\ &= \kappa \left| \epsilon \sum_{l,m}^{2M} U_{k,m} U_{m,l} \frac{e^{\lambda_m t} - 1}{\lambda_m} \right|^2. \end{aligned} \tag{5.69}$$

Else, the expression is harder to compute

$$\boldsymbol{N}_k(t) = \kappa|\epsilon|^2 \sum_{l',m',l,m}^{2M} U_{k+M,m'} U^{-1}_{m',l'} U_{k,m} U^{-1}_{m,l} \frac{1 - e^{\lambda_m t} - e^{\lambda_{m'} t} + e^{(\lambda_m + \lambda_{m'})t}}{\lambda_m \lambda_{m'}}. \tag{5.70}$$

5.4.1 Effect of the drive detuning on the mode dynamics

In this section, we look into the effect of the drive detuning on a single mode. The system Hamiltonian is

$$\hat{H}_{\mathsf{sys}} = \delta\hat{a}^\dagger\hat{a}, \tag{5.71}$$

and the coupling matrix is

$$L = -i \begin{pmatrix} -\delta & 0 \\ 0 & \delta \end{pmatrix}. \tag{5.72}$$

Because $L^\dagger = -L$ and $K = \kappa \mathbb{1}_2$, we have $F(t)F(t)^\dagger = e^{-\kappa t}$. From the expressions of $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\sigma}(t)$ in Eqs. (5.59) and (5.62), we deduce

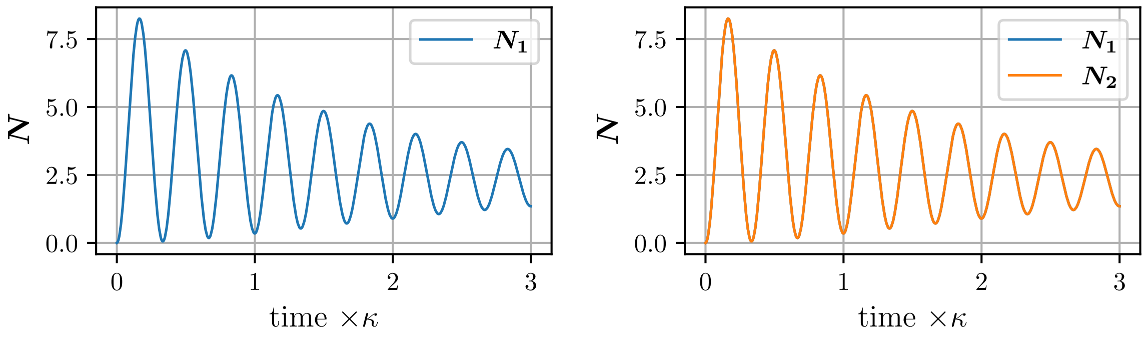
$$\boldsymbol{\alpha}_1(t) = \sqrt{\kappa}\epsilon \frac{e^{i\delta - \frac{\kappa}{2}} - 1}{i\delta - \frac{\kappa}{2}} \tag{5.73}$$

$$\begin{aligned} \boldsymbol{\sigma}(t) &= \boldsymbol{\sigma}_0 \left(e^{-\kappa t} + \kappa \int_0^t e^{-\kappa(t-\tau)} d\tau \right) \\ &= \boldsymbol{\sigma}_0. \end{aligned} \tag{5.74}$$

Calculating the expression of the mean photon number

$$\boldsymbol{N}_1 = \kappa |\epsilon|^2 \left(\frac{1 + e^{-\kappa t} - 2\cos(\delta t) e^{-\frac{\kappa}{2} t}}{\left(\frac{\kappa}{2}\right)^2 + \delta^2} \right), \tag{5.75}$$

reveals that the detuning induces oscillations at frequency $\delta/2\pi$, on a timescale κ^{-1} after which the system reaches an asymptotic photon number of $\kappa|\epsilon|^2 \left(\left(\frac{\kappa}{2}\right)^2 + \delta^2\right)^{-1}$, as shown in Fig. 5.3a.



(a)

(b)

Figure 5.3: Mean photon number $\boldsymbol{N}$ as a function of time. (a) Single mode driven at amplitude $\epsilon = 100\sqrt{\kappa/2}$, applied with detuning $\delta = 2\pi \times 3\kappa^{-1}$. (b) Two coherently coupled modes at a rate $g_{12} = 2\pi \times 3\kappa^{-1}$, driven at amplitude $\epsilon = 100\sqrt{\kappa/2}$ with no detuning. We observe $\boldsymbol{N}_1(t) = \boldsymbol{N}_2(t)$. In both figures, the drive amplitude is $\epsilon = 100\sqrt{\kappa/4\pi}$, i.e both modes have the same number of photons at all times.

5.4.2 Coherent photon conversion

In general for M-coupled modes, if the dissipation rates are equal (i.e $K = \kappa\mathbb{1}_{2M}$) and there is no two-mode squeezing (meaning $L^\dagger = -L$), then

$$\begin{aligned}
&L^\dagger = -L\\
\Rightarrow &F^\dagger(t) = \exp\Big(\Big(-L - \frac{\kappa}{2}\mathbb{1}_{2M}\Big)\,t\Big)\\
\Rightarrow &F^\dagger(t) = F^{-1}(t)\exp(-\kappa t),
\end{aligned} \tag{5.76}$$

and the covariance matrix remains that of vacuum:

$$\begin{aligned}
\boldsymbol{\sigma}(t) &= F(t)\boldsymbol{\sigma}_0F^\dagger(t) + \boldsymbol{\sigma}_0\kappa\int_0^t F(t-\tau)F^\dagger(t-\tau)d\tau\\
&= \boldsymbol{\sigma}_0F(t)F^{-1}(t)e^{-\kappa t} + \boldsymbol{\sigma}_0\kappa\int_0^t F(t-\tau)F^{-1}(t-\tau)e^{-\kappa(t-\tau)}d\tau\\
&= \boldsymbol{\sigma}_0\left(e^{-\kappa t} + \kappa\int_0^t e^{-\kappa(t-\tau)}d\tau\right)\\
&= \boldsymbol{\sigma}_0.
\end{aligned} \tag{5.77}$$

In the case of two driven modes coupled only through coherent photon conversion g_{12}, the coupling Hamiltonian reads

$$\hat{H}_{\text{sys}} = g_{12}\hat{a}_1{}^\dagger\hat{a}_2 + g_{12}{}^*\hat{a}_1\hat{a}_2{}^\dagger. \tag{5.78}$$

As stated previously the covariance matrix remains that of the vacuum, and calculating the mean photon number returns an expression analogous to the single-mode detuning case of Eq. (5.75)

$$\boldsymbol{N}_k = \kappa|\epsilon|^2\left(\frac{1 + e^{-\kappa t} - 2\cos(g_{12}t)e^{-\frac{\kappa}{2}t}}{\left(\frac{\kappa}{2}\right)^2 + (g_{12})^2}\right). \tag{5.79}$$

We observe from this expression that photon conversion also creates oscillations even in the absence of drive detuning, at a frequency $g_{12}/2\pi$, as shown in Fig. 5.3.

We note that since the covariance matrix remains diagonal, there is no correlation between the modes if only photon conversion is turned on. To avoid this, unequal dissipation rates are required, or two-mode squeezing must be turned on.

5.4.3 Two-mode squeezing

If only two-mode squeezing is enabled, the coupling Hamiltonian becomes

$$\hat{H}_{\mathsf{sys}} = g^s_{12}\hat{a}_1{}^\dagger \hat{a}_2{}^\dagger + g^s_{12}{}^* \hat{a}_1 \hat{a}_2. \tag{5.80}$$

For easier calculations, we set $g^s_{12} \in \mathbb{R}^+$. The coupling matrix then reads

$$L = \begin{pmatrix} 0 & 0 & 0 & -ig^s_{12} \\ 0 & 0 & -ig^s_{12} & 0 \\ 0 & ig^s_{12} & 0 & 0 \\ ig^s_{12} & 0 & 0 & 0 \end{pmatrix}, \tag{5.81}$$

which is hermitian. Because the system starts in the vacuum and $K = \kappa \mathbb{1}_{2M}$, the integral expression of $\boldsymbol{\sigma}(t)$ can be simplified to

$$\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}_0 e^{(2L-\kappa)t} + \kappa \int_0^t e^{(2L-\kappa)(t-\tau)} d\tau. \tag{5.82}$$

The diagonalization of F' yields

$$\Lambda = \text{diag}\left(g^s_{12}, g^s_{12}, -g^s_{12}, -g^s_{12}\right) - \frac{\kappa}{2} \tag{5.83}$$

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} i & 0 & i & 0 \\ 0 & -i & 0 & i \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{pmatrix} \tag{5.84}$$

$$U^{-1} = U^\dagger. \tag{5.85}$$

We define two scalar values s_1, s_2

$$\begin{cases} s_1 &= -\frac{\kappa}{2g^s_{12}-\kappa} + e^{\left(2g^s_{12}-\kappa\right)t}\left(1 + \frac{\kappa}{2g^s_{12}-\kappa}\right) \\ s_2 &= -\frac{\kappa}{-2g^s_{12}-\kappa} + e^{\left(-2g^s_{12}-\kappa\right)t}\left(1 + \frac{\kappa}{-2g^s_{12}-\kappa}\right), \end{cases} \tag{5.86}$$

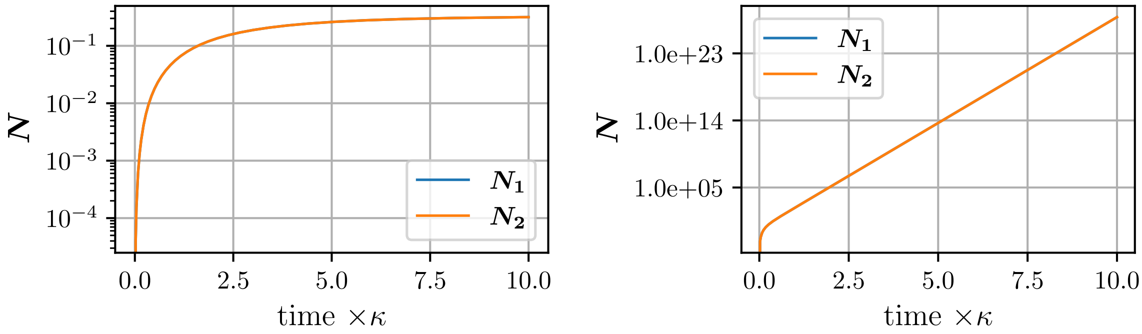
to develop the integrals in Eq. (5.82):

$$\begin{aligned} \boldsymbol{\sigma}(t) &= \boldsymbol{\sigma}_0 U \text{diag}(s_1, s_1, s_2, s_2) U^{-1} \\ &= \boldsymbol{\sigma}_0 \begin{pmatrix} \frac{s_1+s_2}{2} & 0 & 0 & i\frac{-s_1+s_2}{2} \\ 0 & \frac{s_1+s_2}{2} & i\frac{-s_1+s_2}{2} & 0 \\ 0 & i\frac{s_1-s_2}{2} & \frac{s_1+s_2}{2} & 0 \\ i\frac{s_1-s_2}{2} & 0 & 0 & \frac{s_1+s_2}{2} \end{pmatrix}. \end{aligned} \tag{5.87}$$

We set the drive amplitudes to be zero, so the displacement vector $\boldsymbol{\alpha}(t)$ remains zero. Therefore the mean photon number in the mode k is

$$\boldsymbol{N}_k = \frac{s_1 + s_2 - 1}{2}. \tag{5.88}$$

In the expressions of s_1 and s_2 in Eq. (5.86), we observe that if $2g^s_{12} > \kappa$, $\boldsymbol{N}$ diverges asymptotically due to infinitely increasing squeezing, whereas it converges otherwise. This behavior is illustrated in Fig. 5.4.



(a)

(b)

Figure 5.4: Mean photon number $\boldsymbol{N}$ in two modes as a function of time, with (a) $g^s_{12} = 0.05\kappa$ and (b) $g^s_{12} = 0.6\kappa$. We observe that it converges in the first case and diverges in the second.

5.4.4 Effect of simultaneous photon conversion and two-mode squeezing

If the two-mode squeezing and photon conversion pump tones are not in phase with the input drive signal, their coupling rates g^s_{12} and g_{12} take complex values. In this case the propagator matrix generator is

$$F' = \begin{pmatrix} 0 & -ig_{12} & 0 & -ig^s_{12} \\ -ig_{12}{}^* & 0 & -ig^s_{12} & 0 \\ 0 & ig^s_{12}{}^* & 0 & ig_{12}{}^* \\ ig^s_{12}{}^* & 0 & ig_{12} & 0 \end{pmatrix} - \frac{\kappa}{2}\mathbb{1}_{2M}. \tag{5.89}$$

Its diagonalization $\Lambda = U^{-1}F'U$ yields

$$\Lambda = \text{diag}\left(\lambda_+, \lambda_+, \lambda_-, \lambda_-\right) \tag{5.90}$$

$$U = \begin{pmatrix} i(|g^s_{12}|^2 - |g_{12}|^2) & -g_{12} & i(|g^s_{12}|^2 - |g_{12}|^2) & -g_{12} \\ {g_{12}}^*\lambda_+ & -i\lambda_+ & {g_{12}}^*\lambda_- & -i{\lambda_-}^* \\ 0 & {g^s_{12}}^* & 0 & g^s_{12} \\ -{g^s_{12}}^*\lambda_+ & 0 & -{g^s_{12}}^*\lambda_- & 0 \end{pmatrix}, \tag{5.91}$$

where we define

$$\lambda_\pm = \pm\lambda' - \frac{\kappa}{2}, \tag{5.92}$$

with

$$\lambda' = \begin{cases} -i\sqrt{|g_{12}| - |g^s_{12}|} & \text{if } |g_{12}| > |g^s_{12}| \\ \sqrt{|g^s_{12}| - |g_{12}|} & \text{if } |g^s_{12}| > |g_{12}|. \end{cases} \tag{5.93}$$

If $g_{12} = g^s_{12}$, F' is not diagonalizable. By symmetry, the mean photon numbers in modes 1 and 2 are equal: $\boldsymbol{N}_1 = \boldsymbol{N}_2 = \boldsymbol{N}$. The full expression of Eq. (5.70) is too heavy to compute as there are $4^4 = 256$ terms, but depending on the ratio of the photon conversion and two-mode squeezing rates, we can infer different behaviors of $\boldsymbol{N}(t)$:

$$\begin{cases} \text{if } |g^s_{12}| > |g_{12}| \text{ and } 2\sqrt{|g^s_{12}|^2 - |g_{12}|^2} > \kappa & \Rightarrow \left\{ \boldsymbol{N}(t) \text{ diverges when } t \to \infty \right. \\ \text{if } |g^s_{12}| > |g_{12}| \text{ and } 2\sqrt{|g^s_{12}|^2 - |g_{12}|^2} < \kappa & \Rightarrow \left\{ \begin{array}{l} \boldsymbol{N}(t) \text{ doesn't oscillate,} \\ \text{and converges when } t \to \infty \end{array} \right. \\ \text{if } |g^s_{12}| < |g_{12}| & \Rightarrow \left\{ \begin{array}{l} \boldsymbol{N}(t) \text{ oscillates at frequency} \\ \frac{2}{2\pi}\sqrt{|g_{12}|^2 - |g^s_{12}|^2}\text{, and} \\ \text{converges when } t \to \infty. \end{array} \right. \end{cases} \tag{5.94}$$

These three different behaviors are shown in Fig. 5.5 in the blue, orange, and green lines respectively.

This observation of divergences is important because we want the mean number of photons to be kept below 100, in order to stay in the quantum regime. So when implementing the learning models, we will always set the photon conversion rates to be higher than the two-mode squeezing rates.

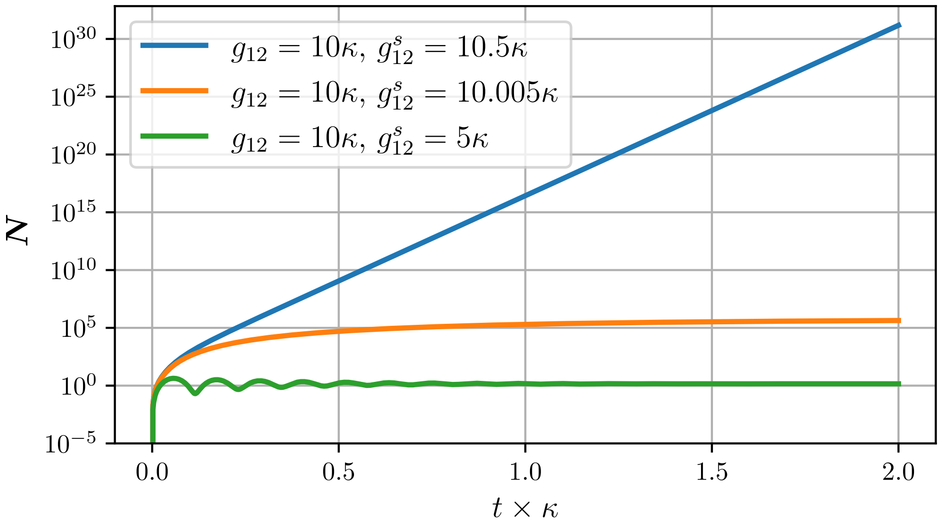


Figure 5.5: Mean photon number $\boldsymbol{N}_1 = \boldsymbol{N}_2 = \boldsymbol{N}$ for 2 modes as a function of time for $|g^s_{12}| < |g_{12}|$ (green), $|g^s_{12}| > |g_{12}|$ and $2\sqrt{|g^s_{12}|^2 - |g_{12}|^2} < \kappa$ (orange), $|g^s_{12}| > |g_{12}|$ and $2\sqrt{|g^s_{12}|^2 - |g_{12}|^2} > \kappa$ (blue). The photon conversion rate is fixed at $g_{12} = 10\kappa$, and the drive amplitudes are set to $\epsilon = 200\sqrt{\kappa/4\pi}$.

5.5 Analog bosonic quantum neural network

5.5.1 Description of the model

We consider a M-mode system, whose evolution of field operator moments $\{\boldsymbol{\alpha}(t), \boldsymbol{\sigma}(t)\}$ is described in Section 5.3.2. All physical parameters can be represented as vectors: $\boldsymbol{\epsilon}$ stores the nearly resonant drive amplitudes, $\boldsymbol{\delta}$ the detunings, $\boldsymbol{g}$ the photon conversion rates, $\boldsymbol{g^s}$ the two-mode squeezing rates and $\boldsymbol{\kappa}$ the single-photon dissipation rates.

As shown in Fig. 5.6, we train two layers of weights, on top of the physical parameters. The first layer is composed of the encoding parameters $\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{bias}}$ into the complex drive amplitudes $\boldsymbol{\epsilon}$, or another physical parameter of our choice. The encoding relation for an $d_{\mathbf{x}}$-dimensional input $\mathbf{x}$ is

$$\boldsymbol{\theta}_{\text{enc}}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot \mathbf{x} + \boldsymbol{\theta}_{\text{bias}}, \tag{5.95}$$

where $\boldsymbol{\theta}_{\text{enc}}$ is the d_{enc}-dimensional parameter to encode $\mathbf{x}$ into, $\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{bias}} \in \mathbb{C}^{d_{\text{enc}}}$, and $\odot$ is the element-wise vector product. If $d_{\mathbf{x}} \neq d_{\text{enc}}$, then we augment the dimension of $\mathbf{x}$ by repeating its values.

Due to challenging experimental implementation of the joint Fock state measurements that we proposed in our QRC work, for the trained network that requires measurements for each training epoch, we consider the local Fock state amplitudes as out-

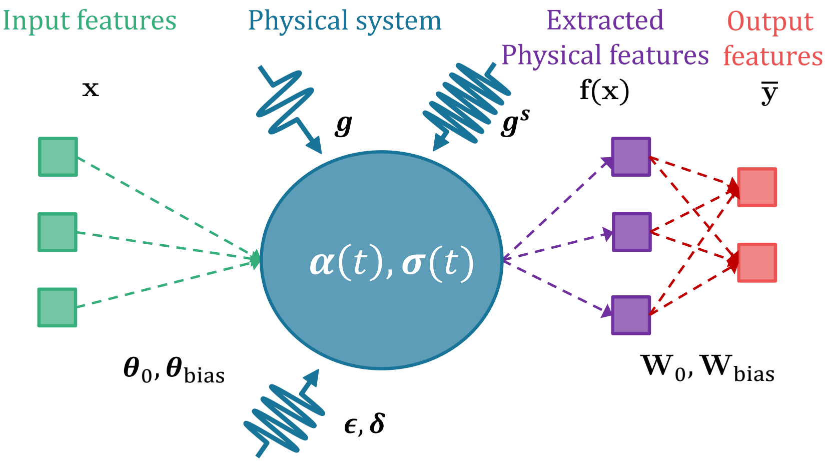

Input features
x
Physical system
g
g^s
Extracted
Physical features
f(x)
Output
features
$\bar{\mathbf{y}}$
$\boldsymbol{\alpha}(t), \boldsymbol{\sigma}(t)$
$\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{bias}}$
$\mathbf{W}_0, \mathbf{W}_{\text{bias}}$
$\boldsymbol{\epsilon}, \boldsymbol{\delta}$

Figure 5.6: Schematic of an analog quantum neural network. The input data vector $\mathbf{x}$ (green squares) is encoded into a parameter $\boldsymbol{\theta}_{\text{enc}}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot \mathbf{x} + \boldsymbol{\theta}_{\text{bias}}$, and the feature vector $\mathbf{f}(\mathbf{x})$ (purple squares) is obtained by measuring a set of local probabilities $\{P_k(n)\}_{k,n}$ of a mode k to contain n photons. The prediction $\bar{\mathbf{y}}$ (red squares) is obtained by multiplying the feature vector by a trained weight matrix $\mathbf{W}_0$, with bias $\mathbf{W}_{\text{bias}}$.

put features. Their probabilities are given by the Gaussian boson sampling formula

$$P_k(n|\boldsymbol{\alpha},\boldsymbol{\sigma}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{\alpha}_k^\dagger \boldsymbol{\sigma}_{k,Q}^{-1}\boldsymbol{\alpha}_k\right)}{\sqrt{\det(\boldsymbol{\sigma}_{k,Q})}\prod_{k=1}^{M} n_k!}\text{lhaf}(\boldsymbol{A}_n), \tag{5.96}$$

where

$$\begin{cases} \boldsymbol{\sigma}_{k,Q} & = \boldsymbol{\sigma}_k + \mathbb{1}_2/2 \\ \boldsymbol{T} & = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \boldsymbol{A} & = \boldsymbol{T}\left(\mathbb{1}_2 - \boldsymbol{\sigma}_{k,Q}^{-1}\right), \end{cases} \tag{5.97}$$

and $\boldsymbol{\alpha}_k \in \mathbb{C}^2$ and $\boldsymbol{\sigma}_k \in \mathbb{C}^{2\times 2}$ are the displacement vector and covariance matrix of the mode k, obtained by keeping only the kth and $k+M$th rows and columns in the Gaussian variables [163]. $\boldsymbol{A}_n$ is formed from $\boldsymbol{A}$ by substituting its diagonal with $\boldsymbol{\alpha}_k^\dagger \boldsymbol{\sigma}_{k,Q}^{-1}$ then repeating the 1st and 2nd rows and columns n times, and lhaf$(.)$ is the loop hafnian function [164]. The measured local Fock states constitute the feature vector $\mathbf{f}(\mathbf{x})$. The final layer is composed of the output weights $\mathbf{W}_0$, with bias $\mathbf{W}_{\text{bias}}$

$$\bar{\mathbf{y}} = \mathbf{f}(\mathbf{x}) \times \mathbf{W}_0 + \mathbf{W}_{\text{bias}}. \tag{5.98}$$

Analytical gradients are obtained through automatic differentiation and the parameters are optimized with the Adam Optimization algorithm [111] in `PyTorch`, based on a code adapted from [81] which performs efficient computations of the loop hafnian function.

One drawback of using Gaussian states is that we cannot simulate non-linear (i.e non-quadratic) terms in the system Hamiltonian, as they create non-Gaussian states. This restricts the number of pump tones we would want to introduce in order to add more trainable parametric couplings: the only allowed ones will be coherent photon conversion and two-mode squeezing. For the remainder of the manuscript, we denote $\boldsymbol{\theta}$ the set of trained parameters in the M-mode system. The physical parameters we choose to train are

- ϵ_k the complex drive amplitude applied to mode k
- δ_k the frequency detuning of mode k
- g_{kl} the photon conversion rate between mode k and l
- g^s_{kl} the two-mode squeezing rate between mode k and l.

Counting the number of trainable parameters thus yields

$$\begin{cases} \boldsymbol{\epsilon} \in \mathbb{C}^M & \to 2M \\ \boldsymbol{\delta} \in \mathbb{R}^M & \to M \\ \boldsymbol{g} \in \mathbb{C}^{\frac{M(M-1)}{2}} & \to M(M-1) \\ \boldsymbol{g^s} \in \mathbb{C}^{\frac{M(M-1)}{2}} & \to M(M-1), \end{cases} \tag{5.99}$$

adding up to $2M(M-1)+3M$. We observe that this number is quadratic in the number of modes M. With such a low number of trainable parameters, we were unsure before starting this work whether training would result in significant performance improvements over quantum reservoir computing with the bosonic modes.

In this chapter, we set the scaling parameter to $\kappa = 2\pi \times 2\,\mathrm{MHz}$ because in experimental superconducting resonators, dissipation rates can be lowered to around this value [165].

Renormalization of physical parameters. In optimization algorithms, it is preferable for all learned parameters to be of the same order of magnitude. To achieve this, we apply a rescaling of physical parameters using a renormalization factor $R \overset{\text{def}}{=} 10\,\mathrm{MHz}$

in all simulations:

$$\begin{cases} t(s) \rightarrow & t \times R \\ \boldsymbol{\delta}(\mathrm{Hz}) \rightarrow & \boldsymbol{\delta}/R \\ \boldsymbol{g}(\mathrm{Hz}) \rightarrow & \boldsymbol{g}/R \\ \boldsymbol{g}^s(\mathrm{Hz}) \rightarrow & \boldsymbol{g}^s/R \\ \boldsymbol{\kappa}(\mathrm{Hz}) \rightarrow & \boldsymbol{\kappa}/R \\ \boldsymbol{\epsilon}(\sqrt{\mathrm{Hz}}) \rightarrow & \boldsymbol{\epsilon}/\sqrt{R}. \end{cases} \tag{5.100}$$

5.5.2 Choice of an analog model

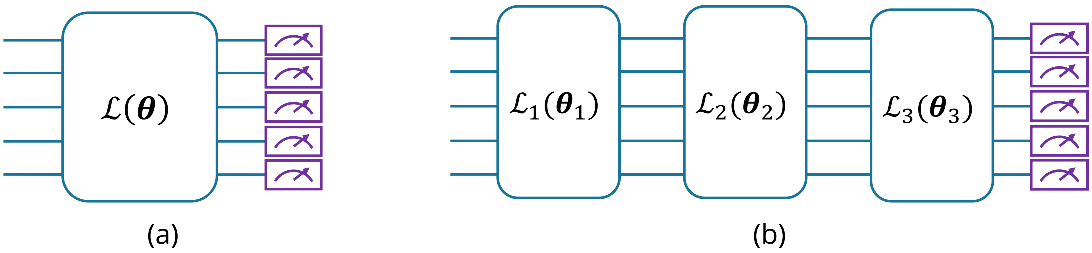

$\mathcal{L}(\boldsymbol{\theta})$
(a)
$\mathcal{L}_1(\boldsymbol{\theta}_1)$
$\mathcal{L}_2(\boldsymbol{\theta}_2)$
$\mathcal{L}_3(\boldsymbol{\theta}_3)$
(b)

Figure 5.7: Two different circuit approaches to parametrically control bosonic modes. Each blue line corresponds to a singular mode. (a) Analog computing, in which there is a single set of parameters $\boldsymbol{\theta}$ parameterizing the Lindbladian evolution described by the superoperator $\mathcal{L}(\boldsymbol{\theta})$. (b) Gate-based computing with three gates in this example, where sequential Lindbladian evolutions described by superoperators $\mathcal{L}_k$ each have their own set of parameters $\boldsymbol{\theta}_k$. The total set of parameters is then $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k\in[1,3]}$.

The goal in this chapter is to establish whether there is an advantage in learning with bosonic modes compared to QRC, even with a quadratic number of trained parameters. The model we described in Section 5.5.1 is analog, meaning we do not use multiple gate-based operations with high predictability and precision, as are considered in parametric quantum circuits. Instead of applying sequential Lindbladian evolutions $\mathcal{L}_k$ each with a different set of parameters $\boldsymbol{\theta}_k$, we apply a single one and harness the complex behaviors in the dynamics for which we do not have an analytical mathematical expression. Indeed, F' cannot be analytically diagonalized in the general case for $M \geq 3$ modes, as its characteristic polynomial is of degree $2M$, and there is no universal solution to finding the roots of 5th and higher degree polynomials. The two different circuit approaches to parametrically learning bosonic modes are illustrated in Fig. 5.7. Our choice of analog computation stems from its relative simplicity: to highlight advantages of learning in bosonic modes, we do not require the use of multiple parametric gate operations.

5.5.3 Possible issues with the back-propagation

The calculation of the field operator displacement $\boldsymbol{\alpha}(t)$ and covariance matrix $\boldsymbol{\sigma}(t)$ in Eqs. (5.62) and (5.65) relies on the matrix U containing the eigenvectors of the propagator generator matrix F' defined in Eq. (5.61). However, as stated in the `PyTorch` documentation [166], gradients involving eigenvectors of a matrix A are only well defined if A has distinct eigenvalues. Gradients become unstable when eigenvalues are nearly degenerate.

A simple example with $M = 2$ modes, identical dissipation rates $\kappa_1 = \kappa_2 = \kappa$ and zero detunings $\delta_1 = \delta_2 = 0\,\mathrm{Hz}$ illustrates this issue. The eigenvalues (λ_-, λ_+) of F' in this case are each two-fold degenerate:

$$\begin{cases} \lambda_\pm = \pm i\sqrt{|g|^2 - |g^s|^2} - \frac{\kappa}{2} & \text{if } |g| > |g^s| \\ \lambda_\pm = \pm\sqrt{|g^s|^2 - |g|^2} - \frac{\kappa}{2} & \text{if } |g^s| > |g| \end{cases}. \tag{5.101}$$

In this scenario, gradient computation will fail due to the degeneracy. To avoid such issues, the initial set of physical parameters $\{\boldsymbol{\epsilon}, \boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{g^s}, \boldsymbol{\kappa}\}$ is chosen such that F' has non-degenerate eigenvalues. The procedure we have arbitrarily chosen, is to set the initial dissipation, photon conversion, and two-mode squeezing rates to be linearly increasing with their index k, i.e for the parameter vector $\boldsymbol{\theta} \in \{\boldsymbol{\kappa}, \boldsymbol{g}, \boldsymbol{g^s}\}$

$$\theta_k = \frac{k-1}{L-1} 0.2 \times \theta_{\text{avg}} + 0.9 \times \theta_{\text{avg}}, \tag{5.102}$$

where L is the length $\boldsymbol{\theta}$, and θ_{avg} is the average value of $\boldsymbol{\theta}$ over all its components θ_k. This linear relation is chosen so that $\theta_1 = 0.9\theta_{\text{avg}}$ and $\theta_L = 1.1\theta_{\text{avg}}$.

5.6 Sin/square waveform classification

We first benchmark the bosonic quantum neural network on the sin/square waveforms classification task, in order to compare its performance to that of the bosonic QRC, which was benchmarked on this task in Section 4.3.3.2. To provide a comparison to QRC, we use two coupled modes, and encode the renormalized input data $\mathbf{x}^{(\nu)} \in [0, 1]$ into the complex drive amplitudes $\boldsymbol{\epsilon}$ according to Eq. (5.95). The input data points are sent one by one, each for a duration $\Delta t = \frac{2\pi}{5\kappa}$.

The loss function applied for this task is the mean-square error

$$\text{MSE}(\bar{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2N_{\text{samples}}} \sum_{k=1}^{N_{\text{samples}}} (\bar{\mathbf{Y}}_k - \mathbf{Y}_k)^2, \tag{2.10 revisited}$$

where N_{samples} is the number of data points used for training, $\bar{\mathbf{Y}} = \mathbf{W}_0 \times \mathbf{F}(\mathbf{X}) + \mathbf{W}_{\text{bias}}$ is the matrix whose rows are the network prediction, and $\mathbf{Y}$ is the target dataset.

Parameter	Initial value	Learning rate
$\mathbf{W}_0$	1	0.01
$\mathbf{W}_{\text{bias}}$	0	0.01
κ	$2\pi \times 2\,\mathrm{MHz}$	none
$\boldsymbol{\kappa}$	$(1 \pm 0.1)\kappa$	none
$\boldsymbol{\delta}$	$0\,\mathrm{Hz}$	0.1
$\boldsymbol{\epsilon}$	$(170 \pm 30)\sqrt{\kappa/4\pi}$	0.1
$\boldsymbol{g}$	45κ	0.1
$\boldsymbol{g^s}$	9κ	0.1
Δt	$0.4\pi\kappa^{-1}$	none

Hyper-parameter	Value
number of modes	2
number of training epochs	500
regularization loss strength $\beta_{\boldsymbol{N}}$	0.02
$\boldsymbol{N}_{\text{thr}}$	3
$\boldsymbol{N}_{\text{tg}}$	2
number of batches	5
dataset size	200
loss function	MSE

Table 5.1: Initial parameter values, their learning rates, and hyper-parameter values for the sin/square waveform classification task.

Regularization loss To prevent the training from pushing the parameters to values that cause photon numbers to diverge, we introduce a regularization term to the total loss function

$$\text{loss}_{\boldsymbol{N}} \overset{\text{def}}{=} \beta_{\boldsymbol{N}} \times \text{MSE}(\boldsymbol{N}_{\text{avg}}, \boldsymbol{N}_{\text{tg}}), \tag{5.103}$$

where $\boldsymbol{N}_{\text{avg}}$ is the average of the photon number expectation values $\boldsymbol{N}$ over the time interval Δt, for the $\mathbf{x} = 0$ and $\mathbf{x} = 1$ valued input. The target average photon number $\boldsymbol{N}_{\text{tg}}$ is set to 2 photons in each mode to ensure that the occupation probabilities of measured Fock states are non-negligible while at the same time not over constraining the learning range. The parameter $\beta_{\boldsymbol{N}}$ is a prefactor that controls the influence of $\text{loss}_{\boldsymbol{N}}$ on the overall learning process. The total optimized loss function is then

$$\text{loss} = \text{MSE}(\bar{\mathbf{Y}}, \mathbf{Y}) + \text{loss}_{\boldsymbol{N}}.$$

The initial parameter values, their learning rates, and the hyper-parameter values are listed in the Table 5.1.

	Quantum reservoir [146]	Bosonic QNN
number of measured states	4	1
number of measurement shots	10^9	10^3

Table 5.2: Number of observables and measurement shots required to reach $100\,\%$ accuracy on the sin/square classification task for the quantum reservoir and for the bosonic QNN.

Results The results are summarized in Table 5.2. We compare the performance of the 2-mode bosonic QNN to quantum reservoir computing (see Section 4.3.3.2), and show that training the drive parameters reduces the number of observables that need to be measured down to just one, i.e the probability of having 0 photons in the first mode $P_1(0)$, compared to 4 observables for the quantum reservoir. This provides a twofold reduction in the number of measurements to perform:

(1) The total number of observables to measure is reduced.

(2) As $P_k(0) > P_k(n > 0)$, fewer measurement shots are needed to determine it with sufficient precision.

Consequently, the bosonic QNN achieves $100\,\%$ accuracy with 1000 measurement shots (Figure 5.8), in contrast to 10^9 shots required for QRC.

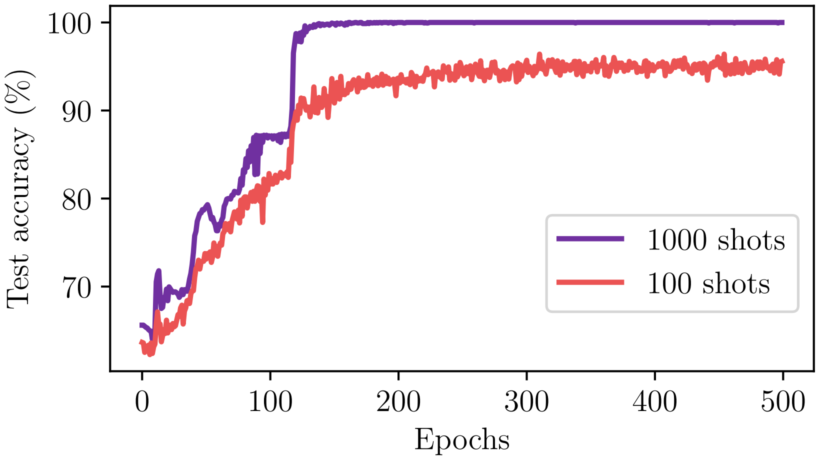


Figure 5.8: Accuracy on the sin/square classification task as a function of the number of training epochs for two different numbers of measurement shots used to determine the probability $P_1(0)$.

We note that sin/square waveform classification is a simple task, hence why measuring a single Fock state in a two-mode system is sufficient for classification. However for more complex tasks, the number of modes and measured observables will need to be higher, to increase the model expressivity.

5.7 Encoding schemes

By considering that the chosen input encoding method influences the nonlinear transformation that the quantum system applies to the input data [66], we find that it is a useful property of our bosonic QNN to leverage. We investigate the optimal encoding scheme in this section, which would minimize the amount of measured observables required for classification. Since the sin/square task is already solved with a minimal amount of resources in Section 5.6, we introduce the spirals classification task illustrated in Fig. 5.9.

5.7.1 Spirals classification task

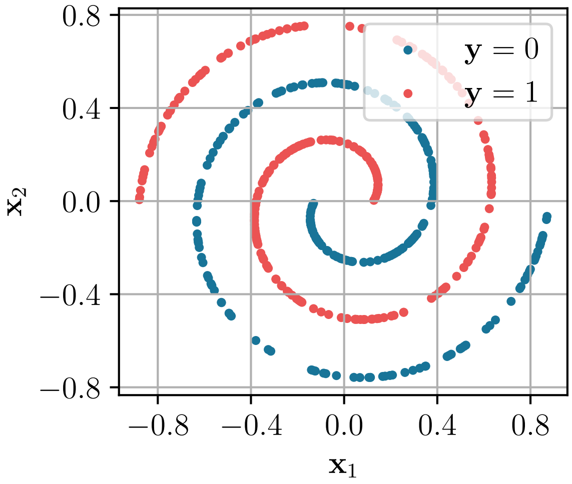


Figure 5.9: The spirals task consists in the classification of points belonging to one of the two spirals, red or blue in the figure. The inputs are data points $\{\mathbf{x}_1, \mathbf{x}_2\}$ and the labels are $\mathbf{y} = 0$ for the blue spiral and $\mathbf{y} = 1$ for the red one.

This task uses a two-class spirals dataset generated from points in polar coordinates according to

$$\begin{cases} \theta(\xi) & \sim \mathcal{U}(0, 3\pi), \\ r(\xi) & = \pm \dfrac{2\theta(\xi) + \pi}{25}, \end{cases} \tag{5.104}$$

where $\mathcal{U}(a, b)$ denotes the uniform distribution on the interval $[a, b]$. Points with a positive (negative) sign in $r(\xi)$ are labeled as class 1 (class 0), and the task consists in correctly guessing these labels. The non-linearity of the task is apparent in that it is impossible to draw a straight line to separate the two spirals, and is known to require more nonlinearity than the sin/square classification. The input data is symmetric with respect to the origin in the 2-dimensional input plane. To incorporate this symmetry into the model, we augment each input point $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$ to $(\mathbf{x}_1, \mathbf{x}_2, -\mathbf{x}_1, -\mathbf{x}_2)^T$. So we have a four dimensional input, and we need 4 input parameters.

In order to have enough complex drive parameters to encode the inputs into, we address the spirals classification task using $M = 4$ coupled modes. We compare the performance when the data is encoded into either the amplitude or phase of the complex drive amplitudes $\boldsymbol{\epsilon}$, the photon conversion rates $\boldsymbol{g}$, or two-mode squeezing rates $\boldsymbol{g^s}$.

5.7.2 Input encoding

We want to encode the augmented $d_{\mathbf{x}} = 4$-dimensional input vector $\mathbf{x}$ into the amplitude or the phase of an d_{enc}-dimensional parameter $\boldsymbol{\theta}_{\text{enc}}$. The amplitude encoding has already been used when solving the sin/square classification task with the QRC and bosonic QNN, and follows

$$\boldsymbol{\theta}_{\text{enc}}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot \mathbf{x} + \boldsymbol{\theta}_{\text{bias}}, \tag{5.105}$$

where $\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{bias}} \in \mathbb{C}^{d_{\text{enc}}}$ and $\odot$ is the element-wise vector product. Instead, encoding into the phase modifies this relation into

$$\begin{aligned} \boldsymbol{\theta}_{\text{enc}}(\mathbf{x}) &= \boldsymbol{\theta}_0 \odot e^{i\boldsymbol{\phi}(\mathbf{x})} + \boldsymbol{\theta}_{\text{bias}} \\ \boldsymbol{\phi}(\mathbf{x}) &= \boldsymbol{\phi}_0 \odot \mathbf{x} + \boldsymbol{\phi}_{\text{bias}}, \end{aligned} \tag{5.106}$$

where $\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{bias}} \in \mathbb{C}^{d_{\text{enc}}}$, and $\boldsymbol{\phi}_0, \boldsymbol{\phi}_{\text{bias}} \in \mathbb{R}^{d_{\text{enc}}}$. We initialize the phase parameters as $(\boldsymbol{\phi}_0)_k = \pi$ and $(\boldsymbol{\phi}_{\text{bias}})_k = 0$ for all $k \in [1, d_{\text{enc}}]$.

However, when encoding the input data into the phase of two-mode squeezing or photon conversion, we find that photon numbers $\boldsymbol{N}$ tend to diverge after around 10 to 200 learning epochs, even when we apply the regularization loss loss$_{\boldsymbol{N}}$ supposed to prevent high $\boldsymbol{N}$. In response, we have conducted a study of the effects of phase difference between the drives of the photon conversion and two-mode squeezing interactions, in order to find the origin of these divergences and suppress them.

5.7.3 Divergence of the mean photon number N

We have already established in Section 5.4.4 for 2 modes that if two-mode squeezing rates $g^s_{kl} \in \mathbb{C}$ have higher amplitude than photon conversion rates $g_{kl} \in \mathbb{C}$, then $\boldsymbol{N}$ diverges in time. In this section we study the dynamical behavior of $\boldsymbol{N}$ in a $(M > 2)$-mode system with complex-valued coupling rates, in order to look for abrupt increases and asymptotic divergences. In all of the cases studied in this section, we set

$$\forall k \in [1, M] \begin{cases} \kappa_k &= \kappa = 2\pi \times 2\,\text{MHz} \\ \delta_k &= 0\,\text{Hz} \\ \epsilon_k &= \epsilon = M \times 100 \times \sqrt{\kappa/4\pi}, \end{cases} \tag{5.107}$$

and the initial state is vacuum.

5.7.3.1 3 modes, $g \in \mathbb{C}, g^s = 0$

Let the photon conversion rates all have equal absolute values, but different phases, i.e

$$g_{kl} = ge^{i\phi^g_{kl}} \quad \text{for } k, l \in \{1, 2, 3\}, \tag{5.108}$$

with $g \in \mathbb{R}^+$. The two-mode squeezing tones are turned off. Then

$$G = \begin{pmatrix} 0 & g_{12} & g_{13} \\ g_{12}{}^* & 0 & g_{23} \\ g_{13}{}^* & g_{23}{}^* & 0 \end{pmatrix}$$
$$L = \frac{1}{i\hbar}\left(\begin{array}{c|c} G & \mathbb{0}_3 \\ \hline \mathbb{0}_3 & -G^T \end{array}\right). \tag{5.109}$$

To compute the eigenvalues of $F' = L - \frac{\kappa}{2}\mathbb{1}_6$, we compute those of G. Its determinant is

$$\det(\lambda \mathbb{1}_3 - G) = \lambda^3 + p\lambda + q, \tag{5.110}$$

with

$$\begin{cases} p & = -3g^2 \\ q & = -2g^3 \cos(\phi'), \end{cases} \tag{5.111}$$

where we have introduced the phase difference term

$$\phi' = \phi^g_{12} + \phi^g_{23} - \phi^g_{13}. \tag{5.112}$$

This term will be important to study destructive and constructive interference phenomena. We solve the cubic equation in Eq. (5.110) with Cardano's formula. The discriminant is

$$\Delta = -(4p^3 + 27q^2) = 108g^6 \sin^2(\phi'). \tag{5.113}$$

We will consider two cases for this discriminant: $\Delta = 0$, or $\Delta \neq 0$.

$\Delta = 0$ case. In this case, where $\phi' \equiv 0 \pmod{\pi}$, there are 3 real eigenvalues of G, one simple and a double:

$$\begin{cases} \lambda_1 & = 2g \\ \lambda_1 & = \lambda_2 = -g. \end{cases} \tag{5.114}$$

The diagonalization of $F' = U\Lambda U^{-1}$ thus yields

$$\Lambda = \text{diag}(2ig, -ig, -ig, 2ig, -ig, -ig) - \frac{\kappa}{2}\mathbb{1}_6, \tag{5.115}$$

and

$$U=\left(\begin{array}{ccc|ccc} 1 & -2 & -1 & 0 & 0 & 0\\ 1 & 1 & 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0 & 0\\ \hline 0 & 0 & 0 & 1 & -2 & -1\\ 0 & 0 & 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right). \tag{5.116}$$

By symmetry, the mean photon number is the same in all modes: $\boldsymbol{N}_1 = \boldsymbol{N}_2 = \boldsymbol{N}_3$. Using Eq. (5.69) we get

$$\boldsymbol{N}_1(t) = \kappa|\epsilon|^2 \times \left(\frac{1+e^{-\kappa t} - 2\cos(2gt)e^{-\frac{\kappa}{2}t}}{\left(\frac{\kappa}{2}\right)^2 + (2g)^2}\right). \tag{5.117}$$

We note that this behavior is the same as in the two-mode case described in Section 5.4.2, but by substituting $g \to 2g$. We interpret this as a constructive interference between different photon conversion processes. Here, no divergence behavior is observed.

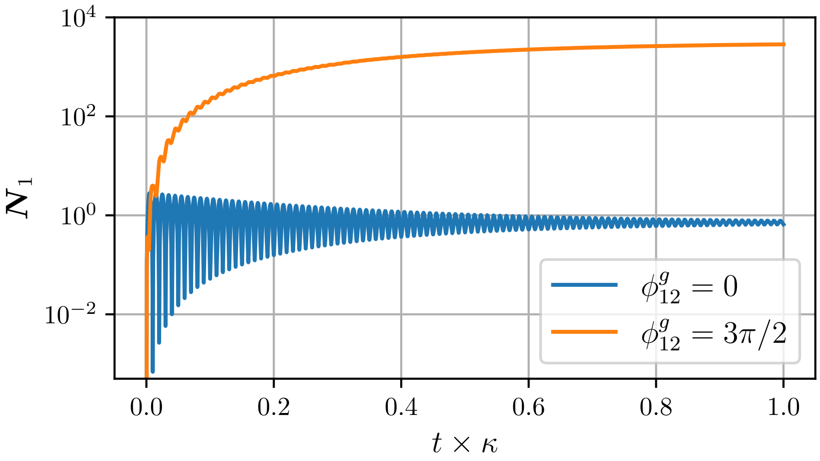


Figure 5.10: Mean photon number in the first mode $\boldsymbol{N}_1$ as a function of time for 3 coupled modes, for two values of the conversion rate between the first two modes $\phi^g_{12} = 0$ and $\phi^g_{12} = \frac{3\pi}{2}$. The conversion rate with the third mode are set to $\phi^g_{13} = \phi^g_{23} = 0$. The conversion rates amplitudes are $|g_{kl}| = g = 50\kappa$ and there is no two-mode squeezing.

$\Delta \neq 0$ case. In this case where $\phi' \not\equiv 0 \pmod{\pi}$, there are 3 real degenerate eigenvalues of G

$$\lambda_{k+1} = 2g\cos\left(\frac{|\phi'|}{3} + \frac{2k\pi}{3}\right) \quad \begin{cases} k & \in [0,1,2] \\ \phi' & \in \{-\pi, \pi\} \end{cases}. \tag{5.118}$$

The full formula of $\boldsymbol{N}_k$ is too big to calculate, but we observe that if $|\phi'| \equiv \frac{(3-4k)\pi}{2} \pmod{3\pi}$, the eigenvalue λ_k becomes zero. Hence the term associated with this eigenvalue does not oscillate, and its values evolve as if there had been no photon conversion, i.e it is divided by κ and not by a term $\approx g^2 + \left(\frac{\kappa}{2}\right)^2$, leading to much higher $\boldsymbol{N}_k(t)$. This behavior is illustrated in Fig. 5.10 for $\phi' = \frac{3\pi}{2}$. We interpret this as a destructive interference between different photon conversion processes, leading to a mean photon number $\boldsymbol{N}_k$ that has similar dynamics to the $g = 0$ case.

5.7.3.2 3 modes, $g_{kl} \in \mathbb{C}, g^s_{kl} \in \mathbb{C}$

In this section we consider both finite conversion and two-mode squeezing interactions. Computing the eigenvalues of F' in order to calculate the mean photon number in Eq. (5.70) in this case is not possible as we would have to factor its 6th-degree characteristic polynomial, so we resort to numerical simulations. Depending on the values of the photon conversion rates and two-mode squeezing rates, we have found different dynamical regimes, which we illustrate in Fig. 5.11. In all of the studies conducted, all the photon conversion rates have identical amplitudes ($|g_{kl}| = g$), as well as the two-mode squeezing rates ($|g^s_{kl}| = g^s$). We set $g > g^s$.

As described in the following sections, we observe that depending on the ratio of different coupling rates, certain combinations of their phases lead to photon number divergences. A divergence is identified by studying $\boldsymbol{N}(t)$ after a time evolution $t = 0.4\pi\kappa^{-1}$. If its smallest value for a certain coupling rate phase combination $\{\phi_{a,k}\}_k$ is exponentially smaller than its highest value for another combination $\{\phi_{b,k}\}_k$, then the photon number diverges in time when the coupling rate phases approach $\{\phi_{b,k}\}_k$.

Destructive interferences of photon conversion rates result in the divergence of the photon number $\boldsymbol{N}$. Figure 5.11a shows the mean photon number $\boldsymbol{N}$ as a function of two photon conversion rate phases ϕ^g_{12} and ϕ^g_{13}. We observe that for certain phase values, the number of photons diverges. We interpret this as a consequence of the destructive interference between multiple photon conversion processes, resulting in the creation of photons through two-mode squeezing processes, and the lack of effective photon conversion to dampen them.

Constructive interferences of two-mode squeezing rates result in the divergence of the photon number $\boldsymbol{N}$ if $|g| < 2|g^s|$. Figure 5.11b shows $\boldsymbol{N}$ as a function of two two-mode squeezing rates $\phi^{g^s}_{12}$ and $\phi^{g^s}_{13}$, for $g < 2g^s$. We observe that for certain phase values, the number of photons diverges. We interpret this as a consequence of constructive interference of the two-mode squeezing processes, resulting in an effective two-mode squeezing $2g^s$ that is higher than the photon conversion rate. In consequence, the number of photons diverges.

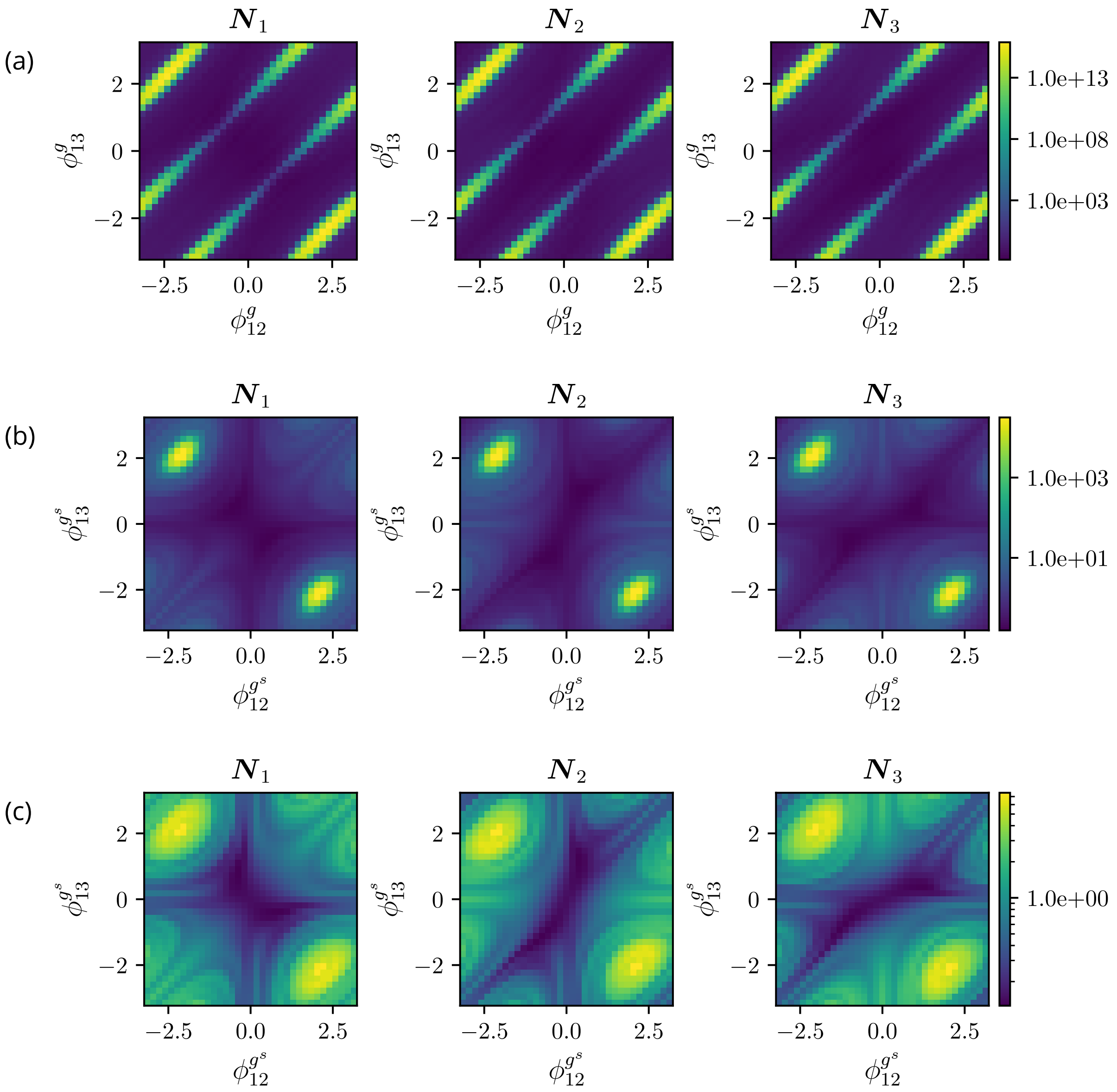


Figure 5.11: (a) The mean photon number $\boldsymbol{N}_k(t)$ in 3 modes as a function of photon conversion rate phases ϕ^g_{12} and ϕ^g_{13} at time $t = 0.4\pi\kappa^{-1}$, when $\phi^g_{23} = 0$. The photon conversion rates all have the same absolute value $|g_{kl}| = g = 50\kappa$ and the two-mode squeezing rates are $|g^s_{kl}| = g^s = 10\kappa$. (b)(c) The mean photon number $\boldsymbol{N}_k(t)$ in 3 modes as a function of the two-mode squeezing rate phases $\phi^{g^s}_{12}$ and $\phi^{g^s}_{23}$ at time $t = 0.4\pi\kappa^{-1}$, when $\phi^{g^s}_{23} = 0$. In (b) $|g_{kl}| = g = 19.5\kappa$ and $|g^s_{kl}| = g^s = 10\kappa$, and in (c) $g = 20.5\kappa$ and $g^s = 10\kappa$.

Absence of the photon number N divergence for the photon conversion rate $|g| > 2|g^s|$. Figure 5.11c shows $\boldsymbol{N}$ as a function of two two-mode squeezing rates $\phi^{g^s}_{12}$

and $\phi_{13}^{g^s}$, for $g > 2g^s$. We observe that the photon number does not diverge, although it is higher for phase values that cause the case $g < 2g^s$ to diverge. We interpret this as constructive interferences in two-mode squeezing tones not having high enough coupling rates to surpass photon conversion rates, which dampens photon creation.

5.7.4 Clamping of the coupling parameters

Taking into account the behavior of the photon numbers $\boldsymbol{N}_k(t)$ with respect to the photon conversion and two-mode squeezing rates observed in Section 5.7.3, we propose a set of heuristic guidelines for choosing the coupling rates to avoid exponential divergences in the photon numbers of the bosonic modes. We stress that these guidelines are not rigorously proven methods for preventing such divergences, but rather practical intuitions that have been effective in the training of the coupling parameters.

Algorithm 1 General guidelines for clamping

```
Clamping is applied element-wise to each component of the coupling rates.
Photon conversion rates are restricted to real, positive values. Complex values may result in destructive interference during photon conversion. In contrast, two-mode squeezing rates can be complex-valued.
The amplitude of the highest two-mode squeezing rate should never be higher than the amplitude of the lowest photon conversion rate.
If the input is encoded in the phase of the two-mode squeezing rates of $M$ modes, then the highest two-mode squeezing amplitude should never be higher than the lowest photon conversion rate amplitude divided by $M - 1$.
**if** the input $\mathbf{x}$ is encoded in one of the coupling rates according to Eq. (5.105) (amplitude encoding) or Eq. (5.106) (phase encoding) **then**
    If the encoded variable $\boldsymbol{\theta}_{\text{enc}}(\mathbf{x})$ requires clamping after a gradient descent update, the bias term $\boldsymbol{\theta}_{\text{bias}}$ is adjusted to enforce the clamping constraints, while $\boldsymbol{\theta}_0$ remains fixed. If this is insufficient to satisfy the clamping conditions, $\boldsymbol{\theta}_0$ is also clamped.
    The values $\boldsymbol{\phi}_0$ and $\boldsymbol{\phi}_{\text{bias}}$ are never clamped, even when phase encoding is used.
    After clamping, the explored values of $\boldsymbol{\theta}_{\text{enc}}(\mathbf{x})$ should deviate as little as possible from their original (pre-clamping) values.
```

From these guidelines, we propose an algorithm to clamp the photon conversion rates $\boldsymbol{g}$ and the two-mode squeezing rates $\boldsymbol{g^s}$. We define the clamping bounds $l^{g}_{\text{min}}, l^{g}_{\text{max}} \in \mathbb{R}^+$ for $\boldsymbol{g}$, and $l^{g^s}_{\text{min}}, l^{g^s}_{\text{max}} \in \mathbb{R}^+$ for $\boldsymbol{g^s}$, and $\boldsymbol{p} \leftarrow \text{clamp}(\boldsymbol{p}, p_{\min}, p_{\max})$ denotes the element-wise operation defined as $p_i \leftarrow \min(\max(p_i, p_{\min}), p_{\max})$ for each element p_i of the real vector $\boldsymbol{p}$. In practice, we find that these clamping rules effectively prevent divergences in the number of photons across all the learning tasks considered in this chapter.

Algorithm 2 Clamping rules

```
Require: $\boldsymbol{g} \in (\mathbb{R}^+)^{\frac{M(M-1)}{2}}$    ▷ To prevent destructive interference in the photon conversion rates
Require: $\boldsymbol{g^s} \in \mathbb{C}^{\frac{M(M-1)}{2}}$    ▷ The squeezing rates are allowed to be complex.
Clamping is applied element-wise to each component of the coupling rates.
$l^{g^s}_{\text{min}} \leftarrow 0\,\text{Hz}$
$l^{g}_{\text{max}} \leftarrow 500\,\text{MHz}$
if the input $\mathbf{x}$ is encoded in $\arg(\boldsymbol{g^s})$ then    ▷ We require that $\max(|\boldsymbol{g^s}|) < \frac{\min(\boldsymbol{g})}{M-1}$
    $l^{g^s}_{\text{max}} \leftarrow$ arbitrary constant value
    $l^{g}_{\text{min}} \leftarrow l^{g^s}_{\text{max}} \times (M-1)$
else    ▷ We require that $\max(|\boldsymbol{g^s}|) < \min(\boldsymbol{g})$
    $l^{g}_{\text{min}} \leftarrow \frac{\max(\boldsymbol{g^s})+\min(\boldsymbol{g})}{2}$
    $l^{g^s}_{\text{max}} \leftarrow \frac{\max(\boldsymbol{g^s})+\min(\boldsymbol{g})}{2}$
if $\mathbf{x}$ is encoded in $\boldsymbol{\epsilon}$ then
    $\boldsymbol{g} \leftarrow \text{clamp}(\boldsymbol{g}, l^{g}_{\text{min}}, l^{g}_{\text{max}})$
    $|\boldsymbol{g^s}| \leftarrow \text{clamp}(|\boldsymbol{g^s}|, 0, l^{g^s}_{\text{max}})$
else if the input $\mathbf{x} \in [0,1]$ is encoded in $|\boldsymbol{g}|$ with $\boldsymbol{g}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot \mathbf{x} + \boldsymbol{\theta}_{\text{bias}}$ then
    $\boldsymbol{g} \leftarrow \text{clamp}(\boldsymbol{g}, 0, l^{g}_{\text{max}})$
    $\boldsymbol{\theta}_0 \leftarrow \text{clamp}(\boldsymbol{\theta}_0, 0, l^{g}_{\text{max}} - l^{g}_{\text{min}})$    ▷ $\boldsymbol{\theta}_0 \in (\mathbb{R}^+)^{\frac{M(M-1)}{2}}$
    $\boldsymbol{\theta}_{\text{bias}} \leftarrow \text{clamp}(\boldsymbol{\theta}_{\text{bias}}, 0, l^{g}_{\text{min}})$    ▷ $\boldsymbol{\theta}_{\textit{bias}} \in (\mathbb{R}^+)^{\frac{M(M-1)}{2}}$
else if $\mathbf{x}$ is encoded in $\arg(\boldsymbol{g^s})$ with $\boldsymbol{g^s}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot e^{i(\boldsymbol{\phi}_0 \cdot \mathbf{x} + \boldsymbol{\phi}_{\text{bias}})} + \boldsymbol{\theta}_{\text{bias}}$ then
    $\boldsymbol{g} \leftarrow \text{clamp}(\boldsymbol{g}, l^{g}_{\text{min}}, l^{g}_{\text{max}})$
    $|\boldsymbol{\theta}_0| \leftarrow \text{clamp}(|\boldsymbol{\theta}_0|, 0, l^{g^s}_{\text{max}})$
    $|\boldsymbol{\theta}_{\text{bias}}| \leftarrow \text{clamp}(|\boldsymbol{\theta}_{\text{bias}}|, 0, l^{g^s}_{\text{max}} - |\boldsymbol{\theta}_0|)$
else if $\mathbf{x}$ is encoded in $|\boldsymbol{g^s}|$ with $\boldsymbol{g^s}(\mathbf{x}) = \boldsymbol{\theta}_0 \odot \mathbf{x} + \boldsymbol{\theta}_{\text{bias}}$ then
    $\boldsymbol{g} \leftarrow \text{clamp}(\boldsymbol{g}, l^{g}_{\text{min}}, l^{g}_{\text{max}})$
    $|\boldsymbol{\theta}_0| \leftarrow \text{clamp}(|\boldsymbol{\theta}_0|, 0, l^{g^s}_{\text{max}})$
    $|\boldsymbol{\theta}_{\text{bias}}| \leftarrow \text{clamp}(|\boldsymbol{\theta}_{\text{bias}}|, 0, l^{g^s}_{\text{max}})$
    if there exists any $\mathbf{x} \in [0,1]$ such that $\boldsymbol{g^s}(\mathbf{x}) \notin [0, l^{g^s}_{\text{max}}]$ then
        $\boldsymbol{\theta}_{\text{bias}}$ is modified such that $|\boldsymbol{g^s}(\mathbf{x})| \in [0, l^{g^s}_{\text{max}}]$ for all $\mathbf{x} \in [0,1]$
        ▷ The updated values of $g^s(\mathrm{x})$ should deviate as little as possible from their original (pre-clamping) values. The detailed clamping procedure is implemented in the source code, specifically in: tests/clamping_demonstrations/abs_encoded_cplx_theta_clamp.ipynb ◁
```

As a result of the clamping rules defined in Algorithm 2, the photon conversion rates are constrained to real values. Counting the number of trainable parameters now yields

$$\begin{cases} \boldsymbol{\epsilon} \in \mathbb{C}^M & \rightarrow 2M \\ \boldsymbol{\delta} \in \mathbb{R}^M & \rightarrow M \\ \boldsymbol{g} \in \mathbb{R}^{\frac{M(M-1)}{2}} & \rightarrow \frac{M(M-1)}{2} \\ \boldsymbol{g^s} \in \mathbb{C}^{\frac{M(M-1)}{2}} & \rightarrow M(M-1), \end{cases} \tag{5.119}$$

adding up to $\frac{3}{2}M(M-1) + 3M$. The number of parameters still scales quadratically with the number of modes.

5.7.5 Results on the spirals classification task

As seen in Section 5.7.4, we have constrained the photon conversion rates $\boldsymbol{g}$ to be real, so we address the spirals classification task with five different encoding schemes: encoding into (1) the amplitudes of the complex drives, (2) their phases, (3) the amplitudes of the two-mode squeezing rates, (4) the phases of the two-mode squeezing rates, and (5) the amplitudes of the photon conversion rates.

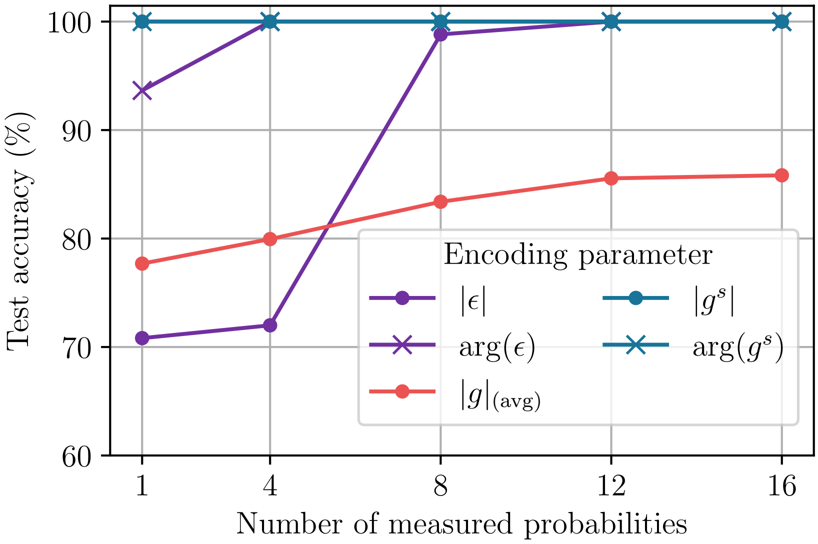


Figure 5.12: Impact of the encoding variable on the performance on the spirals classification task. The accuracy for different encoding schemes is shown as a function of the number of measured probabilities. Since the accuracies obtained when encoding in $\boldsymbol{g}$ vary greatly with the initial physical parameters $\{\boldsymbol{g}, \boldsymbol{g^s}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\}$, we average them over 5 random initial sets of parameters.

We use the Binary Cross Entropy (BCE) with logits loss for this task. It consists of two steps: applying the element-wise sigmoid function $x \rightarrow \frac{1}{1+e^{-x}}$ to the prediction $\bar{\mathbf{y}}$, followed by the BCE computation:

$$\text{BCE}(\bar{\mathbf{y}}, \mathbf{y}) = \mathbf{y} \log(\bar{\mathbf{y}}) + (1 - \mathbf{y}) \log(1 - \bar{\mathbf{y}}), \tag{5.120}$$

where $\mathbf{y}$ denotes the target label. In Figure 5.12 we plot the accuracy of the different encoding schemes as a function of the number of measured Fock state probabilities $P_k(n)$. The set of local Fock state probabilities corresponding to each number are

$$\begin{cases} 1 & \longrightarrow \{P_1(0)\} \\ 4 & \longrightarrow \{P_k(n)\}_{n=1,k\in[1,4]} \\ 8 & \longrightarrow \{P_k(n)\}_{n\in[1,2],k\in[1,4]} \\ 12 & \longrightarrow \{P_k(n)\}_{n\in[1,3],k\in[1,4]} \\ 16 & \longrightarrow \{P_k(n)\}_{n\in[1,4],k\in[1,4]} \,. \end{cases} \tag{5.121}$$

We find that encoding into either the amplitude or phase of the two-mode squeezing rates allows to achieve $100\,\%$ performance with just a single measured probability, $P_1(0)$. In contrast, encoding into the drive phase requires 4 measured probabilities to reach $100\,\%$ accuracy, encoding into the drive amplitudes requires 12 measured probabilities, and encoding into the exchange coupling rate amplitude plateaus at $85\,\%$ accuracy. The initial parameters and hyper-parameters for these simulations are listed in Table 5.3.

Parameter	Initial value	Learning rate
$\mathbf{W}_0$	1	0.1
$\mathbf{W}_{\text{bias}}$	0	0.1
κ	$2\pi \times 2\,\mathrm{MHz}$	none
$\boldsymbol{\kappa}$	$(1 \pm 0.1)\kappa$	none
$\boldsymbol{\delta}$	$(0.5 \pm 0.1$	0.1
$\boldsymbol{\epsilon}$	$400\sqrt{\kappa/4\pi}$	0.1
$\boldsymbol{g}$	$(50 \pm 5)\kappa$	0.1
$\boldsymbol{g}^{\boldsymbol{s}}$	$(10 \pm 1)\kappa$	0.1
Δt	$0.8\pi\kappa^{-1}$	none

Hyper-parameter	Value
number of modes	4
number of training epochs	500
regularization loss strength $\beta_{\boldsymbol{N}}$	0.02
$\boldsymbol{N}_{\text{thr}}$	3
$\boldsymbol{N}_{\text{tg}}$	2
number of batches	5
dataset size	500
loss function	BCE with logits

Table 5.3: Initial parameter values, their learning rates, and hyper-parameter values for the spirals classification task.

To understand why encoding into the two-mode squeezing rates performs better than the photon conversion rates or complex drive amplitudes, we study their influence on the covariance matrix $\boldsymbol{\sigma}(t)$.

5.7.6 Influence of the two-mode squeezing and photon conversion rates on the covariance matrix

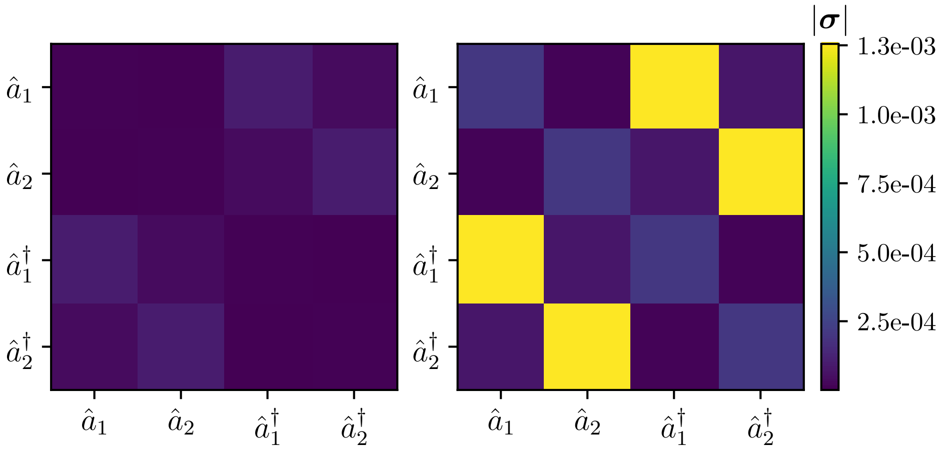


Figure 5.13: Variance matrix of the field operator covariance matrix absolute value $|\boldsymbol{\sigma}|$ for two modes after $\Delta t = 0.8\pi\kappa^{-1}$ of time evolution, with equal dissipation rates in each modes $\kappa_1 = \kappa_2 = \kappa = 2\pi \times 2\,\mathrm{MHz}$. **Left:** Varying the photon conversion rate g from 45κ to 55κ with fixed two-mode squeezing rate $g^s = 10\kappa$. **Right:** Varying the two-mode squeezing rate g^s from 5κ to 15κ with fixed $g = 50\kappa$.

We have established in Section 5.4.2 that starting from the vacuum state, if there is no two-mode squeezing and the dissipation rates of all the modes are equal, the covariance matrix remains constant: $\forall t, \boldsymbol{\sigma}(t) = \boldsymbol{\sigma}_0$. Moreover, the complex drive amplitudes have no influence on the covariance matrix in general.

In this section, we analyze in the 2-mode case the variance of the covariance matrix as a function of the photon conversion rate g and the two-mode squeezing rate g^s. As shown in Fig. 5.13, every term of the covariance matrix exhibits lower variance when g is varied compared to when g^s is varied. Therefore encoding into the two-mode squeezing rates allows for more information to be processed through the covariance matrix, compared to encoding into the photon conversion rates or complex drive amplitudes.

5.7.7 Comparison with the quantum reservoir and a classical model

In order to pin down the advantage brought by the training of the quantum system parameters, we compare the resources in terms of the number of trained parameters, and the number of outputs that need to be measured in order to reach $100\,\%$ accuracy on the spirals task for the quantum reservoir and bosonic QNN. The results are summarized in Table 5.4. We observe that the bosonic QNN needs a significantly smaller number of observables to measure compared to quantum reservoir computing. Furthermore, as a classical software point of comparison, to reach equivalent accuracy, a

	Quantum Reservoir	Bosonic QNN
number of modes M	4	4
number of measured states	36	1
parameters	37	38

Table 5.4: Number of neurons and parameters needed to reach $100\,\%$ accuracy on the spirals classification task using quantum reservoir and bosonic QNN.

classical MLP needs 2 hidden layers with 6 neurons each, resulting in 79 parameters (see Fig. 5.14). We observe that the bosonic QNN requires less parameters than the software MLP, proving its competitiveness in expressivity compared to classical learning models.

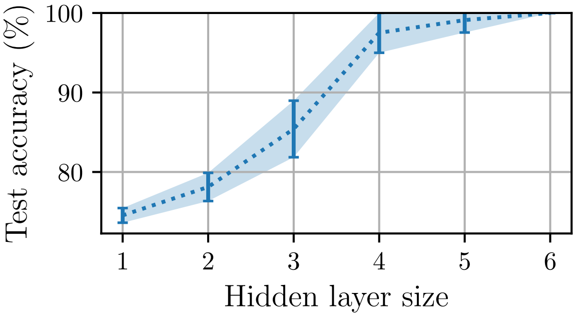


Figure 5.14: Test accuracy of a 2-layer MLP on the spirals classification task as a function of the number of neurons per hidden layer, as a classical software point of comparison to the bosonic QNN. The test accuracies are averaged over 5 different random initializations of the MLP weights, and the error bars correspond to the standard deviation.

5.8 Handwritten digits classification

Training the physical parameters increases the expressivity of the quantum neural network and allows it to solve tasks that are out of reach for quantum reservoir computing on the same hardware. We demonstrate this by solving the handwritten digits recognition task, from the DIGITS dataset, already described in Section 2.1.4. The dataset is imported from the `scikit-learn` Python library [167].

We use 6 modes, pairwise coupled through 15 two-mode squeezing processes whose amplitudes are used for data encoding. 64 pixel images cannot be processed in a single time step, so we use an encoding scheme inspired by the data re-uploading method [168]. After removing the 4 white corner pixels, we divide each input image in four 15-pixel batches, that we apply over 4 time intervals $\frac{\Delta t}{4} = \frac{2\pi}{10\kappa}$, as shown in Fig. 5.15.

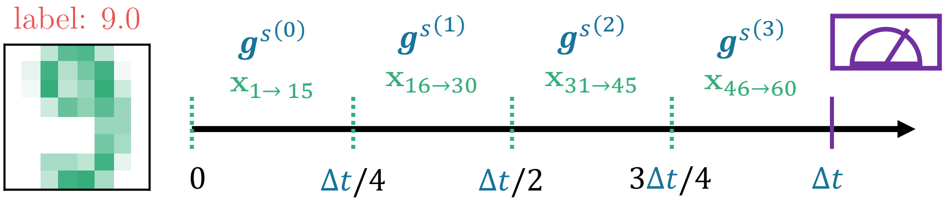


Figure 5.15: Encoding scheme for handwritten digits classification. To create a flattened image vector of size 60, we crop the 4 white corners of the original image. Using 6 modes, we encode 15 pixels into the amplitudes of the 15 two-mode squeezing rates at times 0, $\frac{\Delta t}{4}$, $\frac{\Delta t}{2}$ and $\frac{3\Delta t}{4}$. During each data re-uploading instance, a new set of parameters $\{\boldsymbol{g}, \boldsymbol{g^s}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\}$ is applied. At time Δt, the Fock state probabilities are measured, yielding the feature vector $\mathbf{F}(\mathbf{X})$.

The cost function used in this task is the Cross Entropy, as implemented in `PyTorch` [169]. Predictions $\bar{\mathbf{y}}$ are first passed through the softmax function

$$\forall j \in \{1, \dots, d_{\bar{\mathbf{y}}}\}, \bar{\mathbf{y}}_j \to \frac{\exp(\bar{\mathbf{y}}_j)}{\sum_{k=1}^{d_{\bar{\mathbf{y}}}} \exp(\bar{\mathbf{y}}_k)}, \tag{5.122}$$

where $d_{\bar{\mathbf{y}}}$ is the dimension of the prediction. The cross-entropy loss

$$\text{CE}(\bar{\mathbf{y}}, \mathbf{y}) = \sum_{k=1}^{d_{\bar{\mathbf{y}}}} \mathbf{y}_k \log(\bar{\mathbf{y}}_k) \tag{5.123}$$

between the prediction logits $\bar{\mathbf{y}}_k$ and target labels $\mathbf{y}_k$ is then computed.

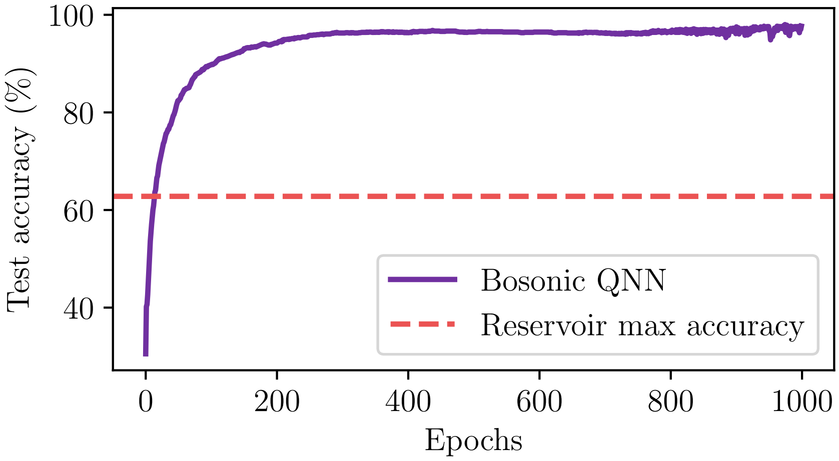


Figure 5.16: Comparison of the bosonic QNN performance to that of quantum reservoir computing on the DIGITS dataset classification task, with 6 modes. The classification accuracy on the test dataset is plotted as a function of the learning epoch. 12 probabilities are measured for the bosonic QNN, while 36 are measured for the quantum reservoir, whose accuracies reaches at best $62.8\,\%$.

Parameter	Initial value	Learning rate
$\mathbf{W}_0$	1	0.01
$\mathbf{W}_{\text{bias}}$	0	0.01
κ	$2\pi \times 2\,\mathrm{MHz}$	none
$\boldsymbol{\kappa}$	$(1 \pm 0.1)\kappa$	none
$\boldsymbol{\delta}$	$0\,\mathrm{Hz}$	0.01
$\boldsymbol{\epsilon}$	$600\sqrt{\kappa/4\pi}$	0.01
$\boldsymbol{g}$	$(50 \pm 5)\kappa$	0.01
$\boldsymbol{g}^{\boldsymbol{s}}$	$(10 \pm 1)\kappa$	0.01
Δt	$0.8\pi\kappa^{-1}$	none

Hyper-parameter	Value
number of modes	6
number of training epochs	1000
regularization loss strength $\beta_{\boldsymbol{N}}$	0.12
$\boldsymbol{N}_{\text{thr}}$	3
$\boldsymbol{N}_{\text{tg}}$	2
number of batches	5
dataset size	1500
loss function	Cross Entropy

Table 5.5: Initial parameter values, their learning rates, and hyper-parameter values for the DIGITS classification task.

Using this method, we achieve in Fig. 5.16 over $97\,\%$ accuracy on the DIGITS classification task, by measuring probability amplitudes $P_k(n)$ for $k \in \{1, 2, 3, 4\}$ and $n \in \{0, 1, 2\}$, resulting in a feature vector $\mathbf{F}(\mathbf{X})$ with 12 components, and training a total of 502 physical parameters and output weights. The initial parameter and hyper-parameter values are listed in Table 5.5.

In comparison, a reservoir computing algorithm with 6 modes and random initial parameters can achieve at best $62.8\,\%$ test accuracy when measuring 36 probability amplitudes $\{P_k(n)\}_{n\in[1,6],k\in[1,6]}$. For this bosonic QRC benchmark, we trained the reservoir with 5 different sets of initial random parameters, and selected the best performance.

Experimentally, measuring this many Fock state occupations would be very challenging. We conclude that learning the physical parameters allows the bosonic QNN to solve tasks for which a bosonic QRC is not expressive enough.

5.9 Discussion

We have demonstrated that the bosonic quantum neural network can be successfully trained by optimizing the complex parameters of the three-wave mixing interactions and nearly resonant drives. Training not only enhances the expressivity of the network, enabling it to solve more complex tasks such as handwritten digits classification, but also drastically reduces the number of observables that need to be measured.

For instance, in the sin/square and spirals classification tasks, the number of variables is reduced to a single one, compared to 4 and 36, respectively in quantum reservoir computing. This makes the experimental implementation of inference significantly more practical after training. Importantly, we show that trained networks can address tasks that are beyond the capabilities of a quantum reservoir of the same size. Given the experimental challenges of scaling up the quantum systems, the ability to solve harder tasks without increasing the system size represents a key advantage of the proposed approach. Furthermore, we have shown that encoding into the two-mode squeezing drive is advantageous compared to encoding in drives of photon conversion, which indicates that quantum coherences have a significant participation in learning.

Chapter 6

Conclusion

In this thesis, we have studied learning with bosonic modes in two different frameworks: quantum reservoir computing, and analog quantum neural networks. In Chapter 3, we explained that bosonic modes have linear dynamics, which poses an obstacle to their application in the processing of nonlinear data. The source of nonlinearity we have chosen to exploit is the measurement of non-Gaussian observables, namely Fock basis states.

Using the quantum reservoir computing framework, we have benchmarked the performance of a two-mode system with Fock basis measurements on the sin/square waveform classification task, and found that a significantly smaller number of features, 4 instead of 17, need to be measured on the quantum system compared to a classical one. To explore the source of this advantage, we have simulated the quantum reservoir in its classical limit by introducing a significant dephasing rate compared to the dissipation rates during its dynamical evolution, which suppresses quantum coherences. In the classical limit, the bosonic reservoir performance is degraded, 9 features need to be measured in order to solve the task. From these tests, we could conclude that quantum coherences play an important role in quantum reservoir computing, increasing the number of accessible states for computation in the Hilbert space. Additionally, we have studied the influence of shot noise stemming from the finite sampling of quantum observables on the reservoir performance. We observed that with 4 measured features, 10^9 measurements are required to recover $100\,\%$ classification accuracy on the sin/square task. To reduce this number, adaptive measurement schemes could be considered, such as requiring less measurement samples for highly occupied states. However, this motivates the consideration of learning frameworks in which less observables need to be measured, and we hoped training the physical parameters would fulfill this need.

To improve on the performances of reservoir computing, we proposed in Chapter 5 to train physical coupling parameters between the modes. In order to obtain tunable parameters, we have introduced parametric coupling between the modes using three-wave mixing interactions. This allowed us to tune photon conversion and two-mode squeezing rates in the system. For the sake of computing efficiency, we have limited ourselves to Gaussian modes whose dynamics can be computed with $\mathcal{O}(M^3)$ time complexity using the Heisenberg picture and Gaussian boson sampling, instead of the exponential time complexity arising in Schrödinger picture simulations. This enables us to efficiently simulate a bosonic quantum neural network with the `PyTorch` package to leverage its automatic differentiation functionality, and train the physical parameters with gradient descent.

Our benchmarking of the two-mode bosonic neural network on the sin/square waveform classification shows that, despite the quadratic number of parameters, learning allows to reduce the number of measured observables from 4 to 1, in order to solve the task. Moreover, only 10^3 measurements are required to keep the same learning performance. This advantage comes from the fact that in the bosonic quantum neural network we measured a single Fock state with high occupation probability, whereas in reservoir computing we needed to measure 4 states with lower occupation probabilities overall. These results show that learning the coupling parameters greatly reduces the resources required for data processing, after the initial training phase.

In the Section 5.7, we have investigated the performance of different input encoding schemes. Because the sin/square task was solved with a single measured observable and 2 coupled modes, we addressed a more complex task, that is spirals classification. In order to have enough complex drive amplitudes to encode the input on, we increased the number of coupled modes to 4.

We considered encoding into either the amplitude or phase of the complex drive amplitudes, the photon conversion rates, or the two-mode squeezing rates. When encoding into coupling parameters, the divergence of the photon number for certain parameter combinations motivated an examination of the constructive or destructive interference occurring when applying complex coupling rates. From the observed phenomena, we devised clamping rules which kept the number of photons from diverging during the training phase. After learning on the spirals task with application of the clamping rules, we observed that encoding into the two-mode squeezing rate amplitude or phase yielded the best performance, solving the spirals task with a single measured Fock state. This advantage can be attributed to its higher influence on the covariance matrix, compared to the photon conversion rate. The bosonic neural network required the optimization of 38 parameters, while the smallest number of parameters a 2-layer MLP requires to solve the same task is 79, illustrating the competitiveness of the bosonic neural network.

Finally, in Section 5.8 we have addressed a more complex tasks, of handwritten digits classification using both the trained bosonic quantum neural network and quantum reservoir computing of equivalent sizes. We have observed an improvement of performance with the trained network achieving $97\,\%$ accuracy when measuring 12 Fock states, compared to the quantum reservoir achieving at best $62.8\,\%$ accuracy when measuring 36 different Fock states. From these tests, we can conclude that training increases the expressivity, and allows the trained network to solve tasks which are too complex for the quantum reservoir.

However, the back-propagation training is not scalable, as it requires to classically simulate the system to determine the gradients. When training an experimental bosonic QNN, the choice of the optimization method will be key to determining the resources required for their training. For larger quantum neural networks, physics-compatible training methods will be crucial, and research is very active in this field. For instance, equilibrium propagation [84, 86] computes gradients of energy-based models from the activities of neurons that can be measured on the physical system. It has been extended to quantum models such as the Ising model and coupled bosonic modes by considering the expectation value of the Hamiltonian as the energy function to minimize [170]. Another promising physics-compatible training method is the forward-forward algorithm [85], which replaces the forward and backward passes of back-propagation by two forward passes, one with positive and the other with negative data.

In my PhD work, we have used Gaussian states for the bosonic QNN because they are simulable in polynomial time with a classical computer, but physics-compatible training methods would remove the constraint on needing a model of the experiment. With non-Gaussian states, we could then increase the number of trainable parameters by incorporating higher-order mixing processes, such as four-wave mixing, and explore more complex quantum states.

Finally, bosonic neural networks may be inherently less susceptible to barren plateaus than parameterized quantum circuits. This is expected due to the more structured way in which information is encoded, avoiding fully random unitary evolutions, and the use of parametric couplings that introduce trainable interactions without excessive scrambling, thereby preserving useful gradient information. Nevertheless, further research is needed to fully understand their scalability, robustness to decoherence, and the potential emergence of barren plateaus in larger architectures.

Chapter 7

Résumé en français

7.1 Introduction

Le cerveau est très efficace en traitement d'information et en résolution de diverse tâches. En plus de la pensée logique, il gère plusieurs sortes de données sensorielles, et peut résoudre des tâches sans être explicitement programmé pour. Désirant de répliquer et même surpasser le pouvoir calculatoire du cerveau, les scientifiques ont développé des modèles d'apprentissage inspirés de systèmes neuronaux biologiques: c'était la naissance de l'apprentissage automatique.

Une neurone biologique est un oscillateur non-linéaire, recevant des pics de signaux électriques d'autres neurones à travers des synapses. Chaque pic reçu contribue à la charge de sa membrane, et à son propre potentiel électrique. Quand ce potentiel dépasse une certaine limite, la neurone se décharge en émettant un pic de signal électrique elle-même. Les synapses peuvent exciter ou atténuer le signal qu'elles transmettent, et l'apprentissage est associé avec un renforcement ou une atténuation de leur poids. De plus, étudier des cerveaux avec des électroencéphalogrammes et des champs magnétiques a révélé que des phénomènes tels que la synchronisation d'oscillations de différentes neurones dans le cerveau sont associés à l'apprentissage de paternes [1, 2, 3, 4].

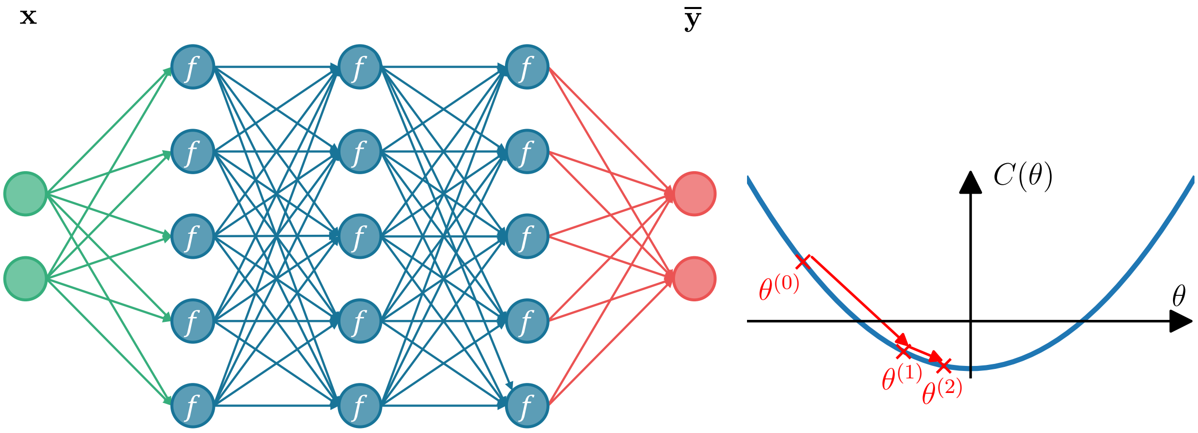
Afin de à la fois comprendre le traitement biologique de l'information et pour améliorer les méthodes actuelles d'apprentissage automatique, des modèles dynamiques d'apprentissage automatique sont en cours de recherche [5]. De plus, les premiers développements en apprentissage automatique étaient basés sur des modèles inspirés par la physique, tels que les réseaux de Hopfield [6] et les machines de Boltzmann [7].

Cependant, le succès de l'apprentissage profond [8] sur plusieurs tâches de reconnaissance d'images ou de traitement de langage a mené vers l'adoption générale de modèles d'apprentissage automatiques simulés numériquement tels que les réseaux de neurones artificielles (ANN) [9] qui implémentent les neurones en tant que fonctions non-linéaires, et les synapses en tant que valeurs scalaires et statiques. Beaucoup d'avancées algorithmiques ont contribué à cette transformation, telles que l'adoption de couches neuronales profondes pour augmenter l'expressivité (c'est-à-dire la capacité à approximer une fonction arbitraire) des modèles d'apprentissage, et l'algorithme de rétropropagation du gradient [10], qui permet des calculs analytiques efficaces de gradients en simulation. Ces gradients calculés permettent l'entraînement de modèles à travers des règles de descente de gradient.

Malgré leur influence toujours grandissante sur la technologie, les réseaux de neurones simulés montrent certaines limites. Contrairement aux cerveaux qui à la fois sauvegardent la mémoire des synapses et traitent l'information, l'architecture von Neumann d'ordinateurs conventionnels sépare leurs unités de mémoire et de calcul. En conséquence, le transfert continuel d'information entre ces unités crée des surcoûts significatifs en énergie et latence: ce phénomène est connu comme le goulet d'étranglement de von Neumann [11]. Ce problème combiné avec l'utilisation des ANN pour traiter des jeux de données toujours grandissant a mené vers un entraînement de plus en plus gourmand en énergie des modèles d'apprentissage modernes [12], au point que Amazon et Google investissent en ce moment dans des centrales nucléaires, dans cet unique but. Dans un contraste marquant, le cerveau humain opère avec une efficacité énergétique remarquable, en consumant uniquement autour de $20\,\mathrm{W}$. Ces considérations ont motivé l'exploration de systèmes neuromorphiques efficaces en énergie implémentant des ANN sur des support physiques dédiés: ce domaine est appelé le calcul neuromorphique. Différents supports physiques ont été étudiées dans ce but, tels que les matrices croisées de memristors en électronique [14], les modulateurs spatiaux de lumière en photonique [15], et les dispositifs spintroniques [16] pour en nommer quelques-uns.

Nous illustrons comment les réseaux de neurones artificielles sont entraînés avec des passes en avant pour calculer les prédictions et une optimisation basé sur les gradients, sur l'exemple du perceptron multicouche (MLP) représenté en Fig. 7.1a. Les neurones appliquant une fonction d'activation non-linéaire f sont groupés par couches, qui sont intégralement connectées entre elles à travers des connections linéaires synaptiques. Dans une passe en avant, la première couche encode un vecteur d'entrée $\mathbf{x}$, puis passe à travers multiples couches cachées, et la dernière couche donne la prédiction du réseau $\bar{\mathbf{y}}$, qui est utilisée pour calculer une fonction de coût C. La fonction de coût quantifie la différence entre la cible et la prédiction: le but d'un algorithme d'apprentissage est de la minimiser, par rapport aux paramètres du modèle.

Dans les réseaux de neurones artificielles, les poids linéaires des connections synaptiques sont les paramètres du modèle. La fonction de coût $C(\boldsymbol{\theta}, \mathbf{x})$ dépend d'un ensemble de paramètres du réseau $\boldsymbol{\theta}$ et des entrées $\mathbf{x}$. L'optimisation par descente de gradient est basée sur le fait que l'extrême d'une fonction a un gradient nul. Elle a prouvé



(a) (b)

Figure 7.1: (a) Illustration d'un perceptron multicouche (MLP) avec trois couches cachées. Les entrées bidimensionnelles $\mathbf{x}$ (cercles verts) sont encodés dans les couches cachées (cercles bleus) avec des connections synaptiques linéaires (flèches), jusqu'aux prédictions bidimensionnelles $\bar{\mathbf{y}}$ (cercles rouges). Chaque couche cachée contient 5 neurones, qui appliquent une fonction d'activation non-linéaire f. Les poids des connections synaptiques constituent les paramètres du modèle. (b) Illustration de l'algorithme de descente de gradient, pour l'optimisation d'une fonction de coût $C(\theta)$ par rapport à un paramètre unidimensionnel θ. En commençant depuis $\theta^{(0)}$, le paramètre est mis à jour de manière itérative (croix rouges) avec la règle $\theta^{(n+1)} = \theta^{(n)} - \gamma \nabla_\theta C(\theta^{(n)})$, où γ est un taux d'apprentissage. L'algorithme converge vers le minimum de cette fonction, car plus il en approche, plus le gradient $\nabla_\theta C$ devient petit.

son efficacité pour entraîner des modèles larges d'apprentissage profond tel que GPT-4 [17] ou ResNet [18]. Pour minimiser une fonction de coût $C(\theta, \mathbf{X})$ par rapport à un jeu de données $\mathbf{X} = \{\mathbf{x}\}$, un exemple d'algorithme de descente de gradient appelé descente de gradient stochastique (SGD) met à jour de manière itérative l'ensemble initial des paramètres $\boldsymbol{\theta}^{(0)}$ avec la relation

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \gamma \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(n)}, \mathbf{x}), \tag{7.1}$$

où γ est un taux d'apprentissage défini entre 0 et 1, et l'entrée $\mathbf{x}$ est choisie aléatoirement dans le jeu de données. Cette règle de mise à jour est montrée en Fig. 7.1b avec une fonction de coût dépendant d'un paramètre unidimensionnel (pour la clarté visuelle), et aucune entrée. Les gradients $\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(n)}, \mathbf{x})$ sont calculés avec l'algorithme de rétropropagation de gradient. C'est l'algorithme le plus commun en apprentissage automatique, et il est expliqué en Section 2.1.3.

D'un autre côté, il semble que les dispositifs quantiques ont le potentiel de transformer le futur dans plusieurs domaines du calcul. L'idée d'utiliser des systèmes quantiques contrôlés précisément pour le calcul [19, 20] a pris forme durant la seconde révolution quantique des années 1980, quand des expériences ont démontré des mesures et manipulations sur des systèmes quantiques individuels [21]. Bien que les premières applications conçues pour les ordinateurs quantiques se concentraient sur la simulations de systèmes physiques, les résultats emblématiques de Peter W. Shor et Lov K. Grover ont démontré que les ordinateurs quantiques pouvaient aussi surpasser les ordinateurs classiques dans certaines tâches algorithmiques, respectivement la factorisation en nombres premiers [22] et la recherche non-structurée [23]. Cette diversité d'applications potentielles combinée avec la maitrise graduelle de la physique quantique expérimentale des dernières décennies a motivé un intérêt grandissant pour la construction d'ordinateurs quantiques utiles, à la fois dans les secteurs académiques et industriels.

En conséquence, les chercheurs se sont demandés si combiner les dispositifs quantiques et l'apprentissage automatique pourrait créer des résultats intéressants. Cette thèse se place dans le contexte de l'apprentissage automatique quantique (QML), un domaine de recherche qui s'est rapidement développé ces 10 dernières années.

7.2 Apprentissage automatique quantique

La perspective de l'apprentissage automatique quantique repose dans l'accroissement exponentiel de l'espace des états (espace de Hilbert) des systèmes quantiques avec le nombre de ses composants, tel que illustré Fig. 7.2 dans le cas d'un ensemble de qubits. En contraste avec le calcul neuromorphique classique où un unique composant correspond à une neurone et le couplage entre deux composants correspond à une synapse, dans le régime quantique, les états de base peuvent être considérés comme des neurones, et les probabilités de transition d'état peuvent être considérées comme des synapses. Ce changement de paradigme fournit un espace des caractéristiques (qui est un environnement conceptuel où chaque dimension représente une caractéristique spécifique des données analysées ou utilisées dans les modèles d'apprentissage automatique) qui est exponentiel par rapport à la taille des systèmes physiques. De par cette propriété, il y a une espoir qu'apprendre avec des systèmes quantiques pourrait produire des modèles d'apprentissage avec une expressivité plus élevée.

De plus, de multiples études empiriques et théoriques ont montré que les modèles d'apprentissage quantiques on tendance à avoir une meilleure capacité à généraliser (c'est à dire la capacité à bien traiter des données nouvelles) que des modèles classiques [24, 25, 26]. Améliorer la capacité à généraliser est important pour des applications où de larges jeux de données fiables sont difficiles à obtenir.

Finalement, une perspective intéressante pour le calcul neuromorphique quantique est d'apprendre sur des données quantiques. En effet, fournir des données quantiques

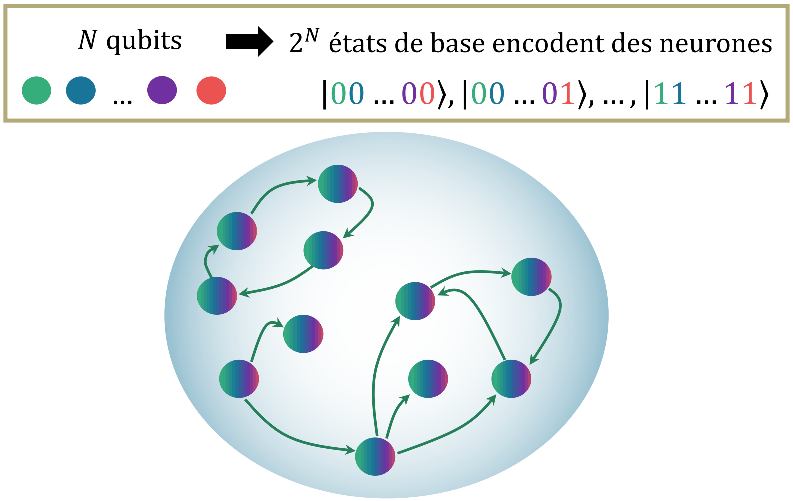


Figure 7.2: Schéma de la croissance exponentielle de l'espace de Hilbert. L'espace de Hilbert est représenté par le cercle bleu clair, et les petits cercles multicolores correspondent aux états collectifs de qubits. Des flèches les reliant symbolisent les probabilités de transition d'état. Le nombre d'états de base évolue en 2^N, où N est le nombre de qubits.

à un réseau de neurones classique requiert des mesures afin d'obtenir une représentation classique de l'état quantique. Avec les réseaux de neurones quantiques, cela serait simplifié car directement traiter les données quantiques avec un dispositif neuromorphique quantique pourrait être moins gourmand en ressources [27, 28, 29, 30]. Cela est particulièrement utile dans le contexte actuel d'une rapide évolution des technologies quantiques, où une large quantité de données quantiques doit être traitée efficacement [31].

Deux approches principales ont émergé en apprentissage automatique quantique, avec différentes méthodes d'optimisation: dans les circuits quantiques paramétrés (PQC) les paramètres physiques sont entraînés, tandis que dans le calcul par réservoir quantique (QRC) un système quantique non entraîné projette les données d'entrée vers un espace dimensionnel plus grand, qui est ensuite classé par régression linéaire sur un ordinateur classique.

7.2.1 Circuits quantiques paramétrés

Optimiser un dispositif physique pour une tâche d'apprentissage automatique implique l'exécution de passes en avant (où le dispositif émet une prédiction en fonction d'une entrée) dont les prédictions sont utilisées pour calculer la fonction de coût $C(\boldsymbol{\theta})$. La fonction de coût est minimisée par rapport aux paramètres du dispositif $\boldsymbol{\theta}$, qui correspondent aux paramètres entraînables du modèle. Tandis que dans les réseaux de neurones simulés classiques la première couche de neurones encode les entrées,

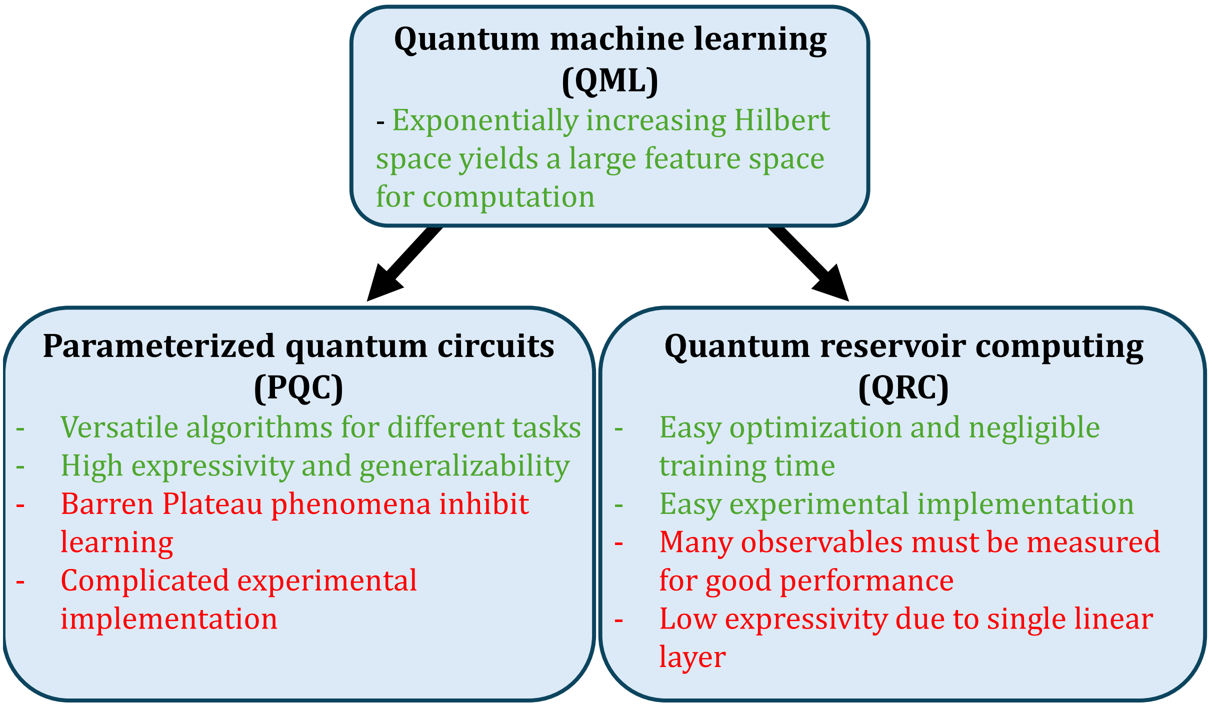

Quantum machine learning (QML)
- Exponentially increasing Hilbert space yields a large feature space for computation
Parameterized quantum circuits (PQC)
- Versatile algorithms for different tasks
- High expressivity and generalizability
- Barren Plateau phenomena inhibit learning
- Complicated experimental implementation
Quantum reservoir computing (QRC)
- Easy optimization and negligible training time
- Easy experimental implementation
- Many observables must be measured for good performance
- Low expressivity due to single linear layer

Figure 7.3: Diagramme comparant les avantages (vert) et désavantages (rouge) dans deux approches différentes à l'apprentissage automatique quantique. Les circuits quantiques paramétrés implémentent l'apprentissage en entraînant des angles de rotation de porte logiques, tandis que les réservoirs quantiques utilisent un système quantique non entraîné pour projeter les données d'entrée vers un espace dimensionnel plus grand, qui est ensuite classé par régression linéaire sur un ordinateur classique.

et les poids synaptiques sont les paramètres entraînés, dans les circuits quantiques, les paramètres des portes logiques unitaires sont utilisés à la fois pour encoder les entrées et fournir des paramètres entraînables. Une prédiction est obtenue à partir d'observables mesurés, et une fonction de coût en est déduite. En utilisant un ordinateur conventionnel, des outils de l'optimisation classique tel que la descente de gradient sont utilisés pour mettre à jour les paramètres des portes logiques. Une telle optimisation hybride dans les circuits quantiques paramétrés (PQC) [32] est illustrée en Fig. 7.4.

Dans les premières implémentation des PQC, les gradients étaient estimés avec des approximations variationnelles du premier ordre

$$\nabla_\theta C \approx \frac{C(\theta + d\theta) - C(\theta)}{d\theta}. \tag{7.2}$$

Cette méthode était imprécise et en conséquence rendait difficile la convergence des algorithmes d'optimisation vers une solution. Pour résoudre ce problème, des règles de décalage de paramètre ont été trouvées, permettant de calculer analytiquement des

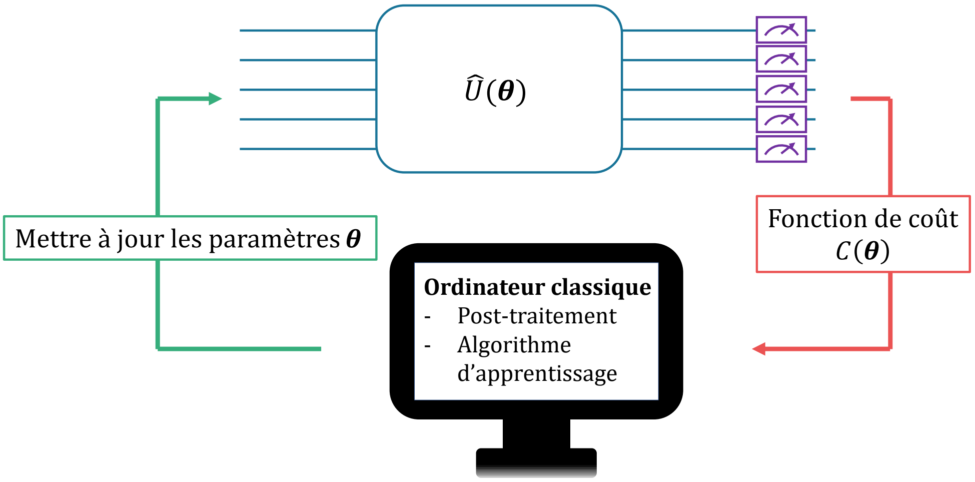

$\hat{U}(\boldsymbol{\theta})$
Mettre à jour les paramètres $\boldsymbol{\theta}$
Fonction de coût $C(\boldsymbol{\theta})$
Ordinateur classique
- Post-traitement
- Algorithme d'apprentissage

Figure 7.4: Schéma de l'optimisation hybride d'un circuit quantique paramétré (PQC). Chaque ligne bleue correspond à un qubit. Pour évaluer une fonction de coût $C(\boldsymbol{\theta})$, des portes logiques avec des paramètres $\boldsymbol{\theta}$ transforment l'état du circuit, avec des opérations unitaires $\hat{U}(\boldsymbol{\theta})$. Puis $C(\boldsymbol{\theta})$ est calculé à partir des observables mesurées dans le circuit (boîtes violettes). Les mises à jour de paramètres sont calculées en utilisant des outils de l'optimisation classique sur un ordinateur classique conventionnel, qui sont réinjectés dans les portes logiques pour la prochaine passe en avant.

gradients à travers deux passes en avant du PQC [33, 34]. Par exemple, le gradient par rapport au paramètre θ d'une porte logique de rotation est [35]

$$\nabla_\theta C(\theta) = \frac{1}{2}\left(C\left(\theta + \frac{\pi}{2}\right) - C\left(\theta - \frac{\pi}{2}\right)\right). \tag{7.3}$$

Les règles de décalage de paramètre ont amélioré la précision des gradients estimés dans les PQC, ce qui en conséquence a augmenté leurs capacité à être finement ajustés [36].

Cependant, le large espace de Hilbert promettant une haute expressivité des circuits quantiques paramétrés est aussi à la source de problèmes dans leur optimisation. En effet, lors de l'implémentation à grande échelle de PQC hautement expressifs, les gradients mesurés avec les paramètres de décalage de gradient disparaissent exponentiellement avec le nombre de dispositifs quantiques [37]. C'est le phénomène des plateau stériles (BP), qui rend l'apprentissage des PQC impossible à grande échelle en ce moment. Il y a une recherche très active pour chercher des architectures de PQC qui pourraient éviter les plateaux désertiques à grande échelle [38, 39, 40, 41].

7.2.2 Calcul par réservoir quantique

Dans ce contexte, le calcul par réservoir quantique a été proposé par Ref. [42] pour l'apprentissage avec des système quantiques. Il est directement inspiré du calcul par réservoir (RC), une architecture développée pour aborder les difficultés d'entraînement des réseaux de neurones classiques lors du traitement de tâches dépendantes du temps [43, 44]. Le calcul par réservoir est appliqué dans un scénario d'apprentissage supervisé, où le jeu de données en entrée $\mathbf{X}$ est associé à un jeu de données cible $\mathbf{Y}$. Le but de l'apprentissage est d'obtenir une prédiction $\bar{\mathbf{Y}}$ aussi proche que possible de la cible. En substance, un ensemble de neurones avec des connections synaptiques aléatoires et non entraînées (appelé le 'réservoir') est utilisé pour projeter $\mathbf{X}$ sur un espace des caractéristiques linéairement séparable durant la passe en avant, qui est ensuite classé avec une régression linéaire.

Curieusement, le réservoir peut être remplacé par un système physique, dans lequel les entrées sont encodés à travers des signaux et les sorties sont obtenues en mesurant certains observables. Le calcul par réservoir a d'abord été implémenté sur des systèmes physiques classiques, allant de l'électronique [45, 46, 47, 48], l'optique [49, 50], à la spintronique [51, 52]. Dans le domaine du calcul neuromorphique avec des systèmes physiques, c'est particulièrement intéressant car l'algorithme de rétropropagation du gradient n'est pas compatible avec l'apprentissage dans le cerveau ou les dispositifs physiques. En effet la rétropropagation du gradient requiert la sauvegarde en mémoire de la sortie de chaque couche de neurones, plutôt que juste la dernière. Dans une architecture de calcul par réservoir, les seuls paramètres appris sont sauvegardés dans un ordinateur classique, les rendant entraînable par n'importe quelle méthode d'optimisation classique.

Dans le QRC le réservoir physique est dans le régime quantique, et ses hyperparamètres sont fixés et aléatoires. Par exemple si le réservoir est un ensemble de spins, alors ses hyperparamètres sont les forces de couplage entre paires de spins. Durant une passe en avant, nous laissons sa dynamique évoluer librement pendant une durée donnée, et mesurer certains observables fournit les caractéristiques de sortie du réservoir $\mathbf{F}(\mathbf{X})$. La prédiction $\bar{\mathbf{Y}} = \mathbf{W}\mathbf{F}(\mathbf{X})$ est alors apprise en utilisant une régression linéaire, où $\mathbf{W}$ est la matrice des poids de sortie. Cet algorithme est illustré en Fig. 7.5.

Impartir l'optimisation vers une régression linéaire en simulation après le réservoir quantique permet d'éviter l'apprentissage des paramètres du système quantique, et donc évite les phénomènes de plateaux stériles. Cependant un réservoir est essentiellement un réseau à une seule couche. Comme l'expressivité des réseaux de neurones augmente avec son nombre de couches [53], l'expressivité des réservoirs quantiques est donc limité comparé aux circuits quantiques paramétrés [54].

Donc, les propositions et implémentations actuelles de QRC n'ont pas comme but de résoudre les mêmes tâches que celles pour lesquelles les PQC ont été conçu. À savoir, leurs propositions principales pour des utilisations pratiques sont pour améliorer la détection quantique et la tomographie [28, 27, 55, 56].

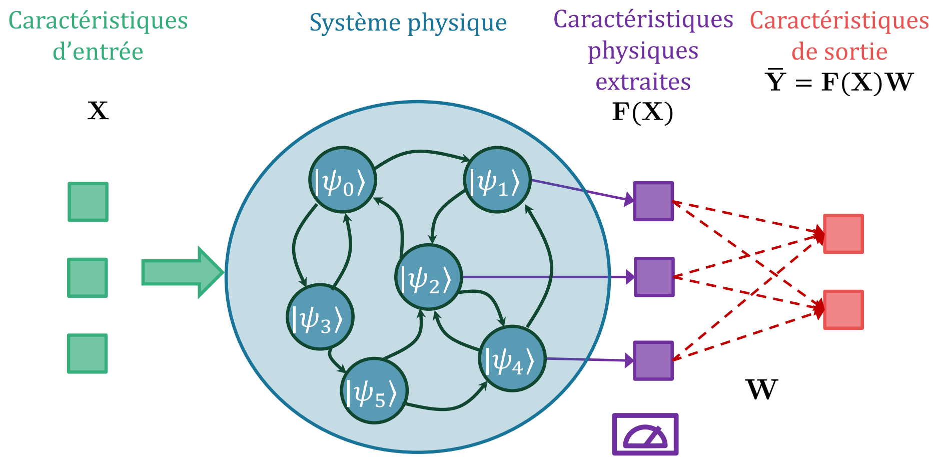


Figure 7.5: Schéma de l'algorithme de calcul par réservoir quantique. Le jeu de données en entrée $\mathbf{X}$ est encodé dans la dynamique du système quantique. Le système maintenant appelé réservoir évolue librement, jusqu'à ce que des observables soient mesurés pour obtenir un vecteur de caractéristiques physiques $\mathbf{F}(\mathbf{X})$. Des connections récurrentes entre les neurones du réservoir (les états de base $|\psi_k\rangle$ du système quantique) symbolisent la propriété de mémoire du réservoir, provenant de sa dynamique. La multiplication du vecteur de caractéristiques $\mathbf{F}(\mathbf{X})$ par une matrice $\mathbf{W}$ donne la prédiction $\bar{\mathbf{Y}}$. $\mathbf{W}$ est appris à travers une régression linéaire, tandis que les hyperparamètres du réservoir sont fixés et aléatoires.

7.3 Calcul par réservoir quantique avec les occupation d'états de Fock d'états bosoniques

Tandis qu'une partie de la communauté se concentre sur l'apprentissage avec des qubits et leurs portes logiques, dans mon travail de doctorat je me suis concentré sur les modes bosoniques. Un mode bosonique est un oscillateur harmonique quantique avec des statistiques bosoniques. Son Hamiltonien est

$$\hat{H} = \omega \hat{a}^\dagger \hat{a}, \tag{7.4}$$

où ω est sa fréquence de résonance, et $\hat{a}$ est l'opérateur de champ. Les états propres de ces oscillateurs quantiques sont les états de Fock notés $|n\rangle$, où n correspond à un nombre de photons. Nos motivations pour utiliser des modes bosoniques dans le calcul neuromorphique provient de leur plus grand nombre d'états de base par composant physique: tandis que les qubits sont des systèmes à deux niveaux, les modes bosoniques ont un nombre infini de niveaux d'énergie. En pratique, nous utiliserons environ 10 niveaux d'énergie par mode.

Des modes bosoniques peuvent être expérimentalement implémentés sur différents supports physiques, tels que des systèmes opto-mécaniques [57], des résonateurs optiques en anneau [58], et les circuits quantiques électro-dynamiques (cQED) [59]. Dans ce manuscrit nous allons considérer les modes électromagnétiques de résonateurs supraconducteurs [60], mais les résultats sont applicables à n'importe quel support physique implémentant des modes bosoniques.

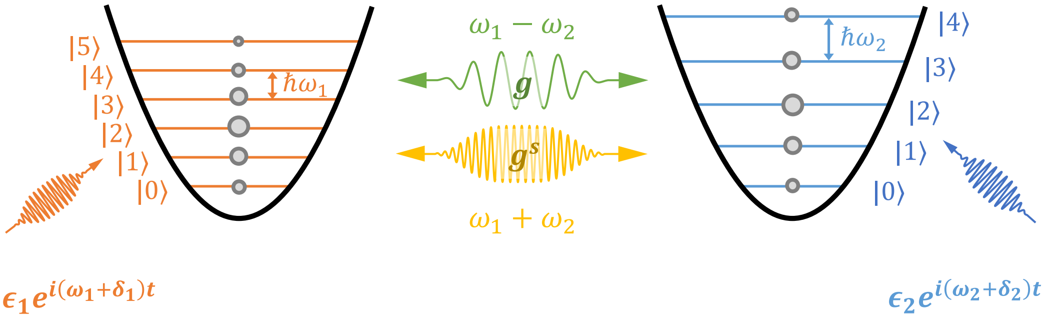


Figure 7.6: Schéma des niveaux d'énergies de deux modes bosoniques, avec des fréquences de résonance respectives ω_1 et ω_2. Nous choisissons d'appliquer à chaque mode des signaux proche de la résonance avec des amplitudes complexes $\epsilon_{1,2}$ et des désaccords de fréquence $\delta_{1,2}$ afin de créer des états cohérents (cercles gris). Les modes peuvent être couplé paramétriquement en introduisant un élément non-linéaire, tel qu'une jonction Josephson pour les circuits supraconducteurs [61, 62]. Nous considérons une mélange à trois ondes qui fournit deux couplages différents en fonction de la fréquence de pompe: de la conversion cohérente de photons et de la compression à deux modes, représentés respectivement par les flèches vertes et jaunes. Un signal de pompe appliqué à la différence de leurs fréquences crée de la conversion de photons à un taux g (flèche verte), tandis que l'appliquer à la somme de leurs fréquences crée une compression à deux modes à un taux g^s (flèche jaune).

Nous voulons implémenter un couplage accordable entre différents modes, afin de contrôler leur réponse dynamique et potentiellement les entraîner en tant que paramètres d'un modèle d'apprentissage. Dans ce but, les modes doivent d'abord être liés par un élément non-linéaire: dans le cas du cQED ceci peut être réalisé avec une jonction Josephson, mais en optique des cristaux non-linéaires peuvent aussi remplir ce rôle [63]. Nous choisissons dans ce manuscrit d'utiliser des signaux proche de la résonance afin d'injecter des photons dans chaque mode individuellement, et des signaux de pompe pour coupler différents modes à des taux dépendant de l'amplitude et la phase du signal de pompe. Figure 1.6 illustre le cas de deux modes couplés, où appliquer des signaux aux fréquences des résonateurs ω_1 et ω_2 permet de créer des états cohérents, et appliquer des pompes de mélange à trois ondes aux fréquences $\omega_1 - \omega_2$ ou $\omega_1 + \omega_2$ crée respectivement de la conversion de photon et de la compression à deux modes. La combinaison de ces phénomènes aboutit à des dynamiques accordables, que nous exploitons pour le calcul neuromorphique.

Cependant, les dynamiques de ces oscillateurs sont linéaires. Cela pose un obstacle à leur fonction de modèle d'apprentissage, qui nécessitent de la non-linéarité pour séparer des données. Pour résoudre ce problème, des travaux précédant mon doctorat ont proposé d'introduire de l'effet Kerr [64, 65], ou d'encoder les entrées dans les phases de pompes paramétriques [66]. Dans mon travail de thèse, nous nous sommes intéressés à la non-linéarité de la mesure. Nous avons considéré des mesures dans la base de Fock, ce qui nous permet d'extraire un grand nombre de caractéristiques. C'est particulièrement intéressant pour le calcul par réservoir où tout l'apprentissage est localisé sur la couche linéaire appliqué à ces caractéristiques. D'autres mesures non-linéaires peuvent être considérés, tel que la mesure de parité [67], qui a l'avantage de laisser le système dans une superposition d'états, mais fournit des mesures moins facile à interpréter. La première partie de mon doctorat fut dédiée à la question de si la mesure d'occupations d'états de Fock dans des modes bosoniques mènerait vers des avancées pour les systèmes neuromorphiques quantiques. L'architecture d'apprentissage automatique quantique que j'ai choisi fut le QRC, à cause de sa simplicité d'implémentation.

Dans la Section 4.3, j'ai montré que la non-linéarité de la mesure d'états de Fock est suffisant pour résoudre des tâches références nécessitant de la non-linéarité tel que la classification de signaux sinus et carrés. En outre, en comparent la performance de réservoirs classiques sur la même tâche, nous avons montré que pour la même puissance de calcul, un nombre plus petit de dispositifs physiques est requis dans le réservoir quantique, et un nombre plus petit de quantités physiques doit être mesuré. Finalement, nous observons que perdre la cohérence quantique baisse la performance du réservoir, ce qui suggère son rôle clé dans l'exploration complète de l'espace des caractéristiques.

Cependant, l'approche du QRC a ses limites. Premièrement, beaucoup d'observables doivent tout de même être mesurés pour avoir assez de caractéristiques de sortie sur lesquelles apprendre. Cela pose un obstacle à un implémentation expérimentale, dans laquelle mesurer même un unique observable est long à cause de la nature stochastique des mesures quantiques. Deuxièmement, apprendre une unique couche linéaire donne une expressivité limitée au réservoir; des contraintes similaires ont aussi été identifiées dans le calcul par réservoir classique [68]. Bien que dans le cas asymptotique, le théorème d'approximation universel [69] garantit qu'une profondeur ou largeur de couches de neurones suffisamment large permet à un réseau de neurones d'approximer n'importe quelle fonction cible, il a été observé que la profondeur joue en général un rôle plus conséquent que la largeur [70, 71]. À titre d'illustration, nous remarquons que les modèles d'apprentissage modernes tels que GPT-4 utilisent environ cent couches de neurones [72].

7.4 Réseau de neurones quantiques analogue avec des modes bosoniques Gaussiens

Dans ce contexte, j'ai passé la seconde partie de mon doctorat à développer un modèle numérique pour apprendre les paramètres du système quantique à travers la rétropropagation du gradient et la descente de gradient. Avant de commencer ce travail, la question que nous cherchions à résoudre était de savoir si entraîner les amplitudes, phases et désaccords de fréquence des couplages paramétriques serait suffisant pour apprendre des tâches, et améliorer la performance du QRC.

Comme la descente de gradient implique l'exécution de multiples passes en avant, cette méthode de simulation pour le système physique devrait être rapide à grande échelle. Cela peut être difficile pour les modes bosoniques d'intérêt, car l'espace de Hilbert exponentiellement grand des systèmes quantiques rend leur simulation classique complexe. L'état d'un système quantique ouvert interagissant faiblement avec un environnement Markovien [73] est modélisé par une matrice densité $\hat{\rho}$, obéissant à l'équation maîtresse de Lindblad

$$\begin{aligned}\frac{d\hat{\rho}}{dt} &= -\frac{i}{\hbar}[\hat{H}(t), \hat{\rho}(t)] + \sum_k \frac{1}{2}[2\hat{C}_k\hat{\rho}(t)\hat{C}_k^\dagger - \hat{\rho}(t)\hat{C}_k^\dagger\hat{C}_k - \hat{C}_k^\dagger\hat{C}_k\hat{\rho}(t)] \\ &= \mathcal{L}(\hat{\rho}),\end{aligned} \tag{7.5}$$

où $\hat{H}$ est l'Hamiltonien hermitien, et $\hat{C}_k$ sont des opérateurs de transition décrivant différents phénomènes de couplage avec l'environnement. $\mathcal{L}$ est un super-opérateur décrivant la dynamique.

Bien qu'il existe des librairies telles que `Dynamiqs` [74] et `torchdiffeq` [75] implémentant la rétropropagation de gradient sur des solveurs d'équations différentielles, le temps de calcul de cette méthode augmente exponentiellement avec le nombre de modes et d'états de base occupés. C'est parce que la matrice densité de M modes a une taille croissant exponentiellement avec $n_{\mathcal{H}}$, où $n_{\mathcal{H}}$ est la limite de l'espace de Hilbert.

En conséquence, pas plus que 2-3 modes peuvent être simulés efficacement sans des techniques d'approximation telles que les cumulants tronqués [55, 76] ou là représentation positive-P [77], qui sont toutes les deux peu fiables pour des temps d'évolution longs. Un de nos buts sera de traiter des jeux de données temporelles, donc le système quantique subira des évolutions dynamiques longues afin de se remémorer des entrées précédentes. Nous n'avons donc pas appliqués ces techniques d'approximation. Nous n'avons pas non plus appliqué des méthodes de réseaux de tenseurs [78, 79], car dans le contexte de ma thèse, nous nous sommes intéressés dans des états Gaussiens qui peuvent être simulé efficacement pour un large nombre de modes.

Un état gaussien a une distribution gaussienne dans l'espace des phases, comme

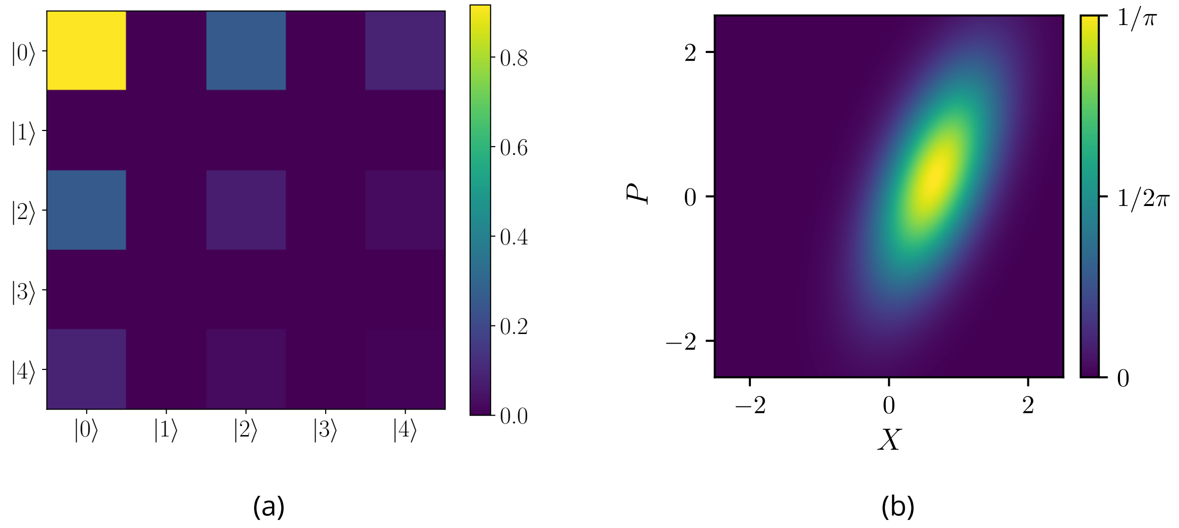


Figure 7.7: Deux représentations différentes d'un unique mode Gaussien comprimé, avec un paramètre de compression $r = 0.3(1 + i)$ (a) Représentation en matrice de densité. Les matrices densité sont à valeurs complexes, mais nous montrons leurs valeurs absolues $|\langle k| \hat{\rho} |l\rangle|$ par soucis de simplicité. L'espace de Hilbert est coupé à l'état de Fock $|4\rangle$, car les états supérieurs ont une amplitude de probabilité négligeable. (b) Fonction de Wigner dans l'espace des phases. Nous observons que sa distribution est gaussienne, ce qui veut dire dans ce cas que l'état est complètement caractérisé par son déplacement $\boldsymbol{\alpha}^R \in \mathbb{C}^2$ et sa matrice de covariance $\boldsymbol{\sigma}^R \in \mathbb{C}^{2\times 2}$.

montré en Fig. 1.7. Cette distribution peut être complètement caractérisée par le vecteur de déplacement $\boldsymbol{\alpha}^R \in \mathbb{C}^{2M}$ et la matrice de covariance $\boldsymbol{\sigma}^R \in \mathbb{C}^{2M\times 2M}$. Cela veut dire qu'un nombre polynomial de variables définit l'état, rendant sa simulation possible classiquement. En nous inspirant du travail de Ref. [80] pour efficacement simuler la dynamique de modes gaussiens, et de Ref. [81] pour analytiquement calculer les occupations d'états de Fock dans des distributions gaussiennes, j'ai développé une librairie Python permettant la simulation numérique efficace de la dynamique et des mesures en base de Fock de modes Gaussiens, avec la librairie `PyTorch`.

Un désavantage de cette méthode est que l'on ne peut pas simuler des termes non-linéaires dans l'Hamiltonien du système, comme ils créent des états non-Gaussiens. Cela restreint le nombre de signaux de pompe que nous voudrions introduire afin d'ajouter des couplages accordables: les seuls que nous pourront permettre seront la conversion cohérente de photon et la compression à deux modes. Donc le but de cette seconde partie de doctorat fut d'établir si il y a un avantage en terme de nombre d'observables mesurés et d'expressivité quand l'on apprend avec des modes bosoniques comparé au QRC, même avec un nombre quadratique de paramètres entraînés $\boldsymbol{\theta}$.

Dans ce but nous utilisons un modèle analogue, donc nous n'utilisons pas une multitude de portes logiques. Plutôt qu'appliquer un séquence d'évolutions Lindbladiennes

$\mathcal{L}_k$ chacunes avec un ensemble différent de paramètres $\boldsymbol{\theta}_k$, nous en appliquons une unique et exploitons la complexité de la dynamique. Les deux différentes approches sont illustrées en Fig. 7.8. Notre choix d'un algorithme analogue provient de sa simplicité relative: pour mettre en avant les avantages de l'apprentissage avec des modes bosoniques, nous n'avons pas besoin d'utiliser plusieurs portes logiques paramétrées.

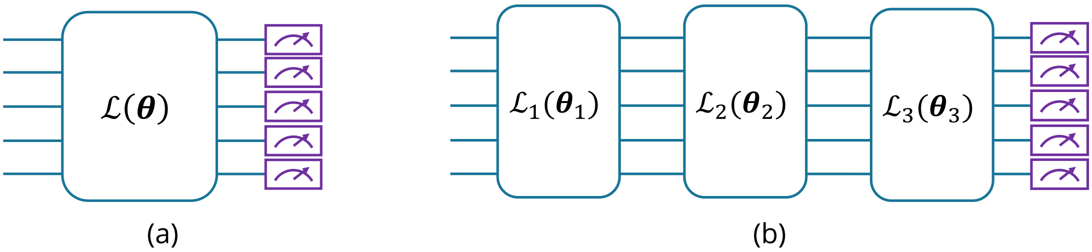

$\mathcal{L}(\boldsymbol{\theta})$
(a)
$\mathcal{L}_1(\boldsymbol{\theta}_1)$
$\mathcal{L}_2(\boldsymbol{\theta}_2)$
$\mathcal{L}_3(\boldsymbol{\theta}_3)$
(b)

Figure 7.8: Deux différentes approches en termes de circuits pour contrôler paramétriquement des modes bosoniques. Chaque ligne bleue correspond à un unique mode. (a) Calcul analogue, dans lequel tous les paramètres $\boldsymbol{\theta}$ de l'évolution Lindbladienne décrits par le super-opérateur $\mathcal{L}$ sont fixes. (b) Calcul basé sur des portes logiques avec trois portes dans cet exemple, où une séquence d'évolutions Lindbladiennes décrits par les super-opérateurs $\mathcal{L}_k$ ont chacun leur ensemble de paramètres $\boldsymbol{\theta}_k$. L'ensemble total des paramètres est donc $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k\in[1,3]}$.

Nous expliquons les avantages d'apprendre les paramètres physiques dans le Chapitre 5. Nous observons que l'apprentissage résout effectivement les limites du QRC, en le fait que moins d'observables doivent être mesurés pour résoudre des tâches, et l'expressivité plus haute permet au réseau de neurones quantiques de traiter des tâches bien plus dures que précédemment possible telle que la classification de chiffres DIGITS [82]. En outre, une étude du choix de paramètre d'encodage est présentée, où nous montrons qu'encoder dans les taux de compression à deux modes augmente la performance sur des tâches hautement non-linéaires.

Ces résultats démontrent que apprendre avec des modes bosoniques est une perspective prometteuse en calcul neuromorphique quantique. Grâce à leur large espace de Hilbert et aux couplages paramétriques, leur dynamique complexe peut être exploitée pour approximer des données non-triviales, et agir en tant que mémoire. Dans des recherches futures, utiliser les qubits en conjonction avec des modes bosoniques pourrait être utile pour créer des états non-Gaussien complexes à travers des protocoles de mesure. Aller d'algorithmes analogues vers des algorithmes basé sur des portes logiques permettra également d'élargir la diversité des tâches pouvant être résolues.

Dans ces travaux nous avons entraîné une simulation numérique de modes bosoniques couplés avec la rétropropagation du gradient, mais dans le cas d'une implémentation sur un support physique cet algorithme n'est pas accessible. Le calcul de gradients avec des règles de décalage de paramètres n'est pas une option non plus, comme aucun d'entre eux n'a été établi pour les modes bosoniques avec des mesures d'états de Fock. Nous pourrions considérer un apprentissage hors-ligne, dans lequel tout l'apprentissage est fait dans une simulation numérique du système physique. Les paramètres appris de cette façon peuvent être injecté dans le système physique pour l'inférence. Deux problèmes principaux surviennent dans cette approche: (i) augmenter l'échelle est complexe car simuler un grand système quantique est difficile, et (ii) des déviations inévitables de la simulation à l'expérience baissent son efficacité. Le deuxième problème peut être atténué en utilisant une algorithme d'apprentissage hybride développé par Ref. [83], où les passes en avant sont effectuées dans le système physique (même durant l'entraînement), mais sa simulation est utilisée pour calculer les gradients. Cette méthode est plus précise que l'apprentissage hors-ligne.

Quoi qu'il en soit, calculer des gradients avec la rétropropagation de gradients n'est pas possible à grand échelle, comme cela requiert une simulation classique du système, si les états du système sont non-Gaussiens. Pour entraîner un réseau de neurones bosoniques en expérience, il sera donc crucial d'explorer des méthodes d'entraînement compatible avec la physique, sur lesquelles la recherche est très active actuellement [84, 85, 86]. Explorer des modes non-gaussiens pour l'apprentissage automatique quantique permettra à la dynamique d'être plus complexe à simuler, ce qui pourrait se traduire par un avantage par rapport aux modèles d'apprentissage classiques [87]. Ajouter des non-linéarités d'ordre supérieur à deux pourra aussi fournir d'avantage de paramètres à entraîner.

Finalement, une recherche plus approfondie sur la susceptibilité aux plateaux désertiques des réseaux de neurones bosoniques devrait être poursuivie. À cause de la manière structurée avec laquelle l'espace de Hilbert est exploré durant leur dynamique, il est possible que l'information sur le gradient ne soit pas perdue lorsque la taille du système augmente.

Appendix A

Dispersive measurement of Fock states with a qubit in experiment

Experimentally, Fock state occupations in the bosonic modes would be measured by coupling a qubit to each mode, and using a dispersive readout. The dispersive coupling of a bosonic mode $\hat{a}$ to a qubit is modeled by the Hamiltonian [140]

$$\hat{H} = \hbar\omega\hat{a}^\dagger\hat{a} + \frac{\hbar\omega_q}{2}\hat{\sigma}_z - \hbar\chi\hat{a}^\dagger\hat{a}\hat{\sigma}_z, \tag{A.1}$$

where ω_q is the bare qubit frequency, and χ is a dispersive shift that makes the qubit resonance frequency depend on the number of photons in the mode. For instance, if the bosonic mode state is the Fock state $|n\rangle$, then partially tracing the bosonic mode system yields the qubit Hamiltonian

$$\begin{aligned}\hat{H}_q &= \mathrm{Tr}\Big(\hat{H}\,|n\rangle\!\langle n|\Big)\\ &= \hbar\omega n + \frac{\hbar\omega_q}{2}\hat{\sigma}_z - \hbar\chi n\hat{\sigma}_z\\ &= \hbar\omega n + \left(\frac{\hbar\omega_q}{2} - \hbar\chi n\right)\hat{\sigma}_z,\end{aligned} \tag{A.2}$$

and we see the qubit resonance frequency is shifted by a factor $\hbar\chi n$. Furthermore, it is possible to simultaneously measure multiple Fock states using frequency multiplexing [171].

Bibliography

[1] H. Sompolinsky, D. Golomb, and D. Kleinfeld. Global processing of visual stimuli in a neural network of coupled oscillators. *Proc Natl Acad Sci U S A*, 87(18):7200–7204, September 1990. ISSN 0027-8424. doi: 10.1073/pnas.87.18.7200.

[2] V. N. Murthy and E. E. Fetz. Coherent 25- to 35-Hz oscillations in the sensorimotor cortex of awake behaving monkeys. *Proc Natl Acad Sci U S A*, 89(12):5670–5674, June 1992. ISSN 0027-8424. doi: 10.1073/pnas.89.12.5670.

[3] Elena Leznik, Vladimir Makarenko, and Rodolfo Llinás. Electrotonically Mediated Oscillatory Patterns in Neuronal Ensembles: An *In Vitro* Voltage-Dependent Dye-Imaging Study in the Inferior Olive. *J. Neurosci.*, 22(7):2804–2815, April 2002. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.22-07-02804.2002. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.22-07-02804.2002.

[4] Antoine Lutz, Jean-Philippe Lachaux, Jacques Martinerie, and Francisco J. Varela. Guiding the study of brain dynamics by using first-person data: Synchrony patterns correlate with ongoing conscious states during a simple visual task. *Proc. Natl. Acad. Sci. U.S.A.*, 99(3):1586–1591, February 2002. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.032658199. URL https://pnas.org/doi/full/10.1073/pnas.032658199.

[5] Kai Malcolm and Josue Casco-Rodriguez. A Comprehensive Review of Spiking Neural Networks: Interpretation, Optimization, Efficiency, and Best Practices, March 2023. URL http://arxiv.org/abs/2303.10780.

[6] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, April 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

[7] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006. doi: 10.1126/science.1127647. URL https://www.science.org/doi/10.1126/science.1127647.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

[9] K. Worden, G. Tsialiamanis, E. J. Cross, and T. J. Rogers. Artificial Neural Networks. In Timon Rabczuk and Klaus-Jürgen Bathe, editors, *Machine Learning in Modeling and Simulation: Methods and Applications*, pages 85–119. Springer International Publishing, Cham, 2023. ISBN 978-3-031-36644-4. doi: 10.1007/978-3-031-36644-4_2. URL https://doi.org/10.1007/978-3-031-36644-4_2.

[10] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL https://www.nature.com/articles/323533a0.

[11] Robert A. Nawrocki, Richard M. Voyles, and Sean E. Shaheen. A Mini Review of Neuromorphic Architectures and Implementations. *IEEE Transactions on Electron Devices*, 63(10):3819–3829, October 2016. ISSN 1557-9646. doi: 10.1109/TED.2016.2598413. URL https://ieeexplore.ieee.org/document/7549034.

[12] Charlotte Debus, Marie Piraud, Achim Streit, Fabian Theis, and Markus Götz. Reporting electricity consumption is essential for sustainable AI. *Nat Mach Intell*, 5(11):1176–1178, November 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00750-1. URL https://www.nature.com/articles/s42256-023-00750-1.

[13] Queenie Wong. Amazon, Google tap into nuclear power to fuel data centers and AI push, October 2024. URL https://www.latimes.com/business/story/2024-10-16/amazon-google-tap-into-nuclear-power-to-fuel-data-centers-and-ai-push.

[14] Qiangfei Xia and J. Joshua Yang. Memristive crossbar arrays for brain-inspired computing. *Nat. Mater.*, 18(4):309–323, April 2019. ISSN 1476-4660. doi: 10.1038/s41563-019-0291-x. URL https://www.nature.com/articles/s41563-019-0291-x.

[15] Bhavin J. Shastri, Alexander N. Tait, T. Ferreira de Lima, Wolfram H. P. Pernice, Harish Bhaskaran, C. D. Wright, and Paul R. Prucnal. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photonics*, 15(2):102–114, February 2021. ISSN 1749-4893. doi: 10.1038/s41566-020-00754-y. URL https://www.nature.com/articles/s41566-020-00754-y.

[16] J. Grollier, D. Querlioz, K. Y. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles. Neuromorphic spintronics. *Nat Electron*, 3(7):360–370, July 2020. ISSN 2520-1131. doi: 10.1038/s41928-019-0360-9. URL https://www.nature.com/articles/s41928-019-0360-9.

[17] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Gi-

ambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024. URL http://arxiv.org/abs/2303.08774.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. URL http://arxiv.org/abs/1512.03385.

[19] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J Stat Phys*, 22(5):563–591, May 1980. ISSN 1572-9613. doi: 10.1007/BF01011339. URL https://link.springer.com/article/10.1007/BF01011339.

[20] Richard P. Feynman. Simulating physics with computers. *Int J Theor Phys*, 21(6): 467–488, June 1982. ISSN 1572-9575. doi: 10.1007/BF02650179. URL https://link.springer.com/article/10.1007/BF02650179.

[21] Serge Haroche and David J. Wineland. The Nobel Prize in Physics 2012, 2012. URL https://www.nobelprize.org/prizes/physics/2012/summary/.

[22] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509, October

1997. ISSN 0097-5397. doi: 10.1137/S0097539795293172. URL https://epubs.siam.org/doi/10.1137/S0097539795293172.

[23] Lov K. Grover. A fast quantum mechanical algorithm for database search, November 1996. URL http://arxiv.org/abs/quant-ph/9605043.

[24] Alberto Manzano, David Dechant, Jordi Tura, and Vedran Dunjko. Approximation and Generalization Capacities of Parametrized Quantum Circuits for Functions in Sobolev Spaces. *Quantum*, 9:1658, March 2025. doi: 10.22331/q-2025-03-10-1658. URL https://quantum-journal.org/papers/q-2025-03-10-1658/.

[25] Jinzhe Jiang, Yaqian Zhao, Rengang Li, Chen Li, Zhenhua Guo, Baoyu Fan, Xuelei Li, Ruyang Li, and Xin Zhang. Strong generalization in quantum neural networks. *Quantum Inf Process*, 22(12):1–12, December 2023. ISSN 1573-1332. doi: 10.1007/s11128-023-04095-x. URL https://link-springer-com.ezproxy.universite-paris-saclay.fr/article/10.1007/s11128-023-04095-x.

[26] Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J. Coles. Generalization in quantum machine learning from few training data. *Nat Commun*, 13(1):4919, August 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32550-3. URL https://www.nature.com/articles/s41467-022-32550-3.

[27] Gerasimos Angelatos, Saeed A. Khan, and Hakan E. Türeci. Reservoir Computing Approach to Quantum State Measurement. *Phys. Rev. X*, 11(4):041062, December 2021. doi: 10.1103/PhysRevX.11.041062. URL https://link.aps.org/doi/10.1103/PhysRevX.11.041062.

[28] Sanjib Ghosh, Andrzej Opala, Michał Matuszewski, Tomasz Paterek, and Timothy C. H. Liew. Quantum reservoir processing. *npj Quantum Inf*, 5(1):1–6, April 2019. ISSN 2056-6387. doi: 10.1038/s41534-019-0149-8. URL https://www.nature.com/articles/s41534-019-0149-8.

[29] Alessia Suprano, Danilo Zia, Luca Innocenti, Salvatore Lorenzo, Valeria Cimini, Taira Giordani, Ivan Palmisano, Emanuele Polino, Nicolò Spagnolo, Fabio Sciarrino, G. Massimo Palma, Alessandro Ferraro, and Mauro Paternostro. Experimental Property Reconstruction in a Photonic Quantum Extreme Learning Machine. *Phys. Rev. Lett.*, 132(16):160802, April 2024. doi: 10.1103/PhysRevLett.132.160802. URL https://link.aps.org/doi/10.1103/PhysRevLett.132.160802.

[30] Danilo Zia, Luca Innocenti, Giorgio Minati, Salvatore Lorenzo, Alessia Suprano, Rosario Di Bartolo, Nicolò Spagnolo, Taira Giordani, Valeria Cimini, G. Massimo

Palma, Alessandro Ferraro, Fabio Sciarrino, and Mauro Paternostro. Quantum extreme learning machines for photonic entanglement witnessing, February 2025. URL http://arxiv.org/abs/2502.18361.

[31] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, and Jarrod R. McClean. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, June 2022. doi: 10.1126/science.abn7293. URL https://www.science.org/doi/10.1126/science.abn7293.

[32] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.*, 4(4): 043001, November 2019. ISSN 2058-9565. doi: 10.1088/2058-9565/ab4eb5. URL https://dx.doi.org/10.1088/2058-9565/ab4eb5.

[33] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum Circuit Learning. *Phys. Rev. A*, 98(3):032309, September 2018. ISSN 2469-9926, 2469-9934. doi: 10.1103/PhysRevA.98.032309. URL http://arxiv.org/abs/1803.00745.

[34] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99(3):032331, March 2019. doi: 10.1103/PhysRevA.99.032331. URL https://link.aps.org/doi/10.1103/PhysRevA.99.032331.

[35] Xanadu. Parameter-shift rules — PennyLane, April 2025. URL https://pennylane.ai/qml/glossary/parameter_shift.

[36] Aram W. Harrow and John C. Napp. Low-Depth Gradient Measurements Can Improve Convergence in Variational Hybrid Quantum-Classical Algorithms. *Phys. Rev. Lett.*, 126(14):140502, April 2021. doi: 10.1103/PhysRevLett.126.140502. URL https://link.aps.org/doi/10.1103/PhysRevLett.126.140502.

[37] Martín Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo. Barren plateaus in variational quantum computing. *Nat Rev Phys*, 7(4):174–189, April 2025. ISSN 2522-5820. doi: 10.1038/s42254-025-00813-9. URL https://www.nature.com/articles/s42254-025-00813-9.

[38] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat Commun*, 12(1):1791, March 2021. ISSN 2041-1723.

doi: 10.1038/s41467-021-21728-w. URL https://www.nature.com/articles/s41467-021-21728-w.

[39] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nat Commun*, 12(1):6961, November 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-27045-6. URL https://www.nature.com/articles/s41467-021-27045-6.

[40] Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martín Larocca, and M. Cerezo. A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits. *Nat Commun*, 15 (1):7172, August 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-49909-3. URL https://www.nature.com/articles/s41467-024-49909-3.

[41] M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë Holmes. Does provable absence of barren plateaus imply classical simulability? *Nat Commun*, 16(1):7907, August 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-63099-6. URL https://www.nature.com/articles/s41467-025-63099-6.

[42] Keisuke Fujii and Kohei Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied*, 8(2), August 2017. ISSN 23317019. doi: 10.1103/PhysRevApplied.8.024030.

[43] Herbert Jaeger and Harald Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, April 2004. doi: 10.1126/science.1091277. URL https://www.science.org/doi/10.1126/science.1091277.

[44] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput*, 14(11):2531–2560, November 2002. ISSN 0899-7667. doi: 10.1162/089976602760407955.

[45] Chao Du, Fuxi Cai, Mohammed A. Zidan, Wen Ma, Seung Hwan Lee, and Wei D. Lu. Reservoir computing using dynamic memristors for temporal information processing. *Nat Commun*, 8(1):2204, December 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-02337-y. URL https://www.nature.com/articles/s41467-017-02337-y.

[46] John Moon, Wen Ma, Jong Hoon Shin, Fuxi Cai, Chao Du, Seung Hwan Lee, and Wei D. Lu. Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat Electron*, 2(10):480–487, October 2019. ISSN 2520-1131. doi: 10.1038/s41928-019-0313-3. URL https://www.nature.com/articles/s41928-019-0313-3.

[47] Yanan Zhong, Jianshi Tang, Xinyi Li, Bin Gao, He Qian, and Huaqiang Wu. Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat Commun*, 12(1):408, January 2021. ISSN 2041-1723. doi: 10.1038/s41467-020-20692-1. URL https://www.nature.com/articles/s41467-020-20692-1.

[48] Linfeng Sun, Zhongrui Wang, Jinbao Jiang, Yeji Kim, Bomin Joo, Shoujun Zheng, Seungyeon Lee, Woo Jong Yu, Bai-Sun Kong, and Heejun Yang. In-sensor reservoir computing for language learning via two-dimensional memristors. *Science Advances*, 7(20):eabg1455, May 2021. doi: 10.1126/sciadv.abg1455. URL https://www.science.org/doi/10.1126/sciadv.abg1455.

[49] Laurent Larger, Antonio Baylón-Fuentes, Romain Martinenghi, Vladimir S. Udaltsov, Yanne K. Chembo, and Maxime Jacquot. High-Speed Photonic Reservoir Computing Using a Time-Delay-Based Architecture: Million Words per Second Classification. *Phys. Rev. X*, 7(1):011015, February 2017. doi: 10.1103/PhysRevX.7.011015. URL https://link.aps.org/doi/10.1103/PhysRevX.7.011015.

[50] Mushegh Rafayelyan, Jonathan Dong, Yongqi Tan, Florent Krzakala, and Sylvain Gigan. Large-Scale Optical Reservoir Computing for Spatiotemporal Chaotic Systems Prediction. *Phys. Rev. X*, 10(4):041037, November 2020. doi: 10.1103/PhysRevX.10.041037. URL https://link.aps.org/doi/10.1103/PhysRevX.10.041037.

[51] Jacob Torrejon, Mathieu Riou, Flavio Abreu Araujo, Sumito Tsunegi, Guru Khalsa, Damien Querlioz, Paolo Bortolotti, Vincent Cros, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D. Stiles, and Julie Grollier. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664):428–431, July 2017. ISSN 1476-4687. doi: 10.1038/nature23011. URL https://www.nature.com/articles/nature23011.

[52] Miguel Romera, Philippe Talatchian, Sumito Tsunegi, Flavio Abreu Araujo, Vincent Cros, Paolo Bortolotti, Juan Trastoy, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Maxence Ernoult, Damir Vodenicarevic, Tifenn Hirtzlin, Nicolas

Locatelli, Damien Querlioz, and Julie Grollier. Vowel recognition with four coupled spin-torque nano-oscillators. *Nature*, 563(7730):230–234, November 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0632-y. URL https://www.nature.com/articles/s41586-018-0632-y.

[53] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the Expressive Power of Deep Neural Networks, June 2017. URL http://arxiv.org/abs/1606.05336.

[54] Nils-Erik Schütte, Niclas Götting, Hauke Müntinga, Meike List, Daniel Brunner, and Christopher Gies. Expressivity Limits of Quantum Reservoir Computing, May 2025. URL http://arxiv.org/abs/2501.15528.

[55] Saeed A. Khan, Fangjun Hu, Gerasimos Angelatos, Michael Hatridge, and Hakan E. Türeci. A neural processing approach to quantum state discrimination, September 2024. URL http://arxiv.org/abs/2409.03748.

[56] Saeed Ahmed Khan, Fangjun Hu, Gerasimos Angelatos, and Hakan E. Türeci. Physical reservoir computing using finitely-sampled quantum systems, November 2021. URL http://arxiv.org/abs/2110.13849.

[57] Pierre Meystre. Quantum Optomechanics. In Pierre Meystre, editor, *Quantum Optics: Taming the Quantum*, pages 325–364. Springer International Publishing, Cham, 2021. ISBN 978-3-030-76183-7. doi: 10.1007/978-3-030-76183-7_11. URL https://doi.org/10.1007/978-3-030-76183-7_11.

[58] R. J. C. Spreeuw and J. P. Woerdman. The driven optical ring resonator as a model system for quantum optics. *Physica B: Condensed Matter*, 175(1):96–110, December 1991. ISSN 0921-4526. doi: 10.1016/0921-4526(91)90698-E. URL https://www.sciencedirect.com/science/article/pii/092145269190698E.

[59] Alexandre Blais, Arne L. Grimsmo, S. M. Girvin, and Andreas Wallraff. Circuit Quantum Electrodynamics. *Rev. Mod. Phys.*, 93(2):025005, May 2021. ISSN 0034-6861, 1539-0756. doi: 10.1103/RevModPhys.93.025005. URL http://arxiv.org/abs/2005.12667.

[60] Philippe Campagne-Ibarcq. *Quantum Backaction and Feedback in Superconducting Circuits*. PhD thesis, Ecole normale supérieure - ENS PARIS, June 2015. URL https://theses.hal.science/tel-01726824.

[61] Danijela Marković. *Applications of the Josephson Mixer : Ultrastrong Coupling, Quantum Node and Injection Locking in Conversion*. PhD thesis, Université Paris sciences et lettres, December 2017. URL https://theses.hal.science/tel-01774258.

[62] B. Huard, H. Pothier, N.o. Birge, D. Esteve, X. Waintal, and J. Ankerhold. Josephson junctions as detectors for non-Gaussian noise. *Annalen der Physik*, 519(10-11): 736–750, 2007. ISSN 1521-3889. doi: 10.1002/andp.200751910-1107. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.200751910-1107.

[63] Grant M. Brodnik, Lindell M. Williams, Haixin Liu, David R. Carlson, Atasi Dan, Jennifer A. Black, and Scott B. Papp. Activating high-power parametric oscillation in photonic-crystal resonators, April 2025. URL http://arxiv.org/abs/2504.07947.

[64] L. C.G. Govia, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, and T. A. Ohki. Quantum reservoir computing with a single nonlinear oscillator. *Physical Review Research*, 3(1), January 2021. ISSN 26431564. doi: 10.1103/PhysRevResearch.3.013077.

[65] W. D. Kalfus, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, T. A. Ohki, and L. C. G. Govia. Hilbert space as a computational resource in reservoir computing. *Phys. Rev. Research*, 4(3):033007, July 2022. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.4.033007. URL https://link.aps.org/doi/10.1103/PhysRevResearch.4.033007.

[66] Johannes Nokkala, Rodrigo Martínez-Peña, Gian Luca Giorgi, Valentina Parigi, Miguel C. Soriano, and Roberta Zambrini. Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing. *Commun Phys*, 4(1):1–11, March 2021. ISSN 2399-3650. doi: 10.1038/s42005-021-00556-w. URL https://www.nature.com/articles/s42005-021-00556-w.

[67] Alen Senanian, Sridhar Prabhu, Vladimir Kremenetski, Saswata Roy, Yingkang Cao, Jeremy Kline, Tatsuhiro Onodera, Logan G. Wright, Xiaodi Wu, Valla Fatemi, and Peter L. McMahon. Microwave signal processing using an analog quantum reservoir computer. *Nat Commun*, 15(1):7490, August 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-51161-8. URL https://www.nature.com/articles/s41467-024-51161-8.

[68] Anthony M. Polloreno. Limits to Analog Reservoir Learning, April 2025. URL http://arxiv.org/abs/2307.14474.

[69] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

[70] Gal Vardi, Gilad Yehudai, and Ohad Shamir. Width is Less Important than Depth in ReLU Neural Networks, June 2022. URL http://arxiv.org/abs/2202.03841.

[71] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *J. Stat. Mech.*, 2020(11):113301, November 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc4de. URL http://arxiv.org/abs/1906.08034.

[72] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H. Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions, May 2023. URL http://arxiv.org/abs/2305.10435.

[73] The Theory of Open Quantum Systems. In Heinz-Peter Breuer and Francesco Petruccione, editors, *The Theory of Open Quantum Systems*, page 0. Oxford University Press, January 2007. ISBN 978-0-19-921390-0. doi: 10.1093/acprof:oso/9780199213900.002.14005. URL https://doi.org/10.1093/acprof:oso/9780199213900.002.14005.

[74] Dynamiqs/dynamiqs. dynamiqs, May 2025. URL https://github.com/dynamiqs/dynamiqs.

[75] Ricky T. Q. Chen. Torchdiffeq, June 2021. URL https://github.com/rtqichen/torchdiffeq.

[76] N. G Van kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, March 2007. ISBN 978-0-444-52965-7. URL https://shop.elsevier.com/books/stochastic-processes-in-physics-and-chemistry/van-kampen/978-0-444-52965-7.

[77] P. D. Drummond and C. W. Gardiner. Generalised P-representations in quantum optics. *J. Phys. A: Math. Gen.*, 13(7):2353, July 1980. ISSN 0305-4470. doi: 10.1088/0305-4470/13/7/018. URL https://dx.doi.org/10.1088/0305-4470/13/7/018.

[78] Roman Orus. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals of Physics*, 349:117–158, October 2014. ISSN 00034916. doi: 10.1016/j.aop.2014.06.013. URL http://arxiv.org/abs/1306.2164.

[79] Yuriel Núñez Fernández, Marc K. Ritter, Matthieu Jeannin, Jheng-Wei Li, Thomas Kloss, Thibaud Louvet, Satoshi Terasaki, Olivier Parcollet, Jan von Delft, Hiroshi Shinaoka, and Xavier Waintal. Learning tensor networks with tensor cross interpolation: New algorithms and libraries. *SciPost Physics*, 18(3):104, March 2025.

ISSN 2542-4653. doi: 10.21468/SciPostPhys.18.3.104. URL https://www.scipost.org/10.21468/SciPostPhys.18.3.104.

[80] Élie Gouzien. *Optique quantique multimode pour le traitement de l'information quantique*. PhD thesis, COMUE Université Côte d'Azur (2015 - 2019), December 2019. URL https://theses.hal.science/tel-02885960.

[81] Jacob F. F. Bulmer, Bryn A. Bell, Rachel S. Chadwick, Alex E. Jones, Diana Moise, Alessandro Rigazzi, Jan Thorbecke, Utz-Uwe Haus, Thomas Van Vaerenbergh, Raj B. Patel, Ian A. Walmsley, and Anthony Laing. The boundary for quantum advantage in Gaussian boson sampling. *Science Advances*, 8(4):eabl9236, January 2022. doi: 10.1126/sciadv.abl9236. URL https://www.science.org/doi/10.1126/sciadv.abl9236.

[82] Julien Dudas, Baptiste Carles, Elie Gouzien, Julie Grollier, and Danijela Marković. Training the parametric interactions in an analog bosonic quantum neural network with Fock basis measurement, April 2025. URL http://arxiv.org/abs/2411.19112.

[83] Logan G Wright, Tatsuhiro Onodera, Martin M Stein, Tianyu Wang, Darren T Schachter, Zoey Hu, and Peter L Mcmahon. Deep physical neural networks trained with backpropagation. *Nature*, 601:549, 2022. doi: 10.1038/s41586-021-04223-6. URL https://doi.org/10.1038/s41586-021-04223-6.

[84] Benjamin Scellier and Yoshua Bengio. Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation. *arxiv*, February 2016. URL http://arxiv.org/abs/1602.05179.

[85] Geoffrey Hinton. The Forward-Forward Algorithm: Some Preliminary Investigations, December 2022. URL http://arxiv.org/abs/2212.13345.

[86] Serge Massar and Bortolo Matteo Mognetti. Equilibrium Propagation: The Quantum and the Thermal Cases. *Quantum Stud.: Math. Found.*, 12(1):6, April 2025. ISSN 2196-5609, 2196-5617. doi: 10.1007/s40509-024-00351-6. URL http://arxiv.org/abs/2405.08467.

[87] Arsalan Motamedi, Hadi Zadeh-Haghighi, and Christoph Simon. Quantumness and Learning Performance in Reservoir Computing with a Single Oscillator, March 2024. URL http://arxiv.org/abs/2304.03462.

[88] Jeongho Bang, James Lim, M. S. Kim, and Jinhyoung Lee. Quantum Learning Machine, March 2008. URL http://arxiv.org/abs/0803.2976.

[89] Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, and Paolo Stornati. Dimensional Expressivity Analysis of Parametric Quantum Circuits. *Quantum*, 5: 422, March 2021. ISSN 2521-327X. doi: 10.22331/q-2021-03-29-422. URL http://arxiv.org/abs/2011.03532.

[90] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 1 edition, May 2014. ISBN 978-1-107-05713-5 978-1-107-29801-9. doi: 10.1017/CBO9781107298019. URL https://www.cambridge.org/core/product/identifier/9781107298019/type/book.

[91] Huang Tian-Long, Wu Yong-Zheng, Ni Ming, Wang Shi, and Ye Yong-Jin. Effects of quantum noise on Shor's algorithm. *Acta Phys. Sin.*, 73(5):050301, 2024. doi: 10.7498/aps.73.20231414.

[92] Jin-Yi Cai. Shor's Algorithm Does Not Factor Large Integers in the Presence of Noise. *Sci. China Inf. Sci.*, 67(7):173501, July 2024. ISSN 1674-733X, 1869-1919. doi: 10.1007/s11432-023-3961-3. URL http://arxiv.org/abs/2306.10072.

[93] Rajeev Acharya, Dmitry A. Abanin, Laleh Aghababaie-Beni, Igor Aleiner, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Nikita Astrakhantsev, Juan Atalaya, Ryan Babbush, Dave Bacon, Brian Ballard, Joseph C. Bardin, Johannes Bausch, Andreas Bengtsson, Alexander Bilmes, Sam Blackwell, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, David A. Browne, Brett Buchea, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Anthony Cabrera, Juan Campero, Hung-Shen Chang, Yu Chen, Zijun Chen, Ben Chiaro, Desmond Chik, Charina Chou, Jahan Claes, Agnetta Y. Cleland, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Sayan Das, Alex Davies, Laura De Lorenzo, Dripto M. Debroy, Sean Demura, Michel Devoret, Agustin Di Paolo, Paul Donohoe, Ilya Drozdov, Andrew Dunsworth, Clint Earle, Thomas Edlich, Alec Eickbusch, Aviv Moshe Elbag, Mahmoud Elzouka, Catherine Erickson, Lara Faoro, Edward Farhi, Vinicius S. Ferreira, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, Suhas Ganjam, Gonzalo Garcia, Robert Gasca, Élie Genois, William Giang, Craig Gidney, Dar Gilboa, Raja Gosula, Alejandro Grajales Dau, Dietrich Graumann, Alex Greene, Jonathan A. Gross, Steve Habegger, John Hall, Michael C. Hamilton, Monica Hansen, Matthew P. Harrigan, Sean D. Harrington, Francisco J. H. Heras, Stephen Heslin, Paula Heu, Oscar Higgott, Gordon Hill, Jeremy Hilton, George Holland, Sabrina Hong, Hsin-Yuan Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Stephen Jordan, Chaitali Joshi, Pavol Juhas, Dvir Kafri, Hui Kang, Amir H. Karamlou, Kostyantyn Kechedzhi, Julian Kelly,

Trupti Khaire, Tanuj Khattar, Mostafa Khezri, Seon Kim, Paul V. Klimov, Andrey R. Klots, Bryce Kobrin, Pushmeet Kohli, Alexander N. Korotkov, Fedor Kostritsa, Robin Kothari, Borislav Kozlovskii, John Mark Kreikebaum, Vladislav D. Kurilovich, Nathan Lacroix, David Landhuis, Tiano Lange-Dei, Brandon W. Langley, Pavel Laptev, Kim-Ming Lau, Loïck Le Guevel, Justin Ledford, Joonho Lee, Kenny Lee, Yuri D. Lensky, Shannon Leon, Brian J. Lester, Wing Yan Li, Yin Li, Alexander T. Lill, Wayne Liu, William P. Livingston, Aditya Locharla, Erik Lucero, Daniel Lundahl, Aaron Lunt, Sid Madhuk, Fionn D. Malone, Ashley Maloney, Salvatore Mandrà, James Manyika, Leigh S. Martin, Orion Martin, Steven Martin, Cameron Maxfield, Jarrod R. McClean, Matt McEwen, Seneca Meeks, Anthony Megrant, Xiao Mi, Kevin C. Miao, Amanda Mieszala, Reza Molavi, Sebastian Molina, Shirin Montazeri, Alexis Morvan, Ramis Movassagh, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Chia-Hung Ni, Murphy Yuezhen Niu, Thomas E. O'Brien, William D. Oliver, Alex Opremcak, Kristoffer Ottosson, Andre Petukhov, Alex Pizzuto, John Platt, Rebecca Potter, Orion Pritchard, Leonid P. Pryadko, Chris Quintana, Ganesh Ramachandran, Matthew J. Reagor, John Redding, David M. Rhodes, Gabrielle Roberts, Eliott Rosenberg, Emma Rosenfeld, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Andrew W. Senior, Michael J. Shearn, Aaron Shorter, Noah Shutty, Vladimir Shvarts, Shraddha Singh, Volodymyr Sivak, Jindra Skruzny, Spencer Small, Vadim Smelyanskiy, W. Clarke Smith, Rolando D. Somma, Sofia Springer, George Sterling, Doug Strain, Jordan Suchard, Aaron Szasz, Alex Sztein, Douglas Thor, Alfredo Torres, M. Mert Torunbalci, Abeer Vaishnav, Justin Vargas, Sergey Vdovichev, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Steven Waltman, Shannon X. Wang, Brayden Ware, Kate Weber, Travis Weidel, Theodore White, Kristi Wong, Bryan W. K. Woo, Cheng Xing, Z. Jamie Yao, Ping Yeh, Bicheng Ying, Juhwan Yoo, Noureldin Yosri, Grayson Young, Adam Zalcman, Yaxing Zhang, Ningfeng Zhu, Nicholas Zobrist, and Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, February 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08449-y. URL https://www.nature.com/articles/s41586-024-08449-y.

[94] Joschka Roffe. Quantum Error Correction: An Introductory Guide. *Contemporary Physics*, 60(3):226–245, July 2019. ISSN 0010-7514, 1366-5812. doi: 10.1080/00107514.2019.1667078. URL http://arxiv.org/abs/1907.11157.

[95] P. Chandra and Y. Singh. Feedforward sigmoidal networks - equicontinuity and fault-tolerance properties. *IEEE*, 15(6):1350–1366, August 2024.

doi: 10.1109/TNN.2004.831198. URL https://ieeexplore-ieee-org.ezproxy.universite-paris-saclay.fr/document/1353274.

[96] K. G. Kapanova, I. Dimov, and J. M. Sellier. A neural network sensitivity analysis in the presence of random fluctuations. *Neurocomputing*, 224:177–183, February 2017. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.10.060. URL https://www.sciencedirect.com/science/article/pii/S0925231216313030.

[97] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing Deep Neural Networks by Noise: Its Interpretation and Optimization, November 2017. URL http://arxiv.org/abs/1710.05179.

[98] Diego García-Martín, Martín Larocca, and M. Cerezo. Effects of noise on the overparametrization of quantum neural networks. *Phys. Rev. Res.*, 6(1):013295, March 2024. doi: 10.1103/PhysRevResearch.6.013295. URL https://link.aps.org/doi/10.1103/PhysRevResearch.6.013295.

[99] 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), October 2019. ISSN 26431564. doi: 10.1103/PhysRevResearch.1.033063.

[100] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry Fields: A Software Platform for Photonic Quantum Computing. *Quantum*, 3:129, March 2019. doi: 10.22331/q-2019-03-11-129. URL https://quantum-journal.org/papers/q-2019-03-11-129/.

[101] Yutaro Enomoto, Keitaro Anai, Kenta Udagawa, and Shuntaro Takeda. Continuous-variable quantum approximate optimization on a programmable photonic quantum processor. *Phys. Rev. Res.*, 5(4):043005, October 2023. doi: 10.1103/PhysRevResearch.5.043005. URL https://link.aps.org/doi/10.1103/PhysRevResearch.5.043005.

[102] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, November 2022. ISSN 03701573. doi: 10.1016/j.physrep.2022.08.003. URL http://arxiv.org/abs/2111.05176.

[103] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113(13):130503, September 2014. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.113.130503. URL http://arxiv.org/abs/1307.0471.

[104] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A Review on Quantum Approximate Optimization Algorithm and its Variants. *Physics Reports*, 1068:1–66, June 2024. ISSN 03701573. doi: 10.1016/j.physrep.2024.03.002. URL http://arxiv.org/abs/2306.09198.

[105] Feng Zhang, Niladri Gomes, Noah F. Berthusen, Peter P. Orth, Cai-Zhuang Wang, Kai-Ming Ho, and Yong-Xin Yao. Shallow-circuit variational quantum eigensolver based on symmetry-inspired Hilbert space partitioning for quantum chemical calculations. *Phys. Rev. Res.*, 3(1):013039, January 2021. doi: 10.1103/PhysRevResearch.3.013039. URL https://link.aps.org/doi/10.1103/PhysRevResearch.3.013039.

[106] Jan Lukas Bosse and Ashley Montanaro. Probing ground-state properties of the kagome antiferromagnetic Heisenberg model using the variational quantum eigensolver. *Phys. Rev. B*, 105(9):094409, March 2022. doi: 10.1103/PhysRevB.105.094409. URL https://link.aps.org/doi/10.1103/PhysRevB.105.094409.

[107] Teppei Suzuki, Takashi Hasebe, and Tsubasa Miyazaki. Quantum support vector machines for classification and regression on a trapped-ion quantum computer. *Quantum Mach. Intell.*, 6(1):31, May 2024. ISSN 2524-4914. doi: 10.1007/s42484-024-00165-0. URL https://doi.org/10.1007/s42484-024-00165-0.

[108] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm, November 2014. URL http://arxiv.org/abs/1411.4028.

[109] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, July 2011. ISSN 1532-4435.

[110] T. Tieleman. Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude, 2012. URL https://cir.nii.ac.jp/crid/1370017282431050757.

[111] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980.

[112] Pere Mujal, Rodrigo Martínez-Peña, Gian Luca Giorgi, Miguel C. Soriano, and Roberta Zambrini. Time-series quantum reservoir computing with weak and projective measurements. *npj Quantum Inf*, 9(1):1–10, February 2023. ISSN 2056-6387. doi: 10.1038/s41534-023-00682-z. URL https://www.nature.com/articles/s41534-023-00682-z.

[113] Tomoya Monomi, Wataru Setoyama, and Yoshihiko Hasegawa. Feedback-enhanced quantum reservoir computing with weak measurements, March 2025. URL http://arxiv.org/abs/2503.17939.

[114] Chuanzhou Zhu, Peter J. Ehlers, Hendra I. Nurdin, and Daniel Soh. Practical few-atom quantum reservoir computing. *Phys. Rev. Res.*, 7(2):023290, June 2025. doi: 10.1103/wsyq-jyxd. URL https://link.aps.org/doi/10.1103/wsyq-jyxd.

[115] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Phys. Rev. A*, 92(4):042303, October 2015. doi: 10.1103/PhysRevA.92.042303. URL https://link.aps.org/doi/10.1103/PhysRevA.92.042303.

[116] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671): 242–246, September 2017. ISSN 1476-4687. doi: 10.1038/nature23879. URL https://www.nature.com/articles/nature23879.

[117] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-Induced Barren Plateaus. *PRX Quantum*, 2(4):040316, October 2021. doi: 10.1103/PRXQuantum.2.040316. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.040316.

[118] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, March 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

[119] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, March 2015. URL http://arxiv.org/abs/1502.03167.

[120] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization, July 2016. URL http://arxiv.org/abs/1607.06450.

[121] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, Madison, WI, USA, June 2010. Omnipress. ISBN 978-1-60558-907-7.

[122] Stefan H. Sack, Raimel A. Medina, Alexios A. Michailidis, Richard Kueng, and Maksym Serbyn. Avoiding Barren Plateaus Using Classical Shadows. *PRX Quantum*, 3(2):020365, June 2022. doi: 10.1103/PRXQuantum.3.020365. URL https://link.aps.org/doi/10.1103/PRXQuantum.3.020365.

[123] Michael I. Jordan. Chapter 25 - Serial Order: A Parallel Distributed Processing Approach. In John W. Donahoe and Vivian Packard Dorsel, editors, *Advances in Psychology*, volume 121 of *Neural-Network Models of Cognition*, pages 471–495. North-Holland, January 1997. doi: 10.1016/S0166-4115(97)80111-2. URL https://www.sciencedirect.com/science/article/pii/S0166411597801112.

[124] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning, October 2015. URL http://arxiv.org/abs/1506.00019.

[125] Zhekang Dong, Chun Sing Lai, Zhaowei Zhang, Donglian Qi, Mingyu Gao, and Shukai Duan. Neuromorphic extreme learning machines with bimodal memristive synapses. *Neurocomputing*, 453:38–49, September 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.04.049. URL https://www.sciencedirect.com/science/article/pii/S0925231221005919.

[126] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[127] George Philipp, Dawn Song, and Jaime G. Carbonell. The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions, April 2018. URL http://arxiv.org/abs/1712.05577.

[128] R. Penrose. On best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52(1):17–19, January 1956. ISSN 1469-8064, 0305-0041. doi: 10.1017/S0305004100030929. URL https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/abs/on-best-approximate-solutions-of-linear-matrix-equations/FDB8762BB13CB93E9E70080B2CFD986A.

[129] S. Bompas, B. Georgeot, and D. Guéry-Odelin. Accuracy of neural networks for the simulation of chaotic dynamics: Precision of training data vs precision of the algorithm. *Chaos*, 30(11):113118, November 2020. ISSN 1089-7682. doi: 10.1063/5.0021264.

[130] Erik Bollt. On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(1):013108, January 2021. ISSN 1054-1500. doi: 10.1063/5.0024890. URL https://doi.org/10.1063/5.0024890.

[131] Pantelis R. Vlachas, Jaideep Pathak, Brian R. Hunt, Themistoklis P. Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Forecasting of Spatio-temporal Chaotic Dynamics with Recurrent Neural Networks: A comparative study of Reservoir Computing and Backpropagation Algorithms, October 2019. URL http://arxiv.org/abs/1910.05266.

[132] Muhammad AbuGhanem. IBM quantum computers: Evolution, performance, and future directions. *J Supercomput*, 81(5):687, April 2025. ISSN 1573-0484. doi: 10.1007/s11227-025-07047-7. URL https://doi.org/10.1007/s11227-025-07047-7.

[133] Zoe Budrikis. 100 years of the Ising model. *Nat Rev Phys*, 6(9):530–530, September 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00760-x. URL https://www.nature.com/articles/s42254-024-00760-x.

[134] Rodrigo Martínez-Peña, Gian Luca Giorgi, Johannes Nokkala, Miguel C. Soriano, and Roberta Zambrini. Dynamical Phase Transitions in Quantum Reservoir Computing. *Phys. Rev. Lett.*, 127(10):100502, August 2021. doi: 10.1103/PhysRevLett.127.100502. URL https://link.aps.org/doi/10.1103/PhysRevLett.127.100502.

[135] Makoto Negoro, Kosuke Mitarai, Keisuke Fujii, Kohei Nakajima, and Masahiro Kitagawa. Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid, June 2018. URL http://arxiv.org/abs/1806.10910.

[136] Makoto Negoro, Kosuke Mitarai, Kohei Nakajima, and Keisuke Fujii. Toward NMR Quantum Reservoir Computing. In *Reservoir Computing*, pages 451–458. Springer, Singapore, 2021. ISBN 978-981-13-1687-6. doi: 10.1007/978-981-13-1687-6_19. URL https://link.springer.com/chapter/10.1007/978-981-13-1687-6_19.

[137] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, May

2011. ISSN 1476-4687. doi: 10.1038/nature10012. URL https://www.nature.com/articles/nature10012.

[138] G. G. Guerreschi and A. Y. Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Sci Rep*, 9(1):6903, May 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-43176-9. URL https://www.nature.com/articles/s41598-019-43176-9.

[139] Antonio Sannia, Rodrigo Martínez-Peña, Miguel C. Soriano, Gian Luca Giorgi, and Roberta Zambrini. Dissipation as a resource for Quantum Reservoir Computing. *Quantum*, 8:1291, March 2024. doi: 10.22331/q-2024-03-20-1291. URL https://quantum-journal.org/papers/q-2024-03-20-1291/.

[140] Baptiste Carles, Julien Dudas, Léo Balembois, Julie Grollier, and Danijela Marković. Experimental quantum reservoir computing with a circuit quantum electrodynamics system, June 2025. URL http://arxiv.org/abs/2506.22016.

[141] Howard M. Wiseman and Gerard J. Milburn. *Quantum Measurement and Control*. Cambridge University Press, Cambridge, 2009. ISBN 978-0-521-80442-4. doi: 10.1017/CBO9780511813948. URL https://www.cambridge.org/core/books/quantum-measurement-and-control/F78F445CD9AF00B10593405E9BAC6B9F.

[142] Rusko Ruskov, Alexander N. Korotkov, and Klaus Mølmer. Qubit State Monitoring by Measurement of Three Complementary Observables. *Phys. Rev. Lett.*, 105(10):100506, September 2010. doi: 10.1103/PhysRevLett.105.100506. URL https://link.aps.org/doi/10.1103/PhysRevLett.105.100506.

[143] Milan Kornjača, Hong-Ye Hu, Chen Zhao, Jonathan Wurtz, Phillip Weinberg, Majd Hamdan, Andrii Zhdanov, Sergio H. Cantu, Hengyun Zhou, Rodrigo Araiza Bravo, Kevin Bagnall, James I. Basham, Joseph Campo, Adam Choukri, Robert DeAngelo, Paige Frederick, David Haines, Julian Hammett, Ning Hsu, Ming-Guang Hu, Florian Huber, Paul Niklas Jepsen, Ningyuan Jia, Thomas Karolyshyn, Minho Kwon, John Long, Jonathan Lopatin, Alexander Lukin, Tommaso Macrì, Ognjen Marković, Luis A. Martínez-Martínez, Xianmei Meng, Evgeny Ostroumov, David Paquette, John Robinson, Pedro Sales Rodriguez, Anshuman Singh, Nandan Sinha, Henry Thoreen, Noel Wan, Daniel Waxman-Lenz, Tak Wong, Kai-Hsin Wu, Pedro L. S. Lopes, Yuval Boger, Nathan Gemelke, Takuya Kitagawa, Alexander Keesling, Xun Gao, Alexei Bylinskii, Susanne F. Yelin, Fangli Liu, and Sheng-Tao Wang. Large-scale quantum reservoir learning with an analog quantum computer, July 2024. URL http://arxiv.org/abs/2407.02553.

[144] Masoud Mohseni, Artur Scherer, K. Grace Johnson, Oded Wertheim, Matthew Otten, Navid Anjum Aadit, Kirk M. Bresniker, Kerem Y. Camsari, Barbara Chapman, Soumitra Chatterjee, Gebremedhin A. Dagnew, Aniello Esposito, Farah Fahim, Marco Fiorentino, Abdullah Khalid, Xiangzhou Kong, Bohdan Kulchytskyy, Ruoyu Li, P. Aaron Lott, Igor L. Markov, Robert F. McDermott, Giacomo Pedretti, Archit Gajjar, Allyson Silva, John Sorebo, Panagiotis Spentzouris, Ziv Steiner, Boyan Torosov, Davide Venturelli, Robert J. Visser, Zak Webb, Xin Zhan, Yonatan Cohen, Pooya Ronagh, Alan Ho, Raymond G. Beausoleil, and John M. Martinis. How to Build a Quantum Supercomputer: Scaling Challenges and Opportunities, November 2024. URL http://arxiv.org/abs/2411.10406.

[145] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nat Comput Sci*, 2(9):567–576, September 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00311-3. URL https://www.nature.com/articles/s43588-022-00311-3.

[146] Julien Dudas, Baptiste Carles, Erwan Plouet, Frank Alice Mizrahi, Julie Grollier, and Danijela Marković. Quantum reservoir computing implementation on coherently coupled quantum oscillators. *npj Quantum Inf*, 9(1):1–7, July 2023. ISSN 2056-6387. doi: 10.1038/s41534-023-00734-4. URL https://www.nature.com/articles/s41534-023-00734-4.

[147] Huawen Xu, Tanjung Krisnanda, Ruiqi Bao, and Timothy C H Liew. The roles of Kerr nonlinearity in a bosonic quantum neural network. *New J. Phys.*, 25(2):023028, February 2023. ISSN 1367-2630. doi: 10.1088/1367-2630/acbc43. URL https://dx.doi.org/10.1088/1367-2630/acbc43.

[148] Yuichiro Mori, · Kouhei Nakaji, Yuichiro Matsuzaki, and Shiro Kawabata. Expressive quantum supervised machine learning using Kerr-nonlinear parametric oscillators. 6:14, 2024. doi: 10.1007/s42484-024-00152-5. URL https://doi.org/10.1007/s42484-024-00152-5.

[149] Chang-Woo Lee and Hyunseok Jeong. Quantification of Macroscopic Quantum Superpositions within Phase Space. *Phys. Rev. Lett.*, 106(22):220401, May 2011. doi: 10.1103/PhysRevLett.106.220401. URL https://link.aps.org/doi/10.1103/PhysRevLett.106.220401.

[150] Abhinav Kala, David Sharp, Minho Choi, Arnab Manna, Prathmesh Deshmukh, Vijin Kizhake Veetil, Vinod Menon, Matthew Pelton, Edo Waks, and Arka Majumdar. Opportunities and Challenges of Solid-State Quantum Nonlinear Optics, November 2024. URL http://arxiv.org/abs/2411.06630.

[151] Darrick E. Chang, Vladan Vuletić, and Mikhail D. Lukin. Quantum nonlinear optics — photon by photon. *Nature Photon*, 8(9):685–694, September 2014. ISSN 1749-4893. doi: 10.1038/nphoton.2014.192. URL https://www.nature.com/articles/nphoton.2014.192.

[152] Jorge García-Beni, Iris Paparelle, Valentina Parigi, Gian Luca Giorgi, Miguel C. Soriano, and Roberta Zambrini. Quantum machine learning via continuous-variable cluster states and teleportation. *EPJ Quantum Technol.*, 12(1):1–14, December 2025. ISSN 2196-0763. doi: 10.1140/epjqt/s40507-025-00352-3. URL https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-025-00352-3.

[153] H. J. Carmichael. Quantum trajectory theory for cascaded open systems. *Phys. Rev. Lett.*, 70(15):2273–2276, April 1993. doi: 10.1103/PhysRevLett.70.2273. URL https://link.aps.org/doi/10.1103/PhysRevLett.70.2273.

[154] Wikipedia. Central limit theorem. *Wikipedia*, April 2025. URL https://en.wikipedia.org/w/index.php?title=Central_limit_theorem&oldid=1287814513#History.

[155] Xing Chen, Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Dafiné Ravelosona, Wang Kang, Weisheng Zhao, Julie Grollier, and Damien Querlioz. Forecasting the outcome of spintronic experiments with Neural Ordinary Differential Equations. *Nat Commun*, 13(1):1016, February 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-28571-7. URL https://www.nature.com/articles/s41467-022-28571-7.

[156] Anja Metelmann. Parametric couplings in engineered quantum systems. *SciPost Physics Lecture Notes*, page 066, March 2023. ISSN 2590-1990. doi: 10.21468/SciPostPhysLectNotes.66. URL https://scipost.org/10.21468/SciPostPhysLectNotes.66.

[157] Ali Anwar, Chithrabhanu Perumangatt, Fabian Steinlechner, Thomas Jennewein, and Alexander Ling. Entangled Photon-Pair Sources based on three-wave mixing in bulk crystals. *Review of Scientific Instruments*, 92(4):041101, April 2021. ISSN 0034-6748, 1089-7623. doi: 10.1063/5.0023103. URL http://arxiv.org/abs/2007.15364.

[158] C W Gardiner and M J Collett. Input and output in damped quantum systems: Quantum stochastic differential equations and the master equation. Technical Report 6, 1985.

[159] Dirac delta function. *Wikipedia*, May 2025. URL https://en.wikipedia.org/w/index.php?title=Dirac_delta_function&oldid=1290332476.

[160] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. Gaussian quantum information. *Rev. Mod. Phys.*, 84(2):621–669, May 2012. doi: 10.1103/RevModPhys.84.621. URL https://link.aps.org/doi/10.1103/RevModPhys.84.621.

[161] Jonatan Bohr Brask. Gaussian states and operations – a quick reference, March 2022. URL http://arxiv.org/abs/2102.05748.

[162] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3):637–657, January 1962. ISSN 0041-5553. doi: 10.1016/0041-5553(63)90168-X. URL https://www.sciencedirect.com/science/article/pii/004155536390168X.

[163] Gerardo Adesso, Sammy Ragy, and Antony R. Lee. Continuous Variable Quantum Information: Gaussian States and Beyond. *Open Syst. Inf. Dyn.*, 21(01n02):1440001, June 2014. ISSN 1230-1612. doi: 10.1142/S1230161214400010. URL https://www.worldscientific.com/doi/abs/10.1142/S1230161214400010.

[164] Andreas Björklund, Brajesh Gupt, and Nicolás Quesada. A faster hafnian formula for complex matrices and its benchmarking on a supercomputer, May 2019. URL http://arxiv.org/abs/1805.12498.

[165] HaiJie Li, YiWen Wang, LianFu Wei, PinJia Zhou, Qiang Wei, ChunHai Cao, YuRong Fang, Yang Yu, and PeiHeng Wu. Experimental demonstrations of high-Q superconducting coplanar waveguide resonators. *Chin. Sci. Bull.*, 58(20):2413–2417, July 2013. ISSN 1861-9541. doi: 10.1007/s11434-013-5882-3. URL https://doi.org/10.1007/s11434-013-5882-3.

[166] Torch.linalg.eig — PyTorch 2.6 documentation. URL https://pytorch.org/docs/stable/generated/torch.linalg.eig.html.

[167] Load_digits, 2017. URL https://scikit-learn/stable/modules/generated/sklearn.datasets.load_digits.html.

[168] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, February 2020. doi: 10.22331/q-2020-02-06-226. URL https://quantum-journal.org/papers/q-2020-02-06-226/.

[169] CrossEntropyLoss — PyTorch 2.7 documentation. URL https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[170] Benjamin Scellier. Quantum Equilibrium Propagation: Gradient-Descent Training of Quantum Systems, June 2024. URL http://arxiv.org/abs/2406.00879.

[171] Antoine Essig, Quentin Ficheux, Théau Peronnin, Nathanaël Cottet, Raphaël Lescanne, Alain Sarlette, Pierre Rouchon, Zaki Leghtas, and Benjamin Huard. Multiplexed Photon Number Measurement. *Phys. Rev. X*, 11(3):031045, August 2021. doi: 10.1103/PhysRevX.11.031045. URL https://link.aps.org/doi/10.1103/PhysRevX.11.031045.