3-24-2023

Realization of Multi-Valued Logic Using Optical Quantum Computing

Sophie Choe
Portland State University

Recommended Citation
Choe, Sophie, "Realization of Multi-Valued Logic Using Optical Quantum Computing" (2023). *Dissertations and Theses.* Paper 6338.
https://doi.org/10.15760/etd.8192

Realization of Multi-Valued Logic Using Optical Quantum Computing

by

Sophie Choe

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:
Marek Perkowski, Chair
John Acken
Xiaoyu Song
Steven Bleiler

Portland State University
2023

Abstract

Quantum computing is a paradigm of computing using physical systems, which operate according to quantum mechanical principles. Since 2017, functioning quantum processing units with limited capabilities are available on the cloud. There are two models of quantum computing in the literature: discrete variable and continuous variable models. The discrete variable model is an extension of the binary logic of digital computing with quantum bits $|0\rangle$ and $|1\rangle$. In the continuous variable model, the quantum state space is infinite-dimensional and the quantum state is expressed with an infinite number of basis elements.

In the physical implementation of quantum computing, however, the quantized energy levels of the electromagnetic field come in multiple values, naturally realizing the multi-valued logic of computing. Hence, to implement the discrete variable model (binary logic) of quantum computing, the temperature control is needed to restrict the energy levels to the lowest two to express the binary quantum states $|0\rangle$ and $|1\rangle$. The physical realization of the continuous variable model naturally implements the multi-valued logic of computing because any physical system always has the highest level of quantized energy observed i.e., the quantum state space is always finite dimensional.

In 2001, Knill, Laflamme, and Milburn proved that linear optics realizes universal quantum computing in the qubit-based model. Optical quantum computers by Xanadu, under the phase space representation of quantum optics, naturally realizes the multi-valued logic of quantum computing at room temperature. Optical quantum computers use optical signals, which are most compatible with the fiber optics communication network. They are easily fabricable for mass production, robust to noise, and have low latency.

Optical quantum computing provides flexibility to the users for determining the

dimension of the computational space for each instance of computation. Additionally, nonlinear quantum optical effects are incorporated as nonlinear quantum gates. That flexibility of user-defined dimension of the computational space and availability of nonlinear gates lead to a faithful implementation of quantum neural networks in optical quantum computing. This dissertation provides a full description of a multi-class data quantum classifier on ten classes of the MNIST dataset.

In this dissertation, I provide the background information of optical quantum computing as an ideal candidate material for building the future classical-quantum hybrid internet for its numerous benefits, among which the compatibility with the existing communications/computing infrastructure is a main one. I also show that optical quantum computing can be a hardware platform for realizing the multi-valued logic of computing without the need to encode and decode computational problems in binary logic. I also derive explicit matrix representation of optical quantum gates in the phase space representation. Using the multi-valued logic of optical quantum computing, I introduce the first quantum multi-class data classifier, classifying all ten classes of the MNIST dataset.

Dedication

To my parents who had devoted their lives to supporting me.

To my brothers who have been a unshaken foundation of support.

To all the professors who nurtured me with their knowledge and wisdom.

To all my classmates who cried and laughed with me and inspired me with their brilliance through this journey.

To everyone at Portland State University who worked tirelessly for the success of the students.

Acknowledgements

I am deeply grateful for everything I have learned from Portland State University.

Dr. Marek Perkowski has opened up a new world of engineering and quantum computing to me from the treasure trove of his knowledge in the field. He helped me develop an engineering mind of decomposing research problems into manageable chunks and constructing a new approach to solving existing problems.

Dr. Steven Bleiler trained me in a logical and methodical thinking process, which became a foundation for understanding complex scientific and engineering concepts. He freely shared his interest in quantum computing from the topological perspective, which led me to the path of quantum computing research.

Dr. John Acken aided me in seeing the landscape of the engineering field with a bird's eye view to understand specific focused areas of research as part of the integrated whole. He trained me to communicate my ideas to a broader audience in the language they understand and relate to.

Dr. Xiaoyu Song helped me to formulate my ideas and thoughts in a rigorous and precise way to better communicate with the research community at large.

Dr. Daniel Hammerstrom trained me through machine learning and computer vision. His kind and stable demeanor as a scholar has taught me to shape myself as a professional researcher.

Dr. John Lipor trained me in the mathematical foundation of machine learning and showed me various tools for analyzing datasets by treating them as matrices.

Dr. Douglas Hall put me on the path of Ph.D. He has been encouraging and supporting me through the years every step of the way and shared his keen insight into the engineering field.

Emily Hahn stayed on top of all the Electrical and Computer Engineering Department administrative requirements I had to follow.

Roxanne Treece showed her professionalism and care for each graduate student at Portland State University and worked tirelessly for our success.

Table of Contents

List of Tables

List of Figures

List Abbreviations/Symbols

- ALU: Arithmetic Logic Unit
- API: Application Programming Interface
- CCD: Charged-Coupled Device
- CNN: Convolutional Neural Network
- CPU: Central Processing Unit
- CTL: Communications Technology Laboratory
- EDVAC: Electronic Discrete Variable Automatic Computer
- EIT: Electromagnetically Induced Transparency
- ELU: Exponential Linear Unit
- ENIAC: Electronic Numerical Integrator and Computer
- GaAs: Gallium Arsenide
- GBS: Gaussian Boson Sampling
- GPU: graphics processing units
- IC: integrated circuit
- I/O: INPUT/OUTPUT
- ITL: Information Technology Laboratory
- LASER: Light Amplification by Stimulated Emission of Radiation
- MERA: Multiscale Entanglement Eenormalization Ansatz

- MML: Material Measurement Laboratory
- MSE: Mean Squared Error
- NISQ: Noisy Intermediate Scale Quantum
- NIST: the National Institute of Science and Technology
- NSF: the National Science Foundation
- NSTC: the National Science and Technology Council
- OTC: Optical Coherence Tomography
- PCA: Principal Component Analysis
- PML: Physical Measurement Laboratory
- QCNN: quantum convolutional neural networks
- QISC: Quantum Information Science Committee
- QML: quantum machine learning
- QNN: quantum neural networks
- QPU: quantum processing units
- qubit: quantum bit
- QuIC: quantum interconnect
- qumode: quantum mode
- ReLU: Rectified Linear Unit
- SDM: spatial division multiplexing

- SVD: Singular Value Decomposition
- TDM: time division multiplexing, temporal domain multiplexing
- TFQ: TensorFlow Quantum
- TTN Tree Tensor Network
- WDM: wave division multiplexing

1 INTRODUCTION

Digital computing has been successful in solving computational problems since 1945 [37]. Since 2017, quantum computing has entered the scene of practical computational tools when IBM launched the first working quantum processing unit (QPU) [45]. The mathematical formalism of quantum mechanical systems offers a higher-dimensional computational space in quantum computing than in digital computing [1][35, 49, 178, 241]. Additionally, the properties unique to quantum mechanics, not available in classical mechanics, are incorporated as components of computation. They include superposition, entanglement, and interference [120, 135, 178]. Superposition is a projective complex linear combination of the computational basis elements, which allows for parallel processing [47]. Entanglement is a phenomenon in which the state of the entire system cannot be described by its components independently [10, 51, 64, 85]. Interference is a phenomenon in which the electromagnetic waves of multiple computational channels affect the overall wave of the system by interacting either constructively or destructively [10, 79]. These properties, along with the extended computational state space, allow for the processing of computational tasks intractable with classical computers or for conducting computations much faster.

The current landscape of quantum computing is defined as Noisy Intermediate-Scale Quantum (NISQ) or near-term quantum [28, 74, 120, 188], where functioning QPUs with limited capabilities are available on the cloud. These QPUs are called near-term devices because they are characterized by small-scale and shallow circuits [203] to minimize hardware errors until they become fully fault-tolerant. For quantum computers to be considered viable computational resources, they need to be

[1]In digital computing, the computational space is a two-point space $\{0, 1\}$. In quantum computing, the computational space is the surface of a higher-dimensional sphere, which is a projective Hilbert space.

- Fault-tolerant: the level of fidelity to true output needs to be above a certain threshold value.

- Universal: capable of carrying out most of the operations necessary for computation.

The available QPUs follow either of the following two models of quantum computing: discrete variable and continuous variable models [35, 120, 241]. The discrete variable model implements the binary logic of computing with two computational basis $|0\rangle$ and $|1\rangle$ as quantum bits [4, 15, 28, 178]. Hence, it is referred to as qubit-based model in this dissertation. This model is implemented using superconductors, trapped-ions, neutral atoms, and quantum dots [45, 19, 40, 5] [2]. The continuous variable model refers to the mode of quantum computing where the computational space is infinite-dimensional [35, 120, 241]. However, a physical quantum system with an infinite-dimensional state space is a theoretically ideal system since the quantum state space of a physical system is always finite-dimensional. This is because the quantized energy levels used as computational basis is always finite with a distinctive highest level of energy. Thus the implementation of the continuous variable model of quantum computing realizes radix$-n$ multi-valued logic.

The availability of working QPUs on the cloud and their simulators has opened a new era of quantum algorithms, specific to near-term devices [3, 28, 74, 106]. The active areas of quantum algorithm research include quantum chemistry, quantum machine learning, and graph theory problems optimization [11, 12]. In most quantum algorithms, the inherent higher dimensionality of the computational space and quantum mechanical properties are explored to solve some of the problems that are intractable in classical computers [12, 52, 68, 92, 103, 134, 213, 215]. In machine learning, how-

[2] Quantum optics is suggested to be another physical system to implement the discrete variable model. However, there is no working QPU readily available to users yet.

ever, classical computers are successfully carrying out tasks with great results. Hence quantum machine learning algorithms are researched to enhance classical algorithms by tapping into the properties of quantum mechanics [30, 49, 71, 72, 98, 106, 120, 247].

With QPUs emerging as a viable tool for computation beyond digital computing, scientists at the National Science Foundation (NSF) and the National Institute of Science and Technology (NIST) are organizing multiple programs to work with academia and industry experts to develop a blueprint for future quantum interconnects (QuIC) [18]. The schematic includes quantum communication, quantum memory, quantum computing, quantum transducers, and quantum sensing as sub-ecosystems within the framework of the existing classical interconnect infrastructure [18]. In this scheme, data transfer among communications devices, quantum computing devices, and quantum sensor devices plays an important role.

To actualize quantum computing, a physical system needs to be created whose state can be described quantum mechanically. The goal is to maintain the fidelity of quantum states as units of computation. Controlled change of quantum states of the system along with readout of the resulting state is regarded as quantum computation [178]. There are several modes of physical systems considered for implementing quantum computing: superconductors [45, 259], neutral atoms [190], trapped ions [41, 210], quantum optics [11, 34], and quantum dots [5, 150]. These quantum mechanical systems are considered for the implementation of quantum computing as stand-alone solutions. In considering quantum computing as a component of the QuIC, quantum optics offers seamless integration into the existing communications/computing infrastructure.

In both fiber optical communications systems and optical quantum computing, the information carriers are light signals [10]. In today's communications, different forms of signals from multiple devices are converted to light signals compatible with

the fiber optics network [266]. Using light signals as information carriers in quantum computing greatly reduces the number of transducers needed for signal conversion. It is my observation that the active research on quantum memory for quantum communications [78, 112, 131, 152, 153, 154, 155, 162, 267] can be used in future optical quantum computers to add the memory component. Memory component is necessary in computer processing units to store intermediary computational states and complex programs. Additionally, optical sensors are widely used to measure different quantities such as temperature, wave signals, bio-metrics, distances, and tomography [10]. Existing optical sensors can be converted to quantum sensors through the process called "squeezing", which converts classical light expressed in terms of waves to quantum light expressed in terms of discrete energy quanta [241].

Optical quantum computers have the advantage of operating at room temperature and being integrable into the existing communications infrastructure for multiplexing and networking [233]. Photons are known to be the most stable and robust to decoherence, i.e., stable in the presence of environmental noise [1]. Optical quantum computing naturally incorporates the interference phenomenon among interacting photons as a component of information processing. In the traditional qubit model [3], interference is something they have to suppress to keep the integrity of the computational states.

In addition to physical benefits, quantum optics offers a new paradigm of information processing from the logic implementation perspective. Quantum optics has properties of higher-dimensional state space that can be used for multi-valued logic realization and flexibility to users to determine the dimension of the computational

[3]The traditional model of quantum computing is based on realizing the binary logic of computing. In optical quantum computing the multi-valued logic of computing naturally arises at room temperature. The distinction is explained more in detail in Chapter 4. Logic Implementation: Multi-Valued Logic.

space [49]. Multi-valued logic and binary logic are naturally embedded within the notion of the cutoff dimension [4]. The cutoff dimension is the highest level of energy that the user has the flexibility of determining for each instance of computation [5]. The same cutoff dimension is applied across all the quantum channels used for computation [6]. Employing the cutoff dimension of three realizes ternary logic and of two, binary. The theoretical underpinning of quantum optics allows the binary logic, ternary logic, and multi-valued logic with higher radices implementations of quantum computing in optical quantum devices.

The concept of "cutoff dimension" is referred to only in one paper in the literature, as a side note to an experiment [120]. The power this concept provides in determining the dimension of the computational space in each instance of computation and the subsequent realization of the multi-valued logic of computing are not known in the research community.

With the working QPUs and their corresponding circuit-building [7] and quantum algorithm developing software tools [8] research in quantum machine learning has become very active. Quantum machine learning models can be developed using variational circuits, wherein the constituent quantum gates are parameterized [61, 183, 207]. The process of "learning" consists in finding an optimal set of parameters for the parameterized quantum gates [49]. A quantum circuit is a collection of quantum gates, which induce a change of states on the original quantum data as quantum computation [207]. Quantum algorithms are implemented on quantum circuits and the computational results are evaluated on classical circuits for opti-

[4] See Section 4.2.2. Multi-Valued Logic Model for a detailed explanation.

[5] This realization is one of my contributions.

[6] In using Xanadu's QPU X8, which is the only known optical quantum computer to which the notion of cutoff dimension can be applied.

[7] Google's Cirq, IBM's Qiskit, and Xanadu's Strawberry Fields

[8] Google's TensorFlow Quantum, IBM's Qiskit, and Xanadu's PennyLane

mization. Within the framework of classical circuits, the machine learning algorithm component is outsourced to QPUs and the resulting vectors are then evaluated on CPUs. Because of this natural integration of classical and quantum, hybrid algorithms combining classical and quantum algorithms together yield successful results.

In implementing quantum machine learning, optical quantum computing offers more types of quantum gates that are not available in the qubit model [49]. Especially in implementing quantum neural networks, the availability of nonlinear quantum gates in addition to linear quantum gates allows optical quantum computers to be able to directly translate classical neural networks into the quantum domain [49, 120, 212]. Multi-valued logic-based optical quantum computers provide additional flexibility in controlling the length of the output vectors of quantum circuits, using different methods of measurement and the notion of cutoff dimension [9] [49].

Optical quantum computing is a viable solution for implementing universal fault-tolerant quantum computing from the physical implementation perspective and the logic realization perspective [10]. Light is abundant, robust to decoherence, has low latency, easy to multiplex and network, and is compatible with the existing communications/computing infrastructure [10]. Optical quantum computers are easily fabricable for mass production at a low cost and operate at room temperature. It naturally realizes the multi-valued logic of computing and avails the users to define the dimension of the computational space for each instance of computing. Additionally, it offers nonlinear quantum gates that are not available in other modes of quantum computing.

Using the advantages of optical quantum computing, I created a multi-class data quantum classifier on all ten classes of the MNIST dataset [49]. The architecture

[9] See Section 6.3 My Research on Quantum Multi-class Data Classifier.

[10] The phase space representation of quantum optics, realized as quantum computing by Xanadu, realizes the multi-valued logic of computing.

supports data with n^m classes where n is the dimension of the computational space for each qumode [11] (cutoff dimension) and m is the number of qumodes used for computation. In Xanadu's X8, using the cutoff dimension 10 and eight qumodes result in the dimension $10^8 = 100,000,000$ for the overall computational space. That means datasets with $100,000,000$ classes can be classified as easily as datasets with 2 classes using my architecture. The multi-class data classifiers in the literature classify only up to four classes [32, 140].

This dissertation is organized as follows: Chapter 2 outlines the research problems this dissertation is addressing, Chapter 3 examines QuIC as defined by the NIST for building future classical-quantum internet, Chapter 4 reviews the mathematical foundation of quantum computing and the multi-valued nature of optical quantum computing, Chapter 5 presents matrix representation of optical quantum gates, and Chapter 6 describes the multi-class data quantum classifier that I created.

[11] Quantum mode which is a channel for light signals to travel through.

2 BACKGROUND

Quantum computing uses the quantized energy levels of electromagnetic fields as a computational basis. There are two models of quantum computing in the literature today: discrete variable and continuous variable models. The discrete variable refers to binary logic and the continuous variable model refers to a quantum computing model endowed with an infinite-dimensional computational space.

The difference between these two models in physical implementation is the number of quantized energy levels selected to represent the computational basis for quantum information processing. To implement the discrete variable model, the lowest two energy levels are selected to represent $|0\rangle$ and $|1\rangle$. To implement the continuous variable model, multiple levels of quantized energy naturally occurring in the system are used to represent $|0\rangle, |1\rangle, \ldots, |n\rangle$.

This distinction between the two models is not widely known in the research community, however. The term "quantum computing" is generally used to mean the discrete variable model of quantum computing realized with the binary system of $|0\rangle$ and $|1\rangle$. In 2001, Knill, Laflamme, and Mulburn proved that optical quantum computing implemented with beamsplitters and phase shifters is universal under the discrete variable model. In the actual implementation of optical quantum computing, Xanadu implemented the phase space representation [12], which is described following the principles of the continuous variable model. The theoretical continuous variable model cannot be actualized in physical systems, however, since the quantum state space of any physical system is not infinite-dimensional. This is due to the fact that no physical system contains an infinite amount of energy. The model realized by Xanadu's X8 implements the multi-valued logic of computing instead of the contin-

[12] A representation using the position and momentum observables.

uous variable model. This distinction between the theoretical continuous variable model and its realization of the multi-valued logic in physical implementation is not widely known in the research community.

The idea of using a physical system whose state can be described quantum mechanically for information processing has first been introduced by Yuri Manin and Paul Benioff separately in 1980 [24, 25, 159]. In 1982, Richard Feynman suggested that a computing device that can accurately simulate nature has to be built on the principles of quantum mechanics. His idea was an extension of the binary logic of digital computing using two different polarization states of photons to represent classical bits as quantum bits (qubits) [76].

Now with working QPUs made available by companies like IBM, Google, Rigetti [13], and Xanadu [14], quantum computing has emerged from the domain of theoretical research into the practical realm of carrying out actual quantum computations [11, 15, 45]. The current landscape of quantum computing is termed "Noisy Intermediate-Scale Quantum" (NISQ) because the available QPUs are not fully fault-tolerant [28, 188, 242]. This means they are prone to hardware errors. Additionally, they are not capable of solving meaningful real-life computational problems yet because they are characterized by short and shallow circuits [120]. Most of the computational problems require more computing power in terms of the number of computational units available and longer series of gate operations. Today's QPUs are equipped with Arithmetic Logic Units (ALU), controlled by classical signals, but lack the memory component to store reusable programs.

Nevertheless, they serve as proof-of-concept for future universal fault-tolerant

[13] QPUs by these companies are implemented under the discrete variable model (binary logic, qubit-based) of quantum computing.

[14] Xanadu's X8 is implemented under the continuous variable model, physically realizing the multi-valued logic of computing.

quantum computing. The availability of functioning QPUs and simulators has opened up a new landscape for quantum information and computation. These devices [15] and associated software tools [16] stimulated research on quantum algorithms for solving real-life problems within the framework of quantum computing.

The goal of today's quantum computing is developing fully universal and fault-tolerant devices [66, 170]. Once it is achieved, those devices need to be multiplexed for more computing power and networked for communications. To achieve this goal, there are several areas of research that need clearer solutions on. Among those, my dissertation addresses the following areas.

- Physical implementation: There are several modes of quantum mechanical systems considered for implementing quantum computing [5, 11, 20, 34, 41, 150, 157, 190, 212, 252, 259]. Out of these candidate materials, which is most cost-effective, easy to operate, and compatible with the existing communications/-computing infrastructure?

- Logic implementation: Optical quantum computing offers a way of realizing the multi-valued logic of computing, bypassing the need to encode computational problems in binary logic.

- Matrix representation of optical quantum computing: In phase space representation [17] of optical quantum computing, the matrix exponential formulas for various optical quantum gates are found in the literature. However, there is no explicit matrix representation of these gates that would allow designers to use them as gate descriptions similarly as it is done in standard quantum computing in technologies such as superconducting.

[15] e.g., Google's Sycamore, IBM's Osprey, and Xanadu's X8 and Borealis

[16] Google's Cirq and TensorFlow Quantum, IBM's Qiskit, and Xanadu's Strawberry Fields and PennyLane.

[17] See Section 4.2.2 Multi-Valued Logic Model

- Multi-class data quantum classifier: Most of the quantum machine learning classifiers are binary classifiers [31, 106, 120]. The multi-class data quantum classifiers in the literature classify up to four classes on the MNIST dataset [32, 140].

In the following subsections, I will outline the background of each of the research problems listed above.

2.1 Physical Implementation of Quantum Computing

With the idea of quantum computing no longer residing in the domain of theoretical research, the NSF and the NIST are putting in concerted efforts to build a blueprint for a quantum computing ecosystem that can be integrated into the existing communications/computing infrastructure [18]. Today's communication system is built on fiber optics cables, to which individual nodes are connected [58, 107, 145, 169, 234, 240]. Electrical signals coming from computing devices are converted to light signals and propagated through fiber optics to reach the destination computing device [266].

To build fault-tolerant universal quantum computers that can seamlessly be integrated into the existing communications/computing infrastructure, the physical material used needs to have the following characteristics:

- Easily integrable into the existing communications/computing infrastructure.
- Easily fabricable for mass production.
- Cost-effective to be accessible to the masses.
- Easily controllable for computation.
- Portable and easily embeddable.

- Easy to be multiplexed for added computing power.
- Easy to be networked for communications.
- Accommodating to the quantum memory component.
- Robust to environmental noise.
- Low-latency.

In the quantum computing research community, there is yet to be established a uniform agreement as to the physical mode that would meet all the criteria mentioned above. In the physical implementation of quantum computing, the quantum computing research community is exploring several different modes of physical systems: superconductors [259], trapped ions [41], quantum optics [11, 34, 157, 212, 252], microwave [20], neutral atoms [190], and quantum dots [5, 150]. Out of these, the question of which choice would enable the most seamless integration with the existing infrastructure has not been fully explored yet. While the quantum computing research community as a whole is finding solutions to engineering challenges and developing quantum algorithms, there is a lack of unified research effort for selecting the most suitable physical implementation mode with consideration of integrating future quantum computing into the existing infrastructure.

Therefore, there is a need to have an integrated unified system of communications/computing that would encompass available computing resources such as CPU, QPU, GPU, and classical and quantum embedded systems. According to NSF and NIST, the research on building a quantum communications/computing infrastructure on the foundation of existing architectures is one of the top priorities in quantum information and science research [18].

For quantum data transfer and communications, the goal is to use the existing fiber optics network [18]. Fiber optical communications system transmits information using pulses of infrared light as signal carriers [10]. For the last-mile connections to end-users, the preferred way of connecting is via fiber optics as well. Existing coaxial cables based on copper are being replaced with fiber optics [58].

2.2 Logic Implementation

In digital computing, binary logic based on the ON and OFF switches of transistors is used for computation. As a natural extension of digital computing in the framework of quantum computing, most of the research has been done on realizing the binary logic model of quantum computing. However, with the advent of optical quantum computing, direct processing of computational problems without the need to encode in binary logic is availed. I briefly outline the history of the binary logic of digital computing below.

With the idea of an abstract computation machine that Turing proposed in 1936, we have been able to realize his notion of computing device [229]. The first programmable digital computer ENIAC (Electronic Numerical Integrator and Computer) was introduced in 1945 [37]. A programmable digital computer EDVAC (Electronic Discrete Variable Automatic Computer), equipped with the memory component to store programs, was developed in 1949 [37]. With the advent of integrated circuits (IC), different digital components necessary for carrying out computations became nano-scale, allowing for desktop computers and embedded systems. Now with the internet, the world has become one interconnected computing unit exchanging and processing information interactively. With quantum near-term devices available to researchers, we are ready to enter into yet another information revolution phase.

The binary logic that digital computing is based on has its root in practical

hardware considerations [37]. The first programmable digital computer ENIAC was based on decimal logic, using the binary operations of the ON and OFF switches with vacuum tubes, crystal diodes, relays, capacitors, and resistors. EDVAC switched from decimal to binary logic for hardware efficiency purposes. In IC, the ON and OFF switches in transistors are natural implementations of the binary logic. Binary digits, rather than other forms of multi-valued logic such as decimal digits, are easier for electronics-based digital computers to store and process information.

From the computer science perspective, it is only natural to limit the concept of quantum computing to the binary logic model since it is a quantum version of the already successful classical computing paradigm. Hence, with the possibility of using the two distinct states of particles as a computational basis, the field of research in the binary logic of quantum computing was very active and kept expanding. The first quantum algorithm demonstrating the power of quantum computing over classical was presented by Deutsch and Jozsa on a theoretical problem [67]. Thereafter, a rich body of literature has been developed in additional algorithms such as Shor's algorithm for prime factorization [213], phase estimation for finding eigenvalues of matrices [122], Simon's algorithm [215] for a query problem, and most notably Grover's search algorithm [92]. A way of solving systems of linear equations using quantum circuits was added to the literature [97].

The success of these algorithms is due to the superposition property of quantum mechanics wherein all the basis states within the computational space are considered simultaneously, leading to parallelism. The superposition property arises from the entanglement property of quantum mechanics wherein the state of the whole system cannot be described separately as a collection of each individual qubit state [10]. Most of the quantum research has been done under the binary logic (qubit-based) model of quantum computing, as a natural quantum extension of classical computing.

Now, existing optical QPUs offer an opportunity to encode computational problems in a whole new way, using multi-valued logic. The computational basis in optical quantum computing under the phase space representation is a set of integer states $\{|0\rangle, |1\rangle, \ldots, |n\rangle, \ldots\}$, describing different quantized energy levels of the electromagnetic field of light [35, 239, 241]. Although it is infinite in theory under the continuous variable model, it is a finite set since there is a maximum number of energy levels observed in reality. This finite set of integer values can be used as a basis for multi-valued logic implementation. With optical quantum computers connected via fiber optics, two distant computing nodes can function as one computer. For example, the reading from quantum sensors can be directly used as inputs in quantum algorithms run on separate computing units. The quantum state inputted from a quantum sensor in Mexico can be transmitted via quantum internet and processed by a quantum algorithm in Belgium.

With improved technology in future optical QPUs, higher numbers of photons can be expressed in an optical system, allowing for a yet higher radix-based multi-valued logic of computation. In the multi-valued logic literature, the practical gates were developed for only ternary [18] and quaternary [19] due to the limitations of hardware platforms. The availability of logic with high radices in optical quantum computing offers a new possibility for both theoretical and applications-related research in which computational problems will be represented using multi-valued variables with high radices. This opens up a possibility of developing algorithms, which operate on such data, expressed in terms of high-radix multi-valued logic.

For each instance of computation, the user can determine the dimension of the computational space using the notion of "cutoff dimension" in Xanadu's X8. Setting

[18] base three
[19] base four

the cutoff dimension at n realizes the multi-valued logic of radix-n for each qumode uniformly [20]. With the advent of optical quantum computers whose dimension of the computational space can be controlled by users, the freedom to choose among binary logic and multi-valued logic of quantum computing for the computational problems at hand is available to users. However, this concept of realizing the multi-valued logic of computing using the cutoff dimension (by controlling the dimension of the computational space) is not recognized by the research community. The multi-valued logic research community has been looking for classical hardware solutions. The quantum computing research community has been focusing on realizing the binary logic of computing. The optical quantum computing research community has been focusing on the infinite-dimensional nature of the quantum state space of light-quanta.

This concept of realizing the multi-valued logic of computing is used in different experiments I performed: classical-quantum hybrid auto-encoder [21] and hybrid multi-class data classifier [49] that will be discussed in detail in later sections.

The idea of using the electromagnetic field of light for quantum computing was first introduced by Lev Vaidman in 1993 for quantum teleportation [232]. Walls and Milburn outlined ways for using quantum optical principles for quantum information in 1994 [236]. Richard Feynman first proposed the idea of using two distinct states of polarization of photons for the binary logic model of quantum computing (1980) [76]. In 2001, Knill, Laflamme, linear optics is universal under the qubit-model of quantum computing. In 2011, Scott Aaronson and Alex Arkhipov proposed that if we can produce indistinguishable photons arriving at the photon detector at the same time, we can use the system for information processing by sampling multiple shots of the operation. The process is called Gaussian Boson Sampling (GBS) [22] [1]. Although

[20] In current technology, it is not feasible to set different cutoff dimensions for different qumodes. The same cutoff dimension is applied to all the qumodes used for computation.

[21] Architecture proposed in "Continuous variable quantum neural networks [120].

[22] This model is just a formal description of the probabilistic nature of quantum computing. The

Aaronson and Arkhipov's model was for the qubit-model of quantum computing, their idea of using indistinguishable photons through a linear optical network for quantum information processing became a basis for realizing the multi-valued logic of quantum computing. In 2019, Xanadu introduced an optical QPU, called X8, which operates on the principle of GBS under the phase space representation [23] of quantum systems [11]. It contains eight quantum modes (qumodes) which are eight separate optical channels. The state of each channel is described with a finite number of light-quanta energy states. In X8, the number of light-quanta observed is 18, including the ground energy state representing zero-point energy. In the new time-domain multiplexed QPU, Borealis by Xanadu, registering events with up to 219 photons and a mean photon number of 125 are observed [157]. That means X8 can realize up to radix-19 multi-valued logic and Borealis up to radix-126.

Although the optics-based quantum computing has been considered from the theoretical perspective since the 1990s, there is only a small body of literature on actual algorithms that are testable and can be expanded upon [49, 120]. Most of the quantum research has been done under the binary logic (qubit-based) model of quantum computing, as a natural quantum extension of classical computing. There is a very small body of literature using ternary or quaternary logic for quantum computing from the theoretical perspective [158]. With the multi-valued quantum computing physically implemented on optical QPUs, there is a potential for formulating computational problems with a greater degree of flexibility than just in binary logic. In Chapter 4 of this dissertation, I present the mathematical formalism of the binary logic and the multi-valued logic of quantum computing.

implementation was suggested using quantum optics, but the general process applies to any physical implementation of quantum computing.

[23] The phase space representation refers to the representation using the position and momentum observables. This is achieved using the Wigner function.

2.3 Optical Quantum Gates

Quantum gates are mathematical expressions of controlled operations in a quantum mechanical system for the purpose of information processing. The quantum states used for quantum computation are physically manipulated to mirror the operation of quantum gates expressed in mathematical forms.

Optical quantum gates are expressed as matrix exponential of the Hamiltonian, describing the energy of the system [120]. There are matrix exponential forms of the optical gates given in literature [35, 239, 241]. However, their explicit matrix representation is missing. In Chapter 5 of this dissertation, I will present some matrix representations of standard optical quantum gates built from the constructor and annihilator matrices, which are building blocks of optical quantum gates.

2.4 Quantum Data Classification

In quantum machine learning (QML), the mapping of data samples is done in a higher-dimensional Hilbert space and then the computational results are brought back down to the classical domain via a measurement operator [201]. A QML circuit is composed of a data encoding circuit, a quantum algorithm circuit, and a measurement operation [49]. The output from the quantum circuit is sent to a CPU wherein all the optimization calculation is done [207]. The updated parameters are sent to the QPU and the quantum gates are recalibrated.

Since the goal of quantum deep learning is to extract hidden features of data from a higher-dimensional computational space, it would be natural to explore the quantum computing model which offers a higher-dimensional computational space.

Due to the limited computing capacities of near-term devices, data samples with a number of features exceeding the capacity cannot be directly encoded to quantum states. Instead, they are preprocessed classically to reduce the size of input vectors

[32, 49, 106, 120, 140]. Therefore, all the existing quantum classifiers on data with a high number of features are classical-quantum hybrid networks.

The multi-class data quantum classifiers in the literature are up to four classes on the MNIST dataset using quantum convolutional neural networks [32, 140]. I invented a ten-class quantum classifier using optical quantum computing motivated by Killoran et al.'s work [120]. The architecture is capable of accommodating up to $18^8 = 11,019,960,576$ classes, using eight qumodes [24]. In a qubit circuit, it would require $\log_2(11,019,960,576) \simeq 34$ qubits. This flexibility of determining the size of the output availed in optical quantum computing is not widely known in the research community. Although we can speculate that this technology can also lead to a much-improved accuracy, sensitivity, and confidence matrices in machine learning, no further research has been done in this area to my knowledge. The published papers referencing Killoran's paper are exploring the qubit-based machine learning algorithms for classification. In Chapter 6 of this dissertation, a detailed explanation of related works and the multi-class data quantum classifier is presented.

[24] Using the maximum radix possible in X8 which is 18.

3 PROPOSED ARCHITECTURAL SOLUTION: QUANTUM INTERCONNECT

In concert with the advances in quantum computing research, NSF and NIST are envisioning the paradigm of integrated classical and quantum computing as part of the classical-quantum hybrid internet. Within that framework, quantum computing devices are regarded as components of a worldwide network, called Quantum Interconnect (QuIC) [18]. QuICs are processes and devices which connect all quantum computing and sensor devices to the quantum communications network. Since quantum effects are naturally occurring phenomena in optics, they are envisioning a scheme to use the currently existing fiber optics communication networks for quantum communications as well. The information carriers for fiber optical communications are the electromagnetic fields of light [58, 107, 169].

Therefore, physical systems implementing quantum computing that are most easily integrable into the existing infrastructure are those that use quantum optics. Optical quantum computing has the following properties:

- Easily integrable into the existing communications/computing infrastructure.
- Easily fabricable for mass production.
- Cost-effective to be accessible to the masses.
- Easily controllable for computation.
- Portable and easily embeddable.
- Easy to be multiplexed for added computing power.
- Easy to be networked for communications.
- Accommodating to the quantum memory component.

- Robust to environmental noise.
- Low-latency.
- Easily multiplexed and networked using optical fibers.

In today's computing, no individual computing device can exist in isolation to fully tap into the availability of rich data and applications offered via connectivity to a broader information community. The interconnected system of communications network, large-scale computers in data centers, individual nodes of end users, and embedded systems all work in concert to compute, process, and exchange information. Future quantum computing devices are going to be connected to the classical-quantum hybrid communications/computing infrastructure. Quantum data transfer between connected devices and the communications network needs to keep the integrity of the quantum state as the correct computational state.

NIST is spearheading a multi-disciplinary research program to this end including Information Technology Laboratory (ITL), Communications Technology Laboratory (CTL), Material Measurement Laboratory (MML), and Physical Measurement Laboratory (PML) [266]. In September 2018, the National Science and Technology Council released a report, "National Strategic Overview for Quantum Information Science," which stated, *"Through developments in [quantum information science], the United States can improve its industrial base, create jobs, and provide economic and national security benefits."* [258]. Research in quantum information science is among the top priorities in scientific research at the federal government level. Quantum optics as related to the envisioned quantum communications network is one of the leading areas of research.

3.1 Components of Quantum Interconnect

Today's communication system is built on fiber optics cables, to which individual nodes are connected [58, 107]. Electrical signals coming from computing devices are converted to light signals and propagated through fiber optics to reach the destination computing device [10]. For an integrated quantum information system to interconnect the world the way classical computing does today, it should be built on the foundation of the existing communications network [18]. The core objective of QuIC is building interconnected quantum information processing architecture on the foundation of the existing classical communications/computing infrastructure. The classical interconnect is composed of communication networks, computing devices, transducers for signal conversion across different devices, and embedded systems. The quantum equivalent of the process is envisioned to be composed of similar components: quantum communications, quantum memory, quantum computing, quantum transducers, and quantum sensors (metrology) [18].

In November of 2019, NSF sponsored a workshop to identify the scientific and community needs, opportunities, and significant challenges for QuIC over the next 2-5 years [18]. The resulting schematic from the workshop on QuIC is depicted in Figure 1.

The main components of QuIC, as envisioned by NSF and NIST are:

- Quantum communications: Connecting quantum computing devices to a network for data transmission and reception.

- Quantum memory: For optical signal regeneration in fiber optics, the memory component is needed to store optical signals. Optics-based quantum memory research is one of the research focus areas by NIST [78, 112, 131, 152, 153, 155]. I believe the research findings in communications memory can be utilized for

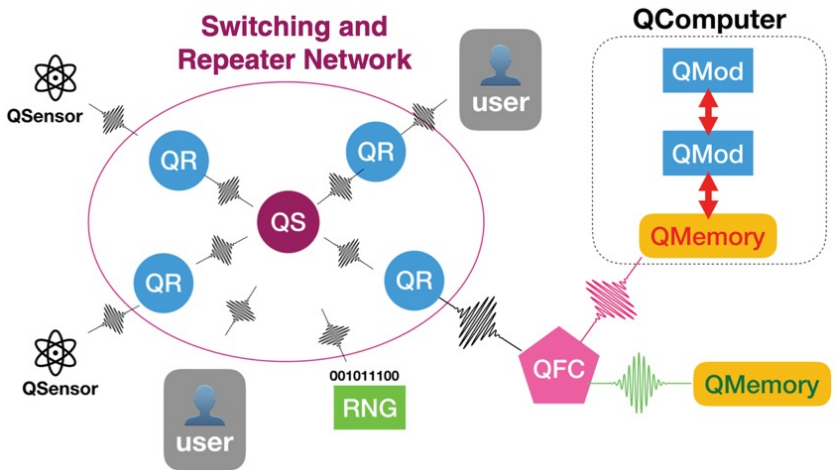

Figure 1: Quantum Interconnect (QuIC) architecture [18]

the memory component in future QPUs as well.

- Quantum computing: Individual computing devices carrying out quantum information processing need to be connected to the communications network.
- Quantum transducers: The devices and processes for converting quantum signals from a quantum computing device to compatible signals to connecting devices [266].
- Quantum sensors (metrology): Quantum devices for measuring physical quantities such as temperature, distance, and topography [268].

3.1.1 Quantum Communications

Quantum communications refer to the process and devices that facilitate data exchange and transfer between quantum computational devices. It is envisioned that the communications networks would carry both classical signals and quantum signals [18].

Today's communications infrastructure is built on fiber optical cables, which are bundled cables of optical fibers [107]. Fiber optical communications system transmits

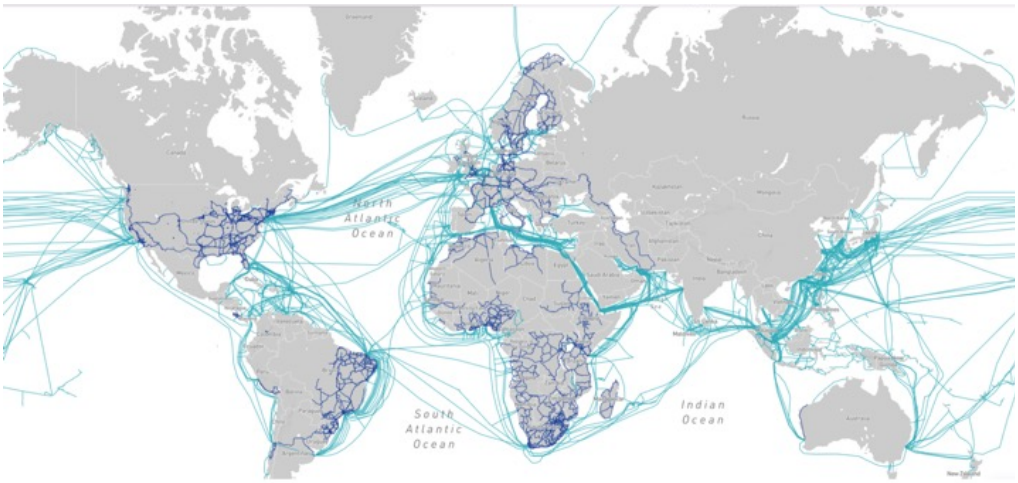

Figure 2: Fiber optics communications network [image source]

information using pulses of infrared light as signal carriers [10]. Figure 3 illustrates the existing submarine fiber optical communications network for intercontinental connections. For the last-mile connections to end-users, the preferred way of connecting is via fiber optics as well [58].

Optical fibers are channels of light paths, composed of ultra-thin glasses as seen in Figure 3. Bundles of optical fibers make a fiber optical cable. Optical fibers fit for long-distance communications were first developed by Corning Glass in 1970, capable of transmitting optical waves with low information loss [10]. In the same year, GaAs (Gallium Arsenide) semiconductor lasers (Light Amplification by Stimulated Emission of Radiation) were developed by an Italian company for transmitting light through fiber optical cables at room temperature [10]. Lasers produce coherent light, whose frequency and waveform can be controlled. The advent of the laser facilitated the transmission of information carrying light over low-loss optical fibers for long-distance optical information transmission.

The laser is an optical device wherein light is amplified via oscillation around a medium placed between two reflecting mirrors. The atoms in the medium, stimulated

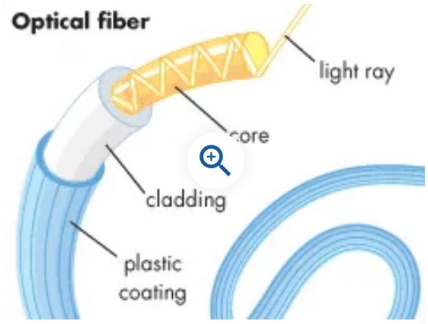

Figure 3: Optical Fiber (Encyclopedia Britannica)

by oscillating light emit light of the same frequency in the same direction. This produces a coherent, monochromatic, and highly directional beam of light, which can be controlled and manipulated as information carriers in long-distance communications [10].

This simultaneous availability in the same year of compact optical sources and low-loss optical fibers led to an efficient worldwide fiber-optics communications system [10]. Since then, optical fiber technology has gone through multiple generations of communication advances with increased bandwidth and negligible latency [107]. *"Among the benefits of optical communication systems are their high reliability over long-distances, low attenuation, low interference, high security, very high information capacity, longer life span and ease of maintenance"* [107]. Therefore, optical communications systems are widely used to transmit voice, video, computation data, and television signals [58].

Since most computing devices today are using electrical signals, transmitting information over fiber optical cables is composed of the following four steps:

- converting an electrical signal to an optical signal using a transmitter
- relaying the signal along the fiber, ensuring that the signal does not become too distorted or weak (repeaters)

- receiving the optical signal
- converting it into an electrical signal

There are two types of optical fibers: single-mode fibers and multi-mode fibers [107]. This provides a potential for transmitting multi-mode quantum computing signals directly over fiber optics cables without the need to encode data. There may be signal attenuation issues that will need to be resolved. However, connecting quantum computing devices through a communication cable consisting of the same number of qumodes provides a way for connecting two quantum computing devices not just for information exchange, but for combining them as one computing unit.

3.1.2 Quantum Memory

Quantum memory is a device that can store quantum states and retrieve them on demand with high fidelity [62, 155]. Optical quantum memories are actively researched for long-distance communications systems. For long-distance signal transfer, overcoming the attenuation of signals is important. For optical signals to travel long-distance through fiber optical cables, repeaters are used to regenerate attenuating signals to maintain fidelity [99, 152]. Repeaters are placed at a certain distance interval in fiber optical cables. The idea of quantum repeaters is based on repeating the same quantum state as a signal within shorter intervals of the communications channel [62].

One of the main components of repeaters is the memory component, storing information carried in optical signals [10, 78, 99, 112, 131, 153, 154]. Under the umbrella of QuIC, NIST has a designated subarea of research for quantum memory and repeaters for long-distance communications [267].

A quantum repeater, for a quantum communications network as envisioned by NIST, requires quantum memories. The role of quantum memory in quantum com-

puting and quantum communication systems is analogous to the role of electronic memories in classical communication and computing systems. The primary goal of quantum memory research is to develop the memory component for long-distance optical quantum signal transmission on fiber optics networks. Thus, the primary areas of research for implementing quantum memory are all optics based: trapped ion [119], electromagnetically induced transparency (EIT) [154], and atomic-ensemble [86].

Out of these areas, the research in EIT is the most advanced in terms of technical readiness, both experimentally and theoretically [267]. EIT works on the principles of cancellation of photon absorption by the medium atoms controlling the electromagnetic field. Such cancellation results in a large degree of transparency [111]. Researchers have demonstrated the ability to control the group velocity of light pulses: ultrafast, ultraslow, or bring light to a full stop. This capability to control light propagation allows for light storage, thus information storage in the form of light.

The principle of EIT is storing information carried by photons in medium atoms by exciting the state of the atoms [111]. An optical pulse is applied to the atom resting at ground state $|g\rangle$, which excites it to a higher energy level state $|e\rangle$. In the process, the photon is absorbed by the atom including the information it carries. When a control pulse is applied to the atom in a higher energy state, the absorbed photon is released with the original information contained within it.

The duration of time needed for the photon to be absorbed by the medium atom and its release through the application of a pulse is considered "storage" of optical information. The light and atom interaction in terms of energy level changes is depicted in Figure 4. It is observed in experiments that light pulses can be coherently stored in medium atoms for up to one minute [154]. This demonstrates that optical memories can successfully store the information carried in optical signals.

Currently, the existing QPUs are basic Arithmetic Logic Units or other special-

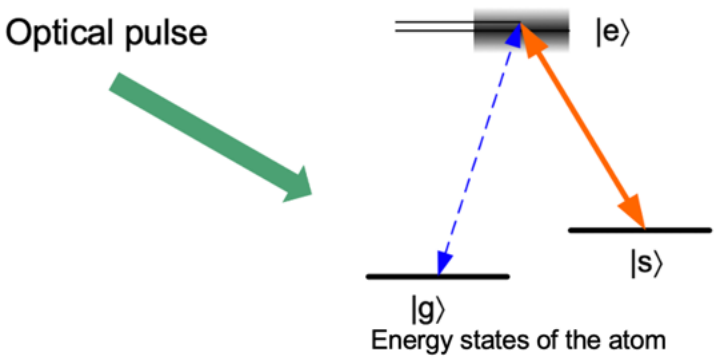

Figure 4: Information storage in EIT [154]

ized processing elements and systems for carrying out computations but are devoid of the memory component capable of storing complex programs. For quantum computers to be practical tools for computation to end users, they need to be equipped with memory components. The schematic of the internal memory component in CPUs is shown in Figure 5.

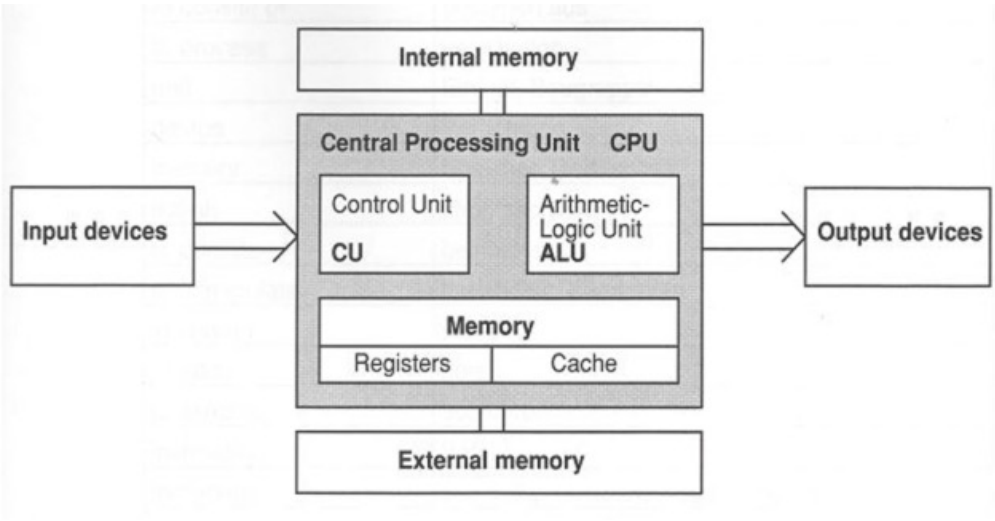

Figure 5: CPU architecture as applied to future quantum computing [image source]

The addition of the memory component will enable QPUs to store programs, hence performing much more complex computations than that is done in the current quantum computing architectures. It is my observation that active research in op-

tical quantum memory can be used for the memory component in optical quantum computers since optical quantum memories are capable of strong light signals.

Xanadu's Borealis, which is built on the principle of time-domain multiplexing has three time-delayed gate operations [157]. Each gate operation is composed of rotation gates and beamsplitters [25]. This is another way of realizing quantum memory by holding computational quantum states for a duration of time. The buffer of time is acting as memory, holding the quantum state until the next operation. The first, second, and third delay loops have a round-trip time of 1τ, 6τ, and 36τ, respectively [271].

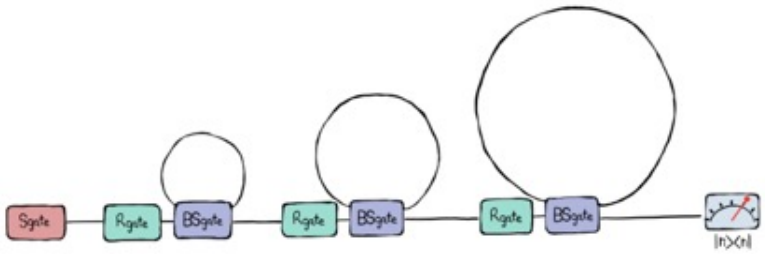

Figure 6: Borealis time buffer as memory (Xanadu website)

3.1.3 Quantum Computers

Quantum computers are devices that carry out information processing using some physical systems whose states are described based on the principles of quantum mechanics. As criteria for the physical implementation of quantum computing, David DiVincenzo proposed the following criteria which have become the main guideline for building quantum computers [19, 70]:

- *Well-characterized and scalable qubits. Many of the quantum systems that we find in nature are not qubits, so we must find a way to make them behave as such. Moreover, we need to put many of these systems together.*

[25] See Section 4.2.2 for optical quantum gates.

- *Qubit initialization. We must be able to prepare the same state repeatedly within an acceptable margin of error.*

- *Long coherence times. Qubits will lose their quantum properties after interacting with their environment for a while. We would like them to last long enough so that we can perform quantum operations.*

- *Universal set of gates. We need to perform arbitrary operations on the qubits. To do this, we require both single-qubit gates and two-qubit gates.*

- *Measurement of individual qubits. To read the result of a quantum algorithm, we must accurately measure the final state of a pre-chosen set of qubits.*

To actualize quantum computing, a controllable physical system operating under the principles of quantum mechanics is used. Controlled change of quantum states of the system and readout of the resulting state is regarded as quantum computation. Current physical systems being explored for realizing future quantum computers include superconductors [259], trapped ions [41], quantum optics [11, 34, 157, 212, 252], microwave [20], neutral atoms [190], and quantum dots [5, 150]. All these candidate systems are explored to realize the binary logic model of quantum computing. The lowest two states of the electromagnetic fields within the system are selected to express the quantum basis states $|0\rangle$ and $|1\rangle$. This requires restricting higher levels of energy states to only two, i.e. the ground state and one level higher excited state. A dilution refrigerator is used for this purpose to keep the ambient temperature to near absolute zero [256].

In the following pages are short descriptions of the physical materials being explored for the implementation of quantum computing.

Superconductor

A superconductor is any material that has no electrical resistance and does not allow magnetic fields to penetrate [222]. They are normally implemented with lossless capacitors, inductors, and Josephson junctions [20]. The electric current in this stable system, in theory, can exist forever [222]. This system with negligible resistance and fluctuation can be used for quantum information processing. The electric current in a superconductor flows without dissipation, i.e, resistance, keeping the electron waves' phase coherence. Below a critical temperature, called the superconducting transition temperature, superconductors exhibit zero-resistance [135].

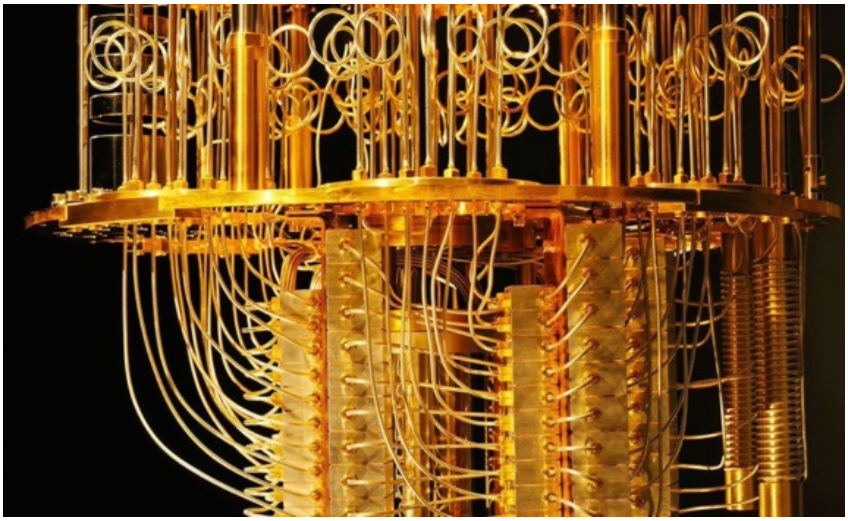

Figure 7: Superconduting quantum computing [image source]

Some of the materials exhibiting superconducting properties include chemical elements such as mercury or lead, alloys such as niobium–titanium, ceramics, and superconducting pnictides [214]. Google, IBM, and Rigetti use superconductors for their quantum computers. The material used to realize the superconducting model of quantum computing may vary from company to company.

However, all the superconductors need to be kept below the superconducting

transition temperature, hence the need for a custom-made dilution refrigerator per material used, which is costly. The schematic of a dilution refrigerator housing qubits is shown in Figure 7. The electron waves used for quantum computations are then accessed via wires from outside, in which a rapid change of temperature occurs. Maintaining the fidelity of quantum computational result states classically at room temperature requires highly sophisticated engineering schemes [256]. In the future universal and fault-tolerant quantum computing era, it would be hard to contain a dilution refrigerator in a personal device such as a desktop, laptop, cell phone, or quantum sensor.

Trapped Ion

Quantum computers using trapped ions are implemented by companies like Honeywell and IonQ [19]. Ions are atoms with a missing electron, created by evaporating a metal and trapped to form a qubit register [41]. The manipulation of the quantum state of ions is done via the use of lasers, which affect the electromagnetic fields of the ions [135]. The schematic of trapped ions is demonstrated in Figure 8.

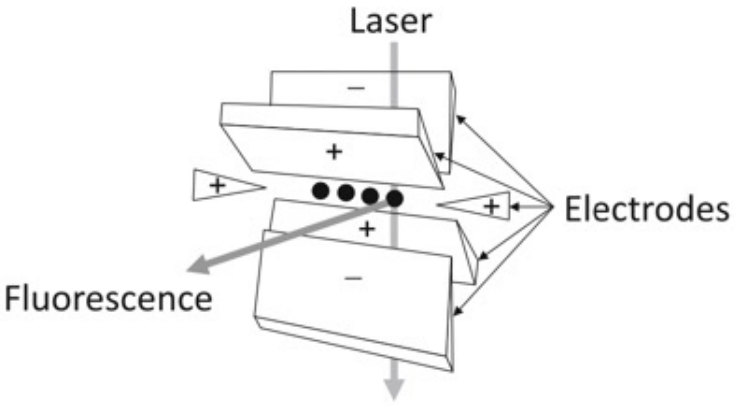

Figure 8: Trapped ion for quantum computing [135]

Trapped ions have a relatively long coherence time and interact easily with neigh-

bors for multiplexing. An ion chain is a one-dimensional array of many ions confined in their respective trap and forms a basis for multi-qubit quantum computing [19]. To manipulate the quantum states of the ions for computation, laser pulses are used. To reduce unwanted fluctuations of the qubit ions from photons or interactions with neighbors, the trapped ions need to be cooled down using laser cooling. For the read-out, a laser beam is applied to the ions to excite the energy state briefly. The ions emit photons while returning to the ground state and the information in the emitted photons is interpreted as the result of the quantum computation [135]. Collecting the emitted photons for readout is done through a lens and a photomultiplier, a device that transforms weak light signals into electric currents [19].

Neutral Atoms

The use of neutral atoms is the technology for quantum computing adopted by Zapata, a quantum computing company based in Massachusetts [217]. Neutral atoms are atoms that have an equal number of protons and electrons, resulting in a neutral net charge. These atoms can be trapped using the standing light waves of an optical lattice created by a network of crossed laser beams, called optical tweezers [135]. The trapped atoms are suspended in an ultrahigh vacuum through the laser cooling process to near absolute zero [40]. They are manipulated for quantum information processing using laser beams. Measurements are based on imaging near-resonant scattered light onto an electron multiplying CCD (Charged-Coupled Device) camera [190].

Neutral atoms are characterized by long coherence times and inherent scalability [190] and are gaining popularity in the research community [40]. The potential materials for realizing neutral atom quantum computing include alkaline-earth atoms and ytterbium.

Quantum Optics

Light is one of the most abundant resources in nature. Depending on the surrounding condition, light exhibits either classical mechanical attributes or quantum mechanical attributes. In quantum optics, the electromagnetic field of light is expressed in terms of light-quanta, which are discrete packets of light-wave, expressing the energy level of the field [69]. The quantum mechanical phenomena of light occur at room temperature and light is robust to noise and decoherence [10].

From the engineering perspective, optical quantum computers are easy to fabricate due to technological advances in fiber optics and lasers. Quantum optical systems can be created using lasers, mirrors, beamsplitters, waveplates, diffraction gratings, and other optical instruments. In quantum optical systems, nonlinear effects as well as linear effects are observed. Linear optics refers to optical systems in which the input light frequency and the output frequency are the same. In nonlinear optical systems, the frequency of the output light is different from that of the input light. These nonlinear effects have been incorporated into Xanadu's X8 and the available nonlinear quantum gates can be used in building quantum algorithms using Penny-Lane [26] [27].

Optical quantum computers have the advantage of operating at room temperature for binary and multi-valued logic, being integrable into the existing communications infrastructure for multiplexing and networking [233], robust to decoherence, and having low-latency. Optical signal processing is more efficient than electrical signal processing for amplification, multiplexing, demultiplexing, switching, filtering, and correlation [107].

[26] Python library by Xanadu for building quantum algorithms on Xanadu's QPUs.

Quantum Dots

Quantum dots are artificially fabricated nano-scale solid-state structures whose size does not exceed $1\mu m$ in each spatial direction [5]. In this structure the motion of charge carriers (electrons and holes) is limited in all three spatial dimensions [150]. Quantum dots are called artificial atoms and their electronic properties can be modified and controlled with modern electronic devices by applying external electromagnetic fields. The electrons confined in this structure exhibit their energy levels in discrete (quantized) units. Either the charge states or spin states of quantum dots can be used to realize quantum bits [150].

3.1.4 Quantum Transducers

Transducers are devices that convert one form of signal from one device to another form of signal that can be used in different devices [266]. Different processes and materials used for quantum computation, storage, and transmission need to be fused for information exchange. For example, with the fiber optic communications system, the electrical signals are converted into optical signals using optical transmitters, transmitted across the fiber optics network, and then converted back to electrical signals at the receiving end. Even within the same optical devices, the frequency and wavelengths of light signals may differ from one device to another. Transducers are used to convert the input energy of one form to the output energy of another form.

Quantum transducers are converters between different quantum state encoding schemes. This is especially necessary for long-distance quantum information transmission across communication networks where the electromagnetic wavelengths between the quantum computing devices and fiber optical networks are different. Various connection points at which quantum transducers may be needed are depicted in Figure

9 [27].

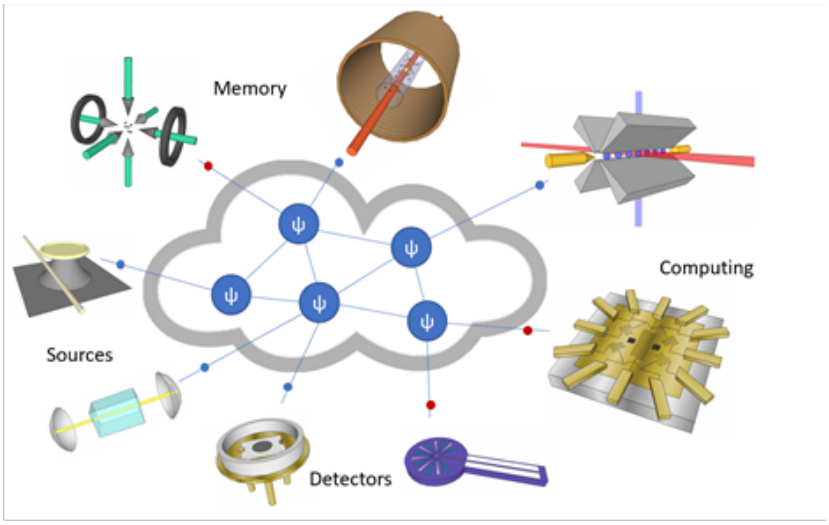

Figure 9: Hybrid quantum network

Fiber optic communication systems consist [107]:

- An optical transmitter to convert an electrical signal to an optical signal
- A cable containing several bundles of optical fibers
- Optical amplifiers to boost the power of the optical signal
- An optical receiver to reconvert the received optical signal back to the original transmitted electrical signal.

If we can build quantum computers based on quantum optics, then the conversion process from electrical signals to optical and the reverse process can be vastly simplified. This will reduce the complexity and number of transducers involved in the entire QuIC infrastructure.

[27]NIST website, credit: Ivana Slattery

3.1.5 Quantum Sensors

Sensors are devices that measure different quantities such as temperature, wave signals, bio-metrics, distances, and tomography. Quantum sensing is measuring these quantities using physical devices employing the use of quantum mechanical properties. According to the subcommittee on Quantum Information Science Committee on Science of the National Science and Technology Council, sensing is arguably the most mature subcategory of quantum technology [268].

Some of the existing quantum sensors identified by the committee include:

- *Atomic clocks for positioning, navigation, networking, and metrology.*
- *Atom interferometers, e.g., gravimeters for remote sensing and accelerometers for navigation.*
- *Optical magnetometers for bioscience, geoscience, and navigation.*
- *Devices utilizing quantum optical effects for local and remote sensing, networks, and fundamental science.*
- *Atomic electric field sensors, e.g., Rydberg atoms for GHz-THz radiation detection.*

In NSF's vision of QuIC, quantum sensors (metrology subsystems) for measuring physical quantities, equipped with quantum embedded systems are included [18]. In today's sensor technology, optical sensors are widely used because optical metrology has the properties of high precision and non-intrusiveness to the physical environment of objects being measured [10]. Some of the applications of optical metrology include semiconductor chip production, fiber optics cable defect check, and medicine [79].

Especially in medicine, optical fibers are used extensively for minimally invasive surgeries, endoscopes, biomedical sensors, and Optical Coherence Tomography (OTC)

[73]. Endoscopes are fiber optics cables that allow medical practitioners to view internal organs as seen in Figure 10.

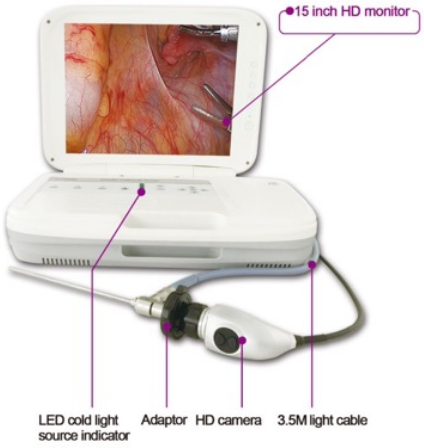

Figure 10: Endoscope: fiber optics are attached to a monitor. [image source]

For quantum sensors, several modes of quantum optics are being considered. They include [268]:

- *Quantum biophotonic: fundamental light–matter interactions and spectroscopy of biological systems*
- *quantum-enhanced remote sensing*
- *quantum spectroscopy*
- *subwavelength quantum microscopy*

In addition to standard quantum image processing, the future quantum chip will add to the functionality of image processing based on advanced quantum algorithms. In all these areas, quantum computers realized as embedded systems can enhance the quality of data collection or enable information processing. In all these areas, the quantum sensors' capabilities will be vastly enhanced with the use of embedded

quantum computing systems for data collection or information processing. This tendency to embed advanced machine learning and control mechanisms to multi-modal sensors is already leading in classical measurements so no doubt the same will happen in future advanced quantum sensors.

3.2 Optical Quantum Computing

The quantization of the electromagnetic field of light is termed quantum optics [236]. Optical quantum computing refers to the mode of quantum computing using quantum optics. In optical quantum computing, the energy of the electromagnetic field of light described in units of discrete energy called light-quanta is used as an information carrier [125]. Classical light is expressed in terms of waves and quantum light is expressed in terms of quantized packets of light energy, namely photons [10].

There have been suggested two different logic models of optical quantum computing: the qubit-based model [232, 76, 124, 236] and the continuous variable model [35, 59, 148, 185, 241]. The definition of the continuous variable model is using a physical quantum system whose quantum state space is infinite-dimensional [241]. In the physical realization of quantum computing, the infinite-dimensional nature of the continuous variable model never gets actualized since there is always the highest level of energy quanta observed. Hence, the physical implementation of the continuous variable model of quantum computing naturally realizes the multi-valued logic of computing, instead. There have been missing a practical way of actualizing a quantum optical system for the multi-valued logic of quantum computing.

In 2011, Scott Aaronson and Alex Arkhipov introduced the concept of Gaussian Boson Sampling (GBS) in the qubit-based model for realizing optical quantum computing. Their idea is to approach quantum computing from the sampling and search problem perspective rather than the decision problem perspective. In Xanadu's im-

plementation of GBS using quantum optics, each qumode contains a multiple number of photons instead of zero or one, naturally implementing the multi-valued logic of computing. Xanadu's X8 emerged as a hardware platform for the realization of multi-valued logic of computing.

In optical quantum computing, a guided path for the information-carrying light signal to travel through is created using a laser. Laser is an acronym for Light Amplification Stimulated Emission Radiation. The amplification is achieved through oscillation around a medium ring placed between two reflectors as seen in Figure 11. Hence a laser is an oscillator.

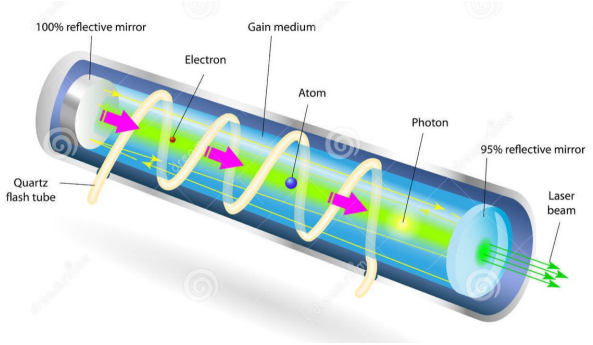

Figure 11: Light Amplification Stimulated Emission Radiation [image source]

When a coherent directed beam of light is emitted from the laser, the process called "squeezing" is applied to convert classical light to quantum [11]. The light beam is directed to pass through a guided path for quantum computations to be performed on. The light path is called quantum mode (qumode), which acts as a basic unit of computing just like digital or quantum bits. A single qumode can be interpreted as a wire in a quantum circuit. Information processing in a qumode is done in time not in space, the same as in standard quantum computing.

The quantum state of a qumode is described as a projective complex linear combination of the light-quanta [49]. Change of quantum states of the field constitutes information processing. The results of quantum computation in optical systems are

extracted via measuring the energy level of the state in number basis [241]. The energy level is interpreted as representing the number of photons in the system after all the computational operations are performed. The notion of photons is proposed as a convenient way of visualizing the energy units of the electromagnetic field. Dirac's original notion of light-quanta is the measure of energy levels of the field in an optical system [69].

These quantized levels of energy are used as computational basis states of optical quantum computing and they naturally implement multi-valued logic of information processing. One of the most notable advantages of optical quantum computing is that these quantized energy levels are exhibited at room temperature. Therefore, optics-based quantum computing is operable at room temperature [11].

Light is abundant in nature and known to be the fastest substance, providing ways for low-latency computation and communications. controllable for both binary and multi-valued computation because it operates at room temperature. Due to the unique properties of quantum optics, optical quantum computers are capable of realizing multi-valued logic of computing, i.e., each variable can have multiple basis states as opposed to just zero and one as in the binary model.

In the current framework of the qubit-based model, most of the algorithms use ancillary qubits for readout of the computation [3, 22, 57, 74, 255]. In optical QPUs, information processing and measurement are done on the same information carriers, bypassing the need for additional wires.

With the advance in optical technology due to fiber optics-based communication systems, more sophisticated optical instruments are being invented and fabricated. In the modern fabrication of VLSI, optical fibers are the material of choice for interconnects [115]. The components of optical interconnects include optical waveguides, vertical cavity surface emitting lasers, and multiple quantum well modulators. The

schematic of an optical system for global on-chip communication is shown in Figure 12.

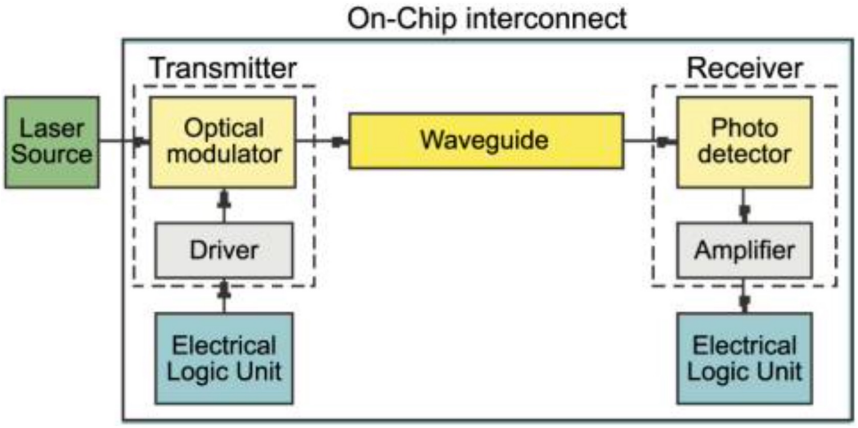

Figure 12: Schematic of optical system for global on-chip communication [115]

This is an example of areas in which technological advances in optics are being propelled. On-going advances in optical instrument technology in multiple areas will bring the cost and barrier down for the fabrication of highly sophisticated optical quantum computers.

3.2.1 Optical Quantum Processing Unit: X8

The Canadian company Xanadu has launched two QPUs using quantum optics called X8 with eight spatial qumodes and Borealis [28] with 216 time-division multiplexed qumodes.

X8 is made of silicon nitrade for photon conductance. It is housed in a conventional server at room temperature and is accessible to the user via Python-based Strawberry Fields from a personal computer. For building quantum algorithms, Xanadu offers a Python-based library, PennyLane [27].

A master controller, connected to a personal computer, controls the number of

[28] See Appendix C.

qumodes to be used, quantum circuit building, information processing, and readout of the measurement results back to the user. For computation, classical light is pumped into the qumodes on the chip, converted into quantum light, is sent through a series of quantum state changes based on a user-defined quantum circuit, filtered, and measured.

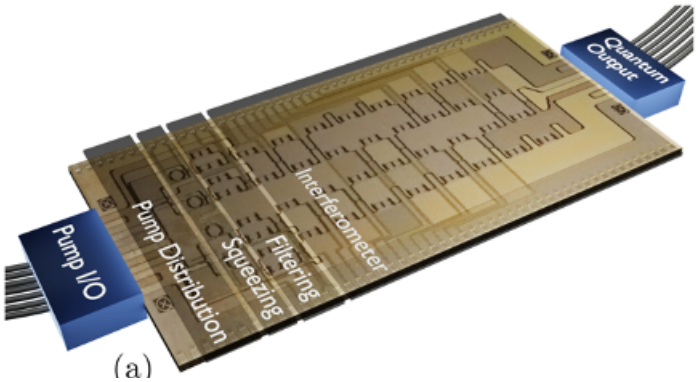

Figure 13: Xanadu's 8-qumode chip [11]

The functional components of X8 are Pump I/O, pump distribution, squeezing, filtering, interferometer, and programmable quantum gates. Pump I/O is a custom-modulated pump laser source that produces a regular pulse train. The pump distributor directs the photons in classical states to appropriate qumodes.

The process of squeezing converts the classical light into a quantum squeezed state. The quantum light is run through a filter to suppress unwanted light, passing only wavelengths close to the signal. The interferometer portion of the chip performs quantum circuit operations. Then the photon counter reads the number of photons in each qumode and sends the result of the computation back to the user.

The general information flow of light is depicted in Figure 14. In this figure, the red arrows indicate classical light propagated through light guides [29]. The first block indicates a block of squeezers realized with ring resonators. The squeezing block

[29]They can be viewed as information channels used as computational wires.

converts classical light into quantum light. The middle block indicates quantum gate operations controlled by classical registers. The measurement is done via photon detectors at the end of the circuit, measuring the exact photon count in each qumode as a computational result.

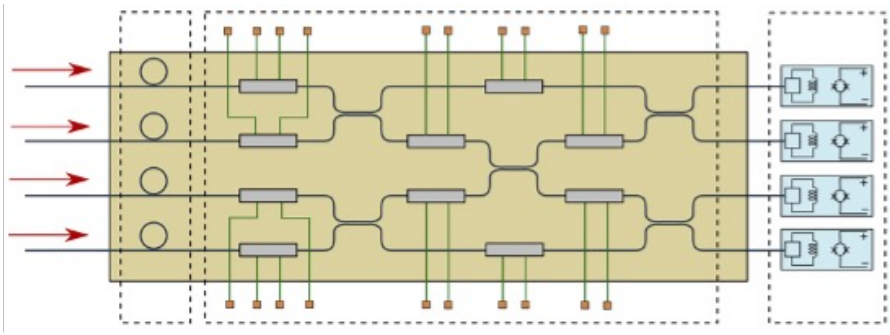

Figure 14: Optical quantum processing unit architecture [11]

Figure 15 illustrates the probability distribution of the number of photons in a qumode generated by X8. All the operations are set to the identity to discount for any quantum operational changes in photon count. The red continuous line indicates the theoretical prediction and the bars indicate the probabilities obtained from experimental samples. For the experiment, 12 runs of 105 samples are sampled [11].

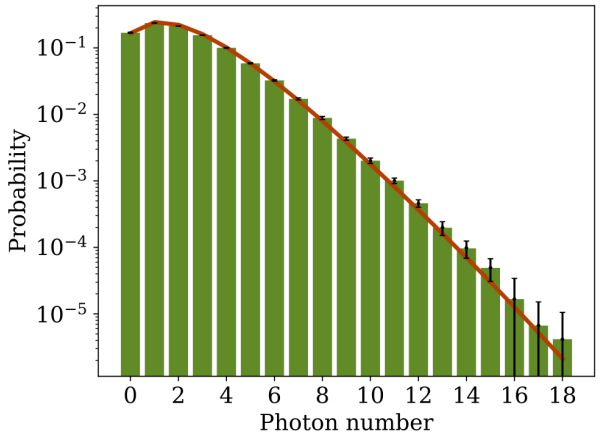

Figure 15: Probability distribution for the total number of photons generated by the device. [11]

3.2.2 Multiplexing

In an optical QPU, the information carrier light coming from a single source needs to be multiplexed to create multiple qumodes for computation. To go from the current NISQ era to the future universal fault-tolerant quantum computing era, multiplexing is one of the areas that need to be improved to accommodate the computing power required to process complex computational problems.

Optical multiplexing is widely used in fiber-optics communications systems. Xanadu's X8 uses spatial-division multiplexing and Borealis uses time-division multiplexing. It is my observation that the optical multiplexing knowledge base and techniques currently employed in communication systems can be adapted for optical QPUs.

Spatial-Division Multiplexing

Spatial-division multiplexing (SDM) is the most natural way of multiplexing a QPU by creating multiple optical guided paths (i.e., qumodes) for light signals to travel through. SDM is used in fiber optical communication systems to accommodate continuing increases in internet data traffic [10]. Within a single optical fiber, multiple optical channels are shared by confining each individual channel to a unique spatial location. These channels can operate at exactly the same wavelength as well as differing wavelengths [173].

Xanadu's X8 has etched-in light paths, each of which constitutes a qumode. The engineering challenge is the increased number of digital registers to control quantum gate operations on each qumode as the number of qumodes increases [34]. As steady technological advances continue in fiber-optical communications systems and laser-based optical sensors, some of the engineering challenges may be overcome in the

future.

Time-Division Multiplexing

This is a scheme that combines multiple time-delayed signals in a single optical channel [240]. To overcome the engineering challenges of dealing with digital registers as the number of qumodes increases, Xanadu's Borealis achieves 216 qumodes using time-division multiplexing [34]. From the same light source, when the pulses are applied with different time intervals, each pulse forms a qumode, which then collectively forms a multiplexed optical QPU.

Wavelength-Division Multiplexing

Wavelength-division multiplexing (WDM) is a technique used to transmit multiple channels at different wavelengths through the same fiber. Its use enhanced the capacity of fiber-optic communication systems so dramatically that data transmission at 1 Tbit/s was realized by 1996 [10]. The benefits of WDM as cited from [94] are :

- *Capacity upgrade: To upgrade the capacity of existing "point to point" fiber optics transmission links.*
- *Transparency: The light path can carry data in any transmission format like asynchronous and synchronous digital data or analog information.*
- *Wavelength reuse: Multiple light paths in the network can use the same wavelength if the wavelength is limited, as long as they do not overlap on any other link.*
- *Wavelength routing.*
- *Wavelength conversing.*

Although WDM is not currently employed in today's near-term QPUs, it remains a viable technique for multiplexing in future QPUs.

4 LOGIC IMPLEMENTATION: MULTI-VALUED LOGIC

The two models of quantum computing in the literature are the discrete variable model and the continuous variable model [120]. The discrete variable model mainly refers to the qubit-based model with two basis states $|0\rangle$ and $|1\rangle$. The continuous variable model refers to a model of quantum computing using a continuous variable system. A continuous variable system is a quantum system with an infinite-dimensional Hilbert space represented by bosonic [30] modes [31] [241]. The information carriers in quantum computing are electromagnetic fields expressed in discrete energy quanta. To implement the discrete variable (qubit-based) model of quantum computing, near subzero temperature is needed to select only the lowest two levels of quantized energy. Implementing the continuous variable model of quantum computing is not practically possible since any physical quantum system always comes with a finite-dimensional Hilbert space. A quantum system exhibits multiple levels of energy quanta, whose mathematical description leads to a finite-dimensional Hilbert space. The state of a natural quantum system is described with a finite number of multiple basis elements, naturally leading to the multi-valued logic of quantum computing.

The logic basis of digital computing is binary logic. This is due to the hardware configuration using the ON and OFF switches realized in transistors inside ICs [37]. As a natural extension of the transition from classical computing to quantum computing, most of the current research effort in industry is in realizing the binary logic of quantum computing [5, 15, 19, 20, 28, 34, 40, 45]. However, using the natural state of quantum mechanical systems wherein multiple levels of energy quanta are observed realizes the multi-valued logic of quantum computing [32]. Especially in

[30] Bosons are elementary particles with spins expressed in integer increments.
[31] Channels or wires.
[32] See Section 4.2.2

optical quantum computing where the components are built with inexpensive optical instruments, this natural realization of the multi-valued logic of computing occurs at room temperature. This new paradigm of computing away from the need to encode information in binary bits may be an added benefit of quantum computing, when we employ optical quantum computing.

4.1 Multi-Valued Logic

The binary logic of computing is based on the semiconductor-based switching circuit technology [8]. Therefore, in modeling real-life problems for the purpose of computation in digital computing, we encode them in binary logic. For example, the ternary states YES, NO, and UNDECIDED cannot be directly computed but are encoded as $00, 01$, and 10 with two bits for example. The decimal logic of $0, 1, \ldots, 10$ are encoded as $0000, 0001, \ldots, 1010$ using four bits.

There has been a growing number of researchers, realizing the limitations of the binary logic of computing both in logic implementation as well as in representing complex digital systems [8, 163, 168, 198, 219]. When signals in a circuit represent more levels than just 0 and 1 with higher radix multi-valued logic, they contain more information leading to data compression. The multi-valued logic of computing can be viewed as compact representations of the information content of signals. Higher radix means fewer digits to represent binary information [8]. The formula for the number of digits needed to represent binary information is given by [8]

$$d = \lceil log_p\,(2^n) \rceil \tag{1}$$

where d is the number of digits, p is the radix, and n is the number of bits for binary logic. For example, in decimal logic of radix-10, only one digit is needed to express

8 while three digits are needed in binary logic as seen in Equation (2).

$$\lceil log_{10}\left(2^3\right)\rceil = \lceil log_{10}\left(8\right)\rceil = \lceil 0.90\rceil = 1 \tag{2}$$

Hence, the basic computational units used in higher radix systems contain more information than in binary logic. This results in the complexity reduction in space, time, and interconnects [8]. In addition, if a memory cell stores more than one bit of information, this increases the density of information per area unit [8].

Optical computers, DNA computers, and quantum computers have been recognized as potential technologies for realizing the multi-valued logic of computing. Optical quantum computing is emerging as a viable hardware platform for actualizing this new mode of information processing, which is not yet widely recognized. Perhaps in the future, the research community will recognize optical quantum computing as a practical hardware platform for the realization of the multi-valued logic of computing and there will be active research in quantum algorithms based on high-radix logic beyond the binary.

4.2 Quantum Computing

In quantum computing, the quantum state of the electromagnetic field of the system is used as a basis of quantum computation. Controlled change of states of the field constitutes "quantum computation". The energy state of the electromagnetic field is expressed in terms of the discrete quanta levels of excited states. The quantized energy levels are used as basis states for computation. Currently, the QPUs by IBM, Google, Rigetti, Honeywell, IonQ, and Zapata all use the binary logic model of quantum computing. This means that the ground state and one higher-up excited state are used as the qubit basis states $|0\rangle$ and $|1\rangle$. Xanadu's X8 implements the

multi-valued model of quantum computing, which means more energy levels of the electromagnetic field are used as basis states. However, Xanadu's goal is to realize the binary logic model of quantum computing by limiting the number of naturally occurring energy quanta [34]. With more recognition of the natural realization of the multi-valued logic in optical quantum computing, the research community may embrace its full expressive power in processing computational problems.

Here, we examine the mathematical formalism of the qubit-based model and the multi-valued model of quantum computing.

Under the mathematical formalism of quantum mechanics, any possible state of a quantum mechanical system is represented by a unit vector in a complex projective Hilbert space [178]. A Hilbert space is a vector space equipped with the inner product operation. To represent a quantum state vector in a Hilbert space, it is customary to use Bra-ket notation, also known as Dirac notation. Column vectors are denoted by kets $|x\rangle$ and the corresponding complex conjugate transpose row vectors by bras $\langle x|$. Suppose $|x\rangle$ is an $n-$dimensional column vector, $|x\rangle = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} \in \mathbb{C}^n$. Its conjugate transpose row vector is $\langle x| = |x\rangle^\dagger = \begin{bmatrix} x_0^*, x_1^*, \dots x_{n-1}^* \end{bmatrix}$. The inner product of two vectors $|x\rangle, |y\rangle \in \mathbb{C}^n$ is expressed as

$$\langle x|y\rangle = |x\rangle^\dagger |y\rangle = \begin{bmatrix} x_0^*, x_1^*, \dots x_{n-1}^* \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \sum_{k=0}^{n-1} x_k^* y_k \tag{3}$$

The inner product of the vector $|x\rangle$ with itself is then

$$\langle x|x\rangle = \sum_{k=0}^{n-1} x_k^* x_k = \sum_{k=0}^{n-1} \|x_k\|^2 = \| \, |x\rangle \, \|^2, \tag{4}$$

which is the norm squared of the original vector. Then the length (norm) of $|x\rangle$ is simply the square root of its inner product.

$$\| \, |x\rangle \, \| = \sqrt{\| \, |x\rangle \, \|^2} = \sqrt{\sum_{k=0}^{n-1} \|x_k\|^2} = \sqrt{\langle x|x\rangle}. \tag{5}$$

This allows the distance between two vectors to be defined as the norm of the difference vector of the two:

$$\begin{aligned} dist(|x\rangle \, , |y\rangle) &= \| \, |x - y\rangle \, \| \\ &= \sqrt{\langle x - y|x - y\rangle}, \end{aligned} \tag{6}$$

which turns the space into a metric space [47]. We can also define the angle between two vectors, using the formula [243]

$$\begin{aligned} &\langle x|y\rangle = \| \, |x\rangle \, \| \| \, |y\rangle \, \| \cos\theta \\ &\implies \theta = \arccos\left(\frac{\langle x|y\rangle}{\| \, |x\rangle \, \| \| \, |y\rangle \, \|} \right) \end{aligned} \tag{7}$$

A clear definition of the distance and angle between two vectors as shown in Equations (6) and (7) allows the generalization of linear algebra and calculus within a quantum state space.

The projective part comes from the nature of quantum mechanical systems where the coefficients $c_k \in \mathbb{C}$ defining a quantum state have to meet the condition $\sum \|c_k\|^2 = 1$. For any quantum state that does not meet the condition, we can simply normalize

it with the normalizing constant $\frac{1}{\sum \|c_k\|^2}$. For a quantum state expressed with a unit vector $|\psi\rangle$, it is equivalent to any other state of the form $\gamma |\psi\rangle$ for some non-zero $\gamma \in \mathbb{C}$ i.e., $\frac{1}{\gamma}$ is the normalizing constant for $\gamma |\psi\rangle \in \mathbb{C}^n$, where $n = 2$ for the qubit model and n =cutoff dimension for the multi-valued model. Here, $\gamma \in \mathbb{C}^*$ is called the global phase, and the original complex Hilbert space is modded out by $\mathbb{C}^*$, producing a complex "projective" Hilbert space [47].

4.2.1 Binary logic Model (Qubit-based Model) [178]

The binary logic model of quantum computing is a natural extension of binary logic digital computing. In realizing the binary logic in a quantum system, the lowest two levels of energy of the system are selected to represent the basis states $|0\rangle$ and $|1\rangle$. The computational basis $\{|0\rangle, |1\rangle\}$ is a quantum version of digital computing bits 0 and 1, hence is called a quantum bit.

Quantum Bits

Embed the classical states $0, 1 \in \mathbb{R}$ as two-dimensional vectors in $\mathbb{C}^2$ using Dirac notation:

$$0 \mapsto \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle, 1 \mapsto \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \tag{8}$$

Any quantum state in this quantum state space is expressed as a projective complex linear combination of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{9}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. This complex linear combination is called superposition. Here α and β are called probability amplitudes.

The concept of "projective" is based on the notion of the global phase in quantum mechanics, where two quantum states are considered to be equal when they differ by a non-zero complex factor.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \gamma\left(\alpha\,|0\rangle + \beta\,|1\rangle\right), \gamma \in \mathbb{C}^*. \tag{10}$$

Any non-unit vector is equivalent to its normalized version, i.e., of length 1. The state space of a qubit, then, is a 2-dimensional complex Hilbert space minus the origin, which is then modded out by the space of all possible global phase values. We take out the origin $(0,0)$ from the space since the $|\alpha|^2 + |\beta|^2 = 1$ condition prohibits both α and β to be zero at the same time. The resulting space is $\mathbb{C}^2 - (0,0)$, modded out by $\mathbb{C}^*$, which is the space of all possible global phase values, obtaining [47]

$$(\mathbb{C}^2 - \{(0,0)\})/\mathbb{C}^* \cong (S^3 \times \mathbb{R}^+)/(U(1) \times \mathbb{R}^+) = S^3/U(1) \cong \mathbb{C}P^1. \tag{11}$$

Geometrically the resulting projective space is of the surface of a spherical space, equivalent to S^2. It is called the Bloch Sphere with $|0\rangle$ as the North Pole and $|1\rangle$ as the South Pole.

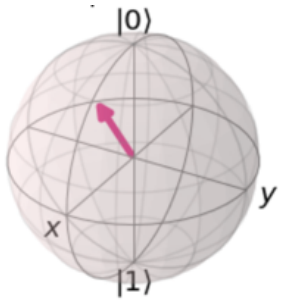

Figure 16: Quantum state space of a qubit (Generated on IBM simulator using Qiskit)

Now any quantum state, which is a point on the Bloch sphere, can be expressed in spherical coordinates: $|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle$, where θ represents the angle

with the $z-$axis and ϕ with the $x-$axis. The value $e^{i\phi}$ is called relative phase [178]. Any state of one qubit would lie on the Bloch sphere as a point, a representative of the entire global phase space $\mathbb{C}^*$.

Compare this qubit space of the Bloch sphere to the classical bit space of $\{0, 1\}$, consisting of two points. This expanded space used for quantum information processing is one of the areas believed to be giving more power to quantum computing than classical.

Multiple Qubits

Increasing the computing power is achieved by adding more qubits. In multiple-qubit systems, the state space is the tensor product space of the individual qubit spaces and the computational basis, the set of all possible combinations of individual qubit computational basis, i.e., the tensor product of the basis states. For example, in a two-qubit system, the set of all possible combinations of the computational basis states would be $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$ where $\otimes$ denotes a tensor product. For simplicity, it is customary to omit the tensor product notation: $|\psi\rangle \otimes |\phi\rangle = |\psi\phi\rangle$. In vector representation,

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

Notice the size of the computational basis and the length of the basis vectors is $2^2 = 4$. The state space of n qubits is a 2^n-dimensional Hilbert space with its computational basis being the set of all combinations of the state each qubit can

be found in. For example, the computational basis of a 3-qubit system is the set $\{|000\rangle, |001\rangle, \ldots, |111\rangle\}$ of size $2^3 = 8$, with each basis vector of length 8. The quantum state in a system of multiple qubits is represented by the tensor product of the individual qubits. Let $|\psi_k\rangle = \alpha_k |0\rangle + \beta_k |1\rangle, k \in \{0, 1, \ldots, n-1\}$ be the quantum state of the k^{th} qubit. Then any quantum state in the whole system can be expressed as

$$\phi_0 \otimes \phi_1 \otimes \ldots \phi_{n-1} = \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \ldots \otimes \begin{bmatrix} \alpha_{n-1} \\ \beta_{n-1} \end{bmatrix} = \begin{bmatrix} \alpha_0\alpha_1 \ldots \alpha_{n-1} \\ \alpha_0\alpha_1 \ldots \beta_{n-1} \\ \vdots \\ \beta_0\beta_1 \ldots \alpha_{n-1} \\ \beta_0\beta_1 \ldots \beta_{n-1} \end{bmatrix}. \tag{13}$$

Each $|\psi_k\rangle$ is a vector of length 2, hence the tensor product of n vectors of length 2 gives us a vector of length 2^n. The resulting tensor product space is The geometric representation of an $n-$qubit system is then [47]

$$\mathbb{C}P^n = (\mathbb{C}^{n+1} - \{(0,0)\})/\mathbb{C}^* \cong (S^{2n+1} \times \mathbb{R}^+)/(U(1) \times \mathbb{R}^+) = S^{2n+1}/U(1) \cong S^{2n}. \tag{14}$$

Quantum Gates

Quantum gates inducing a change of states for information processing are represented by 2×2 rotation matrices on the Bloch sphere. They are of the form $UU^\dagger = U^\dagger U = I$ called unitary matrices, which means $U^\dagger = U^{-1}$. The inverse operation of a unitary matrix is its reverse rotation on the Bloch sphere. Notice all unitary matrices are

invertible, hence all quantum gates in the qubit model are reversible, unlike some classical gates such as the AND and OR gates. The Hadamard gate is used to put the zero state $|0\rangle$ in uniform superposition of $|0\rangle$ and $|1\rangle$.

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\right) = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{15}$$

The Pauli-X, Y, Z gates are used to induce half rotations about the $x, y, z-$axis respectively.

$$X = NOT = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, Y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}, Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \tag{16}$$

Parametrized gates are used to rotate a qubit state according to the given parameter. Examples of parameterized gates are

$$Rx(\theta) = \begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix}, P(\theta) = \begin{bmatrix}1 & 0\\ 0 & e^{i\theta}\end{bmatrix} \tag{17}$$

Quantum gates acting on multiple qubits are tensor products of component unitary gates. Suppose we have n-qubits for computation, each with a unitary gate acting on it. The corresponding unitary gates acting on the entire system are represented by $2^n \times 2^n$ matrices as tensor products of n 2×2 unitary matrices. For example, in a two-qubit system, when we apply the Hadamard gate on the first and the T gate on the second, the tensor product of the two matrices acts on the entire system:

$$
\begin{aligned}
H \otimes T &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} & 1 \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \\ 1 \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} & -1 \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} & 0 & e^{i\frac{\pi}{4}} \\ 1 & 0 & -1 & 0 \\ 0 & e^{i\frac{\pi}{4}} & 0 & -e^{i\frac{\pi}{4}} \end{bmatrix}
\end{aligned} \tag{18}
$$

In a multi-qubit system, we can use one or more qubits as a control for an operation on other qubits. For example, the Pauli$-X$ gate which is equivalent to the classical NOT gate on one qubit can be controlled by another qubit.

Phase Kickback [29]

Unlike in classical computing, the quantum state of the control qubit is affected by the state of the other qubit, on which the controlled operation is performed. This phenomenon called phase kickback, unique to quantum computing, is used to record the evolution of the state change of a quantum circuit on the ancillary qubit. Phase kickback plays an important role in many quantum algorithms including quantum machine learning.

Let U be an arbitrary unitary matrix and $|\psi_k\rangle$ be one of its eigenvectors. Then for its corresponding complex eigenvalue $\lambda_k = e^{i\alpha}$ expressed in Euler formula, U acting on $|\psi_k\rangle$ is equal to multiplying the eigenvector by its eigenvalue $e^{i\alpha} : U|\psi_k\rangle = \lambda_k|\psi_k\rangle =$

$e^{i\alpha}|\psi_k\rangle = e^{i\alpha}I|\psi_k\rangle$. Hence the effect of the gate operation U on one of its eigenvectors is equivalent to the identity operation multiplied by its corresponding eigenvector. However, by the projective nature of the quantum state space, the eigenvalue acts as a global phase, returning us the original state as though no operation was applied. That picture changes when we apply a controlled U gate instead. Consider a two-qubit circuit where an arbitrary unitary gate U is applied to the second qubit controlled by the first. The U operation will be applied only when the control qubit is in the state $|1\rangle$. The Hadamard gate is applied to the first qubit, initialized to the $|0\rangle$ state, to obtain the uniform superposition of $|0\rangle$ and $|1\rangle$.

The states $|0\rangle$ and $|1\rangle$ can be expressed by their outer product matrices up to phase $\frac{1}{\sqrt{2}}$. The operation of the U gate on the second qubit, when its state is one of the eigenvectors of U, conditioned on the $|1\rangle$ state of the first qubit is then given by

$$
\begin{aligned}
&|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{i\alpha}I \\
&= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix}
\end{aligned}
$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{19}$$

Notice the first matrix which applies $e^{i\alpha}$ rotation on $|1\rangle$ on the first qubit and the identity matrix on the second. It is as though the operation of $U = e^{i\alpha} I$ on the second qubit is kicked back to the first qubit when its state is $|1\rangle$.

When this operation is applied to $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |\psi\rangle$, we get

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} |\psi\rangle = \frac{|0\rangle + e^{i\alpha} |1\rangle}{\sqrt{2}} \otimes |\psi\rangle \tag{20}$$

where the original state $|\psi\rangle$ is not affected and the phase $e^{i\alpha}$ is "kicked back" to the $|1\rangle$ state of the first qubit.

Measurement

The last stage of a quantum circuit is measurement, a readout of computational results classically. It is a projection of the quantum computational resulting state onto one of the classical computational basis. Let

$$|\psi\rangle = c_0 |00\ldots0\rangle + c_1 |00\ldots1\rangle + \ldots + c_{2^n-1} |11\ldots1\rangle \tag{21}$$

be the final state of computation in an $n-$qubit system. The probability of getting the k^{th} computational basis state is given by "Born Rule" [178] as

$$prob(k) = \frac{|c_k|^2}{\sum_j |c_j|^2} = \langle\phi|k\rangle \langle k|\phi\rangle = |\langle k|\phi\rangle|^2 \tag{22}$$

In the circuit diagram in Figure 17, each line represents the evolution of the state of one qubit in time, boxes represent quantum gates, and the black arrow

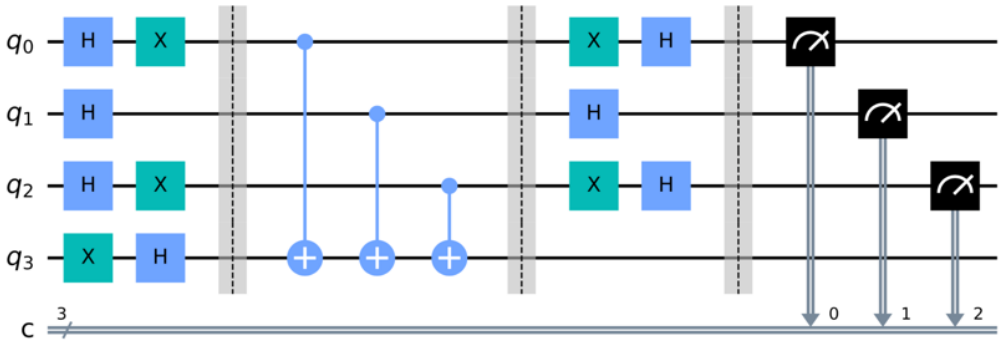

Figure 17: Quantum circuit diagram with measurement (Generated on IBM simulator using Qiskit)

boxes represent measurement. Readout qubits are connected to classical wires for measurement.

In each instance of measurement, we get one of the computational basis elements once. To extract the probability value of the k^{th} element, which is a projection of $|\psi\rangle$ onto $|k\rangle$, we perform multiple shots of the circuit operation. Figure 18 is a histogram of a 4-qubit circuit run for 1000 shots. The higher the number of shots, the closer the result will be to the true probability distribution over the computational basis states.

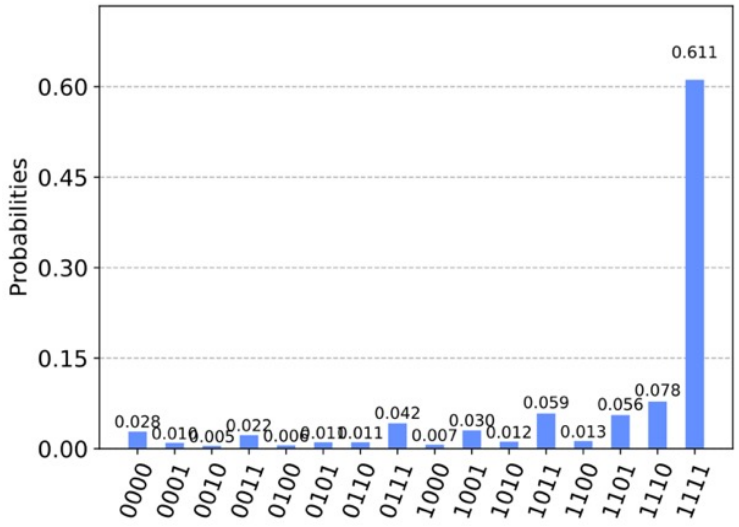

Figure 18: Measurement result of 1000 shots (Generated on IBM simulator using Qiskit)

4.2.2 Multi-valued Logic Model

Another model of quantum computing in the literature is the continuous variable model of quantum computing wherein the computational space is infinite-dimensional [35, 120, 241]. The continuous variable model of quantum computing can be implemented using optical systems, trapped ions, or microwaves [6, 35, 59, 75, 120, 148, 170, 185, 186]. In optical quantum computing, the quantum state of a quantum harmonic oscillator is expressed as a complex linear combination of multiple levels of energy quanta, naturally realizing the multi-valued logic of quantum computing [49]. Theoretically, the quantum state of an optical system is expressed as the sum of an infinite number of basis [241]. However, in the physical implementation of this model, there is always the highest level of energy of real-life physical systems. This means that the quantum state of the system is expressed with a finite number of basis states. Depending on the highest level of energy observed in the system, an optical quantum system is described based on multi-valued basis states $|0\rangle, |1\rangle, \ldots, |n\rangle$ as an implementation of radix$-n$ logic. Thus the "continuous variable" theoretical model is always realized as the multi-valued logic of computing.

In Xanadu's X8, for each instance of computation, the user defines the maximum energy level used for information processing. The maximum energy level allowed for computation defines the dimension of the computational space, and it is called the "cutoff dimension". The notion of "cutoff dimension" is used to control the dimension of the computational space and the size of the output vectors.

More detailed explanations of optical quantum computing and its natural realization of the multi-valued logic of quantum computing are given in following pages.

Quantum Modes

A quantum harmonic oscillator is used as one computational unit in optical quantum computing [35, 125, 241]. It is constructed using a laser for oscillation and a resonator for feedback-loop [11]. The quantum state of its electromagnetic field is used as a unit of quantum computing (qumode) [10]. It can be viewed as a wire The state is described as a projective complex linear combination of the basis states, which are quantized packets of energy [33].

Classical light is converted into quantum light through the process called squeezing [11, 120]. The idea of squeezing is based on the Uncertainty Principle of quantum mechanics. It states that the product of the standard deviation σ_x of position x and the standard deviation σ_p of momentum p has a clearly defined lower bound: $\sigma_x \sigma_p \geq \frac{\hbar}{2}$ for the reduced Planck's constant $\hbar = \frac{h}{2\pi}$. Squeezing the momentum standard deviation σ_p as close to zero as possible will increase the value of the position standard deviation because of the $\frac{\hbar}{2}$ lower bound. A qumode initialized to the vacuum state has no photon present and the Gaussian distribution of the position and momentum of the zero-photon state is clearly defined. The visual representation on the position-momentum plane is depicted in Figure 19.

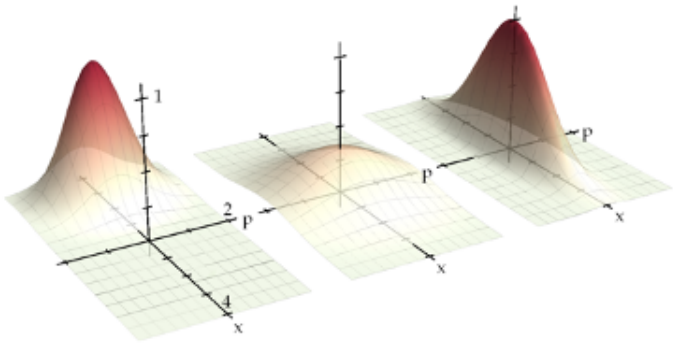

Figure 19: Quantizing classical light: squeezed light [174]

[33] which are called photons, when light is viewed as particles rather than waves.

Then the state of the optical system is expressed as a complex linear combination of number basis. The squeezing process parameterized with the squeezing parameter z on the vacuum state $|0\rangle$ is given by [241]

$$\begin{aligned} S(z)\,|0\rangle &= \frac{1}{\sqrt{\cosh z}} \sum_{n=0}^{\infty} \frac{\sqrt{(2n)!}}{2^n n!} \tanh^n z\,|2n\rangle \\ &= \frac{1}{\sqrt{\cosh z}} \left(|0\rangle + \tanh(z)\,|2\rangle + \cdots + \tanh^n(z)\,|2n\rangle + \cdots\right) \end{aligned} \tag{23}$$

Notice the squeezing operation converted classical light into an infinite sum of even number Fock basis [34], i.e., quantum light:

$$|0\rangle \to \sqrt{\frac{2}{e^z + e^{-z}}} \left(c_0\,|0\rangle + c_2\,|2\rangle + \ldots c_{2n}\,|2n\rangle + \ldots\right) \tag{24}$$

where $c_{2k} = \frac{\sqrt{(2k)!}}{2^k k!} \left(\frac{e^z - e^{-z}}{e^z + e^{-z}}\right)^k$ for $k = 0, 1, 2, \ldots n \ldots$, as k goes to infinity. Here, the states $|0\rangle, |2\rangle, \ldots, |2n\rangle, \ldots$ represent the number of photons found in the optical system.

Once classical light is converted into quantum light as a complex projective linear combination of even Fock (number) basis, optical quantum gate operations are applied to create even and odd Fock (number) basis. Repeated application of the constructor creates a higher number of photons [35]. Notice the process of squeezing processes even numbers of photons in the system. The application of constructor and annihilator operators produces odd numbers of photons.

In phase space representation of quantum optics, the basis states can be represented as functions of the position and momentum observables using the Wigner function [44].

[34] Please see the section **Fock basis** on this page.

[35] See Section 5. Matrix Representation of Optical Quantum Gates

Fock Basis (Number Basis)

A photon is a form of electromagnetic radiation, whose properties are described quantum mechanically [10]. The position wave function describes the light wave electromagnetic strength of the qumode depending on the number of photons present. It is a complex-valued function on real-valued variables: $\Psi : \mathbb{R} \to \mathbb{C} : x \mapsto \alpha$ [120]. As the light waves from different photons interact either constructively or destructively, the wave function with more than one photon displays the constructive and destructive interactions between the light waves of the photons.

Another way of describing an optical quantum system is via the phase space representation wherein both the position and momentum observables are simultaneously used [35, 120, 241]. Consider the lowest energy state of a qumode, where there is no photon present in the system, i.e., the vacuum state. Its wave function of the position variable is given by [44]

$$\Psi_0(x) = \frac{1}{\sqrt[4]{\pi}\sqrt{a}} e^{-\frac{x^2}{2a^2}}. \tag{25}$$

Using the wave function of the position variable and the momentum variable, we can derive a Gaussian distribution of the state on the position - momentum plane. The Wigner function as a function of position and momentum is given [44] by

$$W(x,p) = \frac{p}{h} = \frac{1}{h}\int_{-\infty}^{\infty} e^{-\frac{ipy}{\hbar}} \Psi\left(x + \frac{y}{2}\right) \Psi^*\left(x - \frac{y}{2}\right) dy \tag{26}$$

where h is the Planck constant and $\hbar$ is the reduced Planck constant. The position wave function $\Psi(x)$ for each number of photons and its momentum p is used on the position-momentum plane.

Using the wave function $\Psi_0(x)$, we can derive the Wigner function of the lowest

energy state [44].

$$
\begin{aligned}
W_0(x,p) &= \frac{1}{h}\int \frac{1}{a\sqrt{\pi}} e^{-\frac{ipy}{\hbar}} e^{-\frac{(x+\frac{y}{2})^2}{2a^2}} \left(e^{-(x-\frac{y}{2})^2/2a^2}\right)^* dy \\
&= \frac{2}{h} e^{-\left(\left(\frac{ap}{\hbar}\right)^2+\left(\frac{x}{2}\right)^2\right)}
\end{aligned}
\tag{27}
$$

where $a = \sqrt{\frac{\hbar}{m\omega}}$, m = angular momentum, and ω = angular frequency. It can be simplified by setting $a, h = 1$ as $W_0(x,p) = 2e^{(-(\frac{p}{\hbar})^2 - x^2)}$. The plot of the function $W_0(x,p)$ on the $xp-$plane is shown in Figure 2.5.

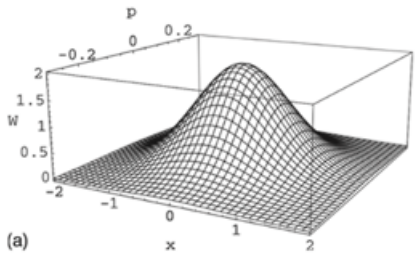

Figure 20: The Wigner function $W_0(x,p)$ on the $xp-$plane [44]

As the value σ_x spreads farther in the $x-$axis, the mathematical description of the qumode can be expressed as a superposition of the numbers of photons that can be found in the system, using Fock basis i.e., number basis. Fock basis is derived from the Wigner functions of the position and momentum variables depending on the number of photons present in the system. The projection of the resulting Wigner functions $W(x,p)$ onto the x-axis gives the probability amplitude of the position x and onto the p-axis gives that of the momentum p. The resulting plot of each energy state is depicted in Figure 21.

The collection of the projection of $W(x,p)$ onto the $xp-$ plane for each energy state is called Fock basis or number basis: $\{|0\rangle, |1\rangle, \ldots, |n\rangle, \ldots\}$ [120].

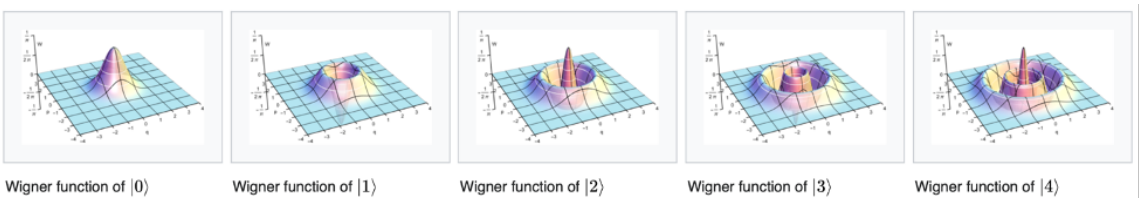

Figure 21: Fock basis [image source]

Quantum States

The quantum state $|\psi\rangle$ of the system of a qumode is expressed as an infinite sum (superposition) of Fock basis states [17]:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle + \ldots + c_n |n\rangle + \ldots, \text{ where } \sum_{k=0}^{\infty} \|c_k\|^2 = 1 \tag{28}$$

and c_k is the probability amplitude of basis $|k\rangle$. The probability of finding the qumode with $k-$ photons is given by $\|c_k\|^2$ when $|\psi\rangle$ is normalized. A single qumode, realized with a harmonic oscillator system, can be in zero particle state $|0\rangle$, 1 particle state $|1\rangle$, 2 particle state $|2\rangle$, so on, and an infinite number of particle state. The coefficients c_k represent the probability amplitude, which gives the probability distribution at measurement.

In a multiple qumode system, the quantum state of the system is represented by the tensor product of the individual qumode states. The resulting space is yet another infinite-dimensional Hilbert space. Let the quantum states of two qumodes be represented each as $|\phi\rangle$ and $|\psi\rangle$.

$$\begin{aligned} |\phi\rangle &= b_0 |0\rangle + b_1 |1\rangle + \ldots + b_n |n\rangle + \ldots \\ |\psi\rangle &= c_0 |0\rangle + c_1 |1\rangle + \ldots + c_n |n\rangle + \ldots. \end{aligned} \tag{29}$$

Then the tensor product of the entire system is

$$
\begin{aligned}
|\phi\rangle \otimes |\psi\rangle = (|\phi\rangle = b_0 |0\rangle + b_1 |1\rangle + \ldots + b_n |n\rangle + \ldots) \\
&\otimes (c_0 |0\rangle + c_1 |1\rangle + \ldots + c_n |n\rangle + \ldots) \\
&= b_0 c_0 |00\rangle + b_0 c_1 |01\rangle + \ldots + b_0 c_n |0n\rangle + \ldots \\
&+ b_1 c_0 |10\rangle + b_1 c_1 |11\rangle + \ldots + b_1 c_n |1n\rangle + \ldots \\
&+ \ldots \\
&+ b_n c_0 |n0\rangle + b_n c_1 |n1\rangle + \ldots + b_n c_n |nn\rangle + \ldots \\
&+ \ldots
\end{aligned}
\tag{30}
$$

Cutoff Dimension

In physical reality, we know that we are not going to have an infinite number of photons in any system, thus also in a qumode. Depending on the technology, there is the highest level of energy observed: in X8, it is 18 photons and in Borealis, 219 [36]. That means the quantum state of an optical system is expressed as a linear combination of a finite number of basis elements.

For each instance of computation using Xanadu's X8, the user is asked to define the "cutoff dimension" as one of the parameters for building a quantum circuit.

```
dev_fock = qml.device("strawberryfields.fock", wires=2, cutoff_dim=2)
```

Figure 22: Number of wires and cutoff dimension as parameters for building a quantum circuit in Xanadu's X8

In order for us to better understand the notion of the cutoff dimension, we look at the definition of the basis and the dimension of a vector space.

Definition 4.1 (Basis) *A basis is a set $\mathcal{B}$ of vectors in a vector space $\mathcal{V}$ wherein*

[36] Although registering events with up to 219 photons are observed, a mean photon count is 125.

every element of $\mathcal{V}$ may be written in a unique way as a finite linear combination of elements of $\mathcal{B}$.

Definition 4.2 (Cutoff Dimension) *The cutoff dimension is the number of Fock basis* [37] *elements in describing the quantum state of an optical system* [38] *for each instance of computation.*

By Theorem 1.3 of [166], every set $F = \{f_1, f_2, \ldots, f_n\}$ of linearly independent vectors in an $n-$dimensional vector space V forms a basis of V. Let n be the cutoff dimension. Then the quantum state $|\psi\rangle$ is in superposition of the set $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$.

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle + \ldots + c_{n-1} |n-1\rangle, \text{ where } \sum_{k=0}^{n-1} \|c_k\|^2 = 1 \tag{31}$$

In vector representation, the state is expressed as a vector of size n =cutoff dimension as

$$|\psi\rangle = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = c_0 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + c_{n-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{32}$$

Notice the vectors representing $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$ are linearly independent.

Suppose cutoff dimension = 3 on a 2 qumode-system. Then for each qumode, the state of the quantum system is represented by a superposition of three Fock basis states: $|\phi\rangle = b_0 |0\rangle + b_1 |1\rangle + b_2 |2\rangle$ and $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle + c_2 |2\rangle$. The quantum state of the entire system is the tensor product of both qumodes:

[37] See P. 55 Fock basis.

[38] So far, it is realized only in Xanadu's X8.

$$
\begin{aligned}
|\phi\rangle \otimes |\psi\rangle = b_0c_0\,|00\rangle + b_0c_1\,|01\rangle + b_0c_2\,|02\rangle \\
+ b_1c_0\,|10\rangle + b_1c_1\,|11\rangle + b_1c_2\,|12\rangle \\
+ b_2c_0\,|20\rangle + b_2c_1\,|21\rangle + b_2c_2\,|22\rangle
\end{aligned} \tag{33}
$$

Notice the Fock basis $\{|00\rangle, |01\rangle, |02\rangle, |10\rangle, |11\rangle, |12\rangle, |20\rangle, |21\rangle, |22\rangle\}$ has $9 = 3^2$ elements.

In an $m-$qumode system, the quantum state of the entire system defined by cutoff dimension n is

$$
\begin{aligned}
&|\psi_0\rangle \otimes |\psi_1\rangle \otimes \ldots \otimes |\psi_{m-1}\rangle = \\
&d_0\,|00\ldots0\rangle + d_1\,|00\ldots1\rangle + c_{n^m-1}\,|n-1, n-1, \ldots, n-1\rangle
\end{aligned} \tag{34}
$$

Assertion 1 The cutoff dimension defines the dimension of the computational space. By definition, the cutoff dimension defines the number of Fock basis elements for a quantum optical system. The number of basis elements is the dimension of the computational space in optical quantum computing. Hence the cutoff dimension defines the dimension of the computational space.

Definition 4.3 (Radix) *The number of independent digits used in the number system is known as Radix or Base of the number system* [272].

Assertion 2 The dimension of the computational space leads to radix$-n$ multi-valued logic.

By definition, the number of basis elements is the dimension of the computational space. For the quantum state $|\psi\rangle = c_0\,|0\rangle + c_1\,|1\rangle + \ldots + c_{n-1}\,|n-1\rangle$ expressed as a complex projective linear combination of $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$ has n independent digits, realizing radix$-n$ logic.

The quantum computing research community has been focusing on algorithm development in binary logic because it was a natural extension of digital computing. However, with working optical QPUs demonstrating multi-valued computational spaces, we are given additional tools to expand our computational problems in more accurate ways than in binary logic. Theoretically, the quantum state space of an optical harmonic oscillator, realizing optical quantum computing, is infinite-dimensional. In real-life physical systems, the space is finite-dimensional with a clearly defined level of the highest energy quanta. In using optical quantum computers, there comes the power for users to define the dimension of the computational space by setting the cutoff dimension. In a multiple qumode system, the size of the computational basis equals n^m where n represents the cutoff dimension and m, the number of qumodes used. Additionally, using different measurement methods in optical quantum computing gives the user the possibility to determine the size of the output of a quantum circuit. This added flexibility of "defining" the size of the computational space and the size of the output vectors can be used as a tool for the exploration of different solution spaces for a given computational problem.

In Xanadu's Borealis, we are able to realize radix-126 logic. With the advance of future optical technology, the increased number of varying light-quanta will mean an increased dimension of computational spaces to the user. With the notion of cutoff dimension, at each instance of quantum computation, the user can limit the number of light-quanta as the number of computational basis states [49].

The term "cutoff dimension" is used as one of the parameters for building quantum circuits with PennyLane on Xanadu's X8. The documentation does not go into detail as to its meaning. The only place wherein the term cutoff dimension is referred to is Killoran's "Continuous variable quantum neural networks". Recognizing the use of cutoff dimension to determine the dimension of computational space, leading to

the realization of the multi-valued logic of computing is one of my contributions.

Quantum Gates

In phase space representation of quantum space in optical quantum computing, the standard quantum gates are represented by matrix exponential of the form $U = e^{-iHt}$ [120], where H is the Hamiltonian of the system, describing the total energy as the sum of kinetic and potential energy.

When describing the action of an optical quantum gate with Hamiltonian expressed as some matrix M, the resulting operation U on the quantum state is represented by the matrix exponential expressions:

$$\begin{aligned} U &= exp(itM) \\ &= \sum_{k=0}^{\infty} \frac{(itM)^k}{k!} \\ &= I + itM + \frac{(itM)^2}{2!} + \cdots + \frac{(itM)^n}{n!} + \cdots \end{aligned} \tag{35}$$

The Hamiltonians for various optical gates are expressed using the constructor $\hat{a}^\dagger$ and the annihilator $\hat{a}$ of photons within a qumode [239]. For the purpose of quantum computation, setting a boundary to the energy levels allowed in the system is crucial since we cannot have an infinite-dimensional matrix. In constructing optical quantum gates, the building blocks are the constructor $\hat{a}^\dagger$ and the annihilator $\hat{a}$ as seen in the standard gates below.

Squeezer with parameter $z : S(z) = exp\left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right)$

Rotation with parameter $\phi : R(\phi) = exp\left(i\phi\hat{a}^\dagger\hat{a}\right)$

Displacement with parameter $\alpha : D(\alpha) = exp\left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right)$ [114]

In addition to these gates, optical quantum computing offers nonlinear gates which are not available in the binary logic (qubit-based) model. These nonlinear gates

include the Kerr gate and the cross Kerr gate. The nonlinear gates implemented in PennyLane include Controlled addition, Controlled phase, Cross Kerr, Cubic phase, Kerr, Quadratic phase, and Two mode squeezing.

The matrix representation of the constructor and annihilator and various gates is presented in Section 5 of this dissertation.

Measurement

Measurement is done by counting the number of photons present in each qumode with a photon detector. At the end of the quantum circuit, a photon detector is attached to each qumode. The quantum state $|\psi_K\rangle = \sum_{i=0}^{n-1} c_i |i\rangle$ of the k^{th} qumode, where $n =$ cutoff dimension, collapses to one of the classical states $i \in \{0, 1, \ldots, n-1\}$ when measured with a photon detector.

Let $|\psi\rangle = \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix}$ be the quantum state of a single qubit or qumode system after desired quantum computational operations are performed. The expectation value measurement method returns $\langle\psi| M |\psi\rangle$ where the operator M is usually the Pauli$-X$, Pauli$-Y$, or Pauli$-Z$ gate. The expectation value of the Pauli$-X$ matrix is

$$\begin{aligned} \langle\psi| X |\psi\rangle &= \begin{bmatrix} \psi_0^* & \psi_1^* \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix} \\ &= \begin{bmatrix} \psi_0^* & \psi_1^* \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_0 \end{bmatrix} \\ &= \psi_0^* \psi_1 + \psi_1^* \psi_0 \\ &= 2\left(Re(\psi_0)Re(\psi_1) + Im(\psi_0)Im(\psi_1)\right) \in \mathbb{R}, \end{aligned} \tag{36}$$

which is a real number. Let $|\psi_k\rangle$ be the quantum state of the k^{th} qubit in a multi-qubit system with m qubits. The expectation value measurement method yields

$[\langle\psi_0| M |\psi_0\rangle, \langle\psi_1| M |\psi_1\rangle, \ldots, \langle\psi_{m-1}| M |\psi_{m-1}\rangle]$ of length m, where M is the quantum gate used for measurement. Similarly, we get a vector of size m for an $m-$qumode system.

Xanadu's QML tool, PennyLane, additionally offers the probability measurement method, which yields an approximation of the probability distribution over the Fock basis states. With the cutoff dimension n and the number of qumodes m, the output vector is of size n^m. Therefore, when using $m-$qumodes for a QML circuit, the user is availed the flexibility of getting an output vector of size m or of size n^m.

Gaussian Boson Sampling [1]

There are two elementary particles in the universe: bosons and fermions. Bosons are subatomic particles with integer spin (i.e., $0, 1, 2, \ldots$) [39] and include photons and carriers of nuclear forces [1]. Fermions are subatomic particles with odd half-integral spin (i.e., $\frac{1}{2}, \frac{3}{2}, \ldots$) [40] and include quarks and electrons [1]. The idea is to create identical photons, pass them through optical elements, and measure the number of photons in each qumode. Aaronson and Arkhipov's model is a model of quantum computing with non-interacting bosons. Aaronson and Alex Arkhipov introduced the concept of Gaussian Boson Sampling (GBS) as a practical way of implementing optical quantum computing. For the physical implementation of their model, they suggest the use of linear optics: to generate identical photons, send them through a linear-optical network composed of beamsplitters and phase shifters (rotation gates), and then measure the number of photons in each qumode [1]. The term stems from Gaussian matrices related to the process, photons being bosons, and the model involving sampling bosons (photons). Although the concept was based on the qubit-based

[39] Encyclopedia Britannica.
[40] Encyclopedia Britannica.

model, the idea of utilizing identical photons as information carriers was utilized in Xanadu's implementation of quantum computing, using quantum optics.

The general information flow of GBS in Xanadu's QPU is demonstrated in Figure 23. In this figure, the blocks R denote rotation gates and the blocks BS denote beamsplitters, forming a linear interferometer. In Xanadu's X8, nonlinear phenomena in quantum optics are incorporated as nonlinear quantum gates as well.

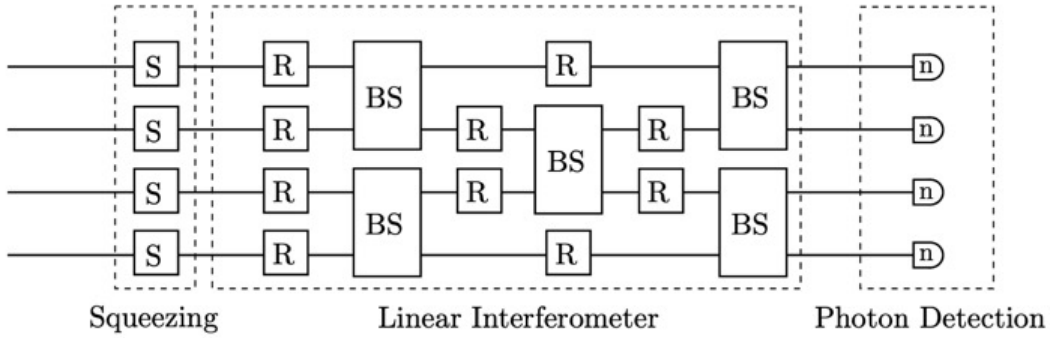

Figure 23: Linear optical architecture with a single linear interferometer [11]

The inspiration comes from Galton's board, where n identical balls are dropped one by one into equally spaced slots as seen in Figure 24. The results exhibit Gaussian distribution [41] with sufficient sample size.

Aaronson and Arkhipov's model, unlike Galton's board, has different locations for photon sources, namely different qumodes. The model constructs m qumodes and n photons where $n \leq m$. Each qumode is allowed to have either 0 or 1 photon, hence this is a qubit-based model [42].

The computational basis state $|S\rangle$ is given by

$$|S\rangle = |s_1, s_2, \ldots, s_m\rangle \text{ where } s_i \in \{0, 1\} \sum s_i = n \tag{37}$$

[41] More specifically Binomial Distribution of which Gaussian is a subset.

[42] If each qumode is allowed to have more than one photon, say p, it would be an implementation of radix$-p$ logic.

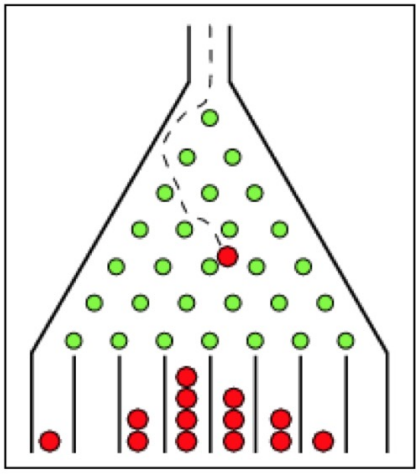

Figure 24: Galton's board as a demonstration of Binomial Distribution

One variant of $|S\rangle$ may be $|1,0,0,1,\ldots,1\rangle$ where there are n ones.

By Lemma 14 of [194], any $m \times m$ unitary matrix can be decomposed as a product $U = U_T \circ \cdots \circ U_1$ where each U_T is an optical element. An optical element is a unitary matrix that acts non-trivially on at most two qumodes and as the identity on the remaining $m-2$ qumodes. The action of an optical element (e.g., beamsplitters and/or rotation gates) on a basis element is given by

$$|s_1, s_2, \ldots, s_m\rangle \mapsto e^{i\theta s_i} |s_1, s_2, \ldots, s_m\rangle \tag{38}$$

The quantum state $|\psi\rangle$ of the entire system is

$$|\psi\rangle = \sum \alpha_s |S\rangle \text{ where } \sum \|\alpha_s\|^2 = 1, \tag{39}$$

meaning it is in superposition of all the possible combinations of qumodes containing one photon.

Measurement is done with photon detectors to count the number of photons found in each qumode. The probability of finding the state in the basis element $|S\rangle$

is given by

$$Pr(S) = \|\alpha_s\|^2 = |\langle\phi|S\rangle|^2. \tag{40}$$

Notice relaxing the condition $n \leq m$ and allowing more than one photon per qumode changes the original GBS into a multi-valued model. In Xanadu's X8, the maximum photon count experimentally observed is 18 where the number of qumodes is 8. Hence X8 can be viewed as an implementation of Aaronson and Arkhipov's model with the modification that $n \geq m$, $n = 18 \times 8 = 144$, and $m = 8$. By the nature of the quantized electromagnetic fields, the multi-valued model of GBS naturally occurs in optical quantum computing whereas the qubit-based model requires additional engineering feats such as temperature control [34].

Example: Optical Quantum Auto-encoder

This section is an illustration of the use of multi-valued logic and quantum computing. Under the qubit-based model, the basis states are only $|0\rangle$ and $|1\rangle$ for a single qumode. When using the multi-valued logic in optical quantum computing, we can employ the ternary logic or higher-radix logic of computing. The circuit in Figure 25 takes advantage of ternary logic by setting the cutoff dimension at three. We can realize radix-n logic by setting the cutoff dimension at n on a single qumode.

Auto-encoder is a network, composed of an encoder that encodes data in a compact manner and a decoder that retrieves the original data. The classical and quantum hybrid auto-encoder, proposed by Killoran et al. [120] is composed of a classical encoder and a one-qumode quantum decoder. The classical encoder compresses the input vectors of length 3 into vectors of length 2. The quantum decoder takes in the vectors of length 2 and outputs vectors of length 3. The training process is to find optimal parameters for the encoding matrices and the decoding quantum

circuit so that the output vectors are as close to the original data as possible. The architecture of the proposed hybrid network is shown in Figure 25.

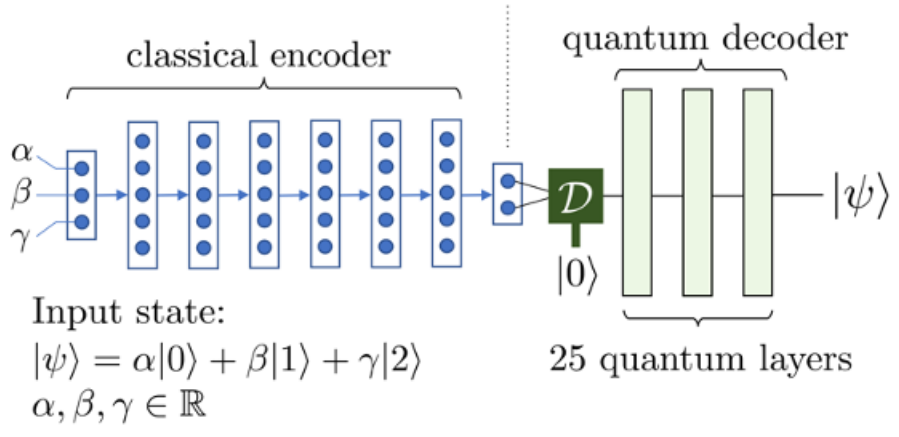

Figure 25: Auto-encoder hybrid network architecture [120]

The classical encoder has 6 hidden layers, each with 5 neurons using Exponential Linear Unit (ELU) as an activation function. The quantum circuit takes in the vectors of length 2 and encodes them in quantum states by using the entries of the vectors as parameters of a displacement gate. The quantum decoder is composed of 25 layers of quantum neural network using squeezers, rotation gates, displacement gates, and Kerr gates. To get output vectors of dimension 3, I set the cutoff dimension to 3 and use the probability method for measurement. The TensorFlow [43]plug-in feature of PennyLane allows us to convert the PennyLane quantum layers into a Keras layer. Then Keras's built-in loss functions and optimizers can be used for training. Mean

[43] *"TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors."* source: Wikipedia

Squared Error loss function was used for the data vector x and the output vector y.

$$MSE = ((x_0 - y_0)^2 + (x_1 - y_1)^2 + (x_2 - y_2)^2)/3 \tag{41}$$

One of the Keras' built-in optimizers, Adam optimizer, with learning rate (i.e., step size) = 0.01 is used as an optimizer. After 300 epochs with batch size 50, the model achieves loss= 0.0736 and accuracy= 90.54%.

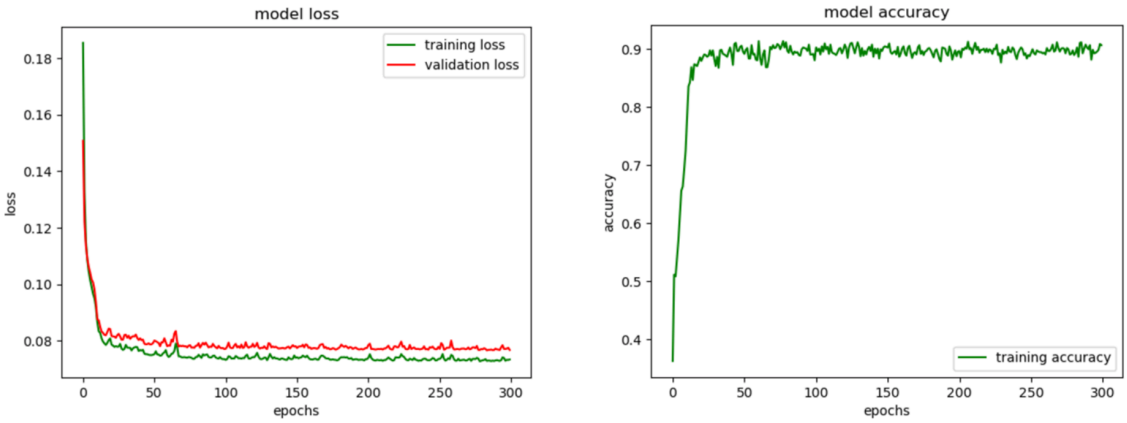

Figure 26: Auto-encoder experimental results

This was Killoran et al.'s demonstration of the power of the use of the cutoff dimension n as the realization of the radix$-n$ multi-valued logic of computation. However, the authors did not relate the results to the concepts or notations of multi-valued logic.

5 DERIVATION OF MATRIX REPRESENTATION OF OPTICAL QUANTUM GATES

In this section, I present a matrix representation of optical quantum gates in phase space representation. Explicit matrix representation of optical quantum gates in the literature is limited to the qubit-model [124], the position and momentum observables representation [120], and the Heisenberg representation.

For example, the unitary matrix representing a beamsplitter operating on two-qumodes under the qubit-model is given by

$$B(\theta,\phi) = \begin{bmatrix} \cos\theta & -e^{i\phi}\sin\theta \\ e^{-i\phi}\sin\theta & \cos\theta \end{bmatrix} \tag{42}$$

by Knill, Laflamme, and Milburn [124]. The entries in the corresponding vector of length two would represent the state of each qumode, implementing a qubit.

The same beamsplitter is represented by yet another matrix by Aaronson and Arkiphov [1]:

$$\begin{bmatrix} \alpha'_S \\ \alpha'_T \end{bmatrix} := \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \alpha_S \\ \alpha_T \end{bmatrix} \tag{43}$$

Here, α_S is the probability amplitude for the state $|10\rangle$ and α_T is for $|01\rangle$. This matrix can be viewed as a special case of the matrix presented by Knill, Laflamme, and Mulburn where $\phi = 0$.

Yet another matrix representation of the beamsplitter presented by Samuel Braunstein and Peter van Loock has a different form [35]:

$$B(\theta) = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \tag{44}$$

The matrix representation of the beamsplitter under the position and momentum observables is given by Nathan Killoran et al. in [120]:

$$\begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix} \mapsto \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix} \tag{45}$$

In Xanadu's documentation, yet another version of the matrix representation of the beamplitter is given: the Heisenberg representation. The Heisenberg representation is a formalism in which the operators depend on time while the wave functions do not [262]. The matrix representation under this formalism is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & -\cos\phi\sin\theta & -\sin\phi\sin\theta \\ 0 & 0 & \cos\theta & \sin\phi\sin\theta & -\cos\phi\sin\theta \\ 0 & \cos\phi\sin\theta & -\sin\phi\sin\theta & \cos\theta & 0 \\ 0 & \sin\phi\sin\theta & \cos\phi\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{46}$$

These matrices under different formalism are fundamentally different from the formula for the beamsplitter under the phase space representation of optical quantum computing wherein the quantum state of a system is expressed in terms of Fock basis [241]:

$$B(\theta,\phi) = exp\left(\theta\left(e^{i\phi}\hat{a}\hat{b}^{\dagger} + e^{-i\phi}\hat{a}^{\dagger}\hat{b}\right)\right) \tag{47}$$

Numerical matrices are usually the starting point to derive practical realizations of quantum circuits from elementary gates [116, 117, 118, 171]. Therefore, my presented result can hopefully help to derive circuits of ternary basic universal gates that

can be used for Grover and other traditional quantum algorithms, implemented in the presented variant of the quantum formalism. For instance, matrices and equations for elementary ternary quantum gates are presented in [118].

In phase space representation of the quantum state in optical quantum computing, a quantum state $|\Psi\rangle$ is a projective complex linear combination of Fock basis $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$ when the cutoff dimension is set to be n. In vector forms, $|\Psi\rangle$ is a vector of length n with each k^{th} entry representing the quasi-probability of the state being found in $|k\rangle$.

The matrix representing a quantum gate is a matrix exponential $U = exp(-itH)$ [120] of size $n \times n$, where the Hamiltonian H is an $n \times n$ matrix itself. The matrix exponential is given by

$$U = exp(-iHt) = I + (-iHt) + \frac{(-iHt)^2}{2!} + \cdots + \frac{(-iHt)^n}{n!} + \cdots \tag{48}$$

where I is the identity matrix of size $n \times n$.

The list of optical quantum gates available on Xanadu's website includes:

- Identity
- Beamplitter: $B(\theta, \phi) = exp\left(\theta\left(e^{i\phi}\hat{a}\hat{b}^\dagger + e^{-i\phi}\hat{a}^\dagger\hat{b}\right)\right)$
- Controlled Addition: $CX(s) = e^{-is\hat{x}\otimes\frac{\hat{p}}{\hbar}}$
- Controlled Phase: $CZ(s) = e^{is\hat{x}\otimes\frac{\hat{x}}{\hbar}}$
- Cross Kerr: $CK(\kappa) = e^{i\kappa\hat{n}_1\hat{n}_2}$
- Cubic Phase: $V(\gamma) = e^{i\frac{\gamma}{3}\hat{x}^3/\hbar}$
- Displacement: $D(\alpha) = exp\left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right)$

- Kerr: $K(\kappa) = e^{ik\hat{n}^2}$
- Quadratic Phase: $P(s) = e^{i\frac{s}{2}\hat{x}^2/\hbar}$
- Rotation: $R(\phi) = exp\left(i\phi\hat{a}^\dagger\hat{a}\right)$
- Squeezing: $S(z) = exp\left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right)$
- Two Mode Squeezing: $exp\left(r\left(e^{-i\phi}\hat{a}\hat{b} - e^{i\phi}\hat{a}^\dagger\hat{b}^\dagger\right)\right)$

The Hamiltonians for various optical gates are expressed using the constructor $\hat{a}^\dagger$ and the annihilator $\hat{a}$ of photons within a qumode [239]. The constructor can be interpreted as a mathematical expression of the action of exciting the energy level of the electromagnetic field to the next excited state or emission of a photon. Similarly, the annihilator describes the action of decrementing the energy level by one or absorption of a photon. The constructor and annihilator operators are building blocks for mathematically describing the Hamiltonian of the system to express various operations as optical quantum gates. Therefore, the matrix representation of these operators is essential in describing optical quantum gates. Some of the standard optical quantum gates in phase space representation [44] and their matrix exponentials are shown below [45] [114]:

Squeezer with parameter z:

$$\begin{aligned} S(z) &= exp\left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right) \\ &= I + \left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right) + \frac{\left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right)^2}{2!} + \cdots + \frac{\left(\frac{z^*\hat{a}^2 + z\hat{a}^{\dagger 2}}{2}\right)^n}{n!} + \cdots \end{aligned} \tag{49}$$

[44] operating on quantum states expressed in terms of Fock basis

[45] In the reference paper, just the exponential formulas are listed, not the explicit matrix exponential expressions.

Rotation with parameter ϕ:

$$\begin{aligned} R(\phi) &= exp\left(i\phi\hat{a}^\dagger\hat{a}\right) \\ &= I + \left(i\phi\hat{a}^\dagger\hat{a}\right) + \frac{\left(i\phi\hat{a}^\dagger\hat{a}\right)^2}{2!} + \cdots + \frac{\left(i\phi\hat{a}^\dagger\hat{a}\right)^n}{n!} + \cdots \end{aligned} \tag{50}$$

Displacement with parameter α:

$$\begin{aligned} D(\alpha) &= exp\left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right) \\ &= I + \left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right) + \frac{\left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right)^2}{2!} + \cdots + \frac{\left(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\right)^n}{n!} + \cdots \end{aligned} \tag{51}$$

Equations (37) through (39) clearly illustrate that in order to find the matrix exponential of these gates, the matrix representation of the constructor and the annihilator is foundational.

The constructor operator $\hat{a}^\dagger$ increases the energy level $|k\rangle$ by one to $|k+1\rangle$ with the coefficient $\sqrt{k+1}$. The mathematical definition of the constructor is [35, 241]

$$\hat{a}^\dagger |k\rangle = \sqrt{k+1}\,|k+1\rangle \;\text{ for } k \geq 0. \tag{52}$$

In a quantum system with cutoff dimension n, $\hat{a}^\dagger$ is represented by an $n \times n$ matrix as

$$\hat{a}^\dagger = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \sqrt{1} & 0 & 0 & \dots & 0 & 0 \\ 0 & \sqrt{2} & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{3} & \dots & 0 & 0 \\ \vdots & & \dots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & \sqrt{n-1} & 0 \end{bmatrix}. \tag{53}$$

Notice this matrix is not linearly independent. However, the matrix exponential form with the first term being the identity matrix renders all optical quantum gates linearly independent. With cutoff dimension n, the vector representation of the state $|k\rangle$ is of length n with all zeros but one in the k^{th} entry. The action of the constructor on the Fock state $|2\rangle$ returns us the Fock state $|3\rangle$ as shown in Equation (54).

$$\hat{a}^\dagger |2\rangle = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \sqrt{1} & 0 & 0 & \dots & 0 & 0 \\ 0 & \sqrt{2} & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{3} & \dots & 0 & 0 \\ \vdots & & \dots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & \sqrt{n-1} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sqrt{3} \\ \vdots \\ 0 \end{bmatrix} = \sqrt{3} |3\rangle . \tag{54}$$

Notice $\hat{a}^\dagger$ indeed allows a Fock basis state to construct the next Fock basis state.

$$\hat{a}^\dagger |0\rangle = \sqrt{1} |1\rangle , \ \hat{a}^\dagger |1\rangle = \sqrt{2} |2\rangle , \dots , \hat{a}^\dagger |n\rangle = \sqrt{n+1} |n+1\rangle , \dots \tag{55}$$

The annihilator $\hat{a}$ is the inverse operator of the constructor $\hat{a}^\dagger$ where it decrements the energy level of the basis state by one with the coefficient $\sqrt{k}$. The annihilator $\hat{a}$ is defined by [35, 241]

$$\hat{a} |0\rangle = 0, \hat{a} |k\rangle = \sqrt{k} |k-1\rangle \ \text{ for } k \geq 1. \tag{56}$$

The matrix representation of $\hat{a}$ is the conjugate transpose of the constructor $\hat{a}^\dagger$. Here,

since the coefficients are all real numbers, it simply is the transpose of $\hat{a}^\dagger$ shown as

$$\hat{a} = \begin{bmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ \vdots & & \dots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sqrt{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{57}$$

An example of the application of its matrix representation on the state $|3\rangle$ is

$$\hat{a}\,|3\rangle = \begin{bmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ \vdots & & \dots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sqrt{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sqrt{3} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sqrt{3}\,|2\rangle\,. \tag{58}$$

Notice $\hat{a}$ indeed annihilates a single photon in a qumode from a $3-$photon state to a $2-$photon state as seen in $\hat{a}\,|3\rangle = \sqrt{3}\,|2\rangle$.

The product of $\hat{a}^\dagger$ and $\hat{a}$ returns the square matrix called the number operator $\hat{n}$ [35, 241], whose eigenstates are the Fock basis states $|k\rangle$.

$$\hat{n} = \hat{a}^\dagger \hat{a} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 2 & 0 & \dots & 0 \\ \vdots & & \dots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & n-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & n-1 \end{bmatrix} \tag{59}$$

The matrix clearly shows $\hat{n}\,|k\rangle = k\,|k\rangle$ where $k \in \{0, 1, \dots, n-1\}$ for cutoff dimension= n, illustrating that the number basis states $|k\rangle$ are eigenstates of the number operator [241].

5.1 Qubit-based Gates

For each instance of computation, the dimension of the computational space is determined by the cutoff dimension n. Then the quantum state of the system is represented by a vector of length n. The corresponding matrices representing quantum gate operations are of size $n \times n$.

In this section are presented 2×2 matrices for cutoff dimension 2, representing binary logic, i.e., qubit-based quantum computing.

Displacement Gate

The matrix representation of displacement with parameter α is given by:

$$\begin{aligned} D(\alpha) &= exp(\alpha \hat{a}^\dagger - \alpha^* \hat{a}) \\ &= exp\left(\alpha \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} - \alpha^* \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \end{aligned}$$

$$
\begin{aligned}
&= exp\left(\begin{bmatrix} 0 & -\alpha^* \\ \alpha & 0 \end{bmatrix}\right) \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\alpha^* \\ \alpha & 0 \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} 0 & -\alpha^* \\ \alpha & 0 \end{bmatrix}^2 + \cdots + \frac{1}{n!}\begin{bmatrix} 0 & -\alpha^* \\ \alpha & 0 \end{bmatrix}^n + \cdots \qquad (60) \\
&= \begin{bmatrix} a & b \\ c & a \end{bmatrix}
\end{aligned}
$$

where the diagonal entries are the same. The expressions for the entries a, b, and c are given in Equations (61), (62), and (63).

$$
\begin{aligned}
a &= 1 + \frac{-\alpha\alpha^*}{2!} + \frac{(\alpha\alpha^*)^2}{4!} + \frac{-(\alpha\alpha^*)^3}{6!} + \cdots \\
&= \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} \sqrt{\alpha\alpha^*}^{2k} \qquad (61) \\
&= \cos\sqrt{\alpha\alpha^*}
\end{aligned}
$$

$$
\begin{aligned}
b &= -\alpha^* + \frac{\alpha\alpha^{*2}}{3!} - \frac{\alpha^2\alpha^{*3}}{5!} + \frac{\alpha^3\alpha^{*4}}{7!} + \cdots \\
&= -\alpha^*\left(1 - \frac{\alpha\alpha^*}{3!} + \frac{(\alpha\alpha^*)^2}{5!} - \frac{(\alpha\alpha^*)^3}{7!} + \cdots\right) \qquad (62) \\
&= -\alpha^* \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} (\alpha\alpha^*)^k
\end{aligned}
$$

$$
\begin{aligned}
c &= \alpha - \frac{\alpha^2\alpha^*}{3!} + \frac{\alpha^3\alpha^{*2}}{5!} - \frac{\alpha^4\alpha^{*3}}{7!} + \cdots \\
&= \alpha\left(1 - \frac{\alpha\alpha^*}{3!} + \frac{(\alpha\alpha^*)^2}{5!} - \frac{(\alpha\alpha^*)^3}{7!} + \cdots\right) \qquad (63) \\
&= \alpha \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} (\alpha\alpha^*)^k
\end{aligned}
$$

Rotation Gate

The matrix representation of the rotation with parameter ϕ is given by:

$$\begin{aligned}
R(\phi) &= exp\left(i\phi\hat{a}^\dagger\hat{a}\right) \\
&= exp\left(i\phi\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\right) \\
&= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & i\phi\end{bmatrix} + \frac{1}{2!}\begin{bmatrix}0 & 0\\ 0 & i\phi\end{bmatrix}^2 + \cdots + \frac{1}{n!}\begin{bmatrix}0 & 0\\ 0 & i\phi\end{bmatrix}^n + \cdots \\
&= \begin{bmatrix}1 & 0\\ 0 & 1 + \frac{(i\phi)^2}{2} + \frac{(i\phi)^3}{3!} + \cdots + \frac{(i\phi)^n}{n} + \cdots\end{bmatrix} \\
&= \begin{bmatrix}1 & 0\\ 0 & e^{i\phi}\end{bmatrix}
\end{aligned} \tag{64}$$

With the rotation gate, we can implement Pauli-Z gate by setting $\phi = \pi$.

$$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi}\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & \cos\pi + i\sin\pi\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = PauliZ \tag{65}$$

Setting $\phi = \frac{\pi}{2}$ implements the S gate or $\sqrt{Z}$ gate.

$$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix} \tag{66}$$

Setting $\phi = \frac{\pi}{4}$ implements the T gate, also known as $\sqrt[4]{Z}$

$$\begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix} \tag{67}$$

Two-qumode Gates

Beamsplitters are 2-qumode gates over a pair of qumodes with parameters θ and ϕ. For the matrix representation of beamsplitters, we decompose each term to find our Hamiltonian matrix to find its matrix exponential:

$$\begin{aligned} B(\theta, \phi) &= exp\left(\theta\left(e^{i\phi}\hat{a}\hat{b}^\dagger + e^{-i\phi}\hat{a}^\dagger\hat{b}\right)\right) \\ &= exp\left(\theta e^{i\phi}\hat{a}\hat{b}^\dagger + \theta e^{-i\phi}\hat{a}^\dagger\hat{b}\right) \end{aligned} \tag{68}$$

In the expressions $\hat{a}\hat{b}^\dagger$ and $\hat{a}^\dagger\hat{b}$ one operator is applied to the first qumode while the other is to the second. In a qumode system, the application of an operator to one qumode while no operator is applied to the other equates the tensor product of the operator and the identity operator to the other.

$$\theta e^{i\phi}\hat{a}\hat{b}^\dagger = \theta e^{i\phi}\left(\hat{a}\otimes I\right)\left(I\otimes\hat{b}^\dagger\right) \tag{69}$$

Hence, we calculate the matrices $\hat{a}\hat{b}^\dagger$ and $\hat{a}^\dagger\hat{b}$ first. The term $\hat{a}\hat{b}^\dagger$ represents an application of the annihilator on the first qumode and the constructor on the second qumode. While the annihilator $\hat{a}$ is applied to the first qumode, the operation on the second is the identity matrix. Then the application of the annihilator on the first qumode is the tensor product of the annihilator and the identity matrix is shown as:

$$\begin{aligned} \hat{a}\otimes I &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 1\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ 0\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 0\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{70}$$

The application of the constructor $\hat{b}^\dagger$ on the second qumode while the identity operation is applied to the first is

$$\begin{aligned} I \otimes \hat{b}^\dagger &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & 0 \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ 0 \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & 1 \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{71}$$

Multiplying them together, we get

$$\theta e^{i\phi} \hat{a} \hat{b}^\dagger = \theta e^{i\phi} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \theta e^{i\phi} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{72}$$

By the same procedure, we get

$$\begin{aligned} \theta e^{-i\phi} \hat{a}^\dagger \hat{b} &= \theta e^{-i\phi} \left(\hat{a}^\dagger \otimes I \right) \left(I \otimes \hat{b} \right) \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \theta e^{-i\phi} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{73}$$

Hence

$$\begin{aligned} B(\theta, \phi) &= exp\left(\theta \left(e^{i\phi} \hat{a} \hat{b}^\dagger + e^{-i\phi} \hat{a}^\dagger \hat{b} \right) \right) \\ &= exp\left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \theta e^{i\phi} & 0 \\ 0 & \theta e^{-i\phi} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cosh\theta & e^{i\phi}\sinh\theta & 0 \\ 0 & e^{-i\phi}\sinh\theta & \cosh\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{74}$$

If we apply two rotation gates on a two-qumode system, parameterized by ϕ_0 and ϕ_1, we get the tensor product of the two rotation gates:

$$
\begin{aligned}
R(\phi_0) \otimes R(\phi_1) &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi_0} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi_1} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 & 0 \\ 0 & 0 & e^{i\phi_0} & 0 \\ 0 & 0 & 0 & e^{i(\phi_0+\phi_1)} \end{bmatrix}
\end{aligned} \tag{75}
$$

According to [120], the operation of beamsplitters followed by rotation gates constitutes an interferometer. The matrix representation of an interferometer on a two-qumode system is then

$$
\begin{aligned}
B(\theta,\phi) \circ (R(\phi_0) \otimes R(\phi_1)) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cosh\theta & e^{i\phi}\sinh\theta & 0 \\ 0 & e^{-i\phi}\sinh\theta & \cosh\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 & 0 \\ 0 & 0 & e^{i\phi_0} & 0 \\ 0 & 0 & 0 & e^{i(\phi_0+\phi_1)} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi_1}\cosh\theta & e^{i(\phi_0+\phi)}\sinh\theta & 0 \\ 0 & e^{i(\phi_1-\phi)}\sinh\theta & e^{i\phi_0}\cosh\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{76}
$$

If we were to apply rotation gates first and then a beamsplitter, we get

$$
(R(\phi_0) \otimes R(\phi_1)) \circ B(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\phi_1}\cosh\theta & e^{i(\phi_1+\phi)}\sinh\theta & 0 \\ 0 & e^{i(\phi_0-\phi)}\sinh\theta & e^{i\phi_0}\cosh\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{77}
$$

5.2 Ternary Gates

By setting the cutoff dimension, we are able to implement multi-valued logic. To realize ternary logic computations in optical quantum computing, we set the cutoff dimension at three.

In implementing the ternary logic of base three, the quantum state of the system is expressed as a complex projective linear combination of three basis states as:

$$|\Psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \alpha_2 |2\rangle , \tag{78}$$

where $\sum_{k=0}^{2} \|\alpha_k\|^2 = 1$. In a two-qumode system, the quantum state of the entire system is

$$\begin{aligned} |\Psi_0\rangle \otimes |\Psi_1\rangle &= (\alpha_{00} |0\rangle + \alpha_{01} |1\rangle + \alpha_{02} |2\rangle) \otimes (\alpha_{10} |0\rangle + \alpha_{11} |1\rangle + \alpha_{12} |2\rangle) \\ &= \beta_0 |00\rangle + \beta_1 |01\rangle + \beta_2 |02\rangle + \cdots + \beta_8 |22\rangle \end{aligned} \tag{79}$$

where $\beta_k = \alpha_{0k_0}\alpha_{1k_1}$ and $|k_0k_1\rangle$ is the state with the first qumode being in the $|k_0\rangle$ state and the second qumode in $|k_1\rangle$.

Ternary logic displacement gate:

$$\begin{aligned} D(\alpha) &= exp(\alpha\hat{a}^\dagger - \alpha^*\hat{a}) \\ &= exp\left(\alpha \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{bmatrix} - \alpha^* \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{bmatrix} \right) \\ &= exp\left(\begin{bmatrix} 0 & -\alpha^* & 0 \\ \alpha & 0 & -\sqrt{2}\alpha^* \\ 0 & \sqrt{2}\alpha & 0 \end{bmatrix} \right) \end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\alpha^* & 0 \\ \alpha & 0 & -\sqrt{2}\alpha^* \\ 0 & \sqrt{2}\alpha & 0 \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} 0 & -\alpha^* & 0 \\ \alpha & 0 & -\sqrt{2}\alpha^* \\ 0 & \sqrt{2}\alpha & 0 \end{bmatrix}^2 + \cdots \\
&= \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}
\end{aligned} \tag{80}
$$

where

$$a = 1 + \sum_{n=1}^{\infty} \frac{3^{n-1}(-\alpha\alpha^*)^n}{(2n)!} \tag{81}$$

$$b = \sum_{n=0}^{\infty} \alpha^* \frac{(-1)^{n+1}(3\alpha\alpha^*)^n}{(2n+1)!} \tag{82}$$

$$c = \sum_{n=1}^{\infty} \sqrt{2}\,(\alpha^*)^2\, \frac{(-3\alpha\alpha^*)^{n-1}}{(2n)!} \tag{83}$$

$$d = \sum_{n=0}^{\infty} \alpha \frac{(-3\alpha\alpha^*)^n}{(2n+1)!} \tag{84}$$

$$e = \sum_{n=0}^{\infty} \frac{(-3\alpha\alpha^*)^n}{(2n)!} \tag{85}$$

$$f = \sum_{n=0}^{\infty} \sqrt{2}\alpha^* \frac{(-1)^{n+1}(\alpha\alpha^*)^n}{(2n+1)!} \tag{86}$$

$$g = \sum_{n=1}^{\infty} \sqrt{2}\alpha^2 \frac{(-1)^n(\alpha\alpha^*)^{n-1}}{(2n)!} \tag{87}$$

$$h = \sum_{n=0}^{\infty} \sqrt{2}\alpha \frac{(-\alpha\alpha^*)^n}{(2n+1)!} \tag{88}$$

$$i = \sum_{n=0}^{\infty} 2\frac{(-\alpha\alpha^*)^n}{(2n)!} - 1 \tag{89}$$

The ternary logic rotation gate is:

$$
\begin{aligned}
R(\phi) &= exp\left(i\phi\hat{a}^{\dagger}\hat{a}\right) \\
&= exp\left(i\phi\begin{bmatrix}0&0&0\\0&1&0\\0&0&2\end{bmatrix}\right) \\
&= \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} + \begin{bmatrix}0&0&0\\0&i\phi&0\\0&0&2i\phi\end{bmatrix} + \frac{1}{2}\begin{bmatrix}0&0&0\\0&i\phi&0\\0&0&2i\phi\end{bmatrix}^2 + \cdots + \frac{1}{n!}\begin{bmatrix}0&0&0\\0&i\phi&0\\0&0&2i\phi\end{bmatrix}^n + \cdots \\
&= \begin{bmatrix}1 & 0 & 0\\ 0 & 1+i\phi+\frac{(i\phi)^2}{2}+\cdots\frac{(i\phi)^n}{n!}+\cdots & 0\\ 0 & 0 & 1+2i\phi+\frac{(2i\phi)^2}{2}+\cdots\frac{(2i\phi)^n}{n!}+\cdots\end{bmatrix} \\
&= \begin{bmatrix}1&0&0\\0&e^{i\phi}&0\\0&0&e^{2i\phi}\end{bmatrix}
\end{aligned}
\tag{90}
$$

We can generalize that the rotation gate R_n with parameter ϕ on multi-valued logic of radix-n would yield an $n \times n$ matrix

$$
R_n(\phi) = \begin{bmatrix}1&0&\dots&0\\0&e^{i\phi}&\dots&0\\&&\dots&\\0&0&\dots&e^{(n-1)i\phi}\end{bmatrix} \tag{91}
$$

Ternary squeezer when the parameter r is a real number:

$$
\begin{aligned}
S(r) &= exp\left(r\left(\hat{a}^2 - \hat{a}^{\dagger 2}\right)/2\right) \\
&= exp\left(r\left(\begin{bmatrix} 0 & \sqrt{1} & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{bmatrix}^2 - \begin{bmatrix} 0 & 0 & 0 \\ \sqrt{1} & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{bmatrix}^2\right)/2\right) \\
&= exp\left(r\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 2 \\ 0 & -2 & 0 \end{bmatrix}/2\right) \\
&= exp\left(\begin{bmatrix} 0 & \frac{r}{2} & 0 \\ -\frac{r}{2} & 0 & r \\ 0 & -r & 0 \end{bmatrix}\right) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \frac{r}{2} & 0 \\ -\frac{r}{2} & 0 & r \\ 0 & -r & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & \frac{r}{2} & 0 \\ -\frac{r}{2} & 0 & r \\ 0 & -r & 0 \end{bmatrix}^2 + \\
&\cdots + \frac{1}{n!}\begin{bmatrix} 0 & \frac{r}{2} & 0 \\ -\frac{r}{2} & 0 & r \\ 0 & -r & 0 \end{bmatrix}^n + \cdots
\end{aligned}
\tag{92}
$$

The matrices representing rotation gates will be of size 3×3 for each qumode. The tensor product of the rotation gates is of size 9×9.

$$
\begin{aligned}
R(\phi_0) \otimes R(\phi_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_0} & 0 \\ 0 & 0 & e^{2i\phi_0} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} \\
&= \begin{bmatrix}
1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & 0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & 0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} \\
0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & \alpha_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & 0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} \\
0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & 0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix} & \alpha_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 \\ 0 & 0 & e^{2i\phi_1} \end{bmatrix}
\end{bmatrix} \\
&= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & e^{i\phi_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & e^{2i\phi_1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & e^{i\phi_0} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & e^{i\phi_0}e^{i\phi_1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & e^{i\phi_0}e^{2i\phi_1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & e^{2i\phi_0} & 0 & 0 \\
0 & 0_0 & 0 & 0 & 0 & 0 & 0 & e^{2i\phi_0}e^{i\phi_1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{2i\phi_0}e^{2i\phi_1}
\end{bmatrix}
\end{aligned}
\tag{93}
$$

6 QUANTUM MULTI-CLASS DATA CLASSIFIER

Machine learning is a method of extracting hidden patterns from data without explicitly programming. A machine learning algorithm "learns" the hidden patterns in data through the process called "training". The "learned" model can be applied to new data samples for predictions. A parameter-based machine learning algorithm is a mathematical expression, whose parameters can be trained to best describe the hidden patterns of data. An algorithm with a set of optimized parameters through training that minimizes the given cost function is called a machine learning model.

Classifiers are supervised machine learning algorithms in which the class label of a new data sample is predicted based on the learned parameters of the model. The quantum classifiers in the literature are mainly binary classifiers. I created a quantum multi-class data classifier capable of classifying up to 11 billion classes in Xanadu's X8 technology.

6.1 Quantum Machine Learning

In supervised machine learning, data samples defined by a finite number of features are mapped to labels:

$$f : \mathbb{R}^n \to \mathbb{R}^m : x \mapsto y \tag{94}$$

where x is a feature vector of length n and y is either a single-valued label or a one-hot encoded vector of length m, where m is the number of classes. The goal is to extract an optimal set of parameters defining the map f from data to labels, minimizing the given objective function.

Machine learning algorithms such as support vector machine, naive Bayes, linear regression, k-nearest neighbors, and stochastic gradient descent have been successfully used for pattern extraction from data. In employing these algorithms in

quantum computing, the inherent higher-dimensionality of the computational space of quantum computing offers the possibility of naturally processing higher-order features of data. Additionally, there are properties of quantum mechanical systems not present in classical systems, which are used as components of quantum computing: superposition, entanglement, and interference. These unique properties of quantum computing may further aid in finding additional features of data that may not be readily available in classical computing.

Quantum machine learning (QML) can be viewed as a way of mapping data into a higher-dimensional Hilbert space for pattern extraction [201]. In classical support vector machines, the kernel method is used to find a separating hyperplane in a higher-dimensional feature space as seen in Figure 27 [127].

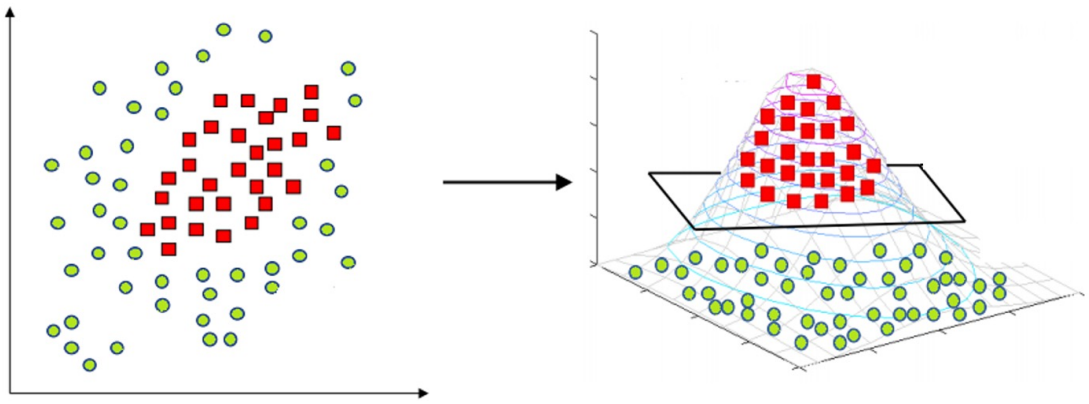

Figure 27: Separating hyperplane in a higher dimension [image source]

In QML, the mapping of data samples is done in a higher-dimensional Hilbert space and then the computational results are brought back down to classical via a measurement operator [201]. Then the classical map f can be viewed as a composition map of data encoding (S), quantum computation (U), and measurement (M).

$$
\begin{aligned}
&S : \mathbb{R}^n \to \mathbb{C}^m : x \mapsto |\phi(x)\rangle \\
&U : \mathbb{C}^m \to \mathbb{C}^m : |\phi(x)\rangle \mapsto |U\phi(x)\rangle \\
&M : \mathbb{C}^m \to \mathbb{R} : |U\phi(x)\rangle \mapsto \langle\phi(x)| U^\dagger M U |\phi(x)\rangle = \langle M\rangle = y \\
&M \circ U \circ S : \mathbb{R}^n \to \mathbb{R} : x \mapsto y
\end{aligned}
\tag{95}
$$

Notice that although the composition map takes $x \in \mathbb{R}^n$ to $y \in \mathbb{R}$, the intermediary computation is happening in $\mathbb{C}^m \cong (\mathbb{R}^2)^m$. In the qubit model, the dimension m of the computational space is defined by $m = 2^q$ where q represents the number of qubits. In optical quantum computing, $m = n^q$ where n represents the cutoff dimension and q is the number of qumodes. Since n can be any integer, the computational space in optical quantum computing can be much larger than that of the qubit model. The enlarged computational space, used as a feature space, offers a potential for finding hidden patterns not readily available in lower-dimensional computational spaces.

In implementing quantum machine learning algorithms, there are three main components: data encoding, quantum algorithm circuit, and measurement as seen in the composition map [149]. The dataflow in executing an algorithm is

- Data encoding: Classical data samples need to be converted into quantum states to be processed by a quantum machine learning circuit. This is usually done by using sample entries as parameters of parameterized quantum gates.

- Quantum circuit: We want the circuit to "learn" the parameters that best fit the hidden features of the data. Hence we use parameterized (variational) gates whose actions are defined by the parameters of the gates. A quantum circuit composed of parameterized (variational) gates is called a parameterized (variational) circuit.

- Measurement: The results from measurement are interpreted as labels.

The schematic of the quantum machine learning circuit, composed of the three distinct components is shown in Figure 28 (The red blocks represent the measurement component).

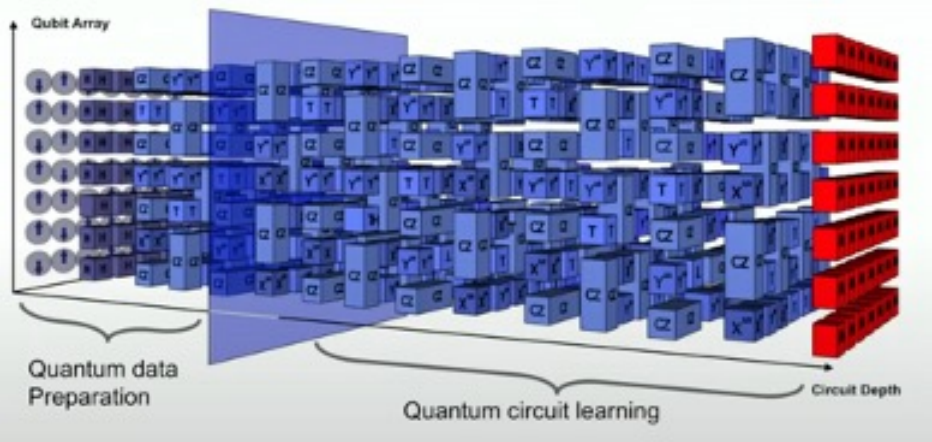

Figure 28: Quantum machine learning circuit (Google Quantum Talk)

The output from the quantum circuit is sent to a CPU wherein all the optimization calculation is done. The updated parameters are sent to the QPU and the quantum gates are recalibrated. Since the entire circuit is composed of classical and quantum, the input data x can be pre-processed using classical machine learning algorithms. By the same token, the output from the quantum circuit can be post-processed classically. In this way, the quantum circuit can be viewed as a sub-circuit inside an overall classical machine learning algorithm, whose component is farmed out to a QPU.

In implementing QML algorithms on near-term devices, there is a limitation of computing power for data encoding of datasets with a large number of features. The current QPUs or simulators are not capable of providing enough qubits or qumodes for accommodating data encoding of all the feature variables. Hence, the QML algorithms in the literature adopt classical dimensionality reduction methods such as

Principal Component Analysis, convolutional neural networks (CNN), or feedforward neural networks to reduce the number of features to be converted to quantum states by a QML circuit.

6.1.1 Data Encoding

The first step in building a QML model is data encoding: converting classical data into quantum states. They are used as input to a quantum circuit, hence the process is also called quantum state preparation. The prepared states are processed through a quantum circuit, which is realized by multiple layers of quantum logic gates.

6.1.2 Variational Quantum Circuit

In QML, training is the process of "learning" optimal parameters of the gates that would produce the classification of new data samples as accurately as possible. In supervised QML, the circuit produces "prediction", which is a close approximation of the true label of patterns, from the measurement operator. The prediction result from the quantum circuit is then processed on a classical circuit for the computation of the objective function, gradients, and new parameters. The updated parameters from the classical circuit are then fed back to the quantum circuit for subsequent iterations.

For a machine learning algorithm to "learn" optimal parameters, its corresponding circuit needs to have parameterized components whose parameters are tunable for optimal results. Hence the quantum circuit for learning is composed of parameterized quantum gates, also called variational quantum gates.

In executing a QML on a QPU, each qubit or qumode is initialized to the $|0\rangle$ state. Then using "state preparation" gates, classical data input x is converted to a quantum state $|\psi\rangle$. State preparation or data encoding is the process of embedding a

classical input $x \in \mathbb{R}^n$ as a quantum state $|\psi\rangle \in \mathbb{C}P^{n-1}$, which is the $n-$dimensional surface space of a higher dimensional sphere S^{2n}. Then a variational circuit, i.e., parameterized circuit, composed of parameterized quantum gates implements a QML algorithm.

6.1.3 Measurement

The measurement operator functions as a projection map from a higher-dimensional Hilbert space to a lower-dimensional classical space $\mathbb{R}^t$. There are several ways of controlling the number t depending on the objective function we want to use for a QML algorithm.

Both in the binary logic and the multi-valued logic model of quantum computing, the expectation value measurement method returns output vectors of size m where m is equal to the number of qubits or qumodes.

Employing the probability measurement method yields the output vectors of length n^m, where n is the cutoff dimension and m is the number of qumodes used. The output vectors of various sizes based on the number of qumodes with cutoff dimensions two, three, and six are shown in Table 1. Notice that with the same number of qumodes, the higher-radix results in the output vectors of greater length. This is a demonstration of the expressive power of the cutoff dimension as the dimension of the computational space.

Number of Qumodes Used	Qubit-based	Ternary	Radix-6
1	2	3	6
2	$2^2 = 4$	$3^2 = 9$	$6^2 = 36$
3	$2^3 = 8$	$3^3 = 27$	$6^3 = 216$
4	$2^4 = 16$	$3^4 = 81$	$6^4 = 1,296$
5	$2^5 = 32$	$3^5 = 243$	$6^5 = 7,776$
6	$2^6 = 64$	$3^6 = 729$	$6^6 = 46,656$
7	$2^7 = 128$	$3^7 = 2,187$	$6^7 = 279,936$
8	$2^8 = 256$	$3^8 = 6,561$	$6^8 = 1,679,616$

Table 1: Varying output vector sizes based on the number of qumodes with cutoff dimensions two, three, and six.

The output vectors of various sizes based on the cutoff dimension in an eight-qumode circuit are shown in Table 2.

Cutoff Dimension	Output vector size
5	$5^8 = 390,625$
6	$6^8 = 1,679,616$
7	$7^8 = 5,764,801$
8	$8^8 = 16,777,216$
9	$9^8 = 43,046,721$
10	$10^8 = 100,000,000$
11	$11^8 = 214,358,881$
12	$12^8 = 429,981,696$

Table 2: Varying output vector sizes in an $8-$qumode circuit based on the cutoff dimensions ranging from five to twelve.

In Xanadu's X8, up to 18 photons are observed, realizing radix-19 multi-valued logic. Then with cutoff dimension 20 [46], classification of data with up to $20^8 = 25,600,000,000$ classes is feasible in theory.

Using a higher radix logic greatly reduces computational resources. For example, to achieve the same computing power as using radix-20 on an eight qumode system, we would need $log_2 25,600,000,000 \sim 35$ qubits.

[46] including the ground state $|0\rangle$.

6.1.4 Optimization

To get an optimal set of parameters for the quantum gates in the circuit, the following calculations are performed on a CPU:

- loss as per the given loss function
- gradients
- optimization
- parameter update

Then the updated parameters are used as new parameters for the variational quantum circuit in the subsequent iteration until an optimal set of parameters is learned. The QPU and CPU interaction with the training process on a QML circuit and the optimization process on a CPU is depicted in Figure 29.

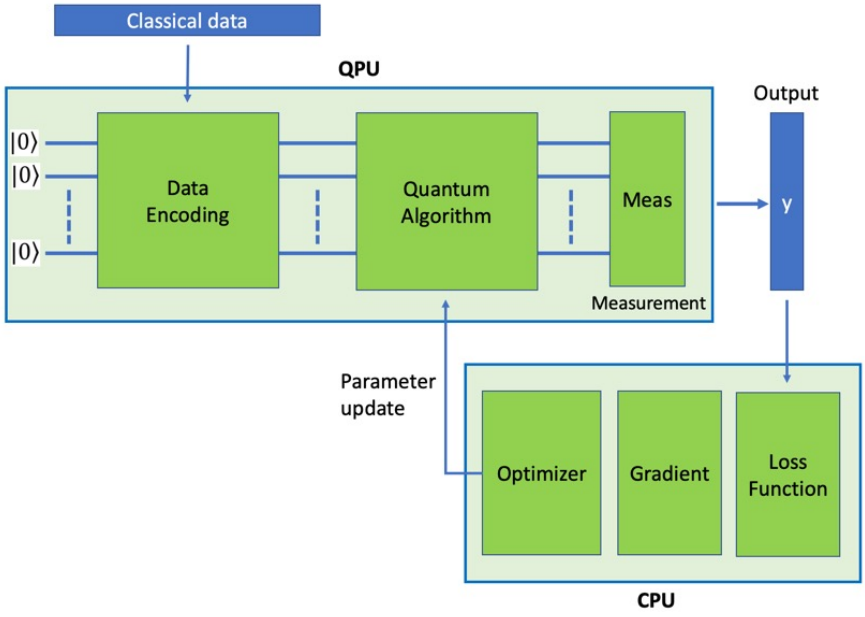

Figure 29: QPU and CPU interaction in variational quantum circuits

6.2 Quantum Deep Learning

Deep learning is one of the machine learning algorithms to extract higher-order hidden features from data, inspired by the workings of the brain [142]. The multi-layered cerebral information processing structure of the neocortex inspired neural networks, modeling the information processing flow of the brain. As in sensory cortices in the brain, deep learning networks use hidden layers to progressively extract higher-order features from data. Deep learning algorithms such as feedforward neural networks, CNN, recurrent neural networks, and deep belief networks are examples that have been implemented in quantum circuits.

To capture the multi-layer structure of the biological neocortex, the deep learning algorithms stack layers of neurons between input data and output of the network. These added layers are called hidden layers and they help the network to progressively process higher-order features. The resulting network is called Artificial Neural Network, Multi-layer Perceptron, or Feedforward Neural Network.

The transition from classical computing to quantum computing is a natural extension of computational spaces from two points (i.e., 0 and 1) to the surface of the Bloch Sphere (qubit-based model) or higher-dimensional spheres (multi-valued model). Hence quantum computing offers its inherent high-dimensional space in which hidden features of data can be explored and extracted [201]. Since the goal of deep learning is to extract hidden features of data from a higher-dimensional feature space, it would be natural to explore quantum computing models which offer higher-dimensional computational spaces. This is a strong argument for multi-valued logic with higher radices.

6.2.1 Quantum Neural Networks

Among deep learning algorithms are neural networks, whose architecture is inspired by the workings of the brain. It mimics the neuronal communication networks, composed of multiple layers, wherein the communication between two layers of neurons is activated by the action potential.

A physical neuron gets input signals from neighboring neurons through many branches called dendrites. The signals are collected into the nucleus of the cell body and then the sum of the signal impulses is propagated to the axon. The electrical pulse coming from the cell body is cascaded down to axon terminals to be passed down to the next cells based on the strength of the pulse. The mechanism in which the pulse reaches the action terminal is nonlinear and is called action potential [100]. This nonlinearity of action potential plays an important role in artificial neural networks. When the signal reaches the threshold of the action potential, it travels through the axon and is cascaded through axon terminals to be passed to other neurons. The whole process is summarized in the following steps:

- summation of input signals
- bias addition to the sum
- nonlinear action potential (activation function).

The synaptic weights w_{ki} represent the strength of the connection between two neurons for passing impulses [197]. The collection of synaptic weights in the k^{th} layer can be expressed as a row vector $w_k = [w_{k1}, w_{k2}, \ldots, w_{km}]$ and the sample inputs as a column vector x_k. Then the dot product of the two vectors is gathered at the summing junction to which a bias value is added. Thereafter a nonlinear activation function is applied, modeled after the nonlinear action potential in biological neurons.

The mathematical expression of the nonlinear function on the affine transformation of original data in each layer is captured in $L(x) = \phi(Wx+b)$ where W represents the matrix of synaptic weights, b bias addition, and $\phi(\cdot)$ a nonlinear activation function. Each k^{th} row of the matrix contains the synaptic weights applied to the input being summed up to the k^{th} neuron of the subsequent layer. The result of the application of the weight matrix is a linear transformation on the input vector. Together with the bias addition component, $Wx + b$ results in the affine transformation of the input x.

- linear transformation: W changes the input state linearly.
- affine transformation: translation by $+b$ on Wx.
- nonlinear activation function: transforms $Wx + b$ nonlinearly.

Different nonlinear functions are used on the affine transformation of input vectors. For the output vectors, it is customary to use the softmax function whose range is $[0, 1]$ so that the results can be interpreted as the probability of getting each entry of the output vector.

In the artificial neural network architecture, there are multiple layers of feature maps between the input layer and output layer. These additional layers can be viewed as extended computational feature space. Instead of mapping input data vectors into output vectors $f : \mathbb{R}^d \to \mathbb{R}^n : x \mapsto y$, where x is an $d-$dimensional feature vector and y, its corresponding $n-$dimensional label, in employing hidden layers, we are mapping the input vectors to intermediary feature spaces.

$$f : \mathbb{R}^d \to \mathbb{R}^{h_1} \to \cdots \to \mathbb{R}^n : x \mapsto x_{h_1} \mapsto \cdots \mapsto y, \tag{96}$$

where h_k is the dimension of the k^{th} hidden layer.

Two main elements of deep learning, which make deep learning unique, are mapping data features into higher-dimensional space and nonlinear activation functions between two layers [7]. The higher-dimensional space to which original features are mapped via feature maps allows for operations that are not readily feasible in the original feature space [201]. Nonlinear activation functions process features beyond linear transformations and provide solutions that may not be linear [253].

In implementing a quantum version of artificial neural networks, the goal is to convert each classical layer $L(x) = \phi(Wx + b)$ to a quantum state $L(|x\rangle) = |\phi(Wx + b)\rangle$ using available gates in the model. A quantum circuit is composed of a data encoding circuit, a quantum neural network circuit, and the measurement operation.

Qubit-based Model

In the binary logic (qubit-based) model of quantum computing, these extended hidden feature spaces can be expressed as

$$\begin{aligned} f :&\mathbb{R}^d \to \mathbb{C}^{2^s} \to \mathbb{C}^{2^{h_1}} \to \cdots \to \mathbb{R}^n : \\ &x \mapsto |x_s\rangle \mapsto |x_{h_1}\rangle \mapsto \cdots \mapsto y, \end{aligned} \tag{97}$$

where s is the number of qubits used for data encoding (state preparation) and h_1 [47], for the hidden layers [48]. In this model, all the gates are unitary, representing rotations on the Bloch Sphere as linear transformations. Each k^{th} layer can be represented as $L_k(|x\rangle) = |U_k(x)\rangle$ where U_k is a unitary linear transformation, missing bias addition and nonlinear activation function. The composition of multiple unitary matrices

[47] In most of the circuits in the literature, $s = h_1 = \cdots = h_n$.

[48] We are not taking the projective nature of the computational space into account for the purpose of examining the degree of freedom in the feature space

is a single unitary matrix. Therefore, employing multiple unitary gates to represent multiple hidden layers of the classical neural networks results in a single layer of linear transformation. Hence, it is not possible to faithfully implement the notion of "deep learning" in the qubit-based model because of the inherent computing framework. Various models of quantum deep learning have been proposed based on the qubit-based model of quantum computing [3, 22, 57, 74, 98, 255].

The quantum neural networks proposed in the literature are mainly built on the binary model. Since all the gates in the qubit-based model are unitary, the nonlinear activation function component of classical neural networks can be implemented only through the measurement operation. However, measurement terminates one cycle of the quantum circuit by projecting the quantum computational result into classical basis states. Hence the notion of "deep learning" cannot be implemented in the discrete variable model. Additionally, there is no direct way to perform affine transformations representing biases.

The architecture of a qubit-based quantum neural network proposed by Farhi et al. is depicted in Figure 30 [74]. Most of the networks proposed in the literature follow the same pattern of multiple layers of parameterized linear transformations and accessing the results using a single readout qubit for binary classification [1, 11, 22, 98].

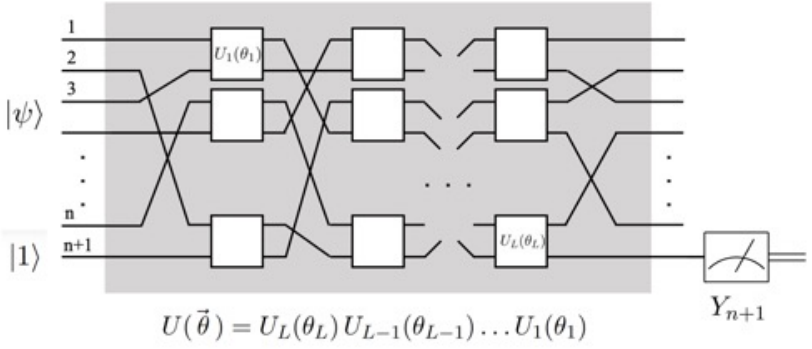

Figure 30: Quantum neural network architecture in the qubit-based model [74]

In this circuit, there are multiple layers of parameterized unitary gates applied, denoted by $U_k(\theta_k)$ for the k^{th} layer operation. However, the composition of the matrices each representing one layer is still a unitary matrix, which is a linear transformation. Let $U_k(\theta_k)$ be the k^{th} layer of the network parameterized by the set of parameters θ_k. The unitary gate $U_k(\theta_k)$ is an implementation of some linear transformation W. The composition $U(\theta) = U_k(\theta_k) \circ U_{k-1}(\theta_{k-1}) \circ \ldots \circ U_1(\theta_1)$ is still a unitary matrix representing a single layer of a neural network. The result of the computation is transferred to a single ancillary qubit for readout.

In this framework, if one were to modify this network to classify datasets of multi-classes, one will need to add more ancillary qubits for readout. For example, to classify datasets of 10 classes like MNIST and Cifar10, the ceiling of $log_2(10) = 4$ qubits is needed. In the optical quantum computing network architecture, the readout is done on each computing qumode, hence the need for extra qumodes is inherently absent in Xanadu's optical quantum computers.

Optical Quantum Neural Networks

The use of nonlinear gates in optical computing was first introduced by Shen et al. [212]. The authors implemented classical neural networks in optical hardware, realizing the nonlinear activation component with nonlinear optical operations. Killoran et al. expanded on Shen's architecture to include bias addition using the displacement gate in a quantum environment [120]. In optical quantum computing, because of the availability of the displacement gate and nonlinear gates, it is possible to build quantum layers that faithfully implement a quantum version of nonlinear activation on the affine transformation of quantum data.

In quantum optics are observed linear optical phenomena as well as nonlinear. When both of these properties are utilized as elements of information processing, there

is a potential for processing information nonlinearly without the need of approximating it with linear expressions. Xanadu has incorporated the linear and nonlinear effects of quantum optics in their QPU, X8 [11, 120]. The modes here can be interpreted as separate electromagnetic fields, whose change in states is used for processing quantum information [125]. In Xanadu's X8, the readout is done on each computing qumode, hence the need for ancillary qumodes is inherently absent.

In building quantum deep learning circuits, implementing the true notion of "deep learning" is feasible in optical quantum computing because of the nonlinear gates available. Shen et al. proposed an architecture for faithfully implementing the neural network layer $L(x) = \phi(Wx + b)$ as a quantum state $L(|x\rangle) = |\phi(Wx + b)\rangle$ on photonic quantum computers [212]. Killoran et al. implemented the idea into practical examples: curve-fitting, classical and quantum hybrid auto-encoder, and hybrid quantum neural network for binary classification on the fraudulent credit card transaction dataset [120]. The author proposed the first multi-class quantum classifier by expanding on Killoran's idea for binary classification [49]. The flexibility of defining the dimension of the computational space (feature map space) allows for the flexibility of determining the size of the output vectors used for inference. Hence, optical quantum computing allows for the classification of n^m classes of labels where n is the cutoff dimension and m, the number of qumodes used.

In optical quantum computing, there is a direct translation of the notion of affine transformations and non-activation functions. The linear transformation W portion of an affine transformation $Wx + b$ is achieved with the composition of interferometers and squeezing gates. The bias addition $+b$ portion is achieved with a set of displacement gates. The non-activation function component is achieved with a set of Kerr gates. A quantum circuit can perform multiple layers of a quantum neural network before producing measurement results.

Architecture

Naturally embedded in optical quantum computing are quantum gates to directly implement the expression $L(|x\rangle) = |\phi(Wx + b)\rangle$. Let U_k denote the k^{th} interferometer, S a set of m squeezers, D a set of m displacement gates, and ϕ a set of m Kerr gates, where m is the number of qumodes.

On an $m-$qumode system, $m - 1$ beamsplitters on pairs of adjacent qumodes and m rotation matrices form an interferometer as seen in the Figure 31.

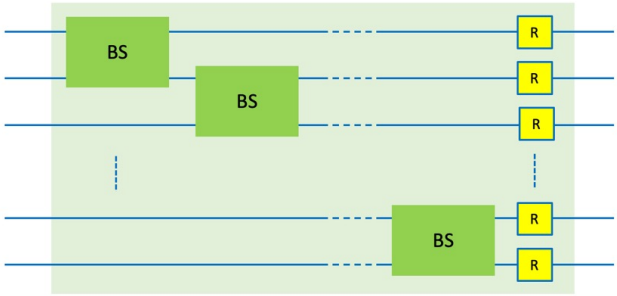

Figure 31: Make-up of an interferometer

The affine transformation $Wx+b$ is implemented by the composition $D \circ U_2 \circ S \circ U_1$ and the nonlinear activation function ϕ by the Kerr gates. The composition $\phi \circ D \circ U_2 \circ S \circ U_1$ acting on a quantum state $|x\rangle$ gives us the desired state $L(|x\rangle) = |\phi(Wx + b)\rangle$. The schematic of the circuit is shown in Figure 32.

The composition of the gates $U_2 \circ S \circ U_1$ implements a quantum version of the linear transformation matrix W. Any matrix M can be factorized using singular value decomposition (SVD) as $M = U\Sigma V^*$, where U and V are orthogonal and Σ is diagonal [257]. The action of a phaseless interferometer U on the quantum state $|x\rangle = \otimes_{k=1}^{m} |x_k\rangle$ has an effect of an orthogonal matrix acting on $|x\rangle$.

The parameterized squeezer $S(r_k)$ acts on the quantum state $|x_k\rangle$ of each k^{th} qumode as $S(r_k)\,|x_k\rangle = \sqrt{e^{-r_k}}\,|e^{-r_k} x_k\rangle$. Collectively they have an effect of a diagonal

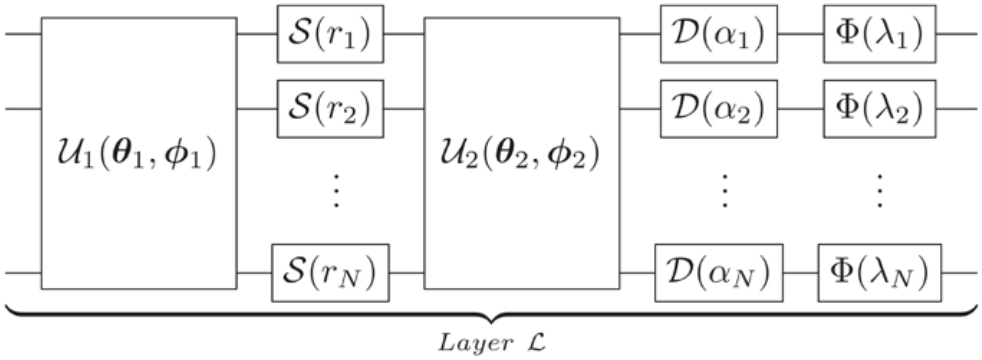

Figure 32: Optical quantum neural network architecture [120]

matrix $S = S(r_1) \otimes S(r_1) \otimes \ldots \otimes S(r_m)$ acting on $|x\rangle = \otimes_{i=1}^{m} |x_k\rangle$.

Orthogonal matrices are just unitary matrices with real entries, inducing length-preserving rotations. Then the transpose of an orthogonal matrix represents the reverse rotation of the original matrix, thus orthogonal. Then the composition $U_2 \circ S \circ U_1$ can be considered as the composition $O_2 \circ \Sigma \circ \left(O_1^T\right)^T$, where O_2 and O_1^T are orthogonal and Σ diagonal. Together, the interferometers and squeezers implement an SVD form of some linear transformation W [120].

The bias addition is realized with displacement gates D. The displacement gate has an effect

$$D(\alpha_k) |\psi_k\rangle = |\psi_k + \sqrt{2}\alpha_k\rangle \tag{98}$$

for each k^{th} qumode. Then $D(\alpha) |\psi\rangle = |\psi + \sqrt{2}\alpha\rangle$ collectively for $\alpha^T = [\alpha_1, \alpha_2, \ldots, \alpha_m]$. For some desired bias b, let $\alpha = \frac{b}{\sqrt{2}}$, then the collection of displacement gates implements the bias addition. The composition $D \circ U_2 \circ S \circ U_1$ acting on the quantum state $|x\rangle$ gives us the affine transformation

$$D \circ U_2 \circ S \circ U_1 |x\rangle = |O_2 \Sigma O_1 x + b\rangle = |Wx + b\rangle . \tag{99}$$

The nonlinear activation function $\phi(\cdot)$ is realized with Kerr gates. The Kerr gate,

parameterized by the parameter κ, is a nonlinear transformation gate. Let n be the cutoff dimension and m the number of qumodes. For the quantum state $|\psi\rangle$ of one qumode, which is a superposition of n Fock basis states, the Kerr gate with parameter κ has an effect

$$\begin{aligned} K(\kappa)\,|\psi\rangle &= \begin{bmatrix} e^{i\kappa 0^2} & 0 & \dots & 0 \\ 0 & e^{i\kappa 1^2} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & e^{i\kappa(n-1)^2} \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} \psi_0 \\ e^{i\kappa 1^2}\psi_1 \\ \vdots \\ e^{i\kappa(n-1)^2}\psi_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} \psi_0 \\ (\cos\kappa + i\sin\kappa)\psi_1 \\ \vdots \\ (\cos\kappa(n-1)^2 + i\sin\kappa(n-1)^2)\psi_{n-1} \end{bmatrix}, \end{aligned} \tag{100}$$

which is nonlinear.

Together, the circuit $L = \Phi \circ D \circ U_2 \circ S \circ U_1$ gives us a quantum version $L\,(|x\rangle) = |\Phi(Wx + b)\rangle$ of a classical neural network $L(x) = \Phi(Wx + b)$.

The quantum version of a neural network is composed of encoding given feature vector x into a quantum state and performing a series of quantum operations equivalent to the classical transformation $L(x)$. The objective is to implement the classical network $L(x) = \phi(Wx + b)$ as a quantum state $L(|x\rangle) = |\phi(Wx + b)\rangle$ using available

quantum gates. In converting classical neural networks into quantum circuits, the key components are the affine transformation $Wx + b$ and the nonlinear activation function $\phi(\cdot)$.

The nonlinear activation function $\phi(\cdot)$ is realized with nonlinear Kerr gates, parameterized by the parameter κ_i.

Together, the circuit $L = \Phi \circ D \circ U_2 \circ S \circ U_1$ gives us a quantum version $L(|x\rangle) = |\Phi(Wx + b)\rangle$ of a classical neural network layer $L(x) = \Phi(Wx+b)$ [120]. Since all these operations are performed at a gate level without being interrupted by measurement operation, it is possible to stack the interferometer - squeezers - interferometer - displacement gates - Kerr gates sequence in multiple layers. Hence the notion of quantum "deep learning" is directly implemented in optical quantum computing.

The complex parameters of the squeezer and the displacement gate, when viewed as $\alpha = ae^{i\theta} = a(\cos(\theta) + i\sin(\theta))$, where a is the magnitude and θ the phase, can be treated as two-parameter gates.

Deep learning algorithms have been successful in extracting higher-order features from data through multiple layers of feature maps [49]. One of the deep learning algorithms, artificial neural networks are inspired by the neocortex, in which the input signals are processed through six layers of neurons progressively processing signal into higher-order features [172].

In optical quantum computing, the expression of extended hidden feature spaces is given by

$$\begin{aligned} f :&\mathbb{R}^d \to \mathbb{C}^{m^s} \to \mathbb{C}^{m^{h_1}} \to \cdots \to \mathbb{R}^n : \\ &x \mapsto |x_s\rangle \mapsto |x_{h_1}\rangle \mapsto \cdots \mapsto y, \end{aligned} \tag{101}$$

[49] Deep learning algorithms include artificial neural networks, CNN, recurrent networks, and spike networks. In this paper, we limit our discussion to artificial neural networks.

where m is the cutoff dimension. The cutoff dimension m can be set to any value higher than 2 that technology allows, hence optical quantum computing avails a much higher-dimensional feature space for processing higher-order features from data.

The availability of nonlinear gates along with the displacement gate in optical quantum computing allows for a direct translation of a layer of classical neural networks into a quantum state $L(|x\rangle) = |\phi(Wx + b)\rangle$. This is achieved with a sequence of interferometers, squeezers, interferometers, displacement gates, and Kerr gates [120]. Stacking the sequence multiple times achieves the notion of multiple hidden layers.

In applying the notion of "deep learning" in the quantum computing paradigm, optical quantum computing avails multiple layers of the quantum neural network circuit without the measurement operation [120]. In the qubit-based model, multiple layers of the quantum neural network reduce to a single layer mathematically.

Most of the data on which machine learning is applied for pattern extraction contain features that are continuous in nature. Then optical quantum computing will be a better tool to use than the discrete variable model in extracting patterns utilizing the properties of quantum mechanics. To implement the notion of quantum deep learning for extracting higher-order patterns from data, I built upon the quantum neural network architecture proposed in the paper from 2019 [120].

The advantage of using optical quantum computing for deep learning is manifold:

- User can define the dimension of the computational space.
- Quantum gates not available in the discrete variable model can be used for computation: displacement gate and nonlinear gates.
- User can define the dimension of the practical computational space.
- No ancillary qumodes are needed for readout.

6.2.2 Quantum Convolutional Neural Networks

CNNs were first introduced by Yann LeCun as a supervised classifier on data composed of image matrices [147]. They are bio-inspired neural networks modeled after the visual cortex of the brain. The local attention in the visual cortex is achieved with the idea of receptive fields [104, 105]. Receptive fields are local clusters of neighboring cells to process localized sensory input information. This concept is implemented in artificial neural networks as a 'filter'. A filter is a smaller sub-matrix mask applied to an input matrix. The entries of the sub-region of the matrix, when a filter is applied, are element-wise multiplied and summed up to produce a single output from the filter region. This operation is called 'convolution'. Several iterations of convolution reduce the size of the input data sample with entries representing higher-order patterns from the sample [80].

In general, a CNN is composed of three types of sub-networks: convolution layers, pooling layers, and fully-connected layers.

Convolution Layers

For each receptive field, a filter of the same size is used to apply a special mathematical operation called convolution. The convolution operation is defined by the sum of element-wise multiplications. The visual representation of the operation is depicted in Figure 33.

For an $n \times n$ image matrix, a kernel (filter) of size $m \times m$ with fixed values is applied where $m < n$. Starting from the top left corner of the image matrix, the kernel is overlaid with element-wise multiplications to create a convolution matrix. The concept of stride is used to determine how to slide over for the subsequent convolution operation. The resulting feature matrix is going to be of a smaller size than

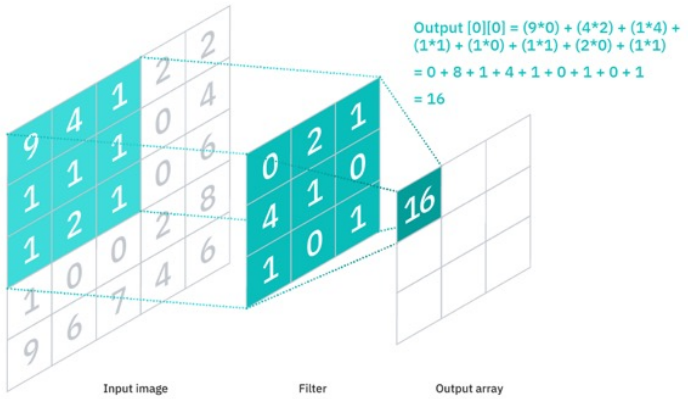

Figure 33: Convolutional operation [image source]

the original. Bias addition and nonlinear activation functions are optional.

Pooling layers

These optional layers are used to reduce the size of the convolution layers. The most commonly used size is 2×2. For each receptive field of 2×2, either max pooling or average pooling is used to reduce the 4 values to a single value. Max pooling selects the maximum value out of the four. Average pooling takes the average of the four. The resulting image matrix is a reduced matrix with a higher-order pattern extracted from the resulting matrix of the convolution layer.

Fully connected layers

The 2D feature matrix is flattened as one vector. When there are multiple feature map matrices, the flattened vectors are concatenated as one vector. Then traditional feedforward neural network layers are applied wherein every neuron is fully connected to every neuron of the next layer via matrix multiplication.

The way how convolutional operations are implemented in machine learning libraries such as TensorFlow and PyTorch [50] is realized as follows:

[50] *PyTorch is a machine learning framework based on the Torch library, used for applications such*

- Flatten the entries of the receptive field.
- Concatenate the flattened vectors as rows of a new data matrix of size number of groups $\times$ kernel size.
- Initialize a filter matrix with size kernel size $\times$ number of filters.
- Left multiply the data vector with the filter vector.
- Each column of the resulting matrix is created as new convoluted values of the new image matrix. Each row contains different values of k^{th} image matrices corresponding to the number of filters.

In converting classical CNNs into quantum networks, the main goal is effectively implementing the convolution operation. The quantum operation sequence of data encoding, quantum gate operation, and measurement should mirror the convolutional layer, pooling layer, and fully connected layer sequence. The fully connected layers are just feed-forward neural network layers, which are implemented with linear matrix multiplications. The main component in consideration is the convolutional layers.

Quantum convolutional neural networks (QCNN) implement the idea of convolution by applying filters to adjacent qubits. To reduce the number of qubits for computation, a pooling layer is implemented usually using controlled gates. The quantum states from the control qubits are used for subsequent layers and the other qubits are discarded. The first such architecture in the qubit-based model was introduced by Cong et al. as seen in Figure 34 [57]. Each U_i block in the diagram is a convolution circuit producing a state change as a result of the operation. The V_i

as computer vision and natural language processing, originally developed by Meta AI and now part of the Linux Foundation umbrella. It is free and open-source software released under the modified BSD license. Although the Python interface is more polished and the primary focus of development, PyTorch also has a C++ interface. PyTorch provides two high-level features: Tensor computing (like NumPy) with strong acceleration via graphics processing units (GPU). Deep neural networks built on a tape-based automatic differentiation system." source: Wikipedia

blocks represent the pooling operations that reduce the size of the overall number of qubits progressively. The F block represents the fully connected neural network layer. In most of the literature, classical pre-processing steps are applied for dimensionality reduction.

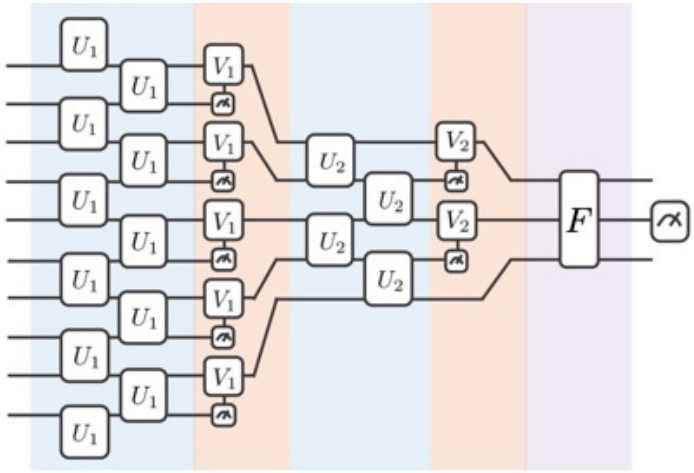

Figure 34: Quantum convolutional neural network architecture [57]

Another way of implementing a QCNN was proposed by Henderson et al. as a data pre-processing technique for classical networks [98]. In this framework, a filter applied to local receptive fields of the input data samples produces multiple channels of feature maps through the use of a random quantum circuit. The resulting feature maps are used as inputs to a CNN. Its architecture is shown in Figure 35.

The (A) layer is a classical CNN with the pre-processing quantum circuit, using QCNN. The actual training is done by the classical network. The (B) layer is a detailed snapshot of the QCNN process. The entries of the image matrix corresponding the 2×2 filter is encoded as quantum states in four qubits. Thereafter, a random quantum circuit is applied to induce a change of states. The measurement is performed on all four qubits, producing an output vector of length four. Then a pooling layer is applied, reducing the output vector of length four to one. The reduced image matrix is used as input to a classical CNN circuit. Training is done purely on the

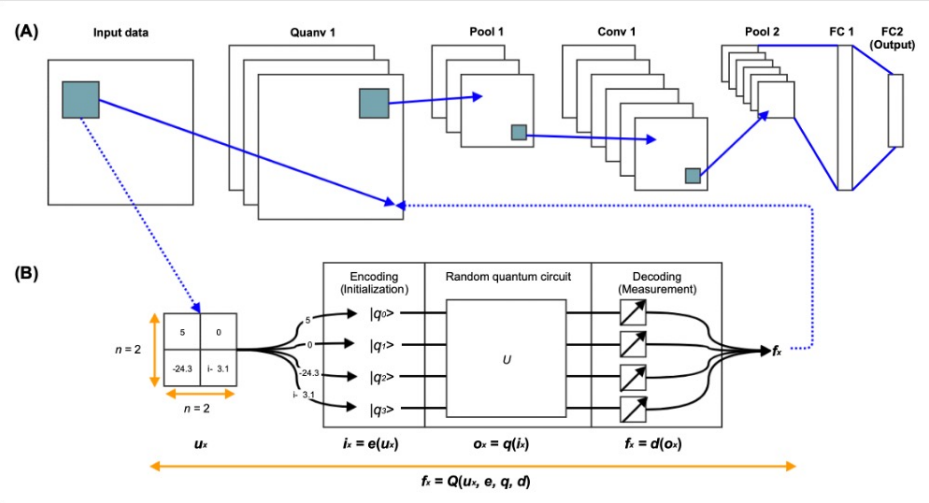

Figure 35: Quantum convolutional neural network as a pre-processing tool [98]

classical circuit.

There are various implementations of quantum neural networks and QCNN in the literature for the purpose of data classification. Due to the computing limitations of the near-term devices, they are classical-quantum hybrid networks. In the next section, we have a closer look at these quantum networks for MNIST data classification.

6.3 Quantum Multi-class Data Classifier

Classifiers are supervised machine learning algorithms providing predictions of the class of given data points, based on the hidden patterns of the given features. Quantum classifiers are quantum machine learning algorithms where data features are mapped into the higher-dimensional quantum computational space. However, due to the limitations of the near-term devices, the datasets with a high number of features cannot be directly processed with quantum circuits only. The shallow circuits in today's technology cannot encode all the values of input data features as quantum states. Quantum classifiers for these datasets are mostly classical-quantum classifiers

where various classical machine learning algorithms are employed for dimensionality reduction.

Among these quantum machine learning algorithms are quantum binary classifiers based on the binary logic of quantum computing. There are, however, only a handful of multi-class data quantum classifiers [32, 49, 140]. We examine various classifier models on the MNIST dataset.

6.3.1 Previous Quantum Classifiers on MNIST Dataset

To my knowledge, there are six quantum classifiers with actual experiments on the MNIST dataset. They can be grouped into three categories: data pre-processing network [98], binary classifiers [106], and multi-data classifiers [32, 140].

Andrea Mari implemented the QCNN pre-processing circuit proposed by Henderson et al. in 2011 [98].

The binary classifiers are:

- Google implementation [269] using TensorFlow Quantum [51] [36].
- IBM implementation using Qiskit [270]
- Hur et al. using QCNN [106]

The multi-class data classifiers in the literature classify only four classes of the MNIST dataset. They both use the qubit-based QCNN architecture:

- Lazzarin et al. using QCNN [140]

[51] *"TensorFlow Quantum (TFQ) is a quantum machine learning library for rapid prototyping of hybrid quantum-classical ML models. Research in quantum algorithms and applications can leverage Google's quantum computing frameworks, all from within TensorFlow. TensorFlow Quantum focuses on quantum data and building hybrid quantum-classical models. It integrates quantum computing algorithms and logic designed in Cirq, and provides quantum computing primitives compatible with existing TensorFlow APIs, along with high-performance quantum circuit simulators."* source: TensorFlow.org

- Bokhan et al. using QCNN [32]

This section outlines a brief description of each classifier.

Quantum Convolutional Neural Network for Pre-processing [160]

This model is a quantum-classical hybrid model wherein training occurs in a classical neural network with one layer. A QCNN is used to pre-process input images, which are saved and loaded as input to the classical layer. The pre-processing network is an implementation of the network proposed in "Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits" [98]. It is implemented with Xanadu's PennyLane using the qubit-based model simulator.

The filter of size 2×2 is encoded as four qubits and the image pixel values in the filter are encoded into quantum states as parameters of the R_y gates. The outputs from the four qubits are interpreted as four channels, producing four different filter outputs. The four separate pre-processed images are flattened as one vector and fed as input to a fully connected feed-forward layer. Although this model classifies all ten classes, the classification is done by a classical neural network, not by a quantum one.

Google Implementation [269]

The Google implementation of a quantum neural network using the MNIST dataset is a binary classifier using the classes 3 and 6. It is implemented using Google's TensorFlow Quantum. The original image samples of size 28×28 are truncated to 4×4 matrices and then the entries are binarized to 0s and 1s. A 4×4 qubit grid is initialized to $|0\rangle$ and the Pauli-X gate (NOT-gate) is applied to the qubits corresponding to the value 1 of the input matrix as seen in Figure 36.

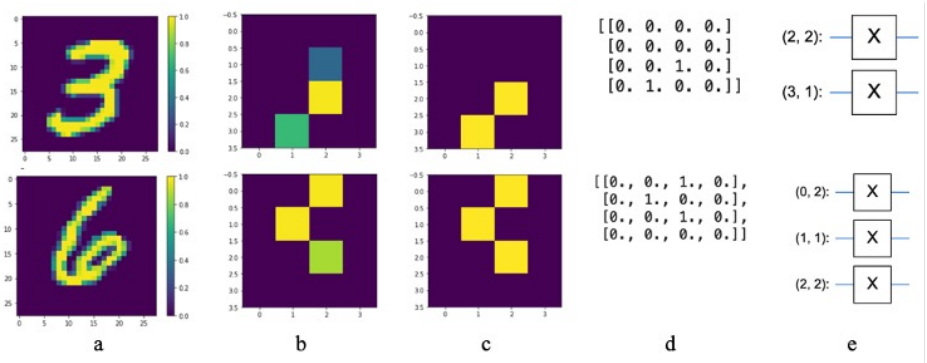

Figure 36: Simplified and binarized data encoding

The network is a pure quantum circuit, composed of XX [52] and ZZ [53]gates with a total of 32 parameters, which are optimized through the "learning" process. An additional qubit for readout is added, which is the only qubit measured after computation. The output of the circuit is single expectation values between $[-1, 1]$. The output values are interpreted as either class 3 or 6. The whole quantum circuit is wrapped as a Keras layer and is run using Hinge loss [54] and Adam optimizer [55]. The training process involves finding optimal values of the parameters for the XX gates and ZZ gates. On 10,338 samples, the model achieves 89.92% training accuracy.

IBM [270]

This binary classifier is a quantum data encoding circuit composed of the Hadam -ard gate and a single R_y gate as seen in Figure 37. It is implemented using IBM's Qiskit [56] and PyTorch [57].

[52] a two-qubit gate with two X-gates in parallel
[53] a two-qubit gate with two Z-gates in parallel
[54] explain Hinge loss
[55] *"Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments"*, Keras documentation
[56] IBM's quantum circuit and algorithm building software
[57] Python-based open-source machine learning framework

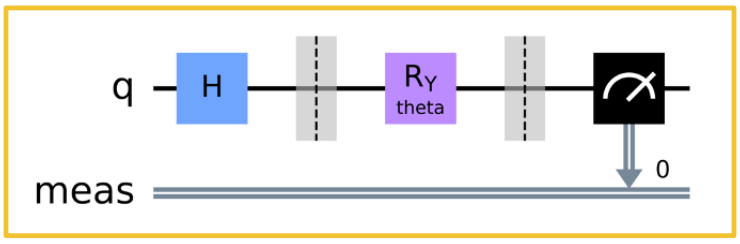

Figure 37: Data encoding circuit for single inputs

It uses a classical CNN to produce a single real-valued output. The output from the CNN is then used as the parameter for the R_y gate. The classical CNN is composed of two convolutional layers each with kernel size (5,5), max pool, and the activation function ReLU (Rectified Linear Unit) [39]. The resulting image matrices are flattened, and then one hidden layer of a feed-forward neural network is applied with the activation function ReLU.

Without any quantum neural network circuit, measurement is performed directly on the quantum encoded data. On 100 samples, the model achieves 100% training accuracy [58].

Hur's MNIST Binary Classifier [106]

This model is a binary classifier whose architecture is based on QCNN proposed by Cong et al. [57]. The classes 0 and 1 are used for the experiments. Image samples are pre-processed via Principal Component Analysis (PCA) or autoencoder to accommodate the capacity of the eight-qubit circuit. For the convolution operations on each adjacent pair of qubits, various combinations of gates are used consisting of R_y gates, Hadamard, Controlled-Z, and Controlled-Not. Pooling layers are applied after quantum convolution layers to reduce the two-qubit state to the one-qubit state. After repeated iterations of convolution and pooling layers, the result is read out from the

[58] It is not clear if the accuracy is achieved by the classical network or the quantum.

last qubit. The readout value from a measurement operation is always binary i.e., 0 or 1. Hence, multiple shots of the classifier will yield real values as probabilities of the sample belonging to class 0 or 1. Based on the qubit-encoding scheme, the model achieves 98.5% classification accuracy.

Lazzarin et al's Classifier on Four Classes [140]

This is a classical-quantum multi-classifier on four classes (0, 1, 2, and 3) of the MNIST dataset. It is based on two different variations of QCNN architectures: Tree Tensor Networks (TTN) and Multiscale Entanglement Renormalization Ansatz (MERA). These models are implemented using Google's Cirq and TensorFlow Quantum. Just like Park's model, PCA and convolution auto-encoders are used to reduce the 28×28 image matrices to vectors of length 8. The convolution auto-encoder is composed of two convolution layers and two fully connected layers. After the number of qubits is reduced through multiple layers of quantum convolution, measurements are performed on two qubits of the final convolution layer. The computational basis of two qubits has the expressive power of $2^2 = 4$ different states, which are interpreted as inference labels for the classes $0, 1, 2$ and 3. The model achieves classification accuracy of up to 93%.

Bokhan et al's Classifier on Four Classes [32]

This model is another multi-class data classifier using the classes $0, 1, 2$, and 3 of the MNIST dataset. It is implemented using Google's Cirq and TensorFlow Quantum. The input images of size 28×28 are rescaled to $16 \times 16 = 256$ and $8 = \log_2 256$ qubits are used. The precise data encoding scheme of converting 256 variables in an eight-qubit circuit is not specified in the paper. The circuit is composed of three main blocks with various filter blocks: preliminary scanning, QCNN layer with pooling, and

regular layers with pooling. The internal circuit of each filter block is composed of R_y gates and controlled-R_y gates, whose parameters are learned through the training process. Then the outputs are connected to four readout ancilla qubits via Toffoli and controlled rotation gates, producing a vector of length four. The measurement result from each readout is interpreted as inference labels. The model achieves 90.03% classification accuracy.

6.3.2 Quantum Optical Binary Classifier for Fraud Detection

The motivation for this part of my work is Killoran et al.'s binary classifier using optical quantum neural networks. I implemented Killoran's architecture using PennyLane and TensorFlow Keras on Xanadu's X8 simulator.

The binary classifier proposed in "Continuous-variable quantum neural networks" is a classical quantum hybrid network [120]. The dataset used contains 284,806 genuine and fraudulent credit card transactions with 29 features, out of which only 492 are fraudulent. The dataset is truncated to 10 features as per the paper and 1,968 samples with 1:3 ratio of fraudulent vs. genuine.

The proposed classical-quantum hybrid model has a classical neural network taking input vectors of size 10 and outputting vectors of size 14, a quantum encoding circuit, and a 2-qumode quantum neural network that outputs vectors of size 2. We can regard the output vectors as the one-hot encoding of binary classification of fraudulent vs. genuine. The architecture of the hybrid network is

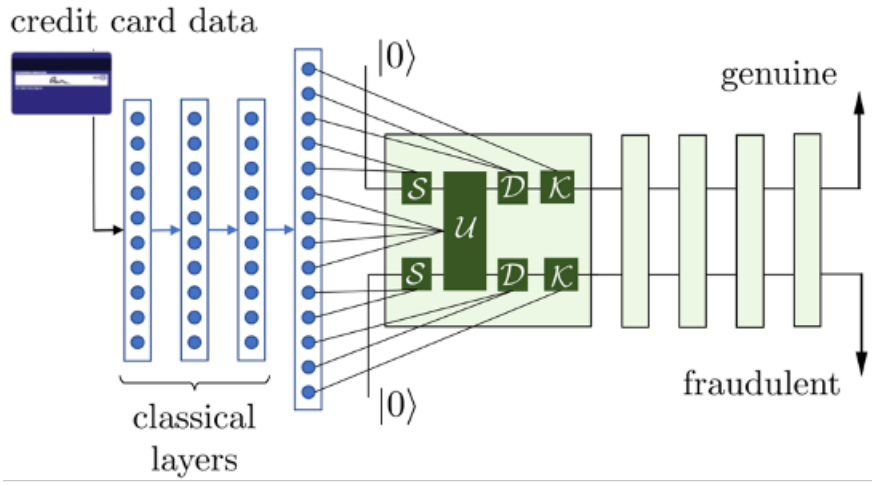

Figure 38: Binary hybrid classifier circuit [120]

The data flow of the circuit is

- Classical network: 2 hidden layers with 10 neurons, each using Exponential Linear Units (ELU) as activation function. Output layer with 14 neurons.
- Data encoding: output vector from the classical network converted to a quantum state by the circuit - squeezers, interferometer, displacement gates, and Kerr gates
- Quantum network: 4 layers of QNN. Each layer is composed of interferometer 1, squeezers, interferometer 2, displacement gates, and Kerr gates.
- Measurement: the expectation value of the Pauli$-X$ gate $\langle\phi_k| X |\phi_k\rangle$ is evaluated for each qumode state $|\phi_k\rangle$ for the k^{th} qumode.

For binary classification, two qumodes are used with the expectation value measurement method, producing a vector of length two. Figure 34 illustrates the convergence of the loss and accuracy of the model. The model achieves 97% train accuracy and 95.18% validation accuracy after 40 epochs.

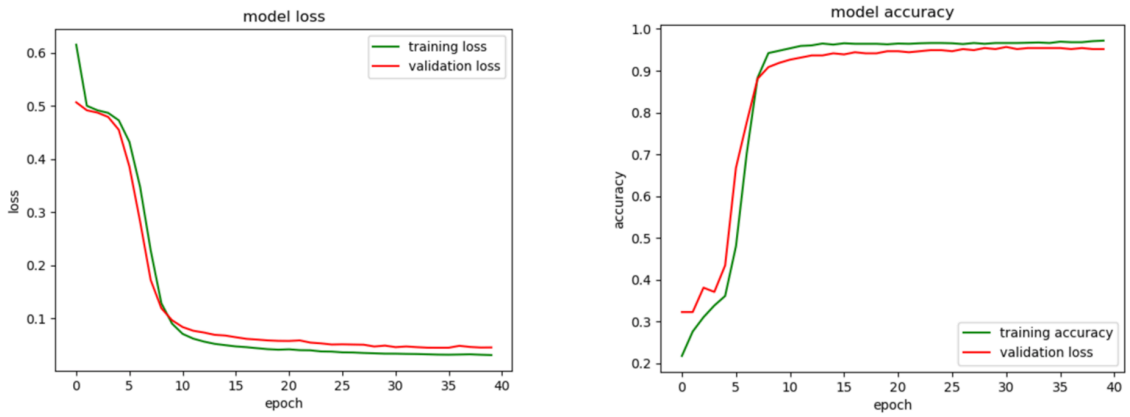

Figure 39: My experimental results

6.3.3 Optical Quantum MNIST Classifier [49]

I introduce the first quantum classifiers capable of classifying all ten classes of the MNIST dataset. In "Continuous variable quantum neural networks", Killoran et al. introduced an optical variational (parametrized) binary classifier which is a classical and quantum hybrid circuit [120]. Taking advantage of the notion of "cutoff dimension" to realize the multi-valued logic of computing, I developed quantum classifiers for multi-class data, which I applied to the MNIST dataset.

The architecture is capable of classifying up to n^m classes, where n is the cutoff dimension and m is the number of qumodes used. In theory, the model can classify datasets with up to $20^8 = 25,600,000,000$ classes, using Xanadu's X8 [59]

The classifiers presented in this section use the flexibility of determining the size of the output vectors in optical quantum computing, based on the cutoff dimension and the number of qumodes used. The output vectors are interpreted as one-hot encoded inference labels of the data samples. Different classifiers using $2, 3, 4, 5, 6,$ and 8 qumodes for computations are experimented on. On the systems of $3, 4, 6,$ and 8 qumodes, the presented multi-classifiers achieve above 99% training accuracy on

[59] In Xanadu's X8, the maximum cutoff dimension allowed is 20 and the maximum number of qumodes for computation is 8.

the truncated datasets of size 600 samples.

There are two models classifying 8 classes $0, 1, \ldots, 7$ and six models classifying 10 classes $0, 1, \ldots, 9$. The measurement outputs from these models are translated as one-hot encoded predictions of image labels. Although neural networks with one qumode are possible, they did not yield good results with different data encoding schemes and up to 25 layers.

The classifiers on 8 classes are built using $3-$qumodes and $8-$qumodes. Different measurement methods were used to produce output vectors of length 8 as seen in Table 3.

number of qumodes	cutoff dimension	measurement method	output size
3	2	probability	$2^3 = 8$
8	2	expectation value	$1 \times 8 = 8$

Table 3: Measurement methods for 8-class models

The 10-class classifiers are built on $2, 3, \ldots, 6-$qumodes. The cutoff dimensions are selected so that the output vectors exceed 10 in length. The label $k \in \{0, 1, \ldots, 9\}$ of an image matrix, when converted into a one-hot encoded vector, becomes a vector of length 10 with all zeros but the k^{th} entry as 1. For each classifier, a different number of zeros are padded to the one-hot encoded labels to match the output size of the circuit. Table 4 lists the number of padded zeros per the number of qumodes used for computation.

The classical neural network is used as a pre-processing step to reduce the original image size $28 \times 28 = 784$ to vectors of lower length that the data encoding circuit can accommodate. For the data encoding circuit, squeezers, an interferometer, displacement gates, and Kerr gates are used. The quantum circuit implementing $L(|x\rangle) = |\phi(Wx + b)\rangle$ as in the binary classifier is used.

number of qumodes	cutoff dimension	output size	number of padding 0's
2	4	$4^2 = 16$	6
3	3	$3^3 = 27$	17
4	2	$2^4 = 16$	6
5	2	$2^5 = 32$	22
6	2	$2^6 = 64$	54
7	2	$2^7 = 128$	118

Table 4: Number of padding zeros for one-hot encoding of labels

The architecture in Figure 40 depicts the data flow of image matrix → classical layers → reduced output vectors → data encoding → QNN → measurement → output vectors as one-hot encoded labels.

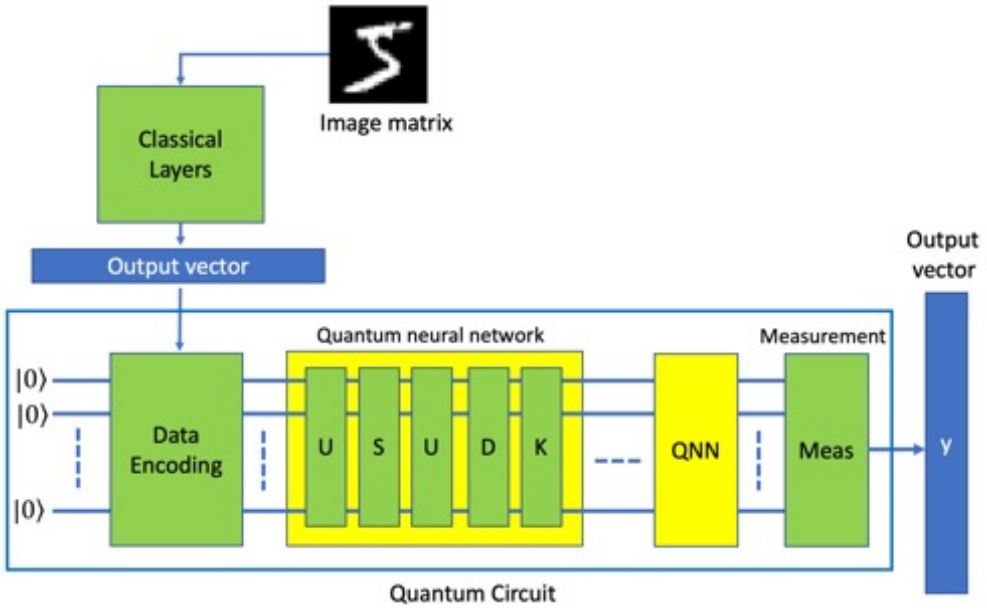

Figure 40: MNIST multi-class data classifier architecture [49]

The boxes U, S, D, and K represent interferometers, a set of m squeezers, a set of m displacement gates, and a set of m Kerr gates respectively, where m =the number of qumodes.

Classical layers

Classical feed-forward neural networks are used for pre-processing data image matrices to parameter vectors for parameterized quantum gates of the data-encoding circuit. The image matrices are flattened to vectors of size $28 \times 28 = 784$ and then reduced to vectors of smaller sizes through multiple layers. Keras dense layer operations with activation function Exponential Linear Unit are used. The output vectors are then encoded as quantum states by the data-encoding quantum circuit.

Data encoding

The output vectors from the classical neural network are in classical states. The quantum data encoding circuit converts classical states into quantum states. The quantum gates used for data encoding are squeezers, an interferometer [60], displacement gates, and Kerr gates. The entries of the classical vector are used as the parameters of these parameterized quantum gates.

An interferometer is composed of beamsplitters on pairs of adjacent qumodes and rotation matrices as seen in Figure 31. Squeezers $S(z)$ and displacement gates $D(\alpha)$ can be considered either one-parameter gates or two-parameter gates when we view the parameters $z, \alpha \in \mathbb{C}$ in Euler formula $z = a + bi = r\,(\cos\phi + i \sin\phi) = re^{i\phi}$. For the purpose of data encoding, we use them as two-parameter gates.

For an $m-$qumode circuit, the number of parameters that these gates can accommodate is $8m - 2$, where m is the number of qumodes.

squeezers	BS	rotation	displacement	Kerr
$2m$	$2(m-1)$	m	$2m$	m

Table 5: Number of parameters for data encoding

Based on these values, the size of the classical network output vectors was de-

[60] composed of beamsplitters and rotation gates

termined.

Quantum circuit

The QNN circuit implements a quantum version of the classical neural network $L(x) = \phi(Wx + b)$ as

$$L(|x\rangle) = |\phi(Wx + b)\rangle = \phi \circ D \circ U_2 \circ S \circ U_i |x\rangle$$

where $|x\rangle$ is a quantum data encoded state of the original data vector x, U_k interferometers, S squeezers, D displacement gates, and $\phi(\cdot)$ Kerr gates respectively. The number of parameters per collection of gates on $m-$qumodes is seen in Table 6.

Interferometer	Squeezers	Displacement	Kerr	Total
$2(m-1)+m$	m	m	m	$2 \cdot 2(m-1)+5m$ $= 9m-4$

Table 6: Number of parameters per QNN layer

On the 8-qumode classifier, 2 layers of QNN are applied. On the rest of the classifiers, 4 layers are applied. The number of parameters for the varying number of qumodes is shown in Table 7.

num of qumodes	per layer	total
2	14	56
3	23	92
4	32	128
5	41	164
6	50	200
8	68	272

Table 7: Number of quantum parameters

Pseudocode

To build a QNN classifier, we first decide on the number of qumodes m to use for computation. Based on that, we can calculate the number of parameters available in the data encoding circuit [61]. Then we can build a classical network with an output size equal to the number of parameters available for encoding.

For the quantum circuit, determine the cutoff dimension n such that n is the least number that satisfies $n^m >$ the number of classes [62] Then convert the outputs from the classical network into quantum states. The parameters of the quantum neural network circuit are initialized randomly. The training process is iterated until an optimal set of parameters is learned. A pseudocode explaining the data flow is shown in Figure 41.

Algorithm 1 Multi-class data quantum classifier

```
procedure QuantumCircuit
    m ← number of qumodes
    p ← number of parameters available for encoding
Classical:
    Flatten the image of 28 × 28 to 784
    Hidden layers to reduce the size of the features
    v ← output vectors of size p
Quantum:
    n ← cutoff dimension                    ▷ minimum n such that n^m > 10
    Data encoding circuit
    parameters of encoding gates ← p.
    Quantum neural network circuit
    param ← initialized parameters of the gates
    while param! = optimal do               ▷ based on the loss function
        for <number epochs> do
            <apply the neural network circuit>
            y ← measurement(result state)
            <calculate loss>
            <calculate gradients>
            <update parameters>
            param ← measurement(updated param)
save optimal parameters                     ▷ Saved model
```

Figure 41: MNIST multi-class data quantum classifier pseudocode

[61] This depends on the optical quantum gates used for data encoding.
[62] For the MNIST dataset, the number of classes is ten.

Parameter update

TensorFlow is a Python-based open-source library for neural networks, managed by Google. Keras is an open-source library that provides an application programming interface (API) over the TensorFlow library. With the PennyLane TensorFlow plug-in functionality, the quantum circuit is converted into a Keras layer and added to the classical layers. Then Keras' built-in loss functions and optimizers can be used for parameter updates. For most of the models, Categorical Crossentropy is used for the loss function and Stochastic Gradient Descent is used for the optimizer. For the 8-qumode classifier, the Mean Squared Error loss function performed better. The updated parameters are then used as the parameters of the quantum gates for the subsequent iteration of training.

Experimetal results

The $4-$qumode classifier yielded the best result of 100% training accuracy on 600 data samples. All the classifiers tested achieve above 95% training accuracy. For the 8-qumode classifier, 300 samples and 2 layers of QNN were used with 50 epochs. For the rest of the classifiers, 600 samples and 4 layers of QNN were used. With the 4-qumode classifier, training accuracy of 100% is achieved in 70 epochs.

num of qumodes	learning rate	training loss	training accuracy
2	0.02	0.4966	97.14%
3^*	0.05	0.0563	99.90%
3	0.05	0.4567	98.04%
4	0.03	0.1199	100.00%
5	0.02	0.2057	98.40%
6	0.02	0.2055	99.35%
8	0.1	0.0199	99.77%

Table 8: Training loss and accuracy

The second line 3^* indicates an $8-$class classifier. The number of parameters

given is the total number of parameters including classical and quantum. All of these classifiers followed the typical loss and accuracy graphs depicted in Figure 42.

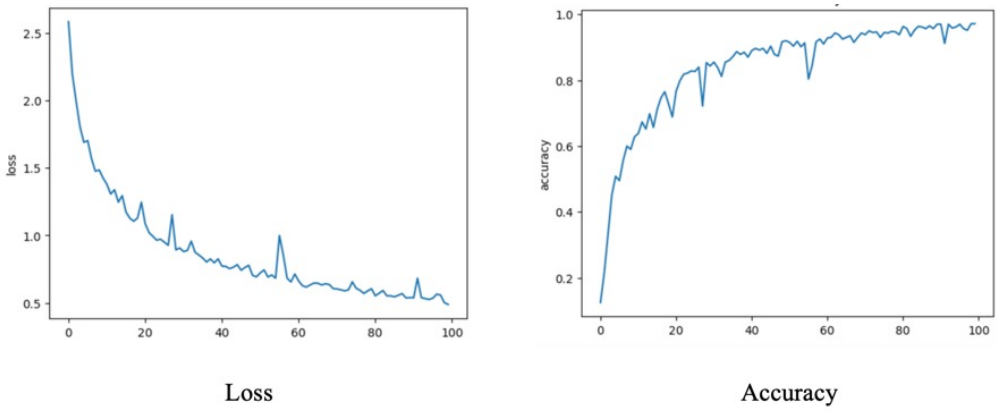

Figure 42: MNIST multi-class data quantum classifier experimental results

Further experiments with different hyper-parameters will bring the validation accuracy as high as the training accuracy.

Comparison with classical circuits

In considering the training accuracy of classical-quantum hybrid circuits, it is not very clear if the accuracy is achieved from the learned parameters from the classical network or the quantum. Here, I set up additional experiments with purely classical circuits without the quantum layers. All of the classical networks have parameters over one million. Figure 43 demonstrates the number of parameters involved in the classical portion of the hybrid network plus the output layer equivalent to the quantum measurement operation.

The results from the experiments in comparison with classical-quantum hybrid classifiers are tabulated in Table 9. The results demonstrate the contribution of the quantum circuit to vastly improve the training accuracy. This is done with a very small number of parameters compared to classical circuits. This illustrates the effectiveness of quantum neural networks and quantum computing in general.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 flatten_19 (Flatten)         (None, 784)               0

 dense_125 (Dense)            (None, 128)               100480

 dense_126 (Dense)            (None, 64)                8256

 dense_127 (Dense)            (None, 32)                2080

 dense_128 (Dense)            (None, 14)                462

 dense_129 (Dense)            (None, 10)                150

=================================================================
Total params: 111,428
Trainable params: 111,428
Non-trainable params: 0
_________________________________________________________________
```

Figure 43: Number of parameters for the classical network (Generated using Keras)

number of qumodes	total number of quantum parameters	with quantum layers	without quantum layers
2	56	0.9714	0.6717
3	92	0.9804	0.7050
4	128	1.0000	0.7983
5	164	0.9840	0.7950
6	200	0.9935	0.7483
8	272	0.9977	0.8647

Table 9: Training accuracy comparison with and without quantum layers

Using all the available optical quantum gates in the PennyLane library allows for data encoding of a larger number of features for quantum machine learning. The number of parameters available for data encoding based on the number of qumodes is listed in Table 10.

If one were to employ eight qumodes for a quantum machine learning circuit, the available number of variables for data encoding is $15 \times 8 - 7 = 113$. Theoretically, stacking multiple layers of these gates will accommodate data encoding of samples

Name	Number of qumodes for a single gate	Number of parameters	Number of parameters for m-qumodes
Beamsplitter	2	2	$2(m-1)$
Controlled Addition	2	1	$m-1$
Controlled Phase	2	1	$m-1$
Cross Kerr	2	1	$m-1$
Cubic Phase	1	1	m
Displacement	1	2	$2m$
Kerr	1	2	m
Quadratic Phase	1	2	m
Rotation	1	2	m
Squeezing	1	2	$2m$
Two Mode Squeezing	2	2	$2(m-1)$
Total			$15m-7$

Table 10: Optical quantum gates in PennyLane library

with any number of features, allowing for purely quantum circuits. I hope more attention from the quantum machine learning research community to the many advantageous features of optical quantum computing will result in the development of algorithms suitable for optical quantum machine learning.

7 SUMMARY AND CONCLUSION

We are in a new era of expanded communications/computing paradigm wherein quantum computing is emerging as an additional computational tool for solving computational problems. Decades of quantum computing research since 1980 have borne fruit in actual working QPUs. In order for quantum computers to be fully integrated into the existing computing infrastructure, providing meaningful solutions to real-life problems, these computers need to be fully fault-tolerant and universal.

For building a holistic classical and quantum computing ecosystem, the information carriers need to be compatible for easy data transfer and communications. The light signals in fiber optics communications networks are most compatible with the light signals in optical quantum computing. Optical quantum computing operates at room temperature, is easily controllable, and is robust to decoherence. Due to continual advances in laser and optical technology, optical quantum computers are easily implementable using lenses, mirrors, waveplates, diffraction gratings, and other optical instruments. The information carriers in optical quantum computing systems are the electromagnetic field modes of light.

The existing optical QPUs, demonstrating the proof-of-concept, contain the control and ALU components. The memory component for storing programs can benefit from the optical quantum memory research for quantum communications spearheaded by NIST. A uniform standard of optical signals for communications and computing facilitates quantum transducer technology. It reduces additional components and devices that need to convert other modes of information signals to optical. With optical metrology, a lot of sophisticated sensors are optics-based. Technological advances in optical instruments are guaranteed. Existing optical metrology technology widely used in sensors can be easily extended to quantum embedded-system metrology.

Optical quantum computing naturally realizes the multi-valued logic of computing at room temperature. It reduces the need to encode real-life computational problems into binary logic, perform computations, and decode to multi-valued real-life solutions. This provides an opportunity to view computational problems in a radically different way than how digital computing has been handling them since the 1940s. The only optical quantum algorithms to date are quantum neural networks [49, 120]. There is a potential to encode real-life problems directly without reformulating to binary logic. There is a potential to implement the quantum version of classical analog circuits. The inherent computing paradigm of optical quantum computers allows users to define the dimension of computational space. Different measurement methods offered by PennyLane add freedom to users to define the size of the output vectors.

Quantum machine learning algorithm research is focused on tapping into quantum mechanical properties for capturing higher-order features of data that may not be available in classical computing. Quantum machine learning circuits are characterized by parameterized (variational) gates, whose gates are learned through training to best capture hidden features. Due to the limitations in existing QPUs, quantum machine learning algorithms to date are classical-quantum hybrid models wherein the size of the data samples is reduced via various dimensionality reduction techniques.

The quantum phenomena naturally occurring in quantum optical systems beyond linear state changes provide vector addition and nonlinear operations in optical quantum computing. True implementation of quantum deep learning is possible in optical quantum computing because of the availability of displacement gate and nonlinear gates. Building a multi-class classification network on the qubit model would require a very complex scheme of using multiple qubits as readout. Due to the rich array of optical quantum gates and multiple measurement operators for producing

different-size output vectors, it is much more natural to build multi-class classification models on optical quantum computers. Xanadu's Pennylane's Tensorflow plug-in feature allows quantum circuits to be converted to Keras layers. Then Keras' built-in loss functions and optimizers are accessible for the optimization process.

My contributions to the quantum computing research community are twofold:

Theoretical contribution

- Demonstration of quantum optics as a highly desirable candidate for the physical implementation of quantum computing.
- Demonstration of the flexibility availed in optical quantum computing to the users to determine the dimension of the computational space.
- Identification of the capabilities of optical quantum computing to directly implement multi-valued logic.

Implementation of optical quantum neural networks

- Implementation of classical-quantum hybrid auto-encoder according to the architecture proposed by Killoran et al [120].
- Implementation of classical-quantum hybrid binary classifier according to the architecture proposed by Killoran et al [120].
- Creation of a novel classical-quantum hybrid multi-class data classifier on real data [49]. Existing multi-class data quantum hybrid classifiers classify up to four classes. My hybrid classifier demonstrated the classification of all ten classes.

As advances in quantum computing technology are realized, quantum machine learning circuits capable of fully encoding data samples will be tested and improved upon.

Quantum computing as a component of overall CPU, QPU, and GPU computational solutions on the foundation of classical and quantum fiber optics communications/computing infrastructure will usher in a new era of the information revolution.

References

[1] S. Aaronson and A. Arkhipov, (2011), *The computational complexity of linear optics*, Proceedings of the 43rd annual ACM Symposium on Theory of Computing.

[2] S. Aaronson and L. Chen, (2017), *Complexity-Theoretic Foundations of Quantum Supremacy Experiments*, Proceedings of the 32nd Computational Complexity Conference, 22:1–67.

[3] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, (2020), *The power of quantum neural networks*, arXiv:[quant-ph]2011.00027.

[4] C. Adami and N. Cerf, (1999), *Quantum Computation with Linear Optics*, Quantum Computing and Quantum Communications. Lecture Notes in Computer Science, 1509, Springer.

[5] J. Adamowski, S. Bednarek, and B. Szafran, (2005), *Quantum Computing with Quantum Dots*, Physics, Chemistry Schedae Informaticae.

[6] G. Adesso, S. Ragy, and A. R. Lee,(2014), *Continuous Variable Quantum Information:Gaussian States and Beyond*, Open Systems and Information Dynamics, 21(1)2: 1440001, World Scientific Publishing Company.

[7] E. AkgünErgün and A. Demir, (2018), *Modeling Course Achievements of Elementary Education Teacher Candidates with Artificial Neural Networks*, International Journal of Assessment Tools in Education 5(3).

[8] Z. Al-Wardi, (2021), *Radix-p Multiple Valued Logic Function Simplification using Higher Radix Representation*, Journal of Physics: Conference Series, 1804:012016.

[9] T. Albash and D. A. Lidar, (2018), *Adiabatic quantum computation*, Reviews of Modern Physics 90, 015002.

[10] M. D. Al-Amri, M. M. El-Gomati, and M. S. Zubairy, (2016), *Optics in our time*, Springer.

[11] J. Arrazola et al., (2021), *Quantum circuits with many photons on a programmable nanophotonic chip*, Nature 591:54–60.

[12] J. M. Arrazola et al., (2021), *Differentiable quantum computational chemistry with PennyLane*, arXiv:2111.09967v2.

[13] M. Artoni, (2016), *Electromagnetically Induced Transparency*, In book: Reference Module in Materials Science and Materials Engineering, Oxford, Elsevier.

[14] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. Srinivasan, (2015), *On the robustness of bucket brigade quantum RAM*, New Journal of Physics NJP(17):123010.

[15] F. Arute et al., (2019), *Quantum supremacy using a programmable superconducting processor*, Nature, 574:505–510.

[16] B. Arvind, N. Dutta, and R. SimonMukunda, (1995), *The Real Symplectic Groups in Quantum Mechanics and Optics*, Pramana - J Phys 45:471–497.

[17] S. Attal, *Lecture 8. Fock Spaces*, Quantum Noise Theory Lectures.

[18] D. Awschalom et al., (2021), *Development of Quantum Interconnects (QuICs) for Next-Generation Information Technologies*, PRX Quantum 2, 017002.

[19] A. Ballon, (2021), *Trapped ion quantum computers*, https://pennylane.ai/qml/demos/tutorial$_t$*rapped*$_i$*ons.html*

[20] J. Bardin, D. Slichter, and D. Reilly, (2020), *Microwaves in Quantum Computing*, IEEE Journal of Microwaves.

[21] M. Beauregard, J. Lévesque, and P. Bourgouin, (2001), *Neural correlates of the conscious self-regulation of emotion*, J. Neurosci. 2001;21:1–6.

[22] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne1, R. Salzmann, D. Scheiermann1, and R. Wolf, (2020), *Training deep quantum neural networks*, Nature Communications.

[23] S. Behnke, (2003), *Hierarchical Neural Networks for Image Interpretation*. Lecture Notes in Computer Science. Vol. 2766. Springer.

[24] P. Benioff, (1980), *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, Journal of Statistical Physics. 22(5):563–591.

[25] P. Benioff, (1982), *Quantum mechanical Hamiltonian models of turing machines*, Journal of Statistical Physics. 29(3):515–546.

[26] V. Bergholm and J. Biamonte, (2010), *Categorical Quantum Circuits*, Journal of Physics A Mathematical and Theoretical 44(24).

[27] V. Bergholm et al., (2018), *PennyLane: Automatic differentiation of hybrid quantum-classical computations*, arXiv:1811.04968.

[28] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W. K. Mok, S. Sim, L. C. Kwek, and A. Aspuru-Guzik, *Noisy intermediate-scale quantum (NISQ) algorithms*, Reviews of Modern Physics, 94:015004.

[29] J. Biamonte, *Explaining quantum phase kickback to estimate eigenvalues*, https://web.cecs.pdx.edu/ mperkows/quantum-seminar/EXPLAINING-QUANTUM-PHASE-KICKBACK.pdf

[30] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, (2017), *Quantum Machine Learning*, Nature, 549.

[31] A. Bisarya, W. E. Maouaki, S. Mukhopadhyay, N. Mishra, S. Kumar, B. K. Behera, P. K. Panigrahi, and D. De, (2020), *Breast Cancer Detection Using Quantum Convolutional Neural Networks: A Demonstration on a Quantum Computer*, medRxiv preprint.

[32] D. Bokhan, A. S. Mastiukova, A. S. Boev, D. N. Trubnikov, and A. K. Fedorov, (2022), *Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning*, arXiv:2203.15368 [quant-ph].

[33] M. Bonamente, (2016), *Three Fundamental Distributions: Binomial, Gaussian, and Poisson*, Statistics and Analysis of Scientific Data, Springer.

[34] J. E. Bourassa, R. N. Alexander, M. Vasmer, A. Patil, I.Tzitrin, T. Matsuura, D. Su, B. Q. Baragiola, S. Guha, G. Dauphinais, K. K. Sabapathy, N. C. Menicucci, I. Dhand, (2020), *Blueprint for a Scalable Photonic Fault-Tolerant Quantum Computer*, arXiv:2010.02905 [quant-ph].

[35] S. Braunstein and P. van Loock, (2005), *Quantum information with continuous variables*, Reviews of Modern Physics 77:513.

[36] M. Broughton et al., (2020), *TensorFlow Quantum: A Software Framework for Quantum Machine Learning*, arXiv:2003.02989 [quant-ph]

[37] B. Brown, (2018), *Why Computers Use Binary Numbers*, Kennesaw State University, College of Computing and Software Engineering.

[38] D. E. Browne and T. Rudolph, (2004), *Resource-efficient linear optical quantum computation*, arXiv:quant-ph/0405157.

[39] J. Brownlee, (2019), *A Gentle Introduction to the Rectified Linear Unit (ReLU)*, Machine Learning Mastery.

[40] B. Brubaker, (2022), *Versatile neutral atoms emerge as an intriguing quantum computing platform*, Research Technology, Physics Today.

[41] C. D. Bruzewicza, J. Chiaverinib, R. McConnell, and J. M. Saged, (2019), *Trapped-ion quantum computing: Progress and challenges*, Applied Physics Reviews 6, 021314.

[42] A. Burgess and M. Florescu, (2022), *Quantum memory effects in atomic ensembles coupled to photonic cavities*, arXiv:2211.08562 [quant-ph].

[43] J. Carolan et al., (2015), *Universal linear optics*, Science, Quantum Optics, 349(6249).

[44] W. Case, (2008), *Wigner functions and Weyl functions for pedestrians*, American Journal of Physics 76:937.

[45] D. Castelvecchi, (2017), *Quantum computers ready to leap out of the lab in 2017*, Nature 541:9–10.

[46] A. Chalumuri, R. KuneRaghavendra, and K. BsManoj, (2021), *A hybrid classical-quantum approach for multi-class classification* Springer, Quantum Information Processing 20(3).

[47] S. Choe, (2015), *Optimal quantum data classification*, Master's thesis in Mathematics, advisor: Dr. Steven Bleiler, Portland State University.

[48] S. Choe, (2022), *Quantum computing overview: discrete vs. continuous variable models*, arXiv:2206.07246 [quant-ph].

[49] S. Choe and M. Perkowski, (2022), *Continuous Variable Quantum MNIST Classifiers, Classical-Quantum Hybrid Quantum Neural Networks*, Journal of Quantum Information Science, 12(2).

[50] Y. J. Choi, A. Vergari, and G. van den Broeck, (2022), *Probabilistic Circuits: A Unifying Framework for Tractable Probabilistic Models*, http://starai.cs.ucla.edu/papers/ProbCirc20.pdf

[51] J. I. Cirac, P. Zoller, H. J. Kimble, and H. Mabuchi, (1997), *Quantum state transfer and entanglement distribution among distant nodes in a quantum network*, Phys. Rev. Lett. 78:3221.

[52] K. B. Clark, (2014), *Basis for a neuronal version of Grover's quantum algorithm*, Frontiers in molecular neuroscience, 7:29. https://doi.org/10.3389/fnmol.2014.00029

[53] K. B. Clark, (2011), *Live soft-matter quantum computing, in Computer Search Algorithms*, ed Salander E. C. (Hauppauge: Nova Science Publishers Inc;), 1–24.

[54] K. B. Clark, (2012), *Bioreaction quantum computing without quantum diffusion*, NeuroQuantology 10:646–654.

[55] K. B. Clark, (2010), *Origins of learned reciprocity in solitary ciliates searching grouped 'courting' assurances at quantum efficiencies*, BioSystems 99:27–41.

[56] P. Clifford and R. Clifford, (2017), *The Classical Complexity of Boson Sampling*, arXiv 1706.01260

[57] I. Cong, S. Choi, M. D. Lukin, (2019), *Quantum Convolutional Neural Networks*, Nature Physics 15:1273–1278.

[58] B. Cyphers, (2019), *The case for fiber to the Home, today: why fiber is a superior medium for 21st century broadband*, Electronic Frontier Foundation.

[59] S. Das, G. Siopsis, and C. Weedbrook, (2018), *Continuous-variable quantum Gaussian process regression and quantum singular value decomposition of non-sparse low rank matrices*, Physical Review A 97:022315.

[60] N. Deb, R. Mishu, F. I. Meem, A. E. M. Ridwan, M. M. Rahman, and M. M. Mary, (2021), *Quantum Error Correction using Quantum Convolutional Neural Network*, Bachelor's degree thesis, Brac University.

[61] A. Delgado, J. M. Arrazola, S. Jahangiri, Z. Niu, J. Izaac, C. Roberts, and N. Killoran, (2021), *Variational quantum algorithm for molecular geometry optimization*, arXiv:2106.13840v2

[62] F. Dell'Anno, S. De Siena, and F. Illuminati, (2006), *Multiphoton Quantum Optics and Quantum State Engineering*, Physics Reports, 428(2–3).

[63] J. Denker et al. (1989), *Neural network recognizer for hand-written zip code digits*, AT&T Bell Laboratories.

[64] M.C. de Oliveira and W.J. Munro, (2004), *Nonclassicality and information exchange in deterministic entanglement formation*, Physics Letters A, 320:352–359.

[65] A. Deshpande et al., (2021), *Quantum computational supremacy via high-dimensional Gaussian boson sampling*, arXiv:2102.12474 [quant-ph].

[66] D. Deutsch, (1985), *Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer*, Proceedings of the Royal Society of London A. 400(1818):97–117.

[67] D. Deutsch, (1989), *Quantum computational networks*, Proc. R. Soc. Lond. A, 425:73.

[68] D. Deutsch and R. Jozsa, (1992), *Rapid solutions of problems by quantum computation*, Proceedings of the Royal Society of London A. 439(1907):553–558.

[69] P. A. Dirac, (1988), *The quantum theory of the emission and absorption of radiation*, Special Relativity and Quantum Theory, Fundamental Theories of Physics volume 3, Springer.

[70] D. DiVincenzo. (2000), *The Physical Implementation of Quantum Computation*, Fortschritte der Physik 48(9–11):771–783.

[71] B. Duan, J. Yuan, C. H. Yu, J. Huang, and C. Y. Hsieh, (2020), *A survey on HHL algorithm: From theory to application in quantum machine learning*, Physics Letters A, 384(24):126595.

[72] T. K. Elgin, Y. Huang, and V. Narayanan, (2018), *A Survey on Quantum Machine Learning*, CS University Maryland

[73] M. Epstein, (1982), *Fiber optics in medicine*, Crit Rev Biomed Eng 7(2):79-120.

[74] E. Farhi and H. Neven, (2018), *Classification with Quantum Neural Networks on Near Term Processors*, arXiv:1802.06002v2.

[75] A. Ferraro, S. Olivares, and M. G. A. Paris, (2005), *Gaussian states in continuous variable quantum information*, Bibliopolis.

[76] R. Feynman, (1982), *Simulating Physics with Computers*, International Journal of Theoretical Physics. 21(6/7): 467–488.

[77] R. Feynman, (1986), *Quantum mechanical computers*, Foundations of Physics. Springer Science and Business Media LLC. 16(6):507–531.

[78] M. Fleischhauer and M.D. Lukin, (2002), *Quantum memory for photons: dark-state polaritons*, Phys. Rev. A, 65:022314.

[79] M. Fox, (2006), *Quantum Optics, an Introduction*, Oxford University Press.

[80] K. Fukushima, (1980), *Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position.* Biological Cybernetics, 36(4):193–202.

[81] C. Genes, A. Mari, D. Vitali, and P. Tombesi, (2009), *Quantum effects in optomechanical systems*, Advances In Atomic, Molecular, and Optical Physics 57:33-86.

[82] A. Géron, (2019), *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Sebastopol, CA: O'Reilly Media. ISBN 978-1-492-03264-9., pp. 448.

[83] V. Giovannetti1, S. Lloyd, and L. Maccone, (2008), *Architectures for quantum random access memory*, Phys. Rev. Lett. 100:160501.

[84] A. V. Gitin, (2012), *Mathematical Fundamentals of Modern Linear Optics*, International Journal of Antennas and Propagation, 2012:15

[85] O. Glöckl, U. L. Andersen, and G. Leuchs, (2006), *Verifying continuous-variable entanglement of intense light pulses*, Physical Review A, 73:012306.

[86] A.V. Gorshkov, A. André, M. Fleischhauer, A. S. Sørensen, and M. D. Lukin, (2007), *Universal Approach to Optimal Photon Storage in Atomic Media*, Physical Review Letters, 98:123601.

[87] D. Gottesman, A. Kitaev, and J. Preskill, (2001), *Encoding a qubit in an oscillator*, Phys. Rev. A 64:012310.

[88] T.M. Graham, Y. Song, and J. Scott et al., (2022), *Multi-qubit entanglement and algorithms on a neutral-atom quantum computer*, Nature 604:457–462.

[89] E. Grefenstette, P. Blunsom, N. de Freitas, K. M. Hermann, (2014), *A Deep Architecture for Semantic Parsing*, arXiv:1404.7296 [cs.CL]

[90] K. Grobe, (2018), *Wavelength division multiplexing*, Reference Module in Materials Science and Materials Engineering, Elsivier.

[91] S. Groblacher, S. Gigan, and M. Paternostro, (2013), *Phase-space behavior and conditional dynamics of an optomechanical system*, Physical Review A, 88:023813.

[92] L. K. Grover, (1996), *A fast quantum mechanical algorithm for database search*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery: 212–221.

[93] Habibi, Aghdam, Hamed, (2017) *Guide to convolutional neural networks : a practical application to traffic-sign detection and classification*, Heravi, Elnaz Jahani. Cham, Switzerland. ISBN 9783319575490. OCLC 987790957.

[94] R. Hakani, (2015), *Optical time division multiplexing (OTDM)*, a Seminar Report, Nirma University, Institute of Technology, Ahmadabad.

[95] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, (2017), *Gaussian boson sampling*. Physical Review Letters, 119:170501, arXiv:1612.01199

[96] C. Hamilton, R. Kruse, L. Sansoni, Sonja Barkhofen, Christine Silberhorn, Igor Jex, (2017), *Gaussian boson sampling*, Physic Review Letters, arXiv:1612.01199

[97] A. W. Harrow, A. Hassidim, and S. Lloyd, (2009). *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. 103(15):150502.

[98] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, (2020), *Quanvolutional neural networks: powering image recognition with quantum circuits*, Quantum Machine Intelligence.

[99] G. Hetet, A. Peng, M. T. Johnson, J. J. Hope, and P. K. Lam, (2008), *Characterization of electromagnetically-induced-transparency-based continuous-variable quantum memories*, Physical review A, Atomic, molecular and optical physics, 77(1):12323.

[100] A. Hodgkin and A. Huxley, (1952), *A quantitative description of membrane current and its application to conduction and excitation in nerve*, The Journal of Physiology. 117(4):500–44.

[101] A. L. Hodgkin and A. F. Huxley AF, (1952), *A quantitative description of membrane current and its application to conduction and excitation in nerve*, The Journal of Physiology, 117(4):500–44.

[102] G. Hinton, (2012), *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - 1(1):1097–1105.

[103] Wenjun Hou and M. Perkowski, (2020), *Quantum-based algorithm and circuit design for bounded Knapsack optimization problem*, Computer Science, Quantum Information and Computation.

[104] D. Hubel, T. Wiesel, (1959). *Receptive fields of single neurons in the cat's striate cortex*. J. Physiol. 148(3):574–91.

[105] D. Hubel, T. Wiesel, (1968), *Receptive fields and functional architecture of monkey striate cortex*, The Journal of Physiology, 195(1):215–243.

[106] T. Hur, L. Kim, and D. K. Park, (2022), *Quantum convolutional neural network for classical data classification*, Quantum Machine Intelligence, 4(3).

[107] F. Idachaba, D. U. Ike, and O. Hope, (2014), *Future Trends in Fiber Optics Communication*, Proceedings of the World Congress on Engineering.

[108] Z. Idris, (2020), *Quantum Physics Perspective on Electromagnetic and Quantum Fields Inside the Brain* The Malaysian journal of medical sciences, MJMS, 27(1):1–5.

[109] S. Jahangiri, J. M. Arrazola, and A. Delgado, (2020), *Quantum Algorithm for Simulating Single-Molecule Electron Transport*, arXiv:2012.09231

[110] P. K. Jain and K. Ahmad, (1995), *5.1 Definitions and basic properties of inner product spaces and Hilbert spaces*, Functional Analysis (2nd ed.). New Age International.

[111] M. T. Johnson, J. J. Hope, and P. K. Lam, (2008), *Characterization of electromagnetically-induced-transparency-based continuous-variable quantum memories*, Physical review A, Atomic, molecular, and optical physics.

[112] B. Julsgaard, J. Sherson, J. Ignacio Cirac, J.r Fiurášek, and E. S. Polzik, (2004), *Experimental demonstration of quantum memory for light*, Nature, 432:482–486.

[113] B. Kannan, A. Almanakly, and Y. Sung et al., (2023), *On-demand directional microwave photon emission using waveguide quantum electrodynamics*, Nature, Physics.

[114] A. Karimi1,2 · M. K. Tavassoly, (2015), *Generation of entangled coherent-squeezed states: their entanglement and nonclassical properties*, Quantum Inf Process.

[115] B. K. Kaushik, (2007), *Future VLSI interconnects: Optical fiber or carbon nanotube - A review*, Microelectronics International.

[116] F. S. Khan and M. Perkowski, (2005), *Synthesis of Ternary Quantum Logic Circuits by Decomposition*, arXiv:quant-ph/0511041.

[117] F. S. Khan and M. Perkowski, (2005), *Synthesis of multi-qudit Hybrid and d-valued Quantum Logic Circuits by Decomposition*, arXiv:quant-ph/0511019.

[118] M. H. Khan, M. Perkowski, R. K. Mujibur, and K. Pawel, (2005) *Ternary GF-SOP Minimization using Kronecker Decision Diagrams and Their Synthesis with Quantum Cascades*, Journal of Multi-Valued Logic Soft Computing, 11:567–602.

[119] D. Kielpinski, V. Meyer, M. A. Rowe, C. A. Sackett, W. M. Itano, C. Monrow, AND D. J. Wineland, (2001), *A decoherence-free quantum memory using trapped ions*, Science, 201(5506).

[120] N. Killoran, T. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, (2019), *Continuous variable quantum neural networks*, Physical Review Research 1:033063.

[121] H. J. Kimble, (2008), *The quantum internet*, Nature, 453(7198):1023-30.

[122] A. Kitaev, (1996), *Quantum measurements and the Abelian Stabilizer Problem*, Electron. Colloquium Comput. Complex. 3.

[123] E. Knill, R. Laflamme, and G. J. Milburn, (2000), *Efficient Linear Optics Quantum Computation*, arXiv:quant-ph/0006088.

[124] E. Knill, R. Laflamme, and G. J. Milburn, (2001), *A scheme for efficient quantum computation with linear optics*, Nature, 409:46–52.

[125] P. Kok and B. Lovett, (2010), *Introduction to Optical Quantum Information Processing*, Cambridge University Press.

[126] S. Kotiyal, H. Thapliyal and N. Ranganathan, (2012), *Mach-Zehnder Interferometer Based All Optical Reversible NOR Gates*, IEEE Computer Society Annual Symposium on VLSI.

[127] D. Krebs, (2007), *Introduction to Kernel Methods*, cs.pitt.edu.

[128] A. Krizhevsky, I. Sutskever, G. E. Hinton, (2012), *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems. 1:1097–1105.

[129] A. Krizhevsky, I. Sutskever, and G. Hinton, (2017), *ImageNet classification with deep convolutional neural networks*, Communications of the ACM. 60(6):84–90.

[130] Y. Kwak, W. J. Yun, S. Jung, and J. Kim *Quantum Neural Networks: Concepts, Applications, and Challenges*, Twelfth International Conference on Ubiquitous and Future Networks (ICUFN).

[131] A.E. Kozhekin, K.Mølmer, and E.S. Polzik, (2000), *Quantum memory for light*, Phys. Rev. A, 62:033809.

[132] P. Lambropoulos and D. Petrosyan, (2007), *Fundamentals of Quantum Optics and Quantum Information*, Springer.

[133] A. M. Lance, T. Symul, W. P. Bowen, B. C. Sanders, T. Tyc, T. C. Ralph, and P. K. Lam, (2004), *Continuous Variable Quantum State Sharing via Quantum Disentanglement*, Physical review A, Atomic, molecular, and optical physics 71(3).

[134] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White , (2010), *Towards quantum chemistry on a quantum computer*, Nature Chemistry, 2:106–111.

[135] R. LaPierre, (2021), *Introduction to Quantum Computing*, The Materials Research Society Series, Springer.

[136] R. R. Larsen, (1971), *Photons on Photons*, Nature, 233:144.

[137] W. A. Latif and S. Ggha, (2019), *Understanding Neurobehavioural Dynamics: A Close-Up View on Psychiatry and Quantum Mechanics*, The Malaysian journal of medical sciences : MJMS 26(1) :147-156.

[138] H. K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, (2017), *Quantum machine learning over infinite dimensions*, Phys Rev Lett, 24;118(8):080501.

[139] S. Lawrence, L. Giles, A. C. Tsoi, and A. D. Back, (1997), *Face Recognition: A Convolutional Neural Network Approach*, IEEE Transactions on Neural Networks, 8(1):98–113.

[140] M. Lazzarin, D. Emilio Galli, E. Prati, (2022), *Multi-class quantum classifiers with tensor network circuits for quantum phase recognition*, Physics Letters A, 434:128056.

[141] Y. LeCun, Yoshua Bengio, (1995), *Convolutional networks for images, speech, and time series*. In Arbib, Michael A. (ed.). The handbook of brain theory and neural networks (Second ed.). The MIT press. pp. 276–278.

[142] Y. LeCun, Yoshua Bengio, Geoffrey Hinton, (2015), *Deep learning*, Nature, 521(7553):436–444.

[143] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, and L.D. Jackel, (1989), *Backpropagation Applied to Handwritten Zip Code Recognition*, Neural Computation 1:541-551.

[144] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, (2009), *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations*, Proceedings of the 26th Annual International Conference on Machine Learning – ICML '09. ACM, 609–616.

[145] X. Li, J. Chen, P. Voss, J. Sharping, and P. Kumar, (2004), *All-fiber photon-pair source for quantum communications: Improved generation of correlated photons*, Optics Express.

[146] H. Liu, C. H. Yu, Y. Wu, S. Pan, S. Qin, F. Gao, and Q. Wen, (2019), *Quantum algorithm for logistic regression*, arXiv:1906.03834 [quant-ph].

[147] C. Y. Liu, C. Chen, C. T. Chang, and L. M. Shih, (2013), *Single-hidden-layer feed-forward quantum neural network based on Grover learning*, Neural Netw. 45:144–150.

[148] S. Lloyd and S. Braunstein, (1999), *Quantum computation over continuous variables*, Physics Review Letters, 82:1784.

[149] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, (2020), *Quantum embeddings for machine learning*, arXiv:2001.03622.

[150] D. Loss and D. DiVincenzo, (1997), *Quantum computation with quantum dots*, Physical Review A.

[151] A. I. Lvovsky, (2019), *Squeezed Light*, A Guide to Experiments in Quantum Optics.

[152] A. I. Lvovsky, B.C. Sanders, and W. Tittel, (2009), *Optical Quantum Memory*, Nature Photonics, 3:706-714.

[153] L. Ma, O. Slattery, and X. Tang, (2018), *Noise reduction in optically controlled quantum Memory*, Modern Physics Letters B, 32(14):1830001.

[154] L. Ma, O. Slattery, and X. Tang, (2017), *Optical quantum memory based on electromagnetically induced transparency*, Journal of Optics, 1:043001.

[155] L. Ma, X. Tang, and O. Slattery, (2022), *Optical quantum memory and its applications in quantum communication systems*, J Res Natl Inst Stand Technol, 125:125002.

[156] I. MacCormack, C. Delaney, A. Galda, N. Aggarwal, and P. Narang, (2022), *Branching Quantum Convolutional Neural Networks*, Physical Review Research, 4:013117.

[157] L. S. Madsen, et al., (2022), *Quantum computational advantage with a programmable photonic processor*, Nature, 606:75–81.

[158] S. B. Mandal, A. Chakrabarti, and S. Sur-Kolay, (2014), *Synthesis of Ternary Grover's Algorithm*, IEEE 44th International Symposium on Multiple-Valued Logic, 2014:184-189.

[159] Y. Manin, (1980), *Vychislimoe i nevychislimoe [Computable and Noncomputable]* (in Russian), Sov.Radio, 13–15.

[160] A. Mari, (2021), *Quanvolutional neural networks*, *https* : *//pennylane.ai/qml/demos/tutorial*$_q$*uanvolution.html*

[161] L. M. Martinez and J. M. Alonso, (2003), *Complex receptive fields in primary visual cortex*, The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry, 9(5):317-31.

[162] A. R. Matanin et al., *Towards highly efficient broadband superconducting quantum memory*, arXiv 2207.14092.

[163] P. K. Meher, J. Valls, T. Juang, K. Sridharan, and K. Maharatna, (2009), *50 Years of Cordic: Algorithms, Architectures, and Applications*, IEEE Transactions on Circuits and Systems: Regular Papers, 56:1893-1907.

[164] M. Menotti et al., (2018), *Nonlinear coupling of linearly uncoupled resonators*, Physics Review Letters.

[165] K. Michielsen, K. De Raedt, and H. De Raedt, (2005), *Simulation of Quantum Computation: A Deterministic Event-Based Approach*, Journal of Computational and Theoretical Nanoscience 2(2).

[166] I. Mikhail, (2009), *Tensor Algebra and Tensor Analysis for Engineers: With Applications to Continuum Mechanics*. Springer.

[167] G.J. Milburn and S. L. Braunstein, (1999), *Quantum teleportation with squeezed vacuum states*, Physical Review A, 60:937.

[168] D. M. Miller M. A. Thornton, (2008), *Multiple Valued Logic: Concepts and Representations*, Morgan Claypool Publishers.

[169] S. E. Miller, (1977), *Photons in fibers for telecommunication*, Science, 195(4283):1211.

[170] D. F. Milne, (2012), *Towards universal quantum computation in continuous-variable systems*, A Thesis Submitted for the Degree of PhD at the University of St. Andrews.

[171] M. Mottonen, J. J. Vartiainen, V. Bergholm, M. M. Salomaa, (2004), *Quantum circuits for general multiqubit gates*, Phys. Rev. Lett. 93:130502.

[172] D. Mumford, (1991), *On the computational architecture of the neocortex I. The role of the thalamo-cortical loop*, Biological Cybernetics, 65:135-145.

[173] S. Murshid, B. Grossman, and P. Narakorn, (2008), *Spatial domain multiplexing: A new dimension in fiber optic multiplexing*, Optics and Laser Technology.

[174] J. Neergaard-Nielsen, (2008), *Generation of single photons and Schrodinger kitten states of light*, PhD Thesis, Technical University of Denmark.

[175] M. Nielsen, (2003), *Quantum computation by measurement and quantum memory*, Phys. Lett. A, 308(2-3):96–100, arXiv:quant-ph/0108020.

[176] M. Nielsen, (2006), *Cluster-state quantum computation*, Reports on Mathematical Physics, 57(1):147-161.

[177] M. Nielsen, (2004), *Optical quantum computation using cluster states*, Phys. Rev. Lett., 93(4):040503.

[178] M. Nielsen and I. Chuang, (2010), *Quantum computation and quantum information*, Cambridge University Press.

[179] M. A. Novotny, (2017), *Inhomogeneous computer interconnects for classical and quantum computers*, US Patent 2017/03715A.

[180] K.S. Oh, K. Jung, (2004), *GPU implementation of neural networks*, Pattern Recognition, 37(6):1311–1314.

[181] K. Parthasarathy, (2010), *What is a Gaussian State?*, Communications of Statistical Analysis, 4(2).

[182] M. Perkowski, (2022), *Inverse Problems, Constraint Satisfaction, Reversible Logic, Invertible Logic and Grover Quantum Oracles for Practical Problems*, Science of Computer Programming, 218:102775.

[183] I. Piatrenka and M. Rusek, (2022), *Quantum Variational Multi-class Classifier for the Iris Data Set*, Computational Science - ICCS Lecture Notes in Computer Science, 13353 Springer, Cham.

[184] J. M. Pino, J. M. Dreiling, C. Figgatt et al., (2021) *Demonstration of the trapped-ion quantum CCD computer architecture*, Nature, 592:209–213.

[185] S. Pirandola, S. Mancini, S. Lloyd and S. L. Braunstein, (2008), *Continuous-variable quantum cryptography using two-way quantum communication*, Nature Physics, 4:726–730.

[186] O. Pfister, (2019), *Continuous-variable quantum computing in the quantum optical frequency comb*, Journal of Physics B: Atomic, Molecular and Optical Physics, 53:012001.

[187] A. Polkovnikov, (2010), *Phase space representation of quantum dynamics*, Annals of Physics.

[188] J. Preskill, (2018), *Quantum computing in the NISQ era and beyond*, Quantum, 2:79.

[189] I. Prokhorenkov and L. Spice, *Quantum field theory*, Department of Mathematics, Texas Christian University.

[190] M. Rasavi and J. Shapiro, (2005), *Long-distance quantum communication with neutral atoms*, Physical review A, Atomic, molecular, and optical physics, 73(4).

[191] R. Raussendorf and H. J. Briegel, (2001), *A one-way quantum computer*, Phys. Rev. Lett., 86(22):5188–5191.

[192] R. Raussendorf, D. E. Browne, and H. J. Briegel, (2002), *The one-way quantum computer — a non-network model of quantum computation*, J. Mod. Opt., 49(8):1299–1306.

[193] P. Rebentrost et al, (2019), *Quantum gradient descent and Newton's method for constrained polynomial optimization*, New Journal of Physics, 21:073023.

[194] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani., (1994), *Experimental realization of any discrete unitary operator*, Phys. Rev. Lett., 73(1):58–61.

[195] J. R. Reimers, L. K. McKemmish, R. H. McKenzie, A. E. Mark, and N. S. Hush (2009), *Weak, strong, and coherent regimes of Fröhlich condensation and their*

applications to terahertz medicine and quantum consiousness, Proc. Natl. Acad. Sci. U.S.A. 106:4219–4224.

[196] B. Rogers, N. LoGullo, G. DeChiara, G. M. Palma, and M. Paternostro, (2014), *Hybrid optomechanics for Quantum Technologies*, Quantum Measurements and Quantum Metrology.

[197] F. Rosenblatt, (1958), *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65(6):386–408.

[198] T. Sasao, (1988), *Multiple-valued logic and optimization of programmable logic arrays*, Computer, 21(4):71-80.

[199] N. Schetakis, D. Aghamalyan, M. Boguslavsky, and P. Griffin, (2021), *Binary classifiers for noisy datasets: a comparative study of existing quantum machine learning frameworks and some new approaches*, arXiv:2111.03372v1 [quant-ph].

[200] P. Sekatski, B. Sanguinetti, E. Pomarico, N. Gisin, and C. Simon, (2010) *Cloning entangled photons to scales one can see*, Physical Review A, 82:053814.

[201] M. Schuld, (2021), *Supervised quantum machine learning models are kernel methods*, arXiv:2101.11020v2.

[202] M. Schuld, V. Bergholm, C. Gogolm, J. Izaac, and N. Killoran, (2018), *Evaluating analytic gradients on quantum hardware*, Physical Review A.

[203] M. Schuld, A. Bocharou, K. Svore, and N. Wiebe, (2018), *Circuit-centric quantum classifiers*, Physical Review A.

[204] M. Schuld, M. Fingerhuth, and F. Petruccione, (2017), *Implementing a distance-based classifier with a quantum interference circuit*, Europhysics Letters, 119(6).

[205] M. Schuld and N. Killoran, (2019), *Quantum machine learning in feature Hilbert spaces*, Physical Review Letters, 122:040504.

[206] M. Schuld, I. Sinayskiy, and F. Petruccione, (2016), *Prediction by linear regression on a quantum computer*, Physical Review A, 94:022342.

[207] M. Schuld, R. Sweke, and J. J. Meyer, (2021), *Effect of data encoding on the expressive power of variational quantum machine learning models*, Physical Review A, 103:032430.

[208] N. Schwaller, V. Vento, C. and Galland, (2022), *Experimental QND measurements of complementarity on two-qubit states with IonQ and IBM Q quantum computers*, Quantum Inf Process, 21:75.

[209] J. M. Schwartz, H. P. Stapp, and M. Beauregard, (2005), *Quantum physics in neuroscience and psychology: a neurophysical model of mind–brain interaction*, Philosophical Transactions of the Royal Society B: Biological Sciences.

[210] A. Serafini, A. Retzker, and M. B. Plenio, (2008), *Manipulating the quantum information of the radial modes of trapped ions: Linear phononics, entanglement generation, quantum state transmission and non-locality tests*, New Journal of Physics.

[211] E. A. Sete, W. J. Zeng, and C. T. Rigetti, (2016), *A Functional Architecture for Scalable Quantum Computing*, IEEE International Conference on Rebooting Computing (ICRC).

[212] Y. Shen, Nicholas C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, (2017), *Deep learning with coherent nanophotonic circuits*, Nature Photonics.

[213] P.W. Shor, (1994), *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124–134.

[214] Q. Si, R. Yu and E. Abrahams, (2016), *High-temperature superconductivity in iron pnictides and chalcogenides*, Nature Reviews Materials, 1:16017.

[215] D. R. Simon, (1997), *On the Power of Quantum Computation*, SIAM Journal on Computing, 26(5):1474–1483.

[216] R. J. Solomonoff, (1964), *A Formal Theory of Inductive Inference. Part II*, Information and Control. 7(2):224–254.

[217] R. Sotelo and T. L. Frantz, (2022)*Preparing for the Quantum Future: Perspectives of an Entrepreneurial Innovator,* IEEE Engineering Management Review, 50(3):13-16.

[218] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, (2014), *Dropout: A Simple Way to Prevent Neural Networks from overfitting*, Journal of Machine Learning Research. 15(1):1929–1958.

[219] R. S. Stancovic, J. T. Astola, and C. Moraga, (2012), *Representation of Multiple-Valued Logic Functions*, Morgan Claypool Publishers

[220] J. B. Stochel, (1997), *Representation of generalized annihilation and creation operators in Fock space*, Universitatis Iagellonicae Acta Mathematica, 34:135–148.

[221] D. Su, I. Dhand, L. G. Helt, Z. Vernon, and K. Brádler, (2019), *Hybrid spatiotemporal architectures for universal linear optics*, Physical Review A, 99:062301.

[222] P. Sutter, (2021), *What is a superconductor?*, LiveScience News.

[223] R. Sweke, F. Wilde, J. J. Meyer, M. Schuld, P. K. F Faehrmann, B. Meynard-Piganeau, and J. Eist, (2021), *Stochastic gradient descent for hybrid quantum-classical optimization*, arXiv:1910.01155 [quant-ph].

[224] C. Szegedy et al., (2015), *Going deeper with convolutions*, IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society. pp. 1–9.

[225] A. Tait, M. Nahmias, B. Shastri, and P. Prucnal, (2014), *Broadcast and weight: an integrated network for scalable photonic spike processing*, Journal of Lightwave Technology.

[226] K. Tan, M. Menotti, Z. Vernon, J. E. Sipe, M. Liscidini, and B. Morrison1, (2019), *Stimulated four-wave mixing in linearly uncoupled resonators*, Optics Letters.

[227] P. Tiwari and M. Melucci, (2018), *Multi-class Classification Model Inspired by Quantum Detection Theory*, arXiv:1810.04491 [cs.LG].

[228] R. R. Tucci, (2008), *An Introduction to Cartan's KAK Decomposition for QC Programmers*, arXiv:quant-ph/0507171.

[229] A. M. Turing, (1937), *On Computable Numbers, with an Application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society. Series 2. 42(1):230–265.

[230] A. M. Turing, (1950), *Computing Machinery and Intelligence*. Mind 49:433-460.

[231] I. Tzitrin, T. Matsuura, R. Alexander, G. Dauphinais, J. E. Bourassa, K. Sabapathy, N. Menicucci, and I. Dhand, (2021), *Fault-tolerant quantum computation with static linear optics*, PRX Quantum, Physics Review Journal.

[232] L. Vaidman, (1994), *Teleportation of quantum states*, Physical Review A, 49:1473.

[233] V. Vaidya et al., (2019), *Broadband quadrature-squeezed vacuum and nonclassical photon number correlation from a nanophotonic device*, Science Advances.

[234] P. Velanas, (2010), *Photonic devices based on optical fibers for telecommunication applications*, Academia.com.

[235] G. Vidal1, (2008), *A class of quantum many-body states that can be efficiently simulated*, Phys. Rev. Lett. 101:110501¿

[236] D. F. Walls and G. J. Milburn, (1994), *Quantum Optics*, Springer.

[237] A. Wallucks, I. Marinković, B. Hensen, R. Stockill, and S. Gröblacher, (2020), *A quantum memory at telecom wavelengths*, Nature Physics volume 16:772–777.

[238] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, (2005), *Experimental one-way quantum computing*, Nature, 434:169–176.

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

[240] H. G. Weber and R. Ludwig, (2008), *Ultra-high-speed OTDM transmission technology*, Optical Fiber Telecommunications V B Optical fiber telecommunications V, Academic Press.

[241] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, (2012), *Gaussian Quantum Information*, Reviews of Modern Physics, 84:621.

[242] S. J. Wei, Y. H. Chen, Z. R. Zhou, and G. L. Long, (2022), *A quantum convolutional neural network on NISQ devices*, AAPPS Bulletin.

[243] E. W. Weisstein, (2022), *Hilbert Space*, From MathWorld–A Wolfram Web Resource.

[244] E. P. Wigner, (1957), *Relativistic invariance and quantum phenomena*, Rev. Mod. Phys. 29:255–268.

[245] E. P. Wigner, (1981), *Review of the quantum mechanical measurement problem, in Quantum Optics, Experimental Gravity, and Measurement Theory*, eds Meystre P., Scully M. O. (New York, NY: Plenum Press), 43–63.

[246] P. Wittek, (2014), *Quantum Machine Learning: What Quantum Computing Means to Data Mining*, Elsevier Insights, Academic Press.

[247] D. de Wolff, (2021), *Accelerating AI at the speed of light Yichen Shen PhD '16 is CEO of Lightelligence, an MIT spinout using photonics to reinvent computing for artificial intelligence*, MIT Startup Exchange, MIT News.

[248] C. H. H. Yang et al., (2021), *Decentralizing Feature Extraction with Quantum Convolutional Neural Network for Automatic Speech Recognition*, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6523-6527.

[249] H. P. Yuen, (1976), *Two-photon coherent states of the radiation field*, Physical Review A, 13:2226.

[250] A. Zhang, X. He, and S. Zhao, (2022), *Quantum classification algorithm with multi-class parallel training*. Quantum Inf Process 21:358.

[251] Y. Zhang et al., (2021), *Squeezed light from a nanophotonic molecule*, Nature Communications.

[252] H. S. Zhong et al., (2020), *Quantum computational advantage using photons*, Science 10.1126.

[253] S. Zhong, Yan Liu, and Yang Liu, (2011), *Bilinear deep learning for image classification*, Proceedings of the 19th ACM international conference on Multimedia, P343 352.

[254] T. Zhou and H. Jia, (2018), *Method to optimize optical switch topology for photonic network-on-chip*, Optics Communications, 413:230-235.

[255] Y. Zoabi, S. Deri-Rozov, and N. Shomron, (2021), *Machine learning-based prediction of COVID-19 diagnosis based on symptoms*, npj Digit. Med. 4(3).

[256] (2019), *Quantum Computing: Progress and Prospects*, Consensus Study Report, The National Academic Press.

[257] Linear Algebra, (2011), *Singular Value Decomposition*, MIT OpenCourseWare

[258] (2018), *National Strategic Overview for Quantum Information Science*, National Science and Technology Council.

[259] (2019), *Quantum Supremacy Using a Programmable Superconducting Processor*, Google AI Blog

[260] (2019), *What is phase kickback*, QuTech Academy.

[261] (2021), *IBM Unveils Breakthrough 127-Qubit Quantum Processor*, IBM Newsroom press release.

[262] (2020), *Heisenberg Representation*, Encyclopedia of Mathematics.

[263] (2021), *10 Best Quantum Computing Companies*, Business Magazine.

[264] (2011), *Meet D-Wave*, About D-Wave.

[265] (2022), *The Quantum Daily's guide to qubit implementations*, The Quantum Insider.

[266] (2022), *Transducer*, National Institute of Science and Technology, Physical Measurement Laboratory, https://www.nist.gov/pml/quantum-networks-nist/technologies-quantum-networks/transducers.

[267] (2022), *Quantum memory and Repeaters*, National Institute of Science and Technology, https://www.nist.gov/pml/quantum-networks-nist/technologies-quantum-networks/quantum-memory-and-repeaters.

[268] (2022), *Bringing Quantum Sensors to Fruition*, a report by the subcommittee on Quantum Information Science Committee on Science of the National Science and Technology Council.

[269] *Google Implementation of MNIST*, https://www.tensorflow.org/quantum/tutorials/mnist.

[270] *IBM Implementation of MNIST*, https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html.

[271] F. Laudenbach and T. Isacsson, (2022), *Xanadu's Discussion Forum*, *https* : $//strawberryfields.ai/photonics/demos/tutorial_borealis_beginner.html$.

[272] Sidhartha, (2021), *What is Radix or Base of a Number System?* https://www.vlsifacts.com/what-is-radix-or-base-of-a-number-system/

Appendix A Mathematical Formalism of Google's Implementation of MNIST Classifier

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \; H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

When the composition $H \circ X$ is applied to $|0\rangle$, we get $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

$$\begin{aligned} H \circ X \, |0\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ -1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \end{aligned}$$

For those qubits whose corresponding pixels values of the image matrix is above

the threshold value 0.5, the Pauli$-X$ gate is applied and the state is changed from $|0\rangle$ to $|1\rangle$. Then the quantum state with the ancilla qubit is

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

Thereafter, the circuit has a layer of parametrized XX gates followed by a layer of parametrized ZZ gates for each pair (readout qubit, k^{th} qubit). Suppose the first qubit has its corresponding pixel value not exceeding the threshold, hence in state $\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle$. Application of the XX gate with parameter t_0 is then

$$\begin{aligned}
&(X \otimes X)^{t_0} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle \right) \\
&= \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) & 0 & 0 & -i\sin\left(\frac{\pi t_0}{4}\right) \\ 0 & \cos\left(\frac{\pi t_0}{2}\right) & -i\sin\left(\frac{\pi t_0}{2}\right) & 0 \\ 0 & -i\sin\left(\frac{\pi t_0}{2}\right) & \cos\left(\frac{\pi t_0}{2}\right) & 0 \\ -i\sin\left(\frac{\pi t_0}{2}\right) & 0 & 0 & \cos\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ i\sin\left(\frac{\pi t_0}{2}\right) \\ -\cos\left(\frac{\pi t_0}{2}\right) \\ -i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\ -1 \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ i \sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= I \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) \otimes \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) & i \sin\left(\frac{\pi t_0}{2}\right) \\ i \sin\left(\frac{\pi t_0}{2}\right) & \cos\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= I \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) \otimes \begin{bmatrix} \cos\left(-\frac{\pi t_0}{2}\right) & -i \sin\left(-\frac{\pi t_0}{2}\right) \\ -i \sin\left(-\frac{\pi t_0}{2}\right) & \cos\left(-\frac{\pi t_0}{2}\right) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \left(I \otimes X^{-t_k}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle \right)
\end{aligned}
$$

Notice applying the XX gate parametrized by t_k is equivalent to applying the X gate parametrized by $-t_k$ to the image qubit. By the same token, on the qubits with corresponding pixel values above the threshold, we get

$$
(X \otimes X)^{t_1} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |1\rangle \right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix}
$$

As the circuit repeats the XX operation on each pair of (readout qubit, k^{th} qubit), the state of the readout qubit remains the same, while the the state of the k^{th} qubit is either $\cos\left(\frac{\pi t_0}{2}\right)|0\rangle + i \sin\left(\frac{\pi t_1}{2}\right)|1\rangle$ or $i \sin\left(\frac{\pi t_1}{2}\right)|0\rangle + \cos\left(\frac{\pi t_0}{2}\right)|1\rangle$ depending on its corresponding pixel value. Denote the state of the 0^{th} qubit $|\psi_0\rangle$. Then the parameterized ZZ gate with parameter s_0 is applied.

$$
(Z \otimes Z)^{s_0} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \right)
$$

$$
\begin{aligned}
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi s_0} & 0 & 0 \\ 0 & 0 & e^{i\pi s_0} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ i\sin\left(\frac{\pi t_0}{2}\right) \\ -\cos\left(\frac{\pi t_0}{2}\right) \\ -i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \\ -e^{i\pi s_0}\cos\left(\frac{\pi t_0}{2}\right) \\ -i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \\ -e^{i\pi s_0}\cos\left(\frac{\pi t_0}{2}\right) \\ -e^{i\pi s_0} e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\ -e^{i\pi s_0} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -e^{i\pi s_0} \end{bmatrix} \otimes \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ -e^{i\pi s_0} \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_0} i\sin\left(\frac{\pi t_0}{2}\right) \end{bmatrix}
\end{aligned}
$$

by the property of tensor products. Now the state of the readout qubit is $|0\rangle - e^{i\pi s_0}|1\rangle$.

Then applying the ZZ gate on the (readout qubit, 1^{st} qubit) pair gives us

$$
\begin{aligned}
&(Z \otimes Z)^{s_1} \left(\begin{bmatrix} 1 \\ -e^{i\pi s_0} \end{bmatrix} \otimes \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \right) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi s_1} & 0 & 0 \\ 0 & 0 & e^{i\pi s_1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ \cos\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_0} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_0} \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \\
&= \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} \cos\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} e^{i\pi s_0} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_0} \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \\
&= \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} \cos\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} e^{i\pi s_0} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} e^{i\pi s_1} e^{i\pi s_0} \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \\
&= \begin{bmatrix} 1 \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \\ e^{i\pi s_1} e^{i\pi s_0} \begin{bmatrix} i \sin\left(\frac{\pi t_1}{2}\right) \\ e^{i\pi s_1} \cos\left(\frac{\pi t_1}{2}\right) \end{bmatrix} \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ e^{i\pi s_1} e^{i\pi s_0} \end{bmatrix} \otimes \begin{bmatrix} i \sin\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_1} \cos\left(\frac{\pi t_0}{2}\right) \end{bmatrix}
\end{aligned}
$$

As the circuit continues the ZZ gate iteration on all 16 qubits, the state of the readout

qubit becomes

$$\begin{bmatrix} 1 \\ e^{i\pi s_{15}} e^{i\pi s_{14}} \dots e^{i\pi s_0} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{i\pi s_{15} + i\pi s_{14} + \dots + i\pi s_0} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{i\pi \sum_0^{15} s_k} \end{bmatrix}$$

Then the circuit applies the Hadamard gate on the ancilla qubit while no operation is applied the data qubit. This is expressed as

$$\begin{aligned} H \otimes I &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ 1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & -1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{aligned}$$

Applying it to the previous state gives us the final state before measurement

$$\begin{aligned} &(H \otimes I) \left(\begin{bmatrix} 1 \\ e^{i\pi \sum_0^{15} s_j} \end{bmatrix} \otimes \begin{bmatrix} i \sin\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_{k+1}} \cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix} \right) \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} i \sin\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi s_{k+1}} \cos\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi \sum_0^{15} s_j} i \sin\left(\frac{\pi t_0}{2}\right) \\ e^{i\pi \sum_0^{15} s_j} e^{i\pi s_{k+1}} \cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{2}} \begin{bmatrix} i\sin\left(\frac{\pi t_k}{2}\right) + e^{i\pi\sum_0^{15} s_j} i\sin\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) + e^{i\pi\sum_0^{15} s_j} e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \\ i\sin\left(\frac{\pi t_k}{2}\right) - e^{i\pi\sum_0^{15} s_j} i\sin\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) - e^{i\pi\sum_0^{15} s_j} e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} (1+e^{i\pi\sum_0^{15} s_j}) i\sin\left(\frac{\pi t_k}{2}\right) \\ (1+e^{i\pi\sum_0^{15} s_j}) e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \\ (1-e^{i\pi\sum_0^{15} s_j}) i\sin\left(\frac{\pi t_k}{2}\right) \\ (1-e^{i\pi\sum_0^{15} s_j}) e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1+e^{i\pi\sum_0^{15} s_j} \\ 1-e^{i\pi\sum_0^{15} s_j} \end{bmatrix} \otimes \begin{bmatrix} i\sin\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix} \\
&= \begin{bmatrix} 1+e^{i\pi\sum_0^{15} s_j} \\ 1-e^{i\pi\sum_0^{15} s_j} \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} i\sin\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix}
\end{aligned}
$$

by properties of tensor products.

Then the final state of the readout qubit is $\begin{bmatrix} 1+e^{i\pi\sum_0^{15} s_j} \\ 1-e^{i\pi\sum_0^{15} s_j} \end{bmatrix}$ and of the k^{th} image qubit $\begin{bmatrix} i\sin\left(\frac{\pi t_k}{2}\right) \\ e^{i\pi s_{k+1}}\cos\left(\frac{\pi t_k}{2}\right) \end{bmatrix}$ up to global phase $\frac{1}{\sqrt{2}}$.

The measurement operator outputs the expectation value of the Pauli$-Z$ gate $\langle\psi| Z |\psi\rangle$ where $|\psi\rangle$ is the final state of the readout qubit. Then up to global phase

$$
\begin{aligned}
&\langle\psi| Z |\psi\rangle \\
&= \begin{bmatrix} \left(1+e^{i\pi\sum_0^{15} s_j}\right)^* & \left(1-e^{i\pi\sum_0^{15} s_j}\right)^* \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1+e^{i\pi\sum_0^{15} s_j} \\ 1-e^{i\pi\sum_0^{15} s_j} \end{bmatrix}
\end{aligned}
$$

$$= \begin{bmatrix} \left(1 + e^{i\pi \sum_0^{15} s_j}\right)^* & \left(1 - e^{i\pi \sum_0^{15} s_j}\right)^* \end{bmatrix} \begin{bmatrix} 1 + e^{i\pi \sum_0^{15} s_j} \\ -\left(1 + e^{i\pi \sum_0^{15} s_j}\right) \end{bmatrix}$$

$$= \left(1 + e^{i\pi \sum_0^{15} s_j}\right)^* \left(1 + e^{i\pi \sum_0^{15} s_j}\right) - \left(1 - e^{i\pi \sum_0^{15} s_j}\right)^* \left(1 - e^{i\pi \sum_0^{15} s_j}\right)$$

$$= \|1 + e^{i\pi \sum_0^{15} s_j}\|^2 - \|1 - e^{i\pi \sum_0^{15} s_j}\|^2$$

Appendix B Sycamore

Google's 63−qubit Sycamore, whose architecture is of a lattice shape as shown in the figure. To select $n-$qubits from the chip, it is customary to pick a $log_2(n) \times log_2(n)$ block.

Google's 63 qubit QPU sycamore uses aluminum for metallization and a thin layer of non-superconducting indium for bump-bonds between two silicon wafers. Conducting electrons on the wafers are condensed to macroscopic quantum state, such that currents and voltage behave quantum mechanically. In order to achieve that, the chips are cooled to below 20 mK in a dilution refrigerator.

The architecture of the chip is of lattice structure where each node represents a qubit. It is composed of nonlinear resonators at 5 - 7 Ghz. As controls, a microwave drive is used to excite the qubit and a magnetic flux control to tune the frequency.

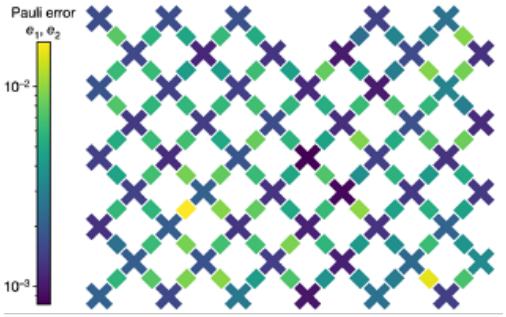

Figure 44: Lattice structure of Sycamore chip [image source]

For readout, a linear resonator connected to each qubit simultaneous readout using frequency- multiplexing technique. The software package offered for quantum circuits is Cirq, based on Python.

Appendix C Borealis [157]

The planar design of $X8$ requires more linear optical components and corresponding control mechanisms for scaling up as more qumodes are added [157]. To overcome these engineering challenges, Xanadu introduced a new QPU, called Borealis, based on time-division multiplexing [63] [271].

In this architecture, temporally spaced batches of light pulses form 216-qumodes. This achieves more number of qumodes for computation with a relatively small number of optical channels.

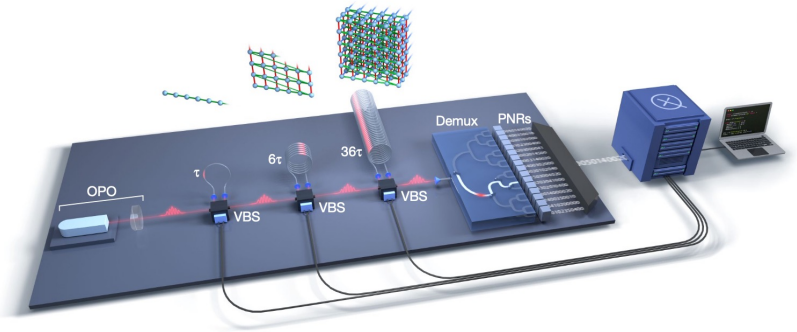

Figure 45: Xanadu's new chip Borealis [157]

From a single source of squeezed light, temporally spaced optical pulses are emitted. Three temporally placed variational beamsplitters (VBS) are placed as quantum operations. The time intervals are 1τ, 6τ, and 36τ. After demultiplexer, the photon number resolutors (PNRs) performs the measurement operation.

[63] also called temporal-domain multiplexing.

Appendix D Quantum Classifier Example Code

```
import tensorflow as tf
import tensorflow_datasets as tfds
from tensorflow import keras
from tensorflow.keras import layers

import pennylane as qml
import numpy as np

import matplotlib.pyplot as plt

######## Loading Data
mnist = keras.datasets.mnist

# datasets are numpy.ndarrays
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()

# Normalize the image data
X_train, X_test = X_train / 255.0, X_test / 255.0

######## Data Preparation
def one_hot(labels):

    depth = 3**3            # 10 classes + 17 zeros for padding
    indices = labels.astype(np.int32)
```

```
    one_hot_labels = np.eye(depth)[indices].astype(np.float32)

    return one_hot_labels

# One-hot encoded labels, each label of length cutoff_dim**2
y_train, y_test = one_hot(Y_train), one_hot(Y_test)

# Using only 600 samples for training in this experiment
n_samples = 600
test_samples = 100
X_train, X_test, y_train, y_test = X_train[:n_samples],
X_test[:test_samples],
y_train[:n_samples],
y_test[:test_samples]

######## Classical Circuit
keras.backend.set_floatx('float32')

model
= keras.models.Sequential([
                        layers.Flatten(input_shape = (28,28)),
                        layers.Dense(128, activation ="elu"),
                        layers.Dense(64, activation ="elu"),
                        layers.Dense(32, activation ="elu"),
                        layers.Dense(22, activation ="elu")
                        ])
```

```
# More than a million parameters for the classical circuit
model.summary()

######## Data Encoding Circuit
def init_layer(x):
    qml.Squeezing(x[0], 0.0, wires=0)
    qml.Squeezing(x[1], 0.0, wires=1)
    qml.Squeezing(x[2], 0.0, wires=2)

    qml.Beamsplitter(x[3], x[4], wires=[0,1])
    qml.Beamsplitter(x[5], x[6], wires=[1,2])

    qml.Rotation(x[7], wires=0)
    qml.Rotation(x[8], wires=1)
    qml.Rotation(x[9], wires=2)

    qml.Displacement(x[10], 0.0, wires=0)
    qml.Displacement(x[11], 0.0, wires=1)
    qml.Displacement(x[12], 0.0, wires=2)

    qml.Kerr(x[13], wires=0)
    qml.Kerr(x[14], wires=1)
    qml.Kerr(x[15], wires=2)

######## Quantum Neural Network Circuit
```

```
def layer(v):
    qml.Beamsplitter(v[0], v[1], wires=[0,1])
    qml.Beamsplitter(v[2], v[3], wires=[1,2])

    qml.Rotation(v[4], wires=0)
    qml.Rotation(v[5], wires=1)
    qml.Rotation(v[6], wires=2)

    qml.Squeezing(v[7], 0.0, wires=0)
    qml.Squeezing(v[8], 0.0, wires=1)
    qml.Squeezing(v[9], 0.0, wires=2)

    qml.Beamsplitter(v[10], v[11], wires=[0,1])
    qml.Beamsplitter(v[12], v[13], wires=[1,2])

    qml.Rotation(v[14], wires=0)
    qml.Rotation(v[15], wires=1)
    qml.Rotation(v[16], wires=2)

    qml.Displacement(v[17], 0.0, wires=0)
    qml.Displacement(v[18], 0.0, wires=1)
    qml.Displacement(v[19], 0.0, wires=2)

    qml.Kerr(v[20], wires=0)
    qml.Kerr(v[21], wires=1)
    qml.Kerr(v[22], wires=2)
```

```
######## Combined Quantum Circuit
num_modes = 3
num_basis = 3

dev = qml.device("strawberryfields.fock",
                 wires=num_modes,
                 cutoff_dim=num_basis)

@qml.qnode(dev, interface="tf")
def quantum_nn(inputs, var):
    init_layer(inputs)   # Encode input x into quantum state

    for v in var:        # Iterative quantum layers
        layer(v)

    return qml.probs(wires=[0, 1, 2])  # Measurement

######## Classical-Quantum Hybrid Circuit
num_layers = 4
num_params = 23
weight_shapes = {'var': (num_layers, num_params)}
qlayer = qml.qnn.KerasLayer(quantum_nn,
                            weight_shapes,
                            output_dim = 4)
```

```
# Add to the classical sequential model
model.add(qlayer)

######## Loss and Optimizer
opt = keras.optimizers.SGD(lr = 0.03)
model.compile(opt,
              loss = 'categorical_crossentropy',
              metrics =['accuracy'])
model.summary()

######## Training
hybrid = model.fit(X_train,
                   y_train,
                   epochs = 100,
                   batch_size = 64,
                   shuffle = True,
                   validation_data = (X_test, y_test))
```