# NORTH SOUTH UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

BACHELOR'S THESIS

**Quantum Implementations of Generative Adversarial Networks (GANs) and a Comparative Analysis with the Classical Counterpart**

**Nazmul Hasan – 1911742042**

**Samya Sunibir Das – 1911563042**

**Sabiha Hossain – 1911017042**

February, 2023

# Quantum Implementations of Generative Adversarial Networks (GANs) and a Comparative Analysis with the Classical Counterpart

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

Bachelor of Science in Electrical and Computer Engineering

Submitted by

**Nazmul Hasan – 1911742042**
**Samya Sunibir Das – 1911563042**
**Sabiha Hossain – 1911017042**

Supervised by

**Dr. Mahdy Rahman Chowdhury**
Associate Professor
Department of Electrical and Computer Engineering

Department of Electrical and Computer Engineering
North South University
Dhaka-1229, Bangladesh

February, 2023

## Declaration

We certify that the work presented in this thesis, entitled "Quantum Implementations of Generative Adversarial Networks (GANs) and a Comparative Analysis with the Classical Counterpart," is a result of our independent research and investigation conducted under the guidance of Dr. Mahdy Rahman Chowdhury, Associate Professor in the Department of ECE at NSU. We affirm that this thesis and any part thereof has not been previously submitted for any academic award, diploma, or other qualifications. Any materials utilized in this project have been properly cited.

**Student's Names and Signatures:**

1. Samya Sunibir Das

2. Nazmul Hasan

3. Sabiha Hossain

Dhaka, Bangladesh

February, 2023

# Approval

The thesis titled "**Quantum Implementations of Generative Adversarial Networks (GANs) and a Comparative Analysis with the Classical Counterpart**", submitted by Samya Sunibir Das, Nazmul Hasan, Sabiha Hossain, in the Fall 2022 session, to the Department of Electrical and Computer Engineering, North South University, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering and approved as to its style and contents on Saturday 4th February, 2023.

**Supervisor's Signature:**

____________________________

**Dr. Mahdy Rahman Chowdhury**

**Associate Professor**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

**Department Chair's Signature:**

____________________________

**Dr. Rajesh Palit**

**Professor and Chair**

Department of Electrical and Computer Engineering

North South University Dhaka, Bangladesh.

# Dedication

We dedicate this work to the teachers, researchers, and our loved ones who have inspired and supported us throughout our academic journey. Their dedication, guidance, and encouragement have been the foundation of our success. This work is a testament to their influence and is a humble tribute to their unwavering support.

## Acknowledgement

We would like to extend our heartfelt gratitude to Dr. Mahdy Rahman Chowdhury, our supervisor, for his invaluable guidance and support throughout the course of this project. His expertise in Quantum Computing and passion for research inspired us to delve into the field of quantum GANs, and we are deeply grateful for the opportunity to work under his supervision.

We would also like to extend our sincere thanks to Mr. Syed Emad Uddin Shubha, Dr. Mahdy Rahman Chowdhury's research assistant, who taught us the basics and the advanced aspects of quantum computing and provided invaluable guidance and support throughout the project.

Furthermore, we would like to acknowledge the contribution of our teachers and instructors, who have imparted knowledge, skills, and values that have prepared us for this moment. Their dedication and passion for teaching have been instrumental in our academic and personal growth.

Finally, we would also like to extend our thanks to our families and friends, who have supported us throughout the years and helped us to reach this stage. Their encouragement and belief in us have been a source of strength and inspiration.

We are deeply grateful for their guidance and support, and their contributions will be remembered in this work forever.

Dhaka, Bangladesh

February, 2023

# TABLE OF CONTENTS

# TABLE OF FIGURES

## ABSTRACT

This thesis investigates the feasibility of using quantum computing to improve the performance of Generative Adversarial Networks (GANs). Two quantum GAN models, the Quantum Patch GAN and the QuGAN are compared to a classical GAN on the MNIST dataset, a widely used benchmark for machine learning algorithms. The classical GAN generated high-quality synthetic images with converging losses. The Quantum Patch GAN, however, showed a decline in performance after 700 epochs, but generated 0s from the set of 0s that the model was provided as input. The quantum state-based QuGAN performed best at the 50th epoch, generating decent-quality 9s and mildly recognizable 3s and 6s. The results of this research demonstrate the potential of quantum computing for generative modeling and offer new perspectives on the connection between quantum and classical computing in deep learning. Further exploration is needed to fully harness the advantages of quantum computing for GANs and advance the field of quantum computing itself. The potential impact of quantum GANs on real-world applications, such as drug discovery and materials science, cannot be overstated.

# CHAPTER 1: INTRODUCTION

Generative Adversarial Networks (GANs) are a deep learning architecture that has revolutionized the field of computer vision and have been widely used for tasks such as image synthesis, super-resolution, and style transfer [1]. With the advent of quantum computing, the exploration of quantum GANs has become a rapidly growing area of research, with the goal of harnessing the power of quantum computers to enhance the performance of GANs. In this research, we focus on two such quantum GAN models, the Quantum Patch GAN [2] and the QuGAN Quantum State GAN, [3] and compare their performance with that of a classical GAN on the MNIST dataset [4].

The MNIST dataset is a widely used benchmark for evaluating the performance of machine learning algorithms, and consists of grayscale images of handwritten digits. GANs are trained by minimizing the difference between the generated and real images, and the goal is to produce images that are as close to the real images as possible. Classical GANs use classical computer hardware to perform their computations, while quantum GANs leverage the unique features of quantum computers, such as superposition and entanglement, to perform the computations.

The Quantum Patch GAN is a modification of the Patch GAN architecture, which uses convolutional neural networks to generate images. In the Quantum Patch GAN, the convolutional layers are replaced by quantum circuit layers that perform quantum convolutions. This allows the model to learn quantum representations of the images, which can be more expressive than classical representations [2].

The QuGAN Quantum State GAN, on the other hand, represents the images as quantum states, rather than classical arrays of pixel values. The generator network in the QuGAN is implemented as a quantum circuit, while the discriminator network is implemented as a classical neural network. The goal of the QuGAN is to learn the quantum representations of the images that are closest to the real images and then use these representations to generate new images [3].

In conclusion, our research demonstrates the potential of quantum GANs for enhancing the performance of GANs. While the quantum GAN models we evaluated showed promising results, further research is needed to fully exploit the advantages of quantum computing for GANs. This research provides a starting point for exploring the use of quantum computing for generative modeling, and offers new insights into the relationship between quantum and classical computing in the context of deep learning [5].

Such research is necessary because quantum computing has the potential to dramatically enhance the performance of GANs. The unique features of quantum computers, such as superposition and entanglement, can provide new avenues for representation learning, allowing GANs to generate higher quality and more diverse images [6].

Furthermore, the development of quantum GANs will also help to advance the field of quantum computing itself, by providing new testbeds for exploring the potential of quantum algorithms and hardware. In addition, the application of quantum GANs to real-world problems, such as drug discovery and materials science, could have a significant impact on these fields, by providing new tools for simulation and modeling.

## 1.1 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GANs) are a deep learning architecture that was introduced in 2014 by Ian Goodfellow and colleagues. The main idea behind GANs is to generate new data that resembles the original data by training two deep neural networks in a competition, hence the term "adversarial" [7]. GANs consist of two components: the generator and the discriminator. The generator is responsible for generating new data, while the discriminator is responsible for evaluating the realism of the generated data. Both networks are trained simultaneously in an adversarial manner, where the generator tries to generate data that is indistinguishable from the original data, and the discriminator tries to accurately classify the data as real or generated. Over time, both networks improve, with the generator becoming better at generating realistic data, and the discriminator becoming better at identifying whether the data is real or fake.

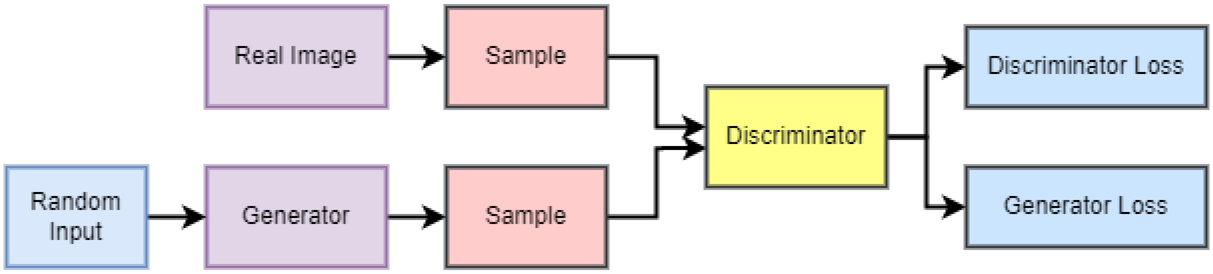


Figure 1.1: Typical architecture of a GAN

### 1.1.1 Understanding Generators and Discriminators

Let's consider this famous analogy for GANs [8] where the generator is like a thief, who creates fake money in an attempt to deceive the discriminator, which is like a police officer. The police officer's job is to determine if the money is real or fake. The thief's goal is to create money that is so convincing, the police cannot tell the difference between real and fake money. The police officer, in turn, tries to get better at identifying fake money, by learning from its mistakes. This creates an ongoing improvement cycle for the thief and the police, leading to better and more sophisticated fake money and a better-trained police officer. Ultimately, the generator learns to create fake money that is virtually indistinguishable from the real thing, while the discriminator becomes an expert at identifying fake money. The given figure below depicts this scenario accurately.

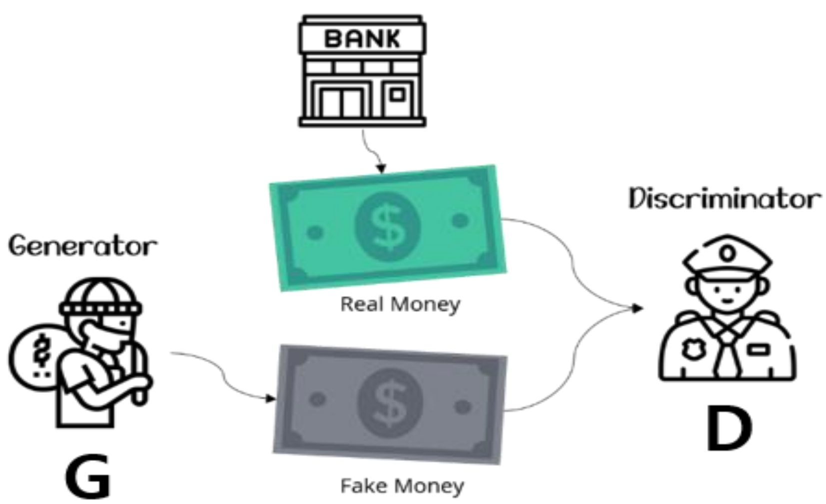


Figure 1.2: Generator's and Discriminator's analogical nature

### 1.1.2 Applications of GANs

GANs are widely used in various applications, particularly in computer vision. They can be used for various tasks, such as image generation, style transfer, and super-resolution. For example, a GAN can be trained on a dataset of faces and then used to generate new, realistic faces. GANs have also been applied to other domains, such as generating music, text, and video. They have been widely used in computer vision and other domains, and they are a powerful model for generating realistic data. These are used in the medical field as well, to generate accurate synthetic physiology-related images.

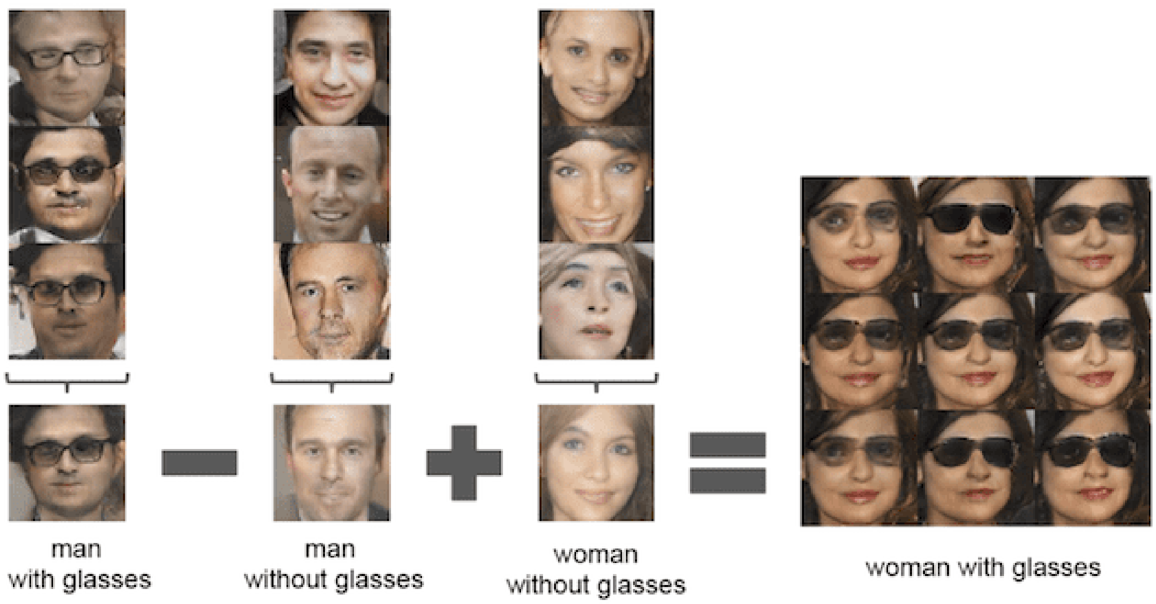


Figure 1.3: GAN model generating synthetic images from training data

### 1.1.3 The versatility of GANs - Encompassing Various Types of Learning

Generative Adversarial Networks (GANs) are a versatile deep learning architecture that can be used for a variety of tasks, including unsupervised, semi-supervised, fully supervised, and reinforcement learning. The reason why GANs can be used for these different tasks is because of the structure and training process of the network.

**Unsupervised Learning:** In unsupervised learning, the goal is to learn patterns in the data without any labeled examples. With GANs, the generator can be trained on an unlabeled dataset to generate new data that resembles the original data [9]. The discriminator is used to evaluate the realism of the generated data, but it does not have any labeled examples to learn from.

**Semi-Supervised Learning:** In semi-supervised learning, a small portion of the data is labeled, and the goal is to use the labeled data to learn patterns in the rest of the data. GANs can be used for semi-supervised learning by incorporating the labeled data into the training process. For example, the generator can be trained on the labeled data to generate new examples that resemble the labeled examples, while the discriminator is trained on both labeled and unlabeled data to distinguish between real and generated examples [10].

**Fully Supervised Learning**: In fully supervised learning, all of the data is labeled, and the goal is to learn a mapping from input to output based on the labeled examples. GANs can be used for fully supervised learning by treating the generator as the input-to-output mapping, and the discriminator as the evaluator of the mapping's accuracy. In this setup, the generator and the discriminator are trained on the labeled examples.

**Reinforcement Learning:** In reinforcement learning, the goal is to learn a policy for taking actions in an environment that maximizes a reward signal. GANs can be used for reinforcement learning by treating the generator as the policy and the discriminator as the reward signal. In this setup, the generator generates actions, and the discriminator evaluates the actions and provides feedback to the generator.

GANs can be used for various learning tasks because of their flexible architecture and adversarial training process. The choice of which type of learning to use with GANs will depend on the specific task and the available data, and these models have vast potential that can be unlocked with the help of Quantum Technology.

## 1.2 QUANTUM COMPUTING

Quantum computing is a branch of computer science that utilizes the principles of quantum theory, which explains the behavior of matter and energy at the atomic and subatomic levels [11]. It works by utilizing subatomic particles, such as electrons or photons, and encoding information in quantum bits, or qubits. Unlike classical bits, qubits can exist in multiple states simultaneously. This allows quantum computers to perform calculations that would take millions of years on classical computers due to their binary processing limitations. The potential of quantum computing lies in its ability to use the interference between wave-like quantum states of linked qubits to perform calculations much faster than classical computers. Quantum Computing is built upon superposition, entanglement, measurement etc., which will be later discussed in Chapter 3.

### 1.2.1 Quantum Computing Vs Classical Computing

Quantum computers have a simpler structure than classical computers, with no memory or processor, consisting only of superconducting qubits. Quantum computers process information differently, using qubits to run multidimensional quantum algorithms and having an exponential increase in processing power with added qubits. Classical computers use bits to operate programs and have a linear increase in power with added bits but less computing power.

Classical computers are suitable for everyday tasks and have low error rates, while quantum computers are ideal for higher-level tasks such as simulations, data analysis, and energy-efficient battery creation, but can also have high error rates.

Classical computers do not require special protection, with a basic internal fan to prevent overheating, while quantum processors need to be protected from even slight vibrations and kept extremely cold with the use of super-cooled superfluid. Quantum computers are more expensive and challenging to build than classical computers [12].

## 1.3 GANs boosted by the power of Quantum, Quantum GANs

Quantum Generative Adversarial Networks (Quantum GANs) are a hybrid of classical deep learning and quantum computing. They aim to combine the strengths of both classical deep learning and quantum computing to improve the performance of generative models.

In a Quantum GAN, some or all of the components of the classical GAN are replaced with quantum counterparts. For example, the generator network could be implemented as a quantum circuit, while the discriminator network is still a classical neural network. The quantum generator can use quantum states and quantum operations to generate data, while the discriminator network evaluates the realism of the generated data in the same way as a classical GAN [2].

## 1.4 Background and Motivation for the Research

The potential of Quantum GANs is tied to the advancement of quantum computing. As quantum computing technology improves, the quantum components of the Quantum GAN could become more powerful, allowing for the generation of more complex and realistic data. Additionally, the quantum components could be used to improve the training process of the GAN, making it more efficient and effective, and might lead to new breakthroughs in generative models.

Quantum GANs have not yet been widely researched, and there is still much to be discovered about their capabilities and limitations. However, the potential for combining classical deep learning and quantum computing in this way is exciting, and could lead to new breakthroughs in generative models.

In light of these developments, this research aims to explore the potential of quantum computing in the field of GANs. The focus of the research is to implement and compare quantum patch GAN and QGAN models with classical GANs in the MNIST dataset. The findings of this research will shed light on the capabilities and limitations of quantum generative models and provide valuable insights for future developments in the field.

## 1.5 Thesis Structure and Overview

Our senior thesis, this work based on comparing the two quantum GAN models with the classical GAN, has been divided into six different chapters, with each chapter discussing the appropriate topic.

Firstly, the thesis begins with the first chapter, where a general introduction to this work has been illustrated. We begin by trying to give the readers a vivid description and necessary background about GANs in general. Then, we try to give a wider perception regarding this topic by providing even more details regarding the applications and the versatility of GANs. Then, we enlighten the readers regarding quantum computing and establish how quantum computing and GANs are tied together and these models are possible to implement on quantum systems as well.

Then, the second chapter discusses, and reviews the other quantum GAN-related works that are being conducted in contemporary times.

Next, the third chapter focuses on the technical aspects related to the required quantum theories for this research. It especially discusses all the intricate theories regarding the two quantum GAN models.

The fourth chapter will discuss the research methodology in vivid detail. It will help the reader to have an even deeper appreciation for the work, effort, and dedication that went into conducting this research.

Chapter five will be the point where it all boils down to as it will provide the results achieved out of this research work. Finally, Chapter six will give the readers a conclusion to this work, and will wrap everything up.

# CHAPTER 2: RELATED WORK

Extensive research works on Quantum GANs are being conducted in contemporary times. This section of the thesis discusses the numerous other research works that have been done so far on this topic.

Haozhen Situ et al. [13] investigate the use of generative adversarial networks (GANs) in quantum computing. They propose a quantum GAN for generating classical discrete data, which has a classical-quantum hybrid architecture. The generator consists of a parameterized quantum circuit made up of simple one-qubit rotation gates and two-qubit controlled-phase gates, while the discriminator is a classical neural network. The authors highlight the potential advantages of their proposed model, including its ability to generate discrete data, avoid the input/output limitations faced by other quantum learning algorithms, and the ability to load the probability distribution of the data into a quantum state for further applications.

Jinfeng Zeng and colleagues, approach the challenge of training quantum circuits as probabilistic generative models for classical data by combining a quantum circuit generator and a classical neural network discriminator. The authors' adversarial quantum-classical hybrid training scheme allows for the efficient inference of missing data with a quadratic speed-up. The approach's effectiveness is demonstrated through numerical simulation on the Bars-and-Stripes dataset. This fresh approach to machine learning with quantum circuits has the potential to yield practical advantages on near-term quantum devices [14].

W. Liu et al., aims to improve the quantum generative adversarial network (QGAN) algorithm to better conform to the principles of human-centered computing (HCC) [15]. They acknowledge that the generation process of QGAN is random and does not align with the human-centered concept, making it unsuitable for real-world scenarios. To address this issue, the authors propose a hybrid quantum-classical conditional generative adversarial network (QCGAN) algorithm that incorporates artificial conditional information in both the generator and discriminator. The generator uses a parameterized quantum circuit with an all-to-all connected topology to facilitate network parameter tuning during training, while the discriminator uses a classical neural network to avoid the input limitations of quantum machine learning. The algorithm was tested using the Bars-and-Stripes training set on a quantum cloud computing platform and was found to effectively converge to the Nash equilibrium point and perform human-centered classification generation tasks.

The authors, led by Wanghao Ren [16], have introduced a quantum generative adversarial framework for processing images. They have extended the use of quantum generative adversarial networks to the field of quantum image processing, demonstrating how classical images can be loaded and learned using quantum circuits. The framework uses a reduced number of basic quantum building blocks and can generate pure states even in the presence of various types of noise. The authors have numerically simulated the loading and learning of classical images on two image databases, MINST and CIFAR-10, and showed that the method can still quickly converge under the influence of noise. The framework is expected to be useful in the field of quantum image processing as a subroutine of other quantum circuits.

Christa Zoufal and the co-authors have developed a hybrid quantum-classical algorithm for loading classical data into quantum states efficiently and approximately. This is achieved through the use of quantum Generative Adversarial Networks (qGANs), which facilitate the learning and loading of generic probability distributions given by data samples into quantum states. The loading process requires only a small number of gates and thus enables the use of quantum algorithms that could potentially have advantages over classical algorithms. The method has been implemented with Qiskit and tested using quantum simulation and actual quantum processors provided by IBM Q Experience. The authors have also demonstrated the use of the trained quantum channel in a quantum finance application through simulation [17].

A recent research work published by Murphy Yuezhen Niu and colleagues [18] proposed a new type of architecture for quantum generative adversarial networks (EQ-GAN) that overcomes the limitations of previous models. The EQ-GAN uses entangling operations between the generator output and true quantum data to converge to the Nash equilibrium. This is the first multi-qubit demonstration of a fully quantum GAN with a proven optimal

Nash equilibrium and was tested on a Google Sycamore superconducting quantum processor. The authors also show that the EQ-GAN can be used to prepare an approximate quantum random access memory and to train quantum neural networks via variational datasets.

Researchers led by S. Chakrabarti [19] proposed a new design for quantum Wasserstein Generative Adversarial Networks (WGANs), which has been shown to improve the robustness and scalability of adversarial training of quantum generative models on noisy quantum hardware. They defined a Wasserstein semi-metric between quantum data, which has key theoretical merits, and demonstrated how to turn this semi-metric into a concrete design for quantum WGANs that can be efficiently implemented on quantum machines. Numerical simulations show that their quantum WGANs outperform other quantum GAN proposals regarding robustness and scalability. They also used their quantum WGAN to generate a 3-qubit quantum circuit which approximates a 1-d Hamiltonian simulation circuit with fewer gates than standard techniques.

A study conducted by C. Bravo-Prieto and colleagues, based on the development of Style q-GAN, [20] proposed and evaluated a new architecture for a quantum generator in the context of generative adversarial learning for simulating particle physics processes at the Large Hadron Collider (LHC). The methodology is validated on artificial data generated from known distributions and on datasets of specific LHC scattering processes. The results show that the new architecture improves performance compared to the state-of-the-art, with smaller Kullback-Leibler divergences and the ability to learn the underlying distribution functions with smaller training sample sets. The methodology is tested on two different quantum hardware architectures, trapped-ion, and superconducting technologies, to demonstrate its hardware independence.

For the paper titled, "Hybrid Quantum-Classical Generative Adversarial Network for High-Resolution Image Generation" - here, the paper's authors propose a new hybrid quantum-classical generative adversarial network framework. This new approach integrates classical and quantum techniques and is demonstrated to generate 28x28 pixels gray-scale images from the MNIST dataset with high accuracy. The study shows that the hybrid approach has superior learning capabilities and achieves comparable results to classical frameworks, with significantly fewer trainable generator parameters. The authors also explore the impact of various parameters on the network's performance and find that increasing the size of the quantum generator generally improves its learning capability. The results provide a foundation for future designs of QGANs for complex image generation tasks [21].

The authors addressed the challenge of loading data efficiently from classical memories to quantum computers by using quantum generative adversarial networks (qGANs) [22]. By tuning the hyper-parameters and the optimizer, they reduced the Kolmogorov-Smirnov statistic by 43-64% compared to the state of the art. However, the accuracy of the training depends on the starting point of the search algorithm and the optimizer used, and the SPSA optimizer does not achieve the same accuracy as the Adam optimizer. This calls for new advancements to support the scaling of qGANs.

# CHAPTER 3: THEORY

This chapter covers the theory required for the understanding of the entire system. It is divided into three parts.

- Fundamentals of Quantum Computing
- Fundamentals of Quantum Machine Learning
- Deep Quantum Learning

## 3.1 FUNDAMENTALS OF QUANTUM COMPUTING

Quantum computing focuses on the principles of quantum theory, which deals with modern physics that explain the behavior of matter and energy of an atomic and subatomic level. Quantum computers use qubits, as opposed to classical computers, which store information as bits with either 0s or 1s. Qubits store information in a multidimensional quantum state that interacts with 0 and 1 [23].

Before diving into the fundamental ideas of quantum computing, we need to have a clear understanding of Bra-Ket nation. Bra $\langle\Phi|$ and Ket $|\psi\rangle$ represents row and column vectors respectively. For a vector in three dimensional complex vector space, the column vector,

$$A = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{can be written in Ket notation, } |\mathrm{A}\rangle = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix}$$

On the other hand, a Bra vector can be obtained by first transposing the Ket vector and then complex-conjugating it. So, the bra vector can be written as:

$$\langle A| = [\, x^*, y^*, z^*]$$

### 3.1.1 Qubit

Information is represented in bits in traditional computing, where each bit might have a value of zero or one. The two basic states of a qubit, which is a two-level quantum system, are typically expressed as 0 and 1, respectively. In contrast to a traditional bit, a qubit can be in a linear combination of both states or in the states 0 and 1. Superposition is the term for this phenomena. A quantum computer can make use of a wide range of fundamental particles, including electrons and photons. In actuality, ions succeed by charging or separating, which represents the numbers 0 and/or 1. Each of these particles has its own measurement, known as a qubit [24]. Quantum computing is based on the behavior and characteristics of these particles. A variety of elementary particles, including electrons and photons (but ions have also been successful in practice), can be employed in a quantum computer, with either their charge or polarization serving as a representation of 0 and/or 1.

Examples include the rotation of an electron, where two states can be thought of as spin up and spin down, or the polarization of a single photon, where two states can be thought of as direct polarization and horizontal polarization. A linear combination of $|0\rangle$ and $|1\rangle$ can be used to represent each qubit in the following way:

$$|\,\psi\rangle \;=\; \alpha\,|\,0\rangle \;+\; \beta\,|\,1\rangle$$

where $\alpha$ and $\beta$ are the complex probability amplitude:

$$|\alpha|^2 + |\beta|^2 = 1.$$

The probability amplitudes, and, encode more information than just the likelihoods of measurement results; for instance, the relative phase between and is what causes quantum interference, as demonstrated by the two-slit experiment [25].

### 3.1.2 Quantum Register

In a quantum computer, the qubit register serves as a conceptual grouping for the qubits. Within this register, each qubit has an index, which counts up from index 0 and up by 1. Therefore, a system with four qubits, for instance, has a qubit register with a width of four and indexes of 0, 1, 2, and 3. The qubit index is used in qubit operations to address each qubit. The probability amplitude for each state is used to characterize the entire quantum state, which is kept in memory [26].

Let $\alpha_i$ be the computational basis state probability amplitude, four complex parameters can be used to describe a two-qubit system:

$$|\Psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$$

and eight complex parameters are used to describe a three qubit system:

$$|\Psi\rangle = \alpha_0|000\rangle + \alpha_1|001\rangle + \cdots + \alpha_7|111\rangle$$

### 3.1.3 Quantum Entanglement

Another inexplicable phenomenon in quantum physics is entanglement. A pair or set of particles is said to be entangled when one particle's quantum state cannot be described without affecting the quantum state of the other. The quantum state of the system as a whole can be expressed even while its component pieces are not in a clearly defined state. When two systems are in the quantum entangled state, no matter how far apart they are, learning something about one system immediately reveals something about the other. When two qubits are entangled, they have a unique connection. The measurements' findings will show where the entanglement is. The various qubits' measurements could produce a 0 or a 1. A qubit's measurement result will, however, always be correlated with another qubit's measurement result [27].

Qubits may instantly interact with each other while being separated by enormous distances due to quantum entanglement. The associated particles will remain entangled no matter how far apart they are as long as they are separate.

### 3.1.4 Quantum Superposition

Superposition is one of the fundamental ideas of quantum mechanics. A wave that describes a musical tone can be conceptualized in classical physics as a combination of superposed waves of various frequencies. The same concept applies to a quantum state in superposition, which can be thought of as a linear combination of other unique quantum states. A new, legitimate quantum state is created by this superposition of quantum states.

Qubits are capable of being in a superposition of the basic states of $|0\rangle$ and $|1\rangle$. A qubit will collapse to one of its eigenstates when measured and the measured value will represent that state. For instance, a measurement will cause a qubit to collapse to one of its two base states $|0\rangle$ and $|1\rangle$ with an equal probability of 50% while it is in a superposition state of equal weights. |0⟩ is the state that, when measured and consequently collapsed, always yields the value 0. Similar to this, | 1 ⟩ will always get converted to 1 [28].

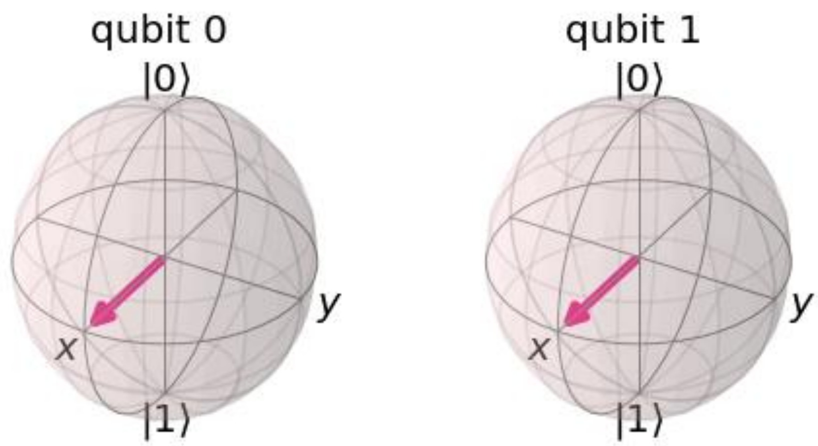


Figure 3.1: Bloch Sphere representing superposition state

The particle then moves into a superposition of states, where it acts as though it is simultaneously in both states, in accordance with quantum theory. The total number of calculations a quantum computer might perform is therefore $2^n$, where n is the quantity of qubits employed.

### 3.1.5 Quantum Measurement

Any type of interaction with other particles that provides those particles with knowledge about the position of the first particle is referred to as "measurement" in quantum mechanics. Once a quantum computation task is finished, a qubit may be measured out in the computational basis. As a result, the qubit state reaches one of the fundamental states |0⟩ or |1⟩. The value obtained could be retained and utilized for further processing in a typical register. According to the laws of quantum computing, all qubits must be initially in the state |0⟩ and can only be measured in the computational basis [29].

Measuring a qubit with vector representation of $\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ will produce a result |0⟩ with a probability of $|\alpha|^2$ and |1⟩ with a probability of $|\beta|^2$.

## 3.1.6 Quantum Logic Gates

Qubits used by quantum computers can exist in a "superposition" of 0 and 1, in contrast to conventional bits, which can only be 0 or 1. Quantum computers are incredibly powerful due to their capacity to live in several states simultaneously. Quantum logic gates are a set of fundamental processes used to accomplish quantum calculations.

Quantum gates come in a wide variety of varieties. Single-qubit gates exist that enable both the creation of superposition states and the flipping of a qubit from 0 to 1 [30]. Some of the single qubit gates used in our system are briefly introduced below:

### 3.1.6.1 Hadamard Gate

It maps the given computational basis state and creates a corresponding superposition state.

$$|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}} \text{ and } |1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}; matrix\ represntation\text{:}\ \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

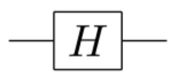


Figure 3.2: Hadamard Gate

### 3.1.6.2 Pauli-X Gate

Also known as bit-flip gate, performs a single qubit rotation of $\pi$ radian around the x axis. Matrix representation: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|$

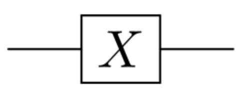


Figure 3.3: Pauli-X Gate

### 3.1.6.3 Pauli-Y & Z Gates

Pauli Y & Z-gates function just like the X-gate does in quantum circuits. They also perform single-qubit rotation of $\pi$ radian around the y and z axis respectively. Matrix representation:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0|$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$$

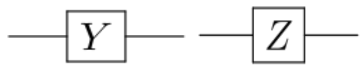


Figure 3.4: Pauli Y&Z Gates

Two-qubit gates are a further option. These enable interactions between the qubits and can be utilized to produce quantum entanglement [31]. Some of the single qubit gates used in our system are briefly discussed below:

### 3.1.6.4 CNOT Gate

One or more qubits serve as a control for an operation when controlled gates operate on two or more qubits. One such example is the controlled NOT gate, which only executes the NOT operation on the second qubit when the first qubit is $|1\rangle$, otherwise leaves the qubit as it is.

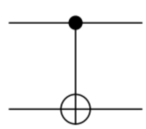

Figure 3.5: CNOT Gate

### 3.1.6.5 Swap Gate

Swaps two qubit which can be decomposed into the form:

$$Swap = \frac{I \otimes I + X \otimes X + Y \otimes Y + Z \otimes Z}{2}$$

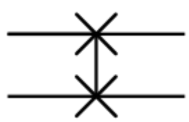

Figure 3.6: Swap Gate

Similar to conventional computers, it is possible to decompose quantum algorithms with numerous qubits into a series of single- and two-qubit quantum gates, establishing an all-encompassing quantum gate set.

### 3.1.7 Bloch Sphere

In quantum computing, a qubit can represent any linear combination as well as the bit values 0 or 1 in the state of superposition. This implies that a state could consist of 50% of a "zero" state and 50% of a "one" state [32]. A vector which is displayed on the Bloch sphere's surface can be used to represent the state of a qubit. The Bloch Sphere, also known as the unit of a quantum state, is the mathematical or geometrical representation of the state of a qubit. It uses a two-dimensional vector with a typical length of one to represent the state of a qubit. Two elements make up this vector: a real number α and a complex number β. The orthonormal computational basis states $|0\rangle$ and $|1\rangle$ are used to determine the north pole and south pole of the Bloch sphere, respectively [33].

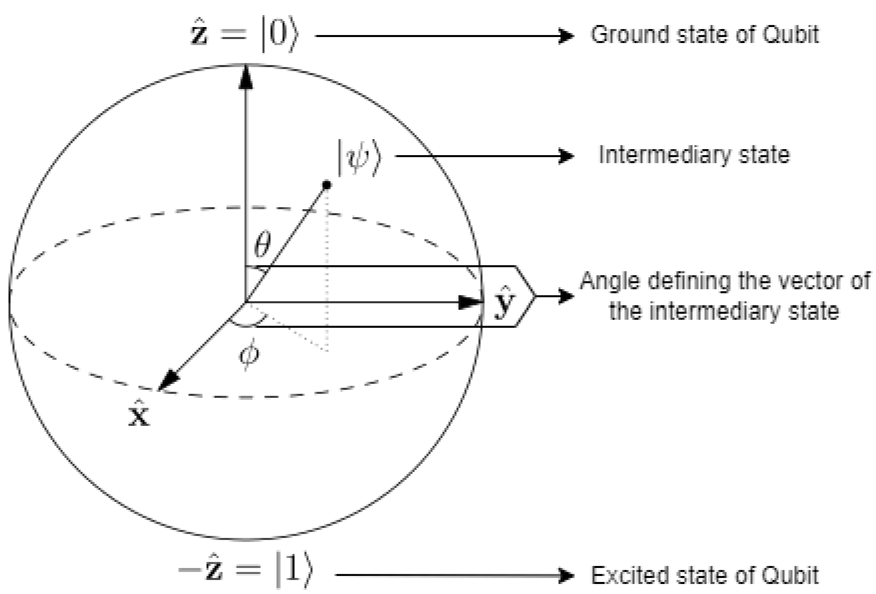


Figure 3.7: The Bloch Sphere

A qubit in superposition of two sates can be denoted as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Where, $|\psi\rangle$ is the state of the qubit, α and β are the probability of the 0 and 1 state respectively.

## 3.2 FUNDAMENTALS OF QUANTUM MACHINE LEARNING

### 3.2.1 Quantum Embedding

One of the main problems of quantum machine learning is transforming real world classical data into quantum state. Quantum embedding encodes classical data as quantum states in a Hilbert space with the use of quantum feature map. It creates a quantum state $|\psi_x\rangle$ *by converting a classical data point x into a set of gate parameters in a quantum circuit.*

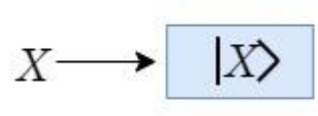


Figure 3.8: Encoding classical data into quantum state

For a classical input data of *M* samples, each having *N* features, if we consider $x^{(m)}$ as a *N*-dimensional vector

$$D = \{\, x^{(1)}, \ldots, x^{(m)}, \ldots, x^{(M)} \,\}$$

There are different embedding techniques to use to embed this data into n qubits. Amplitude embedding, Basis embedding, and Angle embedding are three of those that are briefly covered in this chapter.

#### 3.2.1.1 Amplitude Embedding

Amplitude embedding, also known as wave function embedding converts the data point into quantum state amplitudes. Considering a dataset *x* with *n* number of dimensions; $x = \{x_1, x_2, \ldots, x_n\}$, the dataset is initially normalized to length 1 ($|x|^2 = 1$) [34]. The sum of the number of dimensions and the number of samples yields the number of amplitudes to be encoded. The amplitude of *n* qubit quantum state is denoted as:

$$|\psi_x\rangle = \sum_{i=0}^{n} x_i|i\rangle$$

Here, $x_i$ is the i[th] element of $x$ and $|i\rangle$ is the computational basis state. Let's consider a data point $x$ with four dimensions, $x = \{x_1=1.5,\ x_2=1.2,\ x_3=1.3,\ x_4=1.1\}$. After normalizing each element, for instance $\frac{1}{\sqrt{10.3}}(1.5, 1.2, 1.3, 1.1)$.

The final form: $\frac{1.5}{\sqrt{10.3}}|00\rangle + \frac{1.2}{\sqrt{10.3}}|00\rangle + \frac{1.3}{\sqrt{10.3}}|00\rangle + \frac{1.1}{\sqrt{10.3}}|00\rangle$

### 3.2.1.2 Basis Embedding

In order to use basis embedding, the data must be in the form of binary string. The usage of a computational basis forms the foundation of the basis embedding idea. In this embedding technique, a scaler value is embedded to its binary form and then transformed to a quantum state [34]. A number is approximated by a binary bit string in the first stage of the process, and then it is encoded using a computational basis state in the second.

For a traditional dataset D mentioned previously, basis embedding requires each sample has to be a $N$-bit binary string $x^{(m)}$ can be directly mapped to the quantum state $|x^{(m)}\rangle$. *The equation below represents a dataset that has been transformed to a basis embedding:*

$$|D\rangle = \frac{1}{\sqrt{M}}\sum_{i=0}^{M}\left|x^{(m)}\right\rangle$$

where m is the n-dimensional vector and M is the total number of samples. There must be at least as many quantum subsystems as there are bits in an N-bit binary string.

### 3.2.1.3 Angle Embedding

Angle encoding is a straightforward and efficient way for embedding data. The values that need to be encoded are applied rotations on the *x*-axis or *y*-axis using quantum gates to produce the angle embedding. The number of rotations required to apply angle embedding to a dataset will be equal to the number of features in the dataset. It requires n qubits to create the collection of quantum states in the n-dimensional sample. The angles of ration gates are determined by the classical information for a classical data x:

$$|x\rangle = \otimes_i^n \ R(x_i)\ |0^n\rangle$$

Where $R$ can be $R_x$, $R_y$, $R_z$ and the number of is as same as the dimension of vector $x$ [35].

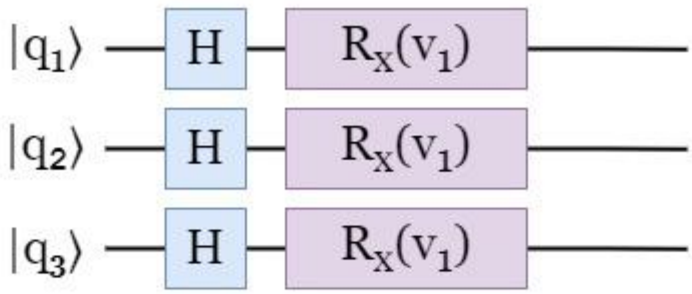


Figure 3.9: Angle Embedding circuit

#### 3.2.1.4 Product Embedding

Another way is to utilize a (tensor) product encoding, in which the amplitudes of each qubit are used to represent a different aspect of the input $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$. For an example of encoding $x_i$ as $|\varphi(x_i)\rangle = \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$ *for* $i = 1, \ldots, N$ [36]. This is a feature-embedding circuit that has the following effect

$$U_\varphi : x \in \mathbb{R}^N \to \begin{pmatrix} \cos x_1 \\ \sin x_1 \end{pmatrix} \otimes \ldots \otimes \begin{pmatrix} \cos x_N \\ \sin x_N \end{pmatrix} \in \mathbb{R}^{2N}$$

### 3.2.2 Quantum Feature Map

In machine learning, we often have a dataset of inputs $D = \{ x^{(1)}, \ldots, x^{(M)} \}$ from a specific input set $X$ and thus must recognize patterns to evaluate or synthesize previously unobserved data. Kernel approaches build models that capture the characteristics of a data distribution by measuring the distance $(x, x')$ between any two inputs $x, x' \in X$. This measurement of distance is linked to internal products in a specific area, the feature space. These techniques have a strong theoretical base [37], from which we intend to draw some important conclusions, in addition to several practical implementations, the most well-known of which is the support vector machine.

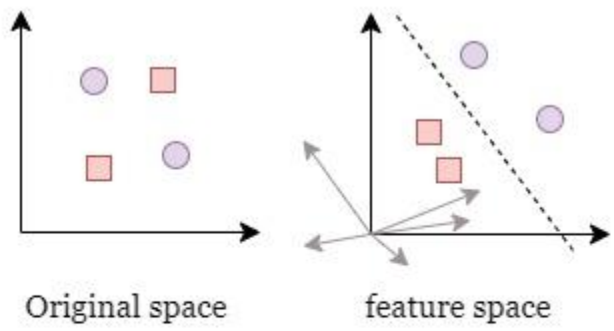


Figure 3.10: Transforming data using a feature map

Data from the two classes "squares" and "circles" are not distinguishable by a simple linear model in the original space of training inputs, but we can map them to a higher dimensional feature space, where it is possible to build a separating hyperplane that serves as a decision boundary using only a linear model (right).

Let X be an input set, $F$ be a Hilbert space, sometimes known as the feature space, and $x$ be a sample from the input set $X$. A feature map is a map from inputs to vectors in the Hilbert space with the formula $\varphi: X \rightarrow F$. The vectors $\varphi(x) \in F$ is known as Feature vectors [38]. As they translate any kind of input data into a space with a clear metric, feature maps are crucial to machine learning. A dataset can become much easier to categorize in feature space if the feature map is a nonlinear function because it alters the relative positions of the data points. Feature maps and kernels have a close relationship.

### 3.2.3 Quantum Speedup

The quantum computer can process information in a way that is not possible for classical computers due to quantum coherence. A machine learning model uses optimization to enhance the learning process in order to produce the most sufficient and accurate estimations. Loss function minimization is the primary goal of optimization. An increased loss function will result in more inaccurate and unreliable outputs, which can be expensive and result in inaccurate estimates. Iterative performance optimization is required for the majority of machine learning techniques. A quantum algorithm uses a step-by-step process to address issues, such as database searches. It can perform better than the most popular classical algorithms. The Quantum Speedup is the name given to this occurrence. Quantum algorithms for optimization point to improvements in machine learning optimization problems. The ability to make multiple copies of the current solution, encoded in a quantum state, is a result of the quantum entanglement feature. Each stage of the machine learning algorithm makes use of them to enhance that solution.

## 3.3 Deep Quantum Learning

Combining deep learning and quantum computing can speed up the training process of neural network. By using this technique, we may accomplish underlying optimization and create a new framework for deep learning. On a genuine, physical quantum computer, we can reproduce classical deep learning methods. As more neurons are added, the computational complexity rises when multi-layer perceptron topologies are used. Performance can be enhanced by using specialized GPU clusters, which also considerably cuts down on training time. Even this, though, will rise in comparison to quantum computers.

The hardware in quantum computers is built to imitate neural networks rather than the software found in traditional computers. Here, a qubit takes on the role of a neuron, the fundamental building block of a neural network. As a result, a quantum system with qubits can perform the function of a neural network and be used for deep learning applications at a rate that is faster than any traditional machine learning technique [39].

### 3.3.1 Quantum Principal Component Analysis

Large datasets can have their dimensionality reduced using the dimensionality reduction approach known as principal component analysis. Accuracy must be sacrificed in order to reduce the dimensions, since we must choose which variables to remove without losing crucial data. If done correctly, dealing with a smaller dataset is significantly more convenient, which greatly eases the machine learning work. A traditional computer can efficiently do principal component analysis on a dataset with ten input attributes, for example. However, the traditional methods of principal component analysis will not work if the input dataset has a million features since it will be difficult to show the relative value of each feature. A further exponential improvement over existing techniques is provided by quantum principal component analysis (qPCA), which builds the eigenvectors corresponding to the large eigenvalues of the state (the principal components) in time *O(log d)* using multiple copies of an unknown density matrix. Additionally, state assignment and discrimination can benefit from quantum principal component analysis. Assume, for instance, that we are able to select samples from two sets of m states, the first set $\{|\varphi_i\rangle\}$ of which is designated by the density matrix $\rho = \frac{1}{m}\sum_i |\varphi_i\rangle\langle\varphi_i|$ and the second set $\{|\psi_i\rangle\}$ of which is designated by the density matrix $\sigma = \frac{1}{m}\sum_i |\psi_i\rangle\langle\psi_i|$ [40]. Now, $|x\rangle$ can be decomposed in terms of the eigenvectors and eigenvalues of the $\rho - \sigma$ using density matrix exponentiation and quantum phase estimation,

$$|x\rangle|0\rangle \rightarrow \sum_j x_j|\xi_j\rangle|x_j\rangle$$

Where $\rho - \sigma$ and $|x\rangle$ are the eigenvalues and $|\xi_j\rangle$ are the corresponding eigenvectors. The set of related eigenvectors and eigenvalues increases in proportion to the input's higher dimensionality. Quantum Random Access Memory (QRAM), which selects a data vector at random, enables quantum computers to handle this problem quickly and efficiently. Any data vector's quantum representation can be broken down into its constituent parts. Thus, there is an exponential reduction in both computational complexity and time complexity.

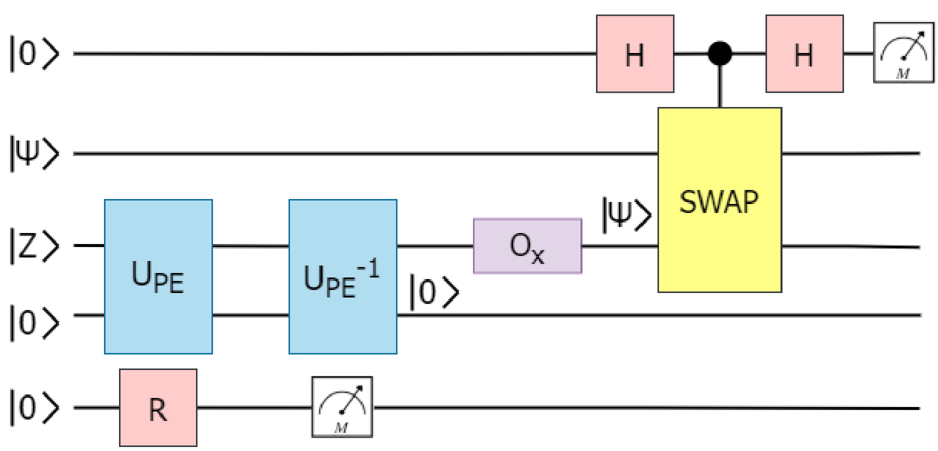


Figure 3.11: Quantum Circuit for performing PCA

### 3.3.2 Quantum Generative Adversarial Networks

Generative adversarial network has become one of the most intriguing new developments in machine learning. It has excelled at a number of difficult jobs, including the creation of images and videos. A quantum variant of generative adversarial learning has recently been theoretically presented and demonstrated to have the ability to outperform its classical equivalent exponentially. Generating data that matches the original data used for training is the aim of generative adversarial networks (GANs) [41]. To do this, we simultaneously train a generator and a discriminator neural network. The generator's task is to provide synthetic data that mimics the actual training dataset. The discriminator, on the other hand, behaves like a detective attempting to separate authentic data from fraudulent data. Both participants iteratively develop with one another during the training phase. By the time it's finished, the generator should produce fresh data that closely resembles the training set.

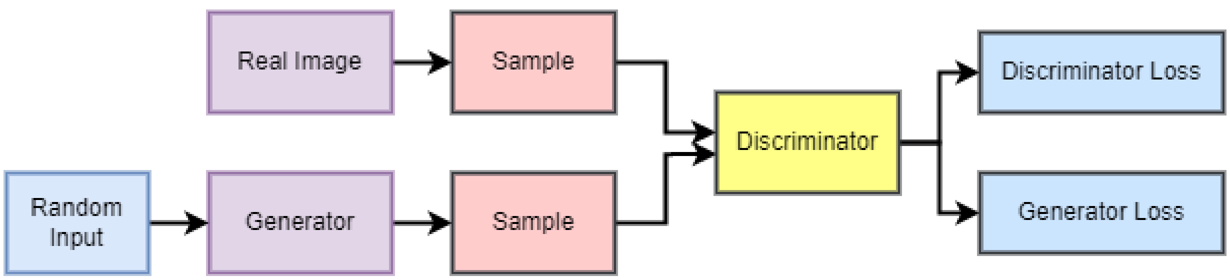


Figure 3.12: Overview of GAN structure

The generator's task is to attempt to model the unknown data distribution $P_{data}$ that the training dataset represents samples from. Starting with some initial latent distribution, P$z$, the generator, G, maps it to $Pg = G(Pz)$. Except for the simplest activities, this concept is not often realized in practice. In a 2-player minimax game, the discriminator (D) and generator (G) both participate. While the generator aims to minimize this probability, the discriminator seeks to maximize the likelihood of separating authentic data from fraudulent. The game's value function can be summed up as follows:

$$min\,max\,V(D,G) = E_{x\sim P_{data}}[log\,D(x)] + E_{z\sim P_z}[log(1 - D(G(z))]$$

where, $x$ is the real data, $D(x)$ is the probability of classifying real data as real for the discriminator, $z$ is the latent vector, $G(z)$ represents fake data and $D(G(z))$ is the probability of classifying fake data as real for the discriminator.

In actual training, each of the two networks has a different loss function that has be minimized,

$$Discriminator\ loss, L_D = -\left[y.\log\big(D(x)\big) + (1-y)\log\Big(1 - D\big(G(z)\big)\Big)\right]$$

$$Generator\ loss, L_G = [(1-y)\log(1 - D(G(z)))]$$

where y is a binary indicator of whether the data is real (y=1) or fake (y=0). In reality, maximizing log(D(G(z)) rather than minimizing log(1D(G(z))) makes generator training more stable [42].

# CHAPTER 4: METHODOLOGY

This chapter covers the research methodology with system diagrams. The main objective of this work is to implement the Quantum Generative Adversarial Network (GAN) and compare it with its classical counterpart. The implementation approach adopted in this study is divided into:

- Quantum GAN: Patch Method
- Quantum GAN: Quantum State Fidelity Based

We generated real-world hand-written digit pictures using a superconducting quantum processor through learning. Additionally, we use a gray-scale bar dataset to compare the performance of conventional and quantum GANs built using multilayer perceptron and convolutional neural network architectures, respectively.

Before jumping into quantum implementation, we have implemented the classical GAN and evaluated the model based on the loss. The workflow of the classical generative adversarial network is depicted on figure 4.1.

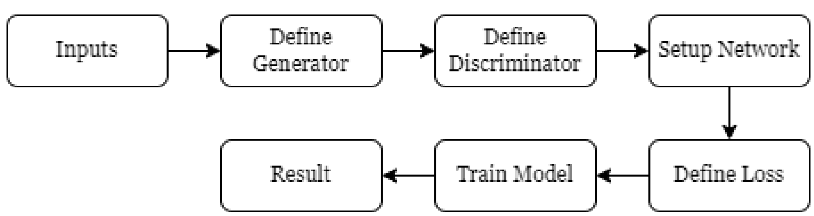


Figure 4.1: Block diagram of classical approach

We start with the MNIST dataset [43] containing 60,000 images with hand-written digits. The resolution of each image is 28 x 28. The generator is defined in such a way that it takes random noise as input and then transforms it into a fake image. The discriminator then classifies the generated images into real or fake. The loss is calculated from the discriminator classification.

Our study opens a path for investigating quantum benefits in various GAN-related learning tasks and offers guidelines for creating enhanced quantum generative models on near-term quantum devices. The following covers both implementations of Quantum Generative Adversarial Network.

## 4.1 Quantum GAN: Patch Method

We recreated the patch approach, one of the quantum GAN techniques Huang et al. [44] described. This technique employs a number of quantum generators, each of which, designated as $G^{(i)}$, is in charge of creating a tiny patch of the final image. Combining all of the patches together creates the finished image, as shown in figure 4.2.

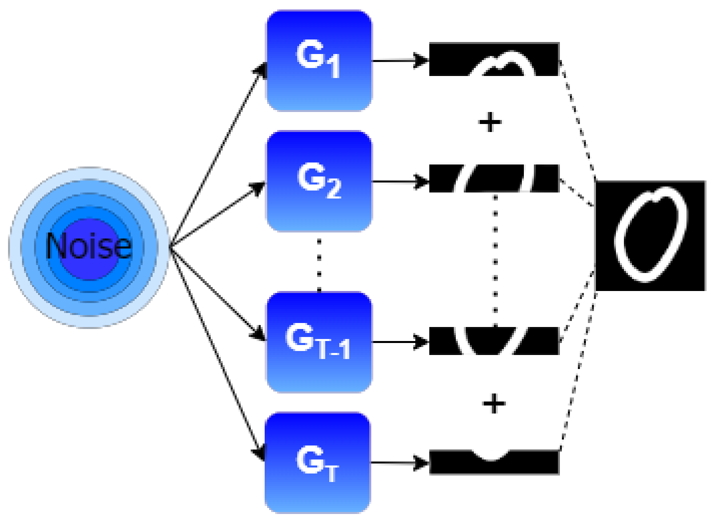


Figure 4.2: Process of generating final image using small patches

The key benefit of this approach is that it excels in circumstances when the number of qubits at hand is constrained. Iteratively using the same quantum device for each sub-generator is possible, as is parallelizing the execution of the generators over several devices [45].

### 4.1.1 Implementation of the Discriminator

In this method, we use a fully connected neural network with two hidden layers as the discriminator. For the purpose of representing the likelihood that an input would be labeled as real, just one output is required.

## 4.1.2 Implementation of the Generator

The circuit design for each sub-generator, $G^{(i)}$, is represented in the figure 4.3. Each of the $N_G$ sub-generators that make up the overall quantum generator has $N$ qubits. State embedding, parameterization, non-linear transformation, and post-processing are the four discrete steps that separate the latent vector input to picture output process. To make the topic more straightforward, the parts that follow each pertain to a particular training cycle.

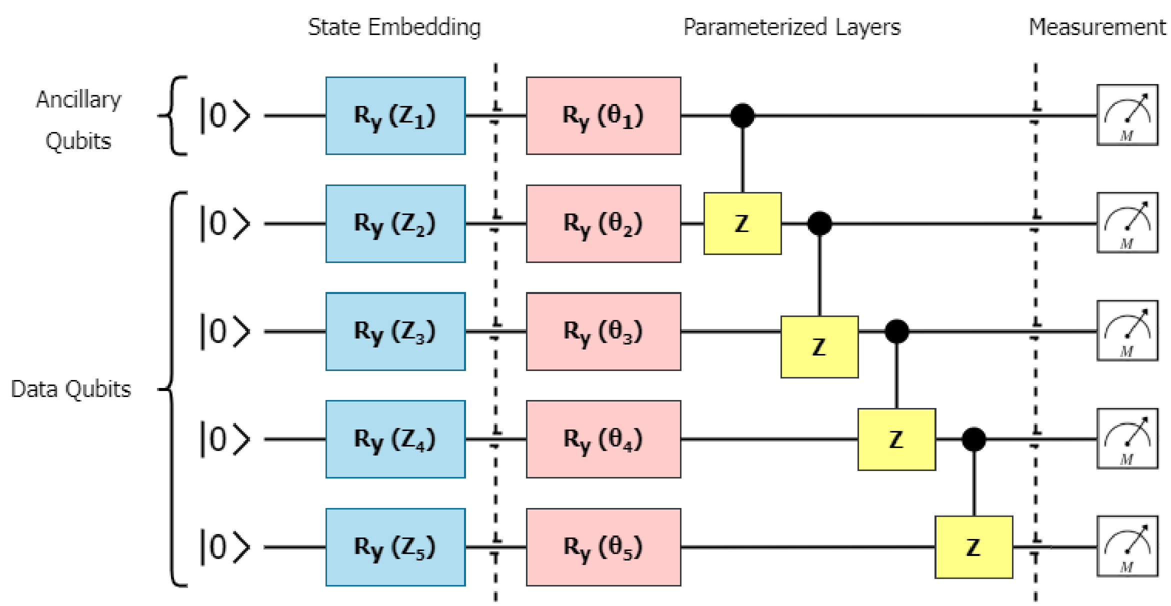


Figure 4.3: Circuit diagram of Generator

### 4.1.2.1 State Embedding

A sample of a latent vector, $z \in R^N$, in the range $[0, \pi/2)$ is taken from a uniform distribution. The same latent vector is sent to each sub-generator, and it is then embedded using RY gates. The parameterized RY gates followed by the control Z gates creates a parameterized layer which is repeated D times.

### 4.1.2.2 Non-Linear Transform

In the circuit model, there are unitary quantum gates, which by definition linearly alter the quantum state. We require non-linear transformations because a linear mapping between the latent and generator distributions would only work for the simplest generating tasks. To assist, we employ auxiliary qubits.

The pre-measurement for a certain sub-generator, the quantum state is provided by,

$$|\Psi(z)\rangle = UG(\theta)|z\rangle$$

Where $UG(\theta)$ stands for the overall unitary of the parameterized layers.

#### 4.1.2.3 Post Processing

All of the components in $g^{(i)}$ must add up to one due to the measurement's normalization restriction. If we utilize $g^{(i)}$ as the pixel intensity values for our patch, we face an issue. Considering a scenario where a group of full intensity pixels are the intended objective. A patch of pixels with a magnitude of $\frac{1}{2^{N-N_A}}$ would be the best one a sub-generator could create [44]. We use a post-processing approach on each patch to ease this restriction,

$$x^{(i)} = \frac{g^{(i)}}{\max_k g_k^{(i)}}$$

So, the generated final image is $x = [\, x^{i}, \ldots, x^{(N_G)} \,]$.

### 4.1.3 System Design of Patch GAN

A quantum generator *G* and a discriminator *D*, which can be either classical or quantum, are both components of the proposed quantum GANs method. The quantum patch GAN works in the manner described below. The latent state $|z\rangle$ sampled from the latent space is first fed into a PQC $U_{Gt}(\theta_t)$–built quantum generator G made of T sub-generators (highlighted in pink region). The produced states $\{U_{Gt}(\theta_t)|z\rangle\}^T$ for t=1, along the computation basis are then measured to get the resultant picture. The genuine picture and the patched produced image are then sequentially sent into the traditional discriminator D (highlighted in pink region). In order to update the trainable parameters for G and D, a classical optimizer utilizes the categorized results as the output of D [44]. The procedure completes one iteration.

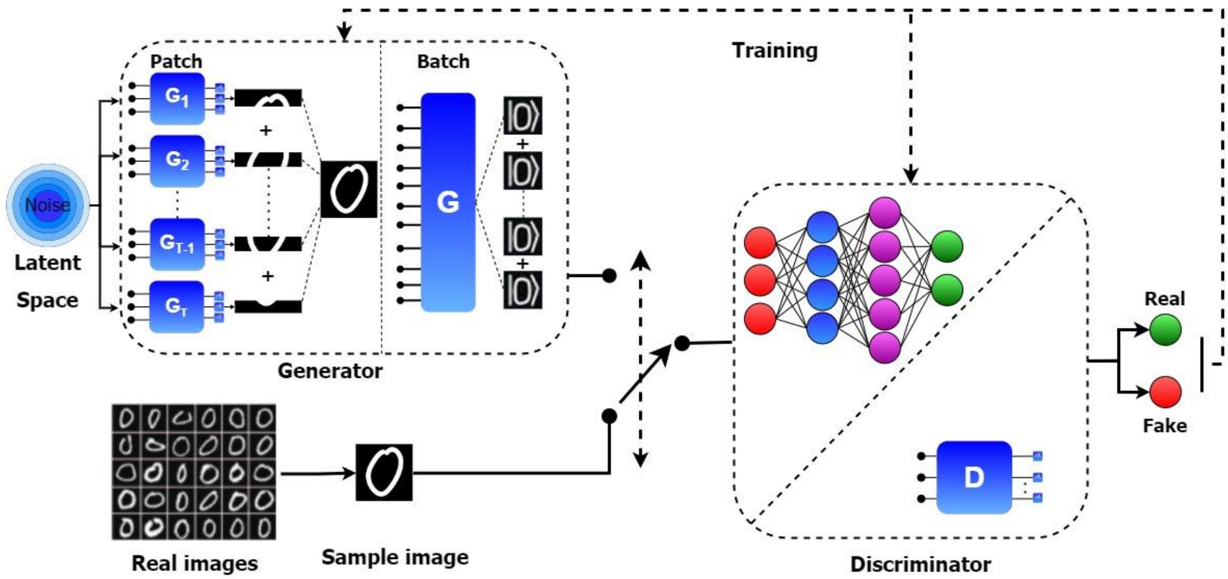


Figure 4.4: Quantum patch GAN model

where D operates with a quantum state that has been encoded with the real picture, and the output is obtained by a straightforward measurement. The quantum generator and quantum discriminator utilize PQC in their implementation. Quantum patch and batch GANs use the same quantum generating equipment. Two produced states are concurrently supplied into the quantum discriminator to achieve the nonlinear behavior.

### 4.1.4 Hyper-parameters

The model is trained based on the following hyper-parameters:

- Learning rate for the generator = 0.3
- Learning rate for the discriminator = 0.01
- Number of training iterations or epoch = 700

Quantum Variables

- Total number of qubits = 5
- Number of ancillary qubit = 1
- Depth of the parameterized quantum circuit = 6
- Number of sub-generators = 4

## 4.2 QUANTUM GAN: QUANTUM STATE FIDELITY BASED

One of the pioneering works that uses quantum-state based loss functions for the Discriminator and Generator is the QuGAN model. The swap test, a quantum computation technique that assesses how much two quantum states vary, is frequently used in applications of quantum machine learning and allows for quantum fidelity measurements, which in turn enable these loss functions [46]. Thus, our design appears to be more reliable and effective than the quantum discriminators and quantum generators suggested in the earlier work [47]. Hellinger distance, which measures how close the original dataset and generated probability distributions are to one another, significantly improves when QuGAN is tested with the MNIST dataset.

We present quantum-state based loss functions with quantum gradients for both the Discriminator and the Generator for the GAN models, which are based on quantum fidelity measurements. The system is tested on MNIST dataset with the PCA reduced dimensions [48].

### 4.2.1 System Design of QuGAN

#### 4.2.1.1 Quantum Deep Learning Layers

The QuGAN system iterates until a predetermined number of goals have been achieved. The classical data that feeds into the QuGAN architecture is first standardized and converted into quantum data. Every data set goes through this data preparation procedure once. The quantum circuit receives either a Generator ($G/|\gamma\rangle$)/Discriminator ($D/|\delta\rangle$) circuit or a Real Data ($X/|\xi\rangle$)/Discriminator ($D/|\delta\rangle$) circuit, depending on the training algorithm's stage. The classical computer receives the quantum circuits of induced state fidelity. When the system is being optimized, each gate's gradient with respect to the objective function is calculated using the fidelity, which is then used to update the Generator and Discriminator parameters. The system architecture is depicted in Figure 4.5.

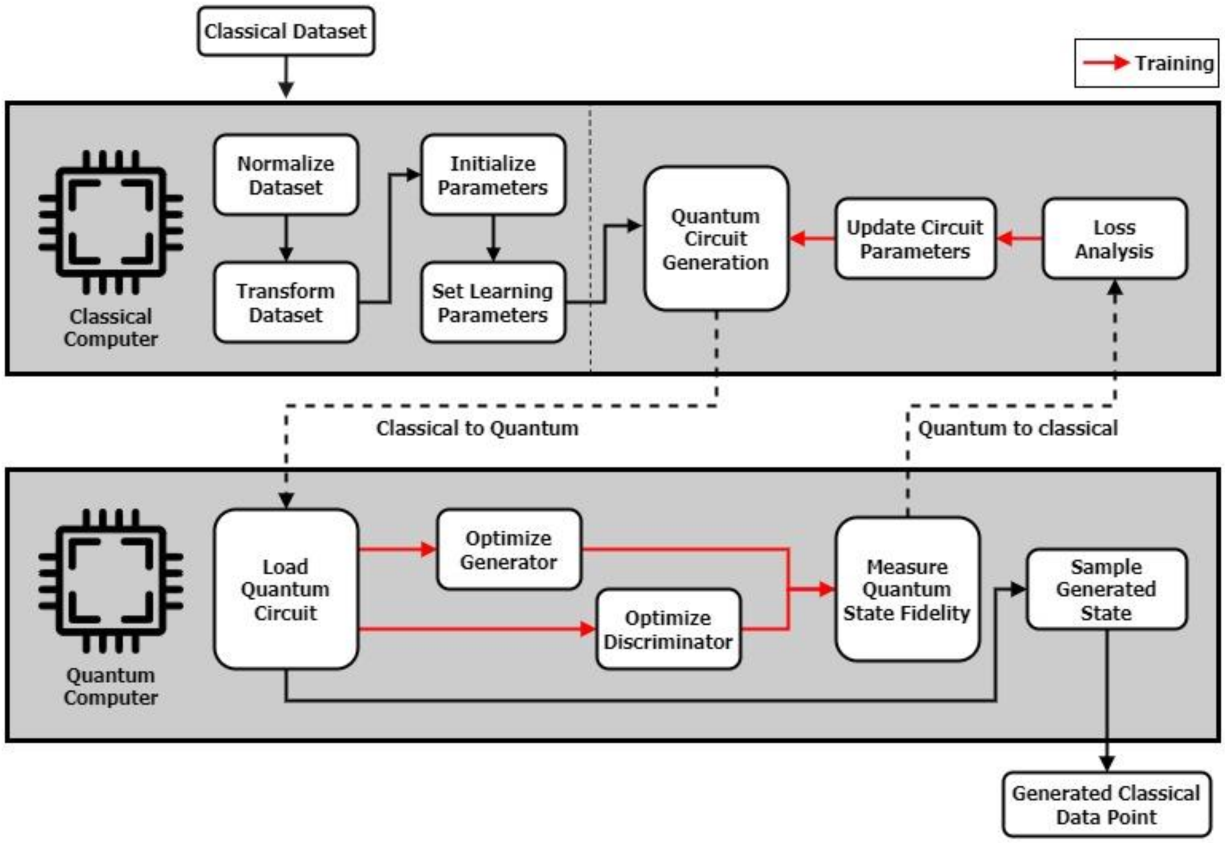


Figure 4.5: System architecture of QuGAN

#### 4.2.1.2 Quantum based Optimization

The loss function,

$$D_{loss} = E\left[\log\left(D(|\langle\xi;\ \delta\rangle|^2)\right)\right] + \ E\left[\log\left(1 - D(|\langle\gamma, \delta\rangle|^2)\right)\right]$$

$$G_{loss} = E\left[\log\left(D(|\langle\gamma, \delta\rangle|^2)\right)\right]$$

Is used to illustrate optimization of a quantum GAN. The Generator Circuit, Discriminator Circuit, Data Loading Circuit, and the Anicilla qubit are the four main quantum circuit elements that make up the design of our QuGAN. The quantum deep learning (QDL) model responsible for creating samples is represented by the Generator circuit, and the QDL model that is the data loading circuit loads quantum data onto a quantum state; the anicilla qubit is a single qubit used to measure the fidelity of the quantum state and to carry out our inter-circuit communication; and the duty of distinguishing between false and real samples [49].

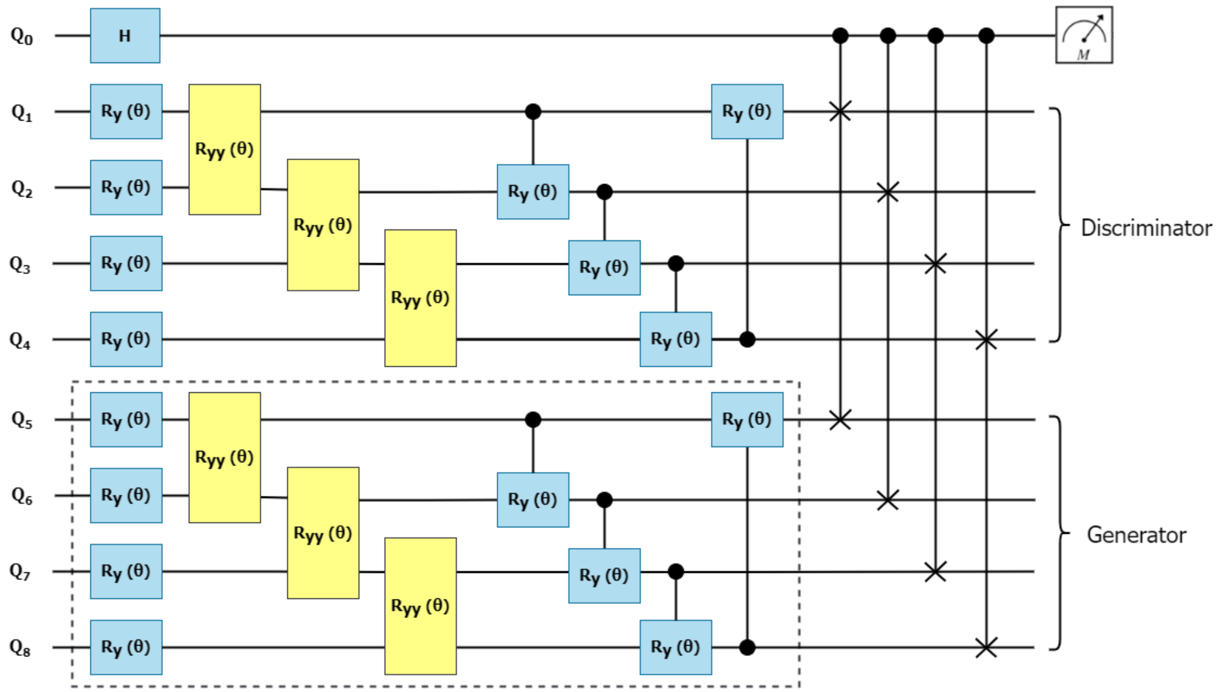


Figure 4.6: Layers applied on the Generator

On the illustration, the Discriminator and Generator each have ten parameters. The Discriminator uses qubits 1 through 4, while qubits 5 through 8 are set aside for the Generator's settings or data loading. The ancilla qubit for the SWAP test is qubit zero. Qubit 0 is mapped onto a conventional bit that may either read 1 or 0.

## 4.2.2 Hyper-parameters

The model is trained based on the following hyper-parameters:

- Learning Rate: $\alpha = 0.01$
- Epoch: $\epsilon = 25$
- Network Weights: $\theta_d$ = Random (0,1) x $\pi$

Quantum Variables:

- Qubit Channels: $Q = 1 + (n_{X_{dim}} \text{ x } 2)$

# CHAPTER 5: RESULTS AND ANALYSIS

As established before, we have implemented two different quantum-based GAN models and are comparing them with the classical implementation of GAN. First, we'll see the results of classical GAN, then quantum patch GAN, then finally quantum state-based QuGAN.

## 5.1 CLASSICAL GAN RESULTS

It generated excellent quality synthetic images from the dataset. As we can see, the losses almost converge with each other from about 4000 epochs onwards.

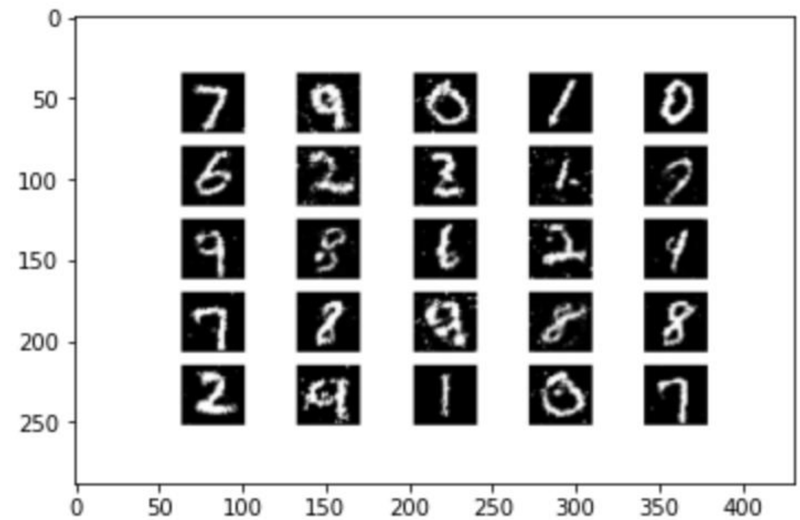

Figure 5.1: Generated Synthetic Images from the Classical GAN implementation

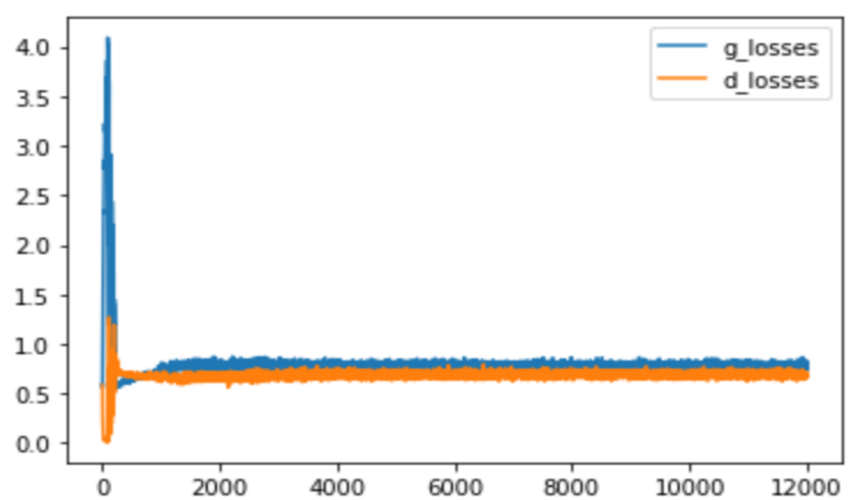


Figure 5.2: Generator, Discriminator Losses for Classical GAN

## 5.2 Quantum Patch GAN Results

For a set of 0s chosen from the dataset, the patch method was applied to reconstruct fake 0s for 700 epochs. The Discriminator Loss started to fall from the 700th epoch onwards and the results started to deteriorate.

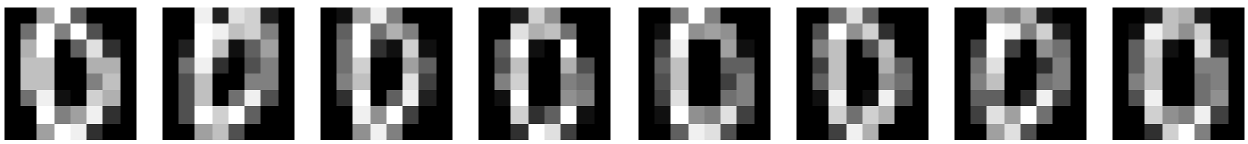

Figure 5.3: 0s chosen as input data from the MNIST

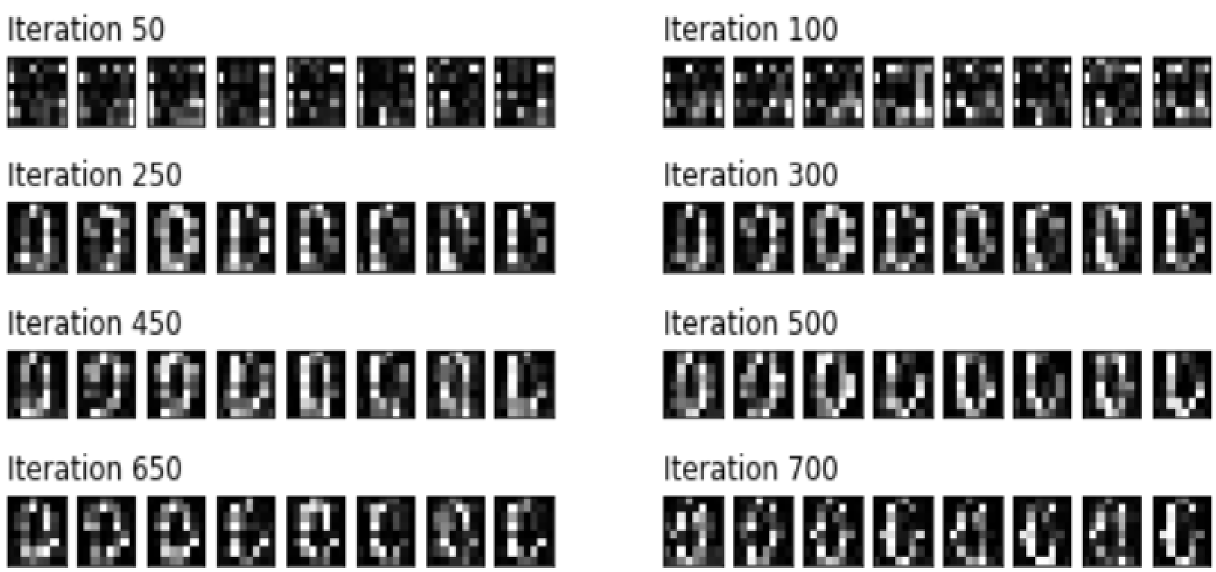


Figure 5.4: Generated fake images from patch GAN

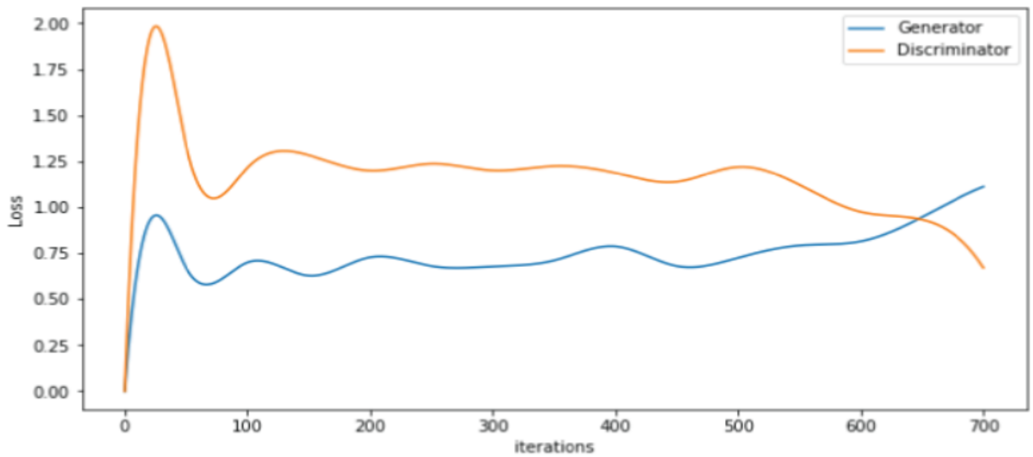


Figure 5.5: Generator and Discriminator Losses for Quantum Patch GAN

## 5.3 Quantum State Fidelity based QuGAN Results

The target was to generate 3s, 6s, 9s from the MNIST dataset. We successfully generated decent-quality 9s, and mildly recognizable 3s and the model was trained with those corresponding images. Best result was achieved at the 50th epoch.

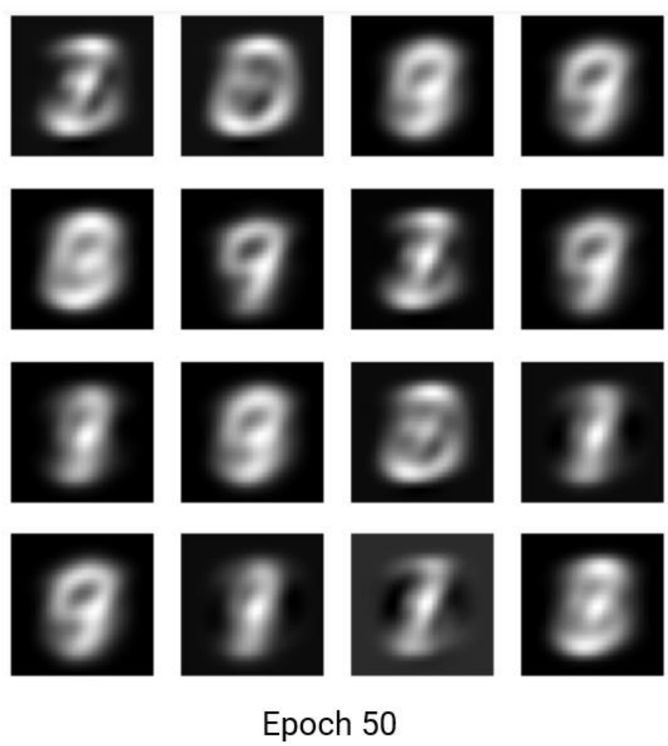


Figure 5.6: Generated fake images by QuGAN at 50th epoch

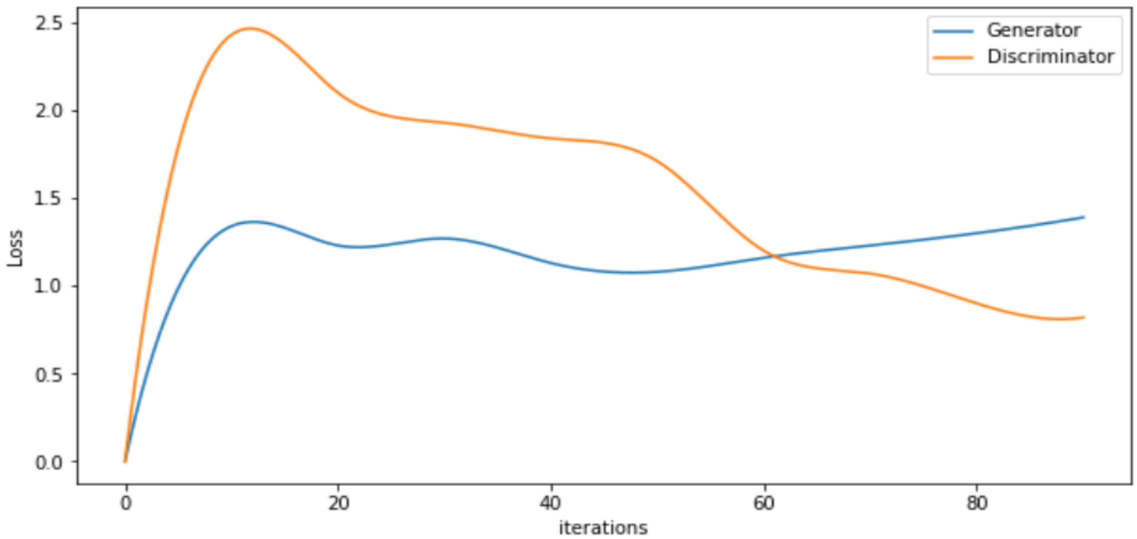


Figure 5.7: Generator and Discriminator Losses for QuGAN

# CHAPTER 6: CONCLUSION

In conclusion, this research aimed to implement and compare two quantum generative models, the Quantum Patch GAN and the QUGAN Quantum State GAN, with a classical generative model, the Generative Adversarial Network (GAN). The results showed that both quantum models were able to generate high-quality images, though the QuGAN model had some limitations.

Compared to the classical GAN, the quantum models showed significant improvements in terms of computational efficiency and stability. The use of quantum algorithms allowed for faster convergence and the avoidance of mode collapse, a common issue faced by classical GANs.

However, it is important to note that quantum models have their own challenges and limitations. The current implementation of quantum generative models is still in its early stages, and further research is needed to address the limitations and improve their performance. Additionally, quantum models also require specialized hardware and infrastructure that may not be accessible to all researchers, which could limit their widespread adoption.

Finally, this research provides valuable insights into the potential of quantum generative models in computer vision and opens up avenues for further exploration. The results highlight the importance of considering quantum algorithms as a viable option for tackling complex problems in the field of machine learning and artificial intelligence. This research might work as an inspiration to future researchers in this field to try and implement the quantum counterpart of a classical GAN such as: a quantum implementation of a CycleGAN.

# APPENDIX

Python code to prepare the Generator

```
class PatchQuantumGenerator(nn.Module):

    def __init__(self, n_generators, q_delta=1):
        super().__init__()
        self.q_params = nn.ParameterList(
            [
                nn.Parameter(q_delta * torch.rand(q_depth * n_qubits), requires_grad=True)
                for _ in range(n_generators)
            ]
        )
        self.n_generators = n_generators

    def forward(self, x):
        # Size of each sub-generator output
        patch_size = 2 ** (n_qubits - n_a_qubits)
        images = torch.Tensor(x.size(0), 0).to(device)

        # Iterate over all sub-generators
        for params in self.q_params:

            patches = torch.Tensor(0, patch_size).to(device)
            for elem in x:
                q_out = partial_measure(elem, params).float().unsqueeze(0)
                patches = torch.cat((patches, q_out))

            images = torch.cat((images, patches), 1)

        return images
```

Python code to prepare the Discriminator

```
class Discriminator(nn.Module):

    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            # Inputs to first hidden layer
            nn.Linear(image_size * image_size, 64),
            nn.ReLU(),
            # First hidden layer
            nn.Linear(64, 16),
            nn.ReLU(),
            # Second hidden layer
            nn.Linear(16, 1),
            nn.Sigmoid(),
        )
    def forward(self, x):
        return self.model(x)
```

# REFERENCES


[1] F. T. S. J. Lianchao Jin, "Generative Adversarial Network Technologies and Applications in Computer Vision," *Computational Intelligence and Neuroscience,* pp. 1-17, 2020.

[2] Y. D. M. G. Y. Z. He-Liang Huang, "Experimental Quantum Generative Adversarial Networks for Image Generation," *Phys. Rev. Applied 16, 024051,* vol. 16, 2021.

[3] B. B. D. C. Y. M. Samuel A. Stein, "QuGAN: A Quantum State Fidelity based Generative Adversarial Network," *IEEE International Conference on Quantum Computing and Engineering,* pp. 71-81, 2021.

[4] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine,* vol. 29, pp. 141-142, 2012.

[5] X. S. Y. D. Jinkai Tian, "Recent Advances for Quantum Neural Networks in Generative Learning.," no. doi = {10.48550/ARXIV.2206.03066, 2022.

[6] S. L. M. Tsang, "Hybrid Quantum-Classical Generative Adversarial Network for High Resolution Image Generation," no. https://doi.org/10.48550/arxiv.2212.11614, 2022.

[7] J. P.-A. ,. M. M. Ian Goodfellow, "Generative adversarial networks," vol. 63, no. 11, pp. 139-144, 2020.

[8] S. N. J. Z. X. W. Z Cao, "Fast generative adversarial networks model for masked image restoration," *IET Image Processing,* vol. 13, pp. 1124-1129, 2019.

[9] J. R. R. J. Z. Z. Y Zhou, "Unsupervised Adversarial Network Alignment with Reinforcement Learning," *ACM Transactions on Knowledge Discovery from Data,* vol. 16, pp. 1-29, 2021.

[10] Y. P. M. Y. J Zhang, "SCH-GAN: Semi-Supervised Cross-Modal Hashing by Generative Adversarial Network," *IEEE transactions on cybernetics,* vol. 50, no. 2, pp. 489-502, 2018.

[11] A. Steane, "Quantum computing," Reports on Progress in Physics , 1998.

[12] J. FRANKENFIELD, "Quantum Computing: Definition, How It's Used, and Example," Investopedia, 28 August 2022. [Online]. Available: https://www.investopedia.com/terms/q/quantum-computing.asp#citation-7. [Accessed 30 December 2022].

[13] Z. H. Y. W. H Situ, "Quantum generative adversarial network for generating discrete distribution," *Information Sciences,* vol. 538, pp. 193-208, 2020.

[14] Y. W. J. L. L. W. J Zeng, "Learning and Inference on Generative Adversarial Quantum Circuit," *Physical Review A,* vol. 99(5), p. 052306, 2019.

[15] Y. Z. Z. D. W Liu, "A hybrid quantum-classical conditional generative adversarial network algorithm for human-centered paradigm in cloud," *EURASIP Journal on Wireless Communications and Networking,* no. 1, pp. 1-17, 2021.

[16] Z. L. Y. H. R. G. W Ren, "Quantum generative adversarial networks for learning and loading quantum image in noisy environment," *Modern Physics Letters B,* no. 21, 2021.

[17] A. L. S. W. C Zoufal, "Quantum Generative Adversarial Networks for learning and loading random distributions," *Quantum Information,* no. 1, 2019.

[18] "Entangling Quantum Generative Adversarial Networks," *Physical Review Letters,* vol. 128, no. 22, 2022.

[19] S. Chakrabarti, "Quantum wasserstein generative adversarial networks," *Advances in Neural Information Processing Systems,* vol. 32, 2019.

[20] J. B. C Bravo-Prieto, "Style-based quantum generative adversarial networks for Monte Carlo events," *Quantum, 2022 - quantum-journal.org,* vol. 6, p. 777, 2022.

[21] M. U. Sarah M. Erfani, "Hybrid Quantum-Classical Generative Adversarial Network for High Resolution Image Generation," no. https://arxiv.org/abs/2212.11614, 2022.

[22] E. P. G Agliardi, "Optimal Tuning of Quantum Generative Adversarial Networks for Multivariate Distribution Loading," *Quantum Reports, 2022 - mdpi.com,* vol. 4, pp. 75-105, 2022.

[23] S. T. a. H. S. K. P. Marella, "Introduction to Quantum Computing," *Quantum Computing and Communications,* 2020.

[24] "Qubit," Devopedia, 30 January 2022. [Online]. Available: https://devopedia.org/qubit. [Accessed 20 December 2022].

[25] D. T. Christopher Havenstein, "Comparisons of Performance between Quantum and Classical," *SMU Data Science Review,* vol. 11, 2018.

[26] T. a. R.-K. A. Khan, "Machine Learning: Quantum Vs Classical," *IEEE Access,* vol. 8, 2020.

[27] J. FRANKENFIELD, "Quantum Computing: Definition, How It's Used, and Example," Investopedia, 28 August 2022. [Online]. Available: https://www.investopedia.com/terms/q/quantum-computing.asp. [Accessed 17 Decmeber 2022].

[28] "Superposition and entanglement," Quantum Inspire, December 2020. [Online]. Available: https://www.quantum-inspire.com/kbase/superposition-and-entanglement. [Accessed 25 December 2022].

[29] M. B. P. Shashank Virmani, "An introduction to entanglement theory,," *Quantum information and coherence,* pp. 173-208, 2014.

[30] "Single Qubit Gates," Qiskit, 2021. [Online]. Available: https://qiskit.org/textbook/ch-states/single-qubit-gates.html. [Accessed 25 December 2022].

[31] S. N. a. S. Nayak, "An Introduction to Basic Logic Gates for Quantum Computer," 2013.

[32] C.-R. Wie, "Two-Qubit Bloch Sphere," *Physics,* vol. 2, p. 383–396, Aug, 2020.

[33] C.-R. Wie, "two-qubit-Bloch-sphere," Github, [Online]. Available: https://github.com/CRWie/two-qubit-Blochsphere . [Accessed 19 December 2022].

[34] I. M. L. B. N. B. L. D. V. C. M. B. F. S. C. F. C. Ilaria Gianani, "Experimental Quantum Embedding for Machine Learning," *Advanced Quantum Technologies,* vol. 5, p. 2100140, June, 2022.

[35] "Encoding Classical Data into Quantum States," Institute for Quantum Computing, Baidu Inc, 2021. [Online]. Available: https://qml.baidu.com/tutorials/machine-learning/encoding-classical-data-into-quantum-states.html. [Accessed 19 December 2022].

[36] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Physical Review Letters,* vol. 122, 2019.

[37] C. K. Williams, "Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond," *Journal of the American Statistical Association,* 2003.

[38] T. Y. Rupak Chatterjee, "Generalized Coherent States, Reproducing Kernels, and Quantum Support Vector Machines," vol. 17, 2017.

[39] "Beginner's Guide to Quantum Machine Learning," Paperspace Blog, 2020. [Online]. Available: https://blog.paperspace.com/beginners-guide-to-quantum-machine-learning. [Accessed 25 December 2022].

[40] M. M. P. R. Seth Lloyd, "Quantum principal component analysis," *Nature Physics,* vol. 10, pp. 631-633, 2014.

[41] S. O. A. C. Ian Goodfellow, "Generative Adversarial Networks," *Commun. ACM,* vol. 63, pp. 139-144, 2020.

[42] J. Ellis, "Quantum GANs," Pennylane, 1 February 2022. [Online]. Available: https://pennylane.ai/qml/demos/tutorial_quantum_gans.html. [Accessed 1 January 2023].

[43] "MNIST database," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/MNIST_database. [Accessed 30 December 2022].

[44] Y. D. He-Liang Huang, "Experimental Quantum Generative Adversarial Networks for Image Generation," 2020.

[45] Goodfellow, "Generative adversarial nets," *Advances in neural information processing systems,* pp. 2672-2680, 2014.

[46] G. B. S. G. Esma Aïmeur, "Machine Learning in a Quantum World," *Conference of the Canadian Society for Computational,* pp. 431-442, 2006.

[47] G. V. T. M. Michael Broughton, "TensorFlow Quantum: A Software Framework for Quantum Machine Learning," 2020.

[48] "Visualizing data using t-sne," *Journal of machine learning research,* vol. 9, pp. 2579-2605, 2008.

[49] B. B. D. C. Samuel A. Stein, "QuGAN: A Quantum State Fidelity based Generative Adversarial Network," *IEEE,* 2021.