Generative Adversarial Game with Tailored Quantum Feature Maps for Enhanced Classification

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Anais Sandra Nguemto Guiawa

December 2023

© by Anais Sandra Nguemto Guiawa, 2023
All Rights Reserved.

To my parents.

Acknowledgments

I would like to extend my heartfelt appreciation to my parents, Nguemto Wakam Jean Jules and Noubiwa Therese Chantal, for their unwavering encouragement and support in both my academic and non-academic pursuits. Without your constant belief in me, I would not be where I am today. Your love and guidance have been instrumental in shaping my path, and I am deeply grateful for everything you have done for me.

To my siblings, Ines, Pierre-Henry, and Cedric, thank you for always cheering me on and providing unwavering support in various ways. Your presence and encouragement have been a constant source of motivation throughout my journey. I appreciate your love and support more than words can express.

I extend my heartfelt gratitude to Lister Ziemi, a cherished family friend for many years, for his invaluable contributions to my academic journey.

I would also like to express my gratitude to my friends, Lindsey Fossouo and Patricia Tongkoua. Your steadfast friendship and unwavering support have been invaluable to me. Thank you for always being there for me, cheering me on, and lending an ear when I needed it the most. Your presence in my life has been a true blessing.

During my time as a Ph.D. candidate, I had the privilege of encountering exceptional mentors and collaborators who played a vital role in my growth and development. Foremost among them is Dr. Vicente Leyton Ortega, who not only taught me the intricacies of research but also guided me with unparalleled kindness and professionalism throughout my training. I owe a great deal of my success to your mentorship and support.

I am also deeply appreciative of my advisor, Dr. Suzanne Lenhart, for her guidance and unwavering support during my tenure as a student at the University of Tennessee. Your

expertise and mentorship have been invaluable in navigating the challenges of academic life. I am truly grateful for your wisdom and encouragement.

I would like to acknowledge another mentor, Dr. Vasileios Maroulas, who has consistently encouraged me to seize opportunities and pursue paths that have proven beneficial to my career. Your guidance and belief in my abilities have been instrumental, and I am thankful for your mentorship.

To my fellow Ph.D. students at the University of Tennessee, Johannes Krotz and Barton Gattis, thank you for your friendship and support. Our interactions and shared experiences have contributed significantly to my growth as a researcher and an individual.

A special mention goes to my "productivity coach," Bounkeu. I want to express my heartfelt gratitude for your dedicated time and effort in helping me realize my full potential. Your guidance and support have been immensely valuable to me.

I would like to thank my friend Jules Audrey Tchamdou Tchatat, who entered my life relatively recently but has been a driving force in helping me reach the finish line of my Ph.D. journey. Your support and encouragement have been pivotal, and I am grateful for your presence in my life.

Lastly, I would also like to extend my thanks to my aunt Chantal Matchuam Wakam for her continuous support and contributions to both me and my family. Your presence and guidance have played a significant role in shaping the person I have become, and I am deeply appreciative of your love and support.

To everyone mentioned and to all those who have supported me along the way, thank you from the bottom of my heart. Your unwavering belief in me and your support have been vital to my success, and I am deeply appreciative of your presence in my life.

Abstract

In the burgeoning field of quantum machine learning, the fusion of quantum computing and machine learning methodologies has sparked immense interest, particularly with the emergence of noisy intermediate-scale quantum (NISQ) devices. These devices hold the promise of achieving quantum advantage, but they grapple with limitations like constrained qubit counts, limited connectivity, operational noise, and a restricted set of operations. These challenges necessitate a strategic and deliberate approach to crafting effective quantum machine learning algorithms.

This dissertation revolves around an exploration of these challenges, presenting innovative strategies that tailor quantum algorithms and processes to seamlessly integrate with commercial quantum platforms. A pivotal proposition within this framework is the creation of a resource-efficient quantum generative adversarial network (QGAN). This adaptation of QGANs, which are typically used to synthesize data according to specific probability distributions, ensures optimal performance even within environments with limited resources.

Furthermore, our research delves deeply into the development of a data encoding procedure specifically designed for NISQ devices. This encoding process, responsible for translating classical data into quantum states to enable quantum algorithm processing, plays a critical role in quantum machine learning. Our goal is to establish an encoding approach that optimally utilizes quantum resources while mitigating the impact of noise and inherent limitations in NISQ devices.

Another key aspect of our study is the seamless alignment of the devised algorithms with the existing architectures of NISQ devices. Given the pivotal role of these devices in contemporary quantum technology, ensuring compatibility is of utmost importance. This

not only facilitates immediate applications but also establishes a robust framework for accommodating future technological advancements.

Through an extensive analysis of these critical dimensions, our objective is to make a substantial contribution to the practical implementation of quantum machine learning algorithms on commercial quantum platforms. We aim to navigate the intricate landscape of NISQ technologies adeptly, thereby facilitating the seamless integration of quantum machine learning into real-world applications. Ultimately, this research endeavor aspires to drive advancements at the nexus of quantum computing and machine learning.

Table of Contents

1 Introduction **1**

1.1 Quantum Computing . . . 2

1.2 Machine Learning . . . 5

1.2.1 Support Vector Machines . . . 7

1.2.2 Generative Adversarial Neural Networks . . . 10

1.3 Quantum Machine Learning . . . 13

1.3.1 Variational quantum algorithms . . . 16

1.3.2 Hybrid quantum classical framework . . . 19

1.3.3 Representing data on a quantum computer . . . 19

1.4 Research Questions and Contributions . . . 21

2 Re-QGAN: an Optimized Adversarial Quantum Circuit Learning Framework **23**

2.1 Introduction . . . 24

2.2 Results . . . 26

2.2.1 Quantum Adversarial Game . . . 26

2.2.2 Generating MNIST handwritten digits . . . 29

2.3 Methods . . . 32

2.3.1 Classical dataset encoding . . . 32

2.3.2 Quantum state generator and discriminator . . . 34

2.3.3 Interaction and optimization . . . 36

2.4 Preparation of an arbitrary two-qubit quantum state with real amplitudes . . . 38

2.5 Discussion . . . 39

3 Custom Quantum Feature Map for Enhanced Classification **42**

3.1 Introduction 42

3.2 Methods 43

3.2.1 Stereographic encoding feature map 43

3.2.2 Variational quantum classifier 46

3.2.3 Kernel based classifier 46

3.2.4 Noise configuration 48

3.3 Results 49

3.4 Discussion 49

Bibliography **53**

Vita **53**

List of Tables

2.1 List of parameters used in the numerical experiments. We present the number of parameters (Y_θs) and entangling gates (CXs) used in every case. In addition, we present the number of components used and the cumulative explained variance (CEV). The training dataset is a subset of 20 images from MNIST. 31

3.1 Average F1 and Accuracy scores for the feature maps without noise. A.E: Amplitude Encoding, S.E: Stereographic Encoding, ZZ: ZZ Feature Map. . . 50

3.2 Average F1 and Accuracy scores for the feature maps with noise. A.E: Amplitude Encoding, S.E: Stereographic Encoding, ZZ: ZZ Feature Map. . . 51

List of Figures

1.1 Illustration of a Bloch Sphere. This sphere is a geometric representation of a qubit, the fundamental unit of quantum information. Each point within or on the surface of the sphere represents a valid quantum state, with the north and south poles corresponding to the basis states $|0\rangle$ and $|1\rangle$. The associated amplitudes and their phases determine a quantum state's location on or within the sphere. Pure quantum states lie on the surface of the sphere, while mixed states, a statistical mixture of pure states, are situated within the sphere. . . 3

1.2 A parameterized quantum circuit, with 2 types of operations: a 1-qubit parameterized gate and a 2-qubit non parametric gate 18

1.3 The hybrid quantum classical framework in quantum computing an quantum machine learning 20

2.1 MNIST benchmark generation problem. In this Figure, we present the re-qGAN generated digits using a subset of the MNIST as the reference dataset. In a) we present the generated digits for each qubit configuration, we compress the reference dataset according to the number of qubits; see Table 2.1 for the level of compression n_{PCA} and re-qGANs characteristics used. In panel b), the learning for the discriminator and generator at various epochs (Epoch 1, 15, and 25), for four qubits, learning the digit '8'. Each epoch represents one learning cycle, with 500 CMA-ES iterations. In the top row of this panel, we have the evolution concerning the number of CMA-ES iterations of the discriminator and generator's loss functions, $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively. In the bottom row, we have the evolution of the variances of $\mathcal{L}_D$ and $\mathcal{L}_G$ throughout the optimization process. In panel c), we present representative elements of the generated samples by re-qGAN for the digits '3', '5' and '8' using four qubits and 50 training samples per digit at different epochs, namely, epochs 1, 15, and 25. Here, the cumulative explained variance for the PCA transformation is $\sim 75\%$. 30

2.2 Sketch of the quantum adversarial game elements. In a), we present an illustration of the SI encoding of a subset of the MNIST. We compress the images using the principal component analysis (PCA) and project them onto a hypersphere to define the representative quantum state amplitudes. On-hardware implementation requires the preparation of every element of the set $\mathcal{C}_Q$ by using the PQC used for the generator. b) Once we get the real-data set, $\mathcal{C}_Q$, we proceed to the sequential discriminator and generator training. Inside, OPT is the classical optimization algorithm that updates the discriminator or generator parameters. The stop criteria depend on the optimizer choice. In c) we present the generator quantum circuit, the ansatz prepares a 2-qubit state and entangles it with the third qubit through a uniformly controlled rotation (UCR). This PQC can be extended to more qubits by applying a cascade of UCRs. For the 4-qubit case, we need to add the UCR U_3 growing the PQC parameter set in 2^3 more elements. At the bottom, the relation between $U_3^{0,1,2}$ and $U_3^{1,2}$. The recurrence can be used for higher levels of UCRs (see text for more details). d) Three-qubit discriminator ansatz, $\mathcal{G}^3$. The ansatz transforms the generated quantum state or $\mathcal{C}_Q$ to a quantum state in a higher dimensional Hilbert space. The measurement in the ancillary qubit serves as a metric to establish the nature of the input. 33

2.3 The expressive power of the 2-qubit discriminator. We represent the discrimination power by its projection on the characteristic plane for each discriminator parameter. In a) the quantum circuit for the two-qubit discriminator, b) the SI encoding model for a single-qubit, and c) heat plots showing how the discriminator can separate data classes, identified by the values of $\langle Z \rangle = \pm 1$. 37

3.1 We compress the images using the principal component analysis (PCA) and project them onto a hypersphere to define the representative quantum state amplitudes. On-hardware implementation requires the preparation of every element of the set $\mathcal{C}_Q$ by using a PQC that prepares quantum states with real amplitudes 45

3.2 we present the parameterized quantum circuit for data loading, the ansatz prepares a 2-qubit state and entangles it with the third qubit through a uniformly controlled rotation (UCR). This PQC can be extended to more qubits by applying a cascade of UCRs. For the 4-qubit case, we need to add the UCR U_3 growing the PQC parameter set in 2^3 more elements. At the bottom, the relation between $U_3^{0,1,2}$ and $U_3^{1,2}$. The recurrence can be used for higher levels of UCRs. 47

Chapter 1

Introduction

Quantum computing, with its potential to solve complex problems far beyond the scope of classical computing, has emerged as a fascinating field of research. Its principles, drawing from the depths of quantum mechanics, offer a profound shift in how we approach computation. Simultaneously, machine learning, a field that enables computers to learn from data and make intelligent decisions, has seen widespread adoption, dramatically impacting multiple sectors. The fusion of these two disciplines – quantum computing and machine learning – introduces a new paradigm known as quantum machine learning, promising breakthroughs in computational speed and efficiency. This study endeavors to explore this dynamic junction, commencing with an in-depth overview of quantum computation, its basic concepts, and intrinsic nuances. This is juxtaposed against the conventional landscape of machine learning, with particular attention to the principles of Support Vector Machines (SVMs) and Generative Adversarial Neural Networks (GANs). Next, we transition into the realm of quantum machine learning, outlining variational quantum algorithms, the hybrid quantum-classical framework, and the art of data representation on a quantum computer, namely quantum encoding. This comprehensive introduction culminates in presenting the research questions that drive this thesis and the significant contributions it aims to provide. The journey of exploring the intersections of quantum and classical worlds and unveiling novel approaches that harness the power of these realms awaits.

1.1 Quantum Computing

Quantum computing embodies a radical departure from traditional computational paradigms, harnessing the principles of quantum mechanics for processing information. In contrast to classical physics, quantum mechanics employs a sophisticated mathematical framework for predicting measurement probabilities on quantum systems. These quantum systems consist of microscale entities like atoms and photons, exhibiting behaviors inconceivable within the confines of classical Newtonian mechanics, such as superposition and entanglement.

At the core of quantum computing is the quantum bit, or 'qubit'. Diverging from a classical bit, which can only occupy the state of either 0 or 1, a qubit can exist in a superposition state. This state is parameterized by complex numbers α and β, which describe the quantum state's amplitudes. Within the mathematical realm of the Hilbert space, a quantum state can be depicted as a linear combination of two orthogonal basis states. These basis states, often represented by the vectors $|0\rangle = (1,0)^{\mathrm{T}}$ and $|1\rangle = (0,1)^{\mathrm{T}}$, encompass the qubit's two-dimensional Hilbert space. Importantly, the Dirac notation introduces 'kets' $|\cdot\rangle$ and 'bras' $\langle\cdot|$ to represent quantum states and their conjugate transposes, respectively.

The complex amplitudes, α and β, at the heart of defining a quantum state, are more than mathematical symbols; they provide a lens to comprehend quantum phenomena in probabilistic terms. Adhering to the equation $|\alpha|^2+|\beta|^2= 1$, these amplitudes ensure the normalization of a quantum state. The importance of this condition lies in its guarantee that the probabilities of all possible measurement outcomes sum to 1, reinforcing the inherent certainty of obtaining one of these outcomes upon measurement. In essence, these amplitudes serve as the connecting link between a quantum state's abstract mathematical representation and its tangible physical reality.

Building on this abstract mathematical representation, we can visualize a qubit's state within the realm of quantum computing through a Bloch sphere. Named by physicist Felix Bloch, this three-dimensional model assigns the two basis states of a qubit, $|0\rangle$ and $|1\rangle$, to its north and south poles, respectively (cf. Fig. 1.1). Within this sphere, any point, guided by the amplitudes α and β and their respective phases, symbolizes a valid quantum state.

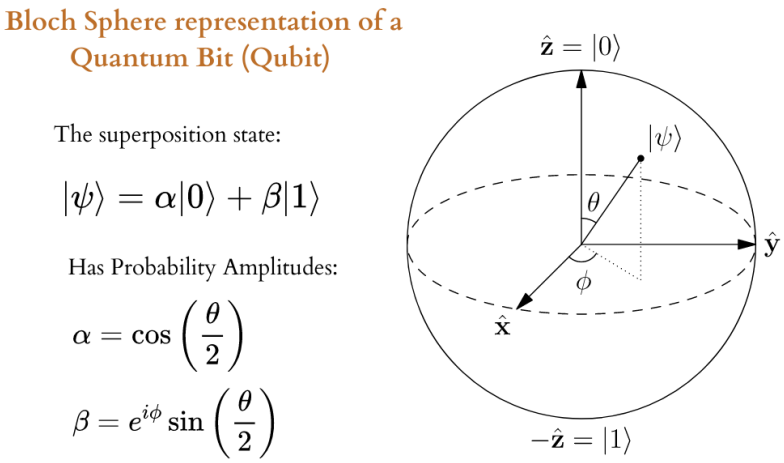


Figure 1.1: Illustration of a Bloch Sphere. This sphere is a geometric representation of a qubit, the fundamental unit of quantum information. Each point within or on the surface of the sphere represents a valid quantum state, with the north and south poles corresponding to the basis states $|0\rangle$ and $|1\rangle$. The associated amplitudes and their phases determine a quantum state's location on or within the sphere. Pure quantum states lie on the surface of the sphere, while mixed states, a statistical mixture of pure states, are situated within the sphere.

While a pure quantum state resides on the sphere's surface, a mixed state, a statistical combination of pure states, finds its place within the sphere. Therefore, the Bloch sphere transcends its role as a mere model to become a compelling tool for intuitively visualizing quantum gates' impact, along with the phenomena of superposition and entanglement, on a qubit's state.

Having this visual representation as a foundation, we can dive into the cornerstone of quantum mechanics: the measurement process. This process refers to the extraction of information from a quantum state. Notably, commercial quantum computers often utilize a type of measurement known as projective or von Neumann measurement. Such a measurement deterministically collapses a quantum state into one of its basis states. For a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, this process yields either of the classical bits 0 and 1, with probabilities dictated by the following relation:

$$p(0) = |\alpha|^2, \quad p(1) = |\beta|^2 \tag{1.1}$$

Equation 1.1 outlines the statistical likelihood, $p(0)$ and $p(1)$, of obtaining each classical outcome upon measuring the quantum state $|\psi\rangle$.

We can consider how these states interact within different quantum computational models when transitioning from individual quantum states. Quantum computing methodologies typically adopt one of the three prevalent models: Quantum Annealing, Gate Model Quantum Computing, or Topological Quantum Computing. Quantum Annealing employs quantum fluctuations to address optimization problems, while the Gate Model, which this dissertation focuses on, uses unitary operations (quantum gates) to manipulate qubits. Topological Quantum Computing, though highly theoretical at this stage, aims to exploit topological states of matter to provide robustness against decoherence.

In the Gate Model of Quantum Computing, quantum gates serve as the fundamental building blocks for creating quantum circuits and performing operations on qubits. This model's power emanates from its ability to leverage quantum phenomena like superposition and entanglement. Superposition permits a qubit to exist in multiple states simultaneously. At the same time, entanglement allows groups of qubits to become interlinked in such a

way that the state of one qubit immediately impacts the state of the others, irrespective of the physical distance between them, as illustrated in the Einstein-Podolsky-Rosen (EPR) paradox.

The inherent capacity of quantum computing to harness superposition and entanglement presents considerable computational gains over classical systems. Its potential extends to diverse application areas, from factoring large integers and sifting through unsorted databases to simulating complex molecular interactions and streamlining systems optimization. The quantum computing revolution has the potential to permeate a wide array of sectors, redefining the way we approach complex problems. By exploring the landscape of quantum computing, we begin to uncover the immense potential of this transformative technology, ready to shape our digital future.

As we delve deeper into this fascinating quantum computing realm, we venture into a parallel discipline that significantly complements it - Machine Learning. Machine Learning, a subset of Artificial Intelligence, capitalizes on data-driven insights to facilitate decision-making and predictions. It is a dynamic field continuously evolving and adapting to new data and scenarios. The following section will elucidate the fundamental concepts of Machine Learning and how the power of quantum computing can be harnessed to elevate its performance and capabilities. Thus, our journey into the synergy of quantum computing and machine learning begins.

1.2 Machine Learning

Machine learning, a subset of artificial intelligence, leverages the power of algorithms to identify patterns within vast data sets, enabling them to make informed predictions about unfamiliar situations. In essence, machine learning transcends the boundaries of traditional artificial intelligence, embodying a nuanced approach to pattern recognition. Humans are innately skilled at recognizing patterns, whether it's interpreting emotions from written text or facial expressions, identifying weather patterns, or diagnosing diseases. These abilities stem from years of observing key indicators and their correlation with future events,

sometimes requiring targeted experimentation. However, the sheer volume of data a human can process remains inherently finite.

The allure of machine learning resides in its proficiency to utilize data sets of staggering scales, exceeding the processing capabilities of individual humans. With an innate knack for managing and scrutinizing vast quantities of data, computers emerge as ideal candidates for pattern extraction, relationship identification, and predictive modeling. Harnessing machine learning techniques, computers can learn from colossal data sets, unveil hidden patterns, and make accurate predictions across diverse fields. This ability to learn from voluminous data sets propels machine learning as a critical tool for addressing intricate problems and extracting insights that might elude human observers.

While machine learning shares certain aspects with conventional statistics, it primarily differentiates itself by the nature of the data it engages with. The patterns machine learning seeks to learn are often exceedingly complex, and our grasp of the underlying dynamics is frequently limited. Traditional statistical methods strive to construct models based on known equations that encapsulate the data generation process. In contrast, machine learning adopts an alternate approach. It commences with a broad, agnostic mathematical model that is then tailored to the specific case using data through a process known as "training."

Scrutinizing the final trained model in machine learning may not offer insights into the physical mechanisms or statistical relationships underpinning the data. Instead, we interpret the model as a black box that has acquired the capability to deliver reliable outputs based on its training data. In this context, the term "model" in machine learning diverges from its usage in physics and statistics, where models are carefully constructed to encapsulate and examine the nature of a mechanism.

Despite its myriad advantages, machine learning does encounter criticisms, predominantly for its lack of transparency. The intricacies of the models can be challenging to interpret or validate. Nevertheless, this opacity also constitutes machine learning's strength, facilitating the models' generic application across various tasks and domains. The adaptability and versatility of machine learning models render them broadly applicable, even in scenarios where the underpinning mechanisms are not fully comprehended. This feature ensures machine learning's crucial role in driving future technological progress.

As we delve deeper into the realm of machine learning, two methodologies merit special attention due to their proven efficacy across numerous applications: Support Vector Machines (SVMs) and Generative Adversarial Networks (GANs). These techniques embody distinct approaches to learning from data, each with its own strengths, challenges, and ideal use cases. SVMs, a key pillar of supervised learning, focus on devising optimal boundaries to separate different categories within a data set. On the other hand, GANs, a paradigm of unsupervised learning, aim to generate new data samples that resemble a given training set. The combined study of SVMs and GANs offers an enlightening exploration of how machines can learn from data. The upcoming sections will detail these methods, illuminating their theoretical foundations and operational mechanisms.

1.2.1 Support Vector Machines

Support Vector Machines (SVMs) [26] occupy a significant position in the realm of supervised learning algorithms in machine learning, catering to both classification and regression analyses. They're constructed upon robust statistical learning frameworks and regarded highly for their predictive prowess. An SVM utilizes a set of training examples labeled into two categories. The model then applies a training algorithm to discern these categories, generating a non-probabilistic binary linear classifier. SVMs cleverly maximize the gap between the categories within a mapped training example space, thereby permitting the categorization of new instances based on their position relative to this gap. They are not restricted to linear classifications; with the kernel trick, they can effectively manage non-linear classifications by implicitly mapping inputs into high-dimensional feature spaces, thus accommodating complex decision boundaries.

In the realm of SVMs, the goal is to categorize data points into one of two classes. Every data point appears as a p-dimensional vector, where 'p' represents the number of attributes or features. The challenge lies in discerning if these data points can be differentiated by a (p-1)-dimensional hyperplane, referred to as a linear classifier. Among the prospective hyperplanes capable of segregating the data, the one maximizing the distance, or margin, between the two classes, is considered optimal. This hyperplane, termed as the maximum-margin hyperplane, defines a maximum-margin classifier or an optimally stable perceptron.

SVMs formalize the construction of one or more hyperplanes in a space that can span high dimensions or even reach infinity. These hyperplanes are utilized for classification, regression, or other tasks like anomaly detection. The SVM endeavors to find a hyperplane that facilitates a comprehensive separation by maximizing the distance to the nearest training data point of any class, referred to as the functional margin. A larger margin reduces the classifier's generalization error, leading to enhanced performance.

In scenarios where the data points aren't linearly separable in the original finite-dimensional space, SVMs employ the kernel trick, which maps the data into a higher-dimensional space where the separation process becomes simpler. The choice of the kernel function, represented as $K(x, y)$, is instrumental in defining the hyperplanes in the higher-dimensional space. The kernel function facilitates the computation of dot products between input data vectors without explicitly mapping them into the higher-dimensional space. By choosing an apt kernel, the SVM can efficiently manage complex decision boundaries and nonlinearly separable data.

The hyperplanes within the higher-dimensional space can be expressed as a set of points whose dot product with a vector in that space remains consistent. These vectors, which define the hyperplanes, can be articulated as linear combinations of the feature vectors' images from the original data, multiplied by parameters α_i. The hyperplane can then be defined using the relation $\sum_i \alpha_i K(x_i, x) = \text{constant}$, where x_i represents the feature vectors in the data. The sum of kernels in this equation is used to gauge the relative nearness of each test point x to the data points belonging to the different classes. This approach empowers SVMs to distinguish between sets that are not convex in the original space, enabling more intricate decision boundaries.

In conclusion, SVMs, by exploiting the concept of hyperplanes and mapping techniques, achieve effective classification in both linearly separable and nonlinearly separable situations, marking them as potent tools in machine learning.

Subsequently, we elucidate the formulation of the SVM algorithm, renowned for its binary classification abilities. It primarily targets the discovery of the maximum-margin hyperplane that separates two classes of data points. This algorithm can be extended to handle non-linearly separable data by using the kernel trick.

The SVM classifier is determined by the support vectors, which are the data points that lie nearest to the maximum-margin hyperplane. Non-linearly separable data is handled using the kernel trick, which substitutes the dot product between input vectors with a non-linear kernel function. Commonly used kernels include polynomial kernels and Gaussian radial basis function (RBF) kernels.

Suppose we desire to learn a nonlinear classification rule equivalent to a linear classification rule for the transformed data points $\varphi(\mathbf{x}_i)$. Moreover, we have a kernel function k that satisfies $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$. The classification vector $\mathbf{w}$ in the transformed space satisfies:

$$\mathbf{w} = \sum_{i=1}^{n} c_i y_i \varphi(\mathbf{x}_i),$$

The c_i values are determined by solving the optimization problem:

$$\begin{aligned} &\text{maximize } f(c_1, \ldots, c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i k(\mathbf{x}_i, \mathbf{x}_j) y_j c_j \\ &\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i. \end{aligned}$$

The coefficients c_i can be determined using quadratic programming. We can find an index i such that $0 < c_i < (2n\lambda)^{-1}$, indicating that $\varphi(\mathbf{x}_i)$ lies on the boundary of the margin in the transformed space, and then solve:

$$b = \mathbf{w}^T \varphi(\mathbf{x}_i) - y_i = \left[\sum_{j=1}^{n} c_j y_j \varphi(\mathbf{x}_j) \cdot \varphi(\mathbf{x}_i) \right] - y_i = \left[\sum_{j=1}^{n} c_j y_j k(\mathbf{x}_j, \mathbf{x}_i) \right] - y_i.$$

Finally, the classification rule is:

$$\mathbf{z} \mapsto \operatorname{sgn}(\mathbf{w}^T \varphi(\mathbf{z}) - b) = \operatorname{sgn} \left(\left[\sum_{i=1}^{n} c_i y_i k(\mathbf{x}_i, \mathbf{z}) \right] - b \right).$$

Having discussed the fundamentals of Support Vector Machines (SVMs) and their applications in linearly separable scenarios, we now focus on non-linear classification. When

a linear hyperplane in the original space cannot separate data points, SVMs employ the kernel trick to transform the data into a higher-dimensional space. This transformation allows for the discovery of complex decision boundaries. We need to find optimal coefficients for the transformed data points to effectively utilize the kernel trick. This section presents a pseudocode implementation of the SVM training process, demonstrating how the coefficients are obtained and the classification rule is established in the transformed space.

Algorithm 1 Training a SVM: Pseudocode illustrating the training process of a Support Vector Machine (SVM). The algorithm takes as input a training set comprising n data points, each represented by a p-dimensional vector $\mathbf{x}_i$ and labeled as $y_i \in \{-1, 1\}$. The regularization parameter λ controls the trade-off between maximizing the margin and minimizing training errors. The algorithm iteratively updates the weight vector $\mathbf{w}$ and the offset b to minimize the loss function, which consists of a regularization term and the hinge loss for misclassified points. The algorithm converges when $\mathbf{w}$ and b reach optimal values or a maximum number of iterations is reached. The algorithm's output is the optimized weight vector $\mathbf{w}$ and the offset b, which define the SVM classifier.

Require: Training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$
Require: Regularization parameter $\lambda > 0$
Ensure: Optimal values for $\mathbf{w}$ and b
Initialize $\mathbf{w}$ and b to some initial values
repeat
    for each i in $1 \dots n$ **do**
        Set $\text{loss}_i = \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b))$
    end for
    Set loss $= \lambda||\mathbf{w}||^2 + \sum_{i=1}^n \text{loss}_i$
    Update $\mathbf{w}$ and b to minimize loss
until $\mathbf{w}$ and b converge or maximum iterations reached
return $\mathbf{w}$, b

1.2.2 Generative Adversarial Neural Networks

Since their inception in 2014, courtesy of Ian Goodfellow and his team [22], Generative Adversarial Networks (GANs) have constituted a robust machine learning framework for the generation of data, mirroring the patterns and statistical nuances of their training set. Over time, GANs have commanded significant attention and achieved impressive strides across a multitude of domains.

The principle underpinning GANs is the antagonistic dynamic between two neural networks: the generator and the discriminator. Together, they partake in a zero-sum game - the generator attempts to forge synthetic data indistinguishable from the real counterpart, while the discriminator endeavors to differentiate between genuine and fabricated samples accurately. This adversarial engagement cultivates the proficiency of GANs in creating realistic, diverse samples that echo the statistical traits of their training data.

Originally proposed within the context of unsupervised learning, the scope of GANs has since broadened to other learning modalities, including semi-supervised, fully supervised, and reinforcement learning. Their versatility is demonstrated by their effectiveness in various tasks, from image synthesis and text generation to music composition and video generation.

The iterative training process intrinsic to GANs involves regular updates to both the generator and discriminator networks. The generator, receiving random noise as input, morphs it into synthetic data samples. Subsequently, the discriminator evaluates the authenticity of these samples, striving to distinguish them accurately from real data. Based on the discriminator's feedback, both networks are updated - the generator seeks to enhance its deception capabilities, while the discriminator aims to improve its accuracy in distinguishing genuine and generated data.

In contrast to conventional models which aim to minimize a predefined loss function to model a specific image, GANs adopt a distinct training approach. Their objective doesn't solely focus on minimizing a specific loss function, but instead, instigates a competitive dynamic between the generator and discriminator. By urging the generator to deceive the discriminator, GANs tap into the potential of unsupervised learning, enabling the model to capture intricate patterns and structures within the training data. This adversarial training process equips the generator to understand the underlying data distribution and generate new samples reminiscent of the training data.

For a comprehensive understanding of the GAN training algorithm, we provide a pseudocode representation, delineating the steps involved in training a GAN model. This pseudocode elucidates the entire training procedure - from initializing the networks and sampling real and noise data, to training the discriminator, and finally updating the generator

using the discriminator's gradients to maximize the log probability of the discriminator being deceived by the generated data.

GANs' remarkable prowess has had transformative impacts on computer vision, natural language processing, and numerous other domains. They have paved the way for innovative possibilities in creativity, data augmentation, and data generation, propelling advancements in a variety of applications and forging new research paths.

Algorithm 2 Pseudocode for the Generative Adversarial Network (GAN) training process. This algorithm accepts the training data, noise dimension, and the number of training iterations as inputs. It initiates the generator and discriminator networks with random weights and executes the specified training iterations. Within each iteration, mini-batches of real data and noise vectors are sampled. The generator fabricates synthetic data samples using the noise vectors, and both real and generated data serve as training material for the discriminator. The generator is updated using the gradients of the discriminator to maximize the log probability of the discriminator being deceived by the generated data, enabling the generator to enhance its proficiency in generating realistic samples that mirror the training data distribution.

Require: Training data X, noise dimension Z, number of training iterations N
Output: Trained generator G
Initialize the generator network G with random weights
Initialize the discriminator network D with random weights
for $n = 1$ to N **do**
    Sample a minibatch of m examples from the training data X
    Sample a minibatch of m noise vectors z from a noise distribution
    Generate synthetic data samples $G(z)$ using the generator network
    Concatenate the real data X and the generated data $G(z)$
    Label the real data with 1 and the generated data with 0
    Train the discriminator D using the concatenated data and corresponding labels
    Sample a new minibatch of noise vectors z from the noise distribution
    Update the generator G by maximizing $\log(D(G(z)))$ using the gradients of D with respect to G
end for

This pseudocode provides a simplified portrayal of the GAN training process. Practical implementations warrant additional considerations such as learning rate schedules, optimization techniques, and network architectures. This code serves as a versatile foundation that can be tailored and expanded based on the specific nuances of any GAN implementation.

As we delve deeper into the world of GANs, it becomes clear that they provide a stepping stone towards another intriguing frontier: quantum machine learning. In the upcoming

section, we will explore how the principles of GANs extend into the quantum realm and the unique advantages that quantum computing can bring to machine learning.

1.3 Quantum Machine Learning

The integration of quantum technologies into the market has given rise to the era of "near-term" quantum computing, which has significantly impacted the field of quantum computing research. It has moved beyond the confines of academia and has become a focus in the research labs of numerous startups and large IT companies. As more computer scientists and engineers join the quantum computing community, there is a growing availability of software toolboxes and quantum programming languages based on classical computational languages, with ongoing development in this area. The realization of quantum technology has become an international, interdisciplinary, and primarily commercial endeavor.

Considerable progress has been made in the development of Noisy Intermediate-Scale Quantum (NISQ) devices, which serve as the initial prototypes of future full-scale quantum computers. These devices typically consist of around 10-100 qubits that may not all interact with each other. As they lack error correction, they can only provide approximate results for computations. Despite their limitations, NISQ devices have the potential to demonstrate the advantages of quantum computing. However, the need to limit algorithms to a small number of qubits and gates significantly influences the design of quantum algorithms.

The ultimate goal is to identify useful computational problems that can be effectively solved by small-scale quantum devices while exhibiting exponential and provable speedups compared to classical algorithms. This pursuit has led to a race to discover what is often referred to as a "killer-app" – a compact yet powerful algorithm specifically tailored for early quantum technologies. Machine learning and optimization, which is a core mathematical problem in machine learning, are frequently mentioned as promising candidates. This may explain the substantial momentum observed in quantum machine learning research in recent years.

Machine learning, which lies at the intersection of statistics, mathematics, and computer science, focuses on teaching computers to learn from prior examples and make predictions

or solve unseen problem instances based on complex and nonlinear relationships present in large datasets. While tasks like image recognition and language processing come naturally to humans, replicating these abilities in machines poses challenges. Machine learning tackles this by enabling computers to extract patterns from data, which inherently encode concepts such as "mountain range" in the case of an image of a mountain panorama.

Machine learning has transitioned into an industry-driven research discipline at a larger scale than quantum computing, largely due to the success of deep learning. Deep learning involves stacking modular algorithms called neural networks to build large architectures that are trained using massive amounts of data and high-performance computing systems. Consequently, machine learning is often seen as the "dark art" of engineering with lucrative salaries.

However, machine learning also has an academic and theory-based aspect. The rise of deep learning has raised the intriguing question of why it works effectively. Traditional learning theory, rooted in statistics, suggested a trade-off between model expressivity and learning power. Yet, modern machine learning neural networks are highly expressive with millions or billions of trainable parameters, and evidence shows that larger models tend to perform better. Understanding why these heavily overparametrized models do not overfit the data they see poses an exciting challenge.

In recent years, significant efforts have been made to develop theories that shed light on this question. While the complete picture is still emerging, it is increasingly evident that a good machine learning method relies on the interplay of various components, such as the data, the model, and the optimization algorithm. These developments in machine learning research create a dynamic landscape for quantum machine learning, offering both excitement and challenges for researchers navigating this field.

The field has matured significantly since its early fragmented state, and there is now a growing body of literature and research that explores the intersection of quantum computing and machine learning.

The emergence of the term "quantum machine learning" gained traction around 2013, with notable contributions and publications by researchers such as Lloyd, Mohseni, Rebentrost, and Wittek. Wittek's monograph [53] titled "Quantum Machine Learning -

What quantum computing means to data mining" played a crucial role in shaping and advancing the field. Subsequently, there has been a substantial increase in interest and research activity, leading to the formation of a vibrant community. International workshops, conferences, special issues, and academic journals have emerged to foster collaboration and knowledge exchange.

The combination of a lucrative market and the potential benefits offered by quantum computing has attracted significant attention from industry. Companies and startups in the quantum computing space actively engage in quantum machine learning research, contributing to its development and practical applications. Open-source software frameworks like PennyLane [6], TensorFlow Quantum [7], Yao [8], and Qiskit [5] have been established to facilitate quantum machine learning experiments and implementations.

Moreover, the influence of students and young researchers who possess coding skills and an interest in machine learning has shaped the academic landscape. Their involvement and choice of machine learning-related research projects have contributed to the rapid growth of quantum machine learning as a distinct discipline.

At present, quantum machine learning is a well-established and active sub-discipline within quantum computing research. It encompasses various sub-areas and benefits from dedicated research teams, quantum computing companies, open-source frameworks, and a growing body of knowledge.

In the context of quantum machine learning, the data used for analysis typically comes from classical systems, such as text, images, or time series of macroeconomic variables. These classical data are then input into a quantum computer for processing and analysis. However, this introduces a challenge known as the quantum-classical interface.

The quantum-classical interface refers to the interaction between classical data and quantum computing systems. In order to analyze classical data using a quantum computer, the data needs to be loaded onto the quantum system. This process involves transferring the data from classical memory to the quantum computer or a quantum-readable memory.

One important consideration in this setting is that the runtime of quantum algorithms is not logarithmic in the number or size of the data samples. The act of loading the data from classical memory onto a quantum computer takes linear time, as each data point needs

to be read in and transferred individually. This linear time requirement is inherent to the quantum-classical interface and is independent of the potential speedup offered by quantum algorithms.

In summary, the quantum-classical interface poses a challenge in quantum machine learning as classical data needs to be loaded onto a quantum computer for analysis, which incurs a linear time cost. This is an important aspect to be aware of when designing and analyzing quantum machine learning algorithms.

1.3.1 Variational quantum algorithms

In recent years, the field of quantum machine learning (QML) has gained significant attention and is being explored as a potential avenue to enhance machine learning algorithms using quantum computing capabilities. QML combines concepts from both quantum computing and machine learning to develop new algorithms and models that leverage quantum properties for improved performance.

The development of QML has been motivated by the limitations and challenges faced in traditional machine learning. While empirical approaches have been successful in driving progress, there is still a lack of satisfactory theoretical explanations for many observed phenomena in deep learning. Additionally, the computational complexity of machine learning algorithms does not always provide meaningful insights, particularly for computationally hard problems like non-convex optimization in neural network training.

These challenges apply to both fault-tolerant quantum computing and near-term quantum computing approaches to machine learning. Fault-tolerant quantum computers rely on meticulously developed computational complexity analysis, which may have limited utility if existing classical algorithms can solve problems heuristically with linear runtimes. On the other hand, near-term quantum computing offers a more practical approach to QML but comes with design limitations, such as the need for lean algorithms that can operate on a few qubits and gates due to the limitations of current hardware.

To address these challenges, researchers have adopted two design decisions in QML. Firstly, hybrid quantum-classical algorithms have been proposed, where quantum computations are integrated into more complex classical pipelines. In this approach, the machine

learning model is implemented as a quantum algorithm, while the training process is performed on a classical co-processor. This hybrid approach allows for leveraging the power of quantum algorithms while still utilizing classical resources for training and optimization.

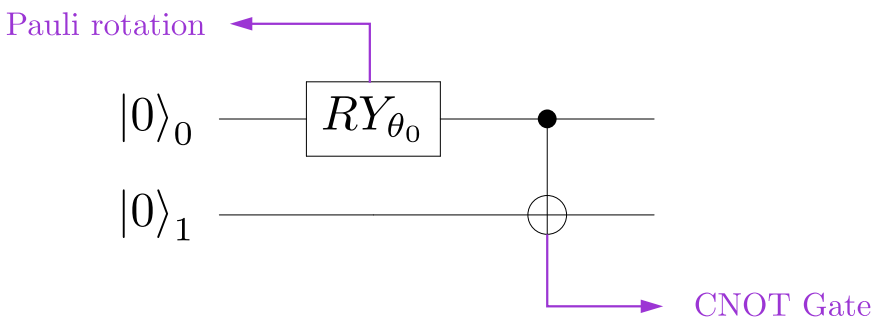
Secondly, researchers have explored the use of hardware-tailored circuits or "quantum models" as alternatives to classical models like neural networks or Gaussian processes. Variational circuits, also known as parametrized circuits or quantum neural networks, have emerged as a promising choice for QML. Variational circuits consist of parametrized and fixed quantum gates arranged in an ansatz or template, similar to the layer structure of a neural network. The parameters of the circuit can be optimized by minimizing a cost function, adapting the gates and circuit to the specific problem at hand. This optimization process is carried out through a classical feedback loop that evaluates the model or queries properties of the model landscape, such as gradients, to iteratively improve the circuit parameters.

The concept of variational circuits in QML has been introduced in various papers, including Refs [21, 11]. Although the idea was proposed earlier, as seen in Ref [10], the recent focus on variational circuits has driven significant advancements in the field.

It's important to note that the introduction of variational circuits has shifted the research objective of QML. Instead of solely aiming for speedups of known classical models, QML now explores a new family of models with unknown potential benefits. This change in perspective has raised new research questions about the nature of quantum circuits as machine learning models, their similarities to classical models like neural networks and support vector machines, the utility of quantum properties such as superposition and entanglement, practical training techniques, and their generalization capabilities.

Overall, QML represents a promising avenue for exploring the intersection of quantum computing and machine learning, providing new opportunities to address the challenges and limitations of classical machine learning algorithms.

Variational quantum algorithms are represented by parameterized quantum circuits as illustrated in figure 1.2 . The parameters of the quantum circuits are usually optimized on a classical processing unit.



Dirac Notation:

$\text{CNOT}((RY_{\theta_0}(|0\rangle_0) \otimes |0\rangle_1)$

Vector-matrix Notation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \left(\begin{bmatrix} \cos\frac{\theta_0}{2} & -\sin\frac{\theta_0}{2} \\ \sin\frac{\theta_0}{2} & \cos\frac{\theta_0}{2} \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times$$

$$\left(\begin{bmatrix} \cos\frac{\theta_0}{2} \\ \sin\frac{\theta_0}{2} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} \cos\frac{\theta_0}{2} \\ 0 \\ \sin\frac{\theta_0}{2} \\ 0 \end{bmatrix}$$

Figure 1.2: A parameterized quantum circuit, with 2 types of operations: a 1-qubit parameterized gate and a 2-qubit non parametric gate

1.3.2 Hybrid quantum classical framework

Hybrid quantum-classical algorithms leverage the synergy between quantum state preparation and measurement with classical optimization techniques, as illustrated in figure 1.3. The primary objective of these algorithms is typically to find the ground state eigenvector and eigenvalue of a Hermitian operator.

In a hybrid quantum-classical algorithm, the quantum component is responsible for preparing and manipulating quantum states. This involves initializing the quantum system in a known state, applying quantum gates and operations to the state, and finally measuring the resulting quantum state. The measurements provide classical information that is used as input for the classical optimization process.

The classical optimization component plays a crucial role in refining the quantum states obtained from measurements. It employs classical optimization techniques, such as gradient-based methods or evolutionary algorithms, to iteratively update the parameters of the quantum system in order to minimize a cost function associated with the ground state eigenvector and eigenvalue estimation.

By combining the strengths of quantum state manipulation and classical optimization, hybrid quantum-classical algorithms provide a powerful approach for tackling problems related to ground state estimation. These algorithms have found applications in various domains, including quantum chemistry, optimization, and machine learning, where determining the ground state properties of a quantum system is essential for solving complex problems.

1.3.3 Representing data on a quantum computer

In the application of quantum computers to extract insights from classical data, a critical aspect to consider is the transformation of this data into quantum states. This process is analogous to the preparation of the initial quantum state in traditional quantum computing, which forms the basis for input encoding for any given algorithm. However, in the context of quantum machine learning, the role of data encoding becomes even more significant, transcending beyond just state preparation.

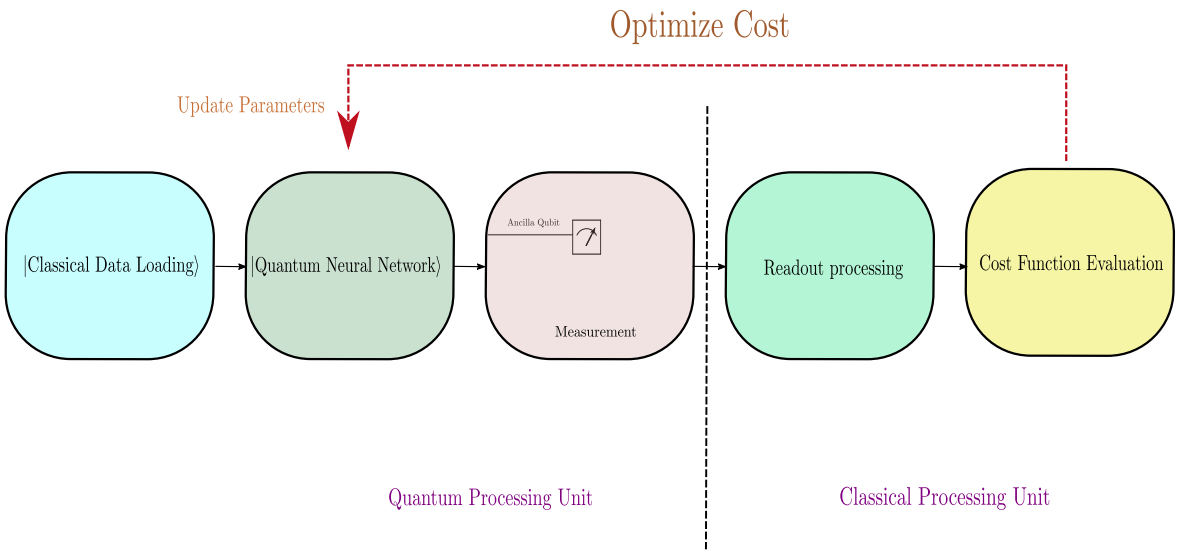


Figure 1.3: The hybrid quantum classical framework in quantum computing an quantum machine learning

The importance of data encoding in quantum machine learning is not confined merely to the initialization of the algorithm; it critically influences the algorithm's runtime. This is exemplified in the case of the Hadamard classifier, where the most time-consuming aspect was the data encoding process. This was followed by a single application of a quantum gate, and subsequently a straightforward measurement. Consequently, the platforms, software, and hardware mediating the transition from classical memory to quantum devices play a critical role in assessing the runtime of quantum machine learning algorithms. Moreover, as quantum machine learning algorithms often yield probabilistic outputs, the entire procedure - encompassing state preparation - may necessitate repetition several times.

Beyond pragmatic considerations, data encoding in quantum machine learning holds crucial conceptual implications. From a theoretical standpoint, data encoding can be perceived as a feature map that transposes input data to the Hilbert space of the quantum system. This is reminiscent of feature maps employed in classical machine learning's kernel methods. Feature maps define a similarity measure that informs the decision-making processes in learning algorithms. By judiciously choosing a feature map or distance metric, one can demarcate data from different classes in the feature space, thereby effectively addressing the learning problem.

In summary, data encoding in quantum machine learning is indispensable both for practical assessments of runtime and conceptual algorithm design. It not only impacts algorithm efficiency but also forms a bridge to classical machine learning methods, such as kernel theory. This is done by leveraging feature maps to project data into the quantum system's Hilbert space, thereby reinforcing the connective tissue between quantum and classical computational paradigms.

1.4 Research Questions and Contributions

Amid the intersection of quantum machine learning and the inherent limitations of Noisy Intermediate-Scale Quantum (NISQ) devices, this study endeavors to design and scrutinize quantum algorithms and procedures specifically tailored for these machines. The investigation is bifurcated into two core components: a resource-efficient, low-depth quantum

circuit Generative Adversarial Network (GAN) and an optimized, low-depth data encoding process.

The first cornerstone of this research revolves around the creation of a quantum circuit GAN. As an acclaimed generative model in the traditional machine learning ecosystem, the ambition is to devise a quantum analogue that requires minimum qubit resources and circuit depth. This is a critical objective, considering the constraints of contemporary NISQ devices - limited qubit connectivity, coherence time, and the susceptibility to introducing noise and errors over extended circuit lengths. Consequently, the design of a quantum circuit GAN that adheres to these restrictions is anticipated to unlock practical and efficient generative modeling possibilities on NISQ devices.

Transitioning to the second pillar of this study, the focus shifts towards the data encoding procedure. As emphasized earlier, data encoding is a vital cog in the wheel of quantum machine learning algorithms. Here, the aspiration is to create a data encoding procedure that is custom-fit for NISQ devices. This equates to devising an encoding scheme that optimally uses resources - minimal qubit numbers and encoding circuit depth. By fine-tuning the data encoding procedure for NISQ devices, the aim is to alleviate the computational load and enhance the overall efficiency of quantum machine learning algorithms on these devices.

Conjoining the resource-conservative, low-depth quantum circuit GAN with the streamlined data encoding procedure, this study intends to grapple with the challenges propagated by NISQ devices and unlock their potential for real-world quantum machine learning applications. The quest is to uncover suitable applications that can effectively harness the abilities of NISQ devices while simultaneously acknowledging their limitations. This quest symbolizes a meaningful stride towards enriching the quantum machine learning domain, setting the stage for the creation of practical and efficient algorithms designed for the quantum hardware of the near future.

Chapter 2

Re-QGAN: an Optimized Adversarial Quantum Circuit Learning Framework

This chapter is adapted from [40]

Adversarial learning represents a powerful technique for generating data statistics. Its successful implementation in quantum computational platforms is not straightforward due to limitations in connectivity, quantum operation fidelity, and limited access to the quantum processor for statistically relevant results. Constraining the number of quantum operations and providing a design with a low compilation cost, we propose a quantum generative adversarial network design that uses real Hilbert spaces as the framework for the generative model and a novel strategy to encode classical information into the quantum framework. We consider quantum generator and discriminator architectures based on a variational quantum circuit. We encode classical information by the stereographic projection, which allows us to use the entire classical domain without normalization procedures. For low-depth ansätze designs, we consider the real Hilbert space as the working space for the quantum adversarial game. This architecture improves state-of-the-art quantum generative adversarial performance while maintaining a shallow-depth quantum circuit and a reduced parameter set. We tested our design in a low resource regime, generating handwritten digits with the

MNIST as the reference dataset. We could generate undetected data (digits) with just 15 epochs working in the real Hilbert space of 2, 3, and 4 qubits. Our design uses native quantum operations established in superconducting-based quantum processors and is compatible with ion-trapped-based architectures.

2.1 Introduction

Quantum Machine Learning [13, 45] has been concerned with applying concepts and tools from quantum computation and quantum information to enhance the treatment and processing of classical machine learning tasks. Several approaches demonstrate a level of advantage when the quantum processor (QPU) is used in classical-quantum hybrid architectures, from the implementation of generative tasks using the QPU as a parameterized sample generator [11] until classification procedures exploiting an entangled measurement scheme in the QPU to enhance the discrimination efficiency [27]. One case is the adversarial networks framework upgrading, known as quantum generative adversarial neural networks (qGANs). This quantum framework uses the QPU to build discriminative and generative models based on parameterized quantum circuits (PQC), where the data is encoded into a quantum state and processed by quantum operations. The adversarial game structure (quantum or classical) requires a classical database known as the real-data, a generative model (generator) that can produce data known as fake-data, and a discriminative model (discriminator) that can measure the difference between real and fake. The generator and discriminator compete; from one side, the generator tends to produce fake-data that can be treated as real-data by the discriminator. On the other side, the discriminator improves the way of distinguishing real- from fake-data.

qGANs were introduced in 2018 in two companion papers [19, 36], presenting a roadmap to build their architecture and sketch their operational structure and advantages over their classical counterpart. Since then, several works have proposed different qGANs architectures in several paradigms and for other applications. In addition, these works have tackled the various problems linked to implementing and training qGANs. For instance, quantum conditional GANs [35] that outperformed their classical counterpart, quantum style-based

GAN (style-qGAN) [16], where the quantum generator model is optimized to generate Monte-Carlo events for high-energy physics applications, quantum dissipative GAN [9] based on dissipative neural networks, entanglement qGAN (EQ-GAN) [41] that entangles target and generated states to achieve the Nash equilibrium in the adversarial game, and qGANs to model classical continuous distributions [44].

Other works treat implementation matters, like strategies using shallow quantum circuits and post-measurement procedures to discriminate the generator's output [52] , studies on the convergence of qGANs and score functions to enhance the convergence time [15], and encoding protocols to efficiently load random distributions into n-qubit quantum states [54]. Most of the proposed qGANs schemes require a non-trivial compilation for their implementation into commercial QPUs, and a resetting according to quantum hardware limitations in connectivity and quantum circuits executions.

One crucial aspect not present in the classical framework is the classical information encoding into quantum information. The adversarial game works in unison with the encoding procedure. Some efforts on this matter cover the effects of data encoding on the expressivity of quantum models [50], and the impact of the encoding on binary quantum classifiers [30], where they evaluate decision boundaries for different encoding strategies under noise influence. A typical procedure is to shrink the classical information into the Hilbert space and return the processed quantum information to the classical world. If the encoding is not one-to-one, we can incur loss and tergiversation. In this work, we address the question of how to design simple qGAN that suit the quantum hardware limitation and at the same time leverage its capabilities, and how we can encode classical information in a quantum state without incurring transformations that yield imprecise information in the de-encoding process.

We contribute to this rich body of work by proposing a novel classical-quantum encoding based on the stereographic projection and a real quantum state qGAN (re-qGAN) that, in general, requires an optimal value of parameters, according to the size of the classical data, and only single-qubit rotations and CXs to be achieved. This design produces less computationally complex quantum neural networks compared to previous architectures. The

real amplitude quantum state preparation, in turn, improves training time and feasibility on near-term quantum devices.

2.2 Results

2.2.1 Quantum Adversarial Game

We propose a generative adversarial quantum circuit learning for an unsupervised learning task. In this work, we test the strategy in image generation using a supervised loss for an unsupervised learning task. We consider the standard structure for quantum generative adversarial algorithms, a real-data agent that encodes the sample of classical information (training sample) into quantum information, a generator that prepares quantum states that eventually will be decoded into classical data, and a discriminator that measures how qualitatively different is the generator's output is from the training dataset. We describe the input data and present the quantum adversarial model in the following.

We consider as input data a collection of images $\mathcal{C} = \{\vec{u}_1, \cdots, \vec{u}_N\}$, that follow an equivalence relation. For simplicity, we convert the images into grayscale. Thus, the input data can be easily formatted into arrays. The vectorized images are compressed by a principal component analysis (PCA) to be projected into the n-qubit real Hilbert (a hypersphere) by the inverse stereographic projection. The stereographic projection allows the use of the projected information, from the real plane where the input data lies to the hypersphere, to define the set of quantum state $\mathcal{C}_Q = \{|u_1\rangle, \cdots, |u_N\rangle\}$ that will represent the classical data. With the state amplitude representation of the classical data, the discriminator's goal is to differentiate this set from any output produced by the generator.

The generator and discriminator agents are designed using parameterized quantum circuits $\mathcal{G}(\grave{})$ and $\mathcal{D}(\grave{})$, respectively, that prepare arbitrary quantum states with real amplitudes. The generator follows a special design that minimizes the number of parameters to the optimal point $2^n - 1$ for full expressivity, given n the number of qubits. On the other hand, the discriminator entangles any pair of quantum states with real amplitudes and uses

Algorithm 3 Quantum adversarial game

Input:

- $\mathcal{C}$: dataset $\{\vec{u}_0, \cdots, \vec{u}_N\}$, with $\vec{u}_k = (u_1, \cdots, u_M)$
- EPOCHS: Number of learning cycles

Initialization:

- SI encoding $\mathcal{C} \to \mathcal{C}_Q$ (Eq. 3.1)
- $n = \lceil \log_2(M) \rceil$, number of qubits
- set $\vec{\theta_D} \in [-\pi, \pi)^{2^{n+1}-1}$ arbitrarily
- set $\{\vec{\theta}_G^k \in [-\pi, \pi)^{2^n-1}\}$ arbitrarily, $k = 1, \cdots, N$
- build $\mathcal{D}^n$ and $\mathcal{G}^n$ (Fig. 2.2(c) and (d))

for $i = 1$ to EPOCHS **do** (Fig. 2.2(b))

$\mathcal{D}^n$ training: $\vec{\theta}_D \leftarrow \arg\min_{\theta_D} \mathcal{L}_D$ (Eq. 2.2)

$\mathcal{G}^n$ training: $\{\vec{\theta}_G^k\} \leftarrow \arg\min_{\{\vec{\theta}_G^k\}} \mathcal{L}_G$ (Eq. 2.3)

end for

- prepare $\mathcal{Q}_Q = \{|q_0\rangle, \cdots, |q_N\rangle\}$ from $\{\vec{\theta}_G^k\}$ (Eq. 2.7)
- SI de-encoding $\mathcal{Q}_Q \to \mathcal{Q}$ (Eq. 2.5)

Output:

- $\mathcal{Q}$: artificial dataset $\{\vec{q}_0, \cdots, \vec{q}_N\}$

an ancillary qubit whose measurement decides whether an input state is equivalent to the training sample.

The discriminator is calibrated with C_Q tuning the parameters $\vec{\theta}_D$ in $\mathcal{D}$ such that the quantity

$$\sigma(|s\rangle, \vec{\theta}) = \langle \psi_D^s(\vec{\theta})|I \cdots I \cdot Z|\psi_D^s(\vec{\theta})\rangle \tag{2.1}$$

becomes 1 for each $|\psi_D^s(\vec{\theta})\rangle = \mathcal{D}(\vec{\theta})|s\rangle \otimes |0\rangle_{ancilla}$ with $|s\rangle \in \mathcal{C}_Q$, and -1 for every other state $|s\rangle \notin \mathcal{C}_Q$. In concurrency, the generator is updated to prepare quantum states more equivalent to the sampling training set with the goal of being undetected by the discriminator, this procedure defines a set of parameters θ_D. We prepare a sample of fake-data produced by $\mathcal{G}$ defining the set of parameters $\{\vec{\theta}_G^1, \cdots, \vec{\theta}_G^N\}$; for a short notation $\{\vec{\theta}_G^k\}_N$. As a first step, the fake sample is compared with the training dataset to calibrate the discriminator. To capture the generator-discriminator adversarial dynamics, we define the cost functions for the discriminator and generator training as

$$\begin{aligned} \mathcal{L}_D(\vec{\theta}_D; \{\vec{\theta}_G^k\}_N) &= \frac{1}{2N}\sum_{l=1}^{N}\left(\left[1 - \sigma(|u_l\rangle, \vec{\theta}_D)\right]^2\right. \\ &\quad \left. + \left[1 + \sigma(|\psi_G(\vec{\theta}_G^l)\rangle, \vec{\theta}_D)\right]^2\right), \end{aligned} \tag{2.2}$$

$$\mathcal{L}_G(\{\vec{\theta}_G^k\}_N; \vec{\theta_D}) = \frac{1}{N}\sum_{l=1}^{N}\left[1 - \sigma(|\psi_G(\vec{\theta}_G^l)\rangle, \vec{\theta}_D)\right]^2, \tag{2.3}$$

respectively, with $|\psi_G(\vec{\theta}_l)\rangle = \mathcal{G}(\vec{\theta}_l)|0\cdots 0\rangle$ as the lth state prepared by the generator. The minimization of the discriminator cost function, $\mathcal{L}_D$, sets a configuration where the elements of $\mathcal{C}_Q$ and the artificial data from $\mathcal{G}$ are recognized with $\sigma = 1$ and $\sigma = -1$, respectively. The minimization of $\mathcal{L}_G$ gives the parameter configuration for the generator $\mathcal{G}$ that produces $\sigma = 1$. In every epoch cycle, the discriminator is optimized to detect the generator's output, and the generator is optimized to fool the discriminator. Once a stop criterion is reached, either a maximum number of iterations or convergence, the set $\{\vec{\theta}_G^k\}_N$ will generate new data indistinguishable to $\mathcal{C}$ after a SI de-encoding procedure. In Algorithm 3 we present the pseudo-code for the quantum adversarial game. This procedure considers the compressed

information when the PCA is applied as to the input information. In the next section, we shall provide more details about the Discriminator's and Generator's training.

2.2.2 Generating MNIST handwritten digits

Generating handwritten digits using the MNIST dataset constitutes a vast common recognition task and a vital benchmark problem in machine learning. In this case, we considered a subset of the MNIST as the training dataset for our re-qGAN model. We compress the MNIST subset to fit the amount of information we can encode for a given number of qubits, so we test the performance of the qGAN in generating low-resolution images. In this experiment, we consider 25 epochs and a subset of 20 images from the MNIST to generate a new digit. In all the numerical experiments, we consider a reduced number of qubits (from 2 to 4) and low resolution (28×28) black and white images in the training dataset compressed by the principal component analysis (PCA) at a degree of explained variance that depends on the number of qubits used, i.e., for n qubits the number of components n_{PCA} must be less than or equal to $2^n - 1$. On the other hand, the maximal number of components depends on the training dataset size. In Table 2.1, we show relevant details of the numerical experiments. In Figure 2.1(a), we present the numerical experiment results and generated images for different sizes in the quantum representation.

In this experiment, the re-qGAN could generate undetectably (from the discriminator viewpoint) samples at the 15th epoch cycle (top row in Fig. 2.1(b)), where we observe a good trade-off between the optimal values of $\mathcal{L}_D$ and $\mathcal{L}_G$. In Figure 2.1(b) we show the variance of $\mathcal{L}_D$ and $\mathcal{L}_G$, and according to the CMA-ES search mechanism, the generator output is closer to a converged value that the discriminator, a favorable scenario in the adversarial game. The fewer epochs for good quality outputs represent an improvement compared to the results in recent proposals, which generally require around 100 epochs to generate undetectable samples [52]. From this observation, we can infer the ability of re-qGAN to learn from a reference database and its potential implementation on limited access quantum devices.

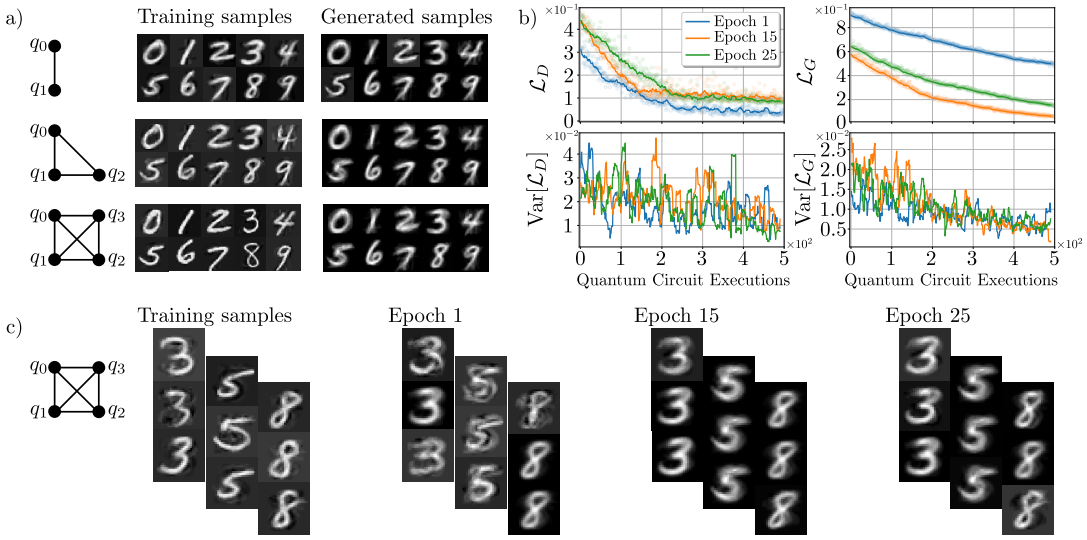


Figure 2.1: MNIST benchmark generation problem. In this Figure, we present the re-qGAN generated digits using a subset of the MNIST as the reference dataset. In a) we present the generated digits for each qubit configuration, we compress the reference dataset according to the number of qubits; see Table 2.1 for the level of compression n_{PCA} and re-qGANs characteristics used. In panel b), the learning for the discriminator and generator at various epochs (Epoch 1, 15, and 25), for four qubits, learning the digit '8'. Each epoch represents one learning cycle, with 500 CMA-ES iterations. In the top row of this panel, we have the evolution concerning the number of CMA-ES iterations of the discriminator and generator's loss functions, $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively. In the bottom row, we have the evolution of the variances of $\mathcal{L}_D$ and $\mathcal{L}_G$ throughout the optimization process. In panel c), we present representative elements of the generated samples by re-qGAN for the digits '3', '5' and '8' using four qubits and 50 training samples per digit at different epochs, namely, epochs 1, 15, and 25. Here, the cumulative explained variance for the PCA transformation is $\sim 75\%$.

Table 2.1: List of parameters used in the numerical experiments. We present the number of parameters (Y_θs) and entangling gates (CXs) used in every case. In addition, we present the number of components used and the cumulative explained variance (CEV). The training dataset is a subset of 20 images from MNIST.

	$\mathcal{G}^n$		$\mathcal{D}^n$		PCA	
n	Y_θs	CXs	Y_θs	CXs	n_{PCA}	CEV
2	3	1	9	4	3	51 %
3	7	4	18	11	7	76 %
4	15	11	35	26	15	96 %

In addition, where we consider a subset of the MNIST as a reference dataset, the generated digit's resolution and quality do not scale with the number of qubits (Fig. 2.1(a) and (c)), which is related to the size of the subset chosen for the learning task. In this case, we found an attractive characteristic to explore, how the subset size works in unison with the number of qubits to generate undetectable data. Looking at the samples generated by re-qGAN for four qubits (Fig. 2.1(c)), we observe a diverse selection of outputs, which indicates that the qGAN is effectively learning the original distribution instead of simply producing random outcomes. We also deal with lower resolution training samples than in Figure 2.1(a) when increasing the training dataset size because of the decreased cumulative explained variance (CEV) in the PCA process. Better resolution outputs would likely require finding a good trade-off between sample size and CEV.

2.3 Methods

2.3.1 Classical dataset encoding

A standard classical-quantum encoding procedure requires mapping an element of the classical data as quantum state amplitude, i.e., we can associate $\vec{u} \in \mathcal{C}$ with an element $\vec{v} = (v_1(\vec{u}), \cdots, v_N(\vec{u}))$, whose components define the amplitudes of a quantum state, therefore $||\vec{v}|| \stackrel{!}{=} 1$. For instance, in the standard amplitude encoding $v_j = u_j/||\vec{u}||$, every single element in the set $\{\alpha\vec{u} : \alpha \in \mathbb{R}\}$ will have the same representation $\vec{v}$, losing information and creating a non one-to-one map between the classical and quantum information. We employed the inverse of the stereographic projection (IS map) to dodge this.

The IS map associates vectors $\vec{u}$ in the real plane $\mathbb{R}^{2^n-1}$ with vectors $\vec{v}$ in $\mathbb{S}^{(2^n-1)} - \{\mathrm{N}\}$ (see Fig. 2.2(b)), using the relationship

$$\begin{aligned} v_i &= \frac{2u_i}{||\vec{u}||^2+1}, \text{ for } i = 1, \cdots, 2^n - 1 \\ v_{2^n} &= \frac{||\vec{u}||^2-1}{||\vec{u}||^2+1}. \end{aligned} \tag{2.4}$$

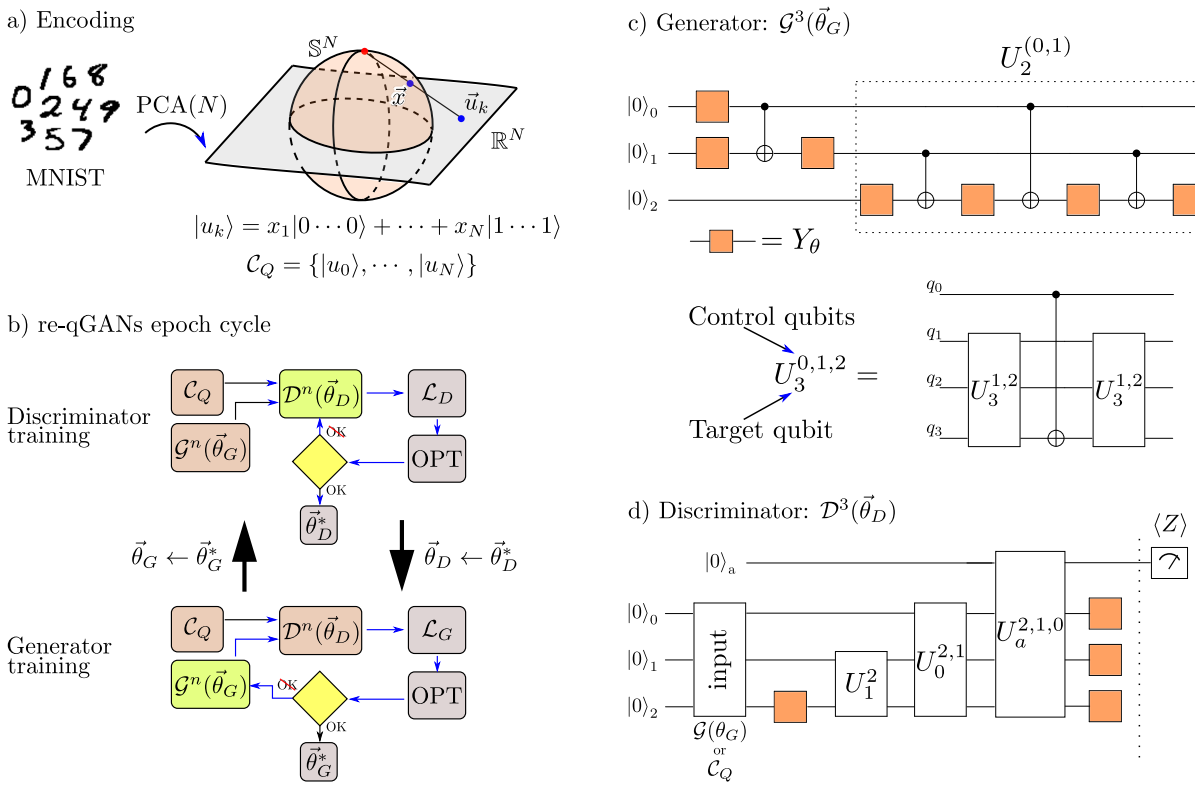


Figure 2.2: Sketch of the quantum adversarial game elements. In a), we present an illustration of the SI encoding of a subset of the MNIST. We compress the images using the principal component analysis (PCA) and project them onto a hypersphere to define the representative quantum state amplitudes. On-hardware implementation requires the preparation of every element of the set $\mathcal{C}_Q$ by using the PQC used for the generator. b) Once we get the real-data set, $\mathcal{C}_Q$, we proceed to the sequential discriminator and generator training. Inside, OPT is the classical optimization algorithm that updates the discriminator or generator parameters. The stop criteria depend on the optimizer choice. In c) we present the generator quantum circuit, the ansatz prepares a 2-qubit state and entangles it with the third qubit through a uniformly controlled rotation (UCR). This PQC can be extended to more qubits by applying a cascade of UCRs. For the 4-qubit case, we need to add the UCR U_3 growing the PQC parameter set in 2^3 more elements. At the bottom, the relation between $U_3^{0,1,2}$ and $U_3^{1,2}$. The recurrence can be used for higher levels of UCRs (see text for more details). d) Three-qubit discriminator ansatz, $\mathcal{G}^3$. The ansatz transforms the generated quantum state or $\mathcal{C}_Q$ to a quantum state in a higher dimensional Hilbert space. The measurement in the ancillary qubit serves as a metric to establish the nature of the input.

With this, we can define a unique quantum state

$$|v\rangle = v_0|0\cdots 00\rangle + v_1|0\cdots 01\rangle + \cdots + v_{2^n-1}|1\cdots 11\rangle \ .$$

The stereographic encoding can accurately represent any vector in the real plane, meaning that encoding is one-to-one. Nevertheless, any variation in the real plane scales with the inverse $||\vec{u}||^{-2}$, i.e., the area is not conserved; we can see this from the relation: $dA_{\text{sphere}} \propto dA_{\text{plane}}/(1+||\vec{u}||^2)$ with dA_{sphere} and dA_{plane} are the area elements of the sphere and the plane, respectively. This feature could make it hard to distinguish the quantum state representation of large vectors, $||\vec{u}|| >> 1$. Without loss of generative power, we scale the training dataset to be close, in magnitude, to a unitary value, keeping $||\vec{u}|| \sim 1$, where $dA_{\text{sphere}} \sim dA_{\text{plane}}$ [20]. In the IS map, we need to establish one scaling factor for all elements in the classical dataset. Once we process the information in the Hilbert space, the answer can be de-encoded back to the real plane by

$$u_i = v_i/(1 - v_{2^n}), \text{ for } i = 1, \cdots, 2^n - 1 \ . \tag{2.5}$$

2.3.2 Quantum state generator and discriminator

We have established the quantum representation of the classical dataset; now, we shall proceed to introduce a parameterized quantum circuit (PQC) able to prepare the elements in C_Q and new quantum states that the discriminator will test.

For the quantum state generator and discriminator, we consider the real amplitude version of the PQC introduced by Möttönen *et al.* [39, 38]. The generator is designed to prepare an n-qubit arbitrary quantum state with the optimal number of parameters, $2^n - 1$. Figure 2.2(b) presents the 3-qubit PQC based on Y_θ rotations and CX gates. The first part of the ansatz prepares an arbitrary two-qubit quantum state. This design was studied in [43] as an optimal preparation. The second part introduces a two-fold uniformly controlled rotation (UCR) $U_2^{0,1}$. In general, the m-fold UCR $U_{q_m}^{q_0,\cdots,q_{m-1}}(\theta_1, \cdots, \theta_{2^m})$ plays a vital role the state preparation. This operator applies the rotation Y_{θ_l} on the target qubit state $|q_m\rangle$ when $|q_0 \cdots q_{m-1}\rangle = |l^*\rangle$ with l^* as the m-bit representation of l, for $l = 1, \cdots, 2^m$. The m-fold

UCR is built recursively following

$$
\begin{aligned}
U_{q_m}^{q_0,\cdots,q_{m-1}}(\theta_1,\cdots,\theta_{2^m}) \quad &= \\
& U_{q_m}^{q_1,\cdots,q_{m-1}}(\theta_1,\cdots,\theta_{2^{m-1}}) \\
& \cdot \; CX_{q_0,q_m} \\
& \cdot \; U_{q_m}^{q_1,\cdots,q_{m-1}}(\theta_{2^{m-1}+1},\cdots,\theta_{2^m}),
\end{aligned}
\tag{2.6}
$$

for $m > 1$, where $U_{q_1}^{q_0}(\theta_1,\theta_2) = [IY]_{\theta_1} CX_{q_0 q_1} [IY]_{\theta_2}$. The bottom in Figure 2.2(c) shows how to generate $U_3^{0,1,2}$ from $U_2^{0,1}$. The n-qubit generator is built with the application of UCRs at different levels,

$$
\begin{aligned}
\mathcal{G}^n(\theta_1,\cdots,\theta_{2^n-1}) \;&=\; [YI\cdots I]_{\theta_1} \\
& \cdot \; U_{q_1}^{q_0}(\theta_2,\theta_3) \\
& \cdot \; U_{q_2}^{q_0,q_1}(\theta_4,\cdots,\theta_7) \\
& \vdots \\
& \cdot \; U_{q_{n-1}}^{q_0,\cdots,q_{n-2}}(\theta_{2^n-2^{n-2}},\cdots,\theta_{2^n-1}).
\end{aligned}
\tag{2.7}
$$

The number of parameters scale as $2^n - 1$ and the number of CX gates as $2^n - n - 1$. The relationship between the quantum state amplitudes generated by $\mathcal{G}^n$ with the parameters $\vec{\theta}_G$ defines a parameterization of $\mathbb{S}^{2^n-1}$. For more details, see section 2.4.

For the discriminator's $\mathcal{D}(\theta_D)$ structure, we consider an ancillary qubit whose measurement decides whether an input state belongs or not to the quantum dataset $\mathcal{C}_Q$. In Figure 2.2(d), we present the four-qubit PQC design, three qubits for the input to discriminate and a fourth qubit to measure. The discriminator aims to project an arbitrary quantum state in $\mathbb{S}^{(2^n-1)}$ into a Hilbert space $\mathbb{S}^{(2^{n+1}-1)}$. For that, we consider the same structure used for the generator with extra local rotations in the last layer. This makes the number of parameters scale as $2^{n+1} + n - 1$ and the number of CX gates as $2^{n+1} - n - 2$. To

illustrate the discrimination mechanism behind $\mathcal{D}^n$, consider the single-qubit quantum state discrimination. We consider a two-qubit discriminator and classical data domain in the real line, from 0 to 10 (see Figure 2.3). Following the SI encoding, see Fig. 2.3(b), we show in Fig. 2.3(c) the different decision boundaries the discriminator defines when one of its parameters is active, i.e., $\neq 0$. The decision boundary is defined by $\langle Z \rangle = 0$ (see dotted green lines in Fig. 2.3(c)).

2.3.3 Interaction and optimization

The discriminator's interaction with $\mathcal{C}_Q$ and the generator's output is as follows. Initially, we calibrate the discriminator by minimizing $\mathcal{L}_D(\vec{\theta}_D; \{\vec{\theta}_G^k\}_N)$ with respect to $\vec{\theta}_D$. Once we calibrate the discriminator, we use the optimal parameter values θ_D^* in the generator optimization, i.e. $\vec{\theta}_D \leftarrow \vec{\theta}_D^*$. In this step, the algorithm minimizes $\mathcal{L}_G(\{\vec{\theta}_G^k\}_N; \vec{\theta}_D^*)$, obtaining the optimal configuration $\{\vec{\theta}_G^k\}_N^*$ that will be used to update $\{\vec{\theta}_G^k\}_N$ in the next epoch, i.e. $\{\vec{\theta}_G^k\}_N \leftarrow \{\vec{\theta}_G^k\}_N^*$. In both training processes, discriminator and generator, the algorithm iterates for a fixed number of steps or quantum circuit executions or until the learning reach a local minimum and the costs, $\mathcal{L}_G$ and $\mathcal{L}_D$, do not decrease.

We consider the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a gradient-free optimization algorithm, to update the generator and discriminator parameters in each learning process. This optimization algorithm uses the principle of biological evolution, where the repeated interplay of variation and selection yields a global minimum. The evolution strategy starts with an initial population of $\Theta = \{\vec{\theta}_1, \cdots \vec{\theta}_P\}$, with $P = 4 + \lfloor 2 \log M \rfloor$ and $\vec{\theta}_k = (\theta_1, \cdots, \theta_M)$. In our case, M is the number of PQC parameters. In each generation (execution of P quantum circuits), a new population Θ' is generated by variation (according to a multivariate normal distribution) of the current parental population Θ. Then, some individuals in Θ' are selected to become the next generation parents based on their $\mathcal{L}_{D/G}$ value. Over the generation sequence, sets of parameters with better and better $\mathcal{L}_{D/G}$ values are generated.

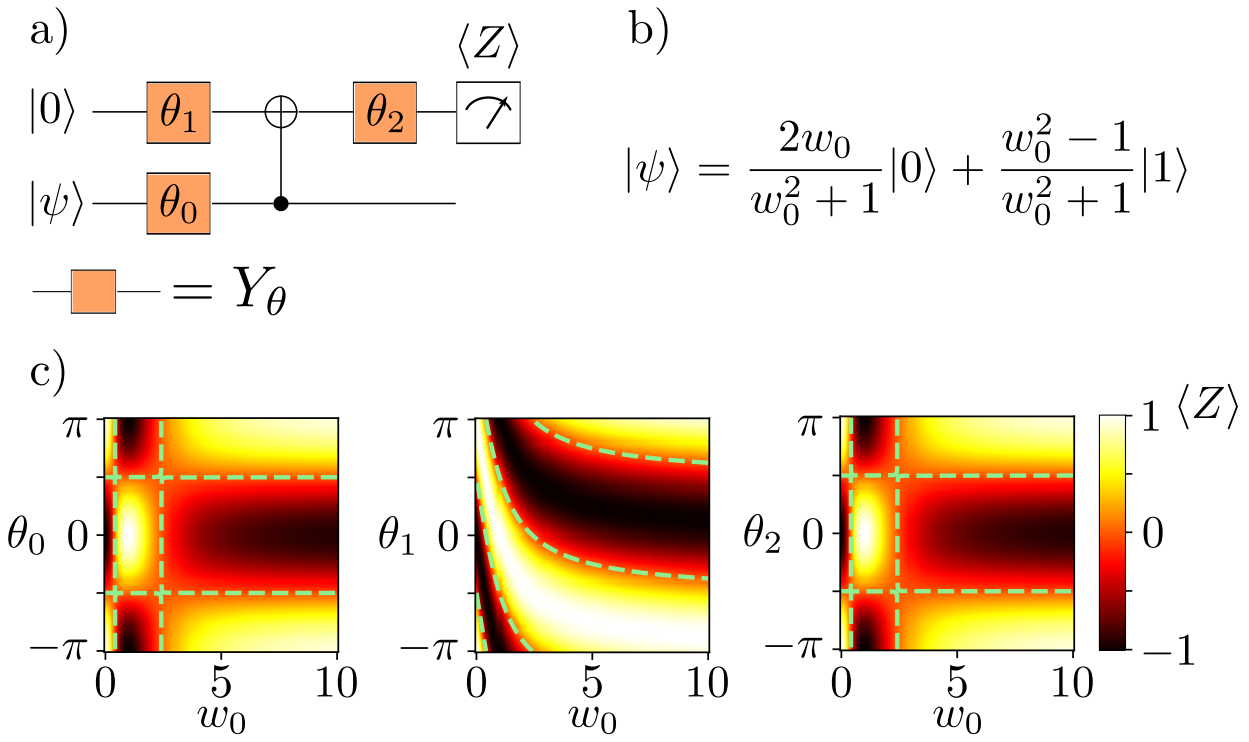


Figure 2.3: The expressive power of the 2-qubit discriminator. We represent the discrimination power by its projection on the characteristic plane for each discriminator parameter. In a) the quantum circuit for the two-qubit discriminator, b) the SI encoding model for a single-qubit, and c) heat plots showing how the discriminator can separate data classes, identified by the values of $\langle Z \rangle = \pm 1$.

If the population has enough information to estimate the covariance matrix reliably, the optimizer updates the covariance matrix that helps compute the new mean value for the distribution presented in Θ'. For a deeper discussion about stochastic research algorithms, see Ref. [23]. In the numerical experiments, we use the CMA-ES python implementation [24].

2.4 Preparation of an arbitrary two-qubit quantum state with real amplitudes

To clarify the relationship between the parameters $\vec{\theta}$ and the real amplitudes $x_1, ..., x_{2^n}$, we consider the two and three-qubit cases. The two-qubit case is straightforward and was studied in [43]. There, the amplitudes follow

$$\begin{aligned} x_1 &= c_{\theta_1} c_{\theta'_1}, \; x_2 = c_{\theta_1} s_{\theta'_1} \\ x_3 &= s_{\theta_1} c_{\theta'_2}, \; x_4 = s_{\theta_1} s_{\theta'_2} \end{aligned} \tag{2.8}$$

with $H_2(\theta_2, \theta_3)^T = (\theta'_1, \theta'_2)$. Where, H_k stands for the Hadamard matrix of order k, $c_\theta = \cos(\theta/2)$, and $s_\theta = \sin(\theta/2)$. The transformation (2.8) defines a parameterization for $\mathbb{S}^3$, for more details see Ref. [43]. For the three-qubit case, the amplitudes follow a pattern

$$\begin{aligned} x_1 &= c_{\theta_1} c_{\theta'_1} c_{\theta''_1}, \; x_2 = c_{\theta_1} c_{\theta'_1} s_{\theta''_1}, \\ x_3 &= c_{\theta_1} s_{\theta'_1} c_{\theta''_4}, \; x_4 = c_{\theta_1} s_{\theta'_1} s_{\theta''_4}, \\ x_5 &= s_{\theta_1} s_{\theta'_2} s_{\theta''_2}, \; x_6 = s_{\theta_1} s_{\theta'_2} c_{\theta''_2}, \\ x_7 &= s_{\theta_1} c_{\theta'_2} s_{\theta''_3}, \; x_8 = s_{\theta_1} c_{\theta'_2} c_{\theta''_3}, \end{aligned} \tag{2.9}$$

where, $H_2(\theta_2, \theta_3)^T = (\theta'_1, \theta'_2)$, and $H_4(\theta_4, \cdots, \theta_7)^T = (\theta''_1, \cdots, \theta''_4)^T$. Defining a parameterization for $\mathbb{S}^7$, therefore, for three (two) qubits $\mathcal{G}^{3(2)}$ covers the entire real Hilbert space. By induction, we can establish that $\mathcal{G}^n$ can prepare an arbitrary quantum state.

An arbitrary real amplitude quantum state $|\psi\rangle = x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle$ in the 2-qubit Hilbert space is prepared following 2.8. This map between the parameter space and the hypersphere is one-to-one. It is also surjective, since we can find for every $\vec{x} \in \mathbb{S}^3$, a vector $\vec{\theta}$ in the parameter space, such that $\vec{\theta} \to \vec{x}$ [43]. Since this is true for $\mathbb{S}^3$, an induction argument will show that this is the case for the n-qubit Hilbert space.

2.5 Discussion

We addressed the question of building a quantum adversarial model per the limitations of commercial quantum processors. We proposed re-qGAN, a re-imagined and optimized architecture for quantum generative adversarial networks, using low-budget resources regarding quantum circuit depth and executions. To avoid any possible loss of information in the classical encoding, we proposed a novel strategy based on stereographic projection. The SI encoding offers a one-to-one map to encode classical data into a real Hilbert space. We can extend this encoding to complex values by embedding the n-qubit Bloch sphere in a $\mathbb{R}^{2^{n+1}-1}$, defining a $\mathbb{S}^{2^{n+1}-1}$, and associating the points of this hyper-sphere with points in the real plane $\mathbb{R}^{2^{n+1}-2}$. In doing so, we increase the size of the real-data dimension we can encode per qubit and simultaneously the number of CXs and local operations in the quantum state preparation. To keep the depth of the quantum circuit at a low level and to avoid an unnecessary boost of quantum imperfection in an eventual on-hardware implementation, we stick with the real amplitude constraint even though it needs more qubits to encode the classical information.

Additionally, as illustrated in figure 2.3, the more affluent input class enabled by the stereographic encoding positively affects the discriminator's classification ability. However, since the inverse stereographic projection is a conformal map, i.e., it does not preserve distances, fluctuations around the north pole can be amplified in the real plane bringing error in the de-encoding process. We can consider a re-scaling factor for the classical input to keep the projection onto the hyper-sphere away from the north pole.

In the quantum adversarial game, The generator is designed for real amplitude quantum state preparation. We require fewer parameters but more qubits than we would if we

considered complex amplitude quantum states to encode classical information. This approach is motivated by the limited set of native gates available in physical devices and, on the other hand, to minimize the noise effects related to the quantum circuits' depth. The relationship between the generator's parameters and the quantum amplitudes defines a hyper-sphere parameterization as illustrated in the 2 and 3 qubit cases in Appendix 2.4. By constructing a generator that covers the entire real Hilbert space, we get a good trade-off between expressive power and computational resource requirements for the adversarial model.

Agents trained in an adversarial game, both in classical or quantum machine learning [18], do not always converge to Nash equilibrium with gradient descent [37]. Therefore, by building an architecture using fewer parameters, we can update the generator and discriminator by gradient-free optimization procedures without error propagation in consecutive evaluations present in gradient-based optimizations. The classical optimization performance depends on the choice of the loss function [32]. In this case, we consider a least-square-based loss function (Eqs. 2.2 and 2.3), with ± 1 as target values. The configuration of the loss functions pushes the discriminator to converge to a local minimum, under adversarial training, that yields the desired outcome of not being able to classify generated samples as fake data.

In the discriminator's design, we considered the projection of the input quantum state into a higher dimensional Hilbert space. In Figure 2.3, we observe an excellent nonlinear behavior in the way the discriminator can separate data classes, where the mean value $\langle Z \rangle$ determines the data class in the ancillary qubit.

We tested our qGAN architecture in the learning task of generating images of handwritten digits, using a subset of the MNIST dataset as our reference database. We successfully generated handwritten digit images using 2,3 and 4 qubits. We use a genetic algorithm (CMA-ES) to update the generator's and discriminator's parameters due to the reduced number of parameters in our model. the re-qGAN algorithm convergences after a fewer number of epochs, in comparison to state-of-the-art qGAN models, such as [52], while using fewer resources in general. Its ability to learn distributions in a few epochs, using a reduced number of parameters and a small training data set, points to its feasibility on real quantum hardware, with potential advantages such as quantum error robustness, from using fewer

operations and qubits. Additionally, the re-qGAN structure uses native quantum operations in most of the superconducting quantum platforms [2], with a small compilation effort in trapped ions platforms [3]. This further points to its feasibility on commercial devices.

Counter-intuitively, the re-qGAN's output resolution does not improve with the number of qubits. We can explore this feature by studying the non-trivial correlation between the size of the reference dataset and the number of qubits. Furthermore, even though re-qGAN's architecture theoretically provides flexibility to add more qubits and thus train it on more complex data, scaling it on quantum hardware will require a good trade-off between computational resources and algorithmic efficiency. More importantly, re-qGAN can be easily implemented on a real quantum machine, such as trapped-ion-based quantum processors, to test its real-world efficiency and benchmark the hardware capabilities.

Chapter 3

Custom Quantum Feature Map for Enhanced Classification

3.1 Introduction

Quantum Machine Learning (QML) [13, 47] has emerged as a field that aims to leverage concepts and tools from quantum computation and quantum information to enhance classical machine learning tasks. In this context, hybrid architectures, where a quantum processing unit (QPU) is used in conjunction with classical methods, have shown promising results. For instance, parameterized sample generators utilizing QPUs have been employed in generative tasks [12], and entangled measurement schemes in QPUs have improved discrimination efficiency in classification procedures [27]. The relationship between kernel methods and quantum computing has been well-established [48]. Havlicek et al. [25] demonstrated that a path to quantum advantage in the use of quantum kernel methods is to construct a quantum feature map that gives rise to a kernel that is intractable classically. Additionally, data encoding procedures [50, 46, 25, 49] have been proven crucial for the implementation of quantum machine learning algorithms on Noisy Intermediate-Scale Quantum (NISQ) devices [30]. The usefulness of a data encoding procedure in the NISQ era can be assessed based on certain criteria. Firstly, the number of gates employed should be at most polynomial in the number of qubits, as well as the size and dimension of the dataset, and other relevant input parameters. For machine learning applications, it is desirable to have a family of state

preparation unitaries with enough free parameters to ensure a unique quantum state for each feature vector, establishing bijectivity. Additionally, in the context of NISQ applications, achieving subpolynomial depth is even more desirable. Furthermore, the data encoding circuit should be hardware-efficient, allowing for the realization of one- and two-qubit gates without significant overhead, such as the need for gate compilation into the computer's native gate set or implementing swap gates to connect disjoint qubits.

Classification tasks serve as popular benchmarks for evaluating the performance of feature maps[14, 17, 51, 33, 34, 42, 28, 31]. Several works [25, 46] propose the application of feature maps in variational quantum classifiers and as kernel estimators for classification algorithms like support vector machines. In this dissertation, we propose a novel implementation of the inverse stereographic projection as a quantum data encoding procedure. We evaluate its performance compared to other popular encoding procedures, including amplitude encoding and the ZZ feature map [25]. Our feature map implementation offers several advantages. Firstly, it ensures bijectivity, allowing for the retention of more information from the feature vectors. Additionally, by restricting ourselves to generating quantum states with real amplitudes, we can achieve a low-depth circuit, which is particularly beneficial since real-valued vectors are commonly encountered in practical applications. Moreover, our data encoding circuit is designed to be hardware-efficient, simplifying its realization on NISQ devices.

3.2 Methods

3.2.1 Stereographic encoding feature map

The stereographic projection[1], is an influential cartographic method that enables the mapping of a sphere onto a plane. This projection is accomplished by selecting a specific point known as the center of projection (N) on the sphere and projecting it onto a plane that is perpendicular to the diameter passing through N. The stereographic projection function is characterized by its smoothness and bijectivity, covering the entire sphere except for the center of projection and encompassing the entire plane. Notably, it exhibits the property of

conformality, ensuring the preservation of angle relationships at the intersection of curves, thereby approximating the preservation of local shapes. This one-to-one characteristic of the map plays a crucial role in encoding data for machine learning tasks within the quantum framework, as it facilitates the preservation of information from different feature vectors[30].

Similarly, the inverse stereographic projection (IS) serves as the counterpart to the aforementioned mapping technique. The IS map establishes associations between vectors $\vec{u}$ in the real plane $\mathbb{R}^{2^n-1}$ and vectors $\vec{v}$ in $\mathbb{S}^{(2^n-1)} - \{\mathrm{N}\}$, according to the relationship:

$$\begin{aligned} v_i &= \frac{2u_i}{||\vec{u}||^2+1}, \text{ for } i = 1, \cdots, 2^n - 1 \\ v_{2^n} &= \frac{||\vec{u}||^2-1}{||\vec{u}||^2+1}. \end{aligned} \tag{3.1}$$

Here, a unique quantum state $|v\rangle$ can be defined as:

$$|v\rangle = v_0|0\cdots00\rangle + v_1|0\cdots01\rangle + \cdots + v_{2^n-1}|1\cdots11\rangle \ .$$

where the amplitudes are determined by the coordinates of the vector resulting from the IS map as illustrated in figure 3.1.

In this work, the feature map has been applied in two distinct scenarios. Firstly, it is employed in conjunction with a variational quantum classifier, where the majority of the learning process occurs within the quantum framework. Secondly, the feature map, together with its corresponding kernel, is computed on a quantum processor and subsequently combined with a classical support vector classifier. In both scenarios, the feature map is implemented as follows. A high-dimensional data vector from $\mathbb{R}^N$ is compressed using principal component analysis, resulting in a vector of dimension $2^n - 1$, where 'n' represents the number of qubits utilized. Subsequently, the inverse stereographic projection is applied to the resulting vector. Once this step is completed, the resulting vector is encoded into the amplitudes of a quantum state prepared through a specific quantum circuit.

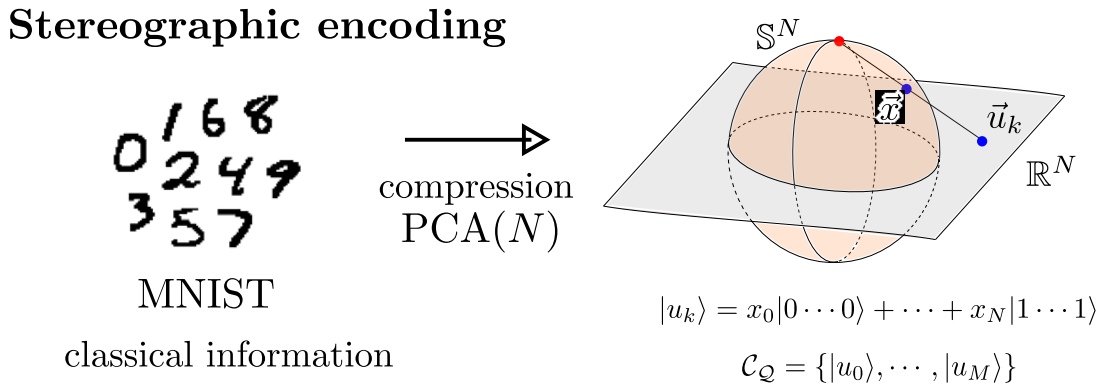


Figure 3.1: We compress the images using the principal component analysis (PCA) and project them onto a hypersphere to define the representative quantum state amplitudes. On-hardware implementation requires the preparation of every element of the set $\mathcal{C}_Q$ by using a PQC that prepares quantum states with real amplitudes

Notably, the quantum circuit employed to load the data exclusively prepares quantum states with real amplitudes, as illustrated in figure 3.2. This choice is driven by the fact that most practical machine learning problems involve real-valued vectors. Moreover, such a quantum circuit is characterized by reduced depth, requiring fewer quantum operations for execution. It also aligns with the hardware capabilities of commercial devices by utilizing commonly employed native gates. These considerations aid in reducing compilation costs and mitigating the accumulation of quantum errors when running the algorithm on NISQ (Noisy Intermediate-Scale Quantum) devices [30].

3.2.2 Variational quantum classifier

The next step in the implementation, in the first setting, is to feed the quantum state prepared by the stereographic encoding ansatz to a variational quantum classifier. This classifier is optimized to learn the appropriate class for each sample point. In practice, we adopt the same variational quantum classifier framework as Halvicek et al. [25]. The variational quantum classifier consists of a low-depth variational unitary, which comprises alternating layers of entangling gates and single-qubit rotations. This structure is a common variational ansatz [29, 21].

For a two-label classification problem, where $y \in \{+1, -1\}$, a binary measurement $\{M_y\}$ is applied to the state $W(\theta)U_{IS\phi}(x)|0\rangle^n$, where $W(\theta)$ is the classifier unitary and $U_{IS\phi}(x)$ is the inverse stereographic projection data encoding unitary. This measurement is implemented by performing measurements in the Z-basis and obtaining an output bit-string $z \in \{0,1\}^n$. The bit-string z is then fed into a chosen boolean function $f : \{0,1\}^n \rightarrow \{+1, -1\}$, which maps the bit-string to the corresponding class label.

3.2.3 Kernel based classifier

In the second setting, the quantum computer is utilized to estimate the kernel $K(\vec{u_i}, \vec{u_j})$ for all pairs of training data $\vec{u_i}, \vec{u_j} \in \mathcal{U}$. Instead of employing a variational quantum circuit to generate the separating hyperplane, we employ a classical support vector machine (SVM) [26] for classification. The optimization problem for the SVM relies on the kernel $K(\vec{u_i}, \vec{u_j})$.

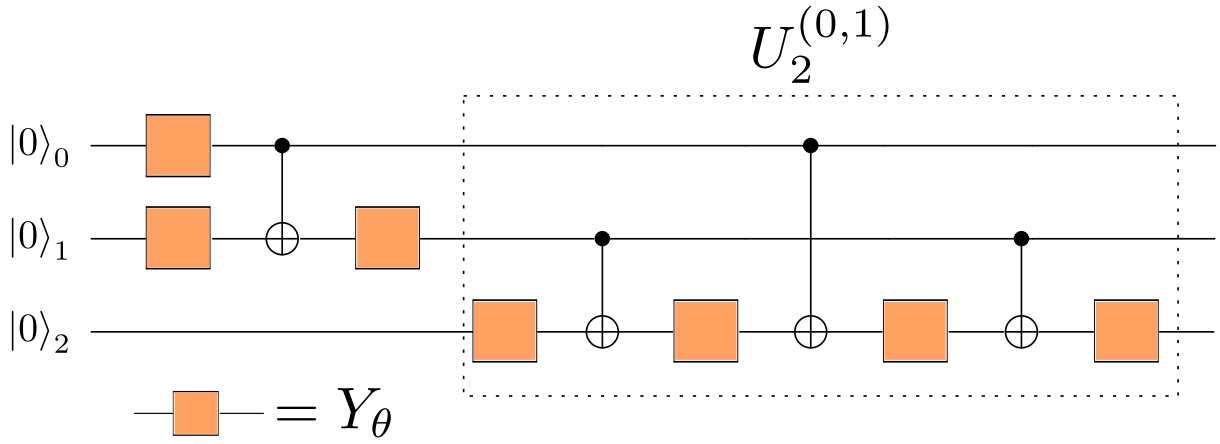


Figure 3.2: we present the parameterized quantum circuit for data loading, the ansatz prepares a 2-qubit state and entangles it with the third qubit through a uniformly controlled rotation (UCR). This PQC can be extended to more qubits by applying a cascade of UCRs. For the 4-qubit case, we need to add the UCR U_3 growing the PQC parameter set in 2^3 more elements. At the bottom, the relation between $U_3^{0,1,2}$ and $U_3^{1,2}$. The recurrence can be used for higher levels of UCRs.

The rationale for employing a quantum computer to estimate the kernel arises from the potential quantum advantage when the quantum kernel is challenging to estimate using classical methods [25].

The estimation of the kernel using the quantum computer proceeds as follows. The kernel entries $K(\vec{u_i}, \vec{u_j})$ correspond to the state fidelities between the different feature vectors. These fidelities can be directly estimated from the transition amplitudes $|\langle\phi(\vec{u_i})|\phi(\vec{u_j})\rangle|^2 = |\langle 0^n|U^\dagger_{IS\phi(\vec{u_i})}U_{IS\phi(\vec{u_j})}|0^n\rangle|^2$.

Initially, we combine the data loading parameterized quantum circuit with a mirrored version of itself and apply the resulting circuit to the initial state $|0^n\rangle$. Subsequently, we perform measurements on the qubits of the resulting circuit in the Z-basis M times and record the number of all-zero bit-strings 0^n. The count of occurrences of the state $|0\cdots0\rangle$ over multiple measurements allows us to determine the state fidelity between the quantum states prepared by each side of the combined parameterized quantum circuit. Dividing this value by the total number of measurements M yields the kernel entry $K(\vec{u_i}, \vec{u_j})$.

3.2.4 Noise configuration

When implementing our approach on NISQ (Noisy Intermediate-Scale Quantum) devices, it is essential to consider the impact of quantum noise. In this study, we incorporated a noise configuration into our quantum simulator implementation.

The noise configuration comprises a depolarizing error channel with an error probability of $p_2 = 0.001$ specifically for 2-qubit gates. This depolarizing noise is applied to all 2-qubit gates (CX gates) in our model. Additionally, we introduced readout errors with the following probabilities:

$$\begin{bmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{bmatrix}$$

Each entry in the matrix represents $P(n|m)$, which denotes the probability of observing a noisy measurement outcome as n given that the true ideal measurement outcome was m. Here, n and m represent integer representations of bit-strings.

3.3 Results

We conducted a benchmark of the feature map against two other well-known data encoding procedures: amplitude encoding and the ZZ feature map. The ZZ feature map encodes each feature into a single qubit and utilizes a Pauli-Z evolution circuit to generate a non-linear kernel [4, 25]. Our experiments employed the MNIST handwritten digits dataset, focusing on correctly classifying images of 0s and 1s.

For the benchmark, we used a training size of 20 images per digit, totaling 40 training samples, and a test dataset size of 11 images for each digit, totaling 22 testing samples. To ensure statistical significance, we repeated the experiment 100 times for each feature map, considering 2, 3, and 4 qubits, respectively.

Another set of experiments was conducted with the same setup as described above, but with the addition of quantum noise. This allowed us to assess the potential performance of the various feature maps on commercial devices.

In both experiments, we observed that the stereographic feature map outperformed the other two feature maps, with or without noise, as shown in Table 3.1 and Table 3.2, for expriments with the variational quantum classifier.

In the second setting, where the quantum framework was used for kernel computation, the stereographic encoding achieved accuracy scores of up to 94% with 3 qubits, compared to scores ranging from 40 to 70% for the vanilla amplitude encoding and the ZZ feature map. It is noteworthy that in this setting, the stereographic encoding exhibited results comparable to high-performing classical kernels such as the Gaussian kernel.

3.4 Discussion

We conducted experiments to evaluate the performance of three feature maps in the task of classifying handwritten digit images. Our experiments were conducted in two settings: utilizing the feature maps with a variational quantum classifier to generate the separating hyperplane, and employing them as kernel estimation tools on a quantum computer to explore the potential classical intractability of quantum kernels.

Table 3.1: Average F1 and Accuracy scores for the feature maps without noise. A.E: Amplitude Encoding, S.E: Stereographic Encoding, ZZ: ZZ Feature Map.

	F1 score			Accuracy		
n	A.E	S.E	ZZ	A.E	S.E	ZZ
2	0.59	0.77	0.5	0.59	0.78	0.49
3	0.55	0.79	0.51	0.55	0.8	0.51
4	0.53	0.8	0.49	0.55	0.79	0.50

Table 3.2: Average F1 and Accuracy scores for the feature maps with noise. A.E: Amplitude Encoding, S.E: Stereographic Encoding, ZZ: ZZ Feature Map.

	F1 score			Accuracy		
n	A.E	S.E	ZZ	A.E	S.E	ZZ
2	0.61	0.77	0.54	0.57	0.77	0.50
3	0.56	0.76	0.55	0.55	0.76	0.49
4	0.59	0.73	0.55	0.53	0.71	0.50

The performance of the ZZ feature map can be attributed to its inherent utilization of more resources for encoding classical information compared to amplitude encoding. This suggests that it may not be suitable for high-dimensional datasets. Amplitude encoding offers an alternative by enabling the encoding of $\log_2(n)$ dimensions into n qubits, providing an exponential advantage. The stereographic feature map further enhances this advantage by mitigating information loss through a one-to-one mapping, resulting in a more expressive feature map and access to a larger region of the feature space.

However, the implementation of the stereographic feature map in this work has certain limitations. For instance, while a quantum circuit that prepares real amplitude quantum states has significantly lower depth (using only half the number of gates) compared to a circuit that prepares any quantum state with complex amplitudes, the depth of our data encoding circuit still exhibits exponential growth relative to the number of qubits. It would be of interest to investigate the feasibility of implementing the stereographic feature map with an even lower-depth circuit, to enhance its usefulness for NISQ device implementation.

Bibliography

[1] AMS :: Feature Column :: Stereographic Projection. 43

[2] IBM Quantum. 41

[3] IonQ | Trapped Ion Quantum Computing. 41

[4] PauliFeatureMap — Qiskit 0.43.0 documentation. 49

[5] Qiskit. 15

[6] Quantum machine learning — PennyLane. 15

[7] TensorFlow Quantum. 15

[8] Yao - Quantum Computing in Julia. 15

[9] Beer, K. and Müller, G. (2021). Dissipative quantum generative adversarial networks. *arXiv:2112.06088 [quant-ph]*. arXiv: 2112.06088. 25

[10] Behrman, E. C., Nash, L. R., Steck, J. E., Chandrashekar, V. G., and Skinner, S. R. (2000). Simulations of quantum neural networks. *Information Sciences*, 128(3):257–269. 17

[11] Benedetti, M., Garcia-Pintos, D., Perdomo, O., Leyton-Ortega, V., Nam, Y., and Perdomo-Ortiz, A. (2019a). A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):1–9. Number: 1 Publisher: Nature Publishing Group. 17, 24

[12] Benedetti, M., Lloyd, E., Sack, S., and Fiorentini, M. (2019b). Erratum: Parameterized quantum circuits as machine learning models (2019 *Quant. Sci. Tech.* **4** 043001). *Quantum Science and Technology*, 5(1):019601. ZSCC: NoCitationData[s0] tex.ids= benedetti2019a. 42

[13] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671):195–202. ZSCC: 0001635. 24, 42

[14] Blank, C., Park, D. K., Rhee, J.-K. K., and Petruccione, F. (2020). Quantum classifier with tailored quantum kernel. *npj Quantum Information*, 6(1):41. 43

[15] Braccia, P., Caruso, F., and Banchi, L. (2021). How to enhance quantum generative adversarial learning of noisy information. *New Journal of Physics*, 23(5):053024. ZSCC: 0000000. 25

[16] Bravo-Prieto, C., Baglio, J., Cè, M., Francis, A., Grabowska, D. M., and Carrazza, S. (2021). Style-based quantum generative adversarial networks for Monte Carlo events. *arXiv:2110.06933 [hep-ph, physics:quant-ph]*. ZSCC: 0000001 arXiv: 2110.06933. 25

[17] Chen, H., Wossnig, L., Severini, S., Neven, H., and Mohseni, M. (2021). Universal discriminative quantum neural networks. *Quantum Machine Intelligence*, 3(1):1. arXiv: 1805.08654. 43

[18] Cheng, J., Yang, Y., Tang, X., Xiong, N., Zhang, Y., and Lei, F. (2020). Generative Adversarial Networks: A Literature Review. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(12):4625–4647. Publisher: Korean Society for Internet Information. 40

[19] Dallaire-Demers, P.-L. and Killoran, N. (2018). Quantum generative adversarial networks. *Physical Review A*, 98(1):012324. ZSCC: 0000125. 24

[20] Dubrovin, B. A., Fomenko, A. T., and Novikov, S. P. (1992). Modern geometry—methods and applications. Part I, volume 93 of Graduate Texts in Mathematics. 34

[21] Farhi, E., Goldstone, J., Gutmann, S., and Neven, H. (2017). Quantum Algorithms for Fixed Qubit Architectures. arXiv:1703.06199 [quant-ph]. 17, 46

[22] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. arXiv:1406.2661 [cs, stat]. 10

[23] Hansen, N. and Auger, A. (2014). Principled Design of Continuous Stochastic Search: From Theory to Practice. In Borenstein, Y. and Moraglio, A., editors, *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 145–180. Springer, Berlin, Heidelberg. 38

[24] Hansen, N., yoshihikoueno, ARF1, Kadlecová, G., Nozawa, K., Rolshoven, L., Chan, M., Akimoto, Y., brieglhostis, and Brockhoff, D. (2023). CMA-ES/pycma: r3.3.0. 38

[25] Havlicek, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., and Gambetta, J. M. (2019). Supervised learning with quantum enhanced feature spaces. *Nature*, 567(7747):209–212. arXiv:1804.11326 [quant-ph, stat]. 42, 43, 46, 48, 49

[26] Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3). arXiv:math/0701907. 7, 46

[27] Huang, H.-Y., Broughton, M., Cotler, J., Chen, S., Li, J., Mohseni, M., Neven, H., Babbush, R., Kueng, R., Preskill, J., and McClean, J. R. (2022). Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186. Publisher: American Association for the Advancement of Science. 24, 42

[28] Jasso, A. V., Modi, A. K., Ferrara, R., Noetzel, J., Deppe, C., Schädler, M., and Fung, F. (2023). Implementation and Realisation of Quantum and Quantum-Inspired Stereographic K Nearest-Neighbour Clustering. 43

[29] Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M., and Gambetta, J. M. (2017). Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature*, 549(7671):242–246. arXiv:1704.05018 [cond-mat, physics:quant-ph]. 46

[30] LaRose, R. and Coyle, B. (2020). Robust data encodings for quantum classifiers. *Physical Review A*, 102(3):032420. 25, 42, 44, 46

[31] Leporini, R. and Pastorello, D. (2021). Support Vector Machines with Quantum State Discrimination. *Quantum Reports*, 3(3):482–499. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. 43

[32] Leyton-Ortega, V., Perdomo-Ortiz, A., and Perdomo, O. (2021). Robust implementation of generative modeling with parametrized quantum circuits. *Quantum Machine Intelligence*, 3(1):17. 40

[33] Li, W. and Deng, D.-L. (2022). Recent advances for quantum classifiers. *Science China Physics, Mechanics & Astronomy*, 65(2):220301. 43

[34] Li, W., Lu, Z., and Deng, D.-L. (2022). Quantum Neural Network Classifiers: A Tutorial. Technical Report arXiv:2206.02806, arXiv. arXiv:2206.02806 [cond-mat, physics:quant-ph] type: article. 43

[35] Liu, W., Zhang, Y., Deng, Z., Zhao, J., and Tong, L. (2021). A hybrid quantum-classical conditional generative adversarial network algorithm for human-centered paradigm in cloud. *EURASIP Journal on Wireless Communications and Networking*, 2021(1):37. ZSCC: 0000000. 24

[36] Lloyd, S. and Weedbrook, C. (2018). Quantum Generative Adversarial Learning. *Physical Review Letters*, 121(4):040502. ZSCC: 0000172. 24

[37] Mazumdar, E., Ratliff, L. J., and Sastry, S. S. (2020). On Gradient-Based Learning in Continuous Games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131. arXiv: 1804.05464. 40

[38] Mottonen, M. and Vartiainen, J. J. (2005). Decompositions of general quantum gates. *arXiv:quant-ph/0504100*. arXiv: quant-ph/0504100. 34

[39] Mottonen, M., Vartiainen, J. J., Bergholm, V., and Salomaa, M. M. (2004). Transformation of quantum states using uniformly controlled rotations. *arXiv:quant-ph/0407010*. arXiv: quant-ph/0407010. 34

[40] Nguemto, S. and Leyton-Ortega, V. (2022). Re-QGAN: an optimized adversarial quantum circuit learning framework. arXiv:2208.02165 [quant-ph]. 23

[41] Niu, M. Y., Zlokapa, A., Broughton, M., Boixo, S., Mohseni, M., Smelyanskyi, V., and Neven, H. (2021). Entangling Quantum Generative Adversarial Networks. *arXiv:2105.00080 [quant-ph]*. ZSCC: 0000001 arXiv: 2105.00080. 25

[42] Park, D. K., Blank, C., and Petruccione, F. (2020). The theory of the quantum kernel-based binary classifier. arXiv:2004.03489 [quant-ph]. 43

[43] Perdomo, O., Leyton-Ortega, V., and Perdomo-Ortiz, A. (2021). Entanglement types for two-qubit states with real amplitudes. *Quantum Information Processing*, 20(3):89. ZSCC: 0000001. 34, 38, 39

[44] Romero, J. and Aspuru-Guzik, A. (2021). Variational Quantum Generators: Generative Adversarial Quantum Machine Learning for Continuous Distributions. *Advanced Quantum Technologies*, 4(1):2000003. ZSCC: 0000038 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qute.202000003. 25

[45] Schuld, M. (2019). Machine learning in quantum spaces. *Nature*, 567(7747):179–181. ZSCC: 0000019. 24

[46] Schuld, M. and Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4):040504. arXiv:1803.07128 [quant-ph]. 42, 43

[47] Schuld, M. and Petruccione, F. (2021a). Introduction. In Schuld, M. and Petruccione, F., editors, *Machine Learning with Quantum Computers*, Quantum Science and Technology, pages 1–21. Springer International Publishing, Cham. 42

[48] Schuld, M. and Petruccione, F. (2021b). Machine Learning. In Schuld, M. and Petruccione, F., editors, *Machine Learning with Quantum Computers*, Quantum Science and Technology, pages 23–78. Springer International Publishing, Cham. 42

[49] Schuld, M. and Petruccione, F. (2021c). Representing Data on a Quantum Computer. In Schuld, M. and Petruccione, F., editors, *Machine Learning with Quantum Computers*,

Quantum Science and Technology, pages 147–176. Springer International Publishing, Cham. 42

[50] Schuld, M., Sweke, R., and Meyer, J. J. (2021). Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430. ZSCC: 0000082. 25, 42

[51] Sergioli, G., Bosyk, G. M., Santucci, E., and Giuntini, R. (2017). A Quantum-inspired Version of the Classification Problem. *International Journal of Theoretical Physics*, 56(12):3880–3888. 43

[52] Stein, S. A., Baheri, B., Chen, D., Mao, Y., Guan, Q., Li, A., Fang, B., and Xu, S. (2021). QuGAN: A Generative Adversarial Network Through Quantum States. *arXiv:2010.09036 [quant-ph]*. ZSCC: NoCitationData[s0] arXiv: 2010.09036. 25, 29, 40

[53] Wittek, P. (2014). *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Academic Press. Google-Books-ID: 92hzAwAAQBAJ. 14

[54] Zoufal, C., Lucchi, A., and Woerner, S. (2019). Quantum Generative Adversarial Networks for learning and loading random distributions. *npj Quantum Information*, 5(1):103. ZSCC: 0000070. 25

Vita

Sandra Nguemto was born and raised in Douala, Cameroon. She began her academic journey at the University of Buea, located in the Southwest region of Cameroon. Sandra excelled during her undergraduate studies and graduated with a Bachelor's Degree in Mathematics, achieving the top position in her class and being recognized as the third best graduating student of the university in 2014.

Seeking further knowledge and opportunities, Sandra pursued a Master's degree in Applied Mathematics at Columbia University's Fu Foundation School of Engineering and Applied Science. She successfully completed her studies and earned her Master's degree in May 2016, further enhancing her mathematical skills and expertise.

In September 2017, Sandra embarked on a Ph.D. program in Mathematics at Oregon State University, working under the guidance of Dr. Juan Restrepo. However, in 2020, due to her advisor's relocation, she decided to transfer to the University of Tennessee Knoxville's Mathematics program. At the University of Tennessee, Sandra had the privilege of collaborating with Dr. Vicente Leyton Ortega, a researcher from Oak Ridge National Laboratory. Dr. Leyton Ortega served as her mentor, particularly in the field of quantum machine learning research.

Throughout her academic journey, Sandra received invaluable support from her advisors, Dr. Leyton Ortega and Dr. Suzanne Lenhart at the University of Tennessee. Their guidance and assistance played a significant role in Sandra's development as a mathematician and researcher.

Sandra is deeply appreciative of the opportunities she has been given and the support she has received along the way. Her dedication to mathematics and her collaborations with esteemed researchers have undoubtedly contributed to her growth and success in the field.