



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

**Μελέτη Κβαντικών Υπολογισμών: Εφαρμογές, Προκλήσεις και
Προοπτικές**

Κίμων-Κωνσταντίνος Χελιώτης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Επιβλέπων
Νικόλαος Τζιρίτας

Λαμία,

2025



UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

Study of Quantum Computation: Applications, Challenges and Opportunities

Kimon-Konstantinos Cheliotis

Master thesis

Nikolaos Tziritas

Lamia

2025



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ 3^η**

**«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ, ΔΙΑΧΕΙΡΙΣΗ
ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ»**

**Μελέτη Κβαντικών Υπολογισμών: Εφαρμογές, Προκλήσεις και
Προοπτικές**

Κίμων-Κωνσταντίνος Χελιώτης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Επιβλέπων
Νικόλαος Τζιρίτας**

Λαμία, 2025

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«τίτλος εργασίας»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

**Μελέτη Κβαντικών Υπολογισμών: Εφαρμογές, Προκλήσεις και
Προοπτικές**

Κίμων-Κωνσταντίνος Χελιώτης

Τριμελής Επιτροπή:

Ονοματεπώνυμο, Νικόλαος Τζιρίτας

Ονοματεπώνυμο, ...

Ονοματεπώνυμο, ...

Επιστημονικός Σύμβουλος:

Ονοματεπώνυμο

Table of Contents

1. Introduction	8
2. Introduction to Quantum Computing	10
2.1 Quantum Mechanics basic principles.....	10
2.2 Qubits	10
2.3 Quantum Gates and Circuits	11
2.4 Fundamental Quantum Algorithms.....	11
2.5 Quantum Annealing.....	12
2.6 Noisy Intermediate-Scale Quantum (NISQ) Computing.....	14
3. Quantum Machine Learning.....	15
3.1 Fundamental differences between Classical and Quantum Machine Learning.....	15
3.2 Quantum Data Encoding	16
3.3 Variational Quantum Circuits	21
4. Quantum Algorithms	24
4.1 Quantum k -Nearest Neighbor.....	24
4.2 Quantum k -means	25
4.3 Quantum Support Vector Machines.....	26
4.4 Quantum Deep Learning	28
4.5 Quantum Principal Component Analysis.....	29
4.6 Quantum Reinforcement Learning.....	31
4.7 Quantum Kernel Methods.....	34
5. Challenges & Limitations in QC	36
5.1 Physical Hardware Limitations	36
5.2 Qubit Quality	36
5.3 Quantum Coherence	37
5.4 Barren Plateaus	39
6. Quantum Platforms and toolkits	42
6.1 QML platforms.....	42
7. Applications of QML	44
7.1 Quantum Finance	44
7.2 Chemical Engineering and Molecular Discovery	44
7.3 Cryptography	45
8. Future Directions & Opportunities	46
9. Conclusions.....	47
Bibliography	48

1. Introduction

Quantum computing represents a paradigm shift in computation, leveraging the principles of quantum mechanics—such as superposition, entanglement, and tunneling—to tackle problems that are intractable for classical machines. Unlike classical bits, quantum bits (qubits) can exist in multiple states simultaneously, allowing quantum systems to process vast amounts of information in parallel. This foundational capability opens the door to solving complex problems across a wide array of domains, including natural language processing, image and video recognition, generative modeling, and other tasks where high-dimensional data and intricate patterns are prevalent.

Of course, achieving the full potential of quantum computing requires overcoming significant technical challenges. Decoherence remains a central obstacle, but advances in quantum error correction (QEC), show that active error suppression can dramatically extend qubit lifetimes. Additionally, research into better qubit materials, like tantalum or isotopically pure silicon, continues to push coherence times further, enhancing stability and reliability.

Chapter 2 introduces the foundational principles of quantum computing, starting with the core concepts of quantum mechanics, qubits, quantum gates, and quantum circuits. It also presents the notion of Noisy Intermediate-Scale Quantum (NISQ) computing and discusses its implications for current quantum technologies.

Chapter 3 explores the field of Quantum Machine Learning (QML), outlining the key distinctions between classical and quantum approaches to ML. This chapter also provides an overview of prominent QML platforms—such as PennyLane, TensorFlow Quantum, and Qiskit—highlighting their capabilities and integrations.

Chapter 4 is dedicated to important quantum algorithms that are relevant to ML applications. These include Quantum k-Nearest Neighbors, Quantum k-means, Quantum Support Vector Machines, Quantum Deep Learning models, and others. The algorithms are examined both from a theoretical perspective and in terms of their practical utility.

Chapter 5 addresses the challenges and limitations inherent in quantum computing, focusing on factors such as qubit fidelity, noise, and decoherence. It also reviews proposed mitigation strategies and outlines promising research directions aimed at enabling scalable and reliable quantum computation.

Chapter 6 provides an overview of current quantum software platforms and development toolkits that are actively used by researchers, startups, and large technology companies to build and test quantum applications.

Chapter 7 discusses real-world applications of Quantum Machine Learning, with emphasis on domains where QML has shown potential—such as finance, chemistry, and biomedical informatics, among others.

Finally, chapter 8 outlines future directions for the field. As quantum computing continues to evolve and remains in its formative stages, breakthroughs are occurring at a rapid pace. This chapter highlights emerging research trends and areas that warrant further investigation.

2. Introduction to Quantum Computing

Quantum Computing utilizes the basic quantum phenomena which include entanglement, superposition and decoherence to process data. A classical computer uses a bit as a unit, this bit is only capable to have two possible binary values, either 0 or 1. A quantum computer processes information using quantum bits, commonly known as qubits.

2.1 Quantum Mechanics basic principles

Quantum mechanics has a variety of concepts that are crucial for QC. In this section we will analyze these concepts in the QC environment because they work fundamentally differently than in classical quantum mechanics and are easier to comprehend.

- **Superposition:** Superposition is a quantum state in which a particle has the capability to exist in more than one position or state simultaneously. Quantum computers take advantage of this by allowing qubits to represent both 0 and 1 simultaneously.
- **Entanglement:** Entanglement is a very weird concept to understand and visualize. A clear way to understand it is that the measurement of one particle affects another particle's measurements, despite them not being interconnected in any way.
- **Quantum Parallelism:** Quantum parallelism allows a quantum computer to evaluate a function over many inputs at once by placing those inputs into a superposition. In classical computing, a function is evaluated on each input sequentially.
- **Decoherence:** Decoherence refers to the breakdown of a quantum system's superposition when surrounding environment intervenes. Managing this effect remains a challenge in building reliable quantum computers.

Employing these core principles of quantum mechanics a quantum computer has the potential to run quantum algorithms more effectively as opposed to their classical counterparts.[1], [2]

2.2 Qubits

Qubits, unlike classical bits, can leverage superposition to exist in multiple states. They can be represented as:

$$\alpha|0\rangle + \beta|1\rangle$$

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

The distinctions between a classical bit and a qubit are that a qubit can take the superposition states: $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ here, α_0 and α_1 are complex coefficients known as amplitudes, which obey the normalization condition: $|\alpha_0|^2 + |\alpha_1|^2 = 1$. A qubit's state is described by a unit vector residing in a two-dimensional complex Hilbert space. In this space, the states $|0\rangle$ and $|1\rangle$ form an orthonormal basis, commonly known as computational basis states. It's impossible to directly ascertain the state of a qubit and determine the specific values of α_0 and α_1 through examination,

as in a classical bit. Quantum theory's inherent randomness means that measuring a qubit will yield the outcome 0 with a probability of $|\alpha_0|^2$ or 1 with a probability of $|\alpha_1|^2$. [1]

2.3 Quantum Gates and Circuits

Similar to the logical gates employed in classical computation theory, quantum computing uses quantum gates applied to qubits, enabling them to exist in a superposition of the binary states 0 and 1. In this section, we provide an overview of quantum gates, highlight several of the most used examples, and describe their implementation within quantum circuits.

As mentioned before, quantum gates operate on qubits including superpositions of computational basis states, which may additionally be entangled. Resulting in a notable distinction in the variety of logic gates between quantum and classical computation. In an isolated quantum computer, logic gate operations evolve the quantum state through unitary transformations up until a measurement is performed. Because of this, quantum gates are represented by unitary matrices and are inherently reversible. Their closest classical counterparts include reversible logic gates such as NOT, SWAP, CNOT, Toffoli, and Fredkin. [1], [3]

A quantum circuit, much like a classical computer's electrical circuit, consists of quantum gates and wires that carry quantum information. Quantum gates manipulate qubits to perform operations. Quantum computers are sometimes realized by various physical platforms, such as optical photons, trapped ions, nuclear magnetic resonance, quantum dots, and superconducting circuits. In research labs, primitive quantum processors have been constructed for the execution of quantum algorithms. The quantum circuit model is fundamental in quantum information and computation, often representing the essence of a quantum computer. A quantum circuit operates on qubits in a state space described by a 2^b -dimensional complex Hilbert space. The computational basis states denoted as $|x_1 \dots x_b\rangle$ correspond to the binary representations of integers. [1]

2.4 Fundamental Quantum Algorithms

2.4.1 *Shor's algorithm*

Peter Shor, in 1994, building on insights from Daniel Simon's earlier work, introduced a groundbreaking polynomial-time algorithm with bounded probability for factoring integers on a quantum computer. As with many classical factoring methods, Shor's algorithm transforms the problem into one of determining the period of a carefully chosen function. Leveraging quantum parallelism, the algorithm constructs a superposition encompassing all possible outputs of the function in a single operation. It then applies the quantum Fourier transform, which, analogous to the classical version, exposes the function's periodicity. A later measurement of the quantum state yields information about this period, which can then be used to efficiently determine a nontrivial factor of the target integer M .

2.4.2 Grover's search algorithm

While Grover's algorithm does not provide the exponential speed-up characteristic of Shor's algorithm, it nonetheless attracted considerable attention due to its effectiveness in search-related tasks. Consider, for instance, a typical routing problem aimed at identifying the shortest path between two cities among NNN possible routes. A classical approach would typically require $O(N)$ operations to exhaustively search through the possibilities. In contrast, Grover's algorithm significantly reduces this to just $O(\sqrt{N})$ operations, offering a meaningful quadratic speed-up.

2.5 Quantum Annealing

As a tailored technique in the quantum computing domain, quantum annealing aims to harness quantum mechanical effects to tackle challenging optimization problems. In this framework, the problem is mapped onto an energy landscape representing all possible solutions, and the system searches for the global minimum by exploiting quantum fluctuations. Key phenomena such as superposition and tunneling allow the system to evaluate multiple candidate solutions simultaneously and avoid becoming trapped in local minima—a common limitation in classical optimization. By navigating the energy landscape using these quantum properties, a quantum annealer can more effectively converge on an optimal or near-optimal solution.

Quantum annealing operates through the gradual transformation of a simple initial quantum system into one that encodes the problem of interest. Initially, the ground state of a simple Hamiltonian serves as the system's starting point—one that typically places all qubits into a uniform superposition of 0 and 1—representing the lowest energy configuration for that Hamiltonian. Then, over time, the Hamiltonian is slowly changed to the problem Hamiltonian, that represents the optimization problem's cost function. Upon completion of the annealing process, the Hamiltonian is dominated by the problem Hamiltonian, and the system is expected to be in its ground state, which corresponds to an optimal solution of the original problem. In practice, the quantum state is then measured to obtain a definite solution. This entire process effectively uses quantum parallelism and tunneling to “guide” the system into a low-energy solution of a complex combinatorial problem.[4]

2.5.1 Applications of Quantum Annealing [5]

Traffic flow optimization: Quantum annealing has recently shown potential in addressing traffic congestion, an increasingly problematic issue causing delays, fuel waste, and environmental pollution. Neukart et al. were pioneers in applying quantum annealing to traffic management, demonstrating how real-world GPS data can be formulated into optimization problems solved by quantum devices, such as D-Wave quantum annealers. Their study illustrated improved car-routing efficiency by redistributing traffic across alternative routes, significantly reducing congestion compared to random assignments.[6] Extending this approach, Yarkoni et al. implemented a real-time quantum-optimized bus fleet, known as the “Quantum Shuttle,” during the 2019 Web Summit in Lisbon. In partnership with Volkswagen and local authorities, they utilized live GPS data and QA algorithms to dynamically route shuttle buses, marking the first commercial use of quantum processors in a real-time public transportation scenario. [7]

Additionally, Toyota Central R&D Labs recently explored traffic signal optimization using quantum annealers, successfully minimizing traffic flow imbalances by formulating the problem through Ising Hamiltonians and solving it on D-Wave systems.[8] Despite promising results, these studies are simplifications of the complex real-world traffic problems, highlighting current hardware limitations such as problem size constraints and communication overhead. Nonetheless, quantum annealing shows strong potential to enhance real-time traffic optimization as quantum technologies continue to mature.

Scheduling and logistics: Quantum annealing has also demonstrated potential applications in scheduling and logistics, which involve allocating resources and tasks efficiently under various constraints. Rieffel et al. explored general mappings of scheduling-type problems onto quantum annealers, testing navigation and scheduling problems using early-generation D-Wave systems, highlighting potential improvements if hardware connectivity could be enhanced.[9] Further, Venturelli et al. addressed the job-shop scheduling problem with the usage of quantum annealing, a classic scheduling task where jobs must be sequenced efficiently on available machines, finding that quantum annealing offered competitive results compared to traditional heuristics, although with limitations.[10] Ikeda et al. extended this to nurse scheduling, comparing quantum annealing against classical methods, revealing that despite some promise, further optimization of annealing parameters was necessary for practical superiority.[11]

Chemistry, physics and biology simulations: Quantum annealing was initially driven by the goal of efficiently simulating complex quantum systems, like molecular interactions and biological processes, that are notoriously difficult to model using classical computers. Streif et al. demonstrated the feasibility of employing a D-Wave quantum annealer to precisely determine the ground-state energies of basic molecular systems, such as hydrogen (H_2) and lithium hydride (LiH). Although promising, they highlighted challenges due to limited hardware capabilities.[12] Mulligan et al. explored quantum annealing for protein design, integrating it with Rosetta software and showing that, despite slower absolute times compared to classical methods, the quantum approach maintained a constant solution time as problem complexity increased.[13] Harris et al. and King et al. successfully simulated complex magnetic systems and exotic phases of matter using quantum annealing processors, validating QA's potential to handle otherwise computationally demanding materials physics simulations.[14], [15] In biology, Li et al. applied quantum annealing to predict transcription factor–DNA binding interactions, outperforming classical methods on smaller datasets but noting limitations with larger data. Finally, QA also demonstrated its value in computational biology classification tasks, where quantum-enhanced Support Vector Machines showed superior accuracy compared to classical versions, particularly on smaller, realistic datasets. These results collectively highlight quantum annealing's promising capabilities in simulation-intensive fields, even though current hardware constraints still limit the complexity of practical problems it can efficiently handle.[16]

2.6 Noisy Intermediate-Scale Quantum (NISQ) Computing

NISQ computing showcases the limitations of current quantum systems, which suffer from significant error rates and a restricted number of logical or physical qubits. John Preskill explains that the *intermediate scale* in NISQ means that the scale of new quantum computers will be greater, which means they will be able to use more than 50 qubits and with the term *noisy* he suggests that these qubits will not be stable enough to be controlled perfectly.[17] A critical aim of the NISQ era is to develop techniques for fault-tolerant devices, as of now currently existing quantum operations and qubits are not error corrected and building such quantum devices might take decades. Mitigating errors in quantum operations through strategies like error correction and noise reduction.[18]

Several kinds of algorithms are already being researched on, promising to exhibit a degree of noise tolerance by employing error mitigation techniques in their programming. Error mitigation strategies are applied in algorithms to counter the effects of computational noise.[19] NISQ algorithms are designed to use quantum circuits with a limited depth, typically up to a few hundred gates, and to minimize the use of complex operations such as multi-qubit controlled unitaries. Such algorithms include Variational Quantum Algorithms (VQAs), the Quantum Approximate Optimization Algorithm (QAOA), QML models, and other methods specifically designed to function within the constraints of NISQ-era devices. These will be examined in detail in later sections. In contrast, well-known algorithms like Grover's and Shor's are generally not suitable for the NISQ regime due to their requirements for low-noise, large-scale fault-tolerant quantum hardware.[20] These algorithms require highly complex and lengthy circuits and can't achieve quantum advantage with only a few noisy qubits whilst being sensitive to noise and errors. Similarly, full-scale Trotterization doesn't fit NISQ criteria since it demands very deep quantum circuits for extended time simulations.[21]

3. Quantum Machine Learning

Machine Learning (ML) is a field about analyzing and classifying large numbers of vectors within high-dimensional spaces.[22] In Quantum Machine Learning, Classical Machine Learning and Quantum Computing are combined to develop Quantum Algorithms for the purpose of solving typical problems of ML. This is done by adapting classical algorithms to run on a Quantum Computer. This emerging field is also benefited by Classical Machine Learning by using well-established methods that can help to extend and improve quantum information theory. [23]

3.1 Fundamental differences between Classical and Quantum Machine Learning

QML has shown Showing strong promise to surpass traditional ML techniques, particularly regarding speed and accuracy for certain problems, particularly through algorithms such as Quantum Support Vector Machines (QSVM), Quantum k-Nearest Neighbor, Quantum k-Means, and Quantum Deep Learning, several of which will be explored in the upcoming sections.

Classical Machine Learning	Quantum Machine Learning
Processes information using bits, each confined to either a 0 or a 1 state	Utilizes qubits capable of superposition, allowing them to encode multiple states simultaneously
Processes information sequentially, one state at a time.	Takes advantage of quantum parallelism to handle multiple computations at once, which may lead to exponential speedups for specific problems.
Systems operate independently, with no inherent correlations between components.	Features entanglement, creating interconnected states between qubits, enabling novel computational possibilities.
Algorithms are based on classical computation principles such as linear algebra and probability theory.	Utilizes quantum algorithms, some of which leverage quantum parallelism for enhanced computational efficiency.
The modification of the classical data occurs by utilizing of logical gates	The modification of the quantum data occurs by using quantum gates

Table 1 Quantum vs Classical ML Fundamental Differences

Figure 1 illustrates the different strategies used for processing in CML and QML. In CML the data simply pass through the algorithm as is, then they're being analyzed, and an output is then generated. On the other hand, in QML the data first have to be encoded into Quantum Data, then

the quantum algorithm handles the quantum information and produces an output which has to be converted into classical data.[24]

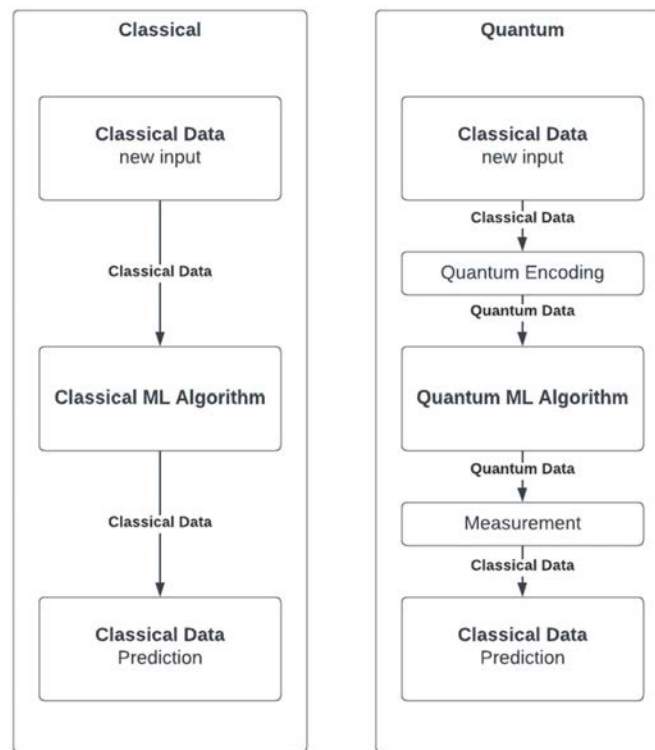


Figure 1 Processes of data in Classical ML and in QML. [24]

3.2 Quantum Data Encoding

As mentioned before, classical bits need to be encoded into quantum bits. There are multiple ways of how qubits can represent the data, so multiple data encodings are possible. The state of the data and the chosen encoding play a crucial role in the performance of the loading process. One common problem with quantum algorithm runtimes is the loading, which in some cases requires exponential time, so to accomplish the promised quantum speed-up the loading of the data needs to be done faster logarithmically or linearly.[25] In this section we will analyze some of the most common Quantum Data Encoding methods. Furthermore, ones that have been proposed recently (Qsample encoding, QRAM-based loading, and dense angle variants) which promise to improve feature representation.

3.2.1 Basis Encoding

Theoretical structure: Basis encoding maps classical bit strings directly onto multi-qubit basis states.[26] In practice, a discrete or binary-quantized feature is represented by preparing each

qubit in $|0\rangle$ or $|1\rangle$ accordingly. This essentially means that every bit becomes a qubit with no superposition or compression.

Scalability and cost: Basis encoding uses one qubit per input bit, so the qubit cost scales linearly with data dimension.[27] Preparing the state is trivial with no gates needed beyond the initialization, so classical hardware can set basis states cheaply. However, encoding real data typically requires first converting to binary form, which may increase bit-length and qubit count. Because each feature is stored in a separate qubit, basis encoding consumes many qubits for high-dimensional data which unlike amplitude encoding offers no exponential compression.

Relevance to QML: Basis encoding, being the simplest embedding, serves as a baseline in many quantum-classical comparisons. It can be useful for discrete problems or when only Boolean features are involved. In theory, a full dataset can be superposed by preparing linear combinations of basis states, but in common QML pipelines each example is encoded separately. In hybrid quantum-classical models, basis encoding effectively feeds classical information into the quantum circuit by setting qubit values. Some studies observe that basis-encoded data can improve classical classification accuracy when combined with quantum kernels, but this often relies on post-encoding quantum processing. [26]

Limitations: Scalability is the main drawback of Basis Encoding as large data requires proportionally many qubits. Without entangling gates, Basis Encoding alone cannot leverage quantum superposition or interference. Furthermore, real-valued or continuous features require normalization and discretization before basis encoding, which may be at the cost of precision. In short, Basis Encoding is simple but limited and is more commonly used where qubit resources are not the bottleneck.

3.2.2 Amplitude Encoding

Theoretical structure: Amplitude encoding maps a real-valued vector $x = (x_0, \dots, x_{n-1})$ onto the amplitudes of an n -qubit state $|\psi_x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$. In effect, each classical feature becomes a probability amplitude. This encoding exploits the full 2^n -dimensional Hilbert space of n -qubits storing $N = 2^n$ data points in a single quantum state.[14][15] Formally amplitude encoding provides an exponential compression of data: n -qubits represent N features with $N = 2^n$.

Scalability and cost: Because $n = \log_2 N$ qubits suffice for N -dimensional data, amplitude encoding is qubit-efficient. However, the circuit depth and gate count to prepare an arbitrary amplitude state can be large unless a specialized mechanism such as a QRAM is available. In general, exact state preparation of $|\psi\rangle$ requires multi-qubit rotations or decomposition that scale exponentially in n . Since the requirements for qubits are growing, compilation cost can be very high. Only some theoretical QRAM models promise polynomial-time preparation, but practical realizations remain unreachable.

Effectiveness in QML: Amplitude-encoded states, if prepared correctly, many classical features are loaded simultaneously, allowing subsequent quantum circuits to operate on a globally embedded dataset. Recent work has showed that amplitude encoding in hybrid networks has superior performance on high-dimensional regression tasks in contrast to angle encoding.[28]

Limitations: Amplitude encoding has limitations such as the measurement number is scaling with the amplitudes. As n -qubit contain 2^n amplitudes which is more expensive. This encoding type is also vulnerable to certain types of noise, like dephasing noise. Moreover, state preparation proves to be a challenge, as real hardware implementation requires deep quantum circuits to load data. In practice, amplitude encoding beyond a few qubits is challenging as the required sequences of multi-qubit rotations grow quickly with dimension. [27], [29]

3.2.3 QSample Encoding

Theoretical structure: Qsample encoding combines features of bot amplitude and basis encodings. It combines a vector of real-valued amplitudes $(\sqrt{p_1}, \dots, \sqrt{p_N})^T$ with a corresponding discrete probability distribution $p_1, \dots, p_N$. The combination of basis encoding as an empirical distribution function and the amplitudes as data weights, close to empirical probabilities. In Qsample encoding, the N features are encoded in the qubits while information of interest is conveyed through the amplitudes. This combination can be viewed mathematically as:

$$|\psi\rangle = \sum_{i=1}^{2^n} \sqrt{p_i} |i\rangle$$

Here, the quantum state's amplitudes encode meaningful probabilistic information, while the qubits themselves represent discrete features or outcomes. By representing both feature indices and probability amplitudes, Qsample encoding effectively models data distributions, making it particularly advantageous in probabilistic quantum models such as QBMs and similar frameworks.

Scalability and cost: Qsample encoding efficiently encodes $N = 2^n$ features into just n qubits, thus scaling logarithmically in the quantity of classical data points. While this qubit-efficiency is advantageous, preparing the quantum state precisely may require complex circuits. The structure of the probability distribution plays a major role in defining the complexity of state preparation; structured or sparse distributions can be efficiently encoded, whereas arbitrary or dense distributions may involve exponentially many gates and higher computational costs.

Limitations: Despite the qubit efficiency, Qsample encoding faces challenges in practical implementations, primarily because of potentially complex and deep quantum circuits required for state preparation. Accurately encoding arbitrary probability distributions can quickly become computationally expensive, especially as the dimensionality grows. Furthermore, slight inaccuracies in amplitude encoding due to noise or gate errors can significantly distort the probability distribution, reducing reliability in real-world scenarios.[27], [30]

3.2.4 Angle or Rotation Encoding

Theoretical structure: Angle encoding uses the phase or rotation of a qubit to represent the information. Rotation gates are used to encode classical information. This can be viewed mathematically as:

$$|x\rangle = \bigotimes_{i=1}^n R(x_i)|0^n\rangle$$

Where $\bigotimes$ is the operation of the tensor product over n qubits, and R is any Pauli gate R_x, R_y, R_z for x, y and z axis rotation which is applied individually to each qubit to encode the feature x_i to the angle rotation for that qubit. This state can also be written as:

$$|x\rangle = \bigotimes_{i=1}^n (\cos(x_i)|0\rangle + \sin(x_i)|1\rangle)$$

Each feature basically controls a qubit's rotation. After encoding, these product states are often followed by entangling gates to exploit quantum correlations.[27]

Scalability and cost: The simplest angle encoding uses one qubit per feature. Thus, encoding an n -dimensional real vector requires n qubits. The cost of preparation is low because each feature needs a fixed single-qubit rotation. The circuit depth of encoding is $O(n)$ for n features if done sequentially. Recent variants such as dense angle encoding reduce qubit count by encoding two values per qubit (using sequential rotations), but at the cost of more complex gate sequences.[27]

Effectiveness in QML: Angle encoding is extremely popular in variational quantum circuits and kernel methods because of its simplicity and robustness. It preserves full precision of continuous features without discretization and is naturally compatible with parameterized circuit ansätze. By itself, angle encoding does not create entanglement, but it provides a rich feature map once followed by entangling gates. In practice, angle-encoded data passed through a variational circuit can express highly nonlinear decision boundaries. Many QWML experiments rely on angle encoding to load data efficiently into circuits. Its effectiveness is evidenced by widespread use in recent studies including image and time-series classification tasks.[27]

Limitations: The main limitation is scalability, since it uses as many qubits as features, angle encoding is not compressive. Because angle encoding maps each feature to one qubit, angle encoding requires more circuit depth or layers to capture complex correlations. The dense angle encoding variant can mitigate qubit cost but introduces additional circuit overhead and potential cross-talk between features. In summary, angle encoding is easy and hardware-friendly, but its linearity limits its applicability to higher-dimensional data.

3.2.5 Quantum Random Access Memory (QRAM)

Theoretical Structure: A QRAM is a theoretical concept based on a classical RAM. It includes the same three components as CC-RAM: A data memory array, an address pointer register, and an output buffer register. Just like a CC-RAM where it receives an address with a memory index

and then loads the data stored there onto the output register. QRAM instead of addresses and output registers, it has qubits. The fundamental functionality of a QRAM is seen in Figure 2 below.

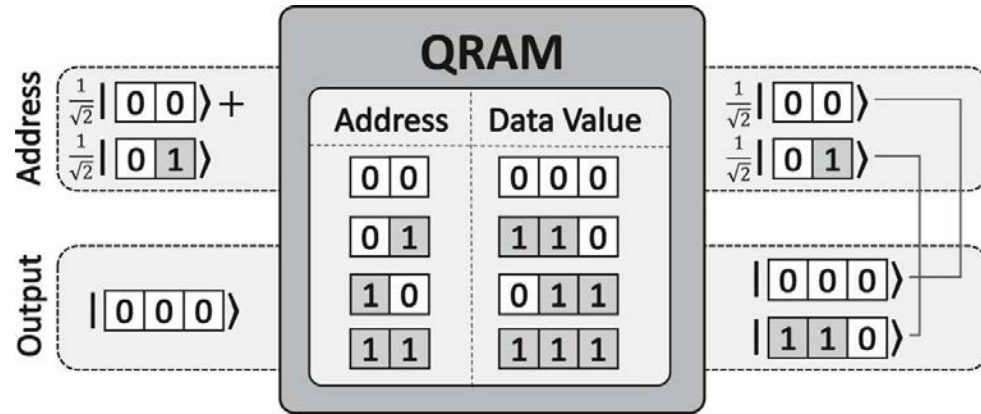


Figure 2 QRAM functionality [31]

The mathematical form of a QRAM is:

$$X \rightarrow \sum_{i=0}^{N-1} \frac{1}{\sqrt{N}} |i\rangle |x_i\rangle,$$

Or

$$\frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |\alpha\rangle_i |0\rangle \xrightarrow{QRAM} \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |\alpha\rangle_i |x_\alpha\rangle$$

Where M denotes the superposition of addresses loaded into the address register, while N stands for the total number of data entries. $|\alpha\rangle_i$ corresponds to the memory address corresponding to the i -th data element to be retrieved, while $|x_\alpha\rangle$ denotes the associated data value linked to that specific address.

Scalability and cost: The number of qubits that is required for the address register is $\lceil \log_2 M \rceil + l$ plus l qubits to encode data values with basis encoding. The number of operations is only $O(\log M)$ thus, rapid decline in power consumption is required for addressing, improving efficiency and reducing the resources that are needed for memory access.

Limitations: The production of a fully functional QRAM has yet to be done, although researchers are actively working on various approaches to build one.

To summarize, Table provides a comparison of the mentioned encoding techniques

Encoding	Num. of qubits	Runtime	Mathematical form
Basis encoding	N	$O(MN)$	$X = \sum_{i=-k}^m b_i 2^i \rightarrow b_m \dots b_{-k}\rangle$
Amplitude encoding	$\log N$	$O(MN)$	$X \rightarrow \psi_x\rangle = \sum_{\{i=0\}}^{\{2^n\}} x_i i\rangle$
QSample encoding	N	$O(2^N) / O(N)$	$X \rightarrow \psi\rangle = \sum_{i=1}^{2^n} \sqrt{p_i} i\rangle$
Angle encoding	$2^N + 1$	$O(N)$	$ x\rangle = \bigotimes_{i=1}^n R(x_i) 0^n\rangle$ or $ x\rangle = \bigotimes_{i=1}^n (\cos(x_i) 0\rangle + \sin(x_i) 1\rangle)$
QRAM encoding	$[\log M] + l$	$O(\log M)$	$X \rightarrow \sum_{i=0}^{N-1} \frac{1}{\sqrt{N}} i\rangle x_i\rangle$

Table 2 Comparison of the mentioned encoding techniques[27]

3.3 Variational Quantum Circuits

Variational Quantum Circuits (VQC) are parametrized quantum circuits, often called ansätze or quantum neural networks (QNN), used as trainable models in hybrid quantum-classical algorithms. In practice, a VQC takes classical inputs, encodes them into a quantum state, applies layers of parametrized gates, and then measures the output. By adjusting the gate parameters via a classical optimizer, VQC's can learn to carry out tasks like classification and energy estimation. These models are crucial to near-term QML because they can run on NISQ hardware. In fact, researchers have shown that VQC's can deliver quantum advantages for certain problems, making them a promising path toward practical quantum enhanced learning.[32]

3.3.1 VQC structure

A VQC typically is composed of three main components: the feature map, which encodes the input data; the variational circuit block (ansatz); and the measurement stage, which produces the output. These are visualized in the figure below:

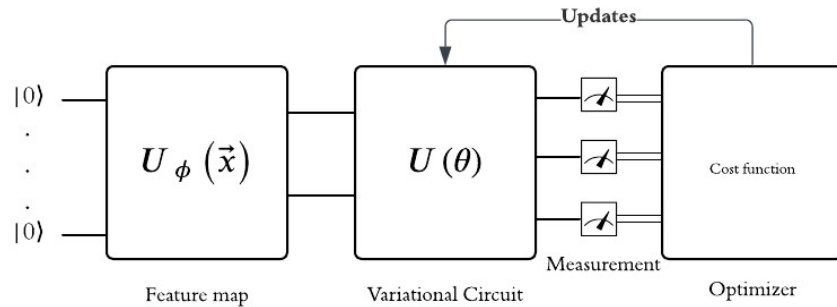


Figure 3 VQC Diagram

Within the feature map, classical input is encoded into a quantum state. One common strategy involves angle encoding, which maps each input feature to a rotational operation on a qubit. Amplitude encoding is also a good approach, where the input vector determines the amplitudes of a multi-qubit state. In either case, the initial state is transformed into a state dependent on data. After the encoding, the circuit applies a variational ansatz which is typically one or more layers of entangling gates interleaved with single-qubit rotations. Each layer contains fixed entangling gates (like CNOTs or CZs) along with parameterized rotation gates. These are the trainable parameters of the VQC. By stacking such layers, the circuit can represent a rich family of quantum states. A good analogy is the hidden layers in a neural network, where entangling gates spread information among qubits while rotation angles tune the model. Finally, the last qubit or qubits are measured. One can measure in the computational basis or measure expectation values of Pauli operators. The measurement outcomes are processed either classically or on a quantum level to yield a prediction or cost. In short, a VQC encodes the input, processes it through parameterized quantum layers, and reads out the result by measurement.[33]

3.3.2 Key contributions and theoretical foundations

Several works have laid the foundation for VQC's and hybrid algorithms.

- In 2014, Farhi et al. proposed the Quantum Approximate Optimization Algorithm (QAOA), a method that alternates layers of problem-specific and mixer Hamiltonians in a parameterized circuit to find approximate solutions of combinatorial problems. QAOA is evidently seen as a VQC tailored for optimization.[34], [35]
- The Variational Quantum Eigensolver (VQE) was proposed by Peruzzo et al. in 2014, which uses a parameterized circuit and classical optimization to find molecular ground-state energies. This was a landmark demonstration of using a VQC on a photonic processor to solve a real quantum chemistry problem. [36]
- The 2019 study by Benedetti et al. explored how parameterized quantum circuits can be applied to general-purpose models in ML, highlighting their expressive capabilities and their use in tasks such as supervised learning and generative modeling[37]
- In 2018, McClean et al. proposed the barren plateau phenomenon, showing that for deep or randomly initialized variational circuits, the cost function's gradient decreases exponentially with the number of qubits. This makes training with gradient-based methods practically impossible for larger systems. Their work explained why many VQC's struggle to train as they scale up. Barren plateaus will be explained at a later section of the thesis.[35]
- Cerezo et al. (2021) analyzed thoroughly the trainability of VQC's. They proved that cost functions defined by global observables induce barren plateaus even in shallow circuits,

in contrast to local cost functions which can avoid the issue. Their results combine trainability and locality.[38]

3.3.3 Applications in QML

VQCs have been explored for many ML tasks. In classification, one prepares a quantum state encoding the input vector and uses the VQC to produce class labels. For example, a simple variational quantum classifier on IBM hardware was demonstrated for the Iris dataset: with only a 5-qubit device, the model achieved faster training and higher capacity than a comparable classical neural network.[39] Similarly, VQCs have been used for regression given input features, the circuit can be trained to output a continuous value (e.g. by correlating a measurement expectation to the target). Quantum circuit models have even been applied to standard time-series and regression benchmarks, often matching or modestly outperforming classical models with fewer parameters.

In the generative setting, VQCs can serve as quantum generators of probability distributions. A notable instance is the Quantum Circuit Born Machine (QCBM), a generative model based on quantum circuits, where a VQC learns to produce samples from a target distribution. Recently, hybrid quantum-classical generative models have shown promise in real-world tasks: for instance, a team used a 16-qubit IBM device to implement a QCBM as part of a molecule-generation pipeline. They paired the QCBM with a classical LSTM network to design candidate drug molecules, discovering several promising inhibitors of the KRAS protein. This work highlights that quantum circuits can enrich classical generative models by capturing complex probability structures. Beyond chemistry, quantum generative adversarial networks (QGANs) and variational autoencoders (with a VQC encoder or decoder) have been proposed for image and data generation. [40]

In summary, VQC's have been applied to classification, regression and generative modeling. Surveys note that VQC has many diverse applications in QML.

Despite their promise, VQC's face serious challenges in practice. Barren plateaus are a major problem as shown by McClean et al, random or deep ansätze tend to have exponentially small gradients in high dimensional Hilbert spaces.[35] This means naïve training of large VQCs often fails. Cerezo et al. further warn that even shallow circuits can succumb to barren plateaus if there is a globalized cost function,[41] Researchers are actively researching the designing of ansätze and cost functions to avoid these flat regions, but it still remains a limitation. Another VQC problem is the noise and hardware limitations. The circuit's outputs are degraded by Noise as it causes decoherence and gate errors. De Palma et al. proved that with realistic noise, shallow VQC's quickly lose their advantage as it is unlikely to surpass high-performing classical algorithms when solving optimization problems with combinable structures such as Max-Cut.[42]

4. Quantum Algorithms

The majority of QML algorithms leverage principles and structures inspired by classical ML algorithms, incorporating quantum physical phenomena such as superposition and entanglement. QML algorithms could be separated into three categories[43]:

- **Quantum versions of classical algorithms:** These are classical ML algorithms augmented by the distinct capabilities of QC.
- **Quantum-inspired ML algorithms:** Algorithms that are inspired by Quantum Mechanics without the use of quantum computation.
- **Hybrid Quantum-Classical models:** Hybrid models combining quantum and classical methods and computational techniques. By exploiting the advantages of both these methods, we create new optimized and low-cost algorithms.

In this section we will try to analyze and explain various QML algorithms while also reviewing the different research and literature implementations.

4.1 Quantum k -Nearest Neighbor

The k -Nearest Neighbours (KNN) is probably the most straightforward classifier in ML. While primarily employed for classification tasks, it also has applications in regression paradigms. The way this algorithm works is by identifying the nearest neighbors to a new input and then the new input is classified based on its closest neighbors. However, it is crucial to know that while KNN offers a simple and versatile approach, its computational demands escalate significantly with the scale of the dataset, posing challenges in handling large datasets efficiently.

The Quantum KNN algorithm has been introduced to extinguish the issue of managing large volumes of data efficiently. The basic idea of the quantum algorithm does not drift off from its classical counterpart, adapts the traditional KNN algorithm, which typically suffers from high time complexity, into a quantum framework. By harnessing the parallelism of quantum computing, QKNN aims to significantly reduce the computational cost compared to its classical counterpart.[44] In our bibliographic research we have found various different implementations of this algorithm.

4.1.1 Key contributions and theoretical foundations

Wiebe et al. [45] suggested two different ways to measure distance within their algorithm, the inner product, which is used to calculate the inner product of non-unit vectors, along with the Euclidean method, which operates in a similar manner, but without implementing the swap test to the algorithm. A key aspect of the inner product process involves the reduction of the scaling factor M . This reduction is efficiently accomplished using the Dürr and Høyer maximum/minimum finding algorithm [46], which combines Grover's algorithm with an exponential search strategy. This hybrid approach is notably effective for identifying the largest or smallest elements in a list. Wiebe et al. particularly highlight that this method provides a near-quadratic improvement in query complexity with respect to M , offering better scalability than

classical approaches. However, they note a critical caveat: if the approximate oracle used in the process inaccurately determines the outcome of the minimum finding algorithm even once, it can compromise the entire algorithm's success. Nonetheless, the risk of such an occurrence is remarkably low, since the number of repetitions needed for amplitude estimation grows slowly, resulting in a high success rate with relatively low computational cost.

Feng et al. [44] introduce a new method of measuring the distance between vectors in the QKNN algorithm which is the polar distance. This method is drawing inspiration from the Polar coordinate system and quantum properties and aims to replace the Euclidean distance. Throughout their work, they do a comparative analysis against the Euclidean distance by testing it on different datasets, and through their results they show that the polar distance far exceeds the accuracy of the Euclidean method on the QKNN algorithm. However, they also note that because this method introduces a new parameter ω , the computational complexity increases and that also makes this method applicable only in supervised ML algorithms.

Li et al. [47] propose the hamming distance as an algorithm for distance calculation. This method employs quantum computation to parallelly compute Hamming distances between samples. A central algorithm which is used to find the shortest distance, which the authors highlight as being more efficient and more appropriate for integer data compared to the conventional Dürr-Höyer minimization method.[46] They also say that in an older proposition for this algorithm by Ruan et al. [48], they introduce two additional parameters to configure and optimize the algorithm, causing a rise in computational overhead. In their work, they resulted in the conclusion that the algorithm achieves a time complexity of $O(\sqrt{M}\log_2 M)$ when used on their testing sample. They point out that a quadratic speedup is realized by the algorithm over classical approaches. Although, some disadvantages of this algorithm are that quantum acceleration is probably not as effective for high-dimensional data.

4.2 Quantum k -means

The k -means algorithm is an unsupervised technique that partitions data into k distinct clusters. The algorithm consists of two phases. Initially, a predetermined number k of centers are randomly selected. Secondly, every data point is assigned to the closest cluster center.[49] Then a distance measurement algorithm takes place to measure the distance between the data point and the cluster centers. Upon assigning all data objects to the clusters, the first phase concludes. Next, the centroid of each cluster is recomputed, and the procedure iterates until the objective function reaches its minimum.[50] The criterion function, can be viewed mathematically as:

$$E = \sum_{i=1}^k \sum_{x \in C_i} |x - x_i|^2$$

Here, x, x_i represent the centroid of cluster C_i , and E denotes the total sum of squared errors (SSE) across all the dataset's data points. The criterion function typically calculates the Euclidean

distance to identify the closest data point to the center of a cluster. The distance of two vectors is measured by the Euclidean distance, where $x = x_1, x_2, \dots, x_n$ and $y = y_1, y_2, \dots, y_n$, and it can be resulted by the calculation: [50]

$$d(x_i, y_i) = \left[\sum_{i=1}^n (x_i - y_i)^2 \right]^{\frac{1}{2}}$$

Quantum k -means (QKM) is the quantum-enhanced variant of the classical k -means algorithm. Because of its infancy, this algorithm does not have a specific implementation. In the following paragraphs we will review and analyze different implementations of QKM.

4.2.1 Key contributions and theoretical foundations

Xiao et al. [51] were among the first to introduce a new Quantum-inspired method for k -means clustering, the k -means quantum-inspired genetic algorithm (KMQGA) and this aimed to replace the conventional k -means variable string genetic algorithm (KMVGA). The traditional k -means has several limitations, the primary one being that the value of k , representing the number of clusters, needs to be defined in advance, and the second one is that the number of clusters affects the result of the algorithm.

Kerenidis et al. [52] proposed Q-means, an algorithm that exhibits convergence and precision comparable to classical K-means. With high probability, it yields a close approximation of the k cluster centroids. They suggest that Their algorithm offers significant computational savings compared to its classical counterpart k -means algorithm, especially for the large datasets. However, the hardware limitations, the lack of quantum simulators and computers prohibit for an accurate thorough test of the q-means algorithm.

4.3 Quantum Support Vector Machines

Support Vector Machines (SVM) stands as a quintessential technique for classical ML, continuing to offer solutions for classification challenges in big data contexts, particularly across multi-domain applications within such environments. At the core of the SVM method is the goal to find the best dividing line, or hyperplane, between two classes. This involves maximizing the distance between the closest members of different classes. Key to this approach are the support vectors, which are point that lie right on the boundaries of each margin. The ideal hyperplane is then placed in the middle of this space, effectively dividing the nearest points of each class. While SVM is effective, it should be noted that it entails complex mathematics and requires significant computing power, which can lead to challenges in resource use and longer processing times. In Figure 2 we observe the optimized decision boundary delineated by a solid line, with dotted lines indicating the margins; the enclosed data points are the support vectors, highlighting the maximal margin between the two distinct classes.

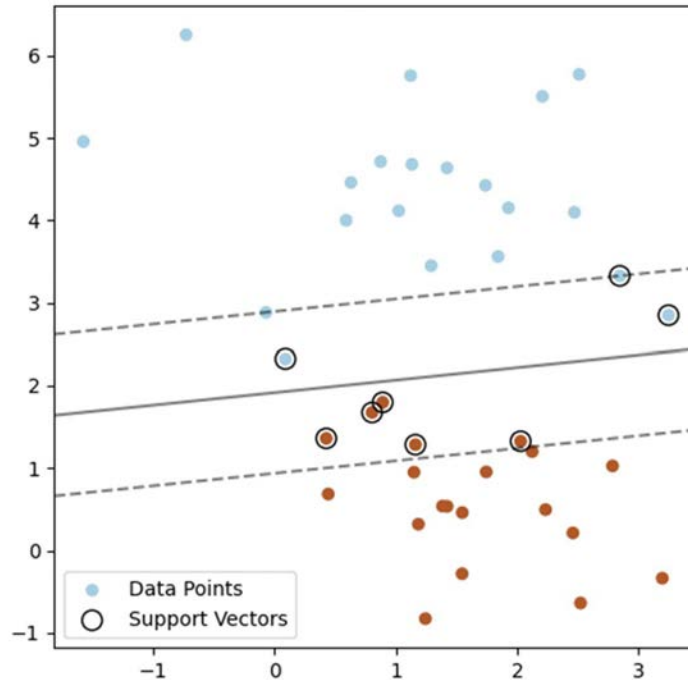


Figure 4 Here the line represents the decision boundary, the dotted lines represent the margins and the points inside the margins represent the support vectors.

The work by Rebentrost et al. [53] presents a significant advancement in the field. The authors demonstrate that in scenarios where classical sampling algorithms require polynomial time, their quantum SVM algorithm achieves an exponential speed-up. Notably this quantum SVM is implemented having a complexity of $O(\log NM)$, where N and M depict the feature size and the number of training data points, individually. The improvement in efficiency with respect to N stems from the fast quantum computation of inner products. In terms of M , Rebentrost et al. reformulate the SVM into an approximate least squares problem, using the quantum matrix inversion algorithm to obtain a solution. This reformulation allows quantum algorithms to efficiently identify the largest eigenvalues and their corresponding eigenvectors in both the training data's kernel (overlap) matrix and the covariance matrix. A crucial insight from their work is that quantum SVMs, with algorithmic complexity that scales logarithmically with both feature size and the number of training samples, provide a powerful example of a quantum algorithm designed for "big data" applications. Rebentrost et al. also observe that their quantum algorithm achieves optimal speedup when the kernel matrix of the training data is largely governed by a few dominant principal components. This discovery marks a notable advancement in QML, especially for support vector machines (SVMs), as it offers promising avenues for addressing existing challenges in handling large-scale datasets. The authors also note that this algorithm offers benefits in preserving data privacy and can act as a foundational component within a broader quantum neural network architecture.

4.4 Quantum Deep Learning

Deep learning encompasses a broad spectrum of ML techniques and architectures. Recognized for their multi-layered, hierarchical structure of nonlinear information processing stages. According to this [54] field can be classified into three primary categories according to the purpose and design of these techniques and architectures:

- **Generative Deep Architectures:** These are designed to analyze patterns or synthesize data by understanding complex correlations in observable data. They can also map the combined statistical distributions of the data and their related categories. By applying Baye's rule, these architectures can be adapted for discriminative purposes.
- **Discriminative Deep Architectures:** The primary aim of these architectures is to enable efficient pattern classification. They achieve this by focusing on the posterior distributions of different categories given the observable data.
- **Hybrid Deep Architectures:** These architectures aim at discrimination but leverage the outputs of generative architectures for improved optimization and regulation. Alternatively, discriminative criteria can be utilized to adjust model parameters within generative models from the first category.

Deep learning leverages computational architectures composed of multiple processing layers—commonly known as neural networks—to model data through hierarchical representations. This layered approach has driven remarkable advancements in domains such as speech recognition and object detection in images. By utilizing the backpropagation algorithm, these systems iteratively adjust their internal parameters to refine representations across layers. Deep learning frameworks are particularly effective in handling modalities like images, text, audio, and speech, often through the use of deep convolutional and recurrent neural networks. A key advantage of these models is their ability learn directly from data, autonomously identifying and extracting the features very important for kinds of tasks such as classification, detection, and generation. [55]

4.5 Quantum Principal Component Analysis

Quantum Principal Component Analysis (QPCA) is the quantum analogue of the classical PCA algorithm, applied to quantum states. In classical PCA, one computes the principal components (largest eigenvectors) of a covariance matrix derived from data. In QPCA, the quantum density matrix ρ which can represent either the covariance of a quantum or amplitude-encoded dataset as the object of analysis.[56] The objective is to identify eigenvectors of ρ with the largest eigenvalues, which are the principal components, using quantum operations. Lloyd et al. [57] showed that given many copies of an unknown low-rank density matrix ρ , one can perform a unitary evolution $e^{-i\rho t}$ on a quantum computer. By combining this density matrix exponentiation with quantum phase estimation, QPCA produces a quantum state that encodes the principal eigenvectors linked to the highest eigenvalues of ρ . Crucially, this procedure can identify the dominant principal components of ρ in time scaling only polynomially in the logarithm of the system dimension (an exponential speedup over naïve algorithms), under ideal assumptions. In essence, QPCA leverages the ability of quantum states to “self-simulate” their own evolution: the copies of ρ act as Hamiltonian that reveals its eigenstructure, analogous to how classical PCA diagonalizes a covariance matrix.[57]

4.5.1 Comparison with Classical PCA

While classical PCA and QPCA share the same goal to find the largest eigenvectors of a data covariance, their models are vastly different:

- **Computational complexity:** Traditional PCA applied to an $N \times N$ covariance matrix generally requires $O(N^3)$ time in standard implementations, or $O(N^2)$ when optimized techniques are used. In comparison, QPCA is theoretically capable of operating in time that scales polynomially with $\log N$, where N denotes the dimensionality of the quantum state.[57] This reflects a potential exponential advantage in terms of dimensional scaling. Nonetheless, this theoretical speedup is contingent upon the ability to efficiently load and access the quantum data.
- **Data access model:** Classical PCA requires explicit data or the full covariance matrix in memory. QPCA assumes that it is possible to generate multiple identical copies of the quantum state ρ sometimes by representing classical data as a quantum state which uses amplitude encoding or a QRAM. [26][28] Such state preparation can be costly: any overhead in preparing ρ must be accounted for in the overall runtime. Tang (2021) [59] showed that the apparent exponential speedup vanishes if one fairly accounts for the cost of state preparation. By introducing a classical model analogous to QPCA’s assumptions, she constructed a purely classical algorithm only polynomially slower than QPCA, therefore, the exponential speedups claimed for their quantum versions are largely due to assumptions made during state preparation.
- **Outputs and measurements:** Classical PCA outputs explicit numeric eigenvectors and values. QPCA outputs quantum states which correspond to the principal components. The classical descriptions of these eigenvectors would require a plethora of measurements, or

quantum state tomography which would negate the theoretical speedup. [57] Instead, QPCA is most useful for tasks where one only needs to use the principal components as quantum states (for example, as inputs to another quantum algorithm) or estimate a few observables.

- **Practical overheads:** While classical PCA is quite straightforward on digital computers. In contrast, QPCA involves subroutines like quantum phase estimation and controlled Hamiltonian simulation, which require deep circuits and high precision. On NISQ devices, these deep circuits lead to decoherence and accumulated gate errors. In practice, near-term implementations use approximate or variational methods instead. As Lin et al. (2024) [60] note that the original QPCA algorithm has yet to be demonstrated experimentally, primarily because preparing several identical quantum states and performing quantum phase estimation remain challenging.

To summarize, classical PCA is conceptually and practically simpler but scales polynomially with data size. QPCA promises exponential scaling in dimensions under quantum access assumptions, with the expense of preparing quantum states, deep circuits, and measurement overheads. The real-world “speedup” thus depends on details of the data model and hardware constraints. [27][30]

4.5.2 The QPCA algorithm

Lloyd et al.'s initial QPCA algorithm. [57] proceeds in several steps:

1. **Density matrix exponentiation:** ρ is given as a density matrix, one first uses multiple copies of ρ to implement the unitary $U = e^{-i\rho t}$. This is done by repeatedly applying partial swap operations between an input register and fresh copies of ρ . Mathematically, one shows that applying an infinitesimal swap Hamiltonian generates $e^{-i\rho t}$ on the input. In effect, the unknown state ρ acts as its own Hamiltonian. Because ρ does not need to be sparse, this technique can exponentiate a low-rank matrix in time $O(\frac{\text{polylog}(N)}{\epsilon})$, an exponential advantage over classical simulation of a dense matrix.
2. **Quantum phase estimation (QPE):** Secondly, one prepares a suitable initial state which could be another copy of ρ or a maximally mixed state and applies QPE using controlled- U gates. The phase estimation circuit entangles an ancilla register or “phase” register with the eigenvectors of ρ . After running QPE for time $O(\frac{1}{\epsilon})$, the joint state becomes: $\sum_j a_j |x_j\rangle |\lambda_j\rangle$, where $\{|x_j\rangle\}$ are the eigenvectors of ρ and $\{\lambda_j\}$ represent approximations of the eigenvalues. In other words, the algorithm produces a superposition over eigenpairs.
3. **Sampling principal components:** If ρ has rank $k \ll N$ (dominated by k large eigenvalues), then after QPE only those k eigenvectors appear with significant amplitude. One can then measure the ancilla to collapse the state onto one of the eigenvectors $|x_j\rangle$,

with likelihood in proportion to its eigenvalue. By repeating the procedure, one can sample the principal components $|x_1\rangle, \dots, |x_k\rangle$. In fact, Lloyd et al. show that if ρ is dominated by a few large eigenvalues, then one obtains copies of the state $\sum_{j=1}^k \lambda_j |x_j\rangle \langle x_j|$. From this, one can estimate expectation values of observables on each $|x_j\rangle$.

The original QPCA finds the largest eigenvalues and eigenvectors of ρ with high accuracy using $O\left(\frac{1}{\epsilon}\right)$ copies of ρ and time $O\left(\frac{\text{polylog}(N)}{\epsilon}\right)$. This is exponentially faster in N than naïve classical diagonalization, under the assumption that preparing ρ costs only $\text{polylog}(N)$.

4.6 Quantum Reinforcement Learning (QRL)

In QRL the classical agent-environment framework is expanded to allow the agent's state and/or the environment's state to be quantum in nature. One can formalize this via a quantum Markov decision process in which the agent's state is a qubit register evolving under unitary actions, and rewards arise from quantum measurements. [61]

This at each step a quantum state $|s_t\rangle$ represent the environment, the agent applies a quantum operation. QRL is the extension of classical reinforcement learning to the quantum regime, In contrast to classical RL, which operates on clearly defined state and action spaces, QRL allows these spaces to exist in superposition, enabling the agent to explore and update multiple action-state pairs simultaneously.[62], [63]

The foundational work by Dong et al. (2008) [64] introduced a QRL model in which the state-action policy is represented as a quantum superposition. The agent selects actions by measuring the quantum state and updates the probability amplitudes using quantum amplitude amplification techniques, analogous to how classical agents adjust probability distributions through reinforcement [64]. This algorithm showed that rewards can be embedded directly into quantum operations, providing an intrinsic balance between exploration and exploitation.

4.6.1 Comparison with classical RL

- **State/Action Representation:** Classical RL represents states and actions as deterministic or stochastic variables. In QRL, these are represented as quantum states within a Hilbert space. For example, superposition of qubit basis states allows for a compact representation of a vast classical state space. The policy itself can be a quantum amplitude distribution: Dong et al. represent the state-action set as a single quantum superposition whose amplitudes determine action probabilities [64]
- **Exploration Strategy:** Classical reinforcement learning agents typically rely on explicit randomness—such as ϵ -greedy strategies or Boltzmann sampling—to facilitate exploration. In contrast, Quantum Reinforcement Learning (QRL) can incorporate exploration intrinsically through the process of quantum measurement. In the framework

proposed by Dong, a measurement performed on a superposed quantum policy produces a probabilistic “eigenaction” with selection probabilities governed by the corresponding amplitude magnitudes. These amplitudes are then adjusted based on received rewards, causing actions with higher rewards to be amplified. Importantly, Grover-inspired QRL approaches update these amplitudes through quantum operations rather than classical heuristics like ϵ -greedy. Studies have also indicated that amplitude-amplification-based QRL exhibits greater resilience to learning rate sensitivity compared to traditional exploration techniques. [64], [65]

- **Data Access Model:** In classical RL, the agent obtains samples sequentially from the environment. Many QRL algorithms assume a quantum-accessible environment (an oracle) so that the agent can query it on superpositions of actions. Under this model, the agent can apply quantum search to identify rewarding action sequences much faster than classical sampling. If the environment remains classical, QRL implementations are typically hybrid: classical observations are encoded into qubits and processed by a VQC policy, but learning reduces to repeated classical experiments. [66]
- **Learning Dynamics and Speed-ups:** Classical RL often updates value functions via Bellman equations or policy gradients. In QRL, analogous updates happen via quantum operations. For example, Dong’s QRL updates probability amplitudes *in parallel* using quantum operations. Similarly, quantum policy-gradient algorithms have been proposed where gradients of a quantum parametrized policy are estimated on a quantum device. In theory, QRL can achieve faster learning: if a quantum agent can perform amplitude amplification over action sequences, it can find high-reward trajectories in $O(\sqrt{N})$ steps ahead of $O(N)$. Dunjko et al. proved that, under “luck-favoring” conditions and with quantum environment access, QRL can obtain quadratic (and in some cases exponential) sample complexity improvements over classical RL. These advantages require ideal quantum states and may not apply universally. [66]
- **Policy Architecture:** In classical deep RL, policies are neural networks. In QRL, policies are often variational quantum circuits. For instance, Chen (2020) implemented a “quantum DQN” where a VQC plays the role of the Q-network. Policy-gradient QRL similarly uses a parametrized quantum policy trained by quantum or classical optimizers. These quantum policies can represent complex decision boundaries with far fewer parameters, but they also face quantum-specific issues like measurement noise and barren-plateau gradients. [67]

4.6.2 Foundational QRL Framework

The prototypical QRL algorithm was introduced by Dong et al. (2008). They proposed mapping each classical state and action to an “eigenstate” of a quantum system, so that the entire state–action set is a quantum superposition. An action is chosen by measuring this superposition: the probability of each action (eigenstate) equals the squared amplitude. After the action and reward,

the algorithm updates the amplitudes in parallel according to the received reward. Dong et al. showed that this yields a built-in exploration–exploitation trade-off and can accelerate learning via quantum parallelism. In summary, Dong et al.’s framework defines the general-purpose QRL paradigm: use quantum superposition to sample actions and use quantum updates to amplify good actions. [64]

4.6.3 Recent Advances

Recent work has explored theoretical and near-term implementations of QRL. Here are some notable examples:

- **Quantum Policy Gradients:** Jerbi et al. at their work designed fully quantum policy-gradient algorithms, these algorithms train advanced policies in reinforcement learning through leveraging quantum interactions with the environment. Nonetheless, they provide only quadratic speed-ups in sample complexity compared to classical methods, and this advantage holds when the trained policies meet certain regularity conditions, meaning that a fully quantum agent can provably learn optimal policies faster in favorable cases. [68]
- **Hybrid Quantum Kernels:** In their study Drăgan et al. use VQC’s as “quantum feature kernels” within a standard RL agent (solving a stochastic FrozenLake environment). By testing multiple ansatze they concluded that all hybrid agents learned faster than classical baselines. Interestingly, common circuit metrics (entanglement, expressibility) did not predict performance, suggesting new design principles are needed for effective QRL architectures. Overall, this work demonstrates practical gains from hybrid QRL on small tasks.[69]
- **Quantum-Encoded Policy Sampling:** Sannia et al. proposed a hybrid approach that uses amplitude amplification to encode and update action-probability distributions for Q-learning. This quantum procedure is used within a reinforcement learning framework to encode the probability distributions that guide action selection. Concretely, their method utilizes Grover’s search algorithm to amplify the probability amplitudes for groups of actions, allowing preparation of complex action distributions in $O(\sqrt{N})$ time rather than $O(N)$. This highlights how quantum accelerators can be integrated into the RL pipeline for faster policy updates.[70]
- **Fully Quantum Markov Decision Process:** Su et al. (2025) built a framework which encodes the agent and environment within quantum components, allowing for consecutive interactions across several time steps. The agent and environment states live in qubit registers, and the environment evolution and reward computation are done by quantum arithmetic. They apply Grover’s algorithm to search over entire multi-step trajectories in superposition identifying high-reward sequences. This work is crucial because it shows how reinforcement learning could be run entirely on a quantum computer by designing an oracle that marks reward-earning trajectories.[71]

4.7 Quantum Kernel Methods

Quantum kernel methods are built on the same principle as classical kernel SVMs where data are transformed into a high-dimensional feature space and pairwise inner products (kernels) are used to learn linear classifiers. In QML, classical information is embedded into quantum states $|\phi(x)\rangle$ by a data-encoding circuit. The inner product $\langle\phi(x)|\phi(x')\rangle$ of two such states, then defines a quantum kernel that measures their similarity. Early work demonstrated that a quantum processor can estimate such kernels, allowing one to train a classical support vector machine on data embedded in a “quantum-enhanced” feature space. [72] As Mengoni et al. explain, quantum kernel methods are simply classical kernel algorithms in which the kernel function is evaluated by a quantum device.[73] Schuld et al. show that any supervised QML model (variational or otherwise) can be reframed as a kernel machine whose kernel is computed quantumly.[74] This unifies many approaches: quantum support vector machines are nothing more than SVMs using these quantum kernels, and quantum classifiers and variational circuits can often be seen as finding weights in the quantum feature space.

Kernel methods only require inner products, thus they avoid directly optimizing quantum circuit parameters and instead reduce learning to convex optimization. Schuld et al. demonstrate that numerous models designed for both near-term and fault-tolerant quantum computing can be substituted with a generalized SVM, where the kernel is capable of computing distances between quantum states. Thus, kernel-based training is assured to yield quantum models that are as good as or better than those obtained through VQC training.[74] Kernel models are guaranteed to achieve at least as low training error as any explicit variational model using the same encoding. This means that a QSVM can often fit the training data perfectly even when a variational classifier struggles. But this comes at a cost, highly expressive quantum kernels can overfit. Linear quantum kernel methods may require exponentially many training points to generalize well on certain tasks. [75]

4.7.1 Recent advances

Continuous-variable kernels: Henderson et al. derive closed-form forms for continuous-variable (CV) quantum kernels. They introduce a notion of “stellar rank” to quantify a kernel’s quantum complexity, showing that higher stellar rank (more exotic states) can improve accuracy on structured tasks. Their simulations reveal that accuracy increases with stellar rank, and that tuning a “bandwidth” parameter can trade off overfitting vs. underfitting. This provides a new interpretability angle: kernels factor into Gaussian and algebraic parts, and complex quantum states can be approximated by simpler finite-rank ones. [76]

Problem-tailored kernels: Rodriguez-Grasa et al. introduce neural quantum kernels by leveraging a quantum neural network (QNN) which is configured and trained to produce the feature map. And so, the resulting “neural EQKs” and “neural PQKs” are designed to reflect the underlying structure of the specific problem at hand. Their results demonstrate that these data-

driven kernels help counteract exponential convergence issues and enhance generalization compared to fixed, non-adaptive kernels. A notable advantage is that only a single kernel matrix—computed on the training set—is required, which significantly reduces computational overhead. This work highlights the potential benefits of learning the kernel representation itself, as opposed to relying on standard quantum encodings. [77]

Benign overfitting and hybrid kernels: Tomasi et al. address overfitting by mixing local and global kernels. They note that many standard kernels (like fidelity kernels) concentrate to nearly the identity matrix, causing overfitting. By combining a “local” kernel (measuring small subsystems) with a “global” kernel (full system overlaps), their local-global kernel exhibits benign overfitting: it can interpolate noisy data while still generalizing well. Their experiments confirm improved generalization on complex tasks using this hybrid kernel. [78]

4.7.2 Limitations and Challenges

Despite showing promise, quantum kernel methods face practical and theoretical hurdles.

Resource demands: Training a QSVM requires many quantum circuit evaluations (for each pair of data points) and sufficient qubit coherence. Jerbi et al. show that for some learning tasks even linear quantum kernels need exponentially many qubits or data to outperform simpler models. Current NISQ devices often lack the depth and qubit count needed to realize the most powerful feature maps.

Overfitting and concentration: Information concentration, as described by Thanasilp et al. proves to be problematic under generic conditions on highly expressive kernels. Consequently, when trained using a polynomial number of measurements, the resulting model generates predictions for unseen inputs that remain invariant to the input data.

In summary, quantum kernel methods reinterpret QML as classical kernel learning with a quantum feature map. They offer clear training guarantees and have motivated experimental QSVM implementations. Recent research has made progress in understanding and improving these kernels. However, challenges remain, as they demand significant quantum resources, hinge on careful feature encoding, and can overfit or collapse to triviality if not designed properly. These limitations temper claims of “quantum advantage” and motivate ongoing work on more robust, interpretable, and efficiently computable quantum kernel methods.

5. Challenges & Limitations in QC

While QC and specifically, QML is seemingly a highly promising research area, it should be emphasized that the challenges the researchers and practical applications are facing are still very hard to overcome. From limitations in the physical level, such as hardware capabilities, noise, temperature, to the theoretical level, like barren plateaus, qubit quality etc. Which will be listing and analyzing in this section, as well as discussing some possible solutions and directions for future research.

5.1 Physical Hardware Limitations

The present status of QC is referred to as NISQ, this is because every quantum computation has problems getting to an exact result because of the environmental noise. The current QML research and experiments neglect noise and promise a near-term compatible result. In controlled simulation, theoretical frameworks and algorithms have demonstrated quantum advantage, although having the same results in a practical application is near-impossible with the current hardware available. Overcoming these hardware limitations is crucial to advance fault-tolerant quantum computing and the implementation of advanced quantum algorithms.[21]

5.2 Qubit Quality

Qubit quality and particularly the accuracy of the qubits performing quantum gates holds equal importance to the quantity of qubits. The difference distinguishing high-quality qubits from lower-quality ones lies in their ability to reliably perform quantum operations without succumbing to noise, errors, or decoherence. Having high-quality qubits is fundamental for achieving quantum advantage, as they make it possible to develop deeper circuits and more complex algorithms. In table 2 we can see some other differences between the low- and high-quality qubits.

Despite recent advancements, the development of reliable qubits continues to pose significant challenges. A notable example is Google's Willow, which performed a specific calculation in just 5 minutes—a task that would require approximately 10^{25} years for a classical supercomputer to complete. Furthermore, Google has stated that scaling up the number of qubits on their chip could contribute to a reduction in computational errors.[79]

Aspect	High-Quality Qubit	Low-Quality Qubit
Coherence Time	Extended coherence duration, enabling quantum states to persist longer before experiencing decoherence	Limited coherence duration, resulting in quicker degradation of quantum states due to external disturbances

Gate Fidelity	High fidelity, meaning quantum gates operate with minimal error.	Low fidelity, causing significant computational inaccuracies.
Error Rates	Very low error rates during qubit manipulation and measurement.	High error rates, reducing reliability of results.
Noise Resistance	Resilient to environmental noise and other disturbances.	Highly susceptible to noise, leading to decoherence.
Connectivity	Fully connected or well-linked qubits, enabling efficient interaction.	Poor connectivity, requiring additional operations like SWAP gates, which increase error likelihood.
Initialization and Measurement	Accurate and repeatable initialization and measurement of quantum states.	Unreliable initialization or measurement, leading to inaccuracies in computation.

Table 3: Comparison of High-Quality and Low-Quality Qubits

5.3 Quantum Coherence

Quantum coherence is a fundamental aspect in QC, fully utilizing the principles of superposition and entanglement. Coherence ensures that the relative phase (and coefficients for example) is kept stable over time, which is necessary for interference effects in quantum algorithms. Maintaining coherence is a challenge due to interactions with the environment, which is called decoherence. Although there are ways to maintain coherence:

- **Cryogenic cooling and isolation:** Most quantum processors are operated at ultralow temperatures (near absolute zero) to freeze out thermal noise. For instance, superconducting qubits reside in dilution refrigerators at $\sim 10\text{--}20$ millikelvin and trapped-ion qubits are held in high vacuum and laser-cooled. Cryogenic conditions vastly reduce environmental vibrations and stray photons, helping qubits maintain coherence longer. Likewise, qubits are physically isolated (using vacuum chambers, magnetic shielding, etc.) to minimize interactions with the outside world.[80]
- **High grade materials:** Enhancing the materials and manufacturing processes used in qubit construction has a significant effect on coherence times. Surface or interface imperfections often act as sources of decoherence by absorbing or scattering the energy of the qubit. Employing ultra-clean, high-purity, and low-defect materials—along with more refined fabrication techniques—has successfully improved coherence duration. Notably, substituting standard niobium with tantalum in superconducting circuits has significantly decreased energy loss, leading to qubit coherence times surpassing 300 microseconds. These developments highlight how selecting superior superconducting

materials or substrates can inherently improve qubit stability. Likewise, the use of isotopically enriched silicon in spin qubits or optimized diamond lattices in NV-center qubits has increased spin coherence by minimizing interactions with noisy atomic nuclei and material defects.[81]

- **Quantum Error Correction (QEC):** Even with excellent hardware, some level of decoherence remains unavoidable. Quantum Error Correction offers a fundamental algorithmic approach to preserving quantum information by redundantly encoding a single logical qubit to multiple physical ones. The key concept is to identify and correct errors—caused by decoherence or other noise sources—before they accumulate. Recent progress has demonstrated that QEC can meaningfully extend the usable coherence time of qubits. A notable example is Google’s 2023 implementation of the surface code, where a logical qubit-maintained coherence for double the time of the best-performing physical qubit, made possible through active error correction. By operating beneath the error threshold and using tens of physical qubits per logical qubit, they achieved exponential reduction of logical error rates as the code size increased. This allowed the encoded qubits to remain coherent for much longer durations—an essential requirement for executing long quantum algorithms reliably. Although QEC introduces considerable overhead, it remains the foundational component of fault-tolerant quantum computing, enabling continuous correction of decoherence-induced errors and supporting arbitrarily long computations.[82]
- **3D Qubit Architectures:** The architecture of the qubit system can also improve coherence. Embedding qubits in three-dimensional structures (e.g. microwave cavities or multilayer circuits) can shield them from environmental noise more effectively than planar layouts. Notably, placing a superconducting qubit inside a 3D electromagnetic resonator was shown to suppress decoherence while still allowing control, resulting in much longer coherence times than earlier 2D designs. Modern quantum hardware is leveraging 3D packaging techniques to reduce cross-talk and isolate qubits (for instance, Fujitsu’s 2025 system uses a 3D unit-cell layout to maintain ultra-low temperatures and vacuum around a larger qubit array). By enlarging the physical mode volume and distancing the qubit from lossy surfaces, 3D architectures have become a proven way to boost stability.[83]
- **Dynamical Decoupling (Pulse Techniques):** Another way to fight decoherence is by continuously refocusing the qubits with sequences of control pulses. Dynamical decoupling (DD) involves applying rapid, precisely timed pulses to a qubit (or qubits) to average out environmental noise and cancel phase errors. DD is a relatively hardware-agnostic, “software” method to extend coherence on the fly. Recent research has shown that periodic decoupling pulses can even freeze a two-qubit system’s state in the short term, neutralizing environmental dephasing and preserving entanglement while the pulses are active. In experiments, applying synchronized pulses to both qubits in an entangled pair fully counteracted certain noise, effectively pausing decoherence for the duration of

the pulse sequence. Techniques like DD are already used to extend memory times in quantum devices, though they require fine-tuned control, as they are often combined with error correction for maximum effect.[84]

Overall, maintaining quantum coherence is both vital and difficult – it requires a multi-pronged effort spanning ultra-clean hardware, environmental isolation, and clever error-correction protocols. Major achievements in recent years—including advances in materials, 3D integration, enhanced error correction, and dynamical decoupling techniques are steadily pushing coherence times longer. Each improvement brings quantum computers closer to the fault-tolerant regime where qubits can retain their quantum character throughout complex computations. Continued research in preserving coherence is expected to remain a cornerstone of progress in quantum computing. Maintaining the delicate balance of quantum coherence is what will ultimately enable practical quantum machines to run intricate algorithms before decoherence can interfere.

5.4 Barren Plateaus

One of the most well-known challenges in training parameterized quantum circuits is the barren plateau phenomenon. This term refers to regions in the optimization landscape that are nearly flat, where the cost function exhibits little to no gradient over wide intervals of parameter values. In effect, when a variational quantum circuit falls into such a region, tuning the gate parameters results in minimal to no improvement in the objective function. As the system size increases, these flat regions become exponentially more pronounced, making optimization increasingly difficult. This issue was first formally identified by McClean et al. in 2018, during their analysis of randomly initialized, so-called “hardware-efficient” circuits. They demonstrated that for many large and random quantum circuits, the expected gradient vanishes and the likelihood of encountering a significant gradient in any given direction decreases exponentially with the number of qubits. The term barren plateau aptly captures this phenomenon—a vast, flat optimization surface where locating a descent direction for the cost function becomes exceptionally difficult.[85], [86]

When a variational quantum circuit encounters a barren plateau, the gradient-based optimization process effectively loses its guiding signal, as all directions in the parameter space appear equally flat. This exponential vanishing of gradients is inversely related to the system size. As a result, for a circuit involving n –qubits, the optimizer would require an exponentially large number of evaluations—or measurement shots—to reliably differentiate an actual decrease in the cost function from statistical fluctuations or noise. Effectively, the algorithm ends up wandering nearly blindly (akin to a random walk) on a flat landscape. This makes training infeasible at scale, undermining the hope that simply increasing circuit depth or qubit count will improve a VQC’s performance. Barren plateaus are now understood as a fundamental trainability barrier tied to the huge dimensionality of the quantum state space – essentially a quantum “curse of dimensionality” where almost all high-dimensional states yield nearly average (and thus uninformative) cost values. [86]

Researchers have identified several factors that can cause the barren plateaus phenomenon. Notably, the choice of cost function plays a critical role. Recent work by Cerezo et al. (as mentioned briefly in section 4.5) demonstrated that using a global cost function (one that depends on all qubits at once) is far more prone to barren plateaus, even for relatively shallow circuits. In such cases the variance of the gradient scales down exponentially as the system size increases, wiping out any learning signal. In contrast, defining the cost with local observables (measuring only a small subset or individual qubits) leads to a much more benign landscape where gradients only vanish polynomially with system size. This was a key insight: by making the cost function local, one essentially restricts the portion of Hilbert space being probed, which Cerezo et al. showed can prevent the worst of the plateau effect for shallow-depth circuits. More generally, other studies have confirmed that highly expressive or entangling ansätze (those that mimic random unitary behavior) tend to produce barren plateaus, whereas circuits with more structure or constraints can maintain trainable landscapes.[38], [87]

Given the threat barren plateaus pose to scalable QML, a variety of mitigation strategies have been proposed and tested.

- **Local Cost Functions:** Designing the VQC's objective so that it involves only local measurements or partial subsets of qubits. This reduces global entanglement in the cost evaluation and has been shown to avoid the exponential gradient collapse seen with global costs. In practice, breaking a global task into local components can maintain larger gradient signals and improve trainability.[38]
- **Structured Ansätze:** Using constrained circuit layouts instead of fully random or hardware-efficient ones. Structured ansatz circuits often exhibit better trainability because they don't explore the entire Hilbert space as freely. Such ansätze can avoid barren plateaus by limiting entanglement growth. In one study, a Hamiltonian Variational Ansatz was found to be absent of barren plateaus due to its layered structure reflecting the problem Hamiltonian. [88]
- **Layerwise Training:** In this strategy, by training shallow circuits and then gradually add more layers, while also optimizing each new layer in turn rather than optimizing all parameters of a deep circuit from scratch. Skolik et al. showed that such incremental training can help avoid the barren plateau problem, because at each step the circuit has a relatively shallow depth, with only a subset of parameters being active, while the gradients remain large enough to be useful. This approach effectively keeps the training landscape out of the flat regime by never dealing with too many unguided parameters at once.[89]
- **Initialization strategy:** Grant et al. showed that by not having all random initial parameter values, and using a smart strategy to initialize the parameters, will limit the effective depth of circuits to avoid becoming trapped in a barren plateau. Their approach employs randomly chosen initial parameters, then determines the remaining parameters resulting in the circuit to be structured in a sequence of low depth blocks effectively

performing identity operations. Thus, enabling the use of maximally compact ansätze - an achievement previously unattainable even for trivial problems.[90]

Each of these techniques attacks the barren plateau problem from a different angle, and they can be combined for better effect. While they are not exterminating the problem of barren plateaus, it significantly alleviates the issue in practice. By using local costs, carefully structured circuits, progressive training, and clever initialization, researchers have managed to extend the feasible size of VQCs that can be trained on current hardware. Barren plateaus remain an active research area and understanding their causes and crafting ways around them is crucial for scaling up QML models. The continual refinement of these error reduction techniques is therefore an important part of making variational quantum algorithms trainable and effective on larger, more complex problems.

6. Quantum Platforms and toolkits

6.1 QML platforms

QML platforms are specialized environments that merge QC and QML, offering researchers and practitioners the tools needed to develop, simulate, and execute algorithms grounded in quantum mechanics. These frameworks support the integration of quantum components—such as circuits, gates, and states—into conventional ML pipelines, thus enabling the investigation of hybrid quantum-classical strategies. By combining the unique computational properties of quantum systems—such as superposition, entanglement, and interference—with the pattern recognition and generalization strengths of classical learning models, QML platforms are designed to address computational problems that are otherwise infeasible for traditional methods. In the following subsection, we examine some of the leading QML platforms, highlighting their distinct features, functional capabilities, and roles in advancing the field of quantum-enhanced ML.

6.1.1 PennyLane

PennyLane is a Python 3 library developed to enable differentiable programming on quantum computing platforms. It accommodates both qubit-based and continuous-variable approaches, while enabling gradient computation for variational quantum circuits through classical backpropagation methods. PennyLane also provides plugins for various quantum simulators and hardware platforms, it also provides integrations with widely used ML frameworks like PyTorch, TensorFlow, Autograd, and JAX.. PennyLane can be applied to various quantum applications such as variational quantum eigensolvers, quantum approximate optimization and QML models. [91]

6.1.2 Paddle Quantum

Paddle Quantum is a cloud-based QML platform that uses Baidu PaddlePaddle as its deep learning framework. It enables the development and training of quantum neural networks, which are a type of artificial intelligence model that can operate on quantum data. Paddle Quantum offers various features and tools to facilitate QML, such as automatic differentiation, multiple optimization methods, simulation with different qubit numbers, and flexible noise models.

6.1.3 TensorFlow Quantum

By simplifying some of the common tasks in deep learning, TensorFlow has helped to advance the field and its knowledge. TensorFlow Quantum is a tool for creating and testing ML models that combine quantum and classical elements. Researchers can use Google's quantum computing resources to explore quantum algorithms, all within the same TensorFlow environment. The main goal is to work with quantum data and create models that use both quantum and classical components.

6.1.4 Qiskit

Qiskit provides essential components - such as Quantum Kernels and Quantum Neural Networks - for different purposes, such as classification and regression. The design has two advantages: it is very user-friendly and enables users to quickly build a first model without much quantum computing expertise. It is also very adaptable, and users can readily customize it to facilitate advanced QML research.

6.1.5 Amazon Braket

Amazon Braket is a cloud service designed to facilitate access to quantum computing by allowing researchers and developers to access a range of quantum devices, including photonic systems, superconducting qubits, and trapped ions, as well as high-performance quantum simulators. The platform enables users to design, test, and execute quantum algorithms across different quantum processing units (QPUs) from providers such as IonQ, Rigetti, and Oxford Quantum Circuits. Additionally, Amazon Braket integrates with classical computing resources, enabling the creation and refinement of hybrid quantum-classical algorithms. By offering a unified interface for exploring diverse quantum technologies, the service supports experimental and theoretical investigations into quantum optimization, QML, and other quantum-enabled applications.

6.1.6 Cirq by Google

Cirq is a QC platform developed by Google, for implementation, design, simulation, and execution of quantum circuits on NISQ devices. It provides a Python-based interface for defining and manipulating quantum circuits, allowing researchers to prototype and test quantum algorithms and applications. Cirq is specifically designed to support near-term quantum hardware, offering tools to address challenges such as qubit connectivity, gate decomposition, and noise mitigation. The framework supports integration with Google's quantum processors, including the Sycamore processor, as well as simulators for testing and validation. By focusing on the practical constraints of current quantum hardware, Cirq facilitates research in QML, quantum optimization, and other areas, while providing a platform for exploring the capabilities and limitations of NISQ devices.

7. Applications of QML

7.1 Quantum Finance

Quantum Computing has shown promise in quantum finance by exploiting quantum optimization algorithms as well as quantum annealers for tasks like optimizing portfolios, finding arbitrage opportunities and performing credit scoring.

Banks often simplify financial problems to make them manageable, reducing potential solutions and, consequently, potentially missing optimal outcomes. Quantum computing, with its unusual ability to handle many possible outcomes simultaneously, might help explore broader solution spaces efficiently. A classic illustration of this is the traveling salesman problem, where a salesperson must find the shortest possible route to visit multiple cities exactly once. Although manageable for a few cities, the problem quickly becomes intractable as the number of cities increases. Quantum computing could, in principle, accelerate finding optimal routes even as complexity escalates. This approach directly translates to practical financial scenarios, including optimizing investment portfolios, foreign exchange arbitrage, and complex risk assessments.[92]

Another promising application of quantum computing lies in Monte Carlo simulations, which are usually useful for risk management and derivatives pricing. As financial models become more complex, involving numerous correlated assets, classical Monte Carlo methods slow significantly, limiting their practicality. Quantum-enhanced simulations could substantially speed up these processes, enabling quicker and potentially more accurate risk assessments and derivative pricing.[93]

Companies are already developing practical quantum finance solutions. Cogniframe, for example, is building a Financial Services Operating Layer atop quantum cloud platforms, targeting complex financial optimizations. Multiverse Computing is similarly leveraging quantum and quantum-inspired algorithms to create comprehensive tools for quantitative finance and macroeconomic modeling.

Despite the exciting potential, quantum finance faces practical challenges, primarily requiring substantial advancements in quantum technology. However, even before fully functional quantum systems become common, quantum-inspired algorithms are already beginning to deliver real-world benefits by simplifying complex financial problems, providing valuable insights, and driving innovative financial solutions.

7.2 Chemical Engineering and Molecular Discovery

Quantum computing has been studied greatly in the field of chemical engineering and molecular discovery, where highly complex simulations are often needed to identify new materials or design effective drugs. The idea of using quantum systems simulating quantum systems, was originally proposed by Richard Feynman in 1982, laid the foundation for such applications. In chemical

simulations, researchers are often interested in determining a molecule's lowest energy state, which is a task that becomes exponentially complex as the atoms increase in numbers. Classical computational chemistry approaches face major challenges due to the combinatorial explosion of interactions between electrons and protons. This limits the scalability and accuracy of simulations, especially for larger molecules.[94]

Quantum computing approaches, such as quantum annealing, have been introduced as potential solutions to these problems. For example, a 5000-qubit machine was launched by D-Wave Systems in 2020, offering new opportunities for tackling optimization problems in molecular simulation.

Companies like OTI Lumionics have already begun leveraging this potential. By modeling material discovery problems directly for D-Wave's hardware, they not only prepare for a quantum future but also benefit today by deriving quantum-inspired algorithms that run efficiently on classical systems. Their partnership with Microsoft's Azure Quantum platform further improved computational performance for molecular simulations, proving valuable even without full-scale quantum computers.

7.3 Cryptography

Quantum computing is already making a significant impact in cybersecurity, driving both public investment and government attention. The World Economic Forum has warned that quantum computing could render current encryption methods obsolete, a concern echoed in the U.S. Quantum Initiative Act (2018), where cybersecurity was a central motivation. The growing technological competition between major powers like the U.S. and China has further underscored the importance of quantum security.

At the heart of this concern lies Shor's algorithm, which showed that a large and fault-tolerant quantum computer could break widely used encryption methods, particularly the RSA public-key cryptosystem, by reducing the complexity involved in integer factorization from exponential to polynomial time. This would compromise many of today's encryption standards and make secure communication vulnerable if and when such quantum machines become a reality.

While such quantum computers do not yet exist, the threat to long-term data security, especially in sectors where classified or sensitive information must remain protected for decades, is real. Even a small probability of future quantum decryption creates a present-day demand for quantum-safe encryption.

8. Future Directions & Opportunities

As quantum computing research advances, we are standing at a promising, yet uncertain future. While current hardware is limited by noise and scalability, advances in both hardware and algorithmic design are opening new doors. In the coming years, several key directions and opportunities are anticipated to shape the future of this field.

- **Toward Fault-Tolerant Quantum Computing (FTQC):** Among the most eagerly awaited milestones is the progression from NISQ devices toward completely error-corrected, fault-tolerant quantum computers. These systems would support error correction at scale, allowing long and deep circuits to be run reliably. Achieving this will require significant progress in both qubit quality and error correction codes, as well as in quantum control engineering. Scientists believe that we will witness the first Early FTQC machines within the next five years.
- **Hardware Innovations:** Improving quantum hardware is a priority across all major players in the field. From superconducting qubits and trapped ions to photonic and topological qubits, different technologies are competing to overcome current limitations. Companies such as Google, IBM, IonQ, Rigetti, Honeywell, and D-Wave are constantly pushing toward better connectivity, coherence times, and qubit counts.
- **Quantum Internet:** The Quantum Internet envisions networking capabilities, enabling secure communications and distributed quantum computing. Through the use of quantum mechanical phenomena like entanglement, teleportation, and the no-cloning theorem.[95] Unlike classical networks, it requires a fundamentally different design, as qubits are fragile and easily disturbed by environmental noise. Key challenges include reliably distributing entanglement over long distances, managing quantum memory resources, and ensuring high-bandwidth communication. To make this vision a reality, standardized interfaces must be developed so that quantum sensors, computers, and web-based applications can seamlessly make data exchanges, paving the way for a truly quantum-enabled web.[96]
- **Quantum Software and Toolkits:** Alongside hardware, software development continues to play a major role. Platforms like Qiskit, Cirq, PennyLane, and Ocean are making quantum programming more accessible. Future tools may allow for hardware-agnostic development, where algorithms can be written once and run on any backend, improving cross-platform compatibility and research efficiency.
- **Standardization and Commercialization:** We may also see future developments in quantum standards, benchmarking, and even regulations, especially as quantum technology moves toward commercialization. Quantum cloud services are already a step in this direction, allowing broader access to quantum computing resources.

Finally, the field still has many open questions, particularly in understanding the full limits of quantum advantage. We are only beginning to explore the landscape of quantum algorithms and how quantum information interacts with areas like AI, neuroscience, or complex systems.

9. Conclusions

To summarize, Quantum Computing has shaken the academic world, with researchers actively trying to overcome current challenges and develop new ways to enter the quantum world computer-wise. Many breakthroughs happen every week, from quantum computers with larger numbers of qubits, to solving hard problems in exponentially shorter times than classical computers. Overall, there is a very promising, yet uncertain future for this area of study, researchers are though hopeful that developing the first large scale quantum computer is going to happen soon.

In this thesis, I explained the basics of quantum computation in simple terms, so the reader can understand and imagine how a quantum computer processes information. Then, I presented the current era of NISQ devices and the impact they have on the development of quantum computing. We explored what QML has to offer and how it may affect our future, especially considering how classical ML and Artificial Intelligence have drastically changed the world in recent years. The most important QML algorithms were introduced to demonstrate how much faster and more accurate they could be when applied to real-world problems. The challenges of developing a large-scale quantum computer are many and difficult to overcome; this is most likely the most important part of the thesis, as unlocking fault-tolerant, noiseless, and coherent quantum computers could lead to transformative changes in computing and beyond. Also important to include were the quantum computing platforms that are currently available. These are the main tools that allow researchers and hobbyists to start developing and simulating quantum programs, potentially encouraging more students to pursue academic careers in the quantum field. Finally, the applications and future of quantum computing were discussed. While many uncertainties remain about its actual capabilities, both theoretical and emerging applications were outlined to highlight the opportunities that lie ahead.

Standing at the dawn of this new computational era, quantum computing holds the potential to reshape science, technology, and society. While there is much more to uncover, the steady progress made so far gives reason for optimism. Continued research, collaboration, and curiosity will be essential in turning today's theoretical possibilities into tomorrow's practical realities.

Bibliography

- [1] Y. Wang, “Quantum Computation and Quantum Information,” *Statist. Sci.*, vol. 27, no. 3, Aug. 2012, doi: 10.1214/11-STS378.
- [2] E. G. Rieffel and W. Polak, “An Introduction to Quantum Computing for Non-Physicists,” Jan. 18, 2000, *arXiv*: arXiv:quant-ph/9809016. Accessed: Feb. 18, 2024. [Online]. Available: <http://arxiv.org/abs/quant-ph/9809016>
- [3] C. P. Williams, *Explorations in quantum computing*, 2nd ed. in Texts in computer science. New York London: Springer, 2011.
- [4] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, “Quantum annealing for industry applications: introduction and review,” *Rep. Prog. Phys.*, vol. 85, no. 10, p. 104001, Oct. 2022, doi: 10.1088/1361-6633/ac8c54.
- [5] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, “Quantum Annealing for Industry Applications: Introduction and Review,” *Rep. Prog. Phys.*, vol. 85, no. 10, p. 104001, Oct. 2022, doi: 10.1088/1361-6633/ac8c54.
- [6] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, “Traffic Flow Optimization Using a Quantum Annealer,” *Front. ICT*, vol. 4, p. 29, Dec. 2017, doi: 10.3389/fict.2017.00029.
- [7] S. Yarkoni *et al.*, “Quantum Shuttle: traffic navigation with Quantum computing,” in *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software*, Virtual USA: ACM, Nov. 2020, pp. 22–30. doi: 10.1145/3412451.3428500.
- [8] D. Inoue, A. Okada, T. Matsumori, K. Aihara, and H. Yoshida, “Traffic signal optimization on a square lattice with quantum annealing,” *Sci Rep*, vol. 11, no. 1, p. 3303, Feb. 2021, doi: 10.1038/s41598-021-82740-0.
- [9] E. G. Rieffel, D. Venturelli, B. O’Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy, “A case study in programming a quantum annealer for hard operational planning problems,” *Quantum Inf Process*, vol. 14, no. 1, pp. 1–36, Jan. 2015, doi: 10.1007/s11128-014-0892-x.
- [10] D. Venturelli, D. J. J. Marchand, and G. Rojo, “Quantum Annealing Implementation of Job-Shop Scheduling,” Oct. 17, 2016, *arXiv*: arXiv:1506.08479. doi: 10.48550/arXiv.1506.08479.
- [11] K. Ikeda, Y. Nakamura, and T. S. Humble, “Application of Quantum Annealing to Nurse Scheduling Problem,” *Sci Rep*, vol. 9, no. 1, p. 12837, Sep. 2019, doi: 10.1038/s41598-019-49172-3.
- [12] M. Streif, F. Neukart, and M. Leib, “Solving Quantum Chemistry Problems with a D-Wave Quantum Annealer,” in *Quantum Technology and Optimization Problems*, vol. 11413, S. Feld and C. Linnhoff-Popien, Eds., in Lecture Notes in Computer Science, vol. 11413. , Cham: Springer International Publishing, 2019, pp. 111–122. doi: 10.1007/978-3-030-14082-3_10.
- [13] V. K. Mulligan *et al.*, “Designing Peptides on a Quantum Computer,” Sep. 02, 2019, *Bioengineering*. doi: 10.1101/752485.
- [14] A. D. King *et al.*, “Observation of topological phenomena in a programmable lattice of 1,800 qubits,” *Nature*, vol. 560, no. 7719, pp. 456–460, Aug. 2018, doi: 10.1038/s41586-018-0410-x.
- [15] R. Harris *et al.*, “Phase transitions in a programmable quantum spin glass simulator,” *Science*, vol. 361, no. 6398, pp. 162–165, Jul. 2018, doi: 10.1126/science.aat2025.
- [16] R. Y. Li, R. Di Felice, R. Rohs, and D. A. Lidar, “Quantum annealing versus classical machine learning applied to a simplified computational biology problem,” *npj Quantum Inf*, vol. 4, no. 1, p. 14, Feb. 2018, doi: 10.1038/s41534-018-0060-8.

- [17] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.
- [18] B. M. Terhal, “Quantum error correction for quantum memories,” *Rev. Mod. Phys.*, vol. 87, no. 2, pp. 307–346, Apr. 2015, doi: 10.1103/RevModPhys.87.307.
- [19] K. Temme, S. Bravyi, and J. M. Gambetta, “Error Mitigation for Short-Depth Quantum Circuits,” *Phys. Rev. Lett.*, vol. 119, no. 18, p. 180509, Nov. 2017, doi: 10.1103/PhysRevLett.119.180509.
- [20] F. Leymann and J. Barzen, “The bitter truth about gate-based quantum algorithms in the NISQ era,” *Quantum Sci. Technol.*, vol. 5, no. 4, p. 044007, Oct. 2020, doi: 10.1088/2058-9565/abae7d.
- [21] J. W. Z. Lau, K. H. Lim, H. Shrotriya, and L. C. Kwek, “NISQ computing: where are we and where do we go?,” *AAPPS Bull.*, vol. 32, no. 1, p. 27, Sep. 2022, doi: 10.1007/s43673-022-00058-z.
- [22] S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi, “Machine Learning Algorithms in Quantum Computing: A Survey,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom: IEEE, Jul. 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207714.
- [23] M. Schuld, I. Sinayskiy, and F. Petruccione, “An introduction to quantum machine learning,” *Contemporary Physics*, vol. 56, no. 2, Art. no. 2, Apr. 2015, doi: 10.1080/00107514.2014.964942.
- [24] A. Zeguendry, Z. Jarir, and M. Quafafou, “Quantum Machine Learning: A Review and Case Studies,” *Entropy*, vol. 25, no. 2, p. 287, Feb. 2023, doi: 10.3390/e25020287.
- [25] M. Weigold, J. Barzen, F. Leymann, and M. Salm, “Data encoding patterns for quantum computing,” in *Proceedings of the 27th Conference on Pattern Languages of Programs*, in PLoP ’20. USA: The Hillside Group, 2022.
- [26] M. Rath and H. Date, “Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy,” *EPJ Quantum Technol.*, vol. 11, no. 1, p. 72, Dec. 2024, doi: 10.1140/epjqt/s40507-024-00285-3.
- [27] M. A. Khan, M. N. Aman, and B. Sikdar, “Beyond Bits: A Review of Quantum Embedding Techniques for Efficient Information Processing,” *IEEE Access*, vol. 12, pp. 46118–46137, 2024, doi: 10.1109/ACCESS.2024.3382150.
- [28] Y. Chen, P. Griffin, P. Recchia, L. Zhou, and H. Zhang, “Hybrid Quantum Neural Networks with Amplitude Encoding: Advancing Recovery Rate Predictions,” Feb. 05, 2025, *arXiv*: arXiv:2501.15828. doi: 10.48550/arXiv.2501.15828.
- [29] N. Nguyen and K.-C. Chen, “Quantum Embedding Search for Quantum Machine Learning,” *IEEE Access*, vol. 10, pp. 41444–41456, 2022, doi: 10.1109/ACCESS.2022.3167398.
- [30] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A*, vol. 103, no. 3, p. 032430, Mar. 2021, doi: 10.1103/PhysRevA.103.032430.
- [31] M. Weigold, J. Barzen, F. Leymann, and M. Salm, “Encoding patterns for quantum algorithms,” *IET Quantum Communication*, vol. 2, no. 4, pp. 141–152, Dec. 2021, doi: 10.1049/qtc2.12032.
- [32] M. Cerezo *et al.*, “Variational Quantum Algorithms,” *Nat Rev Phys*, vol. 3, no. 9, pp. 625–644, Aug. 2021, doi: 10.1038/s42254-021-00348-9.
- [33] C.-Y. Liu, S. Y.-C. Chen, K.-C. Chen, W.-J. Huang, and Y.-J. Chang, “Programming Variational Quantum Circuits with Quantum-Train Agent,” Dec. 03, 2024, *arXiv*: arXiv:2412.01173. doi: 10.48550/arXiv.2412.01173.

- [34] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," Nov. 17, 2014, *arXiv*: arXiv:1411.4028. doi: 10.48550/arXiv.1411.4028.
- [35] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nat Commun*, vol. 9, no. 1, p. 4812, Nov. 2018, doi: 10.1038/s41467-018-07090-4.
- [36] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nat Commun*, vol. 5, no. 1, p. 4213, Jul. 2014, doi: 10.1038/ncomms5213.
- [37] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Sci. Technol.*, vol. 4, no. 4, p. 043001, Nov. 2019, doi: 10.1088/2058-9565/ab4eb5.
- [38] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nat Commun*, vol. 12, no. 1, p. 1791, Mar. 2021, doi: 10.1038/s41467-021-21728-w.
- [39] I. Piatrenka and M. Rusek, "Quantum Variational Multi-class Classifier for the Iris Data Set," in *Computational Science – ICCS 2022*, vol. 13353, D. Groen, C. De Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., in Lecture Notes in Computer Science, vol. 13353, Cham: Springer International Publishing, 2022, pp. 247–260. doi: 10.1007/978-3-031-08760-8_21.
- [40] M. Ghazi Vakili *et al.*, "Quantum-computing-enhanced algorithm unveils potential KRAS inhibitors," *Nat Biotechnol*, Jan. 2025, doi: 10.1038/s41587-024-02526-3.
- [41] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nat Commun*, vol. 12, no. 1, p. 1791, Mar. 2021, doi: 10.1038/s41467-021-21728-w.
- [42] G. D. Palma, M. Marvian, C. Rouzé, and D. S. França, "Limitations of variational quantum algorithms: a quantum optimal transport approach," *PRX Quantum*, vol. 4, no. 1, p. 010309, Jan. 2023, doi: 10.1103/PRXQuantum.4.010309.
- [43] L. Huynh, J. Hong, A. Mian, H. Suzuki, Y. Wu, and S. Camtepe, "Quantum-Inspired Machine Learning: a Survey," Sep. 08, 2023, *arXiv*: arXiv:2308.11269. Accessed: Feb. 03, 2024. [Online]. Available: <http://arxiv.org/abs/2308.11269>
- [44] C. Feng, B. Zhao, X. Zhou, X. Ding, and Z. Shan, "An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance," *Entropy*, vol. 25, no. 1, 2023, doi: 10.3390/e25010127.
- [45] N. Wiebe, A. Kapoor, and K. Svore, "Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning," Jul. 18, 2014, *arXiv*: arXiv:1401.2142. Accessed: Jan. 28, 2024. [Online]. Available: <http://arxiv.org/abs/1401.2142>
- [46] C. Durr and P. Hoyer, "A Quantum Algorithm for Finding the Minimum," Jan. 07, 1999, *arXiv*: arXiv:quant-ph/9607014. doi: 10.48550/arXiv.quant-ph/9607014.
- [47] J. Li, S. Lin, K. Yu, and G. Guo, "Quantum K-nearest neighbor classification algorithm based on Hamming distance," *Quantum Inf Process*, vol. 21, no. 1, p. 18, Jan. 2022, doi: 10.1007/s11128-021-03361-0.
- [48] Y. Ruan, X. Xue, H. Liu, J. Tan, and X. Li, "Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance," *Int J Theor Phys*, vol. 56, no. 11, pp. 3496–3507, Nov. 2017, doi: 10.1007/s10773-017-3514-4.
- [49] A. M. Fahim, A. M. Salem, F. A. Torkey, and M. A. Ramadan, "An efficient enhanced k-means clustering algorithm," *J. Zhejiang Univ. - Sci. A*, vol. 7, no. 10, pp. 1626–1633, Oct. 2006, doi: 10.1631/jzus.2006.A1626.
- [50] S. Na, L. Xumin, and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," in *2010 Third International Symposium on*

- Intelligent Information Technology and Security Informatics*, Jian, China: IEEE, Apr. 2010, pp. 63–67. doi: 10.1109/IITSI.2010.74.
- [51] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, “A quantum-inspired genetic algorithm for k-means clustering,” *Expert Systems with Applications*, vol. 37, no. 7, pp. 4966–4973, Jul. 2010, doi: 10.1016/j.eswa.2009.12.017.
 - [52] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, “q-means: A quantum algorithm for unsupervised machine learning,” 2018, doi: 10.48550/ARXIV.1812.03584.
 - [53] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Phys. Rev. Lett.*, vol. 113, no. 13, p. 130503, Sep. 2014, doi: 10.1103/PhysRevLett.113.130503.
 - [54] L. Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning,” *SIP*, vol. 3, no. 1, 2014, doi: 10.1017/atsip.2013.9.
 - [55] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
 - [56] M. H. Gordon, M. Cerezo, L. Cincio, and P. J. Coles, “Covariance matrix preparation for quantum principal component analysis,” 2022, doi: 10.48550/ARXIV.2204.03495.
 - [57] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum principal component analysis,” *Nature Phys*, vol. 10, no. 9, pp. 631–633, Sep. 2014, doi: 10.1038/nphys3029.
 - [58] N. A. Nghiem, “New Quantum Algorithm for Principal Component Analysis,” Jan. 14, 2025, *arXiv*: arXiv:2501.07891. doi: 10.48550/arXiv.2501.07891.
 - [59] E. Tang, “Quantum Principal Component Analysis Only Achieves an Exponential Speedup Because of Its State Preparation Assumptions,” *Phys. Rev. Lett.*, vol. 127, no. 6, p. 060503, Aug. 2021, doi: 10.1103/PhysRevLett.127.060503.
 - [60] Z. Lin *et al.*, “Hardware-efficient quantum principal component analysis for medical image recognition,” *Front. Phys.*, vol. 19, no. 5, p. 51202, Oct. 2024, doi: 10.1007/s11467-024-1391-x.
 - [61] T. H. Su, S. Shresthamali, and M. Kondo, “Quantum framework for Reinforcement Learning: Integrating Markov decision process, quantum arithmetic, and trajectory search,” *Phys. Rev. A*, May 2025, doi: 10.1103/5lfr-xb8m.
 - [62] N. Meyer, C. Ufrecht, M. Periyasamy, D. D. Scherer, A. Plinge, and C. Mutschler, “A Survey on Quantum Reinforcement Learning,” Mar. 08, 2024, *arXiv*: arXiv:2211.03464. doi: 10.48550/arXiv.2211.03464.
 - [63] V. Dunjko, J. M. Taylor, and H. J. Briegel, “Quantum-Enhanced Machine Learning,” *Phys. Rev. Lett.*, vol. 117, no. 13, p. 130501, Sep. 2016, doi: 10.1103/PhysRevLett.117.130501.
 - [64] D. Dong, C. Chen, H. Li, and T.-J. Tarn, “Quantum reinforcement learning,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 5, pp. 1207–1220, Oct. 2008, doi: 10.1109/TSMCB.2008.925743.
 - [65] G. Kruse, R. Coelho, A. Rosskopf, R. Wille, and J. M. Lorenz, “Benchmarking Quantum Reinforcement Learning,” in *Proceedings of the 17th International Conference on Agents and Artificial Intelligence*, 2025, pp. 773–782. doi: 10.5220/0013393200003890.
 - [66] V. Dunjko, J. M. Taylor, and H. J. Briegel, “Advances in Quantum Reinforcement Learning,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2017, pp. 282–287. doi: 10.1109/SMC.2017.8122616.
 - [67] S. Y.-C. Chen, “An Introduction to Quantum Reinforcement Learning (QRL),” Sep. 09, 2024, *arXiv*: arXiv:2409.05846. doi: 10.48550/arXiv.2409.05846.
 - [68] S. Jerbi, A. Cornelissen, M. Ozols, and V. Dunjko, “Quantum policy gradient algorithms,” *LIPICs, Volume 266, TQC 2023*, vol. 266, p. 13:1–13:24, 2023, doi: 10.4230/LIPICs.TQC.2023.13.

- [69] T.-A. Drăgan, M. Monnet, C. B. Mendl, and J. M. Lorenz, “Quantum Reinforcement Learning for Solving a Stochastic Frozen Lake Environment and the Impact of Quantum Architecture Choices,” Dec. 15, 2022, *arXiv*: arXiv:2212.07932. doi: 10.48550/arXiv.2212.07932.
- [70] A. Sannia, A. Giordano, N. L. Gullo, C. Mastroianni, and F. Plastina, “A hybrid classical-quantum approach to speed-up Q-learning,” *Sci Rep*, vol. 13, no. 1, p. 3913, Mar. 2023, doi: 10.1038/s41598-023-30990-5.
- [71] T. H. Su, S. Shresthamali, and M. Kondo, “Quantum framework for Reinforcement Learning: Integrating Markov decision process, quantum arithmetic, and trajectory search,” *Phys. Rev. A*, vol. 111, no. 6, p. 062421, Jun. 2025, doi: 10.1103/5lfr-xb8m.
- [72] V. Havlíček *et al.*, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: 10.1038/s41586-019-0980-2.
- [73] R. Mengoni and A. Di Pierro, “Kernel methods in Quantum Machine Learning,” *Quantum Mach. Intell.*, vol. 1, no. 3–4, pp. 65–71, Dec. 2019, doi: 10.1007/s42484-019-00007-4.
- [74] M. Schuld, “Supervised quantum machine learning models are kernel methods,” Apr. 17, 2021, *arXiv*: arXiv:2101.11020. doi: 10.48550/arXiv.2101.11020.
- [75] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, “Quantum machine learning beyond kernel methods,” *Nat Commun*, vol. 14, no. 1, p. 517, Jan. 2023, doi: 10.1038/s41467-023-36159-y.
- [76] L. J. Henderson, R. Goel, and S. Shrapnel, “Quantum Kernel Machine Learning With Continuous Variables,” *Quantum*, vol. 8, p. 1570, Dec. 2024, doi: 10.22331/q-2024-12-17-1570.
- [77] P. Rodriguez-Grasa, Y. Ban, and M. Sanz, “Neural quantum kernels: Training quantum kernels with quantum neural networks,” *Phys. Rev. Research*, vol. 7, no. 2, p. 023269, Jun. 2025, doi: 10.1103/xphb-x2g4.
- [78] J. Tomasi, S. Anthoine, and H. Kadri, “Benign Overfitting with Quantum Kernels,” Mar. 24, 2025, *arXiv*: arXiv:2503.17020. doi: 10.48550/arXiv.2503.17020.
- [79] A. Cho, “Google takes key step toward an error-free quantum computer,” *Science*, vol. 386, no. 6727, pp. 1204–1204, Dec. 2024, doi: 10.1126/science.adv2421.
- [80] R. F. Spivey *et al.*, “High-Stability Cryogenic System for Quantum Computing With Compact Packaged Ion Traps,” *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–11, 2022, doi: 10.1109/TQE.2021.3125926.
- [81] A. P. M. Place *et al.*, “New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds,” *Nat Commun*, vol. 12, no. 1, p. 1779, Mar. 2021, doi: 10.1038/s41467-021-22030-5.
- [82] R. Acharya *et al.*, “Quantum error correction below the surface code threshold,” *Nature*, vol. 638, no. 8052, pp. 920–926, Feb. 2025, doi: 10.1038/s41586-024-08449-y.
- [83] H. Paik *et al.*, “Observation of High Coherence in Josephson Junction Qubits Measured in a Three-Dimensional Circuit QED Architecture,” *Phys. Rev. Lett.*, vol. 107, no. 24, p. 240501, Dec. 2011, doi: 10.1103/PhysRevLett.107.240501.
- [84] A. Aaliray and H. Mohammadi, “Quantum speed limit time in two-qubit system by dynamical decoupling method,” *Sci Rep*, vol. 15, no. 1, p. 8338, Mar. 2025, doi: 10.1038/s41598-025-93348-z.
- [85] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes,” *Nat Commun*, vol. 9, no. 1, p. 4812, Nov. 2018, doi: 10.1038/s41467-018-07090-4.
- [86] M. Larocca *et al.*, “A Review of Barren Plateaus in Variational Quantum Computing,” 2024, *arXiv*. doi: 10.48550/ARXIV.2405.00781.

- [87] J. Cunningham and J. Zhuang, “Investigating and Mitigating Barren Plateaus in Variational Quantum Circuits: A Survey,” Feb. 09, 2025, *arXiv*: arXiv:2407.17706. doi: 10.48550/arXiv.2407.17706.
- [88] J. Cunningham and J. Zhuang, “Investigating and Mitigating Barren Plateaus in Variational Quantum Circuits: A Survey,” Feb. 09, 2025, *arXiv*: arXiv:2407.17706. doi: 10.48550/arXiv.2407.17706.
- [89] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, “Layerwise learning for quantum neural networks,” *Quantum Mach. Intell.*, vol. 3, no. 1, p. 5, Jun. 2021, doi: 10.1007/s42484-020-00036-4.
- [90] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, “An initialization strategy for addressing barren plateaus in parametrized quantum circuits,” *Quantum*, vol. 3, p. 214, Dec. 2019, doi: 10.22331/q-2019-12-09-214.
- [91] V. Bergholm *et al.*, “PennyLane: Automatic differentiation of hybrid quantum-classical computations,” Jul. 29, 2022, *arXiv*: arXiv:1811.04968. Accessed: Jan. 21, 2024. [Online]. Available: <http://arxiv.org/abs/1811.04968>
- [92] F. Bova, A. Goldfarb, and R. G. Melko, “Commercial applications of quantum computing,” *EPJ Quantum Technol.*, vol. 8, no. 1, p. 2, Dec. 2021, doi: 10.1140/epjqt/s40507-021-00091-1.
- [93] P. Rebentrost, B. Gupt, and T. R. Bromley, “Quantum computational finance: Monte Carlo pricing of financial derivatives,” *Phys. Rev. A*, vol. 98, no. 2, p. 022321, Aug. 2018, doi: 10.1103/PhysRevA.98.022321.
- [94] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, “Simulated Quantum Computation of Molecular Energies,” *Science*, vol. 309, no. 5741, pp. 1704–1707, Sep. 2005, doi: 10.1126/science.1113479.
- [95] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, “Quantum Internet: Networking Challenges in Distributed Quantum Computing,” *IEEE Network*, vol. 34, no. 1, pp. 137–143, Jan. 2020, doi: 10.1109/MNET.001.1900092.
- [96] S. S. Gill *et al.*, “Quantum computing: A taxonomy, systematic review and future directions,” *Softw Pract Exp*, vol. 52, no. 1, pp. 66–114, Jan. 2022, doi: 10.1002/spe.3039.