

Review

Not peer-reviewed version

---

# A Technical Review of Quantum Computing Use-Cases for Finance and Economics

---

Manqoba Q. Hlatshwayo<sup>\*</sup>, Manav Babel, Dalila Islas-Sanchez, Konstantinos Georgopoulos

Posted Date: 24 November 2025

doi: 10.20944/preprints202511.1802.v1

Keywords: quantum computing; quantum algorithm; finance; economics



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](https://creativecommons.org/licenses/by/4.0/), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# A Technical Review of Quantum Computing Use-Cases for Finance and Economics

Manqoba Q. Hlatshwayo \*<sup>1</sup>, Manav Babel<sup>2</sup>, Dalila Islas-Sanchez<sup>3</sup>  
and Konstantinos Georgopoulos<sup>4</sup>

National Quantum Computing Centre, UK

\* Correspondence: manqoba.hlatshwayo@stfc.ac.uk

## Abstract

Quantum computing has been rapidly evolving as a field, with innovations driven by industry, academia, and government institutions. The technology has the potential to accelerate computation for solving complex problems across multiple industrial sectors. Finance and economics, with many problems exhibiting computationally heavy requirements, is a high-profile sector where quantum computing could have a significant impact. Therefore, it is important to identify and understand to what extent the technology could find utility in the sector. This technical review is written for quantum applications researchers, quantitative analysts in finance and economics, and researchers in related mathematical sciences. It is divided into two parts: (i) survey of quantum algorithms pertinent to problems in finance and economics, and (ii) mapping of several use-cases in the sector to the potential quantum algorithms presented in part (i). We discuss some challenges on the pathway to achieving quantum advantage. Ultimately, this review aims to be a catalyst for interdisciplinary research that will accelerate the advent of practical advantage of quantum technologies to solve complex problems in this sector.

**Keywords:** quantum computing; quantum algorithm; finance; economics

## 1. Introduction

Quantum computing was first theorised in the 1970s and the new research field gained more momentum in the 1980s [1]. At its core, quantum computing leverages the principles of quantum physics to process information. For problems that are intractable for classical computing, like simulation of complex quantum many-body systems, quantum computing is expected to solve them efficiently in time-frame that scales linear with the system size [2,3]. In contrast, classical methods for such problems has computational times to solution that scales exponential with system size. Furthermore, it was theorise that quantum computing could to speed-up computations for classically tractable problems like searching [4]. The landscape of proposed quantum algorithms that have speed-ups over their classical counterparts, ranges from quadratic [4,5] to exponential [6,7]. Such theoretical discoveries along with the rapid development of quantum hardware [8] has led many researchers to believe that quantum computing will have a significant impact across many disciplines and industry sectors.

In particular, finance and economics, where complex calculations, large data-sets, and optimisation problems are ubiquitous, quantum computing could drastically improve the computational efficiency in the sector [9]. For example, intricate models in finance for risk assessment, portfolio optimisation, fraud detection, and high-frequency trading, that generally require substantial use of high performance computing can potentially benefit from quantum computing. Similarly, economic modelling and simulation (particularly in macroeconomic forecasting, game theory, and market equilibrium analysis) could benefit from integration of quantum processors in the computational pipeline. The expected gains with quantum computing methods is two fold: (i) Large-scale, high-dimensional problems can be handled efficiently without resorting to many approximations and truncation as typically

needed for classical computing approaches. (ii) The computational time to solution can be significantly reduced and accuracy improved for some categories of problems [10]. However, there are a few current bottlenecks in realising these potential gains which include the challenge of efficiently loading and reading out data from a quantum computer [11,12].

This review explores some of the promising use-cases in finance and economics that quantum computing could possibly have a significant impact. We discuss the theoretical foundations of pertinent quantum algorithms, recent proof-of-concept implementations, and potential future implications. By examining these specific use-cases, such as option pricing and risk management, this review highlights both the opportunities and limitations of quantum computing in the context of the aforementioned sectors. We aim to provide an objective analysis, providing a technical review of key quantum algorithms, corresponding use-cases, and the overall state-of-the-art of quantum computing for finance and economics.

## 2. Motivation and Contributions

This review seeks to provide a bridge between use-cases, as encountered by practitioners in finance and economics, and their potential solutions using quantum computing. This survey contributes towards stimulating interdisciplinary research to accelerate the development and demonstration of quantum computing *advantage* for applications in this sectors.

What is meant by *advantage* in quantum computing is often debated as there is, currently, no unique definition for it. There are generally three terms commonly used in association with *quantum advantage*:

- *Computational Advantage* - also referred to as *Quantum Supremacy* [13], is a point where quantum computers can efficiently perform a task that is intractable for a classical computer. The task does not have to be directly useful for real-world applications, such as efficient random sampling from a particular probability distribution [14–17].
- *Quantum Utility* - a point when quantum computers can solve a problem more efficiently than classical brute force methods. This definition was first introduced by IBM Quantum in association with their work in simulating spin systems [18].
- *Practical Quantum Advantage* - a point where quantum computers can solve a problem more efficiently than the most advanced classical methods. In this case, the problem/task in question has real-world practical application(s) and impact in areas such as logistics, healthcare, finance and economics. Therefore, Practical Quantum Advantage (PQA) would highlight quantum computing as a “competitive technology” for solving real-world problems, boosting adoption.

Therefore, most researchers consider PQA a monumental milestone that will unlock the benefits of quantum computing to growth of economy and society at large. Oxford Economics has quantified these benefits as an economy-wide productivity boost which is projected to be up to 8.3% by 2055 in the UK, with initial gains from as early as 2034 [19]. Under reasonable assumptions, this is equivalent to each worker in the UK producing an extra one and a half weeks’ worth of output annually, without putting in any additional hours [19]. Furthermore, the report estimate that by 2055, the quantum computing industry will contribute between £5.9 billion and £12.9 billion in total gross value added to the UK’s GDP [19]. Globally, McKinsey [20] has projected that applications of quantum technologies in finance will contribute a total value of \$622 billion by year 2035. Hence, some analysts have predicted that the financial services sector will be one of the first industries to reap the benefits of quantum technologies [21].

Since the general expectation is that quantum computing will generate a lot of value both in the UK and globally, the key question is what is a viable path towards PQA? What are the engineering and algorithmic hurdles to overcome to achieve PQA? And what could accelerate the collaborative effort towards that common goal? This review seeks to contribute towards answering these questions by providing:

1. A survey of state-of-the-art quantum algorithms with potential applications in finance and economics.

2. A mathematical outline of use-cases in the aforementioned sectors and promising quantum approaches towards an efficient solution. Note that this review is among the first to extend the scope of such a survey to cover use-cases in economics.
3. A discussion of the technical challenges towards realising PQA of quantum computing application in these sectors.

The following subsections outlines how our contribution adds upon previously published related survey reports and the benefits of the structure this review.

### 3. Related Survey Reports

The interest in quantum computing use-cases in finance is evident by the rapidly growing publications and investment in the field. The total number of papers, articles, and theses written on quantum computing for finance has grown from under 2,000 in 2014 to over 10,000 in recent years [22]. Commensurate with the growth of the field, regular survey reports of quantum computing in finance progressed from general overviews in the years 2019 to 2020, to specialised applications in Quantum Machine Learning and financial modelling in the years 2021 to 2024. This is a reflection of the field shifting toward practical implementations, with increasing focus on derivatives pricing, risk assessment, and quantum-enhanced financial strategies. Table 1 presents a sample list of related survey reports of quantum computing applications in finance from 2019 to 2025.

**Table 1.** Related survey papers of quantum computing applications in finance.

Year	First Author	Title of Publication
2019	Orús [9]	Quantum computing for finance: Overview and prospects
2020	Egger [23]	Quantum Computing for Finance: State-of-the-Art and Future Prospects
2020	Alcazar [24]	Classical versus quantum models in Machine Learning: insights from a finance application
2020	Hull [25,26]	Quantum Technology for Economists
2021	Pistoia [27]	Quantum Machine Learning for Finance ICCAD Special Session Paper
2022	García [28]	Systematic Literature Review: Quantum Machine Learning and its applications
2022	Albareti [29]	A Structured Survey of Quantum Computing for the Financial Industry
2022	Herman [30]	A Survey of Quantum Computing for Finance
2022	Gómez [31]	A Survey on Quantum Computational Finance for Derivatives Pricing and VaR
2023	Herman [30]	Quantum computing for finance
2023	Chang [32]	The Prospects of Quantum Computing for Quantitative Finance and Beyond
2023	Naik [33]	From Portfolio Optimization to Quantum Blockchain and Security: A Systematic Review of Quantum Computing in Finance
2023	Saxena [34]	Financial modelling using quantum computing
2024	Jacquier [35]	Quantum Machine Learning and optimisation in finance
2024	Claudiu [36]	Enhancing the Financial Sector with Quantum Computing: A Comprehensive Review of Current and Future Applications
2024	Lu [37]	Quantum financing system: A survey on quantum algorithms, potential scenarios and open research issues

Table 1 – continued from previous page

Year	First Author et al.	Title of Publication
2024	Atadoga [38]	The Intersection of Artificial Intelligence And Quantum Computing In Financial Markets: A Critical Review
2024	Gujju [39]	Quantum Machine Learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications
2024	Bunescu [40]	Modern finance through quantum computing—A systematic literature review
2024	Mironowicz [41]	Applications of Quantum Machine Learning for Quantitative Finance
2024	Auer [22]	Quantum computing and the financial system
2025	Corli [42]	Quantum Machine Learning algorithms for anomaly detection: A review

Each of the survey reports above is written for a particular audience as is reflected in the depth of technical details covered and emphasis of either quantum algorithms or the use-cases in finance. In this review, our approach is to present separately the technical depth of the key quantum algorithms and the mathematical formulation of the use-cases in finance and economics. We hope this approach will spur interdisciplinary collaboration that will accelerate the advent of PQA for industrial applications.

#### 4. Structure of the Review

The structure of the review resembles the mapping between use-cases and problem domains shown in Figure 1. Thus, we have divided this report into three parts:

**Part I: Quantum algorithms** - a review of the key quantum algorithms with potential PQA in finance and economics. These are grouped into four problem domains:

- (a) Simulation algorithms are presented in Chapter (14), including Quantum Monte Carlo Integration and quantum solvers for Stochastic Differential Equations.
- (b) Optimisation algorithms are presented in Chapter (17), particularly for combinatorial and convex optimisation problems with state-of-the-art examples.
- (c) Quantum Machine Learning algorithms are presented in Chapter (20), which are quantum extensions of classical Machine Learning techniques for supervised, unsupervised, and reinforcement learning.
- (d) Quantum cryptography methods are presented in in Chapter (24), including discussion of future expected threats to cybersecurity posed by quantum computers once the technology reaches full maturity, and some quantum-safe protocols for data encryption and communication.

The pertinent primitive algorithms that feature as subroutines to quantum algorithms in the above problem domains are outlined in Chapter (6).

**Part II: Use-cases in finance and economics** - a review of the mathematical formulations of potential use-cases of quantum computing in finance and economics. These are grouped into three parts:

- (a) Banking and Investment - presented in Chapter (30), we identify some of the computational bottlenecks for classical methods in banking and investment problems, and use-cases where quantum computing can potentially have an advantage. These include pricing assets and derivatives, portfolio optimisation, hedging, and arbitrage.
- (b) Risk Management and Cybersecurity - presented in Chapter (35), we identify some of the simulation and security challenges in risk management and cy-

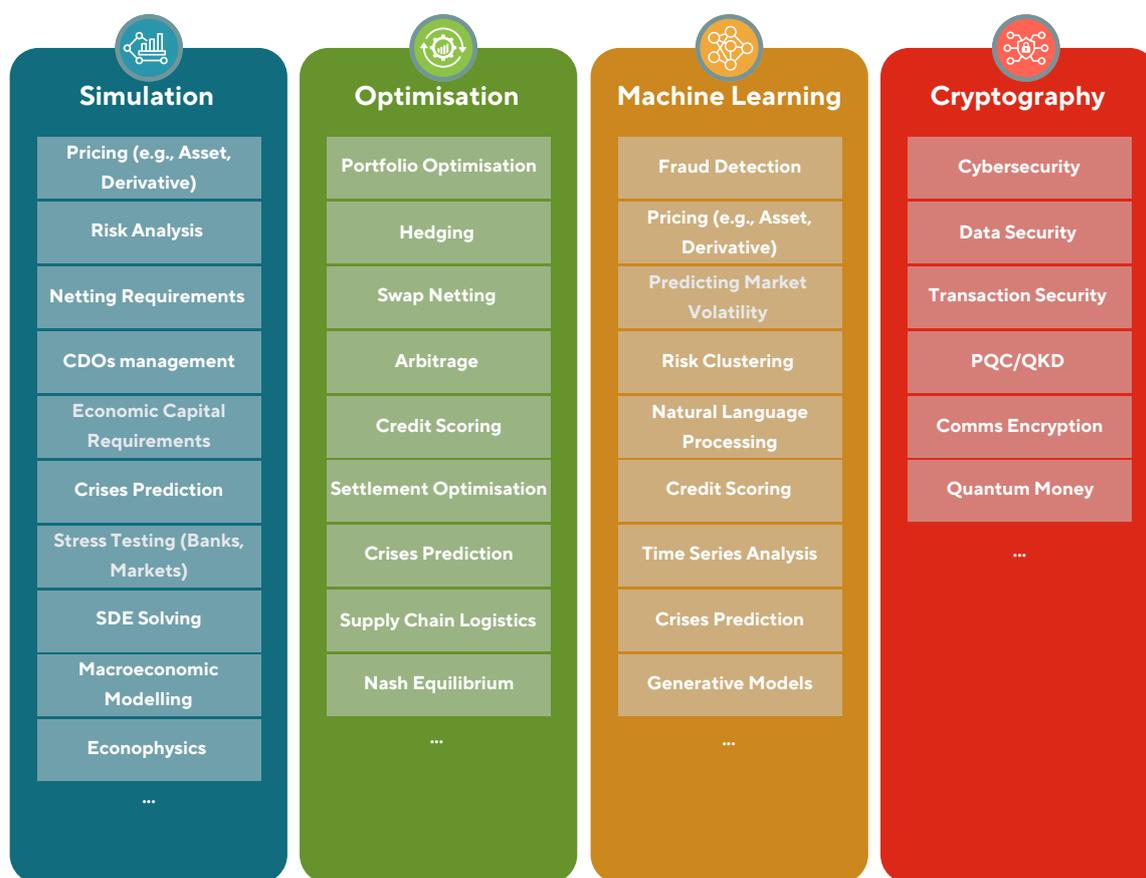
bersecurity, respectively. Similarly to banking and investment, we present the technical formulations of use-cases of quantum computing in this category. This include quantum approaches for risk analysis (Value-at-Risk, credit scoring, etc.) and fraud detection.

- (c) Economics - presented in Chapter (40), the mathematical formulation of potential use-cases of quantum technologies for problems arising in economics are outlined. These include quantum money and macro-economic forecasting.

**Part III: Summary and outlook** - a summary of this survey report, in which we reiterate the benefits to financial organisations in working towards quantum readiness, and provide an outlook of one possible route towards PQA.

Note:

This review is not a comprehensive review of all quantum algorithms nor use-cases in finance and economics. We have selected a subset of use-cases that represent areas of potential high impact with some current published research activity in the form of proof-of-concept proposals and/or demonstrations. Similarly, we have chosen to give technical details for simpler formulations of quantum algorithms for a problem domain and mention more complex approaches in passing. Furthermore, this review is also not intended as a primer on quantum computing or quantum information theory. In the quantum-specific sections, we assume a level of familiarity with quantum mechanics and related terminology, which other sources will be able to provide in more details. In the next section, we will summarise some of the notation used in this review for the reader's convenience. Finally, we hope that financial organisations interested incorporating quantum technologies can work with quantum applications experts to develop relevant solutions using this survey and others as a starting point.



**Figure 1.** An overview of quantum computing use-cases in finance and economics categorised by NQCC's priority problem category [43].

## 5. Notation

In this section we summarise some of the key notation used in this review for clarity. This is not an exhaustive list and some of the notation can be overloaded when the meaning is clear in the context. For example, in some parts of the text, the Hamiltonian may be denoted as  $H$  and it should not be confused as the Hadamard gate. Similarly, the Big-O notation may be denoted as  $O(\cdot)$  or  $\mathcal{O}(\cdot)$  which both mean the same thing. Table 2 give a sample of common notation in quantum computing and analysis of quantum algorithms that are used in this review. Since this review is not an introductory text in the field, a reader who is unfamiliar with these notations can refer to Ref. [44] for more information.

**Table 2.** Common notation in quantum computing and analysis of quantum algorithms.

Symbol	Meaning	Description
$ \psi\rangle, \langle\psi $	Dirac bra-ket notation for general quantum state and its adjoint, respectively.	A general qubit state: $ \psi\rangle = \alpha 0\rangle + \beta 1\rangle$ , where $ \alpha ^2 +  \beta ^2 = 1$ . Here, the basis vectors are given by $ 0\rangle = [1\ 0]^T$ , $ 1\rangle = [0\ 1]^T$
$\langle\phi \psi\rangle$	Inner product	Measures the overlap or amplitude between two quantum states.
$U$	Unitary operator	Reversible transformation that preserves norm: $U^\dagger U = I$ .
$\mathcal{H}$	Hamiltonian	Operator that represents the total energy of a quantum system. It determines the its time evolution via the unitary $U = e^{-i\mathcal{H}t}$ .
$H$	Hadamard gate	Puts a qubit into superposition: $H 0\rangle = \frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ .
$X, Y, Z, I$	Pauli gates	Single-qubit gates: $X$ (bit flip), $Z$ (phase flip), $Y$ (both flip), $I$ (identity).
$CX, CNOT$	Controlled-NOT gate	Two-qubit gate that flips the target if control qubit is $ 1\rangle$ .
$T$	$T$ -gate given by $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	A non-Clifford single-qubit gate that plays two key roles: (i) universality for the set of Clifford gates plus $T$ -gate [44], and (ii) efficient decomposition of Toffoli gates [45] the equivalent of classical NAND gates [46].
$\otimes, \text{Tr}$	Tensor product and Trace	Combines states: $ a\rangle \otimes  b\rangle =  ab\rangle$ . Trace gives sum of matrix diagonal entries; used in measurement and subsystems.
$O(\cdot)$	Big-O notation	Describes asymptotic upper bounds on complexity of an algorithm.
$\Omega(\cdot)$	Big-Omega notation	Describes asymptotic lower bounds on complexity of an algorithm.
$\Theta(\cdot)$	Big-Theta notation	Describes asymptotic average or tight (upper and lower) bounds on complexity of an algorithm.
meas or $M$	Measurement	Collapses qubit to classical bit based on amplitude probabilities.

## 6. Primitive Quantum Algorithms

This chapter offers an introduction to a set of quantum algorithms, here referred to as primitive algorithms, that are used as the *basis* or *fundamental building blocks* from which more complex quantum algorithms, models or frameworks are constructed [47]. As primitive algorithms do not directly solve an end-to-end problem, groups of these fundamental algorithms (also called subroutines) are needed to create a wide range of high-end quantum applications [48] solving several use-cases.

## 7. Quantum Phase Estimation

The first quantum subroutine we will review, which is also one of the earliest to be discovered, is the Quantum Phase Estimation (QPE) which was first formalised in 1995 by Kitaev<sup>1</sup> [50]. This subroutine has been widely used by many other algorithms including Shor's quantum algorithm for factoring large numbers [49,51], Hamiltonian diagonalisation [52], and solving systems of linear equations [6].

QPE serves as a key building block for quantum algorithms of which some have a potential quantum speedup that is exponential due to it. The objective of the algorithm is the following [44]: given a unitary operator  $U$ , estimate the phase  $\theta$  in the representation

$$U |\psi\rangle = e^{2\pi i\theta} |\psi\rangle \quad (1)$$

Here  $|\psi\rangle$  is an eigenvector and  $e^{2\pi i\theta}$  is the corresponding eigenvalue. In order to implement the QPE, the quantum system needs to be separated into registers: the *state register* that contains  $|\psi\rangle$  and the *phase register* that contains the encoded estimate of the phase  $\theta$ .

The QPE subroutine is constructed as follows:

1. **Setup:** initialise the state register with the state  $|\psi\rangle$ . The additional set of  $n$  qubits that form the phase register are set in state  $|0^{\otimes n}\rangle$ . After initialisation, the global state of the system will be:

$$|\psi_0\rangle = |0\rangle^{\otimes n} |\psi\rangle$$

2. **Superposition:** an  $n$ -bit Hadamard gate is applied on the phase register, leaving the global state as:

$$|\psi_1\rangle = \frac{1}{2^{n/2}} (|0\rangle + |1\rangle)^{\otimes n} |\psi\rangle$$

3. **Controlled Unitary Operations:** consider a controlled unitary  $U_C$  that applies the unitary operator  $U$  on the target register (i.e., the phase register) only if its corresponding control qubit is  $|1\rangle$  [44]. Since  $U$  is a unitary operator with eigenvector  $|\psi\rangle$  such that  $U |\psi\rangle = e^{2\pi i\theta} |\psi\rangle$ , it follows that:

$$U^{2^j} |\psi\rangle = U^{2^j-1} U |\psi\rangle = U^{2^j-1} e^{2\pi i\theta} |\psi\rangle = \dots = e^{2\pi i\theta \cdot 2^j} |\psi\rangle.$$

Applying all the  $n$ -controlled operations  $U_C^{2^j}$  with  $0 \leq j \leq n-1$ , and using the relation  $|0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi i\theta} |\psi\rangle = (|0\rangle + e^{2\pi i\theta} |1\rangle) \otimes |\psi\rangle$ , leads to the global state:

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2^{n/2}} \left( |0\rangle + e^{2\pi i\theta 2^{n-1}} |1\rangle \right) \otimes \dots \otimes \left( |0\rangle + e^{2\pi i\theta 2^1} |1\rangle \right) \otimes \left( |0\rangle + e^{2\pi i\theta 2^0} |1\rangle \right) \otimes |\psi\rangle \\ &= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k} |k\rangle \otimes |\psi\rangle \end{aligned}$$

where  $k$  denotes the integer representation of  $n$ -bit binary numbers.

4. **Inverse Fourier Transform:** The Quantum Fourier Transform (QFT) maps an  $n$ -qubit input state  $|x\rangle$  into:

$$\begin{aligned} \text{QFT } |x\rangle &= \frac{1}{2^{n/2}} \left( |0\rangle + e^{\frac{2\pi i}{2} x} |1\rangle \right) \otimes \left( |0\rangle + e^{\frac{2\pi i}{2^2} x} |1\rangle \right) \otimes \dots \\ &\quad \otimes \left( |0\rangle + e^{\frac{2\pi i}{2^{n-1}} x} |1\rangle \right) \otimes \left( |0\rangle + e^{\frac{2\pi i}{2^n} x} |1\rangle \right). \end{aligned}$$

<sup>1</sup> Note that the idea of Quantum Fourier Transform was first used in 1994 by Peter Shor [49], but he did not explicitly formalise it into QPE.

It is evident that the expression of the global state  $|\psi_2\rangle$  is the result of applying QFT on the global expression  $|\psi_1\rangle$  of Step 2. Therefore, to recover the state  $|2^n\theta\rangle$ , an inverse Fourier transform is applied on the phase register [53]. Doing so, it is found that

$$|\psi_3\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k} |k\rangle \otimes |\psi\rangle \xrightarrow{\text{QFT}_n^{-1}} \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-\frac{2\pi i k}{2^n}(x-2^n\theta)} |x\rangle \otimes |\psi\rangle$$

5. **Measurement:** the above expression peaks near  $x = 2^n\theta$ . For the case when  $2^n\theta$  is an integer, measuring in the computational basis gives the phase in the phase register with high probability, as the global state now is:

$$|\psi_4\rangle = |2^n\theta\rangle \otimes |\psi\rangle.$$

For the case when  $2^n\theta$  is not an integer, it can be shown that the above expression still peaks near  $x = 2^n\theta$  with probability at least  $4/\pi^2 \approx 40\%$  [44].

---

### Algorithm 1 Quantum Phase Estimation

---

**Initialising.** Initialise the quantum system. There exist two qubit registers, the phase register initialised in state  $|0\rangle^{\otimes n}$ , and the state register initialised in  $|\psi\rangle$ .

**Superposing.** Apply Hadamard gates on the phase register.

**Applying the unitary.** Apply the controlled unitary operation as shown in the third step of the mathematical analysis.

**Inverse QFT.** Apply inverse quantum Fourier transform in order to decode the state of the phase register.

**Measure.** Measure the phase register in order to get an approximation of the desired phase,  $\theta$ .

---

Algorithm (1) summarises the key steps of the QPE subroutine which scales as  $\mathcal{O}(1/\delta)$ , for desired additive error  $\delta$  [50]. The standard QPE circuit requires  $\mathcal{O}(2^n)$  controlled- $U_C$  operations, which can be a resource challenge for current Noisy Intermediate-Scale Quantum (NISQ) hardware. Hence, QPE is considered a Universal Fault-Tolerant (UFT) quantum algorithm which requires Quantum Error Correction (QEC) to obtain highly accurate phase estimates. In the NISQ-era, practitioners often employ approximate versions of QPE with reduced circuit depth and error-mitigation methods to suppress effects of noise. These variants of QPE includes the Iterative Quantum Phase Estimation (IQPE) which uses a single control qubit repeatedly and employs classical post-processing to update estimates of  $\theta$  [54]. IQPE typically has the same asymptotic error scaling but requires fewer qubits [54], making it more suitable for near-term quantum devices. There has also been proposals for alternatives to QPE such as the quantum eigenvalue estimation via time series analysis [55] and the phase estimation of local Hamiltonians [56]. These methods used expectation values of the evolution operator for various times to estimate the eigenvalues of a Hamiltonian which makes them more feasible to implement on NISQ hardware.

## 8. Quantum Amplitude Algorithms

Another set of widely used quantum subroutines introduced by Brassard et al. [57] are the Quantum Amplitude Amplification (QAA) and Quantum Amplitude Estimation (QAE). These algorithms are an extension of the principles of Grover's search algorithm [4] to a broader range of application.

### 8.1. Quantum Amplitude Amplification

The first amplitude subroutine, QAA also used within Grover's search algorithm [4], enhances the probability amplitude of desired outcomes within a quantum algorithm. The QAA generalizes Grover's unordered search to any algorithm with a known success probability [58], enabling a potential quadratic speedup over classical repetition strategies [57].

Consider a quantum algorithm  $\mathcal{A}$  that produces a state

$$\mathcal{A}|0\rangle = |\psi\rangle = \sqrt{1-a}|\psi_0\rangle + \sqrt{a}|\psi_1\rangle, \quad (2)$$

where  $|\psi_1\rangle$  is the ‘good’ subspace (desired states),  $a$  the probability of measuring a ‘good’ state, and  $|\psi_0\rangle$  the ‘bad’ subspace. The amplification process is done using the Grover-like operator  $Q$ , which acts as a rotation and is defined to be [57]

$$Q = -\mathcal{A}\mathcal{S}_0\mathcal{A}^\dagger\mathcal{S}_\chi, \quad (3)$$

where

$$\mathcal{S}_0 = 2|0\rangle\langle 0| - I \quad \text{and} \quad \mathcal{S}_\chi = I - 2|\psi_1\rangle\langle\psi_1|. \quad (4)$$

The operator  $\mathcal{S}_0$  reflects about the initial state which flips the sign of the amplitudes of the good states  $|x\rangle$  if and only if  $|x\rangle = |0\rangle$ . The operator  $\mathcal{S}_\chi$  reflects about the good state  $|\psi_1\rangle$  which flips the sign of  $|x\rangle$  if and only if  $\chi(x) = 1$ . This implies that [57]

$$\mathcal{S}_\chi|x\rangle = -|x\rangle \iff \chi(x) = 1. \quad (5)$$

The probability amplitude of the good state is  $\sin^2((2m+1)\theta_a)$ , where  $\theta_a = \arcsin(\sqrt{a})$ , with  $0 < \theta_a < \pi/2$ , and the optimal  $m$  number of iterations is  $\lfloor \pi/(4\theta_a) \rfloor$ . Applying  $Q$  iteratively to  $|\psi\rangle$ , the state evolves such that after  $m$  applications, the state is

$$Q^m|\psi\rangle = \sin((2m+1)\theta_a)|\psi_1\rangle + \cos((2m+1)\theta_a)|\psi_0\rangle, \quad (6)$$

Thus,  $Q$  rotates the state in the  $\{|\psi_0\rangle, |\psi_1\rangle\}$  plane per each iteration. If  $a$  is known, QAA achieves a quadratic speed-up with  $\Theta(1/\sqrt{a})$  applications of  $\mathcal{A}$  and  $\mathcal{A}^\dagger$ . For unknown  $a$ , quantum search algorithms like one outlined in Section (9) uses exponentially increasing numbers of iterations to find a good solution. This approach is also expected to use same  $\Theta(1/\sqrt{a})$  applications  $\mathcal{A}$  and  $\mathcal{A}^\dagger$  if  $a > 0$  [57]. Algorithm (2) summarises the QAA subroutine using the operator  $Q$  and a general  $\mathcal{A}$  algorithm.

---

#### Algorithm 2 Quantum Amplitude Amplification

---

**Initialise:** Apply a quantum algorithm  $\mathcal{A}$  to  $|0\rangle$  to prepare the state:

$$|\psi\rangle = \sqrt{1-a}|\psi_0\rangle + \sqrt{a}|\psi_1\rangle$$

**Amplify:** Define the amplification operator and apply it to  $m$  number of iterations to rotate the state:

$$Q^m|\psi\rangle = \sin((2m+1)\theta_a)|\psi_1\rangle + \cos((2m+1)\theta_a)|\psi_0\rangle$$

**Measure:** Measure the final state  $Q^m|\psi\rangle$  to obtain a good state.

**Repeat:** If  $a$  is unknown, use exponentially increasing  $m$  until a good state is found.

---

An extension of QAA, called the *variable time amplitude amplification*, which can be decomposed into multiple steps with different stopping times [59] has been applied as a subroutine that improves the complexity of Harrow et al. [6] Quantum Linear Systems Solver. Improvements to the QAA subroutine have been proposed such as the *oblivious amplitude amplification* [60] which handles the issue that occur when it is not possible to perform the required reflection about the state  $|\psi\rangle$ . Another version of QAA is the *fixed-point amplitude amplification* [61] that ensures amplitude amplification happens regardless of an unknown amplitude. These improved versions of QAA are subroutines for Quantum Singular Value Transformation (QSVT) methods [62] and have been recently compared in Hamiltonian simulation problem [63].

## 8.2. Quantum Amplitude Estimation

The second amplitude subroutine, QAE algorithm first proposed by [57], is closely related to QAA and seeks to estimate the probability amplitude of a specific outcome with high precision. It is a key subroutine for a broad class of Monte Carlo-like estimation tasks [64] and offers a potential quadratic quantum speed-up [5].

The QAE can be described in a similar manner as QAA. Consider an algorithm  $\mathcal{A}$  that outputs a state given by Equation (2), the the QAE seeks to produce an estimate for the amplitude  $a$  of the good state  $|\psi_1\rangle$ , where  $a = \langle \psi_1 | \psi_1 \rangle$ . Using the operator  $Q$  given by Equation (3), QAE estimates the value of  $a$  by using QPE to find the eigenvalues of  $Q$  [57]. Hence, both QAA and QAE use the operator  $Q$ , but the former boosts  $a$  for measurement of  $|\psi_1\rangle$ , while the latter quantifies  $a$  itself.

The key idea is to assign the value of  $a$  to an angle  $\theta$  so that the estimate of  $\theta$  to a given precision  $\delta$  using QPE gives us the estimate of  $a$ . This is done by setting  $a = \sin^2(\theta)$  and then applying QPE to the controlled- $Q$  operations, since  $Q$  acts as a rotation in  $\mathcal{H}_\psi$  with eigenvalues  $e^{\pm i2\theta}$ . The estimate  $\tilde{a}$  with  $M$  evaluations has errors bounds given by [57]

$$|\tilde{a} - a| \leq 2\pi k / \sqrt{2a(1-a)/M} + \pi k^2 / M^2, \quad (7)$$

with probability at least  $8/\pi^2$  for  $k = 1$ , and higher for  $k \geq 2$ . Algorithm (3) summarises the QAE that utilises QPE.

---

### Algorithm 3 Quantum Amplitude Estimation

---

**Initialise:** Apply a quantum algorithm  $\mathcal{A}$  to  $|0\rangle$  to prepare the state:  $|\psi\rangle = \sqrt{1-a}|\psi_0\rangle + \sqrt{a}|\psi_1\rangle$  (same as QAA).

**Define:** Define the amplification operator  $Q$  given by Equation (3). The eigenvalues of  $Q$  are  $e^{\pm i2\theta}$  where  $a = \sin^2(\theta)$  for  $\theta \in [0, \pi/2]$ .

**Estimate:** First apply a total of  $M$  controlled- $Q$  operations on state  $|\psi\rangle$  such that

$$\sum_{k=0}^{M-1} |k\rangle Q^k |\psi\rangle.$$

Let  $\phi = \theta/\pi$ , then apply QPE to estimate the eigenphase  $\tilde{\phi} \in [0, 1)$  which we use to recover the amplitude

$$\tilde{a} = \sin^2(\pi\tilde{\phi})$$

**Output:** Return  $\tilde{a}$  as the estimate of  $a$ , with error bounds [57]:

$$|\tilde{a} - a| \leq \frac{2\pi}{\sqrt{M}} \sqrt{2a(1-a)} + \frac{\pi^2}{M^2}$$

with probability at least  $8/\pi^2$  for  $M \geq 1$ .

---

This version of the QAE algorithm is a relatively resource expensive due to the use of QPE which uses many ancilla qubits for a good estimate, multiple controlled- $Q$  operations, and the inverse QFT. Hence, this results into deep circuits which are not feasible to run on near-term devices. Some variants of QAE that do not make use of QPE include the Maximum Likelihood Amplitude Estimation [65], Faster Amplitude Estimation [66], Iterative QAE [67], Variational QAE [68], and low depth algorithms for QAE [64] which are more feasible on near-term devices. An important application of QAE to finance is to perform Monte Carlo integration which will be discussed in more detail in Section (15). Other applications of QAE include general numerical integration [69], estimating the probability of success of a quantum algorithm [70] and approximate counting [71].

## 9. Quantum Unstructured Search (Grover's Algorithm)

First proposed in 1996 by Lov Grover [4], the Quantum Unstructured Search (QUS)<sup>2</sup> algorithm utilises the QAA subroutine to potentially obtain a quadratic quantum speed-up over classical methods for searching an item in an unstructured dataset.

In this section, we describe the key idea behind the QUS algorithm. For a database  $\mathcal{D}$  of  $N = 2^n$  entries, let  $\{x\} \in \mathcal{D}$  be the set of items in an array, and  $w$  the marked item (i.e., the item the routine searches for). Let  $f$  be a function such that  $f(x) = 1$  if and only if  $x = w$  and  $f(x) = 0$  otherwise. The quantum states can be encoded as binary strings  $x \in \{0, 1\}^n$ . One can then define the oracle  $U_f$  acting on a quantum state,  $|x\rangle$ , as

$$U_f |x\rangle \rightarrow (-1)^{f(x)} |x\rangle. \quad (8)$$

Thus, the oracle does nothing to the unmarked quantum states and negates the phase of the marked state  $|w\rangle$ . Geometrically, the unitary matrix of this oracle corresponds to a reflection about the origin for the marked item in an  $N$ -dimensional vector space.

The QAA subroutine provides the quadratic speed-up of QUS by amplifying the amplitude (and thus, the probability) of the target quantum state, making it more likely to appear after measurement. This procedure is summarised in Algorithm (4) and has been shown have a query complexity of  $O(\sqrt{N})$  which is a quadratic speed-up over classical brute force methods of  $O(N)$  [4].

---

### Algorithm 4 Quantum Unstructured Search (Grover's Algorithm)

---

**Initialisation:** Start with  $|0\rangle^{\otimes n}$ , apply Hadamard gates to create a uniform superposition:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

Each state  $|x\rangle$  has amplitude  $1/\sqrt{N}$ .

**Oracle Application:** Apply the oracle  $U_f$ , which flips the phase of solution states, such that for one solution  $|w\rangle$ , the state becomes:

$$U_f |s\rangle = \frac{1}{\sqrt{N}} \left( \sum_{x \neq w} |x\rangle - |w\rangle \right).$$

This marks solutions by changing their sign.

**Diffusion Operator:** Apply the diffusion operator  $D = 2|s\rangle\langle s| - I$ , where  $I$  is the identity. This reflects the state over  $|s\rangle$ , amplifying solution amplitudes:

$$DU_f |s\rangle = 2\langle s|U_f |s\rangle |s\rangle - U_f |s\rangle.$$

The action of the Grover operator  $G = DU_f$  is to amplify the amplitude of the solution state.

**Iteration:** Repeat the Grover operator  $k$  times  $G^k |s\rangle$  and then measure to find highest probability state  $|w\rangle$ . The optimal  $k$  repetitions to maximise the probability of finding the target state(s) is approximately:

$$k = \frac{\pi}{4} \sqrt{\frac{N}{M}}$$

where  $M$  is the number of solutions.

---

Grover's algorithm has been applied as a subroutine for a various applications including speeding-up nested search of structured problems [72], stochastic pure adaptive search for unconstrained global optimization [73], high-energy physics data at the Large Hadron Collider [74] image pattern matching [75], and homomorphic encryption schemes [76]. The QUS algorithm as described here is considered a fault-tolerant quantum algorithm due the deep circuits (required to reach optimal

<sup>2</sup> Commonly known as Grover's algorithm or quantum search algorithm

$k$  iterations) which are infeasible to run successfully on current NISQ hardware. Hence, similarly to the other primitives reviewed here, improvements have been suggested to the QUS to make it amenable to run on near-term quantum hardware. These include the divide-and-conquer strategy and partial diffusion operators to reduce circuit depth and error accumulation [77,78] and variational approaches [79,80].

## 10. Quantum Walks

Random walks have long served as a foundational model in the study of stochastic processes, with applications spanning physics, computer science, biology, and finance. In its classical form, a random walk describes the trajectory of a particle that takes successive steps in random directions, typically governed by a Markov process [81]. Random walks have also been generalised to graphs [82], in which case they can be viewed as finite-state Markov chains over the set of vertices [83]. A simple example is the one-dimensional symmetric random walk, where a particle at position  $x \in \mathbb{Z}$  moves to  $x + 1$  or  $x - 1$  with equal probability at each time step. In finance, the Geometric Brownian Motion (GBM) is a widely used random walks for modelling market stochastic quantities like asset prices [84,85].

The quantum analog of random walks, known as *quantum walks*, emerged in the early 1900s as a framework for modeling quantum dynamics and as a potential basis for quantum algorithms [86]. Unlike classical random walks, quantum walks evolve according to unitary transformations, preserving coherence and allowing for interference between paths. This leads to fundamentally different behavior, including faster spreading and potential algorithmic advantages [87].

Two main formulations of quantum walks exist: *discrete-time* and *continuous-time*. The Discrete-time Quantum Walk (DTQW) was first formalized by Aharonov et al. [86], incorporating a coin space to account for internal degrees of freedom. In contrast, the Continuous-time Quantum Walk (CTQW), introduced by Farhi and Gutmann [88], operates directly on the graph structure without an explicit coin. In this section, we will review the key ideas behind quantum walks, beginning with their mathematical foundations and then explore their distinctive dynamical properties, algorithmic formulations, bottlenecks, and applications.

### 10.1. Discrete-Time Quantum Walks

Consider a walker on a 1D line, flipping a *quantum coin* to decide whether to turn left or right at each step. This represents one of the simplest quantum walk, also referred to as a *qubit walk* on a finite cycle (with  $N$  states) in discrete time [89]. This quantum walk can be described by repeated application of a unitary evolution operator  $U$  which acts on the Hilbert space  $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_S$  composed of the coin qubit ( $C$ ) with basis  $\{|\uparrow\rangle, |\downarrow\rangle\}$  and state space ( $S$ ) with basis  $\{|x\rangle\}$  for  $x \in \{0, 1, \dots, N - 1\}$ , respectively. The unitary  $U$  is defined as [89]:

$$U = S \cdot (C \otimes I), \quad (9)$$

where  $S$  is the conditional shift operator given by [90]

$$S = |\uparrow\rangle\langle\uparrow| \otimes S^+ + |\downarrow\rangle\langle\downarrow| \otimes S^-. \quad (10)$$

The operator  $S^+ |x\rangle \rightarrow |x + 1\rangle$  moves the walker one step to the right, increasing its position, and  $S^- |x\rangle \rightarrow |x - 1\rangle$  to the left, decreasing its position. The quantum coin operator  $C$  is can be chosen to the Hadamard gate  $H$ , such that

$$H |\uparrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \quad \text{and} \quad H |\downarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle). \quad (11)$$

We note that although this coin is *unbiased*<sup>3</sup>, the quantum walk from this coin operator ( $C = H$ ) will be *asymmetric* with bias towards rightward paths. This is because the leftwards path ( $S^-$ ) undergoes

<sup>3</sup> Meaning the states  $|\uparrow\rangle$  and  $|\downarrow\rangle$  of the coin are evenly distributed up to a phase factor

more destructive quantum interference whereas the rightwards path undergoes more constructive quantum interference [91]. This phenomena has no classical analogue and makes quantum walks more expressive than classical random walks [86,87]. The coin represented by the Hadamard operator alongside a step operator, causes the probability amplitude to become distributed over an increasing space in discrete jumps [92]. This *asymmetry* can be overcome by choosing a *balanced* coin operator [87], for example

$$C = Y = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}, \quad (12)$$

which treats both  $|\uparrow\rangle$  and  $|\downarrow\rangle$  the same, independent of the initial coin state.

Intuitively, in each step the qubit first 'splits' into two paths (C puts it in a superposition) and then one 'half' steps leftwards, while the other steps rightwards. This process, when repeated with  $T$ -steps produces a probability distribution that is ballistic with standard deviation  $\sigma_q \sim T$  whereas the classical random walk has a diffusive distribution with standard deviation  $\sigma_c \sim \sqrt{T}$  [87]. This implies the quantum walk propagates quadratically faster than classical walks [87] which means they can explore more of a graph in fewer steps than classical random walks [93].

Generally, quantum walks in discrete time can be viewed as a Markov chain over the edges of a graph and consist of a series of interleaved reflection operators [94,95]. This formulation of quantum walks generalises the QAA and Grover's algorithm discussed in Sections (8.1) and (9), respectively [96,97]. The bipartite character of the graph introduces the *modularity* property of quantum walks (similarly to classical random walks), which is after an even (odd) number of steps, the probability to find the walker on odd (even) positions is always zero [98]. This property allows for simplified analysis of the performance of quantum walks, in particular to two important metrics: the *mixing time* and *hitting time* of random walks on graphs. The mixing time of a graph is the time it takes for a random walk on the graph to become close to the stationary distribution; that is, the point where the walk "forgets" where it started and behaves as if it started from a random node [99]. The hitting time from node  $u$  to node  $v$  is the expected number of steps a random walk starting at  $u$  takes to reach  $v$  for the first time [81]. Quantum walks are shown to potential have a polynomial quantum speed-up in mixing times [89,93,100] and hitting times for Markov-chain-based search algorithms [94,95,101–105] than their classical random walks counterpart. There has been several proof-of-concept implementations of DTQW on NISQ hardware which includes a Hadamard-based on the 4-cycle and 8-cycle [106] and extensions to multi-qubit systems [107,108].

## 10.2. Continuous-Time Quantum Walks

Consider an undirected graph  $G(V, E)$ , where  $V$  is the set of its vertices and  $E$  the set of its edges. In classical mechanics, a diffusion equation can be used which allows probability to leak to or from neighbouring vertices. The number of neighbouring nodes equals the degree of the node  $j$ , or  $\deg(j)$ . This diffusion operation can be expressed as

$$\frac{d}{dt} p_j(t) = \sum_{k \in V} L_{j,k} p_k(t) \quad (13)$$

where  $t$  is an arbitrary continuous time parameter,  $p_j$  is a function that describes the probabilistic exchange between node  $j$  and its neighbouring nodes. The operator  $L$  is symmetric and Hermitian, called the Laplacian of  $G$ , given by

$$L_{j,k} = \begin{cases} -\deg(j) & j = k \\ 1 & (j, k) \in E \\ 0 & \text{otherwise} \end{cases}$$

Thus, since  $L$  is a Hermitian matrix, it can play the role of a Hamiltonian in Schrödinger's equation as

$$i \frac{d}{dt} |\psi(t)\rangle = L |\psi(t)\rangle \quad (14)$$

where, for simplicity it can be assumed that  $\hbar = 1$  and  $|\psi(t)\rangle$  is the state of the system at arbitrary time  $t$ . First introduced by Farhi and Gutmann in [88], Equation (14) represents the Continuous-time Quantum Walk (CTQW). As evident from the above formulation, the CTQW does not need a quantum coin that drives the evolution, unlike the DTQW. This makes CTQW a construct that shares dynamics with the DTQW, but exhibit different behaviour. In contrast to the quantum coin tossing example described in Section (10.1), the walker now moves smoothly (or continuously), guided by an evolution rule (which is driven by the Hamiltonian, typically the graph Laplacian  $L$ ). This evolution follows the Schrödinger Equation (14) and the walk is continuous without distinct steps. Therefore, we can summarise the fundamental differences between the two quantum walks as:

- **State space:** CTQWs require only the vertex space, while DTQWs models need an auxiliary coin space for evolving. As discribed in Section (10.1), the coin operator drives the symmetry or asymmetry of the quantum walk.
- **Dynamics:** CTQWs have continuous Schrödinger evolution (via Hamiltonians), whereas discrete-time walks rely on alternating coin and shift operators per time step.

In addition, the various mixing operators used in Quantum Approximate Optimisation Algorithm, described in Section (18.4) can be viewed as CTQW connecting the feasible solutions [109]. Similarly to DTQWs, the CTQWs can showcase a quadratic advantage in terms of accelerating diffusion (i.e., spreading) compared to classical continuous-time random walks [110]. For some oracular graph problems, the CTQW can theoretically achieve exponential advantage over classical methods [111].

It was shown by Childs that CTQW can be regarded as a universal computational primitive, with any quantum computation encoded in some graph [112]. Later, Lovett et al. [113] showed that the DTQW can also achieve universal quantum computation. Hence, there has been multiple proposed unifications of quantum walks in continuous and discrete time [114,115], something that is true for the classical versions [116]. For a comprehensive review of quantum walks, see the survey in [115]; and for an overview and a discussion of connections between the various quantum-walk-based search algorithms, see the work by [103].

Quantum walks have since found applications in a variety of areas including quantum search algorithms [97], modeling transport in biological systems [117], and many others [110]. Their utility stems from the combination of quantum superposition and interference, which enable new algorithmic paradigms not feasible in classical settings. Current research into quantum walks faces many challenges. As in any other quantum algorithm, a significant issue is implementation on NISQ devices where noise can disrupt the coherent evolution required for quantum walks. Efforts to mitigate these decoherence effects have been explored in various studies, such as [118], which simulate quantum walks of interacting particles and demonstrate topological protection against noise. However, achieving scalable quantum walks on large graphs or lattices still remains a challenging task [110], as current hardware limitations restrict the size and duration of coherent quantum evolutions.

## 11. Quantum Linear System Solver

Linear systems of equations are ubiquitous in science, engineering, and mathematical fields, hence there several numerical methods developed over the years for solving them efficiently on classical computers [119–121]. The general Linear Systems Problem (LSP) can be formally defines as:

**Definition 1** (LSP). *Given a matrix  $A \in \mathbb{C}^{N \times N}$  and a vector  $b \in \mathbb{C}^N$ , find a vector  $x \in \mathbb{C}^N$  s.t.  $Ax = b$ , or indicate that the system has no solution.*

In other words, Def. (1) describes a matrix-inversion problem since the solution is given by  $x = A^{-1}b$ . The Quantum Linear Systems Problem (QLSP) is quantum version of the LSP, and is defined as:

**Definition 2 (QLSP).** Given a matrix  $A \in \mathbb{C}^{N \times N}$  and a vector  $b \in \mathbb{C}^N$ , prepare the state  $|\tilde{x}\rangle$  on a quantum computer with  $n = \lceil \log_2 N \rceil$  qubits such that

$$\| |\tilde{x}\rangle - |x\rangle \|_2 \leq \epsilon,$$

where  $\epsilon$  is the desired precision. The target quantum state  $|x\rangle = A^{-1}|b\rangle$  with

$$|b\rangle := \frac{\sum_i b_i |i\rangle}{\|\sum_i b_i |i\rangle\|_2}$$

where  $b_i$  are the  $i^{\text{th}}$  component of vector  $b$ .

One of the key difference between the LSP and QLSP is that the former outputs the full solution vector  $x$  whilst the latter outputs a normalized vector proportional to the solution  $|\tilde{x}\rangle \propto A^{-1}|b\rangle$ . Table 3 shows the time complexity of the best known classical algorithm for solving the LSP, with  $A$  matrix sparse [122] and dense [123], compared with a few quantum algorithms for solving the QLSP.

**Table 3.** Comparison of the complexity between various algorithms for the LSP (first two rows) and QLSP. Here,  $N$  is the dimension of the system,  $\epsilon$  is the desired precision,  $s$  and  $\kappa$  are the sparsity and condition number of the matrix  $A$ , respectively. Note that all the quantum algorithms (except one based on QSVE in the 6th row) for the QLSP have a complexity factor of  $\mathcal{O}(\text{polylog}(N))$  in terms of the system size.

Assumption	Algorithm	Complexity
$A$ is symmetric, $s$ -sparse, and positive definite.	Conjugate Gradient [122]	$\mathcal{O}(Ns\sqrt{\kappa} \log(1/\epsilon))$
$A$ can be a dense square matrix	Powers of tensors [123]	$\mathcal{O}(N^\omega)$ with $\omega < 2.3728639$
$A$ is $s$ -sparse, Hermitian with singular values $\in [1/\kappa, 1]$	HHL - Hamiltonian simulation with QPE [6]	$\mathcal{O}(s^2\kappa^2/\epsilon)$
Same as HHL	Variable-time amplitude amplification [124]	$\mathcal{O}(s^2\kappa/\epsilon)$
Same as HHL	Fourier or Chebyshev fitting using LCU [125]	$\mathcal{O}(s\kappa \text{polylog}(s\kappa/\epsilon))$
$A$ is dense, Hermitian with eigenvalues in $[-1, -1/\kappa] \cup [1/\kappa, 1]$	Quantum Singular Value Estimation [126]	$\mathcal{O}(\sqrt{N}^{\cdot 2}/\epsilon)$
$A$ generates Hamiltonian with spectral gap amplification constraints [127]	Adiabatic random method [128]	$\mathcal{O}((\kappa \log \kappa)/\epsilon)$
$A$ general non-Hermitian matrix	Time-optimal adiabatic methods [129]	$\mathcal{O}(\kappa \text{polylog}(\log(\kappa/\epsilon)))$
Same as HHL	Zeno eigenstate filtering [130]	$\mathcal{O}(\kappa \log(\kappa/\epsilon))$
Same as HHL	Quantum discrete adiabatic theorem [131]	$\mathcal{O}(\kappa \log(1/\epsilon))$
Same as HHL	Kernel projection methods [132]	$\mathcal{O}(\kappa \log(1/\epsilon))$

### HHL Algorithm

In this section, we will consider the Harrow-Hassidim-Lloyd (HHL) algorithm [6] which is the first quantum algorithm for solving the QLSP. As mentioned earlier, it does not provide the whole solution vector, but a quantum state  $|\tilde{x}\rangle$  proportional to the solution which we can sample from it with relatively low additional error. Given that the coefficient matrix  $A$  is a sparse and well-conditioned, the quantum algorithm can potentially achieve an exponential quantum speed-up in the dimension of

the matrix  $N$  compared to the best known classical algorithm [6]. However, there are caveats for this speed-up to be possible, which includes [6,11] the following conditions:

1. Matrix  $A$  must be sparse or can be efficiently decomposed into sparse form.
2. The condition number  $\kappa$  of  $A$  must be small and scale as  $\mathcal{O}(\text{poly}(\log N))$ .
3. The elements of  $A$  can be efficiently utilized via black-box oracle calls as needed.
4. The final output is the case where one does not need to know the solution  $\vec{x}$  itself, but rather an approximation of the expectation value of some operator associated with  $\vec{x}$ , e.g.,  $\vec{x}^\dagger M \vec{x}$  for some matrix  $M$ .

It may seem that these constraints alongside with other challenges in quantum computation restricts the possibility of having a PQA as noted by Aaronson [11]. Ongoing research [133] and improved versions of HHL such as shown in Table 3 are promising to deliver PQA.

The main steps of the algorithm are (i) phase estimation, followed by (ii) a controlled rotation, (iii) uncomputation and, finally, (iv) measurement and post-selection [133]. A summary of the HHL algorithm is provided below and for more information, the reader is referred to [6,133]. Consider the  $N \times N$  matrix  $A$  that is  $s$ -sparse and Hermitian, then its spectral decomposition is given

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle \langle u_j|, \quad \lambda_j \in \mathbb{R}, \quad (15)$$

where  $|u_j\rangle$  is the  $j^{\text{th}}$  eigenvector of  $A$  with respective eigenvalue  $\lambda_j$ . In this decomposition, the matrix inversion of  $A$  is then given by

$$A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1} |u_j\rangle \langle u_j|. \quad (16)$$

We can express  $|b\rangle$  in the eigenbasis of  $A$  as

$$|b\rangle = \sum_{j=0}^{N-1} b_j |u_j\rangle, \quad b_j \in \mathbb{C}, \quad (17)$$

where we already have an implicit normalisation constant since we are talking about a quantum state. The goal of the HHL is to exit the algorithm with the readout register in the state approximately  $|x\rangle$  which can also be expressed in the eigenbasis of  $A$  as

$$|x\rangle = A^{-1} |b\rangle = \sum_{j=0}^{N-1} \lambda_j^{-1} b_j |u_j\rangle \quad (18)$$

Therefore, the HHL algorithm encodes  $A$  into a unitary  $U$  given by

$$U = e^{iAt} := \sum_{j=0}^{N-1} e^{i\lambda_j t} |u_j\rangle \langle u_j| \quad (19)$$

Thus, by applying QPE, one can implement the mapping

$$|0\rangle |u_j\rangle \mapsto |\tilde{\lambda}_j\rangle |u_j\rangle, \quad (20)$$

where  $\tilde{\lambda}_j$  is the binary representation of  $\lambda_j$  up to a tolerated precision. Following this, the implementation of a controlled rotation takes place conditioned on  $|\tilde{\lambda}_j\rangle$ . This requires an ancilla register,  $S$ , which

is added to the system initialised in state  $|0\rangle$ . The controlled rotation is of the form of a  $\sigma_y$ -rotation (i.e., a rotation around the  $y$ -axis or quantisation axis) and produces a normalised state of the form

$$\sqrt{1 - \frac{c^2}{\tilde{\lambda}_j^2}} |\tilde{\lambda}_j\rangle |u_j\rangle |0\rangle + \frac{c}{\tilde{\lambda}_j} |\tilde{\lambda}_j\rangle |u_j\rangle |1\rangle, \quad (21)$$

where  $c$  is a normalisation constant. This can be achieved through the application of the quantum operator

$$\sigma_y = e^{-i\theta} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix},$$

where  $\theta = \arccos c/\tilde{\lambda}$ . Thus, enacting the procedure described above in the latter superposition, it is derived that

$$\sum_{j=1}^N b_j |\tilde{\lambda}_j\rangle |u_j\rangle \left( \sqrt{1 - \frac{c^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{c}{\tilde{\lambda}_j} |1\rangle \right).$$

Finally, the first register is uncomputed, giving

$$|0\rangle \otimes \sum_{j=1}^N b_j |u_j\rangle \left( \sqrt{1 - \frac{c^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{c}{\tilde{\lambda}_j} |1\rangle \right). \quad (22)$$

Following the uncomputation, we see that the state of the system (including all the quantum registers) is  $|\tilde{x}\rangle = \sum_{j=1}^N \tilde{\lambda}_j^{-1} b_j |u_j\rangle$ , or in other words, it exists in a state that is close to the solution  $|x\rangle = A^{-1}|b\rangle$ . The state  $|\tilde{x}\rangle$  can be reconstructed within the clock register,  $C$ , by measuring the ancilla register,  $S$ , and post-selecting on the outcome 1 modulo the constant factor of normalisation,  $c$ . The success probability of the final step can be boosted using QAA.

---

#### Algorithm 5 Harrow-Hassidim-Lloyd algorithm for QLSS

---

**Input.** Encode the state vector  $|b\rangle$  and the matrix  $A$  with oracular access to its elements. Define  $t_0 = \mathcal{O}(\kappa/\epsilon)$ ,  $T = \tilde{\mathcal{O}}(\log Ns^2t_0)$  and  $\epsilon$  as the desired precision.

**Initialise.** Prepare the input state  $|\Psi_0\rangle^C \otimes |b\rangle^I \otimes |0\rangle^S$ , where  $|\Psi_0\rangle = \sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+1/2)}{T}\right) |\tau\rangle^C$ .

**Hamiltonian.** Apply the conditional Hamiltonian evolution  $\sum_{\tau=0}^{T-1} |\tau\rangle \langle \tau|^C \otimes e^{iA\tau t_0/T}$  following the input state.

**Apply QFT** to the register  $C$ , denoting the new basis states  $|k\rangle$ , for  $k \in \{0, \dots, T-1\}$ . Define  $\tilde{\lambda} := 2\pi k/t_0$ .

**Controlled rotation.** Append the ancilla register, namely  $S$ , and apply a controlled rotation on  $S$  with  $C$  as the control register, mapping states  $|\tilde{\lambda}\rangle \mapsto |h(\tilde{\lambda})\rangle$ . The state  $|h(\tilde{\lambda})\rangle$  is defined in such a way that it produces an output which denotes whether an inversion has occurred and if that inversion is well-conditioned or not [133].

**Uncomputation.** Uncompute the register  $C$ .

**Measurement.** Measure the register  $S$ .

**Repeat.** Perform  $\mathcal{O}(\kappa)$  rounds of QAA on the HHL algorithm.

**Output.** Result is the state  $|\tilde{x}\rangle$  s.t.  $\| |\tilde{x}\rangle - |x\rangle \|_2 \leq \epsilon$

---

The HHL algorithm was the first QLSS and as shown in Table 3, there has been many improvements on it up to the recent ones with optimal scaling in  $\kappa$  and  $\epsilon$  [131,132]. It is worth noting these state-of-the-art quantum algorithms for the QLSS with optimal scaling for the case when  $A$  is sparse, and there has been a few proposals for the case when  $A$  is dense. For the latter case, Wossing et al. [126] utilised the Quantum Singular Value Estimation (QSVE) algorithm of [134] to achieve a polynomial speed-up over the best known classical algorithm based on powers of tensors with subcubic complexity [123]. However, as noted in [126], the classical algorithm with subcubic complexity [123] is difficult to achieve in practise hence for the dense matrix case the Cholesky decomposition with complexity

$\mathcal{O}(N^3)$  [135] is often used. This implies that for problems with non-sparse  $A$  matrix, such as kernel methods and neural networks in machine learning [136], the potential exponential quantum advantage of HHL-type quantum algorithms is lost.

Following this, Kerenedis and Prakash generalised the QSVE-based linear systems solver to handle both sparse and dense matrices and introduced a technique for spectral norm estimation [137]. Another general framework for solving the QLSP is the Quantum Singular Value Transformation (QSVT) [138] which uses an iterative least square algorithm has been proposed [139,140]. Although this algorithm does not have optimal scaling with  $\kappa$  and  $\epsilon$ , it is general in the sense that it does not assume that  $A$  is invertible nor that it is a square matrix, but uses the Moore-Penrose pseudoinverse  $A^+$  [141] such that the output  $\|\hat{x} - A^+|b\rangle\|_2 \leq \epsilon$  is achieved efficiently [48,139,140]. Furthermore, the QSVT also provides a general framework that unifies [62] a variety of quantum algorithms such as a linear algebraic subroutines like singular-value-threshold projectors [134,138] and matrix-vector multiplication [139]. An outline of QSVT is presented in Algorithm (6).

---

#### Algorithm 6 Quantum Singular Value Transformation

---

**Input:** A  $(1, m, 0)$  block-encoding  $U_A$  of matrix  $A$ , i.e.  $A = (\langle 0^m| \otimes I) U_A (|0^m\rangle \otimes I)$ ; a definite-parity polynomial  $f : [-1, 1] \rightarrow [-1, 1]$  of degree  $d$ ; and a phase sequence  $\Phi = (\varphi_1, \dots, \varphi_d)$  from polynomial synthesis.

**Construct QSVT sequence:** Let  $Z_{|0^m\rangle} := 2|0^m\rangle\langle 0^m| - I$ , and define

$$U_\Phi = \left( e^{i\varphi_1 Z_{|0^m\rangle}} U_A \right) \prod_{j=1}^{(d-1)/2} \left( e^{i\varphi_{2j} Z_{|0^m\rangle}} U_A^\dagger e^{i\varphi_{2j+1} Z_{|0^m\rangle}} U_A \right), \quad d \text{ odd},$$

$$U_\Phi = \prod_{j=1}^{d/2} \left( e^{i\varphi_{2j-1} Z_{|0^m\rangle}} U_A^\dagger e^{i\varphi_{2j} Z_{|0^m\rangle}} U_A \right), \quad d \text{ even}.$$

This yields a block-encoding of  $f^{(SV)}(A)$  with the corresponding odd/even forms.

**Linear combination:** Using that  $-\Phi$  implements  $P^*$ , define

$$P_{\mathbb{R}}(A) = (\langle +| \otimes \langle 0^m| \otimes I) (|0\rangle\langle 0| \otimes U_\Phi + |1\rangle\langle 1| \otimes U_{-\Phi}) (|+\rangle \otimes |0^m\rangle \otimes I),$$

which implements a block-encoding of  $P_{\mathbb{R}}(A)$ , and thus of  $f^{(SV)}(A)$  for any definite-parity real  $f$ .

**Output:**  $P_{\mathbb{R}}(A)$  that is a  $(1, m+1, 0)$  block-encoding of  $f^{(SV)}(A)$ .

---

While solving the QLSP does not provide classical access to the solution vector, this can still be done by utilising methods for vector-state tomography with a cost of  $\mathcal{O}(N/\epsilon^2)$  samples [142]. Applying tomography would no longer allow for an exponential speed-up in the system size  $N$  [11]. However, polynomial speed-ups with tomography for some linear algebra-based algorithms are still possible for certain types of problems [48]. In addition, without classical access to the solution, certain statistics, such as the expectation of an observable with respect to the solution state, can still be obtained without losing the exponential speed-up in  $N$  [6,143]. Another current bottleneck for QLSS is loading the data of  $A$  and  $b$  into quantum gates and state, respectively [11]. In theory, this can be dealt with by using one of the two main quantum models for data access [48,139]: (i) the sparse-data access model and, (ii) the quantum-accessible data structure. The sparse-data access model provides efficient quantum access to the nonzero elements of a sparse matrix. The quantum-accessible data structure, suitable for fixed-sized, non-sparse inputs, is a classical data structure stored in a quantum memory (e.g., Quantum Random Access Memory (QRAM)) and provides efficient quantum queries to its elements [139]. Considering the cost of QRAM with QEC overheads [144], the potential quantum advantage of QLSS may be reduced to polynomial at best [48] or none in the worse-case [11].

Lastly, for LSPs that involve only low-rank matrices, classical Monte Carlo methods for numerical linear algebra (e.g., the FKV algorithm [145]) can be used. These techniques have been used to produce classical algorithms that have an asymptotic exponential speed-up in dimension for various problems

involving low-rank linear algebra (e.g., some machine learning problems) [146,147]. These classical quantum-inspired (dequantised) algorithms seem to have robustness issues [148], making QLSS potentially advantageous for certain cases. In addition, since these results do not apply to sparse linear systems, there is still the potential for provable speed-ups for problems involving sparse, high-rank matrices [48].

## 12. Variational Quantum Algorithms

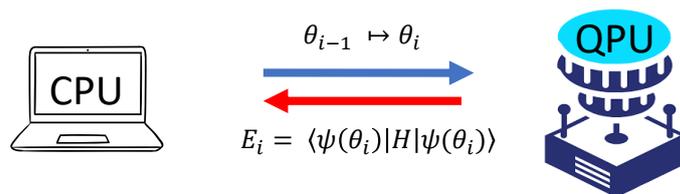
Classical *variational method* is a well-established technique in quantum mechanics [149] for finding the ground state(s) of a quantum system based on the *variational principle*. The methods involves:

1. **Ansatz** - specifying some trial parametrised wavefunction  $|\psi(\theta)\rangle$  where  $\theta$  are the set of variational parameters.
2. **Variation** - minimising the expected energy

$$E = \langle \psi(\theta) | \mathcal{H} | \psi(\theta) \rangle \geq E_0, \quad (23)$$

by varying the parameters  $\theta$ .

Here,  $\mathcal{H}$  is the Hamiltonian for the system, and by the *variational principle* the energy  $E$  after each variation will always be bound below by the true ground state energy  $E_0$  [149]. Hence the set of parameters  $\theta^*$  that minimizes Equation (23) results to a good approximation of true ground-state energy  $E_0$ . In practice, there may be factors preventing the true ground state from being found, such as a failed optimisation, or an inappropriate ansatz that does not include the support basis vectors of the true ground-state [149]. In general, this task is computationally expensive for complex many-body system with Hilbert space that grows exponentially with system size. Hence, by leveraging the high-dimensional capabilities of quantum computers to reduce the complexity of this method lead to the Variational Quantum Eigensolver (VQE) [150]. Figure 2 gives a sketch of the roles the classical and quantum computer plays in VQE algorithm. Essentially, since the computation of the expectation value of Equation (23) is prohibitive classically, it is assigned to the Quantum Processing Unit (QPU) and the task of finding optimal parameters to the CPU [150].



**Figure 2.** A sketch of classical and quantum computer's function in the VQE algorithm. The CPU sets and updates the parameters ( $\theta_{i-1} \mapsto \theta_i$ ) using an optimiser. The new parameters  $\theta_i$  are used to prepare a quantum state  $|\psi(\theta_i)\rangle$  on the QPU which is used to compute the expectation value  $E_i = \langle \psi(\theta_i) | \mathcal{H} | \psi(\theta_i) \rangle$ . Then the CPU takes this output  $E_i$ , and based on the energy difference  $\Delta E_i = |E_i - E_{i-1}|$  decides to either terminate (convergence at  $\Delta E_i \approx 0$ ) or continue loop with new variational parameters ( $\theta_i \mapsto \theta_{i+1}$ ).

The choice of the VQE ansatz can have a significant effect on both the speed and quality of the solution. For example, the ansatz may be chosen to reflect a certain symmetry in the problem set up, or such that it rules out or focuses on a particular region of the Hilbert space, reflecting constraints on possible solutions [151]. Apart from the guideline of symmetry and expressiveness of the ansatz, there is no known general rule for picking the 'best' ansatz as it seems to be problem specific [152]. Another factor that impacts the quality of the solution is the performance of the classical optimiser. One common problem encountered when implementing variational algorithms is that of barren plateau [153], referring to solution landscapes that have exponentially vanishing gradients as a function of the number of qubits. This results in the optimiser unable to efficiently find the optimal

solution or being stuck in a local minimum. A lot of research has been done on the problem [154], but there is not yet a consensus on how to consistently identify or mitigate the issue [154–156].

The VQE algorithm can be considered a special case of general Variational Quantum Algorithms (VQAs) with parametrized quantum circuits (denoted by the unitary  $U(\theta)$ ) and a cost function given by [152]

$$C(\theta) = \sum_k f_k \left( \text{Tr} \left[ O_k U(\theta) \rho_k U^\dagger(\theta) \right] \right). \quad (24)$$

Here,  $\{f_k\}$  is some set of functions with properties defined in [152],  $\{\rho_k\}$  are input states from a training set, and  $\{O_k\}$  are set of observables. The goal is to find optimal set of parameters

$$\theta^* = \arg \min_{\theta} C(\theta), \quad (25)$$

by using a QPU to estimate the cost function  $C(\theta)$  (or its gradient) while leveraging the efficiency of classical optimiser to train the parameters  $\theta$ . This framework naturally applies to quantum algorithm for optimisation problems [48,152,157] (see Chapter (17) for more details). There have been numerous proposed applications of VQA which includes solving the QLSP [158,159], PDEs [160,161], and QML problems (Chapter (20)) [162]. Although quantum computers are efficient in estimating the cost function, the general training VQAs (i.e. finding the optimal ansatz parameters) is an NP-hard non-convex optimisation problem [163,164]. Furthermore, these algorithms are heuristic with no yet provable quantum advantage in complexity of the algorithm [152] but they have empirical proof-of-concept implementations in NISQ era [48] and might obtain PQA with Mega-QuOps<sup>4</sup> QPUs [165].

### 13. Quantum Annealing

The Quantum Annealing (QAnn) algorithm [166,167] is an approach to quantum computing based on Adiabatic Quantum Computing (AQC) [168], which is an alternative to gate-based models of which all previous algorithms outlined in this chapter are derived from. While gate-based algorithms represent any computation as a series of gates, employing the concept of universality [44], AQC is an analogue approach that is suitable for a specific set of optimisation problems [168]. Primarily, the Quadratic Unconstrained Binary Optimisation (QUBO) problems (further discussed in Chapter (17)), form the basis for a rich variety of NP-hard combinatorial optimisation problems tackled by QAnn [169].

Annealing relies on the adiabatic quantum theorem [168], which states that if the system starts out in the  $n$ th eigenstate of a time-dependent Hamiltonian  $H(t=0)$ , and this Hamiltonian evolves *slowly enough*, the system will *remain* in the instantaneous  $n$ th eigenstate during the evolution. More specifically, during the time evolution, the parameters are changed sufficiently slowly that the system adapts to the new parameter configuration quasi-instantaneously. The length of time required for the evolution to succeed depends on the spectral properties of the Hamiltonian path and in particular on the minimum spectral gap [168]. For example, the change from  $H_0$  to  $H_1$  can be achieved by the Hamiltonian:

$$H(t) = (1 - f(s))H_0 + f(s)H_1 \quad (26)$$

where  $f(s) \in [0,1]$  is the the scheduling function and  $s \in [0,1]$  is the annealing parameter. The scheduling function is chosen such that at  $f(s=0) = 0$ , the system is governed by some initial Hamiltonian  $H_0$ , and at  $f(s=1) = 1$  the final dynamics are governed by the *problem* Hamiltonian  $H_1$ . This implies that  $f(s)$  must be strictly increasing and the simplest choice  $f(s) = s$  gives the vanilla AQC. If the system starts out the ground state (eigenstate with minimal energy) of  $H_0$  and slowly evolved, then according to the adiabatic theorem, it will remain the ground state throughout its evolution until it rests in the ground-state of  $H_1$  [168]. This means that, to find the unknown ground

<sup>4</sup> Quantum computers capable of performing about a million error-corrected quantum operations

state of a Hamiltonian for a hard-to-solve problem  $H_1$ , one can start from an easy-to-solve Hamiltonian  $H_0$  with a known ground state and apply the QAnn algorithm.

Typically, the Hamiltonian  $H_0$  has the form  $H_0 = \sum_i \sigma_i^x$  with ground state being the equal superposition of all bitstrings  $\{x\}$  [166]. On the other hand,  $H_1$  is described by the canonical Ising model

$$H_1 = \sum_i h_i \sigma_i^z + \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z. \quad (27)$$

where the biases  $h_i$  and couplings  $J_{ij}$  are determined by the problem [166]. In practice, it is very difficult to fulfil the conditions to achieve adiabaticity, mainly due to unwanted noise on NISQ hardware. Therefore, annealers forego some of the stringent theoretical conditions and heuristically repeat the annealing procedure many times, collecting a number of samples from which the configuration with the lowest energy can be selected as the optimal solution, but with no strict guarantee that the optimal solution will be in this sample set [48,170].

One key issue affecting annealing is the requirement that the problem is reformulated in the correct format (as it is believed the transverse-field Ising Hamiltonian is not universal [57]). For problems which have a native structure (including many graph-based problems, for example), annealing is particularly suitable [170]. For non-native problems, this reformulation may introduce a significant overhead, or such strenuous constraints, that annealing might fail consistently or any potential quantum advantage might be lost [170]. There exist classes of time-dependent Hamiltonians that through adiabatic evolution can approximate arbitrary unitary operators applied to qubits [168,171]. Thus AQC can also realise universal quantum computation for such a class of problems. However, by polynomial equivalence to the circuit model, gate-based devices can also efficiently simulate these problems [168,170].

The main reason behind the interest in QAnn is that it provides a non-classical heuristic, tunnelling, that is potentially helpful for escaping from local minima [168]. Its closest classical analogue is simulated annealing [172], a Markov chain Monte Carlo method that was inspired by classical thermodynamics and uses temperature as a heuristic to escape from local minima. So far QAnn has demonstrated usefulness for solving QUBOs with tall yet narrow peaks in the cost function [173,174]. The overall benefit of QAnn is still a topic of ongoing research [48,170,173]. Furthermore, commercially available quantum machines that implement QAnn, from providers such as D-Wave [175], have allowed for various near-term exploration [176–179].

## 14. Quantum Simulation

While quantum simulation was originally envisioned as a tool for modeling quantum mechanical systems [2,3], recent advances have demonstrated its broader potential to address classical problems whose dimensional complexity renders them inefficiently solvable when tackled with conventional methods [180]. This is the case in the domain of quantitative finance, where the numerical solutions of Stochastic Differential Equations (SDEs) plays a central role in modeling financial products such as pricing options [84,85]. These problems, while inherently classical, often involve high-dimensional integration and probabilistic dynamics that render them computationally intensive [181]. Quantum simulation techniques, including those adapted for solving linear systems and sampling probability distributions, provide promising pathways to address these challenges [9,31]. In the classical regime, SDEs such as those arising in the pricing of financial derivatives under the Black-Scholes-Merton or Heston models, require discretisation techniques like finite differences or Monte Carlo simulations [182]. However, the computational cost grows rapidly with the number of risk factors or underlying assets - a manifestation of the so-called *curse of dimensionality*. Quantum computing offers a potential remedy through quantum algorithms which can achieve polynomial [5] or even exponential speedups over classical counterparts for problems with certain structures [183]. In this review, we will consider quantum algorithms that speed up Monte Carlo Integration (MCI) and quantum solvers for SDEs.

## 15. Quantum Monte Carlo

Monte Carlo methods are very commonly used to approximate solutions to equations which are difficult to solve analytically or when numerical methods scale poorly with the dimension of the problem. These include inference [184], optimisation [185], numerical integration [186], and stochastic expectation values [182]. For the latter, MCI is ubiquitously used in finance for problems such as asset pricing and risk analyses of complex investment portfolios (see use-cases in Chapters (30)-(40)). The major advantage of MCI methods is that their scaling depends on the number of samples and not the dimension of the problem [182]. However, their downside is that they have a slow convergence since its estimation error  $\epsilon$  decays as  $\mathcal{O}(1/\sqrt{N_p})$ , where  $N_p$  is the number of sample paths taken [5]. Routine finance problems such as risk analysis or pricing options often require large amounts of samples to achieve a desired precision [9]. Thus, there is value in exploring and understanding any potential benefits that quantum computing could bring in this space.

### 15.1. QMCI Algorithm

The typical use of MCI in finance is to estimate the expected value of an unknown financial quantity (e.g., price of a financial derivative), which is a function of other nondeterministic variables (e.g., market state) [9]. The methodology, described in more details in Section (32), generally starts with choosing a stochastic model for the underlying random variables from which samples, denoted as  $\{X_1, \dots, X_n\} \subseteq \Omega$ , are taken from. Then the corresponding value of the target quantity, denoted as  $f(X_1, \dots, X_n)$ <sup>5</sup>, is computed based on the samples drawn. The weighted average of all obtained values of the target quantity is given by the expectation [182]

$$\mathbb{E}[f(X)] = \int_{\Omega} p(X)f(X)dX \approx \sum_{i=1}^n p(X_i)f(X_i) \quad (28)$$

where  $p(X)$  is the probability distribution. Note that the random variable is sampled from a space  $X_i \subset \Omega$  that could be multi-dimensional. The Quantum Monte Carlo Integration (QMCI) [5] is the quantum formulation of the MCI method. Outlined in Algorithm (7), the QMCI utilises the Quantum Amplitude Estimation subroutine to compute the expectation given in Equation (28) and can potentially achieve a quadratic speed-up over classical MCI [5]. This is because the estimation error for QMCI decays as  $\mathcal{O}(1/N_q)$ , where  $N_q$  is the number of queries made to a quantum oracle that computes  $f(X_1, \dots, X_n)$ . This is comparable to the complexity in terms of the number of samples  $N_p$  for MCI which has an error that scales as  $\mathcal{O}(1/\sqrt{N_p})$ . Thus, if samples are considered as classical queries, the QMCI requires quadratically fewer queries than classical MCI to achieve the same desired error.

<sup>5</sup> Here  $f(\cdot)$  could represent the payoff function for option pricing, see Section (32.1)

**Algorithm 7** Quantum Monte Carlo Integration

**Define:** Let  $\Omega$  be the set of potential sample paths  $\omega$  of a stochastic process that is distributed according to some probability  $p(\omega)$ , and  $f : \Omega \rightarrow A$  is a real-valued function on  $\Omega$ , where  $A \subset \mathbb{R}$  is bounded.

**Construct:** A unitary operator  $P_l$  to load a discretised and truncated version of  $p(\omega)$ . The probability value  $p(\omega)$  translates to the amplitude of the quantum state  $|\omega\rangle$  representing the discrete sample path  $\omega$ . In mathematical form, it is

$$P_l |0\rangle = \sum_{\omega \in \Omega} \sqrt{p(\omega)} |\omega\rangle \quad (29)$$

**Normalise:** The function  $f$  into  $\tilde{f} : \Omega \rightarrow [0, 1]$ , and construct a unitary  $P_f$  that computes  $\tilde{f}(\omega)$  and loads the value onto the amplitude of  $|\omega\rangle$ . The resultant state after applying  $P_l$  and  $P_f$  is

$$P_f P_l |0\rangle = \sum_{\omega \in \Omega} \sqrt{(1 - \tilde{f}(\omega))p(\omega)} |\omega\rangle |0\rangle + \sqrt{\tilde{f}(\omega)p(\omega)} |\omega\rangle |1\rangle \quad (30)$$

**Perform:** The QAE with unitary  $U = P_l P_f$  and an oracle  $O_f$  that marks states with the last qubit being  $|1\rangle$ . See Section (8.2) for details.

**Result:** Will be an approximation to

$$\mathbb{E}[\tilde{f}] = \sum_{\omega \in \Omega} \tilde{f}(\omega)p(\omega)$$

This value can be estimated to a desired error  $\epsilon$  utilising  $\mathcal{O}(1/\epsilon)$  evaluations of  $U$  and its inverse [5].

**Rescale:** The output  $\mathbb{E}[\tilde{f}]$  to the original bounded range ( $A$ ) to obtain  $\mathbb{E}[f]$ .

## 15.2. Challenges for Quantum Advantage

At the time of this manuscript, no demonstration of quantum advantage with QMCI has been made, primarily due to the limitations of current quantum hardware and algorithmic challenges in key subroutines [12]. The former hardware limitations are due to:

1. **Clock speed:** As noted in [187], to achieve a Practical Quantum Advantage using QMCI over classical MCI, the quantum device would need to be able execute about  $10^7$  layers of  $T$ -gates<sup>6</sup> per second. This implies a required logical clock rate of about 50MHz in order to be competitive with current classical MCI methods. Recent methods [188] have reduced this requirement to 4.5MHz which is still beyond
2. **Fault-tolerant:** The code distance for fault-tolerant implementations needs to be large enough to support  $10^{10}$  error-free logical operations [187]. In addition, quantum algorithms with a proposed quadratic speed-up need to tackle significantly high-dimensional problems to potentially realise some advantage [46] which further complicates fault-tolerant implementations.
3. **Resource:** The estimates are high for fault-tolerant resources needed to achieve competitive performance for finance problems that are challenging for classical methods. The latest estimates for derivative pricing [188] based on Quantum Signal Processing are  $4.7 \times 10^3$  logical qubits,  $4.5 \times 10^7$   $T$ -depth, and  $2.4 \times 10^9$   $T$ -count. This requirements is beyond currently available quantum hardware, and seems to be significantly high in comparison to classical MCI which requires  $4 \times 10^4$  samples and about 10 seconds to achieve the same accuracy [48].

On the other hand, the latter algorithmic challenges are due to:

1. **Loading distribution:** The state-preparation of an arbitrary probability distributions requires exponentially large circuits in terms of the number of qubits [189], which affects the potential quantum speed-up of QMCI. The improved state-preparation algorithm by Grover–Rudolph [190] has been shown to be insufficient to achieve a quantum speed-up [191]. However, for certain distributions, a quadratic speed-up is possible [95,192,193]. It is noteworthy that various non-unitary methods exist for efficient state preparation for large circuits [194,195]. However, these methods are usually incompatible with QAE because of the non-invertibility of the operations

<sup>6</sup> See Table 2 for description of  $T$ -gate.

involved. Furthermore, other alternative methods have been proposed such Quantum Generative Adversarial Networks (QGAN) [196] and tensor networks [197].

2. **Arithmetic:** Computing coherent quantum arithmetic associated with function  $f(\omega)$  can have gate complexity that is similar to classical arithmetic [48]. This step can be elevated by quantum arithmetic free methods based on Quantum Signal Processing [192] and Fourier analysis [193,198].
3. **Estimation:** The first proposed QAE algorithm employed QPE which has a resource requirement beyond the capabilities of NISQ devices (see Section 7). However, non-QPE implementations of QAE have been proposed which can enable near-term implementations of QMCI (see Section 8.2).

As noted in [193], there are many other factors that will drive the adoption of quantum computing in finance beyond exceeding classical performance. Some of these factors will be technical such ease of incorporation of QPU in computational pipeline in finance, and others may be business and/or regulatory related. In any case, it is important to have a well-defined framework [199] to evaluate the goal of achieving Practical Quantum Advantage. In this review, we will consider several use-cases for QMCI in finance and economics which are good candidates for obtaining a PQA in the long-run as the quantum technology matures.

## 16. Quantum Solvers for Stochastic PDEs

Stochastic Differential Equations (SDEs) have become fundamental tools for modeling a wide array of phenomena within finance and economics [200]. The dynamic and often unpredictable nature of financial quantities, such as asset prices, interest rates, and their derivatives, aligns naturally with the framework provided by SDEs, which explicitly incorporate stochastic elements into their structure. Unlike deterministic models based on ordinary differential equations, which yield a unique solution for a given initial condition, SDEs describe the evolution of continuous-time stochastic processes, reflecting the inherent uncertainties present in financial systems. The necessity of employing SDEs in finance arises from the fact that many financial variables exhibit diffusive dynamics, a characteristic well-captured by stochastic processes [200]. For instance, the Heath-Jarrow-Morton model [201], a cornerstone in interest rate modeling, is formulated as an SDEs. Similarly, the Heston [202] model for pricing of options under conditions of stochastic volatility, where the volatility itself is modeled as a random process, leads to a set of coupled SDEs<sup>7</sup>.

Despite the power and versatility of SDEs in capturing the complexities of financial markets, obtaining analytical solutions to these equations is often a formidable challenge for realistic models [200,203]. This is due to various factors which includes nonlinearities in models which complicates integration and inherent high-dimensional from multiple underlying variables [204]. There are various numerical methods to solving SDE with diverse approaches, however, in this review we will only consider the approach of employing the Feynman-Kac formula which gives the expectation value<sup>8</sup> to Equation (28) as solution to a parabolic PDE. This will make it easier to compare with the QMCI algorithms described above.

### 16.1. Feynman-Kac Formula

Originally conceived within the framework of quantum mechanics by Feynman through his path integral formulation [205] and later rigorously formulated in a probabilistic context by Kac [206], this formula provides an alternative perspective for solving the expectation value given by Equation (28). Its original usage had the reverse objective to what we use it for here; that is to solve a complex PDE with high dimensionality or non-constant coefficients by computing the expected value of a carefully constructed stochastic process [200]. However, since quantum computing is more suited to handle high dimensional problems, it makes sense to convert expected values of stochastic processes into parabolic PDEs.

<sup>7</sup> See Section (32.1.2) for more details

<sup>8</sup> This is similar to Equation (145) of Section (32)

The Feynman-Kac formula [200,205–207] establishes a profound connection between parabolic partial PDEs and the expected values of certain stochastic processes. This relationship provides a conversion between the two which can be a powerful tool for analysis and computation.

### Mathematical Formulation

Consider a parabolic PDE for the unknown variable  $u = u(x, t) : \mathbb{R}^n \times [0, T] \mapsto \mathbb{R}$  of the form:

$$\frac{\partial}{\partial t}u + \mu \cdot \nabla u + \frac{1}{2}\text{Tr}[\Sigma\Sigma^T D^2u] - Vu + f = 0 \quad (31)$$

with a terminal condition  $u(x, T) = g(x)$ , where  $g(x)$  is the terminal payoff function. Here the known functions:  $\mu = \mu(x, t)$  is a vector-valued function representing the drift,  $\Sigma = \Sigma(x, t)$  is an  $n \times d$  matrix representing the diffusion coefficient,  $V = V(x, t)$  is a scalar potential function, and  $f = f(x, t)$  is a source term. The operators:  $\nabla u$  is the gradient of  $u$ ,  $D^2u$  is the Hessian matrix of  $u$ , and  $\text{Tr}$  denotes the trace of a matrix. The Feynman-Kac theorem [200,207] states that the solution to the PDE in Equation (31) can be expressed as the conditional expectation:

$$u(x, t) = \mathbb{E}_Q \left[ g(X_T)h(t, T) + \int_t^T f(s, X_s)h(t, r)dr \mid X_t = x \right], \quad (32)$$

where the function  $h(\cdot)$  is given by

$$h(a, b) = \exp\left(-\int_a^b V(s, X_s)ds\right). \quad (33)$$

The stochastic variable  $X_s$  is the solution to the following SDE [200]:

$$dX_s = \mu(s, X_s)ds + \sigma(s, X_s)dW_s^Q \quad (34)$$

with the initial condition  $X_t = x$ , and  $W_s^Q$  is a  $d$ -dimensional standard Brownian motion under a probability measure  $Q$ . The expectation  $\mathbb{E}_Q[\cdot]$  is taken with respect to this probability measure, conditional on the initial state  $X_t = x$ . Intuitively, the Feynman-Kac formula suggests [207] that the value of the function  $u(x, t)$  at a point  $(x, t)$  can be seen as the expected future value, discounted by the potential  $V$ , of a stochastic process  $X$  that starts at  $x$  at time  $t$  and evolves according to the SDE in Equation (34). The terminal condition  $g(X_T)$  contributes at the final time  $T$ , and the source term  $f(s, X_s)$  contributes along the path of the process. The function  $h(a, b)$  acts as a stochastic discount factor, accounting for the accumulated effect of the potential along the random path.

### Example BSM Model

A classic example is the Black-Scholes-Merton (BSM) model for the vanilla European option<sup>9</sup>, where Equation (31) can be formulated as [31]

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0, \quad (35)$$

where  $V(t, S_t) \in (0, T) \times \mathbb{R}^+$  is the option expected value with the terminal condition

$$V(T, S) = f(T, S). \quad (36)$$

Here  $f(\cdot)$  is the payoff function,  $r$  is the risk-free rate, and  $\sigma$  is the volatility of the stochastic underlying asset  $S_t$ . The PDE of the form of Equation (35) can be solved by using quantum algorithms outlined in the ensuing sections.

<sup>9</sup> See Section (32) for more details.

## 16.2. Finite Difference Method

A widely used class of numerical techniques are Finite Difference Method (FDM) [208] that approximate the derivatives appearing in differential equations using differences between the values of the unknown function at discrete points in time and space. When applied to SDEs, the FDMs involve discretising both the spatial and temporal domains into a grid and replacing the continuous derivatives with their finite difference approximations [200]. Various finite difference schemes exist, including explicit and implicit methods, each exhibiting different characteristics in terms of stability and computational requirements [208]. For instance, explicit schemes are generally simpler to implement but may have stricter stability conditions, while implicit schemes, such as the Crank-Nicolson method, are known for their numerical stability, particularly for parabolic PDEs common in finance [208,209]. In situations where the financial model exhibits convection-dominated behavior, upwind schemes are often employed to ensure numerical stability and accuracy [210].

Algorithm (8) outlines a general scheme of the FDM for solving SDEs via the Feynman-Kac formula. For every time step  $t_j$ , the problem boils down to solving a linear system of equations with dimension  $N$  that depends on the discretisation in stochastic variable  $x$ . Therefore, assuming the matrix  $A$  in Equation (38) is well conditioned, we can employ the Quantum Linear Systems Solver discussed in Section (11) to potentially obtain an exponential speed-up [131,132]. For arbitrary cases,  $A$  is not guaranteed to be well-conditioned and the condition number may scale as a polynomial with the dimension:  $\kappa(A) \sim \mathcal{O}(\text{poly}(N))$ . In such cases, both classical and quantum methods can make use of preconditioners to reduce  $\kappa$  before running the linear system solver algorithm. The use of preconditioners for QLSSs has been explored [183,211–213] and have shown promising results. For example, authors in [183] used a wavelet basis as an auxiliary system of coordinates in which the condition number of associated matrices is independent of  $N$  by a simple diagonal preconditioner.

---

### Algorithm 8 Finite Difference Method for solving SDEs

---

1. Define bounds for stochastic variable  $X_t = x$  and impose appropriate boundary conditions for the expectation  $u(x, t)$  [200].
2. Define the finite difference mesh by choosing two natural numbers  $J > 1$  and  $I > 1$ .
  - the constant time step size  $\Delta t = T/(J + 1)$  and asset step size  $\Delta x = x_\infty/(I + 1)$
  - mesh nodes are  $(t_j, x_i) = (j\Delta t, i\Delta x)$  where  $i, j \in [0, I + 1], [0, J + 1]$
3. Define a suitable numerical approximation of the derivatives at each mesh nodes. Let  $u_{ji} = u(t_j, x_i)$  the derivatives are given by [208]

$$\frac{\partial u}{\partial t} \approx \frac{u_{j+1,i} - u_{j,i}}{\Delta t}, \quad \frac{\partial u}{\partial x} \approx \frac{u_{j,i+1} - u_{j,i-1}}{2\Delta x}, \quad \frac{\partial^2 u}{\partial x^2} \approx \frac{u_{j,i+1} - 2u_{j,i} + u_{j,i-1}}{(\Delta x)^2} \quad (37)$$

where the second order Crank-Nicolson method [208] can be used for the discretisation in time.

4. This leads to a  $N = I + 1$  dimensional linear system at each time  $t_j$  given by

$$Au_j = b_j, \quad u_j, b_j \in \mathbb{R}^N \quad \text{and} \quad A \in \mathbb{R}^{N \times N} \quad (38)$$

5. Starting with  $j = J + 1$ , one can solve Equation (31) using either classical or quantum algorithms:
  - CG method [214] with scales as  $\mathcal{O}\left(N\sqrt{\kappa} \log\left(\frac{1}{\epsilon}\right)\right)$
  - QLSS [131] which scales as  $\mathcal{O}\left(\log(N)\kappa \log\left(\frac{1}{\epsilon}\right)\right)$

where  $\kappa$  is the condition number for  $A$  and  $\epsilon$  is the error tolerance.

6. Sequentially repeat step 4-6 for  $j = J, J - 1, \dots, 0$ .
- 

A crucial aspect of employing FDMs for SDEs is the analysis of their stability and convergence in the presence of stochastic terms [200]. Stability ensures that any numerical errors introduced during the computation do not grow uncontrollably as the simulation progresses. Convergence analysis,

on the other hand, aims to determine whether the numerical solution obtained through the FDM approaches the true solution of the SDE as the discretization step sizes in time and space are refined. In general, the convergence of numerical schemes can be characterized in two ways [208,209]: strong convergence, which focuses on the pathwise accuracy of the approximation, and weak convergence, which concerns the accuracy of the moments of the solution. These concepts of strong and weak convergence are extended to the numerical analysis of SDEs as well [200]. Therefore, it is still an active area of research [215,216] to determine how noise from quantum computing affect the stability and convergence analysis of SDEs.

### 16.3. Hamiltonian Simulation

The natural language of quantum computing is in terms of Hamiltonian representation [149], thus an interesting approach to solving a PDE of the form of Equation (31) is to convert it into a Schrödinger or Schrödinger-like equation. For example, in Ref. [217] showed that for the BSM model given by Equation (35), one can perform a change of variables  $S = e^x$  for  $x \in (-\infty, +\infty)$  and reversal of time  $t \rightarrow \tau = T - t$  to reformulate Equation (35) into a Schrödinger-like equation:

$$\frac{\partial V}{\partial \tau} = i\hat{H}V, \quad (39)$$

where  $\hat{H}$  is a non-Hermitian Hamiltonian given by

$$\hat{H} = i\frac{\sigma^2}{2}\hat{p}^2 - \left(\frac{\sigma^2}{2} - r\right)\hat{p} + ir\mathbb{I}. \quad (40)$$

Here,  $\hat{p} = -i\frac{\partial}{\partial x}$  is the momentum operator. Since the associated propagator  $\hat{U}(\tau) = e^{i\tau\hat{H}}$  is non-unitary, the approach of [217] is to decompose  $\hat{H}$  into a sum of Hermitian and anti-Hermitian part,  $\hat{H} = \hat{H}_H + \hat{H}_A$ , with

$$\hat{H}_H = -\left(\frac{\sigma^2}{2} - r\right)\hat{p}, \quad \text{and} \quad \hat{H}_A = i\left(\frac{\sigma^2}{2}\hat{p}^2 + r\mathbb{I}\right). \quad (41)$$

Since, the two parts of  $\hat{H}$  commutes with each other  $[\hat{H}_H, \hat{H}_A] = 0$ , then via the Baker-Campbell-Hausdorff formula [218] the evolution operator can be written as  $\hat{U}(\tau) = e^{i\tau\hat{H}_H}e^{i\tau\hat{H}_A}$ . The non-unitary  $\hat{O} = e^{i\tau\hat{H}_A}$  is then embedding it into an enlarged Hilbert space requiring one additional ancillary qubit, such that the corresponding unitary evolution  $\tilde{U}(\tau)$  is given by

$$\tilde{U}(\tau) = \begin{pmatrix} \hat{O} & \sqrt{1 - \hat{O}^2} \\ \sqrt{1 - \hat{O}^2} & -\hat{O} \end{pmatrix} \quad (42)$$

A proof-of-concept implementation using 8 qubits on a simulator got results in good agreement with analytical solutions for the BSM model[217].

The major drawback with the approach of [217] is that the embedding technique doubles the size of the problem. An alternative Hamiltonian simulation is to make an additional change of variables [216,219]:

$$\tau = \sigma^2(T - t), \quad \text{and} \quad v(x, \tau) = \exp(-ax - b\tau)V(t, S_t), \quad (43)$$

where  $a, b$  are constants given by the BSM model. This allows for the mapping of Equation (35) into the heat equation of the form [216,219]:

$$\frac{\partial v}{\partial \tau} = \frac{1}{2} \frac{\partial^2 v}{\partial \tau^2}. \quad (44)$$

A Wick rotation is performed on the time variable  $\tilde{\tau} \rightarrow -i\tau$  which then can be mapped to a Schrödinger-like equation in imaginary time:

$$\frac{\partial v}{\partial \tilde{\tau}} = -\hat{H}_I v, \quad (45)$$

where the operator  $\hat{H}_1$  is given by

$$\hat{H}_1 = -\frac{i}{2}\hat{q}^2, \quad \text{for } \hat{q} = -i\frac{\partial}{\partial x}. \quad (46)$$

This simulation can be performed using the Quantum Imaginary Time Evolution (QITE) which was first proposed for quantum chemistry problems [220] but has also been applied to other quantum simulation problem [221]. State-of-the-art methods for QITE include accelerated approaches using random measurements [222], and an alternative approach that uses orthogonal basis states to efficiently express the propagated state [223]. The Schrödingerisation method of [219] can be applied to general linear differential equations including the Fokker–Plank equation [224] which describes the time evolution of the probability density function of the velocity of a particle under the influence of drag forces and random forces. Hence, methods of solving SDE via the Fokker–Plank formulation can also be handled by the aforementioned quantum algorithm [216].

### Other Approaches

It is worth mentioning other quantum approaches to solving SDEs, in particular the the variational algorithm of Ref. [161]. In this approach, they first approximate the target SDE by a trinomial tree structure with discretisation to obtain a linear differential equation describing the probability distribution of SDE solutions. The resulting differential equation is then solved by formulating it as the time-evolution of a quantum state embedding the probability distributions of the SDE variables. This approaches utilises the Fokker–Plank equation [224] which also gives the time evolution of the probability distributions of SDE solutions. A proof-of-concept implementation of this approach was shown to be in close agreement with classical methods [161], but the reported scaling of this method does not seem promising for achieving Practical Quantum Advantage for industrial-scale problems.

## 17. Quantum Optimisation

Broadly speaking, optimisation is the act of modifying a process to extremise some property, which might be the occurrence of favourable outcomes, the profit of some business model, expected returns of a portfolio, or any number of other things [225,226]. Performing an optimisation requires us to make some assumptions about the real world. For example, an investor who wishes to optimise their portfolio to maximise returns or minimise risk could begin by stating assumptions about present and future market conditions, such as risk or volatility, and comparative returns [204,227]. Since there is no way to accurately calculate these in real-time, the proper estimation of these values is vital [181,204].

In the financial sector, optimisation is the process of making a financial system more effective by adjusting the variables used for technical analysis, which might involve, for example, reducing certain transaction costs, risks or targeting assets with greater expected returns [228]. Depending on the underlying assumptions or the target of the optimisation, there may be multiple potential optimisation procedures. The success of the chosen procedure will depend on how well one has estimated or quantified the risks, costs, and potential payouts.

Optimisation problems are, therefore, almost ubiquitous, but often very time-consuming or computationally expensive to solve given the number of variables involved [229]. Their high impact, combined with the existence and relative maturity of many quantum algorithms for solving them [150,157,230–232], means there is a lot of potential value in exploring commercially relevant applications on NISQ hardware [150,157]. Furthermore, improvements in hardware mean that the prospect of thorough and reliable benchmarking of various quantum algorithms is closer to reality. A recent paper [233] presented ten optimisation problem classes, as well as a Quantum Optimisation Benchmark Library for recording problem instances and solution track records, with standardised reporting and comparisons to the classical state of the art. This and similar efforts allow more rigorous comparison of different quantum optimisation algorithms, clarifying competing claims of suitability or performance, and enabling laypersons to more confidently explore and implement solutions.

This section discusses how such quantum algorithms may be used to solve two types of optimisation problems [152]: (i) combinatorial optimisation and (ii) convex optimisation. The former is a discrete optimisation problem, consisting of finding the optimal combination of values for some cost function, while the latter is a continuous optimisation problem involving optimising some convex cost function.

## 18. Combinatorial Optimisation Problems

This section presents quantum approaches for solving complex combinatorial optimisation problems which fall under the NP-hard computational class [234,235]. It is believed that there is potential for quantum advantage in this field, with quantum computers approximating solutions faster, providing higher quality solutions, or a mix of both [236]. However, this has not yet been demonstrated on large scale real world problems, and there remain significant theoretical and experimental challenges to overcome.

In general, combinatorial optimisation is a subfield of optimisation where the set of feasible solutions is, or can be reduced to, a discrete set. Typical combinatorial optimisation problems are the travelling salesman problem, the minimum spanning tree problem, and the knapsack problem. In many such problems, exhaustive search is not tractable, and approximation algorithms, or specialised algorithms that quickly rule out large parts of the search space need to be used instead [236].

In the literature, combinatorial optimisation is closely interlinked with Integer Programming (IP), which concerns optimisation or feasibility problems, usually NP-hard, where some or all of the variables are restricted to be integers [235]. A specific subcategory of IP is binary problems, i.e., problems in which the variables are restricted to be either 0 or 1. Trivially, all integer problems can also be represented as binary problems by expressing the integers in binary, at the cost of a logarithmic increase in problem size.

### 18.1. QUBO Formulation

Many IP and combinatorial optimisation problems, such as ones encountered in finance, can be reduced to a set of problems called Binary Quadratic Programming (BQP) problems with at most quadratic cost functions and constraints. By definition, a BQP can be written in the following general format:

$$\min f(x) = \sum a_i x_i + \sum b_{ij} x_i x_j \quad \text{s.t.} \quad \sum c_i^m x_i + \sum d_{ij}^m x_i x_j + e^m \circ 0 \quad (47)$$

where  $x_i$  are binary variables,  $a_i$  and  $b_{ij}$  are the coefficients for the objective function,  $c_i^m, d_{ij}^m, e^m$  are the coefficients for the  $m$ -th constraint, and  $\circ \in \{<, \leq, =, \geq, >\}$ .

A Quadratic Unconstrained Binary Optimisation (QUBO) problem is a BQP with no constraints [237]. It can be written as

$$\min f(x) = \mathbf{x}^T Q \mathbf{x} \quad (48)$$

where  $\mathbf{x}$  is a binary vector and  $Q$  is a real symmetric matrix [237]. BQP problems can be converted to QUBOs using the Lagrangian formulation where each constraint is converted into a penalty term with the appropriate Lagrangian parameter. This ensures that the optimal result is preserved while penalising infeasible solutions [226,238,239]. For example, if we have some constraint  $Ax = b$ , then the corresponding Lagrangian penalty term added to the QUBO would be  $\lambda(Ax - b)^2$ , which penalises infeasible solutions by an amount determined by  $\lambda$ .

It is known that QUBOs are very closely related (and computationally equivalent) to the Ising model, and solving them is similar to finding the ground state of the generalised Ising Hamiltonian [239]. The classical Hamiltonian of such a model is defined as:

$$H = - \sum_{ij} J_{ij} \sigma_i \sigma_j - \sum_j h_j \sigma_j, \quad (49)$$

Where  $J_{ij} \in \mathbb{R}$  represents the interaction between the sites  $i$  and  $j$ , and the spin variables  $\sigma_i$  are in the set  $\{-1, 1\}$ . Moreover, in the Ising model the variables are typically arranged in a lattice where only neighbouring pairs of variables  $ij$  can have non-zero coefficients. One can then follow the substitution  $\sigma_i = 2x_i - 1$  in order to define a QUBO cost function, similar to the form of Equation (47), thus transforming it to a classical Ising Hamiltonian.

This classical Hamiltonian can be quantised by replacing the classical variable  $\sigma_j$  with the Pauli-Z operator  $\sigma_j^z$ . Thus, finding the optimal solution for the QUBO is equivalent to finding the ground state of the corresponding Ising Hamiltonian [239]. This equivalence can be shown from Equations (47) and (49) where:

$$\begin{aligned} H &= \sum_{ij} -J_{ij}(2x_i - 1)(2x_j - 1) - \sum_j h_j(2x_j - 1) \\ &= \sum_{ij} (-4J_{ij}x_i x_j + 2J_{ij}x_i + 2J_{ij}x_j - J_{ij}) - \sum_j (2h_j x_j - h_j) \\ &= \sum_{i=1}^N \sum_{j=1}^i q_{ij} x_i x_j + C. \end{aligned} \quad (50)$$

In the last line of Equation (50) we have used

$$q_{ij} = \begin{cases} -4J_{ij} & \text{if } i \neq j \\ \sum_{ki} 2J_{ki} + \sum_{il} 2J_{il} + 2h_i & \text{if } i = j \end{cases} \quad (51)$$

and

$$C = - \sum_{ij} J_{ij} - \sum_j h_j. \quad (52)$$

As the constant  $C$  does not change the location of the optimum  $x$ , it can be neglected for optimisation and is only important for recovering the value of the original Hamiltonian [239]. In the ensuing subsections, we will review quantum algorithms for solving the optimisation problem defined by Equation (50) that are implementable in NISQ devices.

## 18.2. Variational Quantum Eigensolver

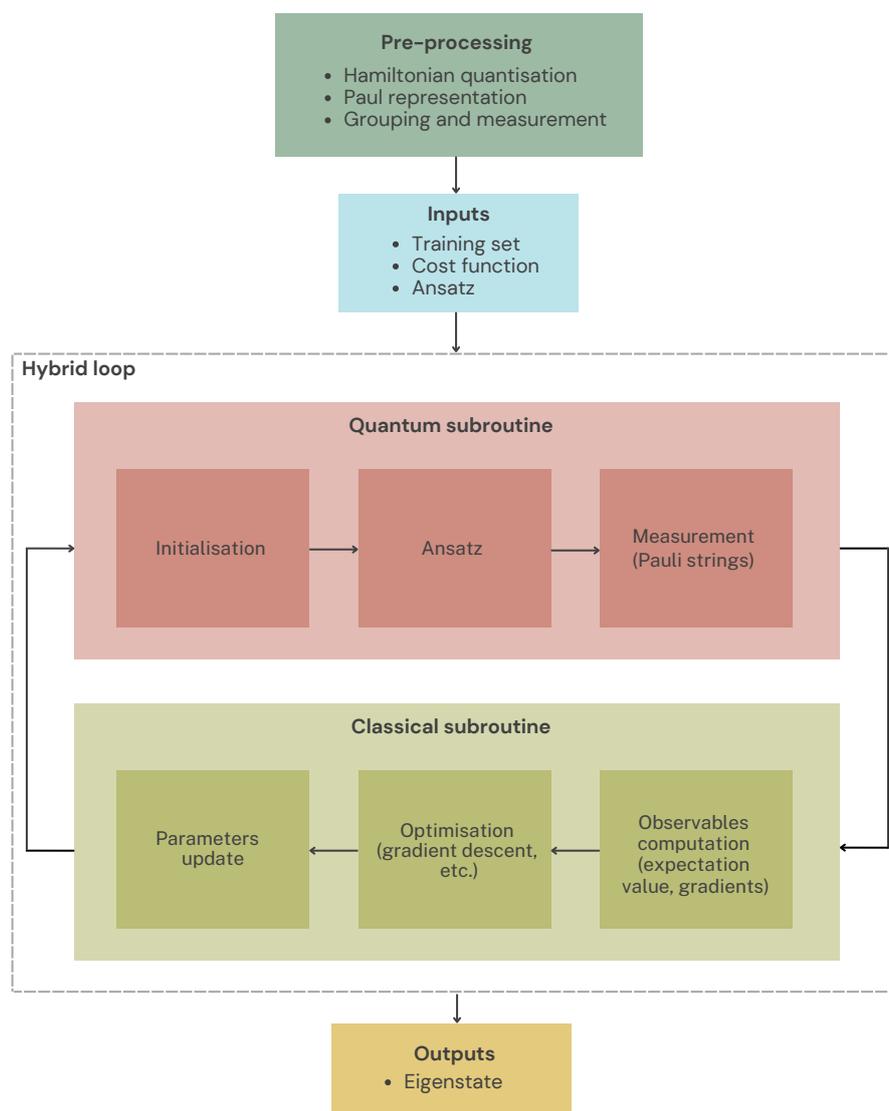
As discussed in Section (12), the Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm that is used for optimisation problems. First proposed by Peruzzo et al. [150], and then extended and formalised by McClean et al. [240], the VQE is among the most widely used algorithm on NISQ hardware. Originally formulated with the aim to estimate the ground state of some Hamiltonian by employing the *variational method* (and more precisely in the Rayleigh-Ritz functional [241,242]) which finds an upper bound for the lowest possible expectation value of an observable with respect to a trial wavefunction. In this formulation, VQE can also be used as the first step in simulating molecules and materials, among other applications [152]. As noted in Section (12), the VQE algorithm can be used for find the optimal parameters

$$\theta^* = \arg \min_{\theta} C(\theta), \quad (53)$$

for a general cost function  $C(\theta)$  generated from a parametrized circuit  $U(\theta)$  [152]. This makes VQE apt for various optimisation problem.

The VQE algorithm, as illustrated in Figure 3, typically starts with a pre-processing step, defining the system whose ground state the process is trying to identify. This could include constructing the Hamiltonian, and additional necessary steps (for example, encoding Pauli operators or measurement weighting [243]). The algorithm will then progress to the hybrid loop, where the classical subroutine handles the parameter updates (e.g. performing a gradient descent based on the returned values), while the quantum subroutine runs the circuit and returns the results. The hybrid loop takes a training set, a

cost function, and an ansatz as input; ansatz is the circuit used to perform the quantum mechanical estimation. visualises this process.



**Figure 3.** The main steps of the Variational Quantum Eigensolver.

A key (and challenging) part of the VQE is choosing a suitable ansatz for the problem at hand. Certain ansätze are more suitable for particular problems because they take into account prior information about the ground-state wave function(s) [150]. Alternatively, evolutionary and noise-resistant techniques exist to optimise the structure of the ansatz and significantly increase the space of candidate wave functions [244]. In a recent paper [245], the authors introduced quantum variational filtering, which can be used to increase the probability of sampling low-eigenvalue states of an observable when the filtering operator is applied to an arbitrary quantum state. Their algorithm, filtering VQE, was applied to combinatorial optimisation problems and outperformed both standard VQE and QAOA [245]. Finally, the Deep VQE (DVQE) method was introduced by [246], which applies a divide-and-conquer method in order to reduce the system dimensionality and solve the effective Hamiltonian further than the approaches. In other words, the DVQE approach can provide the ground state of a given quantum system with a smaller number of qubits.

Another key problem with variational algorithms is that of barren plateaus [153]. This refers to the presence of a flattened energy landscape, with variation only in the near neighbourhood of the optimal solution. As a result, the classical optimiser is unable to converge (see Section 12). A review on barren plateaus can be found in [154] summarises the issue as arising due to the *curse of*

*dimensionality*, where the very strength of quantum computing, the ability to manipulate and compute in a higher-dimensional space, also hinders variational algorithms. This can be circumvented using a variety of techniques, including problem-aware encodings and formulations, and specified ansatz architectures, but it remains an open question as to whether these further restrictions reduce the potential for quantum advantage via this algorithm [154].

### 18.3. Variational Quantum Imaginary Time Evolution

The classical optimisation routine for the VQE (see Section 18.2) attempting to find the ground state in a high-dimensional, non-linear parameter space has a high risk of getting stuck in a local minimum. In order to circumvent this, recent solutions have been using algorithms based on Quantum Imaginary Time Evolution (QITE) [194,220,245,247–250]. This method, developed by [247], has the ability to transform an arbitrary quantum state to the (almost) exact ground state by performing an approximate unitary evolution that mimics the imaginary time propagation. In other words, it is an alternative approach to preparing ground states on quantum computers.

The main idea of QITE is to represent the imaginary time propagator of a system by unitary operators via least-square fitting [194]. It has been shown that one can simulate the QITE on a quantum computer using a variational approach in order to find the ground state energy of the system [220]. Essentially, the energy cost function of a fixed variational ansatz is minimised following the Variational QITE (VQITE) approach [220,251]. The advantage of this method is that it can maintain a fixed circuit depth for the number of imaginary time steps, but the downside is its limited accuracy in finding the ground state. This accuracy very much depends on the variational ansatz, which can be system dependent [252,253].

Given an initial arbitrary state,  $|\psi\rangle$ , the normalised QITE can be defined as [220]

$$|\psi(\tau)\rangle = A(\tau)e^{-H\tau} |\psi(0)\rangle, \quad (54)$$

where  $H$  is the Hamiltonian of the system,  $|\psi(0)\rangle$  is the initial state for time  $\tau = 0$ , and  $A(\tau) = 1/\sqrt{\|\langle\psi(0)|e^{-2H\tau}|\psi(0)\rangle\|}$  is a normalisation factor. The imaginary time  $\tau = it$ , for  $t \in \mathbb{R}$ , is known as a Wick rotation transforming the evolution from relativistic Minkowski spacetime to Euclidean space. The imaginary time dynamics follow the Wick-rotated Schrödinger equation:

$$\frac{\partial |\psi(\tau)\rangle}{\partial \tau} = -(H - E_\tau) |\psi(\tau)\rangle, \quad (55)$$

where  $E_\tau = \langle\psi(\tau)|H|\psi(\tau)\rangle$  results from enforcing normalisation. The solution to Equation (55) gives the time-evolved state as  $|\psi(\tau)\rangle = C(\tau)e^{-H\tau} |\psi(0)\rangle$ , where  $C(\tau)$  is a time-dependent normalisation constant [251]. As  $\tau \rightarrow \infty$ , the smallest eigenvalue of the non-unitary operator  $e^{-H\tau}$  dominates the system so that the state approaches a ground state of  $H$  [220].

In the VQITE, one can use a parametrised ansatz for the time-dependent state as  $|\psi(\theta(\tau))\rangle$  where the parameters  $\theta(\tau)$  vary with time (i.e.,  $\theta_1(\tau), \theta_2(\tau), \dots$ ). This can then be composed with a parametrised circuit where the evolution is projected on the circuit parameters. A limitation of the variational method comes from the fact that the ansatz may not be general enough to describe all states on the desired subspace of the full Hilbert space [254]. Nonetheless, results from [220] show that this is a robust routine for energy minimisation, as long as the errors due to imperfect ansatzes do not cause the routine to get trapped in local minima. In the finance sector, VQITE has been used to simulate the Feynman–Kac partial differential equation [255].

In a more recent paper, an alternative method for VQITE has been developed, which is gradient-free [194], thus avoiding expensive computations that may occur during the optimisation (such as matrix inversion or tensor computations). The VQITE can be used to solve a combinatorial optimisation problem by letting  $H$  encode the combinatorial cost function, in a similar manner to QAOA and VQE. It can also be used for more general optimisation problems, i.e., not only for solving QUBO. Finally, an alternative approach called Adaptive VQITE has been developed by [194], which circumvent the

challenge of finding a good ansatz by iteratively expanding it along the dynamical path, keeping certain metrics (i.e., the MacLachlan distance) under a specific threshold. This ensures that the state can follow the quantum imaginary time evolution in the Hilbert space of the system rather than restricting to a fixed variational ansatz.

#### 18.4. Quantum Approximate Optimisation Algorithm

First introduced by Farhi et al. in [157], the Quantum Approximate Optimisation Algorithm (QAOA) is a quantum algorithm for solving combinatorial problems. It is a VQA and draws inspiration from the concept of adiabatic evolution while being able to run on gate-based quantum computers (as opposed to analogue or annealing devices), characterising it as a finite, Trotterised version of annealing [157]. Hence, QAOA can solve a broader class of problems than quantum annealing, adhering to the universal quantum computation model.

The QAOA is particularly used for problems that can be cast as searching for an optimal bitstring  $z^*$  such that

$$z^* = \arg \min_{z \in \{0,1\}^n} C(z), \quad (56)$$

where  $C(z)$  is the cost function. The algorithm first defines a cost Hamiltonian  $H_C$ , which encodes the solution  $z$  to its ground state. Finding this ground state is a hard problem [157]. In order to approximate it, the QAOA prepares an ansatz wavefunction parametrised by a parameter family  $(\gamma, \beta)$ . This ansatz is embedded into a classical optimisation loop, which finds the optimal values for these parameters by utilising what is known as a mixer Hamiltonian  $H_B$ , which 'mixes up' the quantum state to increase the space searched. It has been shown [157] that good approximate solutions to the problem can be found by preparing the variational state:

$$|\gamma, \beta\rangle = \underbrace{U_B(\beta_p)U_C(\gamma_p) \dots U_B(\beta_1)U_C(\gamma_1)}_p |s\rangle = \prod_{k=1}^p e^{-i\beta_k H_B} e^{-i\gamma_k H_C} |s\rangle, \quad (57)$$

where  $|s\rangle$  is a suitable initial state,  $U_B(\beta_k) = e^{-i\beta_k H_B}$ , and  $U_C(\gamma_k) = e^{-i\gamma_k H_C}$ . In Equation (57), the variable  $p$  can be interpreted as a hyperparameter and defines the number of layers of QAOA blocks. The QAOA with  $p$ -layers, has  $2p$  classical parameters to optimise over since each layer  $k$ , is characterised the set  $\{\gamma_k, \beta_k\}$ .

The preparation step outlined above is followed by a measurement on the computational basis, giving a classical string  $z$ , with which one can evaluate the objective cost function  $C(z)$  of the combinatorial problem. Over many repeated measurement samples, one can estimate the expectation value

$$E(\gamma, \beta) = \langle \beta, \gamma | H_C | \gamma, \beta \rangle, \quad (58)$$

that is provided to the classical optimiser to update  $\{\gamma_k, \beta_k\}$ . Repeating this procedure will provide an optimised string  $z^*$ , with the quality of the result improving as the number of QAOA  $p$ -layers is increased [157]. In principle, assuming absence of noise and other quantum device imperfections, the QAOA can reach the global optimum of any cost function in the limit  $p \rightarrow \infty$  [157], approaching the adiabatic protocol. However, in practice, errors on NISQ devices limit the number of  $p$ -layers that can be implemented coherently. Algorithm (9) gives a concise outline of QAOA for general combinatorial optimisation problem.

---

**Algorithm 9** Quantum Approximate Optimisation Algorithm

---

**Initialising** Define a cost Hamiltonian  $H_C$ , which encodes the solution to the problem as its ground state and a mixer Hamiltonian  $H_B$ .

**Construct unitaries** Construct the Unitaries  $U_B(\beta)$  and  $U_C(\gamma)$ , as defined above.

**Combine unitaries** Construct the unitary  $U_p(\gamma, \beta) = \prod_{k=1}^p U_B(\beta_k)U_C(\gamma_k)$  for a number of layers  $p \geq 1$ .

**Run circuit** Prepare an initial state,  $|s\rangle$  and apply the unitary  $U_p(\gamma, \beta)$  as defined in the previous step.

**Measurement** Measure the final state, estimate the expectation value, and report to the classical optimiser.

**Optimisation loop** Repeat the above steps while the classical optimiser optimises the parameters.

**Result** After many loops, the result will be an approximate solution to the optimal bitstring.

---

The structure of the QAOA ansatz allows for finding suitable candidates for the parameters  $\{\gamma_k, \beta_k\}$  purely classically for certain problem instances [157]. In general, however, finding such parameters is a challenging task as the training of VQAs is NP-hard [152,163,164]. Furthermore, just like other VQAs the QAOA is also susceptible to barren plateaus [152,153]. However, ongoing research has uncovered promising efficient approaches for finding optimal parameter and required circuit depth estimation [256–259].

### 18.5. Quantum Minimum Search

It has been shown in [260] that the Quantum Unstructured Search algorithm (presented in Section (9) in the context of Grover’s algorithm) can be used to find the global minimum of a function operating as a black box. The algorithm finds the index of an item whose value is smaller than the last by a particular threshold index. The resulting value is the next threshold index, and the process continues until the probability of the threshold being the global minimum is sufficiently large. Given an array of marked entries of size  $N$ , the algorithm exhibits a complexity of  $\mathcal{O}(\sqrt{N})$ , and at the end, will yield the index of the minimum value with a probability of at least  $1/2$ .

The original Quantum Minimum Search (QMS) algorithm developed by [260] is described in Algorithm (10). The probability of the algorithm can be improved by running the above routine a number of times. The authors show that by running the algorithm  $t$  times, one can obtain the minimum  $y$  with probability at least  $1 - 1/2^t$ .

One issue with the above algorithm is that it relies on knowledge of the answer space in order to formulate and solve the problem, for example in identifying a feasible encoding into the quantum state space. Solution space knowledge tends to be unavailable for real-world problems, and finding an encoding may have to be done on a case-by-case basis. However, research by Zeng et al. [261] is aimed at alleviating these problems by introducing both a novel encoding mechanism, and an adaptive algorithm used to estimate the number of solutions. These methods were demonstrated on a superconducting qubit device, making it more likely that QUS will be feasible on NISQ-era devices.

---

**Algorithm 10** Quantum Minimum Search

---

**Initialising.** Choose a threshold index  $y$  uniformly at random as  $0 \leq y \leq N - 1$ .

**Threshold and marking.** Initialise the threshold memory as  $\sum_j \frac{1}{\sqrt{N}} |j\rangle |y\rangle$  and mark every item  $j$  whose value is less than the value of  $y$ .

**Quantum search.** Apply the quantum search algorithm described in [262].

**Update threshold.** Let  $y'$  be the outcome of the previous step. Observe the outcome and if  $y' < y$ , then set the threshold to  $y'$ .

**Repeat.** Repeat for  $\mathcal{O}(\sqrt{N})$  from step 2.

**Measurement.** Measure in order to obtain the global minimum,  $y$ , with probability at least  $1/2$ .

---

It is evident that Algorithm (10) works in a similar way to Grover’s algorithm, providing a quadratic speed-up with the same requirements, i.e., a black box approach. The quantum oracle in this case can be described as  $O_{f_y} |x\rangle = (-1)^{f_y(x)} |x\rangle$ , where  $f_y : \mathcal{X} \mapsto \{0, 1\}$  is a classical function

mapping  $f_y(x)$  to 1 if  $y' \leq y$  and 0 otherwise. [263] has also proposed an efficient implementation of the oracle  $O_{f_y}$ .

### 18.6. Quantum Annealing

As introduced above (see Section 13), Quantum Annealing (QAnn) is an approach to quantum computing that is based on the adiabatic model and is best suited to solving QUBO problems. Quantum annealers are available today from providers such as D-Wave [175], and can be used to solve several classes of optimisation problems, including a range of financial and economic optimisation problems. For example a recent paper [264] studied the application of annealing to portfolio optimisation and found that their hybrid method outperformed a classical benchmark. A comprehensive review of annealing for random combinatorial optimisation problems is given in another paper [265].

The Quantum Annealing model consists of adiabatically evolving a system according to the transverse-field Ising Hamiltonian [166]

$$H(t) = A(t) \left( - \sum_i \sigma_i^x \right) + B(t) \left( - \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z \right), \quad (59)$$

where  $J_{ij}$  are the coupling terms of an Ising Hamiltonian,  $h_i$  are magnetic field strengths,  $A(t)$  and  $B(t)$  are functions that control the strength of the transverse magnetic field and the terms within the parentheses are the ones defined in Section (13) as  $H_0$  and  $H_1$ , respectively.

Annealers work by adiabatically shifting the Hamiltonian of the system from some initial  $H_0$ <sup>10</sup> to the problem  $H_1$  Hamiltonian. If the system starts in the ground state of  $H_0$ , the final state will be the ground state of  $H_1$ , encoding the solution of the problem [166,167].

Since the QUBO form can be trivially mapped<sup>11</sup> onto the Ising problem Hamiltonian, quantum annealers (QPUs designed to perform Quantum Annealing) are the native devices of solving QUBO problems. However, there are often problem-specific challenges that must be overcome, including the presence of a too-small energy gap between the ground and excited states, causing a failure of adiabaticity [266], and the need to embed a problem on the hardware's qubit graph [267].

This latter problem in particular can cause issues. As the connectivity of the physical qubits in annealers is fixed, it may not directly correspond to the connectivity of the variables in the problem Hamiltonian [268]. Finding this mapping is an NP-hard problem and can be the main bottleneck in solving problems using annealing [267]. However, tools have been provided by DWave and other companies to automate this process [269], and it is a field of ongoing research.

There are peculiar advantages of quantum annealers as opposed to gate-based QPUs. Since annealers work by 'sweeping' from the initial  $H_0$  to problem  $H_1$  Hamiltonian, they can execute several hundred sweeps very rapidly, returning a set of possible solutions ready for post-processing [270]. This contrasts with gate-based methods such as VQE, which require running lengthy circuits and take much longer [152]. However, annealers are not universal quantum computers, and thus the set of algorithms they can execute is much more limited than gate-based computers, and it is an open question as to whether one technique tends to provide better solutions than the other. For example, a 2024 paper by Q-CTRL [271] claimed to demonstrate that a digitised version of annealing running on IBM devices could outperform annealing for Max-Cut and spin-glass problems in terms of probability of finding the lowest-energy state. However, a later comment by DWave [272] claimed instead that with improvements to the annealing workflow, they were able to outperform the gate-based method in both time to solution and ground state probability. As a result, it is often unclear which method is most suitable for a given problem at the current time, but it is to be hoped that as the field matures, the delineation will become clearer.

<sup>10</sup> Normally chosen to have a ground state which is an equal superposition

<sup>11</sup> By the simple map  $\{-1, 1\} \mapsto \{0, 1\}$  of binary Ising variables to QUBO variables

Separately, Quantum Annealing has inspired similar protocols for digital (gate-based) quantum computers, an example of which is the family known as Counteradiabatic Quantum Optimisation algorithms [273]. These work by adding extra terms to the cost Hamiltonian which have the effect of suppressing excited-state transitions. The latest modification of this algorithm, Bias-Field Digitized Counteradiabatic Quantum Optimization (BF-DCQO) [274], has been used to solve Ising models [275], displaying better performance than either traditional annealing or variational methods like VQE. Another disadvantage of annealing is that it requires problems to be reduced to QUBOs, which can introduce significant overheads. The BF-DCQO algorithm has also been used to solve Higher-order Unconstrained Binary Optimisation (HUBO) problems [276] such as solving protein folding, consistently achieving optimal solutions.

## 19. Convex Optimisation Problems

Convex optimisation problems are those which optimise a convex function over a set of solutions that is given explicitly (i.e., through a set of constraints) or implicitly (i.e., through an oracular function) [226]. Convex optimisation is less common than combinatorial optimisation in finance and economics, but is nevertheless important, and so we provide a brief summary below.

Generally, the optimisation problem under consideration is NP-hard and refers to minimising a convex function  $f : K \rightarrow \mathbb{R} \cup \{\infty\}$ , where  $K \subseteq \mathbb{R}^n$  is a convex set. If the convex function is bounded on  $K$ , one can equivalently consider the problem of minimising a linear function over a different convex set  $K' \subseteq \mathbb{R}^{n+1}$ . Accessing  $K'$  is easy given knowledge of  $f$  and  $K$ , and the parameters involved will be similar. Conversely, for any linear optimisation problem over an unknown convex set  $K$ , there is an equivalent optimisation problem over a known convex set with an unknown bounded convex objective function  $f$  that can be evaluated easily given access to  $K$  [226].

Quantum algorithms for convex optimisation are an active field of research. In earlier stages of the field, a study by [277] presented a quantum algorithm for minimising quadratic functions. In more recent work, there have been suggestions of potential future quantum advantage for semidefinite optimisation, an important subset of convex optimisation problems [278–281]. The quantum algorithms for these problems have been shown to have runtimes that scale polynomially with the desired precision and some geometric parameters [280,281]. In other convex problem classes, many optimisation problems can be solved classically with a logarithmic scaling complexity using interior-point methods [282]. Currently, there is one quantum speed-up which resides partially within this regime of logarithmic scaling algorithms, developed by [142] using interior-point methods and Quantum Linear Systems Solver algorithms<sup>12</sup> (such as improved HHL-type [131]) and tomography to accelerate the computation of the Newton linear system. Further speed-ups to semidefinite convex optimisation problems have also been proposed using multiplicative-weights methods [279,280].

Quantum algorithms have also been formulated for other classes of convex problems, such as, linear programming or second-order cone programming [236]. In the former case, the algorithms include the the quantum interior-points methodology developed by [283] and the simplex method developed by [284]. Under certain conditions, the quantum interior-points method have been shown to provide a potential polynomial speed-ups for the second-order cone programming problem [285]. Finally, recent work by [279] implements subgradient approximation of convex functions via Jordan's algorithm for gradient computation [286]. This approach is quadratically better complexity than the best-known classical randomised algorithm, based on queries to the oracle [279]. However, it is an open question whether this classical quadratic bound is optimal.

## 20. Quantum Machine Learning

The field of Quantum Machine Learning (QML) is an interdisciplinary subset of quantum information that lies at the intersection of Machine Learning (ML), statistics and quantum physics [287–289].

<sup>12</sup> See Section (11) for more details on QLSSs algorithms.

It seeks to utilise the power of quantum computing to potentially augment or provide improved performance over classical ML algorithms [39,290], although some of these improvements are still theoretical and not yet demonstrated experimentally [291–293]. This includes hybrid methods that involve both classical and quantum processing, where some of the classically challenging subroutines are efficiently performed on quantum computers. Developing quantum algorithms for QML is an active area of research with rapid advances in recent years [294]. In this chapter, we will review some of the most common quantum algorithms for supervised and unsupervised QML, and Quantum Reinforcement Learning (QRL).

## 21. Quantum Algorithms for Supervised ML

Supervised machine learning is a type of machine learning that uses labelled data to train algorithms that can classify data or predict outcomes [295]. In supervised learning, the algorithm learns a mapping between the input and output data. This mapping is learned from a labelled dataset, which consists of pairs of input and output data [296]. The algorithm tries to learn the relationship between the input and output data so that it can make accurate predictions on new, unseen data [289]. For example, consider a dataset of images of animals, where each image is labelled with the name of the animal it contains. A supervised learning algorithm can be trained on this dataset to recognise the animal in a new image that it has not seen before.

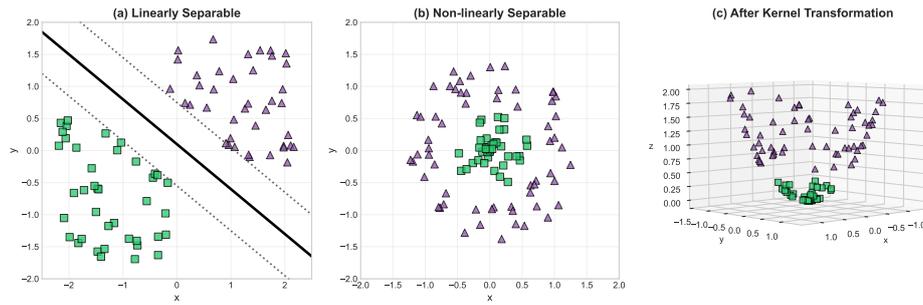
Supervised learning is widely used in various fields such as finance, healthcare, marketing, and more [297]. It can be used for regression or classification tasks [298], and it usually requires large amounts of labelled data to achieve high prediction accuracy in unseen input data [298]. The field of supervised machine learning is very broad, and quantum algorithms corresponding to classical ML are an active area of research [39,299,300]. Therefore, this section will not be a comprehensive review of all supervised QML algorithms but rather will present a few example algorithms that are representative of the broad field.

### 21.1. Quantum Classifiers

Quantum algorithms for classification problems have been an active area of research in recent years. These algorithms aim to improve the efficiency of solving classification problems in machine learning by using quantum computers. A detailed survey of recent works in this area can be found in Ref. [301]; thus, it suffices for us to outline a few common algorithms in this area.

#### 21.1.1. Quantum Support Vector Machine and Kernel Methods

In Machine Learning, Support Vector Machines (SVM) are supervised learning algorithms commonly used for classification and regression tasks. Given two categories (or classes) and a training set containing a number of data-points that are mapped to one of the classes, a SVM algorithm builds a model that assigns new data-point to either one of the two classes depending on their characteristics [302]. Figure 4 (a) is an example of binary classification, where the points that lie at the boundary of each of the margin class hyperplanes (shown with a dotted line) are called the *support vectors*, and they play a crucial role in solving the SVM classification problem. The black line inside the maximum margin hyperplane is the *margin normal vector* that divides the classes with the most significant possible marginal length [303].



**Figure 4.** Example of SVM binary classification. (a) Linearly separable dataset, (b) non-linearly separable case, and (c) the same data transformed into a higher dimensional feature space using kernel methods where linear separation becomes possible. Adapted from [301].

The classification problem can broadly be categorised into two types, with datasets that are: (i) **linearly separable** e.g., Figure 4 (a), and (ii) **non-linearly separable** e.g., Figure 4 (b). For the former class, the SVM binary classification algorithm seeks to find a hyperplane that divides the data-points into two parts by maximising the distance between the support vectors and the hyperplane [303]. For the latter class, one can still use the SVM algorithm by first transforming the data into a form that can be linearly separable. This is done by using the *kernel trick*, which maps the input data set into high-dimensional feature spaces where the data can be linearly separable [304]. It is through this complex process of transforming data into higher-dimensional space and processing it that quantum computing can offer a potential computational advantage over classical techniques. Hence various quantum algorithms for SVM have been theoretically proposed, although practical demonstrations of quantum advantage remain elusive. Below, there is a more in-depth definition of each classification.

#### Linear Classification

: Given a linearly separable training dataset [301]:

$$\mathcal{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_M, y_M)\}, \quad (60)$$

that has  $M$  points where  $\vec{x}_i \in \mathbb{R}^N$  and labels  $y_i \in \{-1, 1\}$ . Let the parameters  $(\vec{w}, b)$  define a hyperplane in  $N$ -dimensional space such each  $x_i$  falls in either  $\vec{w} \cdot \vec{x} + b \geq 1$  or  $\vec{w} \cdot \vec{x} + b \leq -1$  for the  $y_i = 1$  or  $-1$  categories respectively. The goal of SVM is to find a maximum separation (marginal length) between the two classes, which means finding optimal parameters  $\vec{w}^*, b^*$  that construct the decision function [301]

$$f(\vec{x}) = \text{sgn}(\vec{w}^* \cdot \vec{x} + b^*), \quad (61)$$

used to classify data-points in  $\mathcal{D}$  [301]. It is common to consider the *dual problem* [226] with Lagrange multipliers which seeks to maximise

$$L(\vec{\alpha}) = \sum_{j=1}^M y_j \alpha_j - \frac{1}{2} \sum_{j,k=1}^M \alpha_j (\vec{x}_j \cdot \vec{x}_k) \alpha_k, \quad (62)$$

subject to the constraints  $\sum_{j=1}^M \alpha_j = 0$  and  $y_j \alpha_j \geq 0$  [302]. The solution to this problem leads to the construction of the optimal weights and biases in the *primal problem* given by [301]

$$\vec{w}^* = \sum_{j=1}^M \alpha_j^* \vec{x}_j \quad \text{and} \quad b^* = y_k - \vec{w}^* \cdot \vec{x}_k, \quad (63)$$

for an index  $k$  corresponding to non-zero  $\alpha_k^*$  (support vectors) [301]. Therefore, we can substitute Equation (63) into Equation (61) to obtain

$$f(\vec{x}) = \text{sgn} \left( \sum_{j=1}^M \alpha_j^* (\vec{x}_j \cdot \vec{x}) + b^* \right). \quad (64)$$

In general, not all datasets are linearly separable such as the ones shown in Figure 4 (b), and thus, the SVM approach described above will not work. If the dataset is close to being linearly separable, which means it has a few outliers that do not obey the linearity, a common approach is to replace the hard-margin condition  $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$  with a soft-margin condition  $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \varepsilon_i$ , where  $\varepsilon_i$  is the cost [305]. In practice, the soft-margin condition is more often used than the hard-margin condition because most datasets generated in the physical world are noisy [301].

### Non-linear classification

: For datasets that are far from linearity, one can still use SVMs to efficiently perform a non-linear classification using what is called the *kernel trick* which maps the input data set into a high-dimensional feature spaces where the data can be linearly separable [304], as shown in Figure 4 (c). Let  $K(\vec{x}, \vec{y})$  be the kernel function given by [301]

$$K(\vec{x}, \vec{y}) = \phi(\vec{x}) \cdot \phi(\vec{y}), \quad (65)$$

where  $\phi : \mathbb{R}^N \rightarrow \mathbb{R}^{N'}$  is the *feature map* function that maps the  $N$ -dimensional feature space to a higher  $N'$ -dimensional kernel space. The kernel function is used to compute the inner product of vectors in the kernel space [306]. Thus,  $\phi$  does not need to be explicitly defined. Therefore, similarly to Eq.(62), one can use  $K$  to define the *dual problem* of maximising [301],

$$L(\vec{\alpha}) = \sum_{j=1}^M y_j \alpha_j - \frac{1}{2} \sum_{j,k=1}^M \alpha_j K_{jk} \alpha_k, \quad (66)$$

subject to the same constraints as Equation (62). Just like the linear SVM, one uses the optimal Lagrange multipliers  $\vec{\alpha}^*$  to formulate the decision function for the *primal problem* as [301]

$$f(\vec{x}) = \text{sgn} \left( \sum_{j=1}^M \alpha_j^* K(\vec{x}_j, \vec{x}) + b^* \right). \quad (67)$$

Besides the Lagrange multipliers, there are other techniques that can be applied, such as sequential minimal optimisation [306], interior-point methods [307] or gradient descent [308], which is one of the reasons SVM is popular [309]. However, these methods also faces several limitations, such as the fact that it does not handle large datasets well, the computational scaling with complexity, and its dependence on kernel functions.

The Quantum Support Vector Machine (QSVM) proposed by Rebentrost, Mohseni, and Lloyd [302] was one of the first QML quantum algorithms for supervised learning. QSVM can be implemented using several approaches, for instance using the quantum kernel method or quantum feature mapping method. It is an extension of the least-squares version of the classical SVM algorithm [310]. In this formulation, the SVM has equality constraints with slack variable  $e_j$  instead of inequality constraints, which means the hard-margin condition can be transformed as [301]

$$y_j(\vec{w} \cdot \vec{x}_j + b) \geq 1 \quad \implies \quad \vec{w} \cdot \vec{x}_j + b = y_j - y_j e_j. \quad (68)$$

One can then transform the associated Lagrangian of the *dual problem* into a system of linear equations [305], and then apply any Quantum Linear Systems Solver like the HHL-type algorithms described in Section (11). The QSVM algorithm uses quantum circuits to perform the classification task and has been theorised to potentially outperform classical SVMs for certain learning tasks [302]. The

QSVM algorithm consists of three main components: the *quantum feature map* used to map the input data to a quantum state in the kernel space, the *quantum kernel* used to compute the inner product of the quantum states in the kernel space, and the *quantum classifier* used to classify the input data based on the kernel function. In a higher dimensional space, the kernel function is assumed to be classically hard to compute, hence can be efficiently simulated on a quantum computer [302]. This leads to the idea of *covariant quantum kernels*, which have been hypothesised to potentially provide superpolynomial speed-ups for certain specialised problems [311].

#### Covariant Quantum Kernel

: Given an efficient algorithm  $U_{x_i}$  for loading features of the sample  $x_i$ , we can define the associated quantum feature state [312]

$$\Phi(x_i) = U_{x_i} |\psi\rangle \langle\psi| U_{x_i}^\dagger = U_{x_i} V |0\rangle \langle 0| V^\dagger U_{x_i}^\dagger, \quad (69)$$

where  $|\psi\rangle$  is a fiducial state generated by unitary  $V$  which depends on the learning problem and dataset. The quantum kernel for samples  $x_i, x_j$  is defined as [312]

$$K(x_i, x_j) = \text{Tr}(\Phi^\dagger(x_i)\Phi(x_j)) = \left| \langle 0| V^\dagger U_{x_i}^\dagger U_{x_j} V |0\rangle \right|^2. \quad (70)$$

The kernel  $K$  can be simulated on a quantum computer using circuits  $U$  and  $V$ , and the results used in the *dual problem* to speed-up the classification algorithm [311]. We note that the condition that quantum kernel methods can only be expected to do better than classical kernels if they are hard to estimate classically is necessary but not sufficient condition for an end-to-end quantum speed-up [313]. Designing quantum kernels that exploit structure in data is a promising step toward practical applications. Quantum kernels can be optimised on a given dataset with kernel alignment that exploits group structures in the dataset [312].

At the centre of this algorithm sits a non-sparse matrix exponentiation technique for performing matrix inversion of the kernel matrix. A low-rank approximation of the kernel matrix can be used due to the eigenvalue filtering procedure of HHL [6]. In addition, classical SVMs using quantum-enhanced feature spaces to construct quantum kernels have been proposed [314].

#### Variational QSVM

: An alternative proposition for training a QSVM variationally has been explored [314] aiming to target near- to medium-term quantum computers. Let  $\{\vec{x}\}$  denote the classical training data need to be loaded into a quantum computer using a feature map  $\Phi(\vec{x})$ , and  $W(\theta)$  be the parametrised quantum gate, and  $V(\Phi(\vec{x}))$  is the embedding. The PQC can be defined in such a way that, when a measurement is applied, it returns the label  $y \in \{-1, 1\}$  for each data point [314]. The PQC  $|\psi(\theta)\rangle = W(\theta)V(\Phi(\vec{x}))|0\rangle$ , can be trained to find good parameters  $\theta$  that optimises a cost function  $C(\theta)$ . This cost function is constructed such that it penalises misclassified training points. Typically, a classical optimiser, such as gradient descent, is used to update the parameters  $\theta$  and minimise the cost function.

In contrast to the covariant kernel approach which optimises for the associated Lagrange multipliers, the variational classifier approach learns a boundary hyperplane by optimising the parameters  $\theta$  the quantum circuit  $|\psi(\theta)\rangle$  [315,316]. Algorithm (11) summarises the general idea of the QSVM. Alternative to above quantum algorithms for classification have been proposed, which includes the quantum decision tree classifier [88,317–319], quantum nearest neighbour algorithm [299], and quantum annealing classifiers [315,316].

**Algorithm 11** Quantum Support Vector Machine**Input:** Labelled dataset  $\mathcal{D} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_M, y_M)\}$  with  $\vec{x}_i \in \mathbb{R}^N$ ,  $y_i \in \{-1, +1\}$ **Feature Map Encoding:** (a) Choose a quantum feature map  $\phi : \mathbb{R}^N \rightarrow \mathcal{H}$ . (b) Encode each  $\vec{x}_i$  into quantum state  $|\phi(\vec{x}_i)\rangle$ **Kernel Evaluation:** (a) For each pair  $(\vec{x}_i, \vec{x}_j)$ , compute kernel value  $K_{ij} = \langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle$ . (b) Use quantum circuits (e.g., swap test [320] or interference circuits) to estimate  $K_{ij}$ **Dual Optimisation:** (a) Solve the dual SVM problem:

$$L(\vec{\alpha}) = \max_{\vec{\alpha}} \sum_{i=1}^M y_i \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i K_{ij} \alpha_j,$$

subject to constraints:  $\sum_{i=1}^M \alpha_i y_i = 0$ , and  $\alpha_i \geq 0$ . (b) Convert the dual problem into a system of linear equations which could be solved using QLSS [131] for optimal  $\vec{\alpha}^*$ .**Classification:** Use optimal  $\vec{\alpha}^*$  to compute the decision function

$$f(\vec{x}) = \text{sgn} \left( \sum_{i=1}^M \alpha_i^* K(\vec{x}_i, \vec{x}) + b^* \right),$$

which is used to predict the label  $\hat{y} = f(\vec{x})$  for test data  $\vec{x}$ .

The advantage of QSVM lies in potential speed-ups for processing high-dimensional data. In particular, the proposal of [302] has a time complexity that is logarithmic in the dimension of the data  $N$  and number of data-points  $M$ , yielding a potential exponential speed-up over classical kernel methods [321]. Achieving such efficiency in processing high-dimensional data has been a motivation for continued research of QSVM. It includes searching for ways to overcome several challenges with QSVM such as finding a practical efficient data loading protocol, efficiently running the QLSS, and limitations of current NISQ devices [322]. These challenges are shared with general QML methods and they would need to be addressed QSVMs can become practically useful. However, QSVM have been demonstrated in small-scale experiments[323–325], but large-scale implementations relevant for industrial problems still awaits maturity of quantum hardware.

## 21.1.2. Quantum Nearest Neighbour

The nearest neighbour algorithm is a foundational and widely used classification technique in classical ML [326]. The algorithm works by having a large labelled dataset of examples, where each data point is represented by a feature vector. To predict the label or class of an unlabelled data point (also called test vector or query point), the algorithm measures the distance between this query and each dataset entry. It then assigns the label or class of the closest point, often using Euclidean distance [327]. This means that data points with similar features are likely to share the same label.

Formally, given a set of training vectors or examples  $\mathcal{D} = \{(\vec{x}_i, c_i)\}_{i=1}^M$ , where each  $\vec{x}_i \in \mathbb{R}^N$  is a feature vector in a  $N$ -dimensional space, and  $c_i \in \mathcal{C}$  denotes the class label of the point  $\vec{x}_i$ . The task is to classify a new unlabelled query point  $q$  by computing its distance to each training point  $\vec{x}_i$ . For any two points  $\vec{x}, \vec{x}' \in \mathbb{R}^N$ , the Euclidean distance can be defined as [327],

$$d(\vec{x}, \vec{x}') = \sqrt{\sum_{j=1}^N (x_j - x'_j)^2}. \quad (71)$$

One can also use other distance measures, such as the Manhattan distance, cosine similarity, or Hamming distance, but here we will focus on the Euclidean distance as it is commonly used [328]. Using this metric, each training example  $\vec{x}_i$  defines a 'neighbourhood' of nearby points, such that any new point  $\vec{q}$  falling inside this region, will be closest to  $\vec{x}_i$ . Hence, one needs to find the index  $i^*$  that minimises the distance [327],

$$i^* = \arg \min_{i \in \{1, \dots, M\}} d(\vec{q}, \vec{x}_i). \quad (72)$$

Therefore, the nearest neighbour classifier (sometimes called the 1-NN rule) will label  $q$  with the label of  $\vec{x}_{i^*}$ , which is the class of the training sample closest to  $q$  [329]. For the simple 1-NN classifier, only the single closest neighbour is considered to assign a class. However, this approach can be sensitive to noise or outliers [308,330]. To improve robustness, it is useful to take more than one neighbour when making decisions [331].

K-Nearest Neighbour (KNN):

Is one of the most widely used classifiers in practice, due to its simplicity and effectiveness across a range of tasks [329]. It is ‘non-parametric’ and instance-based, meaning it makes no assumptions about underlying distributions and simply uses the stored training instances for decisions [332]. Instead of relying on a single nearest example, KNN considers the  $k$  closest training points to the query and bases the prediction on the most frequent class among those neighbours. The 1-NN rule described above is a special case of KNN with  $k = 1$  ( $k$  is often chosen as an odd number to avoid tie votes).

KNN stores the training set in memory, and given the distance function, classifies a new point by finding the set  $N_k(q)$ , which is the neighbourhood of the query containing the indices of the  $k$  nearest samples to  $q$  [333]. The predicted class  $\hat{y}(q)$  is then the most common label among these neighbours. If we denote by  $n_c$  the number of neighbours in  $N_k(q)$  belonging to class  $c$ , the KNN classification rule can be expressed as [334],

$$\hat{y}(q) = \arg \max_{c \in C} n_c = \arg \max_{c \in C} \sum_{i \in N_k(q)} \mathbf{1}\{c_i = c\}, \quad (73)$$

meaning it selects the class  $c$  that maximises  $n_c$ , where  $\mathbf{1}\{\cdot\}$  is an indicator function taking the value 1 when its argument is true, and 0 otherwise [334].

This method is intuitive but can be slow with large datasets, with time complexity that is linearly proportional to the number of data points and their dimensions [333]. A key bottleneck arises because of the distances computations between the query and all  $M$  training points, resulting in an  $O(NM)$  computational cost for  $N$ -dimensional data (assuming distance calculation in  $O(N)$ ). When  $M$  is very large (as can happen in financial applications with millions of records) or for high  $N$  (e.g. high-frequency trading data with many features), this cost becomes significant, especially if many queries need to be answered in real-time [22,29,37]. This is where quantum computing may offer a potential advantage by simultaneously calculating multiple distances using quantum parallelism [335,336]. However, this advantage is thus far theoretical and contingent upon efficient implementations of quantum data loading techniques, which are not yet practically viable [337,338].

Quantum K-Nearest Neighbour (QKNN)

Is the quantum version of KNN [327], it computes the distances between data points more efficiently and identifies the nearest (or  $K$  nearest) neighbours with a potential quadratic speed-up over classical methods, especially for large datasets [288]. The general approach to QKNN, focus on key components, such as quantum data encoding, distance computation, and quantum search for nearest neighbours.

To represent the data in a quantum state, there are different encoding schemes, but a common choice for distance-based algorithms is amplitude encoding [339]. For example, a data point can be encoded as [340]:

$$|\mathbf{x}\rangle = \frac{1}{\|\mathbf{x}\|} \sum_{i=1}^N x_i |i\rangle,$$

where  $|i\rangle$  are computational basis states. Amplitude encoding is powerful because it packs  $N$  features into just  $\log_2 N$  qubits by using amplitude values. However, preparing such states for arbitrary data can be challenging and may require Quantum Random Access Memory (QRAM)<sup>13</sup> or complex

<sup>13</sup> QRAM is a mechanism to access data (quantum or classical) based on addresses which are themselves a quantum state [341].

rotations [337,342]. Once both the query point  $|x_q\rangle$  and a training point  $|x_i\rangle$  are encoded as quantum states, the distance between them can be related to the inner product of these states. For normalised vectors, the Euclidean distance is given by [336]

$$\|x_q - x_i\|^2 = 2(1 - \langle x_q | x_i \rangle). \quad (74)$$

Thus, knowing the inner product  $\langle x_q | x_i \rangle$  allows one to compute the distance. This overlap can be efficiently estimated on a QPU using a subroutine called the SWAP test, which measure the similarity or fidelity between two quantum states [320,343]. By repeating this test or using QAE, a good estimate of the inner product can be obtained [343].

In practice, many QKNN algorithms use the inner product as a proxy for distance. For instance, one may define a similarity score  $s_i = |\langle x_q | x_i \rangle|$  for each training point  $i$  and maximise the similarity  $s_i$  which is equivalent to minimising distance [344].

Once both the query point  $|x_q\rangle$  and training points  $|x_j\rangle$  are encoded, quantum circuits can be used to compute functions of their distance (via SWAP test [320] or a dedicated distance encoding routine) in superposition. One convenient choice is to produce an output state like [327]:

$$|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^N |x_j, c_j\rangle \otimes (\alpha_j |0\rangle + \beta_j |1\rangle), \quad (75)$$

where  $c_j$  is the class label of the  $j$ th training point, and the amplitudes  $\alpha_j, \beta_j$  depend on the distance between  $x_j$  and  $x_q$ . In an 'ideal' scenario,  $\alpha_j \approx 1$  (and  $\beta_j \approx 0$ ) for the nearest neighbours, while  $\alpha_j$  becomes small (and the ancilla is  $|1\rangle$  with high amplitude) for far-away points [344]. This way, the problem of finding nearest neighbours is encoded into quantum circuits where the solution is identifying the basis states that have an ancilla in state  $|0\rangle$  with high probability amplitude.

Once the data is encoded and distances are described in quantum amplitudes or probabilities, the next task is to actually identify the index of the  $k$  nearest neighbours, i.e., finding  $k$  index  $j$  with ancilla  $|0\rangle$ , meaning a small distance with high probability. One can do this by using the Quantum Unstructured Search (QUS)<sup>14</sup>, which can find a marked item in an unsorted list of size  $N$  in  $O(\sqrt{N})$  steps, which is a potential quadratic speed-up over classical linear search algorithms [4]. Algorithm (12) summarises the QKNN classification procedure outlined above.

---

#### Algorithm 12 Quantum K-Nearest Neighbour

---

**Input:** Labelled dataset  $\mathcal{D} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_M, y_M)\}$ , query point  $\vec{x} \in \mathbb{R}^N$

**Encode Input Data:** (a) For each  $\vec{x}_i \in \mathcal{D}$ , encode into quantum state  $|x_i\rangle$  (b) Encode query point  $\vec{x}$  as quantum state  $|x\rangle$

**Distance Estimation:** (a) For each  $i$ , estimate distance  $d(\vec{x}, \vec{x}_i) = \|\vec{x} - \vec{x}_i\|$  (b) Use a quantum subroutine (e.g. swap test) to estimate  $d$

**Nearest Neighbour Search:** Use QUS to find index  $j^*$  such that  $d(\vec{x}, \vec{x}_{j^*})$  is minimal

**Output:** Predicted label  $\hat{y}$  for input  $\vec{x}$

---

Classically, identifying  $k$  neighbours is  $O(N)$  in the worst case, whereas a QKNN algorithm can potentially achieve  $O(\sqrt{N})$  for fixed  $k$  or  $O(\sqrt{kN})$  in general case [326]. There has been improvements on the QKNN such as a proposal for a memory-efficient quantum subroutine for distance calculation, showing a 50% reduction in qubits required [344]. Furthermore, a recent work demonstrated an efficient swap test on photonic qubits [345]. While such an experiment not directly a QKNN implementation, the swap test [320] is a necessary component for many QKNNs and QML proposals as described here.

The QKNN has applications in finance include economic modelling (searching historical market data to find similarities to the present), or in fraud detection pipelines, rapidly and accurately classifying transactions as likely to be fraudulent or not [33,346,347].

<sup>14</sup> Also known as Grover's search algorithm, see Section (9)

### 21.1.3. Quantum Decision Tree

Decision trees are fundamental tools in classical ML for discovering features and extracting patterns due to their interpretability, fast inference, and natural compatibility with structured data [348]. They work by splitting data creating a tree-like model where each branch represents a decision based on discriminant functions or decision rules [349].

A decision tree defines a function  $f : \mathbb{X} \rightarrow \mathbb{Y}$  by sequentially testing feature values: at each node an attribute  $A$  is chosen as the splitting criterion, and the dataset is divided according to  $A$ 's possible values. This process continues until the subsets are homogeneous enough or other stopping criteria are met [350]. At that point a leaf node assigns a class label (for classification) or a numerical value (for regression) [351]. In other words, the decision tree recursively partitions the feature space into regions, using decision rules at each internal node to split data, until reaching a prediction at leaf nodes [348].

To 'grow' a decision tree, the algorithm must decide which feature provides the best split at each node. This is typically done by optimising an impurity measure or information gain. A common choice is based on Shannon entropy [352]; if  $S$  is the set of training samples at a node, the entropy of  $S$  with respect to the class labels is:

$$H(S) = - \sum_{c \in \mathcal{C}} p(c) \log_2 p(c), \quad (76)$$

where  $p(c)$  is the proportion of samples in  $S$  belonging to class  $c$ , and  $\mathcal{C}$  is the set of class labels [353]. Here, the entropy  $H(S)$  quantifies the uncertainty (impurity) in the node; it is 0 if all samples are of the same class and maximal when classes are uniformly mixed. A split on an attribute  $A$  produces child subsets  $S_v$  (for each value  $v$  of  $A$ ). The information gain achieved by splitting on  $A$  is defined as the reduction in entropy:

$$\text{InfoGain}(S, A) = H(S) - \sum_{v \in \text{Vals}(A)} \frac{|S_v|}{|S|} H(S_v), \quad (77)$$

where  $|S_v|$  and  $|S|$  denote the cardinalities (number of samples) of the child subset  $S_v$  and the parent set  $S$ , respectively, so that  $\frac{|S_v|}{|S|}$  represent the proportion of samples assigned to each branch. A high information gain means that splitting on  $A$  yields more informative subsets. At each new node, entropy is computed for that node's subset and all possible splits, and the best split is taken. This procedure stops when no attribute offers any gain (or other stopping conditions like maximum depth are reached), and then a leaf is created which is often labelled with the majority class in that subset. Other splitting criteria can also be used, such as the Gini impurity

$$G(S) = \sum_c p(c)[1 - p(c)], \quad (78)$$

as a measure of node impurity instead of entropy, yielding a similar formula for Gini gain [348]. Regardless of the metric, the goal is to choose splits that most improve homogeneity in child nodes. This greedy heuristic approach, tends to produce good results in practice, although do not guaranteed to find the global optimum tree, as this is NP-hard [354,355]. The time complexity for building a classical decision tree with  $M$  samples and  $N$  features is typically  $O(M, N, \log M)$  or  $O(NM)$  per node [356]. In the worst case, if the tree is fully grown with  $M$  leaves, the training cost can be  $O(M^2N)$ . Quantum algorithms that can improve this classical scaling have been explored [357].

The Quantum Decision Tree (QDT) algorithm [88,318] aims to enhance the decision tree induction process using quantum subroutines. As this involves encoding classical data, one common approach is to use QRAM data structure [337]; for example, if we have  $M$  data samples  $(\vec{x}_i, y_i)$  where  $\vec{x}_i$  are feature vectors and  $y_i$  are class labels (as shown in Section 21.1.2), one could prepare a state of the form:

$$|\Psi_{\text{data}}\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^M |i\rangle |x_i\rangle |y_i\rangle. \quad (79)$$

A crucial part of building the tree is computing the entropy and information gain for each attribute. In the classical version, to compute the entropy  $H(S)$  of a set  $S$ , the algorithm counts how many samples fall into each class [358]. A quantum algorithm can estimate these counts with potential quadratic speed-up via techniques like Quantum Amplitude Estimation (QAE) [57] (see Section 8.2). In this way, the entropy  $H(S_v)$  for each branch could be obtained with fewer queries to the data [359]. Once the entropies are known, the information gain can be computed similarly to Equation (77);

$$\text{InfoGain}(S, A) = H(S) - \sum_v p_v H(S_v), \quad (80)$$

where  $P_v$  is the projector selecting subset  $S_v$ ,  $p_v = \langle \Psi | P_v | \Psi \rangle$ , and  $H(S_v)$  is estimated via QAE [360].

To find the optimal attribute  $A^*$  (the feature with the highest information gain) from the set of available features  $F = \{A_1, A_2, \dots, A_d\}$ , one can use Quantum Maximum Search (QMxS) algorithms [360] (similar to the Quantum Minimum Search described in Section 18.5) [260] such that

$$A^* = \arg \max_{A \in F} \text{InfoGain}(S, A). \quad (81)$$

After finding the optimal attribute  $A^*$ , the algorithm must partition the data by measuring which samples belong to each branch. Once measured, we obtain classical subsets that must be re encoded into quantum states for processing the next level of the tree [318]. This creates a hybrid quantum-classical recursion, analogous to the classical algorithm's recursion, except that here at each node, one relies on quantum subroutines to decide how to split. In general, the algorithm will construct the same tree as a classical greedy algorithm would, with the difference that it can potentially speed-up the computation of the branch criterion [356,361]. Using the ideas above, Algorithm (13) outlines the QDT procedure.

---

#### Algorithm 13 Quantum Decision Tree

---

**Input:** Training dataset  $S = \{(\vec{x}_i, y_i)\}_{i=1}^M$ , feature set  $F$ , and optional depth limit  $d_{\max}$

**Initialisation:** Prepare quantum state  $|\Psi_{\text{data}}\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^M |i\rangle |x_i\rangle |y_i\rangle$  using QRAM

**Homogeneity Check:** (a) Estimate entropy  $H(S)$  (b) If  $H(S) = 0$ ,  $F = \emptyset$ , or current depth =  $d_{\max}$ : return a majority-class leaf

**Quantum Split Evaluation:** 1. For each feature  $A \in F$ : (a) Apply quantum predicate on  $A$  to split state  $|\Psi_{\text{data}}\rangle$  via ancilla (b) Use QAE to evaluate class probabilities in each split branch (c) Compute entropy  $H(S_v)$  and estimate information gain  $\text{IG}(S, A)$

**Optimal Split Selection:** Use Quantum Maximum Search [360] to find  $A^* = \arg \max_{A \in F} \text{IG}(S, A)$

**Quantum Data Partitioning:** (a) Measure ancilla qubit to collapse  $|\Psi_{\text{data}}\rangle$  into two subsets:  $S_{\text{left}}, S_{\text{right}}$  (b) Re-normalise quantum states for each subset

**Output:** Associated class label

---

In general, for a classical node evaluation with time complexity  $O(MN)$  [362,363], the quantum counterpart has a proposed complexity of  $O(\sqrt{MN})$ <sup>15</sup> [364]. For example, a quantum C5.0<sup>16</sup> algorithm [356] was proposed to run in time  $O(D\sqrt{MN})$  (where  $D$  is tree depth) as opposed to the classical  $O(DMN)$  [356]. A recent improved QDT algorithm [361] has even potentially more significant speed-up specifically for incremental learning scenario. By assuming that new data arrives in small batches, the algorithm can retrain a decision tree with logarithmic complexity in the number of samples  $O(Dk^{2D}N \text{poly log } M)$  (where  $k$  is the number of clusters per node) [361]. This is achieved by using a *quantum q-means* algorithm [366] to find split boundaries and updating the tree, maintaining accuracy comparable to classical trees while reducing its time complexity [361]. Such results may indicate that

<sup>15</sup> This is a rough estimate; a more accurate complexity also depend on the quantum subroutines used.

<sup>16</sup> The C5.0 algorithm is an improved family of recursive decision tree algorithms first developed by Ross Quinlan. The "C" stands for classifier [365].

QDT training can potentially outperform its classical counterpart for large and high-dimensional datasets.

Another source of potential advantage of QDT is when the input data itself is quantum (e.g. when quantum states need classification). In this case, one can avoid the bottleneck of reading out classical features, and directly work with quantum states. An example of this is the algorithm by [367], which treats the unknown quantum state as the input and performs adaptive measurements (based on information gain) to classify the state. A QDT can thus tackle classification of quantum data in its native form, which is a potential advantage for quantum sensing or readout tasks [367].

In finance, decision trees are used for portfolio management, risk assessment, and algorithmic trading (for example, deciding whether to buy/sell based on economic indicators can be framed as a decision tree) [368–371]. A QDT can potentially handle larger feature sets and data streams in real-time, which might be useful for high-frequency trading strategies or analysing large-scale financial data for patterns [41]. The interpretability of decision trees is a plus in finance, where regulators need explanations for decisions (such as in credit approval) [372,373]. Compliance teams also often use decision trees because the resulting rules can be explained to regulators [374].

Despite the promising advantages and application, QDTs face several limitations and challenges. These include the bottleneck of implementing QRAM [341], and the overheads by QAE and Quantum Maximum Search subroutines. Ultimately, the output of the training is a classical decision tree even in the case where the input is quantum data. This means the potential quantum speed-up is only in training of the decision tree and not in the final prediction.

In summary, QDTs promise potential polynomial speed-ups and memory advantages due to quantum subroutines like QAE. However, the quantum subroutines require advanced quantum hardware and are subject to some of the same information-theoretical limits as classical algorithms [18,375,376]. Hence, QDT are most likely to be beneficial in scenarios where data size is massive with dominating cost of processing it, or where the data is quantum mechanical in nature. As quantum technology improves, these disadvantages may be mitigated, but at present, QDTs require further research and development before achieving Practical Quantum Advantage in industrial applications such as finance and economics.

## 21.2. Quantum Algorithms for Regression

Regression in ML is used to solve the problem of fitting a function from the training data and identifying the relationship between independent variables and an outcome [308]. It is also used for predictive modelling to predict continuous outcomes by applying varying attributes in order to understand how the target value changes [377]. Many classifiers algorithms such as discussed in Section (21.1) can be adapted, with only minor modifications, for regression tasks. For example, a SVM classifier seeks a maximum-margin hyperplane for separating discrete classes, whereas its regression counterpart adjust the objective to fit continuous targets [136]. Similarly, while a KNN classifier predicts labels by a majority of vote among nearest neighbours, the regression version of KNN outputs a continuous value computed as the average of the target values of the  $k$  nearest neighbours [378]. Decision tree classifiers use impurity measures to splits and assign a majority class at each leaf, whereas a regression tree uses variance-based criteria (such as minimising measures like mean squared error) to output a numeric prediction [377]. The same adjustments carry over to the quantum versions of those models.

A simple and fundamental case is linear regression [379]: given  $M$  data points  $(\vec{x}_i, y_i)$  with  $\vec{x}_i \in \mathbb{R}^N$  and  $y_i \in \mathbb{R}$ , we assume  $y_i \approx \vec{x}_i^\top \beta$  for some weight vector  $\beta \in \mathbb{R}^N$ . One wants to find  $\beta^*$  that minimises the sum of squared errors [380] such that

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^N} \frac{1}{2} \sum_{i=1}^M (\vec{x}_i^\top \beta - y_i)^2 \quad (82)$$

The solution to Equation (82) is given by the *normal equation* [379]

$$X^T X \beta = X^T y. \quad (83)$$

This is a  $N \times N$  linear system of equations. When  $X^T X$  is invertible (i.e.  $X$  has full column rank), the solution for the optimal coefficients is given in closed form by the Ordinary Least Squares (OLS) estimator:

$$\hat{\beta} = (X^T X)^{-1} X^T y, \quad (84)$$

where each component of  $\hat{\beta}$  represents the linear effect of the corresponding feature on the outcome, holding other features fixed [381]. This solution assumes an unregularised least-squares problem. In many real-world scenarios, however,  $X^T X$  may be ill-conditioned or singular (e.g. when  $N > M$  or features are highly correlated), which can lead to overfitting or unstable solutions [308]. Ridge regression (also called  $L_2$ -regularised linear regression) addresses this by adding a penalty on the size of coefficients [382]. The ridge estimator solves [382]:

$$\min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2, \quad (85)$$

where  $\lambda > 0$  is a regularisation parameter. Therefore, normal equation then becomes

$$(X^T X + \lambda I) \beta = X^T y, \quad (86)$$

yielding the closed-form solution

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y. \quad (87)$$

This shrinks the regression coefficients toward zero to improve generalisation (when  $\lambda = 0$  we recover OLS, and as  $\lambda \rightarrow \infty$  we force  $\beta \rightarrow 0$ ) [383].

Solving the normal equations by direct classical matrix inversion methods is  $\mathcal{O}(N^3)$  in general, or  $\mathcal{O}(MN^2)$  if forming  $X^T X$  via  $M$  data points. There exist more scalable algorithms for large  $M$ , such as iterative solvers or stochastic gradient descent [384], and the state-of-the-art classical solvers can run in time  $\mathcal{O}(MN + \text{poly}(N, 1/\epsilon))$  [385]. Nonetheless, for very high-dimensional data or massive sample sizes, these computations become expensive for classical methods. This has motivated research into quantum algorithms that could potentially speed-up linear regression by exploiting quantum linear algebra subroutines. In the following subsection, we will introduce some quantum algorithms for this task.

### 21.2.1. Quantum Kernel Ridge Regression

One of the earliest and most famous quantum algorithms relevant to regression is the HHL algorithm [6,302] for solving systems of linear equations. Essentially, HHL can be applied to solve the normal equations of linear regression. The HHL or improved QLSS algorithm [131,132] prepares quantum circuit encoding the data, solves the normal equation via efficient quantum matrix inversion protocol, and output a quantum state proportional to the normalized  $|\beta\rangle$  solution vector<sup>17</sup>.

Subsequent research has explored variations like solving weighted least squares and introducing regularisation directly into the QLSS. For instance, quantum ridge regression can be performed by solving Equation (86) with a QLSS such the quantum algorithm by Kerenidis et al. [134] that used this for recommendation systems with a low-rank  $X$  (thus  $\kappa$  small) [385]. More recently, [386] developed Quantum Regularised Least Squares (QRLS) algorithms using QSVT, achieving polynomial improvements in  $\kappa$  and an exponential improvement in precision for ridge regression compared to prior HHL-based methods [386]

<sup>17</sup> For more information see Section (11)

While HHL tackles regression by directly solving the associated linear normal equation, an alternative quantum approach called Quantum Kernel Ridge Regression (QKRR) makes use of kernel methods to perform regression [387]. The QKRR combines the power of kernel methods (to capture non-linear relationships) with quantum computing to efficiently compute or approximate kernel functions in high-dimensional feature spaces. In classical kernel ridge regression [382,388], one maps input data points  $\vec{x}$  onto a high-dimensional feature space via a feature map  $\phi(\vec{x})$ , and perform linear regression in that feature space with an  $L_2$  regularisation. By the kernel trick, one avoids working explicitly with the high-dimensional  $\phi(\vec{x})$ , but instead the solution is obtained in terms of kernel inner products  $K(\vec{x}_i, \vec{x}_j) = \langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle$ . If  $K$  is the  $M \times M$  kernel Gram matrix on the training data and  $y$  is the training target vector, the ridge regression [389] solution in dual form is

$$\vec{\alpha} = (K + \lambda I)^{-1} y, \quad (88)$$

where  $\vec{\alpha}$  are the Lagrange multiplier as defined in Section (21.1.1). The predicted value for a new input  $x_*$  is given by [390]:

$$\hat{y}_* = \sum_{i=1}^n \alpha_i K(x_i, x_*). \quad (89)$$

This procedure can fit very complex functions depending on the choice of kernel, but the downside is that forming and inverting the  $K$  matrix costs  $\mathcal{O}(M^3)$  classically [391]. So, for large datasets, kernel methods become computationally expensive for classical methods.

The quantum version of this algorithm seeks to speed-up kernel evaluation and linear system solving for kernel ridge regression. The QKRR first defines a unitary  $U(x)$  that prepares the state  $|\phi(x)\rangle = U(x) |0 \cdots 0\rangle$  as an encoding of  $\vec{x}$  the classical input data-point<sup>18</sup>. Common choices for implementing  $U(x)$  include basis encoding, angle encoding, and more expressive feature maps [339,392]. The mapping should be data-efficient and ideally enhance linear separability in feature space [339]. QKRR then, estimate the kernel matrix  $K$  for all training points. In principle, one could do this by preparing  $|\phi(\vec{x}_i)\rangle$  and  $|\phi(\vec{x}_j)\rangle$ , then performing a SWAP-test and repeating to get an estimate of the inner product. With assumptions like QRAM, quantum algorithms can access kernel matrix elements or prepare quantum states encoding kernel information, with complexity that can be sublinear in the feature dimension [134,302,393].

The output of the QPU is an estimate for  $K$  which is inserted to Equation (88) to solve for  $\vec{\alpha}$ . One approach is to use a QLSS to solve Equation (88) which outputs a state proportional to the normalized solution  $|\alpha\rangle$ . However, this would require the ability to efficiently perform the linear operation of  $K + \lambda I$  on a QPU. If  $K$  is low-rank or structured, block-encoding techniques [138] can achieve this, and if  $K$  is unitary the LCU [48] methods can be employed.

Finally, given the solution coefficients  $\vec{\alpha}$  as output from QPU, to predict on a new input  $x_*$ , one could compute  $\hat{y}_*$  using Equation (89). In practice, many hybrid approaches just use the QPU for estimating the kernel  $K$  and then solve the ridge regression Equation (88) on a classical computer if  $M$  is not too large. The quantum-classical computational pipeline described above can be summarised in Algorithm (14). In many implementations, Step 1 (quantum kernel computation) is the executed on QPU, and Steps 2-3 are executed on CPUs. Fully quantum implementations that also solve Equation (88) might use an HHL-like QLSS in Step 2, but require  $K$  to be 'well-conditioned' and accessible as an oracle.

<sup>18</sup> For instance,  $\vec{x}$  can be a historical price series or volatility features data-point.

**Algorithm 14** Quantum Kernel Ridge Regression

**Input:** Training data  $\vec{x}_i, y_{i=1..M}$ , feature-map circuit  $U(x)$  preparing  $|\phi(\vec{x})\rangle$ , regularisation  $\lambda$ .

**Quantum kernel computation:** Use a PQC to estimate  $K_{ij} = \langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle$ .

**Solve ridge regression:** Solve  $(K + \lambda I)\vec{\alpha} = \vec{y}$  for  $\vec{\alpha}$  (using classical or QLSS to obtain  $|\alpha\rangle$ ).

**Prediction for new input  $x_*$ :** (a) Prepare state  $|\phi(x_*)\rangle$  (b) compute  $K(x_i, x_*)$

**Output:** Regression model coefficients  $\alpha$  (classical vector).

The primary advantage of QKRR is the ability to handle extremely high-dimensional feature spaces implicitly via quantum kernel methods which have potential speed-ups [387,394]. However, due to the current state of quantum hardware, QKRR experiments have been limited to small data sets, typically demonstrated on systems with modest qubit counts (e.g., [391,395–397]). One concern is that in practice, many proposals end up with complexity  $\tilde{O}(M)$  or worse when all costs are accounted for, meaning a quantum computer might realise Practical Quantum Advantage except in special cases [11,146,398]. Another concern is that quantum kernel methods can sometimes suffer from concentration issues (be nearly uniform in high dimension), making them less useful than hoped for [399].

Despite these challenges, early demonstrations of QKRR on small systems have shown that quantum computed kernels can perform as well as classical kernels on simple tasks. For example, experiments with up to 10 qubits have used a PQC kernel to do regression on molecular data, achieving accuracy on par with classical Gaussian kernels [400].

Future research may include optimising quantum feature maps, understanding generalisation properties of quantum models, and integrating quantum kernel methods with classical frameworks. Whether QKRR can offer a decisive speed-up for practical problems remains an open challenge.

### 21.3. Quantum Neural Networks

Classical neural networks are function approximators composed of layers of interconnected nodes, inspired by biological neuron systems [401,402]. Each node performs an affine transformation on its inputs, followed by a non-linear activation. In a feed-forward network, the  $l$ -th layer takes an input vector  $\mathbf{a}^{(l)}$ , and produces an output  $\mathbf{a}^{(l+1)}$  via:

$$\begin{aligned} \mathbf{z}^{(l+1)} &= \mathbf{W}^{(l)} \mathbf{a}^{(l)} + \mathbf{b}^{(l)} \\ \mathbf{a}^{(l+1)} &= \phi(\mathbf{z}^{(l+1)}), \end{aligned} \tag{90}$$

where  $W^{(l)}$  is a weight matrix,  $\mathbf{b}^{(l)}$  a bias vector (the affine transformation), and  $\phi(\cdot)$  an activation function. The activation function introduces non-linearity; without it, multiple layers would collapse into an equivalent single affine layer. For example, a simple one-layer perceptron<sup>19</sup> with input  $\mathbf{x} \in \mathbb{R}^n$  and output  $y \in \mathbb{R}$  can be written as  $y = \phi(\mathbf{w}^T \mathbf{x} + b)$ , with trainable parameters  $\mathbf{w}$  and  $b$ . Networks are trained by defining a loss function  $L$  and adjusting the weights  $W^{(l)}$ ,  $b^{(l)}$  to minimise  $L$ , typically using gradient-based optimisation [403]. Deep neural networks (many-layered) have achieved a huge success in prediction tasks by learning complex hierarchical features from data [404].

A Quantum Neural Network (QNN) [405] extends the classical neural network concept by using a PQC, where the QNN ‘layers’ can be seen as sequential blocks of quantum gates, some of which are parametrised by adjustable angles analogous to weights. Like input layers in classical networks, QNNs begin by encoding classical data  $\mathbf{x}$  into a quantum state [41]. The choice of encoding influences the QNN’s expressive power<sup>20</sup> [406]. Encodings can be repeated throughout the circuit (data re-uploading) to form multiple ‘layers’, analogous to having multiple layers in a classical network [407].

<sup>19</sup> A perceptron is a mechanism that activates a neuron due to the input of other neurons [402].

<sup>20</sup> Expressive power refers to the model’s capacity to represent complex functions. Different encoding strategies enable QNNs to access different regions of the quantum state space, affecting what functions can be learned.

After encoding, the QNN applies a sequence of parametrised quantum gates  $U(\theta)$  to process the state. These gates depend on a set of  $m$  trainable parameters  $\theta = (\theta_1, \theta_2, \dots, \theta_m)$ . A typical quantum layer might consist of single-qubit rotations on each qubit followed by multi-qubit entangling gates. For example, a layer on qubit  $j$  could include a rotation:

$$R_y(\theta) = \exp(-i\theta_j Y/2) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}. \quad (91)$$

These gates are analogous to weight multiplications in a classical network, with the angle  $\theta_j$  as the tunable weight. In many QNN designs, one layer consists of rotations on all qubits (with independent parameters) followed by a fixed pattern of entangling gates across qubits. The combination of encoding and  $L$  layers of trainable gates produces a final quantum state:

$$|\Psi(\mathbf{x}; \theta)\rangle = U(\theta)U_{\text{enc}}(\mathbf{x})|0\rangle^{\otimes n}, \quad (92)$$

where  $U(\theta)$  represents the total unitary composed of all parametrised gates. This state  $|\Psi(\mathbf{x}; \theta)\rangle$  encodes the model's prediction in its quantum amplitudes. To extract a classical output from the quantum state (analogous to the classical output neuron activations), one measures an observable  $\hat{O}$  and takes the expectation value as the model output. Then, the predicted scalar output can be written as:

$$f(\mathbf{x}; \theta) = \langle \Psi(\mathbf{x}; \theta) | \hat{O} | \Psi(\mathbf{x}; \theta) \rangle. \quad (93)$$

Therefore, for a binary classification, the probability of class label '0' is given by

$$\Pr(y = 0 | \mathbf{x}) = \frac{1 + f(\mathbf{x}; \theta)}{2}. \quad (94)$$

Note that the PQC,  $U(\theta)$ , is linear, whereas a classical neural layer is non-linear due to  $\phi$ . The non-linearity in QNNs enters when one measure and feed results into a classical optimiser; as the measurement probabilities have a non-linear dependence on the quantum state amplitudes [408]. Additionally, if the QNN is a hybrid network, or if intermediate measurements are performed between layers, effective non-linear transformations can be achieved [409]. In most near-term QNN implementations, a variational approach is often employed, relying on classical iterative optimisation to adjust  $\theta$  (see VQA in Section 12).

The training of a QNN involves finding the optimal parameters  $\theta$  such that the QNN's output matches the target outputs on a training dataset (same as in classical ML). This is a hybrid quantum-classical approach; the quantum circuit generates estimates of target outputs based on  $\theta_i$  and a classical optimiser updates the parameters  $\theta_i \mapsto \theta_{i+1}$  to improve the subsequent estimate. Generally the classical optimiser is guided by a loss function  $\mathcal{L}(\theta)$ , for example the mean squared error or cross-entropy between QNN predictions  $f(x_i; \theta)$  and labels  $y_i$  over the dataset [407]. Hence, the parameter update can be done via gradient descent depending on estimates of  $\nabla_{\theta} \mathcal{L}$ . The gradients can be estimated on the QPU using the parameter-shift rule, an analytical quantum subroutine, or via automatic differentiation on quantum simulators [410]. The parameter-shift rule for a rotation parameter  $\theta$  in a gate  $R(\theta) = e^{-i\theta G/2}$  (where  $G$  is a Pauli operator) is [410]:

$$\frac{\partial}{\partial \theta} f(\mathbf{x}; \theta) = \frac{1}{2} (f(\mathbf{x}; \theta + \frac{\pi}{2}) - f(\mathbf{x}; \theta - \frac{\pi}{2})). \quad (95)$$

This implies we can obtain the gradient by evaluating the circuit at two shifted parameter values. This requires additional quantum evaluations for each parameter, which is a key contributor to training cost. The complete QNN training procedure is summarised in Algorithm (15).

**Algorithm 15** Training Quantum Neural Network

**Input:** Training samples  $\{(x_1, y_1), \dots, (x_S, y_S)\}$ , parameters  $\theta = \{\theta_1, \dots, \theta_M\}$ , total iterations  $K$ , and error-tolerance  $\epsilon$ .

**Initialising:** Initialise parameters  $\theta^{(0)} \sim \mathcal{N}(0, \sigma^2)$ , generally randomly chosen from a normal distribution.

**For  $k = 1$  to  $K$  do: For each training sample  $s \in \{1, \dots, S\}$ :** (a) Prepare quantum state  $U_{\text{enc}}(x_s) |0\rangle^{\otimes n}$ . (b) Apply PQC  $U(\theta^{(k)})$ . (c) Measure observable  $\hat{O}$  to obtain prediction  $f_s = \langle \hat{O} \rangle$ . (d) Compute sample loss  $\ell_s = \ell(f_s, y_s)$  (loss for sample  $s$ ).

**Compute global loss:**  $\mathcal{L}(\theta^{(k)}) = \frac{1}{S} \sum_{s=1}^S \ell_s$ .

**Estimate gradients** for each  $\theta_j \in \theta^{(k)}$ : estimate  $\partial \mathcal{L}(\theta^{(k)}) / \partial \theta_j = \frac{1}{S} \sum_{s=1}^S \frac{\partial \ell_s}{\partial \theta_j}$

**Update parameters:**  $\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_{\theta} \mathcal{L}(\theta^{(k)})$  where  $\eta > 0$  is the learning-rate.

**Terminate:** if  $|\mathcal{L}(\theta^{(k+1)}) - \mathcal{L}(\theta^{(k)})| < \epsilon$ . **End for**

**Output:** Set of parameters  $\theta^*$  that defines a QNN for classification task.

QNNs are currently being explored theoretically and experimentally for various financial applications. For example, in financial time-series forecasting, hybrid QNNs have shown potential to capture complex temporal dependencies and outperform classical models in stock market predictions [411,412]. For fraud detection, QNN could learn subtle anomalies in large transactional data [413] by using quantum graph neural networks to flag fraudulent patterns [414]. QNNs have also been suggested for option pricing and risk modelling, where they could learn and approximate stochastic processes or option payoff scenarios that are challenging for classical networks. These applications are still in the early stages, with ongoing research assessing whether QNNs can provide better accuracy or speed compared to classical ML [415].

In theory, if each layer applies  $O(n)$  single-qubit rotations and  $O(n)$  two-qubit gates, the circuit depth might scale as  $O(nL)$  for  $n$  qubits and  $L$  layers. This is analogous to the  $O(N_{\text{params}})$  cost of evaluating a classical neural network with  $N_{\text{params}}$  weights. However, QNNs can operate in an exponentially large space of dimension  $2^n$ , which is where the potential advantage lies. For instance, recent results [416] proved a universal approximation theorem for QNNs and derived that an  $O(\lceil \log_2(1/\epsilon) \rceil)$  number of qubits with  $O(\epsilon^{-2})$  circuit parameters can approximate functions to error  $\epsilon$ . This suggests that to achieve high accuracy, the parameter count in a QNN might grow polynomially in  $1/\epsilon$  while the qubit count grows only logarithmically. In contrast, a classical network might require significantly more neurons to achieve the same expressive power. These are worst-case upper bounds: the actual runtime advantage of QNNs remains an open question and is likely problem-dependent. Whether they can surpass classical neural networks for real-world tasks is an active area of research [417].

A major challenge for QNNs is the barren plateau [153] phenomenon<sup>21</sup>. As the number of qubits or circuit depth grows, the gradients for the cost-function  $\nabla_{\theta_i} C(\theta)$  can vanish exponentially, making training infeasible [153], i.e., the training landscape is almost flat in most areas. Architectures that are designed to avoid barren plateaus [418] may result to quantum circuits that can be simulated classically hence lose quantum advantage [156]. This issue amongst others is a serious limitation on scaling QNN to perform large-scale classification tasks in industrial setting [154,415].

## 22. Quantum Algorithms for Unsupervised ML

Building upon the foundational concepts of QML discussed in Section (21) for supervised learning, in this section we review quantum algorithms for unsupervised ML. Unlike supervised learning, which relies on given labelled data, unsupervised methods aim to discover hidden patterns and structures within unlabelled data by their own [403]. This attribute allows complex data analysis without requiring additional information or prior knowledge of the data, making unsupervised learning aptly tailored for real-world applications, such as in finance [42].

<sup>21</sup> Also discussed in Sections (12) and (18)

To illustrate more explicitly how the unsupervised learning process works, let us revisit the animal data set example from the previous Section (21). In supervised learning, each image comes with a label indicating the type of animal—whether it is a cat, dog, or bird. The algorithm uses this label to learn a mapping function that can correctly predict new, unseen data. In unsupervised learning, however, the collection of images lacks any labels, and the algorithm might analyse intrinsic features of the images, such as colour, texture, shape, or size, to find patterns within the data. For instance, it might group cat images due to shared characteristics like pointy ears and whiskers, even though it does not know the term ‘cat’. In other words, unsupervised learning aims to find a function or model that captures the patterns and structure of the data without any labels guiding the process [39].

Common ML techniques include clustering, where data points are grouped based on similarity and dimensionality reduction, which involves projecting data onto a lower-dimensional space while preserving significant features (such as Principal Components Analysis (PCA) and autoencoders) [419]. Generative models, which can learn the underlying data distribution to generate new data points that resemble the training dataset. Additionally, Reinforcement Learning (RL) techniques, while traditionally categorised separately, can incorporate unsupervised elements for exploration and policy improvement without explicit labels. Extending these techniques in the context of quantum computing is particularly helpful, as they can potentially offer polynomial to exponential speed-ups over classical algorithms [420]. In the following subsections, we will review examples of these techniques and highlight how they are being adapted and implemented in quantum contexts.

### 22.1. Quantum Clustering

Clustering methods rely on mathematical formulations, such as distance metrics (e.g. Euclidean) or statistical models, to evaluate the relationships between data points. These formulations help identify patterns and group data points into distinct clusters by measuring how similar or different they are [39]. Quantum algorithms for clustering have been developed to improve the efficiency of classical methods [421]. Proposed quantum clustering techniques can be applied in areas like fraud detection and customer segmentation (e.g., credit scoring) [421,422]. However, practical implementations on large scale industrial problems remain constrained by current quantum hardware and other algorithmic bottlenecks. In this subsection, we will review some of the common quantum clustering algorithms.

#### 22.1.1. Quantum k-Means Clustering

One of the most well-known algorithms for unsupervised learning is the  $k$ -means clustering algorithm, first introduced by Lloyd [423]. It takes as input a set of vectors  $X = \{x_1, x_2, \dots, x_M\} \subseteq \mathbb{R}^N$ , where  $x_i \in \mathbb{R}^N$  for  $i \in \{1, \dots, M\}$ , with  $M$  being the number of data points and  $N$  the dimensionality (number of features). The aim is to place these vectors or data points in  $k$  subsets so that it minimises sum objective function  $f_k$ . This function is defined as the sum over all points of the squared Euclidean distance from a cluster centroid (subsets) to the members of the cluster [296]

$$f_k = \min_{C_1, \dots, C_k; \mu_1, \dots, \mu_k} \sum_{j=1}^k \sum_{i \in C_j} \|x_i - \mu_j\|^2, \quad (96)$$

where  $C = \{C_1, C_2, \dots, C_k\}$  are the clusters to be determined, and  $\mu_j$  is the centroid of cluster  $C_j$  for  $j \in \{1, \dots, k\}$ . The algorithm initialised  $k$  different centroids (often using methods like k-Means++<sup>22</sup> for better convergence) and iterates over two main steps until convergence. The first step involves assigning each data point  $x_i$  to the cluster with the nearest centroid, where the cluster assignment  $C(x_i)$  is determined by [296]

$$C(x_i) = \arg \min_j \|x_i - \mu_j\|^2. \quad (97)$$

<sup>22</sup> k-Means++ is an initialisation method that selects initial centers with probability proportional to their squared distance from existing centers, improving convergence [424].

Second step is to recompute and update the centroids as the mean of their respective clusters [425]

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i. \quad (98)$$

These steps are repeated until changes in the cluster assignments between iterations are sufficiently small [366]. In other words,  $k$ -means assigns each data point to the cluster whose centroid it is closest to [289].

The classical K-means clustering is a widely used clustering technique [426]; however, despite its simplicity, it suffers from several limitations. These includes issues with the initialisation of centroids which often leads to suboptimal clustering and the algorithm's complexity scaling with number of data points and clusters. Improvements to the initialisation and optimisation stages of  $k$ -means have been proposed such [427]. These aim to mitigate issues such as convergence to local minima and sensitivity to initial centroids, and better analysis of large-scale distributed data [428]. However, in general the  $k$ -means clustering is a an NP-Hard problem [429], hence quantum algorithms are a promising alternative to improve efficiency. Researchers have proposed multiple variants of Quantum k-Means Clustering (QkMC) [366] with potential exponential speed-ups [299,426,430]. By encoding the data into quantum states, these quantum algorithms accelerates the computationally intensive steps of the classical algorithm, typically the distance calculations [289,426].

In this review, we will introduce one variant of the QkMC based on [426]. This a hybrid quantum-classical algorithm where the QPU handles the distance computation while the update step is done on a CPU. Similar to QML methods in Section (21) the classical data can be encoded into quantum states using amplitude encoding<sup>23</sup>. A sequence of controlled unitaries  $U$  prepares the state

$$U |0\rangle_a |0\rangle^{\otimes n} = \frac{1}{\sqrt{2N}} (|0\rangle_a |x\rangle + |1\rangle_a |\mu\rangle). \quad (99)$$

Here, the qubits with subscript ' $a$ ' are the ancillas and the amplitude-encoded states  $|x\rangle$  and  $|\mu\rangle$  are given by

$$|x\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^N x_j |j\rangle, \quad \text{and} \quad |\mu\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^N \mu_j |j\rangle, \quad (100)$$

where  $x_j, \mu_j$  are the  $j$ -th feature components of data point  $x$  and centroid  $\mu$ , respectively. By employing the SWAP test [320] which produces  $\langle x|\mu\rangle$ , the QPU efficiently estimates the Euclidean distance  $d(x, \mu)$  between  $|x\rangle$  and  $|\mu\rangle$  given by Equation (74).

After computing the distances between the data point and all centroids, this is repeated for all  $M \times k$  pairs of vectors, where  $M$  is the number of data points and  $k$  is the number of centroids. The algorithm assigns the data point to the nearest centroid as per Equation (97).

---

#### Algorithm 16 Quantum k-Means Clustering

---

**Input:** Dataset  $X = \{x_1, x_2, \dots, x_M\}$ , and number of clusters  $k$

**Initialising:** Select initial centroids  $\mu = \{\mu_1, \mu_2, \dots, \mu_k\}$  using  $k$ -Means++

**Assignment:** For each data point  $x_i \in X$  and each centroid  $\mu_j \in \mu$ , compute distance  $d(x_i, \mu_j)$  using QPU: (a) Encode  $x_i$  and  $\mu_j$  as quantum states (Equation (100)). (b) Estimate distance using SWAP test [320].. (c) Assign  $x_i$  to the nearest cluster  $C_j$ .

**Update:** For the assigned clusters, update the estimate for the centroids given by Equation (98)

**Repeat** steps 3 and 4 until convergence

**Output:** Cluster assignments and centroids  $\mu$

---

<sup>23</sup> Some proposals to accomplish this by using Flip-Flop QRAM (FF-QRAM) [431] seemed promising but could be impractical for fault-tolerant [341].

Examples of quantum k-means implementations include hybrid methods applied on simulators and full implementations on quantum platforms [432]. In finance, a quantum k-means pipeline is applied to cross-border payment risk supervision [433], enhancing clustering of blockchain transaction data [434]. Recent research of quantum computing for finance recognise clustering methods as one of the unsupervised learning methods applicable to finance [41,346]. Although detailed use-cases are still in early exploratory stages.

Examples of quantum k-means implementations include hybrid methods applied on simulators and full implementations on quantum platforms [432]. In finance, a quantum k-means pipeline is applied to cross-border payment risk supervision [433], enhancing clustering of blockchain transaction data [434]. Recent research of quantum computing for finance recognise clustering methods as one of the supervised learning methods applicable to finance [41,346]. Although detailed use-cases are still in early exploratory stages.

While quantum k-means addresses clustering through iterative centroid updates, quantum spectral clustering takes a fundamentally different approach by leveraging graph theory and eigendecomposition. Classical spectral clustering constructs a similarity graph from the data, computes the Laplacian matrix, and projects data into a lower-dimensional space defined by the smallest  $k$  eigenvectors, where traditional clustering becomes more effective—particularly for non-convex or nested structures [435,436]. The classical algorithm  $\mathcal{O}(M^3)$  complexity for eigendecomposition severely limits scalability [437]. Quantum spectral clustering, as proposed by Kerenidis et al. [438], attempts to accelerate this process using QPE to compute eigenvalues of the normalised Laplacian. However, this approach also requires strong assumptions, such as efficient QRAM [288] and block-encoding, and successful post-selection of the desired eigenvectors. The claimed quantum advantage remains theoretical, as the practical implementation faces significant challenges including QRAM overhead (which alone could eliminate any speedup), the complexity of preparing the block-encoded Laplacian, and the probabilistic nature of post-selection which may require multiple runs. To date, only small proof of concept demonstrations on actual quantum hardware have been reported [439], and there is no large-scale implementations showing advantage. Recent dequantisation results suggest that classical algorithms with similar assumptions might achieve comparable performance, raising questions about whether genuine quantum advantage exists for this application.

## 22.2. Quantum Generative Models

Unsupervised generative models are extensively used in deep learning in order to generate synthetic data and new samples [30]. Classical generative models (e.g., Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs)) aim to learn a target probability distribution  $\tilde{P}(x)$  over a set of possible outcomes  $x \in \Omega$ . This is achieved by constructing a model distribution  $P(x; \theta)$  that approximates  $\tilde{P}(x)$ . Despite their effectiveness, classical generative models require significant computational resources and can struggle to model complex dependencies, especially in high-dimensional spaces [440].

Quantum generative models have been proposed as alternatives to potentially address these limitations [398,441]. For instance: (i) By enabling parallel sampling from different parts of the distribution, which reduces the time required for generating new samples. (ii) Using entanglement to capture complex correlations between variables more naturally, improving the expressiveness of the model. These improvements suggest that quantum generative models could benefit tasks requiring high-dimensional data synthesis, such as generating synthetic financial data for stress testing and model validation or improving portfolio optimisation by generating diverse asset combinations [30].

The training of quantum generative models focuses on minimising a divergence or distance measure, such as the Kullback–Leibler (KL) divergence [442]. Alternatively, one may define a cost function and use maximum likelihood estimation [398] or quantum fidelity-based costs [443] to find its minimum. Below, we will highlight two major techniques: Quantum Boltzmann Machine (QBM) [444] and Quantum Generative Adversarial Networks (QGAN) [445].

### 22.2.1. Quantum Boltzmann Machine

In classical ML, a Boltzmann Machine (BM) [446] is an energy-based model that defines a probability distribution by assigning to each configuration  $x \in \{0, 1\}^d$  a probability [447]. This distribution is given by the Boltzmann distribution [403]:

$$P(x) = \frac{e^{-E(x)}}{Z}, \quad \text{where } Z = \sum_x e^{-E(x)}, \quad (101)$$

with  $E(x)$  denoting the energy of configuration  $x$ , and  $Z$  the partition function ensuring normalisation. For the standard BM, the energy function consists of pairwise interactions among units and bias terms, and can be written as [403]:

$$E(x) = -x^\top Wx - b^\top x, \quad (102)$$

where  $W$  is a weight matrix and  $b$  is a bias vector.

BMs evolved from earlier energy-based models, particularly Hopfield networks<sup>24</sup> [448], which use a similar energy framework but with deterministic dynamics. While Hopfield networks minimise energy through deterministic updates and can get trapped in local minima, BMs introduce stochasticity through the Boltzmann distribution in Equation (101) [449,450], allowing them to escape local minima and model probability distributions more effectively [403,451].

Training a BM is usually performed via Contrastive Divergence (CD) [452] or maximum likelihood estimation [453], which involves adjusting the weights, and is equivalent to minimising the Kullback–Leibler divergence [442] between the data distribution and the model distribution. This is typically achieved through the log-likelihood gradient, which decomposes the learning into two terms: a positive phase that increases the probability of data configurations and a negative phase that decreases the probability of configurations sampled from the model (unobserved configurations). This can be written as [454]

$$\nabla_{W_{ij}} \ell(\theta) = \langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}, \quad (103)$$

$$\nabla_{b_i} \ell(\theta) = \langle x_i \rangle_{\text{data}} - \langle x_i \rangle_{\text{model}}, \quad (104)$$

where  $\langle \cdot \rangle_{\text{data}}$  denotes an expectation under the empirical distribution and  $\langle \cdot \rangle_{\text{model}}$  an expectation under the model distribution [403,447]. These update rules resemble Hebbian learning [455], adjusting parameters so that model statistics align with data statistics.

Through this stochastic formulation, BMs serve as generative models, since they define full probability distributions over the visible variables, and can support probabilistic inference by estimating conditional distributions. For instance, for BMs with visible and hidden units, inference includes computing  $P(h | v)$  or  $P(v | h)$  [403], enabling tasks such as feature extraction or generating new samples from the learned distribution. However, the computational cost of inference and learning of BMs face several challenges, such as: (i) evaluating the partition function  $Z$  requires summation over  $2^d$  states (configurations), which is intractable for large  $d$ . (ii) approximate Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling mix slowly in practice. (iii) The CD, for a fully connected BM with  $n$  units, each Gibbs sampling requires  $O(n^2)$  operations, making training computationally expensive. These limitations have motivated both architectural simplifications (e.g., Restricted Boltzmann Machines) and quantum-inspired approaches to potentially overcome classical sampling bottlenecks [456].

<sup>24</sup> A Hopfield network is an early energy-based model that implements associative memory by storing patterns as local minima of an energy function [448]. Boltzmann Machines extend this framework by introducing stochasticity, enabling generative modelling [447].

In a Quantum Boltzmann Machine (QBM), the classical energy function is replaced by a Hamiltonian  $\hat{H}(w)$ , and the distribution is replaced by the density matrix of a thermal (Gibbs) state [444]:

$$\rho(w) = \frac{e^{-\beta\hat{H}(w)}}{Z(w)}, \quad \text{where } Z(w) = \text{Tr}[e^{-\beta\hat{H}(w)}], \quad (105)$$

and  $\beta = \frac{1}{k_B T}$  is the inverse of Boltzmann constant  $k_B$  times temperature  $T$ . As in the classical case, the learning involves adjusting the weights  $w$  so that expectation values of observables (e.g., marginal probabilities or correlation functions) under  $\rho(w)$  match data statistics with model statistics. Methods include gradient-based optimisers that compute the derivative of the log-partition function and of observable expectation values, such as:

$$\frac{\partial}{\partial w} \log \text{Tr}[e^{-\beta\hat{H}(w)}] \quad \text{and} \quad \frac{\partial}{\partial w} \text{Tr}[\hat{O}\rho(w)]. \quad (106)$$

To perform a thermal state approximation, one approach implements  $e^{-\beta\hat{H}(w)}$  via Trotterisation or variational methods [457]. The estimates of observable averages are compared with target data statistics (e.g., means, correlations) which is used to compute a cost function. Based on the gradients of the cost-function, the parameters  $w$  are updated until convergence. Algorithm (17) illustrates a typical QBM training loop using a variational approach.

---

#### Algorithm 17 Quantum Boltzmann Machine

---

**Input:** Target data statistics (means, correlations), temperature parameter, number of qubits

**Initialisation:** Define an ansatz for  $\hat{H}(w)$  (e.g., local fields and pairwise interactions) and pick random initial  $w$

**Thermal State Approximation:** (a) Construct a PQC  $V(\alpha)$ , with randomly initialised  $\alpha$ , to approximate  $e^{-\beta\hat{H}(w)}$ .

**Sampling and Observables:** (a) Run the circuit  $V(\alpha)$  on a QPU and measure in a relevant basis (e.g.,  $\sigma_z$ ) (b) Estimate observables, such as  $\langle \sigma_z^{(i)} \rangle$  or correlation  $\langle \sigma_z^{(i)} \sigma_z^{(j)} \rangle$

**Loss Computation:** Define a loss function comparing the measured observables to the target data (e.g., least squares)

**Optimisation:** (a) Update  $\alpha$  to improve the approximation of the thermal state. (b) Optionally, update  $w$  to better match target statistics by adjusting the Hamiltonian terms.

**Repeat** until convergence or maximum iterations are reached.

**Output:** Hamiltonian parameters  $w^*$  that define  $\hat{H}(w^*)$

---

QBMs can potentially explore difficult energy landscapes more efficiently [288] through quantum effects like tunnelling and may use fewer parameters to represent certain distributions [443]. However, preparing exact thermal states remains challenging on near-term NISQ devices.

Despite their theoretical advantages, quantum generative models still have several open problems. For example, noise and decoherence limit the circuit depth on current hardware, which in turn constrains model expressivity [444]. Furthermore, the training process can face issues due to barren plateaus, especially in high-dimensional parameter spaces [454]. Researchers have achieved some promising success in mitigating these issues by employing strategies such as carefully designing ansätze, implementing layer-wise training, and using advanced initialisation methods [458]. With hardware steadily improving and novel algorithms emerging [457], quantum generative models stand as a potentially useful tool. Demonstrating a clear quantum advantage in real-world applications remains an open challenge, but progress in error mitigation, algorithm design, and hybrid frameworks is bringing these models closer to practical use.

#### 22.2.2. Quantum Generative Adversarial Networks

Generative Adversarial Networks (GANs) [403] are implicit generative models that learn to generate samples from a target distribution by solving an adversarial minimax game between a

generator  $G$  and a discriminator  $D$ . The generator maps random noise  $z \sim p_z$  to samples  $G(z)$ , while the discriminator attempts to distinguish between real data  $x \sim p_{\text{data}}$  and generated samples. The objective is given by

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (107)$$

At equilibrium under the assumptions of infinite model capacity and an optimal discriminator the generator distribution converges to the data distribution, i.e.,  $p_g = p_{\text{data}}$  [403]. In practice, however, GAN training can be unstable and is susceptible to mode collapse; variants such as the Wasserstein GAN (WGAN) improve training stability by replacing the Jensen–Shannon divergence with the Earth Mover’s distance [459].

Quantum Generative Adversarial Networks (QGAN) extends this framework by parameterising the generator, discriminator, or both as Parametrised Quantum Circuits (PQCs) [445,460,461]. A typical setup employs a quantum generator, where a PQC prepares a variational state

$$|\psi(\theta)\rangle = U(\theta) |0\rangle^{\otimes n}, \quad (108)$$

with measurement outcomes distributed according to  $p_\theta(x)$ . These samples are then evaluated by either a classical discriminator (hybrid QGAN) or a quantum discriminator [460]. The adversarial loss generalises Equation (107) to the quantum–classical setting. Gradients of the quantum generator can be computed exactly for wide classes of gates via the parameter-shift rule [162,462,463], which requires multiple circuit evaluations per parameter. Training proceeds by alternating updates of the generator parameters  $\theta$  and discriminator parameters  $\phi$ , analogous to the classical setting. An outline of the QGAN training loop is shown in Algorithm (18).

---

#### Algorithm 18 Quantum Generative Adversarial Networks

---

**Input:** Dataset  $\mathcal{D} = \{x\}$ ; prior  $p_z$ ; number of qubits  $n$ ; PQC ansatz  $U(\theta)$ ; discriminator  $D_\phi$  (classical or quantum); batch size  $B$ ; shots  $S$ ; learning rates  $\eta_G, \eta_D$

**Initialisation:** Randomly initialise generator parameters  $\theta$  and discriminator parameters  $\phi$

**Sample real data:** Draw a minibatch  $\{x^{(i)}\}_{i=1}^B \sim p_{\text{data}}$  from  $\mathcal{D}$ .

**Generate quantum samples:** For  $i = 1, \dots, B$ : draw  $z^{(i)} \sim p_z$ ; prepare  $|\psi(\theta)\rangle = U(\theta)|0\rangle^{\otimes n}$ ; measure in the computational basis with  $S$  shots to obtain  $x_\theta^{(i)} \sim p_\theta$  (apply any required classical post-processing if modelling continuous variables).

**Discriminator step:** Form the (hybrid) adversarial loss (e.g., JS or Wasserstein variant)

$$\mathcal{L}_D = -\frac{1}{B} \sum_{i=1}^B \left( \log D_\phi(x^{(i)}) + \log(1 - D_\phi(x_\theta^{(i)})) \right).$$

Compute  $\nabla_\phi \mathcal{L}_D$  (classical backprop if  $D$  is classical; parameter-shift or other gradient rule if  $D$  is quantum) and update  $\phi \leftarrow \phi - \eta_D \nabla_\phi \mathcal{L}_D$ .

**Generator step:** Hold  $\phi$  fixed and define

$$\mathcal{L}_G = -\frac{1}{B} \sum_{i=1}^B \log D_\phi(x_\theta^{(i)}).$$

Estimate  $\nabla_\theta \mathcal{L}_G$  using the parameter-shift rule [162,462,463], requiring multiple circuit evaluations per parameter, and update  $\theta \leftarrow \theta - \eta_G \nabla_\theta \mathcal{L}_G$ .

**(Optional) WGAN variant:** Replace  $\mathcal{L}_D, \mathcal{L}_G$  with Wasserstein losses and include a gradient penalty term [459].

**Repeat** steps 3–7 for  $t = 1, 2, \dots, T$

**Until** convergence or maximum iterations.

**Output:** Trained generator parameters  $\theta^*$  (and discriminator  $\phi^*$ ).

---

QGANs have been proposed for applications in finance, particularly for learning and efficiently loading probability distributions of asset price changes [196]. Exact state preparation of arbitrary distributions requires  $O(2^n)$  gates, but QGANs can approximate state loading with  $O(\text{poly}(n))$  gates after training. Zoufal et al. demonstrate this in the context of European option pricing, where the learned generator prepares a distribution matching asset returns, enabling quantum amplitude estimation to evaluate expected payoffs more efficiently [196].

Beyond finance, QGANs have been investigated for tasks such as image [461] and molecular generation [464]. Empirically, hybrid QGANs have matched or exceeded classical baselines on specific tasks while using fewer trainable parameters in the quantum modules [464]. Theoretically, quantum sampling may enable more efficient exploration of probability landscapes [445], although such claims remain speculative.

Despite their promise, QGANs inherit both the adversarial instability of classical GANs and the limitations of NISQ hardware. Deep variational circuits are prone to barren plateaus, where gradients vanish exponentially with system size, hindering training [153,465,466]. In addition, adversarial training amplifies shot noise, requiring a large number of measurement shots to estimate gradients and loss functions accurately. The parameter-shift rule itself adds further overhead, as each parameter update requires multiple circuit evaluations [463]. Surveys of QGANs [467] emphasise that while they show strong conceptual potential for finance and other data-intensive applications, realising quantum advantage remains contingent on overcoming these limitations.

#### Other Approaches

Several other quantum generative approaches have been developed. Such as, Quantum Circuit Born Machine (QCBM) [398,468], where a PQC directly encodes a probability distribution via the Born rule. QCBMs are also trained by minimising the KL divergence between the target and model distributions through circuit parameter optimisation.

In addition, Quantum Bayesian Network [469,470] extend classical Bayesian networks by replacing conditional probability distributions with quantum channels between nodes, enabling the representation of quantum correlations and entanglement in structured probabilistic models. While classical Bayesian network inference is NP-hard for general DAGs, QBNs may achieve polynomial speedups for certain network topologies through quantum amplitude amplification [469]. Recent work has also explored quantum-like Bayesian networks that use quantum probability theory for classical data, showing advantages in modeling paradoxical human decision-making and cognitive phenomena [471,472].

### 22.3. Quantum Dimensionality Reduction

Dimensionality reduction [473] has long been an essential tool for processing large-scale datasets in ML, signal processing, and other data-driven fields. They compress high - dimensional data into lower-dimensional representations while preserving essential information. In classical frameworks, popular methods include Principal Components Analysis (PCA) [474], Singular Value Decomposition (SVD)-based approaches [475], and nonlinear manifold learning [476]. However for extremely high-dimensional data, these techniques can become computationally expensive. In this section we review the Quantum Principal Components Analysis (QPCA) [477] approach for QDR.

#### 22.3.1. Quantum Principal Component Analysis

In Quantum Dimensionality Reduction (QDR) [477], the fundamental idea is to represent high-dimensional data as quantum states and perform transformations (e.g., QPE, QSVT) that isolate essential components (principal components, singular values, etc.). Given a dataset  $\{x_1, \dots, x_M\}$  where  $x_i \in \mathbb{R}^N$ , the goal is to find lower-dimensional representation  $\{y_1, \dots, y_M\}$  where  $y_i \in \mathbb{R}^d$  (with  $d \ll N$ )

that retains most of the ‘useful’ structure. For instance, PCA [477] finds an orthonormal projection matrix  $W \in \mathbb{R}^{N \times d}$  that minimises

$$\min_W \sum_{i=1}^M \left\| \mathbf{x}_i - WW^T \mathbf{x}_i \right\|^2, \quad (109)$$

subject to  $W^T W = I_d$ . To employ quantum algorithms, one typically represents classical vectors as normalised quantum states. As detailed in previous sections, a frequently used technique is amplitude encoding:

$$|\psi_i\rangle = \frac{1}{\|\mathbf{x}_i\|} \sum_{j=1}^N x_i^{(j)} |j\rangle, \quad \text{where } \|\mathbf{x}_i\| = \sqrt{\sum_{j=1}^N |x_i^{(j)}|^2}. \quad (110)$$

Considering a statistical mixture of states  $\{|\psi_i\rangle\}$  with probabilities  $\{p_i\}$  results in the density matrix:

$$\rho = \sum_{i=1}^M p_i |\psi_i\rangle \langle \psi_i|. \quad (111)$$

Thus QDR seeks to approximate the most significant eigenvectors or singular values of this density matrix (or a related operator) with quantum subroutines.

One of the earliest approaches of Quantum Principal Components Analysis (QPCA) was introduced by Lloyd et al. [477]. The classical data vectors are normalised and encoded into  $n$ - qubits quantum states, which are then combined into a density matrix  $\rho \in \mathbb{C}^{2^n \times 2^n}$  given by Equation (111). The QPCA algorithm aims to approximate the largest eigenvalues and eigenvectors of  $\rho$ ; its spectral decomposition is:

$$\rho = \sum_{k=1}^{2^n} \lambda_k |u_k\rangle \langle u_k|, \quad \text{where } \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \quad (112)$$

and  $\sum_k \lambda_k = 1$ . The QPCA of [477] primarily utilises the QPE on an operator encoding  $\rho$  to extract eigenvalues  $\lambda_k$  and filter out the dominant subspace. This procedure is summarised in Algorithm (19). This algorithm potentially had exponential speed-up in dimension compared to best known classical algorithms at the time [288,477]. This potential speed-up along with other QML has since been ‘dequantised’ by quantum-inspired classical methods [478]. Furthermore, to achieve an  $\epsilon$  approximation eigenvalue, QPE requires  $O\left(\frac{1}{\epsilon}\right)$  calls to oracle with block-encoding or QRAM access to  $\rho$  [477]. The overhead in building the necessary oracles for QPE may undermine practical quantum speed-ups. Since, implementing QPE on NISQ hardware is infeasible, research has focused on alternative near-term approaches [479–481]. Another related method is tensor PCA [48,482], studied through the spiked tensor model [483] as a higher-order analogue of the spiked covariance model from matrix PCA. A quantum algorithm [482] has been proposed with provable runtime guarantees in this model, achieving a quartic speedup over classical methods and recovering the signal at lower signal-to-noise ratios [48]. However, the spiked tensor model is not directly linked to clear practical applications, and have unfavorable runtime scaling for large systems, limiting the practical impact of these results.

---

#### Algorithm 19 Quantum Principal Components Analysis

---

**Input:** Data density matrix  $\rho$ , rank cutoff  $r$ .

**Data Loading:** Construct or load the density matrix  $\rho$  that encodes your data distribution.

**Quantum Phase Estimation:** (a) Implement an oracle  $U_\rho$  (or a block-encoding of  $\rho$ ) for phase estimation. (b) For each eigenstate  $|u_k\rangle$ , QPE encodes the eigenvalue  $\lambda_k$  into the phase register.

**Eigenvalue Filtering:** (a) Measure the phase register to approximate  $\lambda_k$ . (b) Post-select or project onto those eigenstates with the largest eigenvalues.

**Repeat** the QPE as needed to refine precision.

**Output:** The top  $r$  eigenvectors  $\{|u_1\rangle, \dots, |u_r\rangle\}$  (principal components) and corresponding eigenvalues.

---

## Other Approaches

Another type of approach for QDR employs Variational Quantum Algorithm [152] which are apt for NISQ devices and circumvent QPE. One popular example is the Variational Quantum Auto-Encoder (VQAE) [484], which compresses a quantum state  $|\psi_i\rangle \in \mathcal{H}_N$  into a smaller subspace  $\mathcal{H}_d$  where  $d \ll N$ , and then reconstruct it with minimal fidelity loss. Training involves iteratively adjusting the circuit parameters  $U(\theta)$  to minimise a reconstruction-error-based cost function.

Beyond variational methods, a general method is provided by Quantum Singular Value Transformation (QSVT) [48,138], which takes a block-encoding of a matrix and a polynomial  $f$  and produces a block-encoding of the singular value transformation  $f^{(SV)}(A)$ . By interleaving applications of the block-encoding and its adjoint with phase rotations derived from a classical sequence, QSVT enables polynomial transformations of singular values and thereby encompasses a wide range of dimensionality-reduction primitives.

Quantum dimensionality reduction holds promise for tackling large-scale data challenges. While algorithms like QPCA, Variational Quantum Autoencoders, and QSVT show asymptotic advantages, major issues, like data loading, hardware noise, and circuit complexity make it difficult to achieve definitive quantum advantage in real-world scenarios.

## 23. Quantum Reinforcement Learning

Reinforcement Learning (RL) [291,485] is an area of Machine Learning that aims to train models or *agents* via the notion of actively taking actions in an *environment* that maximises the reward. It is one of the three fundamental ML paradigms, alongside supervised and unsupervised learning [486]. It is used to find the optimal behaviour or policy within a specific environment to achieve a long-term goal.

One of the main differences of RL compared to other ML paradigms is that it eliminates the need for labelled input/output pairs. Instead, the agent learns from scalar reward signals, emphasising the exploration-exploitation trade-offs [487]. This makes it suitable for problems where optimal actions are not known a priori. Additionally, there is no need for sub-optimal actions to be explicitly corrected. The focus is on finding a balance between discovering new policies within an environment (exploration) while effectively using current knowledge (exploitation) [488]. RL can include approaches such as value-based, policy gradient and model-based methods. In this section we will focus on the first two: value-based and policy gradient methods.

The main points in RL can be identified as follows:

- **Input:** the initial state<sup>25</sup> of the model.
- **Training:** based upon the input, the model will follow (or evolve) through a sequence of events and produce an output. Based on the output, the model will receive a reward or penalty signal.
- **Output:** an action selected from possible actions, aiming to maximise cumulative reward over time. The best solution or policy is decided based on maximising the expected return.

Basic RL is modelled as a discrete-time, finite-state Markov Decision Process (MDP)<sup>26</sup> [489]. Such an MDP contains a set of environment and agent states  $S$ , a set of actions,  $A$ , and the one-step dynamics of the environment [485]. The probability of transition at time  $t$  from state  $s \in S$  to  $s' \in S$  under action  $a \in A$  [485] can be defined as

$$P_a(s'|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}. \quad (113)$$

<sup>25</sup> The 'state' refers to all the information available to the model/agent, defined by the environment.

<sup>26</sup> When an environment is an MDP, it means that only the current state contains all the information needed to predict the future, not the entire history of states and actions that came before [485]. This property is called the *Markov property*.

Over an episode (a sequence of interactions terminating in a terminal state), the agent accumulates a sequence of rewards [485]. The return from time  $t$  is defined as the total discounted reward:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \quad (114)$$

which the agent aims to maximise, where  $\gamma \in [0, 1]$  is a discount factor, and  $R_{t+k+1}$  denotes the reward at timestep  $t + k + 1$ .

An RL agent interacts with its environment in discrete time steps [485]. At each time,  $t$ , it receives the current state  $S_t \in S$  and a reward  $R_t$ . Following this, it chooses an action  $A_t \in A$  which evolves the environment to a new state  $S_{t+1}$  and generates a new reward  $R_{t+1}$ . The reward is computed according to the function  $R_a(s, s')$  and depends on the transition  $R_{t+1} : (S_t, A_t) \rightarrow S_{t+1}$ . The set of successive decisions creates a history [487] within the MDP model as  $h_n = (S_0, A_0, S_1, A_1, \dots, S_{n-1}, A_{n-1}, S_n)$ .

A policy  $\pi$  is a strategy defining the agent's behaviour, mapping states to a probability distribution over actions [490]. Under a given policy  $\pi$ , the state-value function  $V_\pi(s)$  is the expected return when starting from state  $s$  and following  $\pi$  thereafter [485,490,491]. Similarly, the action-value function (Q-function)  $Q_\pi(s, a)$  is the expected return starting from state  $s$ , taking action  $a$ , and then following  $\pi$ . The goal of RL is to learn a mapping from states to actions that maximise the expected cumulative reward [485,491], or in mathematical terms, learn a policy  $\pi : S \times A \rightarrow [0, 1]$ ,

$$V_\pi(s) = \sum_{a \in A} \pi(a|s) \left[ R_a(s) + \gamma \sum_{s' \in S} P_a(s'|s, a) V^\pi(s') \right], \quad (115)$$

where  $V^\pi$  is the state-value function under policy  $\pi$ , and  $\pi(a|s)$  defines the probability of the agent selecting action  $a$  at state  $s$  according to  $\pi$ .

RL can be applied to applications within finance, such as pricing, hedging, portfolio or investment allocation [492–494], managing asset liability, or exploring and optimising tax revenues and their implications.

### Value-Based Methods

Ref. [495] Value-based methods form a fundamental class of RL algorithms. These methods learn to estimate the value of states or state-action pairs, then derive optimal policies from these learned values. This approach transforms the control problem into a value estimation problem.

The core of value-based methods is Q-learning [485], this algorithm aims to learn the optimal action-value function  $Q_*(s, a)$ . In its tabular form, the learning update rule is:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \quad (116)$$

where  $\alpha$  is the learning rate,  $r$  is the immediate reward,  $\gamma$  is the discount factor, and  $s'$  is the next state.

Deep Q-Learning (DQL) extends Q-learning to high-dimensional state spaces using neural networks [496]. The network  $\hat{Q}(s, a; \theta)$  approximates the optimal action-value function, according to the Bellman optimality equation [485]:

$$Q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q_*(S_{t+1}, a') | S_t = s, A_t = a], \quad (117)$$

where  $Q_*(s, a)$  represents the maximum expected return achievable from state-action pair  $(s, a)$ , following the optimal policy. At time  $t$ , the agent observes state  $S_t \in S$ , selects action  $A_t \in A$ , receives reward  $R_{t+1}$ , and transitions to new state  $S_{t+1}$ . The network parameters  $\theta$  are optimised by minimising the loss:

$$L(\theta) = \mathbb{E} \left[ (y - \hat{Q}(s, a; \theta))^2 \right], \quad (118)$$

where  $y = r + \gamma \max_{a'} \hat{Q}(s', a'; \theta^-)$  and  $\theta^-$  are the parameters of a target network that is periodically updated from  $\theta$  to stabilise training [496].

Despite their success of these algorithms, classical value-based methods face several scalability challenges. Tabular Q-learning becomes intractable for large state spaces, requiring memory and updates proportional to  $|S| \times |A|$ , making it impractical for continuous or high-dimensional domains [485]. Even with function approximation, Q-learning with non-linear approximators can diverge due to the deadly triad of off-policy learning, bootstrapping, and function approximation [497]. Sample inefficiency plagues both approaches, Q-learning requires visiting each state-action pair multiple times for convergence, while DQL often needs millions of frames despite using experience replay [496]. Some of these limitations motivate exploring quantum computing approaches that could potentially provide polynomial or exponential speedup.

Recent advances in QML comes in the form of hybrid quantum-classical machine learning systems [498]. These hybrid QML models use PQC with a few qubits that can be trained to solve certain benchmarking environments, exploring whether PQCs might provide advantages for specific learning tasks.

In [499], it was proposed to represent the state set with a quantum superposition state; the eigenstate obtained by randomly observing the quantum state is the action. The probability of the action is determined by the probability amplitude, which is updated in parallel according to the rewards. The probability of the 'good' action is amplified by using iterations of Grover's algorithm (see Section 9). In addition, they showed that the approach makes a good trade-off between exploration and exploitation using the probability amplitude, and could potentially speed up learning. In [500], it was shown that the computational complexity of a projective simulation can be quadratically reduced for specific problem instances [404]. This speedup applies to particular structured environments rather than providing a general advantage.

Quantum Deep Q-Learning (Q-DQL) replaces the neural network in DQL with a PQC. The quantum circuit takes the classical state  $s$  as input (encoded into a quantum state  $|\psi(s)\rangle$  via amplitude encoding, basis encoding, or other quantum state preparation methods) and outputs expectation values corresponding to Q-values for each action. For each action  $a$ , the Q-value is obtained by measuring an appropriate observable  $\hat{O}_a$  on the quantum state  $|\psi(s; \theta)\rangle$  [501]:

$$Q_\theta(s, a) = \langle \psi(s; \theta) | \hat{O}_a | \psi(s; \theta) \rangle. \quad (119)$$

where  $\hat{O}_a$  is the observable for action  $a$  [501].

The parameters  $\theta$  are updated using gradient-based optimisation, where gradients  $\nabla_\theta Q_\theta(s, a)$  can be estimated using the parameter-shift rule [498, 502]. For each parameter  $\theta_i$ , the circuit is evaluated at  $\theta_i \pm \frac{\pi}{2}$  to obtain derivative estimates. This hybrid learning loop is analogous to training a classical neural network in DQL [501].

Below is a more detailed version of the hybrid Q-DQL process, Algorithm (20). The quantum circuit replaces the classical neural network in computing the approximate Q-values.

---

#### Algorithm 20 Quantum Deep Q-Learning

---

**Input:** A PQC  $Q_\theta$ , learning rate  $\alpha$ , discount factor  $\gamma$ , replay buffer  $\mathcal{D}$

**Initialise:** Environment, circuit parameters  $\theta$ , target parameters  $\theta^-$ , and observe initial state  $s$

**Encode state:** (a) Encode  $s$  into quantum state  $|\psi(s)\rangle$  via chosen encoding scheme (b) Measure observables  $\{\hat{O}_a\}$  to obtain  $Q_\theta(s, a)$  for all actions  $a$  (c) Select action  $a$  using  $\epsilon$ -greedy policy based on  $Q_\theta(s, \cdot)$  (d) Execute  $a$  in the environment, observe reward  $r$  and next state  $s'$  (e) Store transition  $(s, a, r, s')$  in replay buffer  $\mathcal{D}$

**Sample a mini-batch from  $\mathcal{D}$ :** (a) Compute target values Equation (118) (b) Update  $\theta$  via gradient descent using parameter-shift rule (c) Periodically update target parameters:  $\theta^- \leftarrow \theta$  (d) Set  $s \leftarrow s'$

**Output:** Trained quantum circuit parameters  $\theta$ .

---

Ref. [502] demonstrated Q-DQL on grid-world environments using quantum devices. They achieved comparable performance to classical methods for small problems. Ref. [495] applied PQCs to Atari games, showing that data re-uploading encoding can match classical performance with fewer parameters. Ref. [503] implemented Q-DQL on real quantum hardware, achieving learning despite hardware noise.

Although quantum methods offer potential polynomial speedup for search within value iteration [499,504], actual performance benefits remain heavily contingent on hardware capabilities and problem structure [501,505]. The prospective advantages include enhanced function approximation and potential polynomial or exponential speed-ups in very specific scenarios [443,499]. However, most implementations show no advantage over classical methods on current hardware [506].

### Policy Gradient Methods

Policy gradient methods take a fundamentally different approach from value-based methods. Rather than learning value functions and deriving policies, these methods directly parameterise and optimise the policy  $\pi(a|s; \theta)$  [507]. This direct optimisation often provides better convergence properties for continuous action spaces.

The core idea is to maximise the expected return:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[ \sum_{t=0}^T \gamma^t r_t \right], \quad (120)$$

where  $\tau$  denotes a trajectory sampled from policy  $\pi_{\theta}$ .

The policy gradient theorem provides the gradient [485]:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) G_t \right], \quad (121)$$

where  $G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$  is the return from timestep  $t$ .

The REINFORCE algorithm [508] is one of the most commonly used policy gradient algorithms. It implements Equation (121) using neural networks to parameterise the policy [408]. The main idea is to modify the policy to favor sequences of actions that led to high returns, and those actions are more likely to happen in the future [509]. The update rule is:

$$\Delta \theta = \eta \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) G_t, \quad (122)$$

where  $\eta$  is the learning rate [408].

Policy gradient methods suffer from high variance in gradient estimates [507]. Sample efficiency is poor, as thousands of trajectories may be needed for reliable gradient estimation [510]. The variance results into unstable policy updates which increase convergence time [408]. Credit assignment is challenging when rewards are sparse or delayed [511]. Neural network policies face vanishing gradients in deep architectures [512].

Following the approach of replacing neural networks with PQCs demonstrated in hybrid quantum-classical RL [498], the REINFORCE algorithm can be adapted as follows.

The quantum policy is defined as:

$$\pi_{\theta}(a|s) = \frac{e^{\beta \langle O_a \rangle_{s,\theta}}}{\sum_{a'} e^{\beta \langle O_{a'} \rangle_{s,\theta}}}, \quad (123)$$

where  $\langle O_a \rangle_{s,\theta}$  is the expectation value of observable  $O_a$  for state  $s$  and parameters  $\theta$ . The inverse temperature  $\beta$  controls exploration [498].

The state  $s$  is encoded into the circuit through data re-uploading or amplitude encoding [513]. For discrete actions,  $\lceil \log_2 |A| \rceil$  qubits represent the action space [514]. Continuous actions can be represented through expectation values parametrisising distributions.

The gradient  $\nabla_\theta \log \pi_\theta(a|s)$  is computed using the parameter-shift rule:

$$\frac{\partial}{\partial \theta_i} \langle O_a \rangle = \frac{1}{2} (\langle O_a \rangle_{\theta_i + \pi/2} - \langle O_a \rangle_{\theta_i - \pi/2}). \quad (124)$$

The above procedure is summarised in Algorithm (21).

---

#### Algorithm 21 REINFORCE with Parametrised Quantum Circuit

---

**Input:** Episodes  $N$ , learning rate  $\eta$ , discount  $\gamma$

**Initialise:** PQC parameters  $\theta$

**For episode = 1 to  $N_{\text{episodes}}$ :** (a) Initialise  $s_0$  (b) Collect trajectory  $\tau = \{(s_0, a_0, r_0), \dots, (s_T, a_T, r_T)\}$ :

**For  $t = 0$  to  $T$ :** (a) Encode  $s_t$  into quantum state via chosen encoding (b) Measure PQC to sample  $a_t \sim \pi_\theta(\cdot|s_t)$  (c) Execute  $a_t$ , observe  $r_t, s_{t+1}$

**Policy gradient update:** For each  $(s_t, a_t, G_t)$  in trajectory, (a) Compute  $\nabla_\theta \log \pi_\theta(a_t|s_t)$  via parameter-shift rule (b) Update:  $\theta \leftarrow \theta + \eta \nabla_\theta \log \pi_\theta(a_t|s_t) G_t$

**Output:** Optimised PQC parameters  $\theta$ .

---

While replacing a classical neural network with a PQC in quantum algorithms is a conceptually straightforward extension, it introduces profound practical challenges that go beyond standard hardware limitations. Such as the known difficulties of training variational quantum circuits (e.g. barren plateaus and the high cost of gradient estimation) [466,515] which in a agent-environment feedback loop, inaccurate Q-value or policy gradient creates a vicious cycle where errors are constantly amplified. Or the fact that there is no canonical way to encode a classical state  $s$  and action  $a$  into a quantum circuit. The choice of encoding scheme (e.g. amplitude encoding, angle encoding or more complex data re-uploading techniques) determines the model's expressive power and its ability to generalise [516].

For these reasons, while early theoretical work suggested potential quadratic speed-ups in idealised settings [517,518], the path to a Practical Quantum Advantage in Quantum Reinforcement Learning is exceptionally challenging. Progress will require not only better quantum hardware but also fundamental breakthroughs in noise-resilient and stable training techniques specifically designed for a dynamic quantum-classical feedback loop.

## 24. Quantum Cryptography

Modern classical encryption methods rely on the difficulty of solving certain mathematical problems [519]. They are time-tested and regularly updated to account for new attacks or increased computational power. However, quantum algorithms exist that can efficiently solve these problems, breaking the protocols that rely on them [520]. While quantum computers large and powerful enough to implement these algorithms do not yet exist, data is still vulnerable through so-called "harvest now, decrypt later" attacks [521], where it is scooped up *en masse* for decryption when quantum computers are more powerful in what is now known as *Q-day* [522]. In this chapter, we will review some of the basic concepts for classical encryption protocols, simple quantum algorithms that can break these protocols, and a brief outlook to post-quantum cryptography and quantum secure communications. For a comprehensive review of quantum cryptanalysis and resource estimation see [48].

## 25. Classical Protocols

Classical encryption protocols are used for a wide range of applications, from securing communications to digitally signing files which confirms its authenticity. They rely on several different problems in mathematics, such as prime factorisation, and discrete logarithms [519]. In this section, we review

two common protocols: the Rivest-Shamir-Adleman (RSA) [523] and Diffie-Hellman Key Exchange (DHKX) protocol [524,525].

### 25.1. Rivest-Shamir-Adleman (RSA)

The RSA is one of the most famous and widespread family of encryption algorithms [519,526]. The RSA is a public-key encryption protocol used for both secure one-way communication, and digitally signing files to prove authorship [523,526]. The basic formulation is outlined in Algorithm (22), and consists of a *private* and *public* key for encryption and decryption. The RSA protocol can be implemented in two ways:

- Allow the sender to encrypt messages using the public key to the holder of the private key, so that only the receiver can decrypt them, enabling secure one-way communication.
- The holder of the private key can encrypt data and release it so that anyone can decrypt it with the public key, thus digitally signing the message as proof that it has not been tampered with.

The basic encryption and decryption of RSA uses modular exponentiation [519]. The encryption function for some message  $m$  using the public key  $(n, e)$  is

$$c(m) = m^e \bmod n. \quad (125)$$

While the decryption function for some ciphertext  $c$  using the private key  $d$  is

$$m(c) = c^d \bmod n. \quad (126)$$

The values of the public key and private key are computed according to the RSA Algorithm (22). The most commonly-chosen value is  $e = 2^{16} + 1$ , since smaller values of  $e$  make the algorithm faster but sometimes less secure [526].

---

#### Algorithm 22 The RSA Algorithm

---

Randomly choose two large secret prime numbers  $p$  and  $q$ .

Compute  $n = pq$ , which is used as the modulus. Its length in bits is the *key length*.

Compute  $\lambda(n) = \text{lcm}(p-1, q-1)$ , where  $\text{lcm}$  is the least common multiple.

Choose an integer  $e$  such that  $1 < e < \lambda(n)$  and  $e$  and  $\lambda(n)$  are coprime.

Calculate  $d$  such that  $de \equiv 1 \pmod{\lambda(n)}$

Output *public key*  $(n, e)$  and *private key* is  $d$

---

To break the RSA encryption, one needs to factor the public key  $n$  into the two randomly chosen primes  $(p, q)$ . From this, an attacker could compute the secret  $d$  and decrypt the message. The RSA is thought to be secure for large enough  $n$  because the factorisation problem is believed to be intractable for classical computation<sup>27</sup> [519,523]. Brute-force methods to break RSA are possible, for example in 2010, a team of researchers factorised a 768-bit number over about 2000 core-years [528]. However, current recommended bit lengths are at least 2048, rendering the protocol theoretically secure from classical brute-force methods [526].

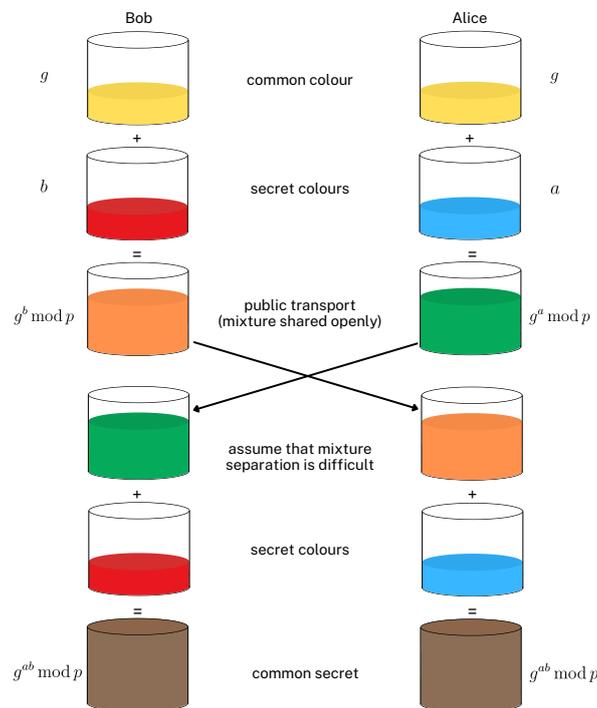
### 25.2. Diffie-Hellman Key Exchange (DHKX)

The DHKX protocol [524,525] was proposed early than RSA and aimed at bootstrapping a secure communications channel. In the early days of cryptography (stretching back to antiquity), establishing such a channel required the presence of a pre-existing one. For example, to send secure letters, the parties involved would need to meet in-person to decide on the security protocol to use. How, then, could a secure channel be established, without having a second one to decide on the security of the first? The key innovation of DHKX is to allow two parties to set up an informationally secure channel

---

<sup>27</sup> Meaning no polynomial-time method for factoring large numbers has been found and there is no proof that none exist [519,527].

over a physically insecure one [524]. The algorithm provides both parties with a private key, which is used to encrypt messages. The parties communicate in such a way that even if an eavesdropper intercepts all communications, they will be unable to deduce the private key.



**Figure 5.** The DHKX protocol, illustrated with mixing paint which represents the numbers Alice and Bob send between themselves to arrive at a shared secure key.

---

### Algorithm 23 Diffie-Hellman Key Exchange

---

The two parties, Alice and Bob, publicly (insecurely) agree on two numbers, a modulus  $p$  and a base  $g$ .

Alice chooses a secret integer  $a$ , and then insecurely sends Bob  $A = g^a \bmod p$ .

Bob chooses a secret integer  $b$ , and then insecurely sends Alice  $B = g^b \bmod p$ .

Both parties compute the secret key  $s$ :

- Alice computes it via  $s = B^a \bmod p$
- Bob computes it via  $s = A^b \bmod p$

Now  $s$  can now be used to encrypt and decrypt messages, establishing a secure channel.

---

The DHKX protocol, which has since been improved manifold times [529], is presented in Algorithm (23) and also illustrated in Figure 5. Using this protocol Alice and Bob can to send each other a secure-key  $s$  in the presence of an eavesdropper (Eve) with access to the public information  $\{A, B, p, g\}$ . This is because for Eve to obtain shared key

$$s = A^b \bmod p = B^a \bmod p, \quad (127)$$

they would have to solve for  $a$  in

$$g^a \bmod p = A \quad (128)$$

which is a discrete logarithm problem [530]. Just like the aforementioned prime factorisation problem, the discrete logarithm problem is intractable; there is no known efficient classical algorithms that can solve it in polynomial time [530,531]. As a result, Alice and Bob have collaboratively constructed a shared secret over an insecure channel.

## 26. Quantum Attacks

Several quantum algorithms exist that can efficiently solve the mathematical problems on which some classical cryptographic protocols depend on. These include

- **Shor's algorithm** (Algorithm 24), which can efficiently factorise numbers and breaks RSA (Algorithm 22) and DHKX (Algorithm 23) [51,532].
- **Grover's algorithm**(Algorithm 4), which can efficiently search unstructured lists and drastically decreases the time taken to brute-force symmetric-key encryption schemes [533,534].
- **Quantum annealing** which supports various algorithms which could also be used to break encryption, but as yet has only been demonstrated on particularly small problems with foreknowledge of the solution [535,536].

Here, we give a brief outline of the key ideas behind Shor's factoring algorithm and quantum annealing based approaches for breaking classical encryption.

### 26.1. Shor's Algorithm

The mathematician Peter Shor first proposed the *quantum factoring algorithm* [51], now known as Shor's algorithm. It is one of the well-known quantum algorithms with a concrete theoretical advantage over classical computers [10]. In the abstract, it can solve the hidden subgroup problem, of which prime factorisation and discrete logarithms are instances, in polynomial time [10,537]. It is a hybrid algorithm: mainly classical, with a quantum subroutine which provides the speedup. Algorithm (21.1.2) outlines Shor's algorithm for factoring  $N$  (odd, composite large number).

---

#### Algorithm 24 Shor's algorithm

---

Pick a random number  $1 < a < N$ .

Calculate  $K = \gcd(a, N)$ , where  $\gcd(\cdot)$  is the greatest common divisor.

If  $K \neq 1$ , then we are done.

Otherwise, use a quantum subroutine, Quantum Phase Estimation (see Section 7) to find the order  $r$  of  $a$  modulo  $N$ , i.e.:  $a^r \equiv 1 \pmod{N}$

If  $r$  is odd or  $a^{r/2} \equiv -1 \pmod{N}$ , return to step 1.

Calculate  $g = \gcd(N, a^{r/2} - 1)$  and  $g = \gcd(N, a^{r/2} + 1)$ .

If  $g \neq 1$ , we are done. Otherwise, return to step 1.

---

The various conditional branches are due to the implied properties of the calculated numbers. For example, if the calculated  $r$  were odd, then  $a^{r/2}$  would not be an integer, and so the algorithm would not work. For a more in-depth explanation of the algorithm, see [44,51,532]. The quantum subroutine, QPE, provides an exponential speed-up in solving the following problem [44,51]: given coprime integers  $a, N$ , find the order  $r$  of  $a \pmod{N}$ , which is the smallest positive integer such that  $a^r \equiv 1 \pmod{N}$ . However, as of writing, Shor's algorithm has only been implemented and used to factor very small primes [538–540] due to the large quantum resource requirements of QPE (see Section 7). Early fault-tolerant estimates for factoring a 2048-bit number required about 20 million physical qubits running over eight hours, depending on the chosen Quantum Error Correction code [541]. This estimate has recently been improved to less than 1 million physical qubits running over a week [542]. This large physical qubit requirement and error correction quality assumptions is significant beyond current state-of-the-art, hence the projected timelines are very uncertain, ranging from around 2030 [543] to 2039 and beyond [544]. Thus, while it is unlikely that Shor's algorithm will be used relatively soon to break modern encryption (based on major hardware providers' technology roadmaps), there is ongoing research and infrastructure preparations for alternative encryption that is safe from quantum attacks (see Section 27).

### 26.2. Quantum Annealing

Due to some of the limitations of Shor's algorithm, particularly the large quantum resource requirements for QPE, alternative quantum-based approaches to integer factorisation have been devel-

oped [48]. Quantum Annealing (QAnn) (see Section 13) has emerged as a promising alternative [545], with a recent demonstration of factorisation of an 80-bit number, the largest yet [546]. While this is still orders of magnitude away from any practical application, this result demonstrates some of the key advantages QAnn holds over traditional gate-based quantum algorithm, especially in the NISQ-era. While Shor's algorithm requires thousands of logical qubits, translating to millions of physical qubits (as described in Section 26.1), QAnn requires far fewer qubits, scaling at most as  $O(\log^2(N))$  with the integer  $N$  [545]. However, QAnn requires statistical sampling, and has an unknown time complexity, while Shor offers time complexity guarantees in finding the solution [10,545,546].

In this section, we will outline one of the simplest applications of QAnn for factorisation: By reformulating the problem as an optimisation problem in QUBO form as shown in [547]. In this approach, the problem of factorising  $N = pq$  into the primes  $p, q$ , can be phrased as the problem of minimising the cost function

$$C(p, q) = (N - pq)^2, \quad (129)$$

where  $p, q$  are represented in binary. After adding auxiliary variables to convert the quartic into a quadratic cost function, the authors in [547] were able to factorise  $N = 4137131$  into  $p = 2029$  and  $q = 2039$  using 93 variables. Alternative approaches that have been proposed range from implementing full-adder circuits [548] to integrating annealing into a hybrid workflow [546]. The latter approach was used to achieve the current record-breaking result above of factorising an 80-bit integer.

Significant challenges remain to be solved, such as algorithmic development for efficient implementation and connectivity of annealing hardware, before QAnn-based approaches become a viable method of solving the factorisation problem [546,548]. Nevertheless, the advantages of annealing-based approaches mean it is possible that it might be the first to factor 1024 or 2048-bit RSA. Hence, alongside Shor's algorithm, organisations should also monitor progress on algorithmic and hardware advances in Quantum Annealing.

## 27. Post-Quantum Cryptography

In response to the growing threat of a future quantum computer that renders modern encryption insecure, there has been a huge amount of research and development into so-called Post-Quantum Cryptography (Post-QC), which refers to classical encryption protocols that aim to be secure against both classical and quantum attacks [549–552]. This has culminated in the 2024 announcement by the US National Institute of Standards and Technology (NIST) of a set of three standard Post-QC protocols recommended for use in key establishment and digital signing [553–555]. Even before this, however, post-quantum protocols were beginning to be adopted by large technology companies, notably Google [556] and Apple [557]. As a result, there is not only a large body of literature and code available for easy implementation, but a number of organisations providing Post-QC as a service, minimising the need for customised (and therefore potentially unsafe) implementations. The details of these protocols are outside the scope of this paper, but we present below a brief list of the key families as a guide to further perusal:

- Lattice-based [558] which employs difficult lattice problems such as the Shortest Vector Problem.
- Code-based [559], which uses error-correcting codes to encrypt data.
- Multivariate [560], which is based on the hard problem of solving multivariate polynomials over finite fields.
- Hash-based [561,562], which utilises the hard problem of reverse-engineering hash functions which map arbitrary inputs to fixed-length outputs.

These algorithms are generally complex, and it is not recommended that each organisation implement their own version themselves, but rather partner with a trusted provider. The UK's National Cyber Security Centre has published a migration roadmap for Post-QC [563] which has estimated completion between 2031 to 2035. These timelines are in line with estimates of  $Q$ -day happening between 2030

to 2039 [522,543,544]. Financial organisations should therefore proactively work on the migration to Post-QC encryption.

## 28. Quantum Communications

An important issue with cryptography protocols is that there is every chance that new algorithms, either quantum or classical, will be created which can break them. For example, the Supersingular Isogeny Key Encapsulation (SIKE) algorithm [564,565] was considered secure for more than a decade<sup>28</sup> before it was broken in 2022 by a team of researchers using a legacy Intel chip [567]. Of course, the same potential for new algorithms also exists for classical cryptographic protocols, but the novelty of quantum protocols means they have had less time to be tested.

This uncertainty in encryption implies that existing communication networks can always be intercepted and encrypted data stored in anticipation of novel decryption attacks in the future. Thus, it is desirable an alternate way of encrypting data by generating and distributing a secret key in a way which is inherently protected by the laws of physics. This what is achieved by the Quantum Key Distribution (QKD) which was first proposed by Charles Bennett and Gilles Brassard in 1984 [568] forming what is now known as the BB84 protocol [569]. The security of the protocol can be formally established under the assumption of an idealized, error-free quantum communication channel. The theoretical proof relies on two foundational premises: (1) the quantum-mechanical principle that any attempt to extract information from non-orthogonal states necessarily perturbs the signal<sup>29</sup>, consistent with the no-cloning theorem [570]; and (2) the availability of an authenticated, publicly accessible classical communication channel [569].

The key idea of the BB84-QKD is to send a secure key encoded into quantum states in a way that it would be impossible for an eavesdropper to copy it without being detected. The information is exchange as outline in Algorithm (25) is encoded in polarised states of photons and measurement basis given by:

$$M_{+} = \begin{cases} |0\rangle & \text{horizontal } \rightarrow \\ |1\rangle & \text{vertical } \uparrow, \end{cases} \quad (130)$$

for the rectilinear basis and

$$M_{\times} = \begin{cases} |+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}} & \text{north-east } \nearrow \\ |-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{south-east } \searrow, \end{cases} \quad (131)$$

diagonal basis. Alice chooses a random binary key, encodes it into polarisation of photons prepared in a randomly chosen basis and sends to Bob via a public quantum communication channel (assumed to be error-free). Bob measures these photons in his own randomly chosen set of measurement basis to generate a set of bit strings. Alice and Bob then communicate via a classical (potentially insecure) channel to compare their chosen set of preparation and measurement basis. They sift out the results of the cases where the preparation and measurement basis differ and keep the rest. In principle, assuming error-free quantum channel and no eavesdropper (Eve), the remaining set of bit strings would be almost identical (possible some very small error). However, if Eve has intercepted their message by measuring the photons in a random basis and then sending 'spoof' photons prepared in that basis to Bob, there would be a significant Quantum Bit Error Rate (QBER) rate given by [568]

$$\text{QBER} = \frac{\text{Number of mismatched bits}}{\text{Total compared bits}}. \quad (132)$$

There is a threshold for statistically acceptable QBER, for BB84 is about 11%, thus anything above it would mean Eve has listened to the message. For Eve to be undetected, she would need to have

<sup>28</sup> SIKE was a finalist in NIST's competition for post-quantum protocols [566].

<sup>29</sup> In other words, there is no physical process that can clone a quantum state; see Section (41) for more details.

a device that makes exact copies of the photon states without measurements which is forbidden by the no-cloning theorem [570]. If the QBER is below the BB84 threshold (meaning no eavesdropper in quantum channel), then Alice and Bob can proceed to use classical information reconciliation and privacy amplification techniques to construct a shared private key [568,571].

---

#### Algorithm 25 BB84 - Quantum Key Distribution

---

**Goal:** Alice to send to Bob a key to encrypt data via a public channel that could be intercepted by an eavesdropper, Eve.

**Preparation:** Alice generates a random set of  $\{K_A, M_A\}$  of a key of binary  $n$  bits string  $K_A \in \{0, 1\}^n$  and measurement basis  $M_A \in \{M+, M\times\}^n$ . The random key  $K_A$  is encoded into  $n$  photon states polarised as either  $\{|\uparrow\rangle, |\rightarrow\rangle\}$  if  $M_A = M+$  or  $\{|\nearrow\rangle, |\searrow\rangle\}$  if  $M_A = M\times$ .

**Transmission:** Alice sends to Bob this sequence of  $n$  photons via a public channel.

**Measurement:** Bob generates a random sequence of measurement basis  $M_B \in \{+, \times\}^n$  and measures each photon to obtain a key  $K_B \in \{0, 1\}^n$ .

**Sifting:** Alice and Bob share via a public channel their randomly chosen measurement basis  $M_A$  and  $M_B$ , respectively. They sift out the results of bits that were measured in the wrong basis as they were prepared and keep the matching  $k$  basis set  $M_{AB} = M_A \cap M_B$  where  $k < n$ .

**Error checking:** Alice randomly chooses  $k/2$  bit strings of the set  $M_{AB}$  and shares the values with Bob who does the same via a public channel. They use these values to compute an estimate of the QBER given by Equation (132). If the QBER is above some tolerant threshold ( $\sim 11\%$  for BB84), they both abort or try again afresh another time because it is highly likely that Eve is listening to their communication.

**Key Distillation:** If QBER is below the threshold, they then use classical post-processing (error correction and privacy amplification) [569] to generate the final shared key.

**Output:** Shared private key between Alice and Bob  $K_{AB}$  that can be used to encrypt their messages.

---

The BB84 protocol is one of the simplest QKD, and more sophisticated methods exist utilising quantum superposition and entanglement to increase fidelity and decrease the number of photons which need to be sent [569]. In particular, the 1991 protocol by Ekert [572] (known as E91) uses entangled Bell states to create a more secure QKD. Implementations of such QKD already exist and are available commercially; for a review of the progress and state-of-the-art of the field see Refs. [573–575]. The relative lack of a widespread quantum network channels means their use is limited to local communications and research. However, an example of a fully-realised implementation is the quantum network in the Chinese city of Hefei, which allows for secure QKD over 4,600 kilometres [576]. It is built with a combination of fibre-optics and ground-to-satellite links and is currently the largest of its kind in the world [576].

Another related set of quantum communication protocols are *quantum money* [26,577] (see Section 41). A recent experimental implementation reported a quantified time advantage over optimal classical cross-checking protocols for secure financial transactions [578]. This advantageous verification speed-up of *quantum money* and security of QKD could potentially fuel rapid industrial adoption for applications requiring high security, privacy and minimal transaction times, like financial trading and network control. However, the pace of widespread adoption of QKD schemes will also be dependent at the rate at which quantum-capable networks infrastructure are built, which involves government regulations.

## 29. Quantum Random Number Generation

Random Number Generators (RNGs) are vital in not only many cryptographic protocols, but also in wider areas of fintech including blockchain technologies and smart contracts [579–581]. True RNG can be achieved via random physical processes, such as inter-event times in atomic decay, voltmeter readings, and semiconductor thermal noise [582]. However, this physical approach has two set of problems:

- **Provable Distribution:** The generated sequence must approximate independent, uniformly distributed variables. In bit-stream implementations, this entails equal probability for 0 and 1,

alongside bitwise independence. Since such statistical properties cannot typically be proven, validation relies on empirical statistical testing to assess conformity with the desired distribution and independence criteria [582].

- **Impractical:** Connecting computational algorithms to physical RNGs is generally more costly and slower (less rapid generation of samples) than *pseudo*-RNGs [583].

Pseudo-RNG are chaotic (but deterministic) functions which generate sequences of samples that are independent and identically distributed (i.i.d.), and uniformly distribution over the interval (0, 1)[584]. They take a seed (sometimes a random input such as movements of computer mouse) and generate a sequence of samples that are difficult to distinguish from true random numbers. The difficulty in predicting the pattern of pseudo-RNGs depends on if the seed is known and properties of the pseudo-random functions [584]. In general, if the seed is not known then many pseudo-random functions produces samples that are almost impossible to predict their pattern. Hence, some RNGs uses a physical process for a true random seed and proceed to generate pseudo-random numbers are 'unpredictable' [582]. More formally, a cryptographically secure pseudo-RNG is believed to be unpredictable if it passes the next-bit test<sup>30</sup> introduced by Blum and Micali [585]. However, just like other cryptography protocols, the unprecitability of pseudo-RNG depends on some hardness assumption which have not been proven [582].

There is a further serious issue when there is a lack of trust of the output of a RNG; it is impossible to look at a sequence of bits and be certain as to whether it is random or not [582,586]. In a zero-trust environment (such as communicating over the internet), there must be a way to certify randomness to all parties involved. Quantum technologies offer alternative sources of randomness which are certified by the laws of quantum physics [587,588]. In particular, there are proposed protocols for Quantum Random Number Generator (QRNG) certified by Bell's inequality violation [587,589]. However, as pointed out in [590], these protocols have a limitation of unable to circumvent trusting a third party to implement all its requirements faithfully. A recent alternative approach proposed by Aaronson and Hung [588] re-purposes the random circuit sampling experiments [14] to construct a certified randomness protocol employing the Linear Cross-Entropy Benchmark threshold. Generating random numbers from random circuit sampling' distribution with entropy above this entropy threshold is computationally hard classically, but is possible using quantum computing. This is what guarantees that the output is truly generated from a QPU not a pseudo-classical simulator, provided you can efficiently verify these distributions. As noted in [588], the drawback of this approach is that it requires a classical computer to perform the verification at a cost that scales exponential with the number of qubits used. This limits its practical implementations to about  $n \sim 60$  qubits [588]. A recent experiment [590] based on this entropy verification has certified over 70 000 bits using 56 qubits of a trapped-ion QPU. This is the largest experiment of of its kind and more research in quantum verification is needed to lower the computational costs [591–593]. Furthermore, the limited adversarial model and protocol assumptions make them not yet ready for industrial deployment [590], but are a promising stepping stone towards more secure future communications and cryptography protocols.

### 30. Banking and Investment

The financial services sector has undergone a profound transformation over the last two decades [594], driven by technological advancements, the proliferation of data, and the increasing demands for efficiency and scalability [595]. Central to this evolution is the integration of computational tools which includes Machine Learning (ML) algorithms, Artificial Intelligence (AI) models for big data, and blockchain technologies [596,597] that are reshaping the landscape of banking and asset management [595,598]. These tools enable financial institutions to handle vast amounts of data, make more informed decisions, and provide personalised services at scale. Even though these classical methods have delivered significant positive results in the sector, they still face computational bottlenecks

<sup>30</sup> Given the first  $k$  bits of its output, no polynomial-time algorithm can predict the next bit with probability significantly better than 0.5 [585].

for certain hard-problems which quantum computing can potentially overcome [9,10,30]. According to a 2023 study by McKinsey [20], quantum computing use-cases in banking and asset management are projected to generate value of about 302 billion<sup>31</sup> USD and 80 billion USD, respectively by the year 2035. These two categories represent a combined 61% of the total projected value of 622 billion USD by year 2035 [20]. In this chapter, we will give a technical overview of some of the use-cases that can benefit from quantum computing in finance that approximately fall under the two categories of banking and investment.

### 31. Asset Pricing

In finance, the accurate pricing of traded assets is a fundamental problem with significant implications for risk management, investment strategies, and regulatory compliance [599]. Asset pricing involves determining the value of different types of financial instruments, such as stocks, bonds, and derivatives. According to Cochrane [181] asset pricing theory seeks to explain why assets that offer uncertain future payments have the prices they do, and why some offer higher returns than others. The theory can be used both to describe how prices work in real-world markets (positively) and to suggest what prices should be if markets were perfectly rational (normatively) [181]. When actual prices don't match the theory's predictions, it could mean the model needs improvement, or that the market is mispricing the asset — creating opportunities for smart investors [600]. The theory is also useful for pricing assets whose values are not directly observable, such as new projects, private investments, or complex financial products, helping guide both private and public financial decisions [181].

The growing complexity of modern financial products, including exotic options, structured derivatives, and interest rate instruments, has necessitated the use of advanced computational techniques to obtain reliable asset pricing [601]. The non-linearity, path-dependence, and high dimensionality inherent in these instruments make asset pricing a challenging task for classical computing. A growing number of practitioners are exploring the use of quantum computing to alleviate some of computational bottle-necks of traditional classical methods [12,602].

In this section, we present the mathematical formulation of assets pricing models and describe some computational methods which includes quantum algorithms. Many of the quantum algorithm employed for this use-cases are described in more detail in Part I of this review and cited papers, hence to make the connection at the problem formulation level. In a similar spirit, this review will not be a comprehensive exegesis of finance models but will present pertinent information to make the case for potential benefit of using quantum computing.

#### 31.1. Pricing Models

There are different approaches to developing a robust pricing models which we will not discuss here. For this review we will follow the approach of Cochrane [181] and formulate the asset pricing problem in terms of stochastic discount factors of the expected asset payoff. This is summarised by the formula

$$p_t = \mathbb{E}(m_{t+1} \cdot x_{t+1}), \quad (133)$$

where  $p_t$  and  $x_{t+1}$  is the asset's price and payoff at times  $t$  and  $t + 1$ , respectively. The stochastic discount factors at time  $t + 1$  are given by

$$m_{t+1} = f(\text{data}, \text{parameters}) \quad (134)$$

where  $f(\cdot)$  is an economic model of the market. These economic models are generally based on two main pricing principles [181]: (i) general equilibrium asset pricing and (ii) rational asset pricing. The former focuses on asset pricing through supply and demand, whereas the latter principle assigns derivative prices such that they are arbitrage-free with respect to more fundamental securities prices.

<sup>31</sup> The total figure is composed of 190 billion USD in corporate, 90 billion USD in retail, 20 billion USD in investment banking, and 2 billion USD in operations.

Irrespective of which economic model you choose for  $f(\cdot)$ , the task of selecting the type of empirical representation for Equation (133) is separate and can be reformulated in terms of returns, price-dividend ratios, expected return-beta relationships, moment conditions, and the distinctions between continuous- and discrete-time frameworks. Essentially, the main benefit of this approach is that it provides a single theoretical framework that covers a variety of cases. Table 4 outlines examples how Equation (133) translates to different pricing problems.

**Table 4.** Examples of how the general asset pricing problem of Equation (133) translated to other pricing problems. Here,  $d_{t+1}$  is the stock dividend and other variables are explicitly defined in [181].

Asset	Price $p_t$	Payoff $x_{t+1}$
Stock	$p_t$	$p_{t+1} + d_{t+1}$
Return	1	$R_{t+1}$
Price-dividend ratio	$\frac{p_t}{d_t}$	$\left(\frac{p_{t+1}}{d_{t+1}} + 1\right) \frac{d_{t+1}}{d_t}$
Excess return	0	$R_{t+1}^e = R_{t+1}^a - R_{t+1}^b$
Managed portfolio	$z_t$	$z_t R_{t+1}$
Moment condition	$\mathbb{E}(p_t z_t)$	$x_{t+1} z_t$
One-period bond	$p_t$	1
Risk-free rate	1	$R^f$
Option	C	$\max(S_T - K, 0)$

The ability to switch between different representations helps in understanding empirical results, as many seemingly distinct approaches are deeply connected. Using discount factors often simplifies analysis compared to portfolio-based approach [181]. For instance, it's easier to verify the existence of a positive discount factor than to compare all possible portfolios. Discount factors also shift the perspective from mean-variance geometry to state-space geometry, which Cochrane [181] highlights as a key intuition behind many traditional results. Because of these advantages, the discount factor framework and state-space view are widely used in academic research and advanced financial practice.

### 31.2. Computational Methods

The approach of Cochrane [181] decomposes the asset pricing problem into three parts:

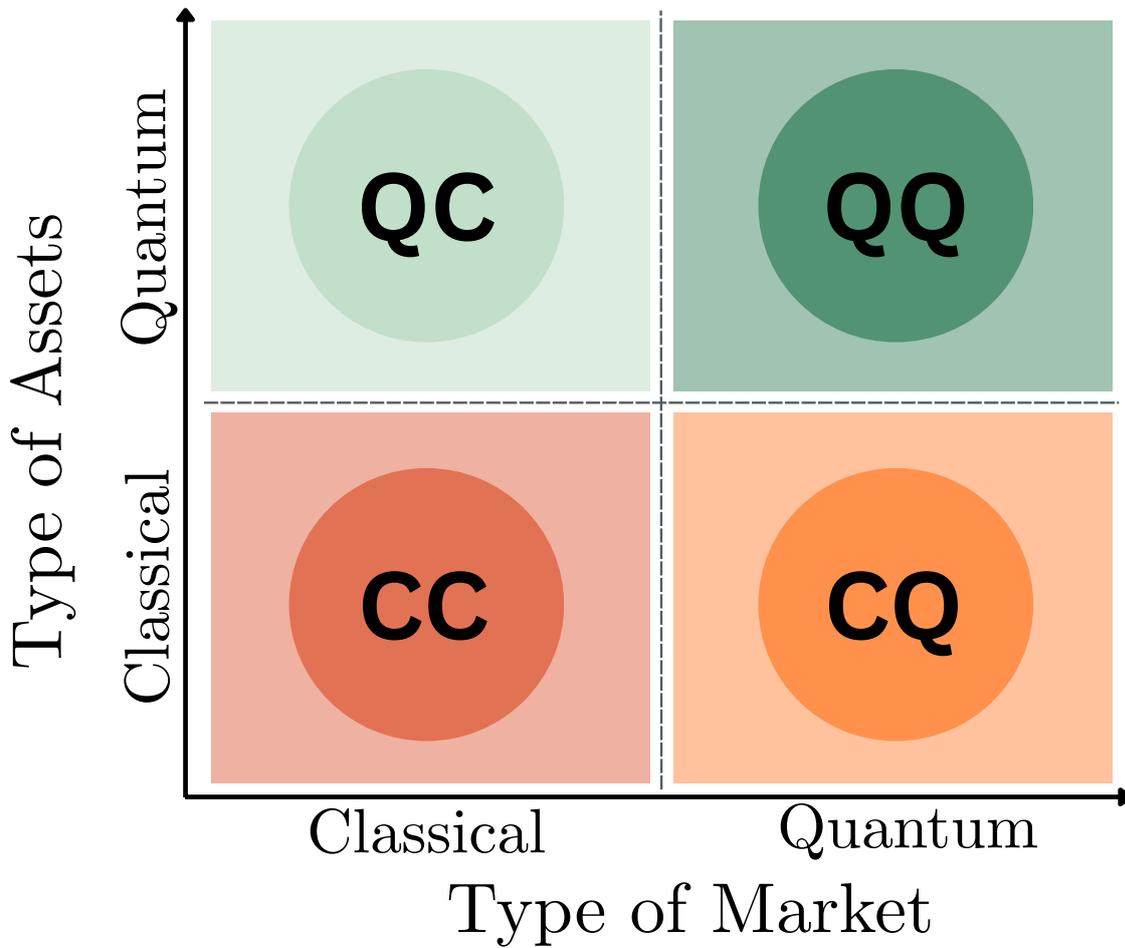
1. Choosing a model for simulating the payoff  $x_{t+1}$ , which is a function of underlying stochastic variables. For example, the payoff for option pricing depends on the spot value  $S_t$  of the underlying asset which can be modeled using the Black-Scholes-Merton (BSM) [84,85] or its variants.
2. Choosing a model of simulating the discount factors  $m_{t+1}$  which are functions of market data and model parameters which incorporates the risks associated with the asset. For example, option pricing has discount factors depending on the filtration  $\mathcal{F}_t$  which represents the market information assumed to be known until time  $t$  (see Section 32.1).
3. Computing the expectation  $\mathbb{E}(\cdot)_{t+1}$  which is conditional on information at time  $t$ .

This abstraction of the problem makes it easy to see where is the root of the computational bottlenecks associated with asset pricing, thus where can quantum algorithm offer an advantage. In the case where part (2) is 'easy', that is, a standard model with parameters fitted on historic data accurately describes the discount factors, then part (3) can be tackled by common computational methods such as Monte Carlo Integration to simulate many paths for the stochastic variables (like consumption, returns, or prices) using random draws and computing sample averages [181,203,603]. The benefit is that MCI is generally flexible, easy to implement and can handle high-dimensional problems since its accuracy is independent to the problem dimension [182]. The main downside is that it has a slow convergence, and needs a large number of samples for accurate simulations [5]. This challenge can be alleviated by employing to Quantum Monte Carlo Integration which, as discussed in Section (15), can potentially have a quadratic speed-up over MCI methods [5].

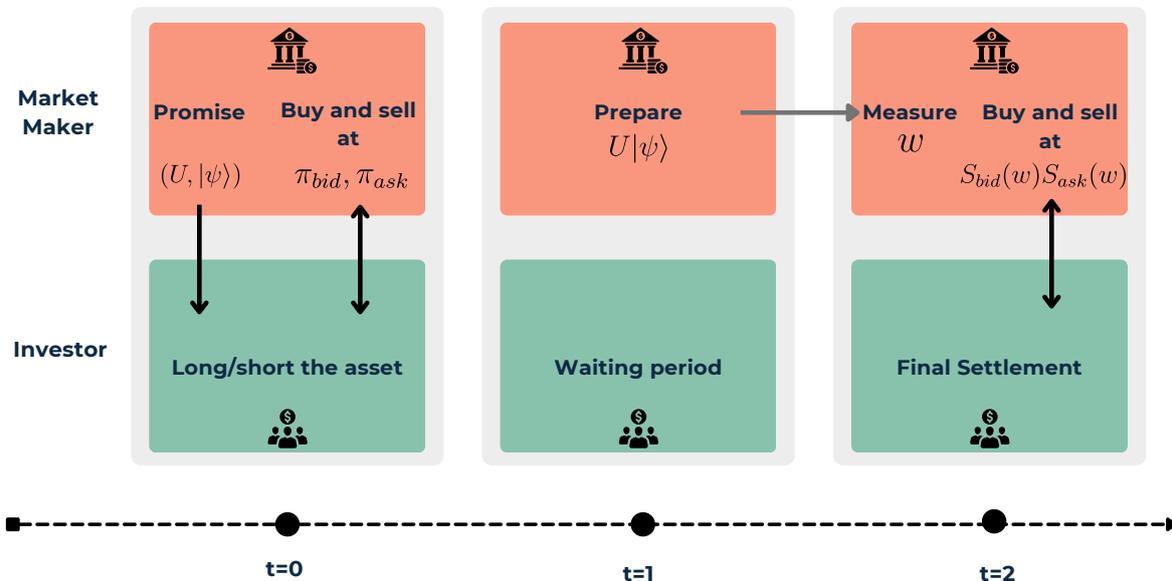
In the case where part (2) is 'hard', that is standard models do not accurately capture the discount factors, then ML techniques can be used to learn from historic market data. Examples include the use of Recurrent Neural Networks (RNN) to run time series predictions [604,605]. However, RNNs are computationally intensive, especially in complex scenarios, hence QML techniques are being considered as potentially more efficient alternatives [9,30,35,41]. A natural approach would be to extend to quantum Recurrent Neural Networks (RNN) as shown in [606] where the learning is performed by parametrised quantum circuits. This approach has demonstrated comparable accuracy in preliminary tests with small datasets; however, its effectiveness at scale remains to be fully validated. Other classical ML techniques for asset pricing have shown great potential to improve the accuracy of learning from historic market data. For example, the authors in [604] studied a collection of ML methods, including linear regression, generalised linear models, principal component regression, and neural networks. An equivalent study of QML for quantitative finance problems which includes asset pricing was done by [41]. A notable example is Quantum Generative Adversarial Networks (QGAN) which can be trained to learn the conditional probability distributions of the asset's value from historical market data [196]. A proof-of-concept implementation of QGAN for the option pricing problem was done by [196,607] using log-normal probability distributions. However, training QGANs can be time consuming for industrial applications, hence some quantum-inspired methods such as tensor networks can be a more efficient alternative [197]. The research is still ongoing and the solution is likely to be a hybrid quantum-classical approach for case of asset pricing.

### 31.3. Quantum Assets

Thus far, we have considered classical assets (described by real values) traded in a classical market (described by real-valued functions). A quantum solution to the asset pricing problem is the real-valued output of a quantum algorithm that embeds the classical variables into quantum states and associated probability amplitudes. An interesting recent theoretical work by Bao and Reberthost [608], has shown that it is possible to define assets that are quantum in nature, referred to as *quantum assets*, as shown in Figure 6(a). These are described by quantum states/density operators in a Hilbert space. Furthermore, asset pricing theorems for quantum assets can be derived analogous to classical asset pricing theorem [608].



(a) A generalisation of types of assets and markets.



(b) A toy model of how quantum assets can be traded in a quantum market discussed in the text.

Figure 6. The results in this section are based on QQ in (a), and the trading toy model (b) for quantum assets. Both figures are adapted based on [608].

### Mathematical Formulation

Assume the market is *quantised* meaning stochastic events are described by probabilities associated with quantum states in the Hilbert space. A quantum asset class, which sits under QQ in Figure 6(a), can be defined as follows [608]:

**Definition 3 (Quantum Asset).** Given a Hilbert space  $\mathcal{H}$ , with  $\dim(\mathcal{H}) = N$ , a quantum asset is a positive semidefinite Hermitian matrix  $\mathcal{A} \in \mathbb{C}^{N \times N}$ , such that for all  $|\psi_{t+1}\rangle \in \mathcal{H}$ , the future payoff is given by

$$x_{t+1} = \langle \psi_{t+1} | \mathcal{A} | \psi_{t+1} \rangle \geq 0 \quad (135)$$

The quantum asset can be bought/sold at a price  $p_t = \pi > 0$ , computed using Equation (133), in a quantum market. It follows that a given a classical asset  $\mathcal{A}_{cl} \in \mathbb{R}^N$ , its quantum asset embedding is the diagonal matrix  $\mathcal{A} = \text{diag}(\mathcal{A}_{cl}) \in \mathbb{R}^{N \times N}$ . A quantum asset assumes a diagonal form in its eigenbasis, a structure that naturally lends itself to financial interpretation [608]. Within this framework, each eigenstate may be regarded as a fundamental event specific to the quantum asset, distinct from the elementary events defined in a classical probability finite sample space  $\Omega$ . The associated eigenvalues represent the respective outcomes or payoffs realised when these corresponding events occur. Figure 6(b) illustrates a reasonable trading scenario in a quantum market with a market maker (top panels) with access to a quantum computer interacting with an investor (bottom panels). The toy model, shown in Figure 6(b), has the following interactions [608]:

**t=0** : The market marker announces the following:

1. A  $n$ -qubit state  $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |w_i\rangle$  and unitary  $U \in \mathbb{C}^{N \times N}$  where  $N = 2^n$ .
2. The initial bid and ask prices denoted by  $\pi_{bid}$  and  $\pi_{ask}$ , respectively.
3. Two function  $\mathcal{S}_{bid}(w)$  and  $\mathcal{S}_{ask}(w)$  that will determine the bid and ask prices at  $t = 2$ , where  $w \in \{0, 1\}^n \subset \Omega$  and  $\mathcal{S}_{bid}, \mathcal{S}_{ask} : \Omega \mapsto \mathbb{R}_+$ .

**t=1** : The market marker uses a quantum computer to prepare the state  $|\psi\rangle$  and then evolves it by the unitary  $U$  to get the final unobserved quantum state  $U|\psi\rangle$ .

**t=2** : The market marker performs a measurement on the final quantum state in the computational basis of  $|\psi\rangle$  to get measurement outcome  $w$  occurring with probability  $|\langle w | U|\psi \rangle|^2$ .

The asset pricing problem involves computing expectations of the form of Equation (133). An equivalent expectation for the quantum asset  $\mathcal{A}$  at time  $t = 1$  viewed at  $t = 0$  is given by

$$\mathbb{E}_\rho[\mathcal{A}] = \text{Tr}\{\rho\mathcal{A}\}, \quad (136)$$

where the final quantum state is described by the density operator  $\rho = U|\psi\rangle\langle\psi|U^\dagger$ . Note that in this formulation  $\rho$  becomes associated with the probability measure of which Bao and Rebentrost [608] uses to propose the *fundamental theorem of quantum asset pricing* which is analogous to the first fundamental theorem of asset pricing based on arbitrage theory [181]. They also describe a path to formulate *quantum derivative* assets and equivalent pricing scheme [608].

Similar ideas of a quantum market have also been considered in [609] and stochastic finance based on quantum mechanics have been explored in [610]. As quantum technologies advance, we may see practical implementation and adoption of such defined quantum financial instruments<sup>32</sup>. However, the research in this field is still in its infancy, we can only speculate that if quantum assets gets practically demonstrated and adopted, they will most likely going to be a relatively small but significant part of the global financial economy similar to crypto-currencies.

<sup>32</sup> This may include quantum money described in Section (41).

## 32. Derivative Pricing

Derivatives are financial instruments whose value is derived from the performance of underlying assets, indexes, interest rates, or other financial variables [204]. Common examples of underlying assets include stocks, bonds, currencies, and commodities. Derivatives allow investors to hedge risks, speculate on price movements, or access assets and markets that might otherwise be difficult or costly to trade. Common types of derivatives include the following [204]:

1. **Options Contracts:** These contracts give the buyer the right (but not the obligation) to buy (call option) or sell (put option) an asset at a predetermined price before or at expiration.
2. **Credit Derivatives:** These are used to transfer credit risk. A common type is the Credit Default Swap (CDS), where the buyer pays a premium for protection against a credit event like a default.
3. **Swaps:** Contracts in which two parties agree to exchange streams of cash flows over a set period. For example, an interest rate swap involves exchanging fixed-rate interest payments for floating-rate ones.
4. **Futures Contracts:** Agreements to buy or sell an asset at a predetermined price at a specific time in the future.
5. **Forward Contracts:** Similar to futures but traded over-the-counter instead of on exchanges, providing more flexibility in terms and conditions.

Efficient pricing of derivative represents a significant opportunity of improvement in financial operations. The projected size of global derivatives market by 2033 is \$64.24 billion [611] which coincides with predicted time-lines for Universal Fault-Tolerant quantum computing [20]. This a growth from \$30.57 billion in 2024 at a CAGR of 8.6% during the forecast period from 2025 to 2033 [611]. According to Business Research Insights [611], the projected growth of the derivatives market is attributed to various factors which includes: (1) growing demand for price volatility, (2) increasing changes in demand and supply which influence product demand, (3) rapid industrialisation of many underdeveloped countries, and (4) rising technological developments in financial sectors of which quantum computing may play a significant role.

In this section, we will review some of the use-cases of quantum computing for derivative pricing. In particular, we will focus on the computational methods for pricing [options](#), [collateralised debt obligations](#), and [swap netting](#).

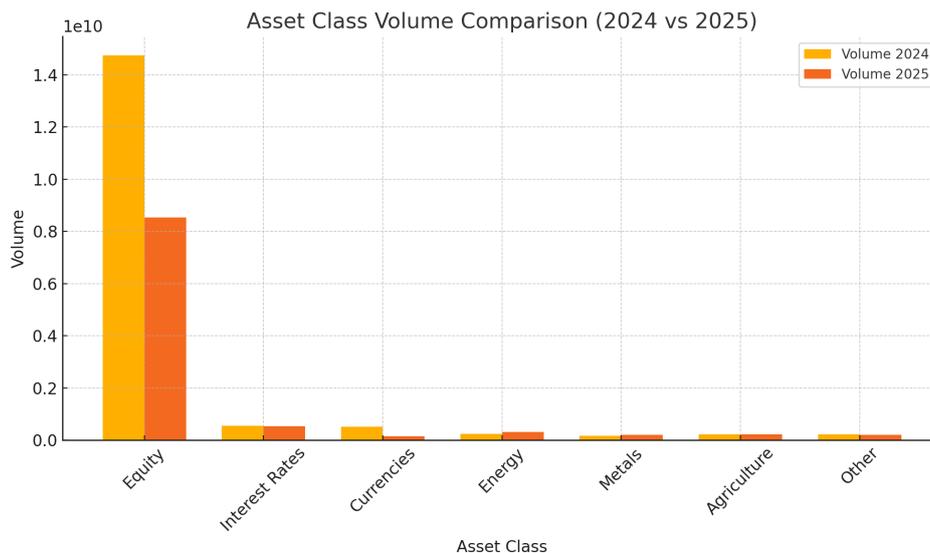
### 32.1. Option Pricing

Options are financial derivatives that provide the holder with the right, but not the obligation, to buy or sell an underlying asset at a predetermined price (the strike price) before or at the expiration date [601]. Option pricing is the process of determining the fair value of an option [612–614], which depends on various factors such as the underlying asset's price, volatility, time to expiration, and risk-free interest rate. The pricing of options is critical for investors and traders as it helps make informed decisions about hedging, speculation, and portfolio management [204].

According to the Futures Industry Association (FIA)<sup>33</sup> worldwide volume of Exchange-Traded Derivatives (ETDs)<sup>34</sup> has been on the order of 10 billion contracts in recent years. Figure 7 lists some of the most traded ETDs and their relative percentage volumes in 2025. Note that these data does not include the Moscow Exchange and its volume and open interest history [615]. This data is presented to underscore the potential opportunity space for quantum computing for efficient pricing of options contracts.

<sup>33</sup> FIA is a prominent global trade organisation that represents the interests of the futures, options, and derivatives markets, including futures commission merchants and principal traders.

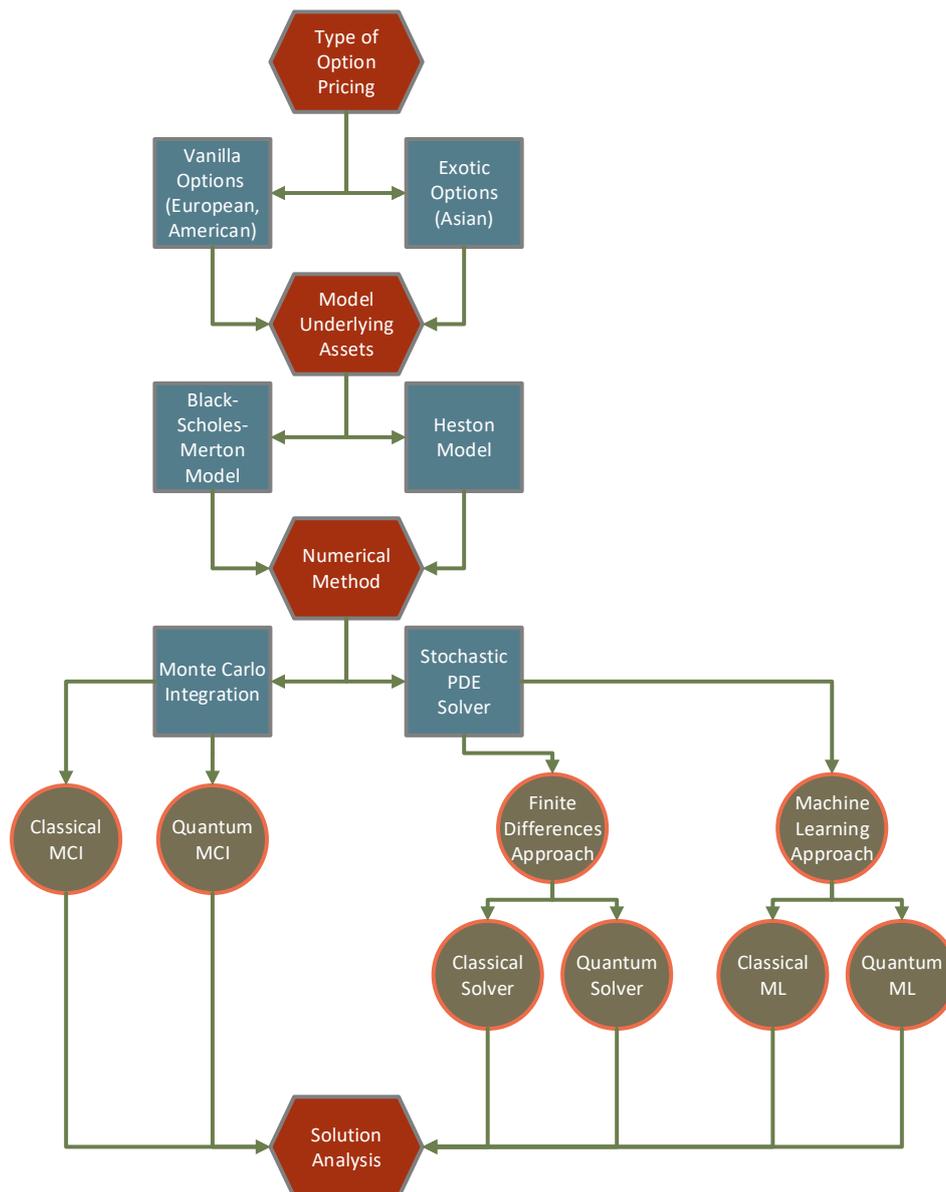
<sup>34</sup> An ETD is a financial contract that is listed and traded on a regulated exchange. In other words, these are derivatives that are traded in a regulated environment.



**Figure 7.** FIA ETD data of asset class trading volume of futures and options in January 2024 and 2025 [615]. Although, there was a decline of nearly 40% from 2024 to 2025, the grand total volume is still in the order of billions of trade per year of which more that 80% are based on underlying equity assets.

Options represent a versatile class of financial instruments, structured as contracts between a buyer and a seller. The buyer compensates the seller with a premium in exchange for specific contractual rights [204]. *Call options* confer upon the holder the right, but not the obligation, to purchase an underlying asset at a predetermined price within a specified period. Conversely, *put options* grant the holder the right to sell the asset under similar conditions. Each call option transaction involves a bullish buyer and a bearish seller, whereas put options feature a bearish buyer and a bullish seller [616]. Market participants engage in options trading for various strategic purposes. Speculative trading enables investors to leverage their positions in an asset at a lower capital outlay compared to direct ownership of the underlying security. Additionally, options serve as a risk management tool, allowing investors to hedge against portfolio volatility [204]. In some instances, options holders can generate income by purchasing call options or assuming the role of an options writer [617]. Furthermore, options provide a direct mechanism for investing in commodities such as oil.

For options traders, three critical metrics serve as key indicators for making informed investment decisions; fair value, daily trading volume and open interest [617]. It is with the efficient computing of fair value of an option that quantum computing can potentially make a difference. In this subsection, we will describe the mathematical formulation of the option pricing problem by following the workflow illustration in Figure 8. First, we describe some different [types of options](#) with their payoff. Second, we consider some traditional models for underlying assets such as the BSM and Heston model. Lastly, we outline some quantum algorithms for option pricing which includes QMCI and [quantum solvers](#) for Stochastic Differential Equation (SDE).



**Figure 8.** An illustrative workflow of option pricing which involves: (1) choosing the type of option, (2) selecting an appropriate model of the underlying assets, and (3) choosing a numerical method for the computation. This section will present the potential quantum advantage offered by quantum algorithms to make the numerical methods efficient.

### 32.1.1. Types of Options

Options, also known as contingent claims or non-linear derivatives, are financial instruments whose payoffs exhibit a non-linear relationship with the value of the underlying asset [181,204]. The options pricing problem generally fall into two classes: **Vanilla Asset Pricing** and **Exotic Asset Pricing**. The option value corresponds to the premium that the buyer must pay to acquire the rights conferred by the option contract. In derivative pricing, mathematical models establish the relationship between the underlying asset and the fair value of the option which makes both the option seller and buyer break even [181,204].

**Definition 4** (Vanilla Asset Pricing). Let  $v_t$  and  $S_t$  denote the value of the option and the underlying asset at time  $t$ , respectively. At the expiring date  $T$  of the option contract, or before under certain circumstances, the option holder receives a payment, referred as payoff given by

$$v_t = h(T, S_T) \quad (137)$$

where  $h$  is a non-linear function of  $T$  and value of underlying asset  $S_T$  at expiration.

#### European Option

on an underlying asset grants the holder the right, but not the obligation, to purchase or sell the asset at a predetermined future date, referred to as the expiry date ( $T$ ), and at a specified price, commonly known as the strike price ( $K$ ) [181,204]. The payoff for the option is given by [181]:

$$c_T = \max(S_T - K, 0) \quad \text{and} \quad p_T = \max(K - S_T, 0), \quad (138)$$

Where  $c_T$  and  $p_T$  are *call* and *put* values at maturity. For the call option, we say that [204]:

$$c_T \text{ is } \begin{cases} \text{in the money,} & \text{if } S_T > K \\ \text{out the money,} & \text{if } S_T < K \\ \text{at the money,} & \text{if } S_T = K \end{cases} \quad (139)$$

This similarly applies to the put option. Since the strike price  $K$  is fixed for the duration of the contract, the option pricing problem is to calculate the fair value of the contract at the pricing date  $t = 0$  given the expected value  $S_T$  at maturity under certain assumptions about the market [181].

**Definition 5** (General Asset Pricing). *Let  $v_t$  and  $S_t$  denote the value of the option and the underlying asset at time  $t$ , respectively. At any-time  $0 < t < T$ , the option value is given by*

$$v_t = V(t, S_t) \quad (140)$$

where  $V$  is a non-linear function of  $t$  and value of underlying asset  $S_t$ .

#### American Option

on an underlying asset grants the holder the right, but not the obligation, to exercise the buy (sell) at the asset at a strike price  $K$  at any time  $t_e \in [0, T]$  and receive a payoff of [601]:

$$c_{t_e} = \max(S_{t_e} - K, 0) \quad \text{or} \quad p_{t_e} = \max(K - S_{t_e}, 0), \quad (141)$$

for a call and put option, respectively. Other exotic options include the Asian option, where the payoff is based on the average value of the underlying asset over specific discrete times or within a defined period [181,204,601].

#### 32.1.2. Models for Underlying Assets

Derivative pricing largely depends chosen stochastic model for the underlying asset which takes into account many dynamic factors that affects the price of the asset [181]. In this report, we give the mathematical description for two traditional models: Black-Scholes-Merton (BSM) model [84,85] and Heston model [202]. These models computes the value of the option price  $v_t = V(t, S_t)$  at time  $t$  based on the value of the underlying asset  $S_t$  at that time. They both assume that  $S_t$  follows a Geometric Brownian Motion (GBM) with a probability measure  $P$  defined in the filtered probability space [618]. These give rise to SDEs given in Table 5 where the variable are defined to be [84,85,202]:

$S_t$ – underlying asset value	$v_t$ – instantaneous variance	(142)
$\mu$ – drift parameter	$\kappa$ – mean reversion speed	
$\sigma$ – asset volatility	$\chi$ – volatility of the volatility	
$\theta$ – long-term variance	$dW_t^{S,v}$ – increment of Wiener/Brownian motion	

**Table 5.** Traditional models: BSM and Heston for stochastic dynamic of the value of the underlying asset  $S_t$ .

	BSM	Heston
Assumption	constant volatility $\sigma$	stochastic volatility $\sqrt{v_t}$
SDE	$dS_t = \mu S_t dt + \sigma S_t dW_t^S$	$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S$ $dv_t = \kappa(\theta - v_t)dt + \chi \sqrt{v_t} S_t dW_t^v$

For the Heston model, the Brownian motions  $W_t^S$  and  $W_t^v$  are correlated as [202]

$$dW_t^S dW_t^v = \rho dt \quad (143)$$

where  $\rho$  is the correlation coefficient that captures the dependency between asset value and variance. Since these models must be arbitrage-free, thus one has to ensure that the discounted prices satisfy the Martingale property under the risk neutral probability measure  $Q$  [181,204]. Combining the martingale property with Itô's lemma<sup>35</sup>, we obtain the *vanilla European option* value  $v_t$  as a conditional expectation given by [204,618]:

$$v_t = V(t, S_t) = e^{-r(T-t)} \mathbb{E}_Q[h(T, S_T) | \mathcal{F}_t], \quad (144)$$

where  $r$  is the risk-free/discounting rate that replaces the drift  $\mu$  in Table 5 and  $\mathcal{F}_t$  is the filtration which represents the market information assumed to be known until time  $t$ . Similarly, the *American option* is given by the conditional expectation [204,618]:

$$v_t = V(t, S_t) = \max_{\tau \in [t, T]} e^{-r(T-\tau)} \mathbb{E}_Q[h(\tau, S_\tau) | \mathcal{F}_t], \quad (145)$$

where  $\tau$  denotes the stopping time when the option is exercised.

These models are typically solved ( $v_t$  computed) using numerical methods and in this review we will focus on QMCI techniques and quantum solvers for SDEs described in the ensuing subsections.

### 32.1.3. Quantum Monte Carlo Integration

A very common approach to computing the expectation of Equations (144) and (145) is to rewrite them as an integral and employ the classical MCI. In particular, the risk-neutral valuation of Equation (144) is given by [181,618]:

$$V = e^{-r(T-t)} \int_{\mathbb{R}} h(T, y) g(y | \mathcal{F}_t) dy, \quad (146)$$

where  $g(\cdot)$  is the conditional probability density function of the underlying asset value  $S_t$ . In general, the problem of evaluating Equation (146) can be reduced into computing the discretised integral expectation of the form [618]:

$$\mathbb{E}[f(x)] = \int_{\Omega} p(x) f(x) dx \approx \sum_i p(x_i) f(x_i), \quad (147)$$

where  $f(x)$  and  $p(x)$  are the general payoff functions and conditional probability distribution, respectively. The random variable  $x \in X$  (corresponding to the asset value  $S_t$ ) is sampled from a space  $X \subset \Omega$  that could be multi-dimensional.

Since Equation (147) is the same as Equation (28), thus we can employ the Quantum Monte Carlo Integration described in Algorithm (7) on a quantum computer. Note that we can rescale the function  $f$  to output values between  $[0, 1]$  as required by qubit systems, and by linearity of expectation values we can rescale back at the end of the computation [31].

<sup>35</sup> Itô's lemma or Itô's formula is an identity used in stochastic calculus to find the differential of a time-dependent function of a stochastic process. It serves as the stochastic calculus counterpart of the chain rule.

The pricing vanilla European options is one of the most straightforward applications implemented on quantum hardware. In the literature, they are often regarded as an initial benchmark or, more precisely, a proof of concept demonstrations on quantum computers. Examples include Refs. [619–622] where implementations on either quantum simulators or hardware were made. Recent approaches include the QMCI engine [193] which is a modular and extendable, and general quantum algorithm for numerical integration [69].

#### 32.1.4. Stochastic PDE

An alternative approach is to reformulate Equations (144) and (145) into PDEs using the Feynman–Kac formula [207] as shown in Section (16). For example, the value of the European option in the BSM model can be written as [31]:

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0, \quad (148)$$

where  $V(t, S_t) \in (0, T) \times \mathbb{R}^+$  with the terminal condition

$$V(T, S) = h(T, S). \quad (149)$$

Hence the PDE of the form of Equation (148) can be solved by using quantum algorithms as discussed in Section (16). In this review

#### Finite Difference

A widely used method for solving PDEs is the *finite differences* methods described in Algorithm (8). The above BSM example of solving Equation (148) requires defining bounds for underlying asset and imposing boundary conditions depending on specific payoff. For instance, one can choose the bounds  $S_t \in [0, S_\infty]$  where  $S_\infty = 4K$  and boundary conditions [31]:

$$V_{j,I+1} = S_\infty - e^{-r(T-t_j)} K, \quad \text{where } V_{j,0} = 0. \quad (150)$$

Following Algorithm (8), the problem boils down to solving a  $N = I + 1$  dimensional linear system at each discretised time-point  $t_j$  given by [31]:

$$AV_j = b_j, \quad \text{where } V_j, b_j \in \mathbb{R}^N \quad \text{and} \quad A \in \mathbb{R}^{N \times N}. \quad (151)$$

Here, the mesh nodes are  $(t_j, S_i) = (j\Delta t, i\Delta S) = \left(j \cdot \frac{T}{(J+1)}, i \cdot \frac{S_\infty}{(I+1)}\right)$  for  $i, j \in [0, I + 1], [0, J + 1]$ . The user chosen natural numbers  $J > 1$  and  $I > 1$  define the mesh-size. Generally the finer the mesh-size, the more accurate the calculation but this also increases the computational cost since it results in a high-dimensional linear systems of equations. This is where quantum computing can potentially make the computational pipeline more efficient. By employing the Quantum Linear Systems Solver (QLSS), the theoretical speed-up can be up to exponential in terms of the size of the linear system  $N$  [6,131]. However, as discussed in Section (11) the QLSS has challenges of efficient data-loading on and read-out from a quantum computer which might limit its Practical Quantum Advantage [11].

#### Hamiltonian Simulation

Another approach is to reformulate Equation (148) into a Schrodinger or Schrodinger-like equation as discussed in Section (16). These methods perform a carefully chosen change of variables that transform the problem into:

- Performing non-unitary Hamiltonian simulation using embedding techniques [217].
- Performing imaginary-time evolution of the heat equation [219,614].

The advantage of these methods is that they can circumvent the mesh discretisation issues of finite difference methods, whilst also having the same potential exponential quantum speed-up due to

quantum Hamiltonian simulation [31,48]. The drawback is that the quantum resource estimates are so high for Universal Fault-Tolerant implementations [48], however the use of quantum analogue system has shown promising results [216,219].

### Quantum Machine Learning

Another interesting approach is to reformulation the problem of solving linear PDEs of the form of Equation (148) into a learning problem that can be solved using deep learning techniques [623,624]. This can be translated to variational QML technique for solving PDEs [625]. However, unlike the previous methods this approach does not have known theoretical quantum speed-up and may have empirically determined efficiency just like other VQA methods discussed in Section (12).

### 32.2. Collateralised Debt Obligations

A Collateralised Debt Obligation is a complex financial product backed by a pool of loans and other assets and sold to institutional investors [626]. The utility of the tool is to free up capital and shift risk. A typical CDO pool is divided into three categories, otherwise called tranches [626]: junior, mezzanine, and senior. The tranches of CDOs are named to reflect their risk profiles. The senior tranches are generally safest because they have the first claim on the collateral, but they offer the lowest interest rates, with the opposite being true for junior tranches. Although the senior tranche is protected from loss, other default events can cause the CDO to collapse, such as events that caused the 2008 financial crisis [627].

#### Mathematical Formulation

The goal is to compute a fair price of a CDO where the return is consistent with the expected loss for investors of each tranche (denoted by  $k$ ). Following the notation in [627], let  $L$  denotes the total loss for the portfolio and  $L_k$  denote the loss suffered by the holders of tranche  $k$ . Denote the lower and upper attachment point for tranche  $k$  by  $K_{L_k}$  and  $K_{U_k}$ , respectively. When defaults occur, the buyer of the tranche  $k$  will bear the loss in excess of  $K_{L_k}$ , and up to  $K_{U_k} - K_{L_k}$ . It follows that

$$L_k = \min[K_{U_k} - K_{L_k}, \max(0, L - K_{L_k})] = h(L) \quad (152)$$

where  $h(\cdot)$  is a non-linear function of  $L$  similar to the payoff function in Equation (137) for the vanilla option pricing. Investors care about the fair spread  $r_k$  of each tranche given by [626,627]:

$$r_k = \frac{1}{N_k} \mathbb{E}[L_k = h(L)], \quad (153)$$

where  $N_k = K_{U_k} - K_{L_k}$  is the notional value of the tranche  $k$  of the portfolio. Since a fair price of a CDO is consistent with the expected losses at each tranche, the fair spread  $r_k$  is considered the expected return for a chosen tranche.

#### Models for Total Loss

Analogous to the option pricing problem where models are based on Geometric Brownian Motion for the stochastic value of the underlying asset  $S_t$ , the total loss  $L$  for a CDO is computed based on the *Conditional Independence Approach* [628]. Conditional independence models [628] are often used for estimating the chances of parties defaulting on the credit in the CDO pool. In such models, given the systemic risk  $z$ , the default risks  $\{X_1, \dots, X_n\}$  of the assets involved are independent.  $z$  is then used as a latent variable to introduce correlations between the default risks. The distributions of the default risks and the systemic risk are usually either Gaussian [628] or normal-inverse Gaussian [629]. In particular, for a CDO comprising of  $n$  assets with correlated default events, the total expected loss is [626,627]:

$$\mathbb{E}[L] = \int_{-\infty}^{\infty} \sum_{i=1}^n \lambda_i p_i(z) f(z) dz, \quad (154)$$

where  $\lambda_i$  is the loss that would incur for asset  $i$  with probability  $p_i(z)$  under systematic risk level  $z$  with probability density function  $f(z)$ . In practice one integrates  $z$  from  $-3\sigma$  to  $+3\sigma$  which covers 99.73% of a Gaussian distribution with a variance  $\sigma$  [627]. The expected loss in Equation (154) can be then be used to compute the fair spread in Equation (153).

### Quantum Approach

The MCI is generally the most preferred method for computing expectations of the form in Equation (154) since its scaling does not depend on the dimension of the problem but the number of samples used. As noted on Section (15), the QMCI can potentially have a quadratic speed up of this computation [5,193]. A small scale proof-of-concept implementation of CDO on IBM quantum computers was performed in Ref. [627]. As quantum computers scale up, it is possible that some improved versions of QMCI may deliver Practical Quantum Advantage.

### 32.3. Swap Netting

In general, a swap is a derivative contract between two parties who exchange sequences of monetary flows for a set period of time [630]. One of the most common types of swaps is the interest rate swap, where institutions exchange fixed and floating interest payments to either hedge against interest rate fluctuations or take advantage of market movements [630,631]. Usually, at the time the contract is initiated, at least one of these flows is determined by a random or uncertain variable, such as an interest rate, foreign exchange rate, equity price or commodity price. Swap netting involves netting the present values of cash flows that two parties owe each other under swap contracts, so that only the net amount is exchanged. There are two main categories of swaps [632]:

1. Settlement netting – netting the periodic cash flows due on the same date.
2. Close-out netting – netting all current and future obligations in the event of a default.

Swap netting is a method of reducing credit, settlement and other risks of financial contracts by reducing them into a net obligation [630]. When a financial institution has thousands to millions of swaps with multiple counterparties, calculating net exposures across all these positions can be computationally demanding. This is because each swap may have different tenors (lengths), currencies, payment schedules, interest rate indices, optionality (e.g., callable swaps), etc [630]. This leads to high-dimensional cash flow matrices represented as a complex graph problem [633].

### Mathematical Formulation

Let the  $\mathcal{S}$  be the set of swaps for a given counterparty with the clearing house. We can introduce a partition  $\mathcal{P}$  on  $\mathcal{S}$  such that [634]:

$$\mathcal{S} = \cup_{k=1} M_k, \quad \text{where } M_k \in \mathcal{P} \subset \mathcal{S}. \quad (155)$$

The elements of each  $M_k$  will be swaps that are similar enough to be netted with each other. In a fine-grained partition these may be swaps with identical economic terms. However, the general problem is to consider partitioning into a smaller number of larger subsets in order to select nettable groups within them [634]. Assuming there are  $n$  swaps in each  $M_k$ , we denote notional of swap  $i \in \{1, 2, \dots, n\}$  by  $\mathcal{N}_i$  and define the direction  $d_i$  such that

$$d_i = \begin{cases} +1 & \text{if the clearing house pays the counterparty} \\ -1 & \text{if the counterparty the pays clearing house} \end{cases} \quad (156)$$

where a fixed cash flow payment is between the two parties. The objective is to select subsets of  $M_k$  with a maximum notional that can be netted, such that within each subset, the fixed legs and float legs are similar to each other. This is an optimisation problem that is solved by numerical methods.

## Quantum Approach

Mapping the problem of swap netting to a quantum computer was first proposed in Ref. [634]. The authors define a binary decision variable  $x_i$  for each swap  $i$  which has value is one if the corresponding swap is in the chosen group of swaps to be netted and zero otherwise. The optimisation problem for choosing the swaps to be netted can be formulated as a QUBO given by [634]:

$$x = \max_{x_i \in \mathbb{B}^N} \left[ \sum_{i \in M} x_i \mathcal{N}_i - \alpha \left( \sum_{i \in M} x_i d_i \mathcal{N}_i \right)^2 - \beta \sum_{i,j \in M} x_i x_j I_{ij} \right]. \quad (157)$$

The first term aims to maximise the total notional value of swaps within each group. The second term introduces a penalty for groups whose notional amounts deviate significantly from offsetting each other. The third term applies a penalty based on the incompatibility between pairs of swaps. This pairwise incompatibility, denoted as  $I_{ij}$ , must be precomputed for every swap pair and is non-negative. The incompatibility function can also be interpreted as a way to prioritise different attributes (like interest rate, duration, etc.) and can be weighted according to the priorities set by the clearing house [634]. The constants  $\alpha$  and  $\beta$  control the relative weighting of these terms and can be set either manually using theoretical insights or calculated using methods such as sub-gradient descent. This process, defined for the general component  $M$ , is applied to each  $M_k$  within the partition  $\mathcal{P}$  to generate netting proposals for the group and then aggregating all the outputs for each subset  $k$  into a final netting for the full set  $\mathcal{S}$ .

As noted in [634], similar but non-identical swaps may not be fully netted due to factors like coupon blending, which can produce interest rates diverging from market values. Furthermore, different partitioning strategies and tuning of constants  $\alpha$  and  $\beta$  can yield multiple netting proposals. The QUBO of Equation (157) can be solved using quantum algorithms described in Chapter (17) on a quantum annealer which may also generate various solutions for a subset, allowing selection of the proposal that best aligns with market conditions.

## 33. Investment Optimisation

### 33.1. Portfolio Optimisation

Portfolio optimisation is one of the most common optimisation problems in fintech. It is the process of choosing and weighting assets to hold in a portfolio such that some given objective is achieved. This problem has a huge amount of flexibility, but the canonical model was proposed in 1952 by Markowitz [228], wherein the most common objectives are to maximise returns for a given level of risk, or vice versa. This trade-off between risk and return is due to the asset volatility: a volatile asset can increase in price dramatically, providing a high return, or it may drop precipitously, losing money. The trade-off is characterised by a curve known as the *efficient frontier* [228].

The above is of course a simplification, and in the decades since the Markowitz model was proposed, a vast number of additional variables and constraints have been taken into account, including transaction costs, dynamic optimisation (where the portfolio is rebalanced regularly), liquidity, time to maturity, asset class (e.g. stocks, bonds, futures, options), regulatory and compliance requirements, and so on [635]. The resulting optimisation problem can be of staggering complexity, and very difficult to solve using traditional computational methods.

### Mathematical Formulation

We consider a simplified portfolio optimisation problem to illustrate the formulation and solution approach. Given a pool of assets, with some expected returns  $\mu_i$  (mean of historical returns) and risk  $\sigma_i$  (variance of historical returns), combine them to either maximise return for some level of risk, or minimise risk for some level of return, creating a optimised static portfolio<sup>36</sup>. There are now two

<sup>36</sup> Static means we ignore the possibility of time-varying allocations.

main approaches [636]: optimising weightings, where the problem is to pick the right amount (either continuous or discrete) of each asset, or optimising inclusion, where we specify an amount of each asset and simply choose whether to include it or not. In the case of maximising return for a given level of risk  $R$  by optimising the weightings  $x_i$ , the optimisation problem would be [636]:

$$\max \sum_i \mu_i x_i \quad \text{such that} \quad \sum_i \sigma_i x_i = R. \quad (158)$$

While in the case of minimising risk for a given level of return  $P$  by optimising inclusion, it would be [636]:

$$\min \sum_i \sigma_i w_i b_i \quad \text{such that} \quad \sum_i \mu_i w_i b_i = P, \quad (159)$$

where  $w_i$  is the amount of each asset, and  $b_i$  is the binary variable determining whether or not it is included in the portfolio. Instead of just choosing or weighting assets, it is also possible to choose whether to hold long or short positions, in a technique known as long-short minimum risk parity optimisation [637]. Briefly, this is where the decision to go long or short (buy or sell) assets is optimised in the interest of minimising risk. This is prioritised over maximising return as the shorting strategy can lead to theoretically unlimited losses, which must be mitigated against.

Depending on the formulation, there are numerous classical techniques for solving this problem which includes convex [228] or mixed-integer programming [638,639], meta-heuristic methods [640], principal component-based methods [641], or genetic algorithms [642]. These methods have varying degrees of efficiency and accuracy, but all face computational bottlenecks as optimisation problems with many variables and constraints [229]. Hence, quantum computing has a potential to offer efficient computation of approximate solutions, higher quality solutions, or some mix of both [236].

### Quantum Approach

There are various quantum algorithms, discussed in Chapter (17), that can be used to solve this problem. The constrained optimisation problems above can be reformulated as a QUBO, allowing the application of Quantum Annealing [643] and variational gate-based methods [644]. Such approaches can also account for additional constraints [645,646] via, for example, using mixing operators (see Section 18.4). Other approaches utilise an adaptation of Grover's algorithm [263], exhibiting the expected quadratic speedup over analogous classical techniques, while yet more [647,648] formulate the problem as dynamic in time, making decisions for the same portfolio as it evolves in time.

Quantum walks may also be used. There has been a proposition in [646] where an algorithm based on a Continuous-time Quantum Walk (CTQW) structure is used in the context of portfolio optimisation. This Quantum Walk Optimisation Algorithm (QWOA) uses a CTQW Hamiltonian defined by a graph adjacency matrix [88,649] to evolve the system. The advantages of QWOA for the portfolio optimisation problem lie in its flexibility in 'connecting' only the solutions in the valid subspace, the ability to eliminate degenerate portfolio states (thus significantly reducing the search space), and complete global symmetry amongst valid solutions (eliminating the bias of one valid solution over another due to mixing asymmetry) [646]. This latter is compared positively to approaches like QAOA in [650], which retain this bias in the mixing operator over non-trivial feasible solutions. The results from numerical simulations show that the QWOA for portfolio optimisation leads to an improved performance by reducing the asset search space by a significant factor while also showcasing a higher convergence rate and solution quality [646].

### 33.2. Hedging

A special case of portfolio optimisation is *hedging*, which is a strategy that seeks to reduce risk in a portfolio by ensuring that changes in the value of some assets are countered by opposite changes in the values of other assets [651]. The most common method of hedging is through derivatives, which are securities whose value depends on some function of the values of one or more underlying assets (see

Section (32). Hedging strategies are fundamental in financial risk management, providing mechanisms to mitigate exposure to adverse market movements [652].

#### Mathematical Formulation

Consider a portfolio with value  $V$  exposed to market risk. The investor has access to a set of hedging instruments with return vector  $\mathbf{r} = (r_1, r_2, \dots, r_n)$  and wishes to determine the position vector  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  to minimise risk. A classical variance-based formulation of the hedging problem is [228,653]:

$$\min_{\mathbf{w}} \text{Var}(V + \mathbf{w}^\top \mathbf{r}). \quad (160)$$

Assuming the return vector  $\mathbf{r}$  has covariance matrix  $\Sigma$ , the problem simplifies to:

$$\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^\top \mathbf{p} = C, \quad (161)$$

which are linear budget constraints. Here,  $\mathbf{p}$  is the price vector of the hedging instruments and  $C$  denotes the total capital allocated for hedging. This is a combinatorial optimisation problem with a quadratic risk term determined by the covariance matrix  $\Sigma$  which is computed using historical pricing information [652].

#### Quantum Approach

To solve the hedging problem on a quantum computer, the general approach is to reformulate it as a QUBO by first discretising the continuous decision variables  $\mathbf{w}$  [654]. Let each  $w_i$  be expressed in binary form as:

$$w_i = \sum_{k=1}^K 2^{k-1} x_{ik}, \quad \text{where} \quad x_{ik} \in \{0, 1\}. \quad (162)$$

The variance minimisation objective becomes [654]:

$$\min_{\mathbf{x} \in \{0,1\}} \mathbf{x}^\top T^\top \Sigma T \mathbf{x}, \quad (163)$$

where  $T$  is a transformation matrix mapping binary vectors  $\mathbf{x}$  to real-valued weight vectors  $\mathbf{w}$ . We can incorporate the budget constraint within the QUBO framework by introducing a penalty term:  $\lambda(\mathbf{p}^\top T \mathbf{x} - C)^2$ , where  $\lambda$  is a Lagrange multiplier regulating the constraint's influence. The final QUBO formulation is [654]:

$$\min_{\mathbf{x} \in \{0,1\}} \mathbf{x}^\top T^\top \Sigma T \mathbf{x} + \lambda(\mathbf{p}^\top T \mathbf{x} - C)^2. \quad (164)$$

There are various quantum algorithms that can be used to solve (164), as discussed in Chapter (17). One approach, presented in [654], is to map the solution to finding a maximum-independent set in a specially constructed graph. In this graph, the vertices represent assets with significant correlations amongst them being represented by the connecting edges. In order to solve the problem, the authors use two methods. In the first instance, the asset exhibiting minimal risk is picked by finding a maximum-independent set in the above graph in similar terms to Equation (164). The second method proposes a modified solution to the optimisation problem through a graph colouring formulation [654], finding a colouring minimising the edge weights connecting same-colour nodes.

#### 33.3. Settlement

In finance, settlement (or transaction settlement) is the process by which a set of securities (be it assets, bonds or derivatives) are exchanged for a form of compensation [655]. The problem is to settle as many transactions as possible in a complex graph of transactions, using the minimum amount of resources (such as time, number of transactions, individual communications, and so on) [656]. It is a challenging problem due to the constraints posed by legal requirements, as well as the complexity added by further parameters that need to be taken into account (i.e., collateralised assets or credit

information). However, an optimal solution results in the least expenditure of resources, which, if performed at a large organisation like a central bank, can be a significant saving.

### Mathematical Formulation

For the settlement optimisation problem, one needs to maximise the weighted sum  $S$  of settled transactions [657]:

$$S = \max_{\vec{x}} \sum_{i=1}^T w_i x_i, \quad (165)$$

where  $T$  is a set of transactions,  $x_i$  is a binary variable indicating that the transaction is being settled, and  $w_i$  is the weight for each transaction  $i$  that reflects the total monetary value of the transaction. If, instead of maximising the total value of transactions settled, we wish to settle the total *number* of transactions, we can set  $w_i = 1 \forall i$ . In the first instance, the optimisation of equation (165) is subject to the inequality constraint

$$\vec{b}_p + \sum_{i \in \Gamma_p} x_i \vec{v}_{ip} \geq \vec{l}_p, \quad \forall p, \quad (166)$$

where  $P$  is the number of parties that take part in the  $T$  transactions, the  $i$ th transaction involves a party  $p$  and is described by the vector  $\vec{v}_{ip}$ ,  $\Gamma_p$  is the set of transactions of party  $p$  and  $\vec{b}_p$  is a balance vector, which encodes the securities and currencies that party  $p$  owns before any transactions settle [657].

In practise, the inequality constraints of equation (166) can be transformed to equality constraints by defining the settlement optimisation as [658]:

$$S = \max_{\vec{x}} \sum_{i=1}^T w_i x_i, \quad \text{subject to} \quad \vec{b}_p + \sum_{i \in \Gamma_p} x_i \vec{v}_{ip} = \vec{l}_p + \vec{s}_p, \quad \forall p, \quad (167)$$

where  $\vec{s}_p \in \mathbb{R}_{\geq 0}$  is a vector including slack variables.

### Quantum Approach

It has been shown [658] that well-known quantum methods, such as VQE or QAOA, can be extended to solve problems that reside within the Mixed Binary Optimisation (MBO) class, which combines binary and continuous variables. The work proposes hybrid quantum-classical heuristics to address the problem of settlement optimisation, allowing the modelling of the constraints and options described above.

The constrained optimisation problem in equation (167) can be mapped to a QUBO by transforming the equality constraint into a quadratic penalty term scaled by a penalty coefficient  $\lambda > 0$  [658]:

$$\max_{\vec{s}_p \geq 0, \vec{x}} \left[ \sum_{i=1}^T w_i x_i - \lambda \sum_{p=1}^P \left( \vec{b}_p + \sum_{i \in \Gamma_p} x_i \vec{v}_{ip} - \vec{l}_p - \vec{s}_p \right)^2 \right]. \quad (168)$$

In this final form, one can solve the transaction settlement optimisation problem using any of the QUBO-solving techniques mentioned above, such as Quantum Annealing or QAOA. For example, the aforementioned study, as well as a second [659], approach the problem using QAOA, and are able to find the optimal solution for the problems under consideration by running on real QPUs. However, as in many of the other use-cases in this paper, constraints include noise and limitations on qubit number and circuit depth, which prevent competitive comparisons with state-of-the-art classical solvers on real-world problems.

#### 33.4. Arbitrage

Arbitrage refers to the exploitation of market inefficiencies such as trading the same asset in different markets in order to take advantage of different prices [660]. Such assets might include securities, currencies, or commodities. Arbitrage tends to be more speed-reliant than other trading

strategies: as the act of trading influences prices, being faster than competitors to identify and exploit an opportunity leads to much better returns [661]. For example, an FCA report in 2020 highlighted that races between different firms to exploit an arbitrage opportunity can be over in a matter of tens of milliseconds [662].

In general, arbitrage can be far more complex, involving multiple interconvertible assets across multiple markets, potentially over an extended period, and allowing for complications such as fees, regulatory requirements, and liquidity [663]. Identifying optimal strategies faster than competitors is a crucial and high-impact problem, as the very act of exploiting an opportunity causes it to lessen. Apart from the potential profitability, an accurate and rapid analysis of market distortions causing arbitrage opportunities can also provide a significant amount of information about other potential opportunities.

### Mathematical Formulation

One formulation of the problem of finding an optimal arbitrage strategy is described in [664] and summarised below. We have a set of assets represented as nodes of a graph. There are a number of directed edges between the nodes, representing potential conversions, with edge weights  $\log c_{ij}$ , where  $c_{ij}$  represents how many units of asset  $j$  a unit of asset  $i$  can be converted to.

To find the most profitable arbitrage opportunity, we want to maximise the product of the conversion rates in the cycle. We define  $x_{ij}$  as the binary variable for including the  $ij$  edge in the final cycle. Thus, the profit of a given cycle can be written as  $\sum x_{ij} \log c_{ij}$ , ignoring constants, where the logarithm accounts for converting a product to a sum. For conceptual simplicity, we will consider only cyclical strategies, where we start and end with the same asset. To enforce this, we need two constraints: that all nodes have the same number of entrances as exits, and that all nodes are passed through exactly once, represented respectively as [664]:

$$\sum_{(i,j) \in E} x_{ij} = \sum_{(j,i) \in E} x_{ji} \quad \text{and} \quad \sum_{(i,j) \in E} x_{ij} \leq 1, \quad \forall i \in V. \quad (169)$$

From here, it is also easy to add a risk constraint by adding a risk penalty for each conversion, to limit cycle length, or to enforce certain transaction structures.

There exist classical algorithms (such as Bellman-Ford's algorithm [665] and Floyd-Warshall's algorithm [666]) which run in polynomial time and return the first viable cycle they find, but the problem of finding the *optimal* cycle is NP-hard [667]. As mentioned above, arbitrage is a particularly competitive strategy, and so successfully implementing an algorithm to reduce the time taken would be a significant prize.

### Quantum Approach

In a graph-based formulation, the problem lends itself to being converted to a QUBO and solved using any of the aforementioned quantum methods such as variational (QAOA/VQEs) or annealing methods. For example, one study [668] formulated the objective function and penalties as a QUBO, and solved using both QAnn and simulated QAOA. They found that the QAnn formulation returned the optimal result, while QAOA often did not return feasible solutions to the problem. Like many other current approaches, the feasibility and advantage of a quantum approach will become more visible as hardware improved, but promising results in proof-of-concept studies like the above suggest this is a worthwhile field of study.

However, it is important to note that such quantum approaches are unlikely to be suitable for the most rapid trades or short-lived opportunities in liquid markets such as oil. Calls to quantum hardware take tens of milliseconds at the very least, stretching to hours if queuing times are included [669] which is a significant speed disadvantage compared to state-of-the-art classical methods. As mentioned before, this is one field where the time to solution is vital, meaning such latency is fatal. While it is possible that as quantum hardware improves and becomes more widespread, this speed disadvantage will decrease, the greater potential of quantum computing for arbitrage is for slower trades in less

liquid markets, and the aforementioned ability to glean information about market distortions from the results.

### 34. Natural Language Processing

Natural Language Processing (NLP) is now widely used in many sectors, especially those driven by large volumes of unstructured textual data (e.g., press releases, regulatory filings, social media sentiment or financial news) [670]. Its uses range from short-term trading algorithms to option pricing models (see Sections 31 and 32). NLP aims to interpret and extract key text information, enabling better understanding and generation of human language.

#### Mathematical Formulation

Natural language can be modelled as a sequence of tokens  $X = (w_1, w_2, \dots, w_n)$  from a vocabulary  $V$ , each token  $w_i \in V$  is represented as a vector in  $\mathbb{R}^d$  via an embedding function  $E : V \rightarrow \mathbb{R}^d$ . Distributional semantics states that words appearing in similar contexts have similar meanings [671], which embedding methods implement by mapping such words to nearby vectors [672], which are often compared using measures like the dot product  $w_i^\top w_j$  [673]. For example *word2vec* [674–676] training optimises word vectors so that the inner product  $\langle E(w), E(w') \rangle$  is high if  $w$  and  $w'$  share contextual neighbours [677]. Thus, semantic similarity is quantified geometrically (e.g. via cosine similarity) in the embedding space. Embeddings can be static (each word has a single vector) or contextual. In contextual embeddings, introduced by transformer-based models like Bidirectional Encoder Representations from Transformers (BERT) [678] or Generative Pre-trained Transformers (GPT) [679], the representation of a word depends on its surrounding words. A contextual embedding is a function  $F$  mapping the whole sequence  $X$  to context-aware vectors  $(h_1, \dots, h_n)$ , where  $h_i = F_i(X) \in \mathbb{R}^d$  encodes the meaning of  $w_i$  in context. These embeddings capture both semantics and syntax by encoding word order and co-occurrence patterns from large collections of text data.

Syntactic structure is often modelled by formal grammars or parse trees. In a parse tree  $T$  for sentence  $X$ , words are leaves and nonterminal nodes represent phrase constructs (such as noun phrases, verb phrases) with a root of type 'sentence'. Transformers achieve this through self-attention mechanisms: given token  $w_i$  with an associated query vector  $q_i$  and other tokens  $w_j$  with keys  $k_j$ , the attention weight from  $w_i$  to  $w_j$  is  $\text{Softmax}(q_i \cdot k_j / \sqrt{d_k})$  [677]. This attention weight determines how much information from token  $j$ 's value vector contributes to  $w_i$ 's new representation.

Mathematically, a transformer layer updates the sequence representation by a function,

$$h_i^l = \sum_{j=1}^n \alpha_{ij} (W_V h_j), \quad (170)$$

where  $\alpha_{ij} = \text{Softmax}_j((W_Q h_i) \cdot (W_K h_j) / \sqrt{d_k})$  are attention weights and  $W_Q, W_K, W_V$  are learned projection matrices. Through layered feed-forward networks, transformers produce rich contextual embeddings that encode semantic relations and syntactic structure [677].

#### Classical Methods

The process of NLP has undergone significant developments due to the advancements in the area of deep learning [680]. Deep neural networks can be used to create complex language models [681–683] that benefit tasks such as, such as market sentiment analysis, event-driven forecasting, and credit risk assessment [684]. However, large models (including RNNs, and GPTs) demand substantial computational resources to develop and train [685,686], resulting in high energy usage and limited scalability. Hence, significant challenges remain in terms of bias, energy consumption, computational resources or environmental and societal impact [687].

## Quantum Approach

One way to construct a model for NLP is by combining meaning and grammar into a single language model (which can then be represented in a diagrammatic framework<sup>37</sup> [689]), known as Distributional Compositional Categorical (DisCoCat). It has also been shown that this very framework can be used by categorical quantum mechanics [690,691] and can be represented by quantum circuits utilising ZX-calculus [692]. This makes the model 'quantum-native', meaning it can naturally be run on a quantum computer via quantum circuits. In [672], the grammatical structure is explicitly encoded into a quantum circuit provide a structured way to interpret how word meanings combine to form sentence level sentiment. This, a difference with other large language models, which learn these relationships implicitly from vast amounts of data.

Alternative approaches to NLP arise from QNNs and enhancing the capacity of current NLP models' neural architectures [346,606,693]. QNN-based approaches replace parts of existing deep learning architectures with quantum subroutines, offering a potential solution to some of NLP's scalability and efficiency issues. This intersection is often referred to as Quantum Natural Language Processing (QNLP) [690]. Experiments have been carried out in NISQ devices for a variety of simple NLP tasks [677,694–699], such as topic classification using single-qubit rotations and entangling gates or encoding Word2Vec embeddings [674] in a QRAM with a 'modest' number of qubits. These experiments suggest QNLP can in principle perform NLP tasks, although noise and limited circuit depth in NISQ devices is currently a key challenge [677,690,700].

While classical deep learning still dominates NLP in both research and industry, there has been growing interest in exploring QNLP methods [672,699,701], particularly for applications like sentiment analysis in finance [672]. For instance, one can use these methods to perform sentiment analysis to predict market trends, analyse investor sentiment, and improve decision-making in trading. This usually involves extracting positive, neutral, or negative sentiments from financial text data.

Moreover, a potential advantage of QNLP (especially in scaling to more complex datasets) relies on faster and more efficient data processing compared to classical NLP approaches [677]. This is very important in finance, as timely analysis of massive amounts of textual data can provide a competitive advantage [702]. The aim is that this explicit approach might eventually lead to more robust or efficient models for specific financial NLP tasks. If scaled up, QNN-based NLP could add new capabilities to classical models [690]; however, demonstrating a clear quantum advantage in NLP financial tasks remains an ambitious goal. Further development in encoding schemes, possibly leveraging dimensionality reduction or hybrid classical-quantum architectures, as well as more stable qubits and error-corrected quantum systems are crucial to fully realise the potential of QNLP in finance.

## 35. Risk Management and Cybersecurity

In modern finance, risk management and cybersecurity have become foundational pillars for sustaining operational integrity, client trust, and regulatory compliance [703]. Financial institutions face a complex landscape of threats ranging from credit and market risk to sophisticated cyberattacks targeting digital infrastructure. As the financial sector becomes increasingly digitised and interconnected, safeguarding assets and ensuring resilience against uncertainties has transcended from a compliance necessity to a strategic imperative [704].

Risk management entails identifying, assessing, and mitigating potential financial losses arising from market volatility, operational failures, credit defaults, and systemic crises [705]. An effective risk management framework enables firms to allocate capital efficiently, optimise portfolio returns relative to risk exposure, and maintain solvency under stress conditions. Concurrently, cybersecurity protects the confidentiality, integrity, and availability of financial data and systems against malicious activities such as hacking, ransomware, and insider threats [706]. Failures in either domain can result in catastrophic financial loss, reputational damage, and systemic contagion effects that extend beyond

<sup>37</sup> Diagrammatic structure is a graphical high-level representation of how meanings and grammatical structures interact to form the meaning of a sentence, see [688] for a detailed description.

individual institutions. The increasing reliance on automated trading, cloud-based infrastructures, and real-time payment systems has further amplified vulnerabilities, necessitating advanced and proactive defenses. Financial institutions are faced with two kinds of limitations: on one hand, classical computational methods for risk assessment often rely on complex simulations and/or optimisation algorithms which have several bottlenecks related to problem dimensionality and computation time [707]. Quantum computing has the potential to alleviate these bottlenecks as it can naturally perform computations in a high-dimensional Hilbert space. On the other hand, classical cryptographic methods (e.g., RSA) depend on the computational difficulty of problems like integer factorisation and discrete logarithms, which are theoretically vulnerable to advances in quantum computing (see Section 24). Therefore, the inevitable solution is to incorporate cryptographic systems that are quantum-safe.

In this chapter, we will consider some use-cases of quantum technologies in risk-management and cybersecurity. According to a study by McKinsey [20], the projected value of quantum technology applications for risk-management and cybersecurity will be about \$80 billion by 2035. Hence, strategic investments in quantum research and early adoption frameworks will likely define the resilience and competitiveness of financial institutions in the coming decades.

### 36. Value at Risk and CVaR

Risk analysis in finance can be either quantitative or qualitative [708]. Under quantitative risk analysis, a risk model is built using simulation or deterministic statistics to assign numerical values to risk. Inputs that are mostly assumptions and random variables are fed into a risk model. For any given input range, the model generates a range of outputs or outcomes [708]. The model's output is analysed using graphs, scenario analysis, and/or sensitivity analysis to make decisions to mitigate and deal with the risks. On the other hand, qualitative risk analysis [709] is an analytical method that does not identify and evaluate risks with numerical and quantitative ratings. Qualitative analysis involves a written definition of the uncertainties, an evaluation of the extent of the impact (if the risk ensues), and countermeasure plans in the case of a negative event occurring [709]. Risk assessment enables corporations, governments, and investors to assess the probability that an adverse event might negatively impact a business, economy, project, or investment.

One of the most important metrics for quantitative risk assessment are the Value-at-Risk (VaR) and Conditional Value at Risk (CVaR) [710–714]. The VaR is a measure of the worst expected loss on a portfolio of instruments resulting from market movements over a given time horizon and a pre-defined confidence level [715]. This metric is most commonly used by investment and commercial banks to determine the extent and occurrence ratio of potential losses in their institutional portfolios. Risk managers can use VaR to measure and control the level of risk exposure for compliance. For example the Basel III regulations require banks to perform stress tests using VaR [716]. The associated metric CVaR, also known as the *Expected Shortfall*, is a measure of the average of all potential losses exceeding the VaR at a given confidence level. This risk assessment measure quantifies the amount of tail risk an investment portfolio has. The metric is derived by taking a weighted average of the "extreme" losses in the tail of the distribution of possible returns, beyond the VaR cutoff point. CVaR is used in portfolio optimisation for effective risk management [717].

#### Mathematical Formulation

Consider a portfolio with uncertain future profit/loss  $L$ , then we can define the VaR as:

**Definition 6.** Given a confidence level  $\alpha \in (0, 1)$ , the portfolio Value-at-Risk is

$$VaR_{\alpha} = \inf\{l \in \mathbb{R} : \mathbb{P}(L \leq l) \geq \alpha\} = \inf\{l \in \mathbb{R} : F_L(l) \geq \alpha\}, \quad (171)$$

where  $F_L(l) = \mathbb{P}(L \leq l)$  is the Cumulative Distribution Function (CDF) of  $L$ .

In other words, the VaR is the smallest number  $l$  such that the probability the loss exceeds  $l$  is at most  $1 - \alpha$ . In practise,  $\alpha = 0.999$ , thus VaR is the loss level that will not be exceeded with 99.9% confidence. As outlined in Algorithm (26), the VaR can be computed using MCI [717,718]. However, as discussed in Section (15) the MCI methods have a slow convergence, hence quantum algorithms that have a potential quadratic speed-up are desirable.

---

**Algorithm 26** Computation of VaR via Monte Carlo Simulation
 

---

**Require:** Number of simulations  $M$ , portfolio value  $V_0$ , model parameters (e.g.,  $\mu, \sigma$ ), confidence level  $\alpha$

**Ensure:** Estimated Value-at-Risk  $VaR_\alpha$

- 1: **for**  $i = 1$  to  $M$  **do**
  - 2:   Simulate one portfolio return  $r_i$  based on the assumed distribution (e.g., normal:  $r_i \sim \mathcal{N}(\mu, \sigma^2)$ )
  - 3:   Compute simulated profit and loss:  $L_i = V_0 \times r_i$
  - 4: **end for**
  - 5: Sort  $\{L_i\}$  in ascending order
  - 6: Find the  $(1 - \alpha)$  quantile of the sorted  $\{L_i\}$
  - 7: Set  $VaR_\alpha = -\text{Quantile}(L, 1 - \alpha)$
  - 8: **return**  $VaR_\alpha$
- 

We can also define the CVaR based on the VaR:

**Definition 7.** Given the profit-loss variable  $L$  that is integrable and continuous CDF, then the CVaR at confidence level  $\alpha \in (0, 1)$  is defined as

$$CVaR_\alpha = \mathbb{E}[L | L \geq VaR_\alpha] = \frac{1}{1 - \alpha} \int_\alpha^1 VaR_u du. \quad (172)$$

This shows CVaR as an average of VaRs beyond the quantile  $\alpha$  [717] and can be computed using MCI.

### Quantum Approach

The most straightforward quantum approach to employ the QMCI for computing the VaR and CVaR. As shown in Section (8), the QAE has two loading operators:

$$\mathcal{P}_l |0\rangle_n = |\psi\rangle_n = \sum_{i=0}^{N-1} \sqrt{p_i} |i\rangle_n, \quad \text{with} \quad \sum_{i=0}^{N-1} p_i = 1, \quad (173)$$

which prepares the initial state with discretised probabilities  $p_i$  using  $n$  qubits, where  $N = 2^n$ . The second operator  $\mathcal{F}_l$  which applies the function  $f : \{0, \dots, N-1\} \mapsto [0, 1]$  on corresponding states as:

$$\mathcal{F}_l |i\rangle |0\rangle = |i\rangle \left( \sqrt{1 - f(i)} |0\rangle + \sqrt{f(i)} |1\rangle \right). \quad (174)$$

These operators combine to give

$$\mathcal{F}_l \mathcal{P}_l |0\rangle_n |0\rangle = \sum_{i=0}^{N-1} \sqrt{p_i} \sqrt{1 - f(i)} |i\rangle |0\rangle + \sum_{i=0}^{N-1} \sqrt{p_i} \sqrt{f(i)} |i\rangle |1\rangle, \quad (175)$$

thus by measuring the probability of the last qubit  $|1\rangle$  we get an estimate for the expectation value  $\mathbb{E}[f(X)] = \sum_{i=0}^{N-1} p_i f(i)$ , where  $X$  is mapped into  $\{0, \dots, N-1\}$ . In Ref. [710], this method is extended by defining a function

$$f(i) = \begin{cases} 1 & \text{if } i \leq l \\ 0 & \text{otherwise} \end{cases} \quad (176)$$

where  $l \in \{0, \dots, N-1\}$ . Therefore, Equation (175) reduces to:

$$\mathcal{F}_l \mathcal{P}_l |0\rangle_n |0\rangle = \sum_{i=l+1}^{N-1} \sqrt{p_i} |i\rangle |0\rangle + \sum_{i=0}^l \sqrt{p_i} |i\rangle |1\rangle. \quad (177)$$

The probability of measuring the last qubit  $|1\rangle$  is given by  $\sum_{i=0}^l p_i = \mathbb{P}[X \leq l]$ . The authors in [710] then use a bisection search over  $l$  to find the smallest level  $l_\alpha$ , which corresponds to the  $\text{VaR}_\alpha(X)$ , such that  $\mathbb{P}[X \leq l_\alpha] \geq \alpha$  in at most  $n$  steps. Similarly, an appropriate function  $f(\cdot)$  for CVaR can be defined, and then the same procedure is followed. This quantum algorithm has an error that scales as  $\mathcal{O}(1/M)$  which is a quadratic speedup over MCI described in Algo. (26). A proof-of-concept implementation of this QMCI for VaR and CVaR on quantum computers was performed in [710,719]. Additionally, Ref. [719] offers estimates for the total number of qubits needed, the anticipated circuit depth, and how these factors relate to the expected runtime, assuming reasonable projections for future fault-tolerant quantum hardware.

### 37. Credit Risk Analysis

Credit risk analysis is a fundamental discipline in finance concerned with the assessment of the likelihood that a borrower will default on contractual debt obligations [720]. It plays a critical role not only in individual lending decisions but also in the strategic management of financial institutions' portfolios and the determination of necessary capital reserves [720]. With the globalisation of financial markets and the increasing complexity of credit products, the methodologies for assessing, pricing, and managing credit risk have evolved substantially.

The two major branches of credit risk analysis are: the quantification of Economic Capital Requirement (ECR) and the development of robust credit scoring systems. The ECR refers to the amount of capital a financial institution needs to ensure that it stays solvent with a high degree of confidence over a specified time horizon [721]. Credit scoring, on the other hand, operationalises credit risk evaluation at the transaction level by using statistical and machine learning techniques to predict a borrower's probability of default [722]. Credit scoring models are essential not only for underwriting but also for regulatory compliance, particularly under frameworks like Basel II and III, which link credit risk quantification directly to minimum ECR [716]. Traditional models such as logistic regression remain prevalent [722,723], although recent advancements have integrated ensemble methods and deep learning to improve predictive performance [724,725].

However, classical computational bottlenecks arise in these systems due to the increasing size and complexity of datasets, the non-linear interactions between multiple risk factors, and the need for real-time or near-real-time analytics [726]. Given these limitations, researchers have been exploring how quantum computing can offer speed-ups, accuracy, and efficiency [727]. In this section, we will state the mathematical formulations of these credit risk analysis, and outline some proposed quantum computation approaches to these problem which may potentially offer an advantage.

#### 37.1. Economic Capital Requirement

As described above, the ECR is an important risk metric which quantitatively captures the amount of capital required to remain solvent at a given confidence level [716,721]. It captures unexpected losses arising from credit events and serves as a buffer against adverse outcomes. Institutions allocate economic capital across their credit portfolios based on internal models that estimate the distribution of potential losses, taking into account default probabilities, loss given default, and exposure at default [728]. The result reflects the amount of capital that the firm should have to support any risks that it takes on its investment portfolio. The measurement process for economic capital involves converting a given risk to the amount of capital that is required to support it. The calculations are based on the institution's financial strength (or credit rating) and expected losses. Financial strength is the probability of the firm not becoming insolvent over the measurement period and is otherwise known as the confidence level in the statistical calculation [728].

### Mathematical Formulation

Let  $L_{\mathbf{x}}$  denote the random variable representing the total *loss* of the institution's portfolio of  $K$  assets, denoted by  $\mathbf{x} = (x_1, \dots, x_K) \in \mathbb{R}^K$ , over a fixed time horizon (typically one year). The distribution of  $L_{\mathbf{x}}$  is characterised by its CDF  $F_{L_{\mathbf{x}}}(l) = \mathbb{P}(L_{\mathbf{x}} \leq l)$  as described in Section (36). The ECR is commonly defined as the difference between the VaR and the expected loss [721]:

$$\text{ECR}_{\alpha}[L_{\mathbf{x}}] = \text{VaR}_{\alpha}(L_{\mathbf{x}}) - \mathbb{E}[L_{\mathbf{x}}], \quad (178)$$

where  $\mathbb{E}[L_{\mathbf{x}}]$  denotes the expected (mean) loss of the portfolio. Alternatively, in more robust risk management frameworks [729], the CVaR can be used to define ECR as:

$$\text{ECR}_{\alpha}[L_{\mathbf{x}}] = \text{CVaR}_{\alpha}(L_{\mathbf{x}}) - \mathbb{E}[L_{\mathbf{x}}]. \quad (179)$$

In practice, institutions seek to allocate their portfolio of assets denoted to minimise the ECR, subject to regulatory, liquidity, and investment constraints. A general optimisation problem can be stated as [730,731]:

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{X}} \text{ECR}_{\alpha}[L_{\mathbf{x}}] \\ & \text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m, \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j = 1, \dots, p, \end{aligned} \quad (180)$$

where  $\mathbb{X}$  is the feasible set determined by internal risk policies and external regulations, and:

- $g_i(\mathbf{x})$  are inequality constraints representing risk limits (e.g., sector exposure caps, maximum portfolio volatility, leverage restrictions).
- $h_j(\mathbf{x})$  are equality constraints ensuring portfolio balance (e.g., full investment constraint  $\sum_{k=1}^K x_k = 1$ , or specific regulatory ratios).

### Quantum Approach

We see that the problem of ECR is two fold:

1. First, for a given portfolio of assets  $\mathbf{x}$ , compute the associated  $\text{ECR}_{\alpha}[L_{\mathbf{x}}]$  by modeling its total loss  $L_{\mathbf{x}}$  using *Gaussian conditional independence models*<sup>38</sup> [732] such as discussed in Section (32.2) for Collateralised Debt Obligations. The problem is solved in three-steps:
  - (a) Compute  $\mathbb{E}[L_{\mathbf{x}}]$  which is the same form as Equation (154), and thus it has been shown by [627] that this can be efficiently performed on a quantum computer using QMCI algorithm.
  - (b) Compute the  $\text{VaR}_{\alpha}(L_{\mathbf{x}})$  just as in Section (36) using QMCI algorithm [710,719].
  - (c) Combine the above to compute the  $\text{ECR}_{\alpha}[L_{\mathbf{x}}]$  by using Equation (178).
2. Second, repeat the above to find an optimal portfolio  $\mathbf{x}^* \in \mathbb{X}$  that solves the optimisation problem (180). This can be done by converting the problem into a QUBO and solve it using quantum algorithms described in Chapter (17).

A proof-of-concept computation of ECR using QMCI on quantum computers was performed in [719] with comparative accuracy to classical methods. Additionally, [719] offers quantum resource estimates for future fault-tolerant quantum hardware. However, unlike the pipeline described above, the authors in [719] used classical numerical integration for computing  $\mathbb{E}[L_{\mathbf{x}}]$  and did not proceed to solve the optimisation problem (180). Therefore, at the best of our knowledge, there is still an opportunity for researchers to explore the benefits of the end-to-end pipeline involving multiple quantum algorithms.

<sup>38</sup> This scheme resembles the one employed for regulatory purposes in the Internal Ratings-Based approach to credit risk under Basel II and subsequent frameworks.

### 37.2. Credit Scoring

The process of credit scoring, otherwise called defining creditworthiness, is one of the most difficult problems in finance due to the large amount of data necessary to determine key independent features that can influence the end result [722,724,725]. Credit scoring can be identified as a potential use-case both for quantum optimisation as well as Quantum Machine Learning techniques. A typical credit score is influenced by payment history, accounts owed, length of credit history, new credit, and credit mix [733]. For business credit, this information can be gathered from historical business data, public filings, and payment collections. Currently, credit scores are often predicted by using classical regression models or machine learning algorithms [734,735]. In order to get more accurate results, it is helpful for the machine learning algorithms to train on more data; however, this process is computationally intensive. Some researcher are exploring the potential for QML to be able to efficiently resolve critical features to determine a person's creditworthiness.

#### Mathematical Formulation

One approach of credit scoring is to reformulate it as a combinatorial optimisation problem aiming to select the most influential (or highly affecting) features for identifying creditworthiness [736]. In this approach, one considers a dataset of past credit applicants represented by a matrix  $U \in \mathbb{R}^{m \times n}$ :

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mn} \end{bmatrix} \quad (181)$$

Each row represents an applicant and each column a feature. The outcome vector  $V \in \{0, 1\}^m$  represents credit decisions:

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad (182)$$

where  $v_i = 0$  indicates acceptance and  $v_i = 1$  indicates rejection. Suppose we begin with  $n$  original features and aim to identify a subset of  $K$  features to guide our credit decision-making process<sup>39</sup>. To avoid introducing bias in the selection, we desire an exhaustive yet efficient search. However, the number of possible subsets of size  $K$  is combinatorial, given by  $C(n, K)$ . Even with early pruning, the search space remains prohibitively large, necessitating a strategy that prioritises promising candidate subsets [736]. The aim is to select columns from the data matrix  $U$  that exhibit strong correlation with the target vector  $V$ , while being mutually uncorrelated among themselves. We assume a correlation metric yielding values in the range  $[-1, 1]$  is available. In practice, "correlation" may encompass domain-specific rules, such as mandatory inclusion of certain features, that go beyond simple statistical association [736].

Let  $\rho_{ij}$  denote the correlation between the  $i$ -th and  $j$ -th columns of  $U$ , and let  $\rho_{V_j}$  be the correlation between the  $j$ -th feature and the target  $V$ . We introduce binary variables  $x_j \in \{0, 1\}$  to indicate whether feature  $j$  is selected:

$$x_j = \begin{cases} 1, & \text{if feature } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad (183)$$

<sup>39</sup> There are several motivations for feature reduction [736]: financial costs associated with data collection, the presence of redundant information, and features so weakly associated with the target outcome that they behave like noise.

We thus define the objective function [736]:

$$f(\mathbf{x}) = - \left[ \alpha \sum_{j=1}^n x_j |\rho_{V_j}| - (1 - \alpha) \sum_{j=1}^n \sum_{\substack{k=1 \\ k \neq j}}^n x_j x_k |\rho_{jk}| \right] \quad (184)$$

where  $\alpha \in [0, 1]$  is the parameter that balances influence and independence such that:  $\alpha = 0$  and  $\alpha = 1$  prioritises the independence and influence that features has on the marked class, respectively [736].

### Quantum Approach

Let  $\mathbf{x} = (x_1, \dots, x_n)$  be a collective vector of the binary variables, and by using the property  $x_j x_j = x_j$ , the objective function in Equation (184) can be expressed as a quadratic form:

$$f(\mathbf{x}) = -\mathbf{x}^\top \mathbf{Q} \mathbf{x} \quad (185)$$

The optimal subset  $\mathbf{x}^*$  is given as a QUBO:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left( -\mathbf{x}^\top \mathbf{Q} \mathbf{x} \right) \quad (186)$$

This problem can then be solved using quantum optimisation techniques for QUBOs described Chapter (17). A proof-of-concept implementation on a QAnn using German credit data was shown to yield comparative accuracy as tradition feature selection methods [736]. In addition, the QUBO feature selection framework in [736] yielded a smaller feature subset with no loss of accuracy compared to a common classical methods like recursive feature elimination. This paves the way for leveraging quantum computing to systematically reduce the dimensionality of large feature sets, particularly as QPUs become more advanced.

Other more common proposals for credit scoring on quantum computers are based on QML includes *systemic quantum score* using quantum kernels [737], a hybrid quantum-classical neural network approaches [738,739], and quantum-inspired evolutionary method for feature selection [740]. Although still in its early stages and constrained by current quantum hardware limitations and theoretical hurdles, the findings to date highlight the promising potential of quantum computing for efficient credit scoring. Improvements in accuracy, processing speed, inclusivity, and feature optimisation point to a promising future for quantum-enhanced credit risk assessment in finance [737]. As quantum technology advances, continued progress in algorithm development, data representation, and hybrid quantum-classical methods will be essential to achieving Practical Quantum Advantage and integrating these innovations into the financial industry.

## 38. Fraud Detection

Fraud detection is a mission-critical function in financial institutions, safeguarding against losses and ensuring regulatory compliance. It refers to identifying events such as transactions or withdrawals which are likely to be fraudulent, based on data about them. Fraud poses a huge threat to economic systems - the UK Finance Annual Fraud Report recorded more than \$1.5 billion fraud-related losses in the United Kingdom alone in 2023, while total global losses were estimated at approximately \$485.6 billion [741]. To crack down on this, the UK Economic Crime and Corporate Transparency Act 2023 and other similar legislation are imposing more responsibilities on various financial organisations to combat fraud [742]. Currently, classical machine learning techniques are used to detect and prevent fraud, but they often suffer from problems including a high false positive rate reaching 80% [743], and long inference times. Given the high impact of even small improvements in terms of time and money saved, organisations are turning to more advanced algorithms including quantum computing, to boost performance.

## Classical Methods

Classical machine learning techniques have high performance variability due to the model used, apart from other factors such as data selection, formatting, and preprocessing, model choice and hyperparameter fixing. Some particular models and algorithms are popular due to their interpretability, robustness, and computational efficiency. Logistic regression is suited to binary classification tasks like credit scoring, and is easy to set up and use [744]. Decision trees are useful for providing transparency into decisions, although they may be prone to overfitting (where models generalise insufficiently) [745]. This may be overcome using random forests, which are ensembles of decision trees and aggregate multiple tree outcomes [746]. Finally, support vector machines [747] and k-Nearest Neighbors [331] can be computationally expensive, but can identify subtle patterns associated with fraud. Each method has a range of benefits and drawbacks, enumerating which is outside the scope of this paper, and correct model selection and preparation is vital in setting up a functioning and high-performing fraud detection pipeline. A recent survey paper [748] summarises developments in both types of fraud, and in the ML pipelines used to detect them. In particular, it appears deep learning techniques including Convolutional Neural Networks (CNNs) and ensemble methods have experienced a surge in popularity due to their greater robustness and more rapid classification [749,750].

## Quantum Approaches

Quantum approaches to fraud detection can be split into two domains: quantum algorithms applied to classical data, and those applied to data that has been loaded into a quantum format. For the former, the data are simply a table of numbers, which may refer to observed, calculated, or encoded properties of a particular transaction. Quantum  $k$ -means and spectral clustering algorithms are known to be potentially more efficient over classical counterparts [420] (see Section 22.1). A recent hybrid quantum-classical framework implements classification through QRAMs enhanced by quantum feature selection, demonstrating improved accuracy through complementary exploration of the feature space [751]. These implementations are conceptually simple as the data remain unchanged, necessitating little novel pre-processing. A simple example of this is quantum neural networks was demonstrated in Ref. [752]. In the report, they constructed a classical deep neural network, and replaced one of the layers with a Parametrised Quantum Circuit. This enabled them to achieve similar accuracy scores on the same data, using fewer parameters, resulting in a more performant pipeline [752].

For fully quantum implementations, the data must be encoded in a quantum format. There are a number of ways to do this; one common method is using Quantum Generative Adversarial Networks (QGAN) [753]. A GAN is a combination of a generator and a discriminator model. The generator takes as input a noise vector, and outputs a data vector, while the discriminator attempts to distinguish the real and generated data. Over the course of training, the generator learns the data's underlying distribution. The quantum version of this uses a QNN (see Section 21.3) in place of the generator. In essence, it takes an equal superposition of qubits and learns the circuit parameters to convert this into a quantum-encoded datapoint. Thus, the data are indirectly encoded into a quantum format. This enables the method to potentially exploit a quantum advantage in sampling efficiency: classically sampling from a complex probability distribution is computationally expensive, while the quantum version is far easier [445,467,754]. The aforementioned study [753] reported results that achieve performance on par with currently existing classical methods, while potentially circumventing their key drawbacks of training instabilities and sampling inefficiency. Additional approaches include QAE for density-estimation based anomaly detection, quantum-enhanced kernels for clustering methods [755], and Quantum Boltzmann Machine [444].

Although there is some optimism on these quantum methods, a recent study [415] raised doubts about the proper benchmarking of these quantum approaches. The authors developed a benchmarking suite and used it to compare classical and quantum ML algorithms; they found that out-of-the-box classical models tended to outperform quantum ones on the small datasets used. On the other hand,

it is known that results from small amounts of data do not necessarily hold for larger amounts, due to the properties of deep learning techniques [415]. Thus, it is possible that as the scale increases, the benefits of the quantum approach become apparent. As a result, it is still an open question as to the precise benefit that is to be gained from a quantum approach to fraud detection.

### 39. Quantum Safe Cryptography

One very frequent and important vulnerability that financial services companies tend to have is a reliance on legacy software, with commensurate security issues and time-consuming patches and upgrades, which includes the security of that software [756]. A key problem exposed by this is the need to upgrade security to Post-QC protocols<sup>40</sup> before quantum computers capable of breaking current encryption algorithms become feasible [563]. The expected delay before this occurs may tempt organisations to put off this transition. However, waiting for malicious actors to pose a present risk by possessing quantum computers is highly dangerous, not only due to the risk of being caught off-guard, but due to the threat from ‘harvest-now decrypt-later’ attacks where information is gathered and held until it can be decrypted [521].

Given the complexity of the field, attempts to enact this transition entirely internally are likely to lead to substandard, and insecure, implementations. The simplest path is to rely on recommendations from national or international bodies, including NCSC [563] NIST [757], ETSI [758], and the IETF [759], which are actively working on developing and disseminating such standards. However, as mentioned above, Post-QC is still a relatively novel field, and while roll-out has begun on a large scale, there is every chance that novel attacks will be developed that can break these protocols. One must therefore balance the potential risk of a switch with the likelihood of harvest now /decrypt later attacks. Furthermore, security is a weak-link problem: any system is only as secure as its most insecure point, and for cryptography, that point is human error. Post-QC protocols cannot be seen as a panacea, and must be implemented in tandem with rigorous and regular training and testing of personnel. If this is carried out, organisations’ data stands a good chance of remaining secure against hostile actors equipped with quantum computers.

### 40. Economics

Economics, at its core, is the study of how individuals, firms, and societies allocate scarce resources to meet their needs and desires [760]. The field spans microeconomic inquiries into individual behavior and market mechanisms to macroeconomic studies of national income, inflation, and employment. Central to economics is the development and analysis of models that help explain complex systems, predict outcomes, and inform policy. The key objectives of the field include efficiency in resource allocation, economic growth, equitable distribution of wealth, and the stabilisation of economies through informed decision-making [25,760–763].

To achieve these objectives, economists rely heavily on mathematical modeling, statistical analysis, and computational simulations [764]. These tools are vital in forecasting market trends, understanding consumer behavior, optimising financial portfolios, and solving general equilibrium models. However, as datasets grow larger and models become more sophisticated—especially in areas like game theory [765], and agent-based modeling [766,767] limitations of classical computing become apparent. For example, in supply chain economics [768], solving for the optimal allocation of resources in real-time can be computationally expensive, especially as the number of variables increases. Quantum technologies offers a potential efficient alternative for economic modeling [25]. In this chapter we will outline some of the potential use-cases of quantum computing in economics which includes quantum money and economic modeling.

<sup>40</sup> These are quantum-safe protocols and some are outlined in Section (27).

## 41. Quantum Money

The concept of quantum money, first introduced by Stephen Wiesner in the 1980s [577], represents one of the earliest applications of quantum information to cryptography. At its core, quantum money exploits the *no-cloning theorem*, which asserts that it is fundamentally impossible to create an exact copy of an arbitrary unknown quantum state [570]. This property provides a natural basis for unforgeability, a critical requirement for any monetary system. Quantum money has become a foundational example of how quantum information can provide security guarantees impossible in classical systems.

### 41.1. Wiesner Private-Key Quantum Money

In Wiesner's original formulation [577], a quantum banknote is a tuple  $(s, |\psi_s\rangle)$ , where:

- $s \in \{0, 1\}^m$  is a classical serial number.
- $|\psi_s\rangle \in \mathcal{H}^{\otimes n}$  is an  $n$ -qubit quantum state.

Each qubit is chosen from one of two mutually unbiased bases:

$$\{|0\rangle, |1\rangle\} \text{ (computational basis), } \{|+\rangle, |-\rangle\} \text{ (Hadamard basis),} \quad (187)$$

where  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  and  $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ . The specific choice of basis and value for each qubit is known only to the issuing bank and stored in a secure database indexed by the serial number  $s$ . The bank randomly selects: A basis  $b_i \in \{Z, X\}$  for each qubit, and a bit  $v_i \in \{0, 1\}$  indicating the qubit value in that basis. It then generates the full state as:

$$|\psi_s\rangle = \bigotimes_{i=1}^n |\phi_i\rangle, \quad \text{where } |\phi_i\rangle \in \{|0\rangle, |1\rangle\} \text{ or } \{|+\rangle, |-\rangle\}. \quad (188)$$

The bank maintains a secret mapping:

$$\text{DB}[s] = ((b_1, v_1), \dots, (b_n, v_n)). \quad (189)$$

To verify a note, the bank measures each qubit in the basis  $b_i$  and checks whether the outcome matches  $v_i$ . Any discrepancy leads to rejection.

The security of Wiesner's scheme stems from the no-cloning theorem, which can be formally stated as follows:

**Theorem 1** (No-Cloning Theorem [570]). *There does not exist a unitary operator  $U$  and a fixed state  $|e\rangle$  such that:*

$$U(|\psi\rangle \otimes |e\rangle) = |\psi\rangle \otimes |\psi\rangle$$

for all  $|\psi\rangle \in \mathcal{H}$ .

The implication is that any adversary who tries to duplicate an unknown quantum state  $|\psi_s\rangle$  without knowledge of the preparation bases will likely alter it irreversibly. Measuring a qubit in the wrong basis yields a random result, destroying the original state and producing a statistically distinguishable forgery<sup>41</sup>.

However, Wiesner's quantum money has at least three notable limitations [25]. First, it requires verification through an online authority, reducing its practicality compared to traditional cash. Second, it is based on a private-key scheme, meaning the issuer must keep certain verification details secret. Third, the scheme is technologically difficult to implement practically given its quantum memory and hardware requirements. Hence, the ensuing section will describe modern approaches to quantum money that have a decentralised verification protocol which in principle addresses the first two challenges with Wiesner's protocol.

<sup>41</sup> See Section (28) where a similar idea is used for Quantum Key Distribution

### 41.2. Modern Public-Key Quantum Money

To overcome the limitations of Wiesner's quantum money, the concept of public-key quantum money emerged, aiming to create a system where anyone could verify the validity of a quantum banknote without needing to consult the issuing authority [25]. This development represents a significant step towards realising a more practical and widely usable form of quantum currency. The concept of *publicly verifiable quantum money* was first proposed by Aaronson [769] and later developed in [770–772] and other works, which provided schemes where the authenticity of a quantum banknote can be verified without access to a secret database. These schemes typically have the following design:

- A key generation algorithm  $\text{Gen}() \rightarrow (pk, sk)$ ,
- A quantum state generator  $\text{Mint}(sk) \rightarrow |\psi\rangle$ ,
- A verification procedure  $\text{Verify}(pk, |\psi\rangle) \in \{0, 1\}$ .

Here, the pair of generated keys  $(pk, sk)$  are the public-key and secret-key, respectively. The security of such schemes is often based on computational assumptions resistant to quantum attacks, such as the hardness of lattice problems or quantum-secure hash functions [772,773]. An alternative approach is the work of Zhandry [774], who has proposed a public-key quantum money scheme based on abelian group actions. The key idea is to create quantum states (called 'bolts') that are unclonable and non-reproducible, even by the entity who generated them — a stronger notion than traditional quantum money. This can be formalised as:

- Bolt Generation:  $\text{Gen}() \mapsto \{|\psi_s\rangle, s\}$  which outputs a quantum state  $|\psi_s\rangle$  ('bolt') and serial number  $s$ .
- Verification:  $\text{Ver}(|\psi_s\rangle) \mapsto s$  or  $\perp$  which returns the serial number if valid or rejects, respectively.

The security of the *quantum lightning* is that no efficient quantum adversary can create two bolts  $\{|\psi_1\rangle, |\psi_2\rangle\}$  such that both verify to the same serial number [773]. This goes beyond the no-cloning theorem and ensures computational unclonability even with access to the generation process. Making bill serial numbers public and unique allows anyone to verify the total money supply, offering transparency. Unlike physical cash—where a central bank could secretly print duplicates—this system prevents such manipulation. It reduces the need to fully trust central banks, which is especially useful in countries with a history of inflation.

There are currently a variety of proposed quantum money schemes, which can be categorised by the type of money:

1. **Quantum Bill** - Identifiable quantum money units with unique serial numbers, traceable across transactions (less anonymous) [577,769,771,775].
2. **Quantum Coins** - Indistinguishable quantum money units that offer user anonymity (untraceable) [776–779].
3. **Quantum Bolts** - Public quantum money with the additional guarantee that even the issuer cannot create duplicates with the same serial number [772,774,780,781].
4. **Quantum Smart Contracts**<sup>42</sup> - hybrid classical-quantum payment system based on classical blockchains capable and public-key quantum money quantum bolts (decentralised) [783,784].

Most of the modern constructions aim to preserve the physical unforgeability offered by quantum mechanics while enabling decentralised verification, a crucial property for practical deployment. However, some of these protocols have been shown to be insecure [770] due to many factors which include information leakage from the verification procedure which can allow attackers to learn enough to forge the quantum money [772], unrealistic hardness assumptions (e.g., hidden subspace problems) [771], or inadequate modeling of adversaries especially quantum-capable ones [775].

On one hand, theorists continue to perform rigorous cryptanalysis of such proposed schemes i.e. Bilyk et al. recently showed that Zhandry's quantum bolts scheme may be insecure [785]. On

<sup>42</sup> A smart contract is, in essence, a mechanism that enables parties to deposit funds and automatically release them once certain algorithmically verifiable conditions are met [782]. This makes it a formal tool for enforcing monetary incentives.

the other hand, experimentalists are performing proof-of-concept demonstrations of quantum money using coherent states of light [786,787] and quantum S-tokens which eliminates reliance on quantum memories and long-distance quantum communication [578,788]. The later approach by [578] has reported a quantified time advantage over optimal classical cross-checking protocols for transactions. This could potentially be adopted for applications requiring high security, privacy and minimal transaction times, like financial trading and network control [578].

Quantum money presents a compelling blend of quantum physics and cryptography, where fundamental quantum principles like measurement uncertainty and the no-cloning theorem enforce properties that are otherwise unattainable in classical systems. The ongoing research into publicly verifiable schemes [25,26] may result into practical quantum-secure finance.

## 42. Economic Forecasting

Economic forecasting is the process of attempting to predict the future condition of the economy using a combination of widely followed indicators [789]. Government officials and business managers use economic forecasts to determine fiscal and monetary policies and plan future operating activities, respectively. In this section, we explore some of the potential quantum computing use-cases for economic forecasting areas which includes time series analysis, synthetic data generation, and predicting financial disasters.

### 42.1. Time Series Analysis

Time series analysis, a fundamental pillar of quantitative analysis in economics and finance, involves examining sequences of data points collected over time to discern patterns, understand underlying dynamics, and ultimately forecast future values [790]. This methodology plays a crucial role for financial analysts in making informed investment decisions by providing insights into historical trends and enabling the prediction of future movements in key variables such as quarterly sales figures and daily asset returns. The consistent collection of data points at defined intervals allows for the observation of how variables evolve, revealing critical information about market behavior and economic activity.

In contemporary financial and economic landscapes, the sheer volume and intricate nature of data present significant analytical challenges. Modern markets are characterised by non-linear relationships, high volatility, and complex interdependencies that often strain the capabilities of traditional computational methods [791]. This increasing complexity necessitates the exploration of novel computational paradigms capable of handling these intricate datasets and extracting meaningful insights with greater efficiency and accuracy. Given the inherent computational intensity of advanced time series analysis in finance and economics, quantum computing presents a compelling avenue for exploration [792].

First, let us introduce some key concept in time series analysis before describing the associated quantum algorithms that can potentially provide a performance boost. A time series is considered *covariance stationary* if its statistical properties, including the expected value (mean), the variance, and the covariance between values at different time points (autocovariance), remain constant over time [791]. Many time series encountered in economics and finance are non-stationary, meaning their statistical properties change over time, often exhibiting trends or seasonality [793]. In such cases, it is often necessary to transform the series into a stationary one, for example, by using techniques like differencing, before applying certain time series models [790,791]. Other key concepts include *seasonality* (patterns that repeat over a period), *trends* (general upwards or downward direction over time), and *noise* (random ups and downs) [791]. There are various models time series models that work better for some data-sets depending on their feature. In this review, we will outline a few classical models and associated quantum approaches for time-series analysis.

## Classical Time Series Models

A wide variety of time series forecasting models exist, utilising different mathematical approaches and varying in complexity. Among the most commonly used are the Auto-Regressive Integrated Moving Average (ARIMA) and Smoothing model families. In this review, we will outline the ARIMA models which is a class of models that predicts the future values of a given time series based on its own past values (lags) and the lagged forecast errors [794].

First, the Auto-Regressive (AR) models of order  $p$ , denoted as  $AR(p)$ , posit that the current value of a time series is linearly dependent on its  $p$  most recent past values, along with a constant term and a random error term. Mathematically, this can be expressed as:

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \epsilon_t \quad (190)$$

where  $y_t$  is the value at time  $t$ ,  $c$  is a constant,  $\phi_i$  are the auto-regressive parameters, and  $\epsilon_t$  is the error term.

Second, the Moving Average (MA) models [794] of order  $q$ , denoted as  $MA(q)$ , suggest that the current value of the time series is a linear combination of the past  $q$  error terms and the current error term. The general equation for an  $MA(q)$  model is given by:

$$y_t = \mu + \sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t \quad (191)$$

where  $\mu$  is the mean of the series,  $\theta_i$  are the moving average parameters, and  $\epsilon_t$  represents the error term at time  $t$ .

Third, Auto-Regressive Moving Average (ARMA) models [794] of order  $(p, q)$ , denoted as  $ARMA(p, q)$ , combine the auto-regressive and moving average components to model stationary time series. An  $ARMA(p, q)$  model can be written as:

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t, \quad (192)$$

encompassing the dependencies on both past values of the series and past error terms.

Fourth, for time series that exhibit non-stationarity, the ARIMA model of order  $(p, d, q)$  is widely used [795]. The 'Integrated' (I) part, denoted by  $d$ , refers to the number of times the raw observations are differenced to achieve stationarity. The ARIMA model essentially fits an  $ARMA(p, q)$  model to the  $d$ -th difference of the series. Using the backshift operator  $B$ , where  $B^k Y_t = Y_{t-k}$ , the general form of an  $ARIMA(p, d, q)$  model is [795]:

$$\phi_p(B)(1-B)^d Y_t = \theta_q(B) a_t, \quad (193)$$

where

$$\phi_{(p,q)}(B) = 1 - \phi_1 B - \dots - \phi_{(p,q)} B^{(p,q)} \quad (194)$$

are the auto-regressive  $p$  and moving average  $q$  polynomials in the backshift operator, respectively, and  $a_t$  is the white noise error term.

Fifth, to model the time-varying volatility often observed in financial time series, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model of order  $(p, q)$  is employed [796]. The  $GARCH(p, q)$  model posits that the conditional variance of the error term at time  $t$ , denoted by  $\sigma_t^2$ , depends on the  $q$  most recent lagged squared error terms (ARCH terms) and the  $p$  most recent lagged conditional variances (GARCH terms) [796]. The mathematical formulation for the conditional variance is [796]:

$$\sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2, \quad (195)$$

where  $\omega$  is a constant,  $\alpha_i$  are the coefficients for the ARCH terms, and  $\beta_j$  are the coefficients for the GARCH terms

Sixth, the Kalman Filter provides a different framework for time series analysis by employing a state-space representation to estimate unobserved variables [797]. It is a recursive algorithm that operates through prediction and update steps based on a system's dynamic model and noisy measurements. The core of the Kalman Filter lies in two key equations: the *state transition equation* and *measurement equation* given by [797]

$$x_{k+1} = \Phi x_k + w_k \quad \text{and} \quad z_k = H x_k + v_k, \quad (196)$$

which describes how the system's state evolves over time and relates the measurements to the system's state, respectively. Here,  $x_k$  is the state vector,  $\Phi$  is the state transition matrix,  $w_k$  is the process noise,  $z_k$  is the measurement vector,  $H$  is the measurement matrix, and  $v_k$  is the measurement noise.

The increasing volume of financial and economic data, coupled with the need to analyse high-dimensional time series (i.e., series with many variables observed over time), poses significant computational demands and scalability issues for classical methods. As the number of data points and the number of variables increase, the computational time and resources required for parameter estimation, forecasting, and validation can grow substantially, potentially hindering real-time analysis and decision-making [792]. Hence, quantum computing approaches have become attractive to many researchers and practitioners.

### Quantum Approach

There has been numerous proposals for performing time-series analysis on a quantum computer. A quantum analogue to the classical ARIMA model is proposed in [798] where the potential speed-up comes from employing the QFT as a subroutine (see Section 7). However, most proposals for time series analysis on quantum computers are based on QML methods discussed in Chapter (20). In particular, QNN which can be designed with fewer parameters than their classical counterparts while potentially achieving comparable or even improved accuracy, especially for complex time series signals such as explored in [799]. Proof-of-concept time series predictions of important economical factors such as the amount of rainfall fall and electric load power were implemented using QNN in [792] and the reported results were comparable to classical methods used in the study. Another variant of QML methods are approaches are based on quantum RNN explored in [800] where they also reported comparative results to classical methods when trained on real-world data.

One may ask the question: how do these QML models compare with classical methods for time series forecasting. An answer is found in a benchmarking study [801] where they found that in overall, state-of-the-art classical models currently outperforms their counterpart quantum models. However, most of the quantum models were able to achieve comparable results to classical methods, and for one data set two quantum models outperformed the classical ARIMA model [801]. These benchmarking results highlight the need for more research to develop better quantum models so that as quantum hardware increases in capabilities they may have a good chance to achieve Practical Quantum Advantage.

#### 4.2.2. Synthetic Data Generation

Synthetic Data Generation (SDG) has become a vital tool for training ML models [802], particularly in fields like finance where access to sensitive data is limited. SDG allows for ML practitioners to generate artificial datasets that resemble real-world data while avoiding many of the regulatory and logistical challenges associated with accessing sensitive financial information [803]. Banks and financial institutions, for example, can benefit from larger and more diverse datasets for risk modelling [804] and stress testing [805], especially when relevant data are scarce or do not reflect rare market events [804]. Customer behaviour analysis can also become less constrained by strict data protection rules and avoid exposing sensitive consumer information since artificial datasets preserve

important statistical relationships. Therefore, researchers and regulators often use synthetic datasets to compare or benchmark different predictive models, enabling a more transparent comparison of outcomes.

### Mathematical Formulation

Most SDG techniques focus on learning the underlying distribution  $p(\mathbf{x})$  of real data samples  $\mathbf{x}$  [802,806] (e.g., multiple financial variables such as asset prices or macroeconomic indicators), and construct a model  $G_\theta(\mathbf{z})$  (generative model) which transforms variables  $\mathbf{z}$  from a known prior distribution  $p(\mathbf{z})$  [803], such as a Gaussian or uniform distribution into synthetic samples  $\hat{\mathbf{x}} = G_\theta(\mathbf{z})$  [804]. The objective is typically to minimise a discrepancy measure between the real data distribution  $p(\mathbf{x})$  and the synthetic distribution  $q_\theta(\mathbf{x})$  (see section 22.2). In finance, this typically means learning a high-dimensional distribution that captures complex patterns, such as correlations across multiple assets [807] or autocorrelations in time series [808]. Additionally, financial use cases often involve constraints; for instance, a synthetic dataset used in stress testing needs to accurately reflect tail risk behaviour; this can be encouraged by the use of penalty functions or regularisers [809–811].

### Classical Method

GANs have been the leading choice for these tasks [812], using a generator to create artificial data and a discriminator to distinguish between genuine and synthetic samples [813]. Over the past decade, various extensions of GANs have improved performance. However, challenges such as training instability [814] and the considerable computational resources required to handle large-scale data persist [804,815]. VAEs have also been explored, showing promise in producing correlated asset returns and improve model's interpretability [816]. However, they can sometimes struggle with retaining sharp details of complex distributions [804].

### Quantum Approach

More recently, researchers have been exploring quantum computing for synthetic data generation [817]. For example, the use of QGANs [818] to reduce mode collapse and can potentially capture intricate financial patterns more fully. Other approaches, such as Quantum Boltzmann Machines, rely on quantum Hamiltonians to define their energy landscapes [819], which can help to explore complex financial dependencies more efficiently than classical methods. Quantum Annealing-based is also explored by using quantum tunnelling to sample from distributions that reflect realistic market conditions [9,820]. Despite promising theoretical insights, practical applications remain limited by hardware constraints and the often difficult task of mapping financial data to the relevant quantum models.

A proof-of-concept demonstration can be found in [196], where they used a QGAN framework to learn and approximate classical probability distributions, such as log-normal or bimodal distributions (which are commonly used in financial modelling) to load these distributions into quantum circuits efficiently. This approach can significantly reduce the resource overhead required for data loading (which is a bottleneck in quantum algorithms). The QGAN framework prepares quantum states that approximate these distributions with high fidelity by learning a quantum representation of the target probability distribution [818]. The main application involves pricing financial derivatives, specifically European call options [181], by integrating the learned quantum states that represent the learned asset price distribution into a QAE algorithm. This integration shows a potential pathway for quantum computing to transform computationally intensive financial tasks, providing faster and more accurate derivative pricing and risk management methods, even with near-term quantum devices [702]. However, as with other quantum-based models, current NISQ hardware limitations and the complexities of scaling remain a challenge [821].

Despite these challenges, quantum computing could improve SDG for finance as quantum technology improves [822]. Progress in QEC and hardware scalability may facilitate the development of more powerful quantum models that can handle large-scale financial data. This might deliver

improvements in areas such as high-frequency trading, stress testing, and modelling rare but critical market behaviour [805].

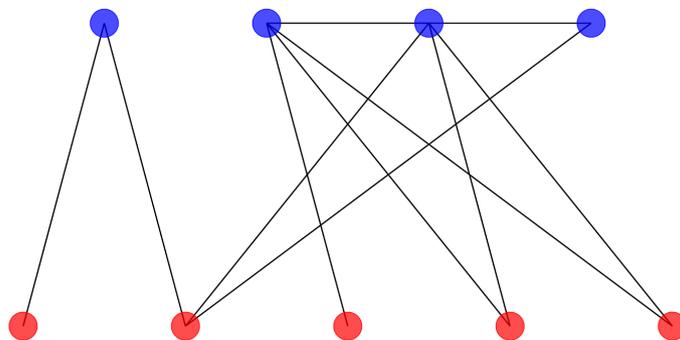
### 42.3. Predicting Financial Crises

A financial crisis is often associated with a panic or a bank run, during which investors sell off assets and withdraw cash from savings accounts, because they fear the value of those assets will drop, or the bank will collapse, respectively [823]. In such a crisis, asset prices see a steep decline in value, businesses and consumers are unable to pay their debts, and financial institutions experience liquidity shortages [824]. Other situations that may be labelled a financial crisis include the bursting of a speculative financial bubble, a stock market crash, a sovereign default, or a currency crisis [824]. A crisis may be limited to banks or spread throughout a single economy, the economy of a region, or economies worldwide.

The problem of predicting crises is therefore of great importance, and can be approached using a number of techniques, from modelling the evolution of markets, to detecting outlier events or identifying inconsistencies in the equilibrium of market values [824]. Mathematical models for predicting crises typically involves non-linear terms and interactions between many components [825]. These non-linear terms make prediction the behaviour difficult and computationally demanding for traditional classical methods. Below, we outline one promising approach that can be naturally mapped onto Quantum Annealing to potentially achieve a computational advantage [826].

#### Mathematical Formulation

The problem of predicting financial crises can be represented by a financial network graph [825] where nodes represent entities and the edges being how the worth of each entity is connected or depends upon other entities (see example in Figure 9). Analysing such networks is invaluable for predicting financial crises.



**Figure 9.** An example of a random financial network with 4 institutions (blue/top-row) and 5 assets (red/bottom-row).

Following the formulation in [346,825], we assume a financial network that has  $E$  entities and  $A$  assets. Let the vector  $\vec{w}_e \in \mathbb{R}_{\geq 0}^A$  denote the worth of each asset that the entity  $e \in [0, E - 1]$  owns. We also assume assets in  $A$  are co-owned by multiple entities and a matrix  $D \in \mathbb{R}_{\geq 0}^{E \times A}$  containing the percentage of ownership by an entity  $E$  of a co-owned asset. Furthermore, entities can have cross-holdings in other financial entities in the network, represented by a matrix  $C \in \mathbb{R}_{\geq 0}^{E \times E}$  and the self-ownership of an entity can be found via the diagonal matrix  $\tilde{C}$  such that  $\tilde{C}_{jj} := I - \sum_i C_{ij}$ . From the above definitions, it follows that one can define the market valuations for the assets associate with an entity as [825]:

$$\vec{v} = \tilde{C}(I - C)^{-1}(D\vec{w} - \vec{\lambda}(\vec{v}, \vec{w})), \quad (197)$$

where  $\vec{\lambda}$  is the *failure vector* that models a drop in the equity valuation including effects beyond a certain critical point. When such a failure occurs, there is generally loss of investor confidence as the entity is unable to cover operating costs. Hence, the goal is to find an equilibrium value for  $\vec{v}$  where

the entity is considered stable, which has been shown to be an NP-Hard optimisation problem [825] as the inter-dependencies of the network can be very complex for a large financial network.

### Quantum Approach

The first proposal to tackle this problem on quantum computers was by Orus et al. [827] where they reformulated it as a QUBO. First, they define a cost function:

$$F(\vec{v}) \equiv \left[ \vec{v} - \tilde{C}(I - C)^{-1} \left( D\vec{w} - \vec{\lambda}(\vec{v} - \vec{w}) \right) \right]^2 \geq 0. \quad (198)$$

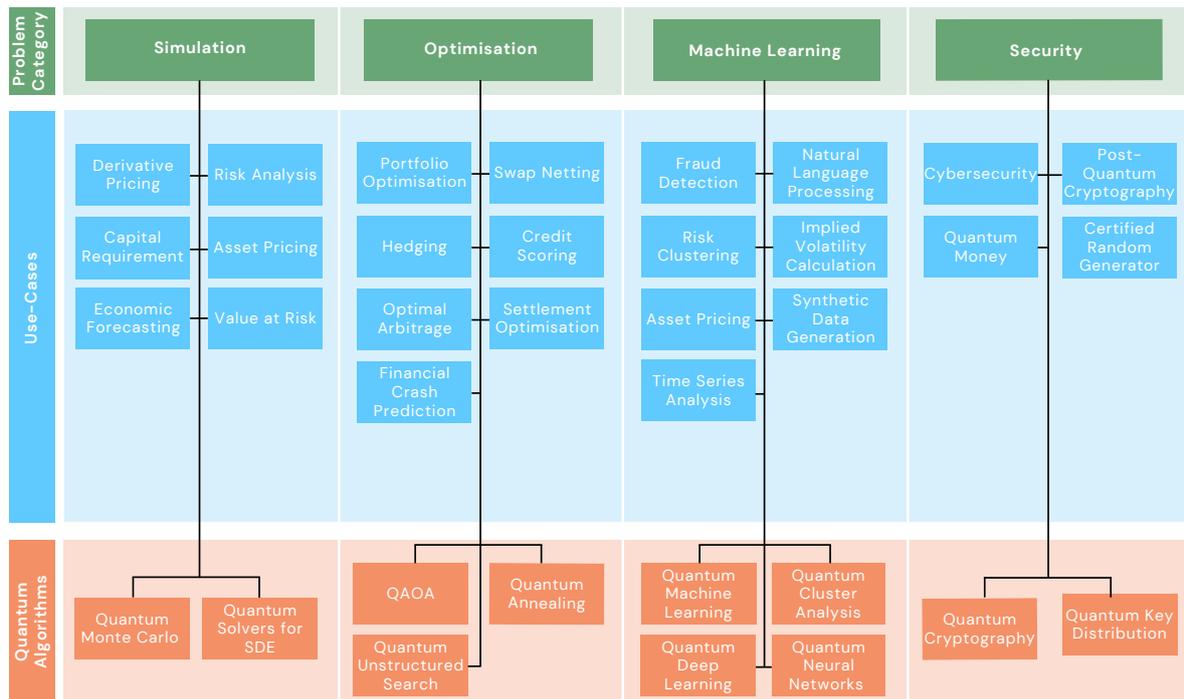
which is strictly larger than zero away from equilibrium and zero (i.e., minimum) at equilibrium value of  $\vec{v}$ . Hence, the classical cost function  $F(\vec{v})$  will be minimised when the vector  $\vec{v}$  is at equilibrium. One can approximate the values of the vector by binary variable  $v_i \approx \sum_{k=-q}^q 2^k x_{i,k}$  with  $2q + 1$  classical bits. It is straightforward to define an associated binary vector  $\mathbf{x}^*$  that gives the minimum for the following Higher-order Unconstrained Binary Optimisation (HUBO) cost function [827]:

$$f_Q(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j, \quad (199)$$

where  $Q$  incorporates all the non-linear constraints. As shown in [827], Equation (199) can then be approximated by Pauli-z matrices and two-qubit interactions<sup>43</sup> which reduces the problem into a QUBO that can be run on a QAnn to predict a potential massive failure of financial institutions after a small shock to the system. Ding et al. [826] also experimentally implemented this approach for assessing financial equilibrium but not the crash itself. Fellner et al. [828] formulated the problem in a similar way to Equation (199), but instead use gate-based computers and QAOA to solve it. The current bottleneck for these quantum methods is mapping the HUBO problem to available quantum hardware that is small in scale [826,828], hence we expect that future implementations to be different as the hardware scales up.

<sup>43</sup> The goal is to find an effective Hamiltonian with the same low-energy subspace and at most two-qubit interactions [827].

### 43. Summary and Outlook



**Figure 10.** A mapping between use-cases and applications for each of the main problem categories in quantum finance and economics.

In this review, we have surveyed some use-cases of quantum computing for finance and economics. Our approach is to first outline the technical constructions of the quantum algorithms in four problem domains that are relevant for finance and economics, namely: simulation, optimisation, machine learning, and cryptography. Following, we review the mathematical formulation of the use-cases (i.e., computational problems encountered in finance and economics) and map their quantum computing solution(s) to the quantum algorithms described in the first part. This is summarised in Figure 10, where the four problem categories (top row) are mapped to several use-cases (middle row), which are then mapped to pertinent quantum algorithms (bottom row). The benefit of this structure is that it makes it easier to work out the technical details of the use-case formulation/solution and quantum algorithm separately. This allows one to easily spot opportunities of reformulating the use-case into another problem class, and thus utilising a different quantum algorithm. For example, the option pricing problem is often formulated as computing an expectation value, which makes Quantum Monte Carlo Integration suitable as a solver. However, as discussed in Section (32.1), the same problem can be formulated as a Stochastic Differential Equation, which can be solved by quantum solvers for PDEs, which includes Hamiltonian simulation. Furthermore, we hope this structure will help experts in other technical fields, i.e., with little or no background in quantum computing, to collaborate and contribute in improving quantum algorithms and application so that we can achieve Practical Quantum Advantage (PQA). In the ensuing section, we will discuss a potential path to achieving PQA.

### 44. Path Towards Practical Quantum Advantage

The definition for PQA and related concepts is given Section (2). Here, we give an overview of some of the key ingredients towards achieving the goal of PQA in finance and economics. While about to complete this manuscript, we were made aware of Ref. [199] that proposes overlapping ideas for the path to PQA as described below.

### *Scaling Quantum Hardware*

The core limiting factor towards PQA is the scale and noise-tolerance of quantum hardware. A lot of hardware engineers, physicists, quantum theorists, control system specialists and application engineers are working to solve this challenge. The NQCC hosts an annual *Quantum Computing Scalability Conference* [829] which discusses various aspects of this challenge, including Quantum Error Correction (QEC), qubit and gate performance, integrated photonics & qubit addressability, cryo-engineering technology, control systems, and quantum computing networking. It is an incredible challenge to scale up quantum hardware to meet the requirements of practical applications, albeit significant progress has been made thus far.

Many hardware manufacturing companies, especially ones with full-stack services, have made their development road-map public. Most of the gate-based companies anchor their targets towards achieving Universal Fault-Tolerant (UFT) with net-positive QEC. The idea of QEC comes from the *quantum threshold theorem* (or quantum fault-tolerance theorem) which states that a gate-based quantum computer with a physical error rate below a certain threshold can, through application of QEC schemes, suppress the logical error rate to arbitrarily low levels [830–832]. This has been the foundational theory to why quantum computers can be made fault-tolerant, which is analogous to von Neumann's threshold theorem for classical computation [833]. Hence, for such systems, an important milestone in the NISQ era is proof-of-concept demonstrations of QEC, where a net positive benefit has already been achieved in a small scale experiments [834–838].

The next major milestone is for these systems to scale up to MegaQuOp machines capable of performing a million coherent operations [165]. This is along the path from NISQ-era to UFT. The ambitious goal of the UK's National Quantum Strategy Mission 1 is to have TeraQuOp quantum machines which are capable of performing a trillion coherent operations by year 2035 [839]. If that is achieved, the resulting productivity gains could provide significant economy-wide boosts, as projected by the Oxford Economics [19]. Along with key progress in the quantum stack, such milestones will more likely meet or exceeded McKinsey's \$622 billion projected value generated by quantum computing in finance by year 2035 [20].

Although other non-gate-based quantum computers do not have the same qubit-based road-maps, they still have scalability challenges. For example, photonic QPUs seek to scale *measurement based quantum computing* with large error-corrected cluster states [840]. Irrespective of the QPU modality, the scaling of quantum hardware and error correction are expected to be the keys that unlock Practical Quantum Advantage.

### *Application Benchmarks*

In the meantime, while we still do not have UFT quantum computers or large-scale Quantum Annealing, what will speed-up the advent of PQA is having readily accessible Up-to-date Application Benchmarks (UAB). As quantum computers are advancing, so do classical systems and associated algorithms. In some instances, where a quantum advantage has been claimed over the best known classical algorithm, researchers may find a classical method that achieves better results than originally compared to. A notable example is Ewin Tang's quantum-inspired classical algorithm for recommendation systems [146] which showed that the previously claimed quantum exponential speed-up by Kerenidis and Prakash [134] is no longer valid compared to her new algorithm. A similar case exists for practical applications benchmarks, when certain performance achievements are claimed by a quantum method, there may exist better classical implementations that achieves equal or better performance. A recent example is the ML vs QML benchmarking in [415] which found that across all experiments in that case, the out-of-the-box classical models systematically out performed the quantum models considered.

Therefore, it would be beneficial for speeding up research if there were central repositories with UAB for classical, hybrid, and quantum methods. We note that there are sites that have some benchmarking problems like Kaggle data-sets [841] for ML, sparse-matrix collections that arise in real-

world applications [842,843], and published benchmark papers for optimisation problems [233,236]. However, to the best of our knowledge we have not yet seen sites/publications with UAB that provides both problem data and state-of-the-art performance metrics for classical, hybrid, and quantum algorithms. This remains a gap in the community which if filled, would spur progress towards achieving PQA sooner than otherwise.

### *Quantum Middleware*

The term Quantum Middleware (QM) refers to the middle layers of the quantum computing stack, sitting between quantum hardware (which occupies the bottom of the stack) and quantum applications (which sit at the top) [844–846]. It is generally comprised of software packages that provide layers of abstraction which enable applications engineers to focus on developing solutions using quantum algorithms while masking hardware complexities.

In the path towards PQA, QM will become very important in the quantum ecosystem as it can potentially speed up and simplify the process of developing industry applications. Some key desirable features for QM are:

1. Tools for minimising quantum computation errors which include quantum error suppression, quantum error-mitigation, and Quantum Error Correction.
2. Circuit compilation tools that perform optimal circuit compression and embedding to specific quantum hardware.
3. Heterogeneous computation [847] which involves the use of different types of computing hardware such as CPUs, GPUs, and QPUs. The QM will essentially be an orchestrator that manages the execution of tasks on distributed computing resources.
4. Application libraries for various classes of problems with state-of-the-art algorithms.

As use-cases become increasingly complex and quantum hardware becomes more sophisticated, most applications engineers will rely on QM to speed-up the development process.

## **45. Quantum Algorithms**

In this review, we discussed the technical details of quantum algorithms in four problem domains: simulation, optimisation, machine learning, and cryptography. Here, we summarise and give key takeaways from these quantum algorithms.

### *Quantum Simulation*

Originally developed for solving problems that are quantum in nature [2,3], i.e. they can be expressed as Hamiltonian dynamics in Hilbert space, quantum simulation has been shown to be useful tool for non-quantum problems [180]. In Chapter (14), we focused on two classes of algorithms: QMCI and quantum solvers for SDE.

### *Quantum Monte Carlo Integration*

This is the quantum analogue of the classical Monte Carlo Integration which has the potential of a quadratic speed-up over its classical counterpart [5,193]. In general, the MCI is favoured in many industry applications, especially in finance and economics because they can better handle non-linear and high-dimensional problems as they scale independently of the problem size. However, as discussed in Section (15), MCI has a slow convergence, requiring a large number of samples for accuracy, which can be computationally intensive and time-consuming. Thus, the potential quadratic speed-up offered by QMCI is attractive to researchers, albeit it is not trivial to realise it practically. There are hardware and algorithmic challenges with QMCI, discussed in Section (15), which must be solved in order achieve PQA.

## Quantum Solvers for SDEs

The fields of finance and economics uses Stochastic Differential Equations to model quantities affected by randomness such as stock prices and economic crises. In Section (16) we discussed how SDEs can be reformulated as PDEs via the Feynman-Kac formula. Quantum solvers for PDEs can then be used such as ones based on finite-difference methods [848]. We also outlined more sophisticated methods of mapping the PDE to a Schrödinger-like equation [219] and then solving it via Hamiltonian simulation methods. It is currently not fully known if such methods can realise a PQA, especially for real-world problems [48], hence research and proof-of-concept demonstrations are ongoing.

## Quantum Optimisation

Quantum Optimisation, explored in Chapter (17), is another widely-studied application domain of quantum computing, with a huge diversity of use-cases across a range of industrial sectors [236]. A subset relevant to finance and economics are discussed in Chapters (30-40), which includes portfolio optimisation, optimal arbitrage identification, and others. Quantum algorithms such as the QAOA and VQE, as well as Quantum Annealing are common choices due to their flexibility, as well as their suitability for running on NISQ hardware. Similarly to the above, a number of recent papers have proposed a theoretical quantum advantage [236,849], but limitations posed by current hardware, as well as improving classical algorithms, have prevented incontrovertible proof of an empirical advantage. There also remain some open problems in the field, including dealing with barren plateaus [154] and ensuring constraint satisfaction [236].

## Quantum Machine Learning

As explored in Chapter (20), QML shows promise across different ML paradigms, from supervised classification and regression to unsupervised clustering and reinforcement learning.

A significant limitation in many QML algorithms is the assumption of efficient data loading into quantum states through QRAM to provide rapid access to large volumes of classical data [11,337]. While algorithms such as quantum k-means [299] and quantum nearest neighbour [327] report polynomial or even exponential speed-ups under QRAM assumptions [11], no fault-tolerant, scalable QRAM has yet been realised [341]. This could change as researchers collaboratively work on the development of QRAM [850]. Moreover, data encoding (e.g., amplitude or angle encoding) is non-trivial and may negate any theoretical speed-up if performed inefficiently [477]. Another challenge is the limitations of current hardware, with short coherence times, gate errors and highly affected by noise.

Adding to this is the lack of consistent algorithm and application benchmarks (such as UAB described above) for assessing quantum advantage in QML. Many QML demonstrations are on synthetic or contrived datasets with idealised assumptions about data encoding or oracle access [415,801]. Without consistent/accepted benchmarks, it remains challenging to ascertain whether demonstrated improvements reflect a genuine quantum advantage or merely misrepresents results. Nonetheless, identifying problem setups/formulations/types where quantum computing could provide an advantage, and the associated assumptions, is beneficial in-and-on itself both for problem/algorithm selection and bottlenecks identification.

## Quantum Cryptography

Cryptography and cybersecurity is one of the most well-known sectors of impact by quantum computing, with Shor's algorithm - see Section (26.1) - being widely discussed and researched since it was conceived [540]. As discussed in Chapter (24), most modern encryption relies on hard mathematical problems being intractable on classical computers. However, some of these problems are theoretically solvable on a fully fault-tolerant quantum computer in a reasonable amount of time. As a result, multiple national agencies and technology companies are performing intensive research into new and updated *quantum-safe* protocols which are resistant to such attacks [706]. This has culminated in the recent release of a few standard algorithms, which many organisations have already started incorporating [553-555]. Given the current resource estimates of quantum hardware requirements to

execute known quantum decryption algorithms (in terms of number of qubits, gate depth, and the likely requirement for QEC [542]), it seems unlikely that a cryptographically relevant decryption will happen in the next few years [543,544]. However, financial institutions which generally deals with sensitive data, it is recommended that they plan for a migration of their systems to quantum safe protocols [563]. A sooner migration will lower the risk of *harvest now, and decrypt later* type of security attacks [521].

## 46. Use-Cases

This technical review is dedicated to use-cases of quantum computing in finance and economics. These can be broadly categorise into use-cases that: (i) have a potential advantage, and (ii) a speculative adoption.

### *Potential Advantage*

Most of the quantum algorithms examined here exhibit theoretical speed-ups or potential better performance compared to current classical algorithms. Therefore, organisations or practitioners may find it optional to use quantum or hybrid quantum-classical approaches for their computational needs. If quantum computing is incorporated, the sector stands to benefit from two kinds of quantum advantage:

- *Speed-up*: we can broadly classify use-cases that depend of quantum simulation and quantum optimisation as top candidates to realise faster runtimes; see Chapters (14) and (17). These include use-cases in option pricing, portfolio optimisation, hedging, economic forecasting, etc; see Chapters (30) to (40). It should be noted that the analysis performed in this review does not take into account delays caused by internal processes and regulatory requirements. It might be possible that for some of these use-cases the difference in potential speed-up is not significant enough to matter when all the non-computational factors are considered. However, we expect that large organisations with global operations or companies with large-scale portfolios will need quantum solutions to remain competitive in fast-changing markets.
- *Accuracy*: in general, use-cases based on QML may have the benefit of better accuracy of model predictions compared to classical models; see Chapter (20). This has a huge impact to use-cases like fraud detection where the classification accuracy of the model is far more important than the speed of training [851]. The same rationale applies to risk analysis, economic forecasting, etc; see Chapters (35) and (40).

However, as noted in Section (44), both quantum hardware and algorithms would need to significantly improve to realise these benefits. Hence, in the long term, quantum computing is expected to provide direct computational advantages and also to deliver substantial indirect benefits by accelerating the development of novel computational methods and inspiring new insights in classical algorithm development.

### *Speculated Adoption*

In this review, we also considered use-cases of quantum computing whose adoption can only be speculated at this point. These use-cases include:

- *Quantum Money* [26] - a form of currency that is extremely difficult or even impossible to forge due to security guarantees from quantum mechanics.
- *Quantum Assets* [608] - a class of assets that are quantum in nature and traded in a quantum market enabled by quantum computers.

These use-cases represents the potential transformative impact of quantum computers beyond speed-up and accuracy benefits. The former use-case, quantum money, has a long history dating back to 1980s when Stephen Wiesner proposed the first formulation of private-key quantum money [577]. As discussed in Section (41), the field has grown over the years to the point that there now exist several experimental proof-of-concept demonstrations of quantum money [578,786–788]. A notable recent

experiment is that of Jiang et al. [578], where they reported a time advantage over optimal classical cross-checking protocols for transactions over significantly long inter-city distances. If these results are reproducible, they will stand as one of the first realisation PQA in finance and economics. This is because, the protocol could potentially be used for applications requiring high security, privacy and minimal transaction times, like financial trading and network control [578].

Therefore, one might speculate that quantum money might create a niche market of specialised usage. In the medium-term quantum money might become like crypto-currencies in the sense that it will only be utilised by a small fraction of the global market. However, irregardless of adoption to the economy, we expect that the concepts of both quantum money and quantum assets will feature directly or indirectly in many other technological advancements. For example, the authors in [783] have shown how quantum money can be coupled with block-chain technologies like Bitcoin to enhance the formation of smart contracts. Therefore, we predict that there is a significant chance of unforeseeable breakthroughs that would make these proposals become mainstream use-case in the global economy, thus are optimistic of such line of research.

#### 47. Further Research Directions

Given the findings and use-cases discussed, several important research directions seem promising for future exploration. For example, quantum computing methods for predicting financial crises [824,826] seem feasible in near-term quantum hardware and could have a significant impact. Furthermore, as global markets are becoming more interconnected, classical methods may soon reach the limit of what they could simulate in a reasonable time when considering large data-sets and huge number of constraints for optimisation problems.

Other interesting long-term use-cases are quantum money [25] and quantum assets [608]. While practical implementations of these use-cases are still in their infancy, their theoretical frameworks and/or experimental advancements are promising and certainly an area for further exploration.

The pace of the field is exemplified by the fact that while this manuscript was under internal review, a number of papers showing promising progress were published which include quantum enhanced computation of institutional algorithmic bond trading [852], sampling-based VQA for portfolio construction [853], regulatory considerations of quantum computing applications in financial services [854], a new record two-qubit gate fidelity of 99.99% by a trapped-ion quantum computer [855], and a claim by Google and collaborators of a Practical Quantum Advantage [856]. All these recent developments, including this year's Nobel Prize award in physics [857] being related to experiments that have contributed to quantum hardware, attest to the optimism that quantum computing will deliver real-world value. However we still maintain, as recently noted by Eisert and Preskill [858], that the field still has some major gaps to fill in the between NISQ and future fault-tolerant application-scale quantum computers that will unlock wide-spread Practical Quantum Advantage.

**Author Contributions:** Konstantinos Georgopoulos (KG) conceived the work and its structure, and carried out an initial survey of the landscape for fintech problems with potential solutions in quantum computing. The landscaping was then continued and completed by Manqoba Hlatshwayo (MQ), Manav Babel and Dalila Islas-Sanchez. MQ led the delivery of the technical content. All authors contributed equally to the writing of the manuscript. KG was responsible for overall supervision of the work. The manuscript was then proof-read and revised by all authors.

#### References

1. Phys, N.R. [40 years of quantum computing](https://doi.org/10.1038/s42254-021-00410-6). *Nature Reviews Physics* **2022**, *4*, 1–1. <https://doi.org/10.1038/s42254-021-00410-6>.
2. Feynman, R.P. [Simulating physics with computers](https://doi.org/10.1007/bf02650179). *International Journal of Theoretical Physics* **1982**, *21*, 467–488. <https://doi.org/10.1007/bf02650179>.
3. Lloyd, S. [Universal Quantum Simulators](https://doi.org/10.1126/science.273.5278.1073). *Science* **1996**, *273*, 1073–1078. <https://doi.org/10.1126/science.273.5278.1073>.

4. Grover, L.K. [A fast quantum mechanical algorithm for database search](#). In Proceedings of the Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212–219.
5. Montanaro, A. [Quantum speedup of Monte Carlo methods](#). *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2015**, 471, 20150301, <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2015.0301>. <https://doi.org/10.1098/rspa.2015.0301>.
6. Harrow, A.W.; Hassidim, A.; Lloyd, S. [Quantum Algorithm for Linear Systems of Equations](#). *Phys. Rev. Lett.* **2009**, 103, 150502. <https://doi.org/10.1103/PhysRevLett.103.150502>.
7. Berry, D.W. [High-Order Quantum Algorithm for Solving Linear Differential Equations](#). *Journal of Physics A: Mathematical and Theoretical* **2014**, 47, 105301.
8. QUANTUMPEDIA – The Quantum Encyclopedia. [A Brief History of Quantum Computing](#). Accessed 2025-07-15.
9. Orús, R.; Mugel, S.; Lizaso, E. [Forecasting financial crashes with quantum computing](#). *Phys. Rev. A* **2019**, 99, 060301. <https://doi.org/10.1103/PhysRevA.99.060301>.
10. Aaronson, S. [How Much Structure Is Needed for Huge Quantum Speedups?](#) *arXiv* **2022**, [arXiv:quant-ph/2209.06930]. <https://doi.org/10.48550/ARXIV.2209.06930>.
11. Aaronson, S. [Read the fine print](#). *Nature Physics* **2015**, 11, 291–293. <https://doi.org/10.1038/nphys3272>.
12. Bouland, A.; van Dam, W.; Joorati, H.; Kerenidis, I.; Prakash, A. [Prospects and challenges of quantum finance](#), 2020. <https://doi.org/10.48550/ARXIV.2011.06492>.
13. Preskill, J. [Quantum computing and the entanglement frontier](#). *arXiv preprint* **2012**, [arXiv:quant-ph/1203.5813]. <https://doi.org/10.48550/ARXIV.1203.5813>.
14. Arute, F.e.a. [Quantum supremacy using a programmable superconducting processor](#). *Nature* **2019**, 574, 505–510. <https://doi.org/10.1038/s41586-019-1666-5>.
15. Zhong, H.S.; Wang, H.; Deng, Y.H.; Chen, M.C.; Peng, L.C.; Luo, Y.H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. [Quantum computational advantage using photons](#). *Science* **2020**, 370, 1460–1463. <https://doi.org/10.1126/science.abe8770>.
16. Madsen, L.S.; Laudenbach, F.; Askarani, M.F.; Rortais, F.; Vincent, T.; Bulmer, J.F.F.; Miatto, F.M.; Neuhaus, L.; Helt, L.G.; Collins, M.J.; et al. [Quantum computational advantage with a programmable photonic processor](#). *Nature* **2022**, 606, 75–81. <https://doi.org/10.1038/s41586-022-04725-x>.
17. Gao, D.e.a. [Establishing a New Benchmark in Quantum Computational Advantage with 105-qubit Zhongchongzhi 3.0 Processor](#). *Physical Review Letters* **2025**, 134, 090601. <https://doi.org/10.1103/physrevlett.134.090601>.
18. Kim, Y.; Eddins, A.; Anand, S.; Wei, K.X.; van den Berg, E.; Rosenblatt, S.; Nayfeh, H.; Wu, Y.; Zaletel, M.; Temme, K.; et al. [Evidence for the utility of quantum computing before fault tolerance](#). *Nature* **2023**, 618, 500–505. <https://doi.org/10.1038/s41586-023-06096-3>.
19. Oxford Economics. [Ensuring that the UK can capture the benefits of quantum computing](#), 2025. Commissioned by IBM and Oxford Quantum Circuits.
20. Gschwendtner, M.; Morgan, N.; Soller, H. [Quantum technology use cases as fuel for value in finance](#), 2023. Last accessed: 14-10-2024.
21. Ménard, A.; Ostojic, I.; Patel, M.; Volz, D. [A game plan for quantum computing](#). *McKinsey Quarterly* **2020**.
22. Auer, R.A.; Angela, D.; Leonardo, G.; Joon, S.P.; Koji, T.; Andras, V. [Quantum computing and the financial system](#); Number no 149 in BIS papers, BIS, Bank for International Settlements: [Basel], 2024.
23. Egger, D.J.; Gambella, C.; Marecek, J.; McFaddin, S.; Mevissen, M.; Raymond, R.; Simonetto, A.; Woerner, S.; Yndurain, E. [Quantum Computing for Finance: State-of-the-Art and Future Prospects](#). *IEEE Transactions on Quantum Engineering* **2020**, 1, 1–24. <https://doi.org/10.1109/tqe.2020.3030314>.
24. Alcazar, J.; Leyton-Ortega, V.; Perdomo-Ortiz, A. [Classical versus quantum models in machine learning: insights from a finance application](#). *Machine Learning: Science and Technology* **2020**, 1, 035003. <https://doi.org/10.1088/2632-2153/ab9009>.
25. Hull, I.; Sattath, O.; Diamanti, E.; Wendin, G. [Quantum Technology for Economists](#). *arXiv* **2020**, [arXiv:econ.GN/2012.04473]. <https://doi.org/10.48550/ARXIV.2012.04473>.
26. Hull, I.; Sattath, O.; Diamanti, E.; Wendin, G. [Quantum Technology for Economists](#); Contributions to Economics, Imprint: Springer: Cham, 2024.
27. Pistoia, M.; Ahmad, S.F.; Ajagekar, A.; Buts, A.; Chakrabarti, S.; Herman, D.; Hu, S.; Jena, A.; Minssen, P.; Niroula, P.; et al. [Quantum Machine Learning for Finance ICCAD Special Session Paper](#). In Proceedings of the 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD). IEEE, 2021. <https://doi.org/10.1109/iccad51958.2021.9643469>.

28. García, D.P.; Cruz-Benito, J.; García-Peñalvo, F.J. Systematic Literature Review: Quantum Machine Learning and its applications. *arXiv* **2022**, [arXiv:quant-ph/2201.04093]. <https://doi.org/10.48550/ARXIV.2201.04093>.
29. Albareti, F.D.; Ankenbrand, T.; Bieri, D.; Hänggi, E.; Lötscher, D.; Stettler, S.; Schöngens, M. A Structured Survey of Quantum Computing for the Financial Industry. *arXiv* **2022**, [arXiv:q-fin.GN/2204.10026]. <https://doi.org/10.48550/ARXIV.2204.10026>.
30. Herman, D.; Googin, C.; Liu, X.; Sun, Y.; Galda, A.; Safro, I.; Pistoia, M.; Alexeev, Y. Quantum computing for finance. *Nature Reviews Physics* **2023**, *5*, 450–465. <https://doi.org/10.1038/s42254-023-00603-1>.
31. Gómez, A.; Leitão, A.; Manzano, A.; Musso, D.; Nogueiras, M.R.; Ordóñez, G.; Vázquez, C. A Survey on Quantum Computational Finance for Derivatives Pricing and VaR. *Archives of Computational Methods in Engineering* **2022**, *29*, 4137–4163. <https://doi.org/10.1007/s11831-022-09732-9>.
32. Chang, Y.J.; Sie, M.F.; Liao, S.W.; Chang, C.R. The Prospects of Quantum Computing for Quantitative Finance and Beyond. *IEEE Nanotechnology Magazine* **2023**, *17*, 31–37. <https://doi.org/10.1109/mnano.2023.3249501>.
33. Naik, A.S.; Yeniaras, E.; Hellstern, G.; Prasad, G.; Vishwakarma, S.K.L.P. From portfolio optimization to quantum blockchain and security: a systematic review of quantum computing in finance. *Financial Innovation* **2025**, *11*, 88. <https://doi.org/10.1186/s40854-025-00751-6>.
34. Saxena, A.; Mancilla, J.; Montalban, I.; Pere, C. *Financial modeling using quantum computing*, 1st ed. ed.; Packt Publishing: [Place of publication not identified], 2023. Includes bibliographical references and index.
35. Jacquier, A.; Oleksiy, K. *Quantum machine learning and optimisation in finance*, second edition. ed.; Packt Publishing Ltd.: Birmingham, UK, 2024. Includes bibliographical references and index.
36. Claudiu, B.; Cosmin, E.; Otniel, D.; Andrei, A., *Enhancing the Financial Sector with Quantum Computing: A Comprehensive Review of Current and Future Applications*. In *Proceedings of 22nd International Conference on Informatics in Economy (IE 2023)*; Springer Nature Singapore, 2024; pp. 195–203. [https://doi.org/10.1007/978-981-99-6529-8\\_17](https://doi.org/10.1007/978-981-99-6529-8_17).
37. Lu, Y.; Yang, J. Quantum financing system: A survey on quantum algorithms, potential scenarios and open research issues. *Journal of Industrial Information Integration* **2024**, *41*, 100663. <https://doi.org/10.1016/j.jii.2024.100663>.
38. Atadoga, A.; Ike, C.U.; Asuzu, O.F.; Ayinla, B.S.; Ndubuisi, N.L.; Adeleye, R.A. The Intersection of AI And Quantum Computing In Financial Markets: A Critical Review. *Computer Science; IT Research Journal* **2024**, *5*, 461–472. <https://doi.org/10.51594/csitrj.v5i2.816>.
39. Guju, Y.; Matsuo, A.; Raymond, R. Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications. *Physical Review Applied* **2024**, *21*, 067001. <https://doi.org/10.1103/physrevapplied.21.067001>.
40. Bunescu, L.; Vârtei, A.M. Modern finance through quantum computing—A systematic literature review. *PLOS ONE* **2024**, *19*, e0304317. <https://doi.org/10.1371/journal.pone.0304317>.
41. Mironowicz, P.; H., A.S.; Mandarino, A.; Yilmaz, A.E.; Ankenbrand, T. Applications of Quantum Machine Learning for Quantitative Finance. *arXiv* **2024**, [arXiv:quant-ph/2405.10119]. <https://doi.org/10.48550/ARXIV.2405.10119>.
42. Corli, S.; Moro, L.; Dragoni, D.; Dispenza, M.; Prati, E. Quantum machine learning algorithms for anomaly detection: A review. *Future Generation Computer Systems* **2025**, *166*, 107632. <https://doi.org/10.1016/j.future.2024.107632>.
43. Georgopoulos, K. Fintech Use-case Library, Quantum Readiness Delivery. Internal NQCC Report.
44. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press, 2010. <https://doi.org/10.1017/CBO9780511976667>.
45. Selinger, P. Quantum circuits of T-depth one. *Physical Review A* **2013**, *87*, 042302. <https://doi.org/10.1103/physreva.87.042302>.
46. Babbush, R.; McClean, J.R.; Newman, M.; Gidney, C.; Boixo, S.; Neven, H. Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage. *PRX Quantum* **2021**, *2*, 010103. <https://doi.org/10.1103/PRXQuantum.2.010103>.
47. Arnault, P.; Arrighi, P.; Herbert, S.; Kasneci, E.; Li, T. A typology of quantum algorithms, 2024, [arXiv:quant-ph/2407.05178].
48. Dalzell, A.M.; McArdle, S.; Berta, M.; Bienias, P.; Chen, C.F.; Gilyén, A.; Hann, C.T.; Kastoryano, M.J.; Khabiboulline, E.T.; Kubica, A.; et al. Quantum algorithms: A survey of applications and end-to-end complexities, 2023, [arXiv:quant-ph/2310.03011].

49. Shor, P.W. [Algorithms for quantum computation: discrete logarithms and factoring](#). In Proceedings of the Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 1994, pp. 124–134. <https://doi.org/10.1109/SFCS.1994.365700>.
50. Kitaev, A.Y. [Quantum measurements and the Abelian Stabilizer Problem](#), 1995, [arXiv:quant-ph/quant-ph/9511026].
51. Shor, P.W. [Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer](#). *Society for Industrial and Applied Mathematics* **1997**, 26, 1484–1509. <https://doi.org/10.1137/S0097539795293172>.
52. Abrams, D.S.; Lloyd, S. [Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors](#). *Physical Review Letters* **1999**, 83, 5162–5165. <https://doi.org/10.1103/physrevlett.83.5162>.
53. Griffiths, R.B.; Niu, C.S. [Semiclassical Fourier Transform for Quantum Computation](#). *Physical Review Letters* **1996**, 76, 3228–3231. <https://doi.org/10.1103/physrevlett.76.3228>.
54. Dobšiček, M.; Johansson, G.; Shumeiko, V.; Wendin, G. [Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark](#). *Phys. Rev. A* **2007**, 76, 030306. <https://doi.org/10.1103/PhysRevA.76.030306>.
55. Somma, R.D. [Quantum eigenvalue estimation via time series analysis](#). *New Journal of Physics* **2019**, 21, 123025. <https://doi.org/10.1088/1367-2630/ab5c60>.
56. Clinton, L.; Bausch, J.; Klassen, J.; Cubitt, T. [Phase estimation of local Hamiltonians on NISQ hardware](#). *New Journal of Physics* **2023**, 25, 033027. <https://doi.org/10.1088/1367-2630/acc26d>.
57. Brassard, G.; Høyer, P.; Mosca, M.; Tapp, A. [Quantum amplitude amplification and estimation](#), 2002. <https://doi.org/10.1090/conm/305/05215>.
58. Høyer, P. [Arbitrary phases in quantum amplitude amplification](#). *Physical Review A* **2000**, 62. <https://doi.org/10.1103/physreva.62.052304>.
59. Ambainis, A. [Variable time amplitude amplification and quantum algorithms for linear algebra problems](#). In Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012); Dürr, C.; Wilke, T., Eds., Dagstuhl, Germany, 2012; Vol. 14, *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 636–647. <https://doi.org/10.4230/LIPIcs.STACS.2012.636>.
60. Berry, D.W.; Childs, A.M.; Cleve, R.; Kothari, R.; Somma, R.D. [Exponential Improvement in Precision for Simulating Sparse Hamiltonians](#). In Proceedings of the Proceedings of the forty-sixth annual ACM symposium on Theory of computing, New York, NY, USA, 2014; STOC '14, p. 283–292. <https://doi.org/10.1145/2591796.2591854>.
61. Yoder, T.J.; Low, G.H.; Chuang, I.L. [Fixed-Point Quantum Search with an Optimal Number of Queries](#). *Physical Review Letters* **2014**, 113, 210501. <https://doi.org/10.1103/physrevlett.113.210501>.
62. Martyn, J.M.; Rossi, Z.M.; Tan, A.K.; Chuang, I.L. [Grand Unification of Quantum Algorithms](#). *PRX Quantum* **2021**, 2, 040203. <https://doi.org/10.1103/PRXQuantum.2.040203>.
63. Toyozumi, K.; Yamamoto, N.; Hoshino, K. [Hamiltonian simulation using the quantum singular-value transformation: Complexity analysis and application to the linearized Vlasov-Poisson equation](#). *Physical Review A* **2024**, 109, 012430. <https://doi.org/10.1103/physreva.109.012430>.
64. Giurgica-Tiron, T.; Kerenidis, I.; Labib, F.; Prakash, A.; Zeng, W. [Low depth algorithms for quantum amplitude estimation](#), 2020. <https://doi.org/10.48550/ARXIV.2012.03348>.
65. Suzuki, Y.; Uno, S.; Raymond, R.; Tanaka, T.; Onodera, T.; Yamamoto, N. [Amplitude estimation without phase estimation](#). *Quantum Information Processing* **2020**, 19. <https://doi.org/10.1007/s11128-019-2565-2>.
66. Nakaji, K. [Faster Amplitude Estimation](#). *Quantum Information & Computation* **2020**, 20, 1109–1123, [arXiv:quant-ph/2003.02417]. <https://doi.org/10.26421/qic20.13-14-2>.
67. Grinko, D.; Gacon, J.; Zoufal, C.; Woerner, S. [Iterative quantum amplitude estimation](#). *npj Quantum Information* **2021**, 7. <https://doi.org/10.1038/s41534-021-00379-1>.
68. Plekhanov, K.; Rosenkranz, M.; Fiorentini, M.; Lubasch, M. [Variational quantum amplitude estimation](#). *Quantum* **2022**, 6, 670. <https://doi.org/10.22331/q-2022-03-17-670>.
69. Shu, G.; Shan, Z.; Xu, J.; Zhao, J.; Wang, S. [A general quantum algorithm for numerical integration](#). *Scientific Reports* **2024**, 14. <https://doi.org/10.1038/s41598-024-61010-9>.
70. Rotello, C. [Quantum algorithm for approximating the expected value of a random-exist quantified oracle](#). *arXiv* **2024**, [arXiv:quant-ph/2412.00567]. <https://doi.org/10.48550/ARXIV.2412.00567>.
71. Aaronson, S.; Rall, P. [Quantum Approximate Counting, Simplified](#). In *Symposium on Simplicity in Algorithms*; Society for Industrial and Applied Mathematics, 2020; pp. 24–32. <https://doi.org/10.1137/1.9781611976014.5>.

72. Cerf, N.J.; Grover, L.K.; Williams, C.P. Nested quantum search and structured problems. *Physical Review A* **2000**, *61*, 032303. <https://doi.org/10.1103/physreva.61.032303>.
73. Bulger, D.; Baritomba, W.P.; Wood, G.R. Implementing Pure Adaptive Search with Grover's Quantum Algorithm. *Journal of Optimization Theory and Applications* **2003**, *116*, 517–529. <https://doi.org/10.1023/a:1023061218864>.
74. Armenakas, A.E.; Baker, O.K. Application of a Quantum Search Algorithm to High- Energy Physics Data at the Large Hadron Collider. *arXiv* **2020**, [arXiv:quant-ph/2010.00649]. <https://doi.org/10.48550/ARXIV.2010.00649>.
75. Tezuka, H.; Nakaji, K.; Satoh, T.; Yamamoto, N. Grover search revisited: Application to image pattern matching. *Physical Review A* **2022**, *105*, 032440. <https://doi.org/10.1103/physreva.105.032440>.
76. Fernández, P.; Martín-Delgado, M.A. Implementing the Grover algorithm in homomorphic encryption schemes. *Physical Review Research* **2024**, *6*, 043109. <https://doi.org/10.1103/physrevresearch.6.043109>.
77. Zhang, K.; Rao, P.; Yu, K.; Lim, H.; Korepin, V. Implementation of efficient quantum search algorithms on NISQ computers. *Quantum Information Processing* **2021**, *20*. <https://doi.org/10.1007/s11128-021-03165-2>.
78. Zhang, K.; Yu, K.; Korepin, V. Quantum search on noisy intermediate-scale quantum devices. *Europhysics Letters* **2022**, *140*, 18002. <https://doi.org/10.1209/0295-5075/ac90e6>.
79. Liu, C.Y. Practical Quantum Search by Variational Quantum Eigensolver on Noisy Intermediate-Scale Quantum Hardware. In Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2023, pp. 397–403. <https://doi.org/10.1109/csci62032.2023.00071>.
80. Piron, R.; Goursaud, C. Hybrid Grover Search for AUD on a NISQ Device. In Proceedings of the 32nd European Signal Processing Conference (EUSIPCO), 2024, pp. 2102–2106. <https://doi.org/10.23919/EUSIPCO63174.2024.10714952>.
81. Spitzer, F. *Principles of random walk*, 2. ed., 1. softcover printing ed.; Number 34 in Graduate texts in mathematics, Springer: New York, NY, 2001. Literaturverz. S. [395] - 402.
82. László, L. Random Walks on Graphs: A Survey, Combinatorics, Paul Erdos is Eighty. *Bolyai Soc. Math. Stud.* **1993**, *2*.
83. Norris, J.R. *Markov Chains*; Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997. <https://doi.org/10.1017/CBO9780511810633>.
84. Black, F.; Scholes, M. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* **1973**, *81*, 637–654. <https://doi.org/10.1086/260062>.
85. Merton, R.C. Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science* **1973**, *4*, 141. <https://doi.org/10.2307/3003143>.
86. Aharonov, Y.; Davidovich, L.; Zagury, N. Quantum random walks. *Physical Review A* **1993**, *48*, 1687–1690. <https://doi.org/10.1103/physreva.48.1687>.
87. Kempe, J. Quantum random walks - an introductory overview. *Contemporary Physics*, Vol. 44 (4), p.307-327, **2003** **2003**, *44*, 307–327, [arXiv:quant-ph/quant-ph/0303081]. <https://doi.org/10.1080/00107151031000110776>.
88. Farhi, E.; Gutmann, S. Quantum computation and decision trees. *Physical Review A* **1998**, *58*, 915–928. <https://doi.org/10.1103/physreva.58.915>.
89. Ambainis, A.; Bach, E.; Nayak, A.; Vishwanath, A.; Watrous, J. One-Dimensional Quantum Walks. In Proceedings of the Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2001; STOC '01, p. 37–49. <https://doi.org/10.1145/380752.380757>.
90. Douglas, B.L.; Wang, J.B. Efficient quantum circuit implementation of quantum walks. *Phys. Rev. A* **2009**, *79*, 052335. <https://doi.org/10.1103/PhysRevA.79.052335>.
91. Krovi, H. Symmetry in quantum walks, 2007. <https://doi.org/10.48550/ARXIV.0711.1694>.
92. Georgopoulos, K.; Emary, C.; Zuliani, P. Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers. *Physical Review A* **2021**, *103*. <https://doi.org/10.1103/physreva.103.022408>.
93. Aharonov, D.; Ambainis, A.; Kempe, J.; Vazirani, U. Quantum Walks on Graphs. In Proceedings of the Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2001; STOC '01, p. 50–59. <https://doi.org/10.1145/380752.380758>.
94. Magniez, F.; Nayak, A.; Roland, J.; Santha, M. Search via Quantum Walk. *SIAM Journal on Computing* **2011**, *40*, 142–164. <https://doi.org/10.1137/090745854>.
95. Szegedy, M. Quantum speed-up of Markov chain based algorithms. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004, pp. 32–41. <https://doi.org/10.1109/FOCS.2004.53>.

96. Ambainis, A.; Kempe, J.; Rivosh, A. [Coins Make Quantum Walks Faster](#). In Proceedings of the Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, USA, 2005; SODA '05, p. 1099–1108.
97. Shenvi, N.; Kempe, J.; Whaley, K.B. [Quantum random-walk search algorithm](#). *Phys. Rev. A* **2003**, *67*, 052307. <https://doi.org/10.1103/PhysRevA.67.052307>.
98. Reitzner, D.; Nagaj, D.; Bužek, V. [Quantum Walks](#). *Acta Physica Slovaca. Reviews and Tutorials* **2011**, *61*. <https://doi.org/10.2478/v10155-011-0006-6>.
99. Levin, D.A.; Peres, Y.; Wilmer, E.L. *Markov chains and mixing times*; American Mathematical Society: Providence, Rhode Island, 2009. Includes bibliographical references and indexes. Description based on print version record.
100. Moore, C.; Russell, A. [Quantum Walks on the Hypercube](#). In Proceedings of the Randomization and Approximation Techniques in Computer Science; Rolim, J.D.P.; Vadhan, S., Eds., Berlin, Heidelberg, 2002; pp. 164–178.
101. Ambainis, A. [Quantum Walk Algorithm for Element Distinctness](#). *SIAM Journal on Computing* **2007**, *37*, 210–239, [<https://doi.org/10.1137/S0097539705447311>]. <https://doi.org/10.1137/S0097539705447311>.
102. Ambainis, A.; Gilyén, A.; Jeffery, S.; Kokainis, M., [Quadratic Speedup for Finding Marked Vertices by Quantum Walks](#). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*; Association for Computing Machinery: New York, NY, USA, 2020; p. 412–424.
103. Apers, S.; Gilyén, A.; Jeffery, S. [A Unified Framework of Quantum Walk Search](#), 2019. <https://doi.org/10.48550/ARXIV.1912.04233>.
104. Childs, A.M.; Eisenberg, J.M. [Quantum Algorithms for Subset Finding](#). *Quantum Info. Comput.* **2005**, *5*, 593–604.
105. Somma, R.D.; Boixo, S.; Barnum, H.; Knill, E. [Quantum Simulations of Classical Annealing Processes](#). *Phys. Rev. Lett.* **2008**, *101*, 130504. <https://doi.org/10.1103/PhysRevLett.101.130504>.
106. Razzoli, L.; Cenedese, G.; Bondani, M.; Benenti, G. [Efficient Implementation of Discrete-Time Quantum Walks on Quantum Computers](#). *Entropy* **2024**, *26*, 313. <https://doi.org/10.3390/e26040313>.
107. Chawla, P.; Singh, S.; Agarwal, A.; Srinivasan, S.; Chandrashekar, C.M. [Multi-qubit quantum computing using discrete-time quantum walks on closed graphs](#). *Scientific Reports* **2023**, *13*. <https://doi.org/10.1038/s41598-023-39061-1>.
108. Nandi, B.; Singha, S.; Datta, A.; Saha, A.; Chakrabarti, A. [Robust implementation of discrete-time quantum walks in any finite-dimensional quantum system](#). *Modern Physics Letters A* **2025**, *40*. <https://doi.org/10.1142/s021773232550035x>.
109. Ruan, Y.; Marsh, S.; Xue, X.; Li, X.; Liu, Z.; Wang, J. [Quantum approximate algorithm for NP optimization problems with constraints](#), 2020. <https://doi.org/10.48550/ARXIV.2002.00943>.
110. Qiang, X.; Ma, S.; Song, H. [Quantum Walk Computing: Theory, Implementation, and Application](#). *Intelligent Computing* **2024**, *3*. <https://doi.org/10.34133/icomputing.0097>.
111. Childs, A.M.; Cleve, R.; Deotto, E.; Farhi, E.; Gutmann, S.; Spielman, D.A. [Exponential Algorithmic Speedup by a Quantum Walk](#). In Proceedings of the Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2003; STOC '03, p. 59–68. <https://doi.org/10.1145/780542.780552>.
112. Childs, A.M. [Universal Computation by Quantum Walk](#). *Physical Review Letters* **2009**, *102*, 180501. <https://doi.org/10.1103/physrevlett.102.180501>.
113. Lovett, N.B.; Cooper, S.; Everitt, M.; Trevers, M.; Kendon, V. [Universal quantum computation using the discrete-time quantum walk](#). *Physical Review A* **2010**, *81*, 042330. <https://doi.org/10.1103/physreva.81.042330>.
114. Childs, A.M.; van Dam, W. [Quantum algorithms for algebraic problems](#). *Rev. Mod. Phys.* **2010**, *82*, 1–52. <https://doi.org/10.1103/RevModPhys.82.1>.
115. Venegas-Andraca, S.E. [Quantum walks: a comprehensive review](#). *Quantum Information Processing* **2012**, *11*, 1015–1106. <https://doi.org/10.1007/s11128-012-0432-5>.
116. Klebaner, F.C. *Introduction to Stochastic Calculus with Applications*, 3rd ed.; Imperial College Press, 2012; [<https://www.worldscientific.com/doi/pdf/10.1142/p821>]. <https://doi.org/10.1142/p821>.
117. Mohseni, M.; Rebentrost, P.; Lloyd, S.; Aspuru-Guzik, A. [Environment-assisted quantum walks in photosynthetic energy transfer](#). *The Journal of Chemical Physics* **2008**, *129*. <https://doi.org/10.1063/1.3002335>.
118. Giri, M.K.; Mandal, S.B. [Realizing topological quantum walks on NISQ digital quantum computer](#). *Physica Scripta* **2025**, *100*, 035111. <https://doi.org/10.1088/1402-4896/adb244>.

119. Spedicato, E. *Computer Algorithms for Solving Linear Algebraic Equations*; Number v.77 in Nato asi Subseries F: Ser., Springer Berlin / Heidelberg: Berlin, Heidelberg, 1991. Description based on publisher supplied metadata and other sources.
120. Allgower, E.L.; Georg, K., *Introduction to Numerical Continuation Methods*; Society for Industrial and Applied Mathematics, 2003; chapter 4, pp. 28–36. <https://doi.org/10.1137/1.9780898719154.ch4>.
121. Gould, N.I.M.; Scott, J.A.; Hu, Y. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Transactions on Mathematical Software* **2007**, *33*, 10. <https://doi.org/10.1145/1236463.1236465>.
122. Shewchuk, J.R. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Technical report, Carnegie Mellon University, USA, 1994.
123. Le Gall, F. *Powers of tensors and fast matrix multiplication*. In Proceedings of the Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation. ACM, 2014, ISSAC '14, pp. 296–303. <https://doi.org/10.1145/2608628.2608664>.
124. Ambainis, A. *Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations*, 2010. <https://doi.org/10.48550/ARXIV.1010.4458>.
125. Childs, A.M.; Kothari, R.; Somma, R.D. *Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision*. *SIAM Journal on Computing* **2017**, *46*, 1920–1950, [<https://doi.org/10.1137/16M1087072>]. <https://doi.org/10.1137/16M1087072>.
126. Wossnig, L.; Zhao, Z.; Prakash, A. *Quantum Linear System Algorithm for Dense Matrices*. *Phys. Rev. Lett.* **2018**, *120*, 050502. <https://doi.org/10.1103/PhysRevLett.120.050502>.
127. Somma, R.D.; Boixo, S. *Spectral Gap Amplification*. *SIAM Journal on Computing* **2013**, *42*, 593–610. <https://doi.org/10.1137/120871997>.
128. Subaşı, Y.; Somma, R.D.; Orsucci, D. *Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing*. *Phys. Rev. Lett.* **2019**, *122*, 060504. <https://doi.org/10.1103/PhysRevLett.122.060504>.
129. An, D.; Lin, L. *Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm*. *ACM Transactions on Quantum Computing* **2019**, *3*, 1–28, [arXiv:quant-ph/1909.05500]. <https://doi.org/10.1145/3498331>.
130. Lin, L.; Tong, Y. *Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems*. *Quantum* **2020**, *4*, 361. <https://doi.org/10.22331/q-2020-11-11-361>.
131. Costa, P.C.; An, D.; Sanders, Y.R.; Su, Y.; Babbush, R.; Berry, D.W. *Optimal Scaling Quantum Linear-Systems Solver via Discrete Adiabatic Theorem*. *PRX Quantum* **2022**, *3*, 040303. <https://doi.org/10.1103/PRXQuantum.3.040303>.
132. Dalzell, A.M. *A shortcut to an optimal quantum linear system solver*, 2024, [arXiv:quant-ph/2406.12086].
133. Dervovic, D.; Herbster, M.; Mountney, P.; Severini, S.; Usher, N.; Wossnig, L. *Quantum linear systems algorithms: a primer*, 2018. <https://doi.org/10.48550/ARXIV.1802.08227>.
134. Kerenidis, I.; Prakash, A. *Quantum Recommendation Systems*, 2016, [arXiv:quant-ph/1603.08675].
135. Dabrowski, A., Ed. *2013 signal processing: algorithms, architectures, arrangements, and applications (SPA)*; IEEE: Piscataway, NJ, 2013. Literaturangaben.
136. Geron, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*, third edition ed.; Data science / machine learning, O'Reilly: Beijing, 2023. Literaturangaben. - Index.
137. Kerenidis, I.; Prakash, A. *Quantum gradient descent for linear systems and least squares*. *Phys. Rev. A* **2020**, *101*, 022316. <https://doi.org/10.1103/PhysRevA.101.022316>.
138. Gilyén, A.; Su, Y.; Low, G.H.; Wiebe, N. *Quantum Singular Value Transformation and beyond: Exponential Improvements for Quantum Matrix Arithmetics*. In Proceedings of the Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, New York, NY, USA, 2019; STOC 2019, p. 193–204. <https://doi.org/10.1145/3313276.3316366>.
139. Chakraborty, S.; Gilyén, A.; Jeffery, S. *The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation*. In Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019); Baier, C.; Chatzigiannakis, I.; Flocchini, P.; Leonardi, S., Eds., Dagstuhl, Germany, 2019; Vol. 132, *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 33:1–33:14. <https://doi.org/10.4230/LIPIcs.ICALP.2019.33>.
140. Gribling, S.; Kerenidis, I.; Szilágyi, D. *Improving quantum linear system solvers via a gradient descent perspective*, 2021. <https://doi.org/10.48550/ARXIV.2109.04248>.

141. Barata, J.C.A.; Hussein, M.S. The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory. *Brazilian Journal of Physics* **2011**, *42*, 146–165. <https://doi.org/10.1007/s13538-011-0052-z>.
142. Kerenidis, I.; Prakash, A. A Quantum Interior Point Method for LPs and SDPs. *ACM Transactions on Quantum Computing* **2020**, *1*. <https://doi.org/10.1145/3406306>.
143. Kiani, B.T.; De Palma, G.; Englund, D.; Kaminsky, W.; Marvian, M.; Lloyd, S. Quantum Advantage for Differential Equation Analysis. *Physical Review A* **2022**, *105*, 022415, [arxiv:math-ph, physics:quant-ph/2010.15776]. <https://doi.org/10.1103/PhysRevA.105.022415>.
144. Hann, C.T.; Lee, G.; Girvin, S.; Jiang, L. Resilience of Quantum Random Access Memory to Generic Noise. *PRX Quantum* **2021**, *2*, 020311. <https://doi.org/10.1103/prxquantum.2.020311>.
145. Frieze, A.; Kannan, R.; Vempala, S. Fast Monte-Carlo Algorithms for Finding Low-Rank Approximations. *J. ACM* **2004**, *51*, 1025–1041. <https://doi.org/10.1145/1039488.1039494>.
146. Tang, E. A Quantum-Inspired Classical Algorithm for Recommendation Systems. In Proceedings of the Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, New York, NY, USA, 2019; STOC 2019, p. 217–228. <https://doi.org/10.1145/3313276.3316310>.
147. Chia, N.H.; Gilyén, A.; Li, T.; Lin, H.H.; Tang, E.; Wang, C., Sampling-Based Sublinear Low-Rank Matrix Arithmetic Framework for Dequantizing Quantum Machine Learning. In Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing; Association for Computing Machinery: New York, NY, USA, 2020; p. 387–400.
148. Le Gall, F. Robust Dequantization of the Quantum Singular Value Transformation and Quantum Machine Learning Algorithms. *computational complexity* **2025**, *34*. <https://doi.org/10.1007/s00037-024-00262-3>.
149. Sakurai, J.J.; Napolitano, J. *Modern quantum mechanics*, third edition ed.; Cambridge University Press: Cambridge, United Kingdom, 2021.
150. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O’Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* **2014**, *5*. <https://doi.org/10.1038/ncomms5213>.
151. Wecker, D.; Hastings, M.B.; Troyer, M. Progress towards practical quantum variational algorithms. *Physical Review A* **2015**, *92*. <https://doi.org/10.1103/physreva.92.042303>.
152. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nature Reviews Physics* **2021**, *3*, 625–644. <https://doi.org/10.1038/s42254-021-00348-9>.
153. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nature Communications* **2018**, *9*. <https://doi.org/10.1038/s41467-018-07090-4>.
154. Larocca, M.; Thanasilp, S.; Wang, S.; Sharma, K.; Biamonte, J.; Coles, P.J.; Cincio, L.; McClean, J.R.; Holmes, Z.; Cerezo, M. Barren plateaus in variational quantum computing. *Nature Reviews Physics* **2025**, *7*, 174–189. <https://doi.org/10.1038/s42254-025-00813-9>.
155. Holmes, Z.; Sharma, K.; Cerezo, M.; Coles, P.J. Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus. *PRX Quantum* **2022**, *3*, 010313. <https://doi.org/10.1103/prxquantum.3.010313>.
156. Cerezo, M.; Larocca, M.; García-Martín, D.; Diaz, N.L.; Braccia, P.; Fontana, E.; Rudolph, M.S.; Bermejo, P.; Ijaz, A.; Thanasilp, S.; et al. Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing. *arXiv* **2023**, [arXiv:quant-ph/2312.09121]. <https://doi.org/10.48550/ARXIV.2312.09121>.
157. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm, 2014. <https://doi.org/10.48550/ARXIV.1411.4028>.
158. Bravo-Prieto, C.; LaRose, R.; Cerezo, M.; Subasi, Y.; Cincio, L.; Coles, P.J. Variational Quantum Linear Solver, 2019. <https://doi.org/10.48550/ARXIV.1909.05820>.
159. Huang, H.Y.; Bharti, K.; Rebentrost, P. Near-term quantum algorithms for linear systems of equations, 2019. <https://doi.org/10.48550/ARXIV.1909.07344>.
160. Lubasch, M.; Joo, J.; Moinier, P.; Kiffner, M.; Jaksch, D. Variational quantum algorithms for nonlinear problems. *Phys. Rev. A* **2020**, *101*, 010301. <https://doi.org/10.1103/PhysRevA.101.010301>.
161. Kubo, K.; Nakagawa, Y.O.; Endo, S.; Nagayama, S. Variational Quantum Simulations of Stochastic Differential Equations. *Physical Review A* **2021**, *103*, 052425, [arxiv:quant-ph/2012.04429]. <https://doi.org/10.1103/PhysRevA.103.052425>.
162. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309. <https://doi.org/10.1103/PhysRevA.98.032309>.

163. Bittel, L.; Kliesch, M. Training Variational Quantum Algorithms Is NP-Hard. *Phys. Rev. Lett.* **2021**, *127*, 120502. <https://doi.org/10.1103/PhysRevLett.127.120502>.
164. Huembeli, P.; Dauphin, A. Characterizing the loss landscape of variational quantum circuits. *Quantum Science and Technology* **2021**, *6*, 025011. <https://doi.org/10.1088/2058-9565/abdbc9>.
165. Preskill, J. Beyond NISQ: The Megaquop Machine. *ACM Transactions on Quantum Computing* **2025**, *6*, 1–7. <https://doi.org/10.1145/3723153>.
166. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355–5363. <https://doi.org/10.1103/PhysRevE.58.5355>.
167. Farhi, E.; Goldstone, J.; Gutmann, S.; Sipser, M. Quantum Computation by Adiabatic Evolution, 2000. <https://doi.org/10.48550/ARXIV.QUANT-PH/0001106>.
168. Albash, T.; Lidar, D.A. Adiabatic quantum computation. *Rev. Mod. Phys.* **2018**, *90*, 015002. <https://doi.org/10.1103/RevModPhys.90.015002>.
169. Glover, F.; Kochenberger, G.; Du, Y. A Tutorial on Formulating and Using QUBO Models, 2018. <https://doi.org/10.48550/ARXIV.1811.11538>.
170. Crosson, E.J.; Lidar, D.A. Prospects for quantum enhancement with diabatic quantum annealing. *Nature Reviews Physics* **2021**, *3*, 466–489. <https://doi.org/10.1038/s42254-021-00313-6>.
171. Biamonte, J.D.; Love, P.J. Realizable Hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A* **2008**, *78*, 012352. <https://doi.org/10.1103/PhysRevA.78.012352>.
172. Laarhoven, P.J.M.; Aarts, E.H.L. *Simulated annealing: theory and applications*; Mathematics and its applications, Reidel, 1987.
173. Denchev, V.S.; Boixo, S.; Isakov, S.V.; Ding, N.; Babbush, R.; Smelyanskiy, V.; Martinis, J.; Neven, H. What is the Computational Value of Finite-Range Tunneling? *Phys. Rev. X* **2016**, *6*, 031015. <https://doi.org/10.1103/PhysRevX.6.031015>.
174. Kechedzhi, K.; Smelyanskiy, V.N. Open-System Quantum Annealing in Mean-Field Models with Exponential Degeneracy. *Phys. Rev. X* **2016**, *6*, 021028. <https://doi.org/10.1103/PhysRevX.6.021028>.
175. Cohen, E.; Tamir, B. D-Wave and predecessors: From simulated to quantum annealing. *International Journal of Quantum Information* **2014**, *12*, 1430002. <https://doi.org/10.1142/s0219749914300022>.
176. Streif, M.; Neukart, F.; Leib, M., Solving Quantum Chemistry Problems with a D-Wave Quantum Annealer. In *Quantum Technology and Optimization Problems*; Springer International Publishing, 2019; pp. 111–122. [https://doi.org/10.1007/978-3-030-14082-3\\_10](https://doi.org/10.1007/978-3-030-14082-3_10).
177. Ajagekar, A.; Humble, T.; You, F. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Computers and Chemical Engineering* **2020**, *132*, 106630. <https://doi.org/10.1016/j.compchemeng.2019.106630>.
178. Phillipson, F.; Bhatia, H.S., Portfolio Optimisation Using the D-Wave Quantum Annealer. In *Computational Science – ICCS 2021*; Springer International Publishing, 2021; pp. 45–59. [https://doi.org/10.1007/978-3-030-77980-1\\_4](https://doi.org/10.1007/978-3-030-77980-1_4).
179. Carugno, C.; Ferrari Dacrema, M.; Cremonesi, P. Evaluating the job shop scheduling problem on a D-wave quantum annealer. *Scientific Reports* **2022**, *12*. <https://doi.org/10.1038/s41598-022-10169-0>.
180. An, D.; Liu, J.P.; Wang, D.; Zhao, Q. A Theory of Quantum Differential Equation Solvers: Limitations and Fast-Forwarding. *arXiv* **2023**. <https://doi.org/10.48550/arXiv.2211.05246>.
181. Cochrane, J.H. *Asset Pricing*; Princeton University Press: s.l., 2009. Description based upon print version of record.
182. Glasserman, P. *Monte Carlo Methods in Financial Engineering*; Applications of mathematics: stochastic modelling and applied probability, Springer, 2004.
183. Bagherimehrab, M.; Nakaji, K.; Wiebe, N.; Aspuru-Guzik, A. Fast Quantum Algorithm for Differential Equations. *arXiv* **2023**, [arXiv:quant-ph/2306.11802]. <https://doi.org/10.48550/arXiv.2306.11802>.
184. Neal, R.M. Probabilistic inference using Markov chain Monte Carlo methods, 1993. Department of Computer Science, University of Toronto Toronto, ON, Canada, Last accessed: 20-05-2022.
185. de Mello, T.H.; Bayraksan, G. Monte Carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science* **2014**, *19*, 56–85. <https://doi.org/10.1016/j.sorms.2014.05.001>.
186. Christian P. Robert, G.C. *Monte Carlo Statistical Methods*; Vol. 2, Springer, 2004.
187. Chakrabarti, S.; Krishnakumar, R.; Mazzola, G.; Stamatopoulos, N.; Woerner, S.; Zeng, W.J. A Threshold for Quantum Advantage in Derivative Pricing. *Quantum* **2021**, *5*, 463. <https://doi.org/10.22331/q-2021-06-01-463>.

188. Stamatopoulos, N.; Zeng, W.J. Derivative Pricing using Quantum Signal Processing. *Quantum* **2024**, *8*, 1322. <https://doi.org/10.22331/q-2024-04-30-1322>.
189. Plesch, M.; Brukner, i.c.v. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A* **2011**, *83*, 032302. <https://doi.org/10.1103/PhysRevA.83.032302>.
190. Grover, L.; Rudolph, T. Creating superpositions that correspond to efficiently integrable probability distributions, 2002. <https://doi.org/10.48550/ARXIV.QUANT-PH/0208112>.
191. Herbert, S. No quantum speedup with Grover-Rudolph state preparation for quantum Monte Carlo integration. *Phys. Rev. E* **2021**, *103*, 063302. <https://doi.org/10.1103/PhysRevE.103.063302>.
192. McArdle, S.; Gilyén, A.; Berta, M. Quantum state preparation without coherent arithmetic. *arXiv* **2022**, [arXiv:quant-ph/2210.14892]. <https://doi.org/10.48550/ARXIV.2210.14892>.
193. Akhalwaya, I.Y.; Connolly, A.; Guichard, R.; Herbert, S.; Kargi, C.; Krajenbrink, A.; Lubasch, M.; Keever, C.M.; Sorci, J.; Spranger, M.; et al. A Modular Engine for Quantum Monte Carlo Integration. *arXiv* **2023**, [arXiv:quant-ph/2308.06081]. <https://doi.org/10.48550/ARXIV.2308.06081>.
194. Gomes, N.; Mukherjee, A.; Zhang, F.; Iadecola, T.; Wang, C.Z.; Ho, K.M.; Orth, P.P.; Yao, Y.X. Adaptive Variational Quantum Imaginary Time Evolution Approach for Ground State Preparation. *Advanced Quantum Technologies* **2021**, *4*, 2100114, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/qute.202100114]. <https://doi.org/10.1002/qute.202100114>.
195. Rattew, A.G.; Sun, Y.; Minssen, P.; Pistoia, M. The Efficient Preparation of Normal Distributions in Quantum Registers. *Quantum* **2021**, *5*, 609. <https://doi.org/10.22331/q-2021-12-23-609>.
196. Zoufal, C.; Lucchi, A.; Woerner, S. Quantum Generative Adversarial Networks for learning and loading random distributions. *npj Quantum Information* **2019**, *5*. <https://doi.org/10.1038/s41534-019-0223-2>.
197. Pereira, A.; Villarino, A.; Cortines, A.; Mugel, S.; Orus, R.; Beltran, V.L.; Scursulim, J.V.S.; Brito, S. Encoding of Probability Distributions for Quantum Monte Carlo Using Tensor Networks. *arXiv* **2024**, [arXiv:quant-ph/2411.11660]. <https://doi.org/10.48550/ARXIV.2411.11660>.
198. Herbert, S. Quantum Monte Carlo Integration: The Full Advantage in Minimal Circuit Depth. *Quantum* **2022**, *6*, 823. <https://doi.org/10.22331/q-2022-09-29-823>.
199. Lanes, O.; Beji, M.; Corcoles, A.D.; Dalyac, C.; Gambetta, J.M.; Henriët, L.; Javadi-Abhari, A.; Kandala, A.; Mezzacapo, A.; Porter, C.; et al. A Framework for Quantum Advantage. *arXiv* **2025**, [arXiv:quant-ph/2506.20658]. <https://doi.org/10.48550/ARXIV.2506.20658>.
200. Evans, L.C. *An introduction to stochastic differential equations*; American Mathematical Society: Providence, RI, 2014. Description based upon print version of record.
201. Heath, D.; Jarrow, R.; Morton, A. Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation. *Econometrica* **1992**, *60*, 77. <https://doi.org/10.2307/2951677>.
202. Heston, S.L. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies* **1993**, *6*, 327–343. <https://doi.org/10.1093/rfs/6.2.327>.
203. Wilmott, P. *Paul Wilmott Introduces Quantitative Finance*; Wiley & Sons, Incorporated, John, 2013; p. 728.
204. Wilmott, P. *Paul Wilmott on Quantitative Finance*; Wiley & Sons, Incorporated, John, 2013; p. 1500.
205. Feynman, R.P. Space-Time Approach to Non-Relativistic Quantum Mechanics. *Reviews of Modern Physics* **1948**, *20*, 367–387. <https://doi.org/10.1103/revmodphys.20.367>.
206. Kac, M. On distributions of certain Wiener functionals. *Transactions of the American Mathematical Society* **1949**, *65*, 1–13.
207. Del Moral, P., *Feynman-Kac Formulae*. In *Feynman-Kac Formulae*; Springer New York, 2004; pp. 47–93. [https://doi.org/10.1007/978-1-4684-9393-1\\_2](https://doi.org/10.1007/978-1-4684-9393-1_2).
208. Mazumder, S. *Numerical methods for partial differential equations*; Academic Press, Imprint of Elsevier: London, 2016.
209. Larson, M.G.; Bengzon, F. *The Finite Element Method: Theory, Implementation, and Applications*; Vol. 10, *Texts in Computational Science and Engineering*, Springer: Berlin, Heidelberg, 2013. <https://doi.org/10.1007/978-3-642-33287-6>.
210. Xu, S.; Zhao, J.; Wu, H.; Zhang, S.; Müller, J.D.; Huang, H.; Rahmati, M.; Wang, D. A Review of Solution Stabilization Techniques for RANS CFD Solvers. *Aerospace* **2023**, *10*, 230. <https://doi.org/10.3390/aerospace10030230>.
211. Clader, B.D.; Jacobs, B.C.; Sprouse, C.R. Preconditioned Quantum Linear System Algorithm. *Physical Review Letters* **2013**, *110*, 250504. <https://doi.org/10.1103/PhysRevLett.110.250504>.
212. Shao, C.; Xiang, H. Quantum circulant preconditioner for a linear system of equations. *Phys. Rev. A* **2018**, *98*, 062321. <https://doi.org/10.1103/PhysRevA.98.062321>.

213. Tong, Y.; An, D.; Wiebe, N.; Lin, L. Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions. *Physical Review A* **2021**, *104*, 032422. <https://doi.org/10.1103/physreva.104.032422>.
214. Nazareth, J.L. Conjugate Gradient Method. *WIREs Computational Statistics* **2009**, *1*, 348–353. <https://doi.org/10.1002/wics.13>.
215. Chang, M.H. *Quantum stochastic*, first published 2015 ed.; Number 37 in Cambridge series on statistical and probabilistic mathematics, Cambridge University Press: New York, 2015. Title from publisher's bibliographic system (viewed on 05 Oct 2015).
216. Jin, S.; Liu, N.; Wei, W. Quantum Algorithms for Stochastic Differential Equations: A Schrödingerisation Approach. *Journal of Scientific Computing* **2025**, *104*. <https://doi.org/10.1007/s10915-025-02970-6>.
217. Gonzalez-Conde, J.; Rodriguez-Rozas, A.; Solano, E.; Sanz, M. Efficient Hamiltonian simulation for solving option price dynamics. *Physical Review Research* **2023**, *5*, 043220. <https://doi.org/10.1103/physrevresearch.5.043220>.
218. Hunacek, M. Lie groups: an introduction through linear groups, by Wulf Rossmann. Pp. 265. £29.95 (pbk). 2006. ISBN 0 19 920251 6 (Oxford University Press). *The Mathematical Gazette* **2008**, *92*, 380–382. <https://doi.org/10.1017/s0025557200183561>.
219. Jin, S.; Liu, N.; Yu, Y. Quantum simulation of partial differential equations: Applications and detailed analysis. *Physical Review A* **2023**, *108*, 032603. <https://doi.org/10.1103/physreva.108.032603>.
220. McArdle, S.; Jones, T.; Endo, S.; Li, Y.; Benjamin, S.C.; Yuan, X. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information* **2019**, *5*. <https://doi.org/10.1038/s41534-019-0187-2>.
221. Yeter-Aydeniz, K.; Moschandreou, E.; Siopsis, G. Quantum imaginary-time evolution algorithm for quantum field theories with continuous variables. *Physical Review A* **2022**, *105*, 012412. <https://doi.org/10.1103/physreva.105.012412>.
222. Kolotouros, I.; Joseph, D.; Narayanan, A.K. Accelerating quantum imaginary-time evolution with random measurements. *Physical Review A* **2025**, *111*, 012424. <https://doi.org/10.1103/physreva.111.012424>.
223. Jouzdani, P.; Johnson, C.W.; Mucciolo, E.R.; Stetcu, I. Alternative approach to quantum imaginary time evolution. *Physical Review A* **2022**, *106*, 062435. <https://doi.org/10.1103/physreva.106.062435>.
224. Pavliotis, G.A. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*; Springer New York, 2014. <https://doi.org/10.1007/978-1-4939-1323-7>.
225. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer, 2006.
226. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press, 2004. <https://doi.org/10.1017/CBO9780511804441>.
227. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680, [<https://www.science.org/doi/pdf/10.1126/science.220.4598.671>]. <https://doi.org/10.1126/science.220.4598.671>.
228. Markowitz, H. Portfolio Selection. *The Journal of Finance* **1952**, *7*, 77–91.
229. Shapiro, A.; Dentcheva, D.; Ruszczyński, A.P. *Lectures on stochastic programming : modeling and theory*, third edition. ed.; MOS-SIAM series on optimization, Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104: Philadelphia, Pennsylvania, 2021. <https://doi.org/10.1137/1.9780898718751>.
230. et al., N.M. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology* **2018**, *3*, 030503. <https://doi.org/10.1088/2058-9565/aab822>.
231. et al., A.K. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242–246. <https://doi.org/10.1038/nature23879>.
232. Guerreschi, G.G.; Smelyanskiy, M. Practical optimization for hybrid quantum-classical algorithms, 2017, [[arXiv:quant-ph/1701.01450](https://arxiv.org/abs/1701.01450)].
233. Koch, T.; Neira, D.E.B.; Chen, Y.; Cortiana, G.; Egger, D.J.; Heese, R.; Hegade, N.N.; Cadavid, A.G.; Huang, R.; Itoko, T.; et al. Quantum Optimization Benchmark Library – The Intractable Decathlon, 2025, [[arXiv:quant-ph/2504.03832](https://arxiv.org/abs/2504.03832)]. <https://doi.org/10.48550/ARXIV.2504.03832>.
234. Hromkovič, J. *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*; Springer Science & Business Media, 2013.
235. Nemhauser, G.; Wolsey, L., *Linear Programming*. In *Integer and Combinatorial Optimization*; John Wiley and Sons, Ltd, 1988; chapter I.2, pp. 27–49, [<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118627372.ch2>]. <https://doi.org/10.1002/9781118627372.ch2>.

236. Abbas, A.e.a. Challenges and opportunities in quantum optimization. *Nature Reviews Physics* **2024**, *6*, 718–735. <https://doi.org/10.1038/s42254-024-00770-9>.
237. Kochenberger, G.; Hao, J.K.; Glover, F.; Lewis, M.; Lü, Z.; Wang, H.; Wang, Y. The Unconstrained Binary Quadratic Programming Problem: A Survey. *J. Comb. Optim.* **2014**, *28*, 58–81. <https://doi.org/10.1007/s10878-014-9734-0>.
238. Okada, S.; Ohzeki, M.; Taguchi, S. Efficient partition of integer optimization problems with one-hot encoding. *Scientific Reports* **2019**, *9*. <https://doi.org/10.1038/s41598-019-49539-6>.
239. Lucas, A. Ising formulations of many NP problems. *Frontiers in Physics* **2014**, *2*. <https://doi.org/10.3389/fphy.2014.00005>.
240. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics* **2015**, *18*.
241. Rayleigh, J. In finding the correction for the open end of an organ-pipe. *Phil. Trans.* **1870**, *161*, 77.
242. Ritz, W. Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *Ann. Phys.* **1909**, *1909*, 1–61. <https://doi.org/doi:10.1515/crll.1909.135.1>.
243. Tilly, J.; Chen, H.; Cao, S.; Picozzi, D.; Setia, K.; Li, Y.; Grant, E.; Wossnig, L.; Rungger, I.; Booth, G.H.; et al. The Variational Quantum Eigensolver: a review of methods and best practices, 2021. <https://doi.org/10.48550/ARXIV.2111.05176>.
244. Rattew, A.G.; Hu, S.; Pistoia, M.; Chen, R.; Wood, S. A Domain-agnostic, Noise-resistant, Hardware-efficient Evolutionary Variational Quantum Eigensolver, 2019. <https://doi.org/10.48550/ARXIV.1910.09694>.
245. Amaro, D.; Modica, C.; Rosenkranz, M.; Fiorentini, M.; Benedetti, M.; Lubasch, M. Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology* **2022**, *7*, 015021. <https://doi.org/10.1088/2058-9565/ac3e54>.
246. Fujii, K.; Mizuta, K.; Ueda, H.; Mitarai, K.; Mizukami, W.; Nakagawa, Y.O. Deep Variational Quantum Eigensolver: a divide-and-conquer method for solving a larger problem with smaller size quantum computers, 2020. <https://doi.org/10.48550/ARXIV.2007.10917>.
247. Motta, M.; Sun, C.; Tan, A.T.K.; O'Rourke, M.J.; Ye, E.; Minnich, A.J.; Brandão, F.G.S.L.; Chan, G.K.L. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics* **2019**, *16*, 205–210. <https://doi.org/10.1038/s41567-019-0704-4>.
248. Gomes, N.; Zhang, F.; Berthussen, N.F.; Wang, C.Z.; Ho, K.M.; Orth, P.P.; Yao, Y. Efficient Step-Merged Quantum Imaginary Time Evolution Algorithm for Quantum Chemistry. *Journal of Chemical Theory and Computation* **2020**, *16*, 6256–6266, [<https://doi.org/10.1021/acs.jctc.0c00666>]. PMID: 32877181, <https://doi.org/10.1021/acs.jctc.0c00666>.
249. Sun, S.N.; Motta, M.; Tazhigulov, R.N.; Tan, A.T.; Chan, G.K.L.; Minnich, A.J. Quantum Computation of Finite-Temperature Static and Dynamical Properties of Spin Systems Using Quantum Imaginary Time Evolution. *PRX Quantum* **2021**, *2*, 010317. <https://doi.org/10.1103/PRXQuantum.2.010317>.
250. Huang, Y.; Shao, Y.; Ren, W.; Sun, J.; Lv, D. Efficient quantum imaginary time evolution by drifting real time evolution: an approach with low gate and measurement complexity, 2022. <https://doi.org/10.48550/ARXIV.2203.11112>.
251. Yuan, X.; Endo, S.; Zhao, Q.; Li, Y.; Benjamin, S.C. Theory of variational quantum simulation. *Quantum* **2019**, *3*, 191. <https://doi.org/10.22331/q-2019-10-07-191>.
252. Grimsley, H.R.; Economou, S.E.; Barnes, E.; Mayhall, N.J. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications* **2019**, *10*. <https://doi.org/10.1038/s41467-019-10988-2>.
253. Ryabinkin, I.G.; Yen, T.C.; Genin, S.N.; Izmaylov, A.F. Qubit Coupled Cluster Method: A Systematic Approach to Quantum Chemistry on a Quantum Computer. *Journal of Chemical Theory and Computation* **2018**, *14*, 6317–6326, [<https://doi.org/10.1021/acs.jctc.8b00932>]. PMID: 30427679, <https://doi.org/10.1021/acs.jctc.8b00932>.
254. Poulin, D.; Qarry, A.; Somma, R.; Verstraete, F. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Phys. Rev. Lett.* **2011**, *106*, 170501. <https://doi.org/10.1103/PhysRevLett.106.170501>.
255. Alghassi, H.; Deshmukh, A.; Ibrahim, N.; Robles, N.; Woerner, S.; Zoufal, C. A variational quantum algorithm for the Feynman-Kac formula. *Quantum* **2022**, *6*, 730. <https://doi.org/10.22331/q-2022-06-07-730>.
256. Galda, A.; Liu, X.; Lykov, D.; Alexeev, Y.; Safro, I. Transferability of optimal QAOA parameters between random graphs. In Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), 2021, pp. 171–180. <https://doi.org/10.1109/QCE52317.2021.00034>.

257. Shaydulin, R.; Safro, I.; Larson, J. [Multistart Methods for Quantum Approximate optimization](#). In Proceedings of the 2019 IEEE High Performance Extreme Computing Conference (HPEC), 2019, pp. 1–8. <https://doi.org/10.1109/HPEC.2019.8916288>.
258. Shaydulin, R.; Hadfield, S.; Hogg, T.; Safro, I. [Classical symmetries and the Quantum Approximate Optimization Algorithm](#). *Quantum Information Processing* **2021**, *20*. <https://doi.org/10.1007/s11128-021-03298-4>.
259. Streif, M.; Leib, M. [Training the Quantum Approximate Optimization Algorithm without access to a Quantum Processing Unit](#), 2019. <https://doi.org/10.48550/ARXIV.1908.08862>.
260. Durr, C.; Hoyer, P. [A Quantum Algorithm for Finding the Minimum](#), 1996. <https://doi.org/10.48550/ARXIV.QUANT-PH/9607014>.
261. Zeng, Y.; Dong, Z.; Wang, H.; He, J.; Huang, Q.; Chang, S. [A general quantum minimum searching algorithm with high success rate and its implementation](#). *Science China Physics, Mechanics and Astronomy* **2023**, *66*. <https://doi.org/10.1007/s11433-022-2060-3>.
262. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. [Tight Bounds on Quantum Searching](#). *Fortschritte der Physik* **1998**, *46*, 493–505, [https://doi.org/10.1002/\(SICI\)1521-3978\(199806\)46:4/5<493::AID-PROP493>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P).
263. Gilliam, A.; Woerner, S.; Gonciulea, C. *Quantum* **2021**, *5*, 428. <https://doi.org/10.22331/q-2021-04-08-428>.
264. Morapakula, S.N.; Deshpande, S.; Yata, R.; Ubale, R.; Wad, U.; Ikeda, K. [End-to-End Portfolio Optimization with Quantum Annealing](#), 2025, [arXiv:quant-ph/2504.08843].
265. Kim, S.; Ahn, S.W.; Suh, I.S.; Dowling, A.W.; Lee, E.; Luo, T. [Quantum Annealing for Combinatorial Optimization: A Benchmarking Study](#), 2025, [arXiv:quant-ph/2504.06201].
266. Guéry-Odelin, D.; Ruschhaupt, A.; Kiely, A.; Torrontegui, E.; Martínez-Garaot, S.; Muga, J. [Shortcuts to adiabaticity: Concepts, methods, and applications](#). *Reviews of Modern Physics* **2019**, *91*. <https://doi.org/10.103/revmodphys.91.045001>.
267. Gomez-Tejedor, A.; Osaba, E.; Villar-Rodriguez, E. [Addressing the Minor-Embedding Problem in Quantum Annealing and Evaluating State-of-the-Art Algorithm Performance](#), 2025, [arXiv:quant-ph/2504.13376].
268. Ohzeki, M. [Breaking limitation of quantum annealer in solving optimization problems under constraints](#), 2020. <https://doi.org/10.48550/ARXIV.2002.05298>.
269. DWave. [minorminer documentation](#). Accessed 06-08-2025.
270. DWave. [Annealing Implementation and Controls](#), 2025. Accessed: 2025-08-04.
271. Sachdeva, N.; Hartnett, G.S.; Maity, S.; Marsh, S.; Wang, Y.; Winick, A.; Dougherty, R.; Canuto, D.; Chong, Y.Q.; Hush, M.; et al. [Quantum optimization using a 127-qubit gate-model IBM quantum computer can outperform quantum annealers for nontrivial binary optimization problems](#). *arXiv* **2024**, [arXiv:quant-ph/2406.01743]. <https://doi.org/10.48550/ARXIV.2406.01743>.
272. McGeoch, C.C.; Chern, K.; Farré, P.; King, A.K. [A comment on comparing optimization on D-Wave and IBM quantum processors](#). *arXiv* **2024**, [arXiv:quant-ph/2406.19351]. <https://doi.org/10.48550/ARXIV.2406.19351>.
273. del Campo, A. [Shortcuts to Adiabaticity by Counterdiabatic Driving](#). *Physical Review Letters* **2013**, *111*. <https://doi.org/10.1103/physrevlett.111.100502>.
274. Cadavid, A.G.; Dalal, A.; Simen, A.; Solano, E.; Hegade, N.N. [Bias-field digitized counterdiabatic quantum optimization](#). *Physical Review Research* **2025**, *7*. <https://doi.org/10.1103/physrevresearch.7.1022010>.
275. Visuri, A.M.; Cadavid, A.G.; Bhargava, B.A.; Romero, S.V.; Grabarits, A.; Chandarana, P.; Solano, E.; del Campo, A.; Hegade, N.N. [Digitized counterdiabatic quantum critical dynamics](#), 2025, [arXiv:quant-ph/2502.15100]. <https://doi.org/10.48550/ARXIV.2502.15100>.
276. Romero, S.V.; Cadavid, A.G.; Nikačević, P.; Solano, E.; Hegade, N.N.; Lopez-Ruiz, M.A.; Giroto, C.; Yamada, M.; Barkoutsos, P.K.; Kaushik, A.; et al. [Protein folding with an all-to-all trapped-ion quantum computer](#), 2025, [arXiv:quant-ph/2506.07866].
277. Jordan, S.P. [Quantum Computation Beyond the Circuit Model](#), 2008. <https://doi.org/10.48550/ARXIV.0809.2307>.
278. Brandao, F.G.; Svore, K.M. [Quantum Speed-Ups for Solving Semidefinite Programs](#). In Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), 2017, pp. 415–426. <https://doi.org/10.1109/FOCS.2017.45>.
279. Van Apeldoorn, J.; Gilyén, A.; Gribling, S.; de Wolf, R. [Quantum SDP-Solvers: Better Upper and Lower Bounds](#). In Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), 2017, pp. 403–414. <https://doi.org/10.1109/FOCS.2017.44>.

280. Brandão, F.G.S.L.; Kaveh, A.; Li, T.; Lin, C.Y.Y.; Svore, K.M.; Wu, X. **Quantum SDP Solvers: Large Speed-Ups, Optimality, and Applications to Quantum Learning**. In Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019); Baier, C.; Chatzigiannakis, I.; Flocchini, P.; Leonardi, S., Eds., Dagstuhl, Germany, 2019; Vol. 132, *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 27:1–27:14. <https://doi.org/10.4230/LIPIcs.ICALP.2019.27>.
281. van Apeldoorn, J.; Gilyén, A. **Improvements in Quantum SDP-Solving with Applications**. In Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019); Baier, C.; Chatzigiannakis, I.; Flocchini, P.; Leonardi, S., Eds., Dagstuhl, Germany, 2019; Vol. 132, *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 99:1–99:15. <https://doi.org/10.4230/LIPIcs.ICALP.2019.99>.
282. Arkadi Nemirovski. **Interior Point Polynomial Time Methods in Convex Programming (Lecture Notes)**, 2004. Accessed: 2025-08-07.
283. Nesterov, Y.; Nemirovskii, A. *Interior-Point Polynomial Algorithms in Convex Programming*; Society for Industrial and Applied Mathematics, 1994; [<https://epubs.siam.org/doi/pdf/10.1137/1.9781611970791>]. <https://doi.org/10.1137/1.9781611970791>.
284. Nannicini, G. **Fast Quantum Subroutines for the Simplex Method**. In Proceedings of the Integer Programming and Combinatorial Optimization; Singh, M.; Williamson, D.P., Eds., Cham, 2021; pp. 311–325.
285. Kerenidis, I.; Prakash, A.; Szilágyi, D. **Quantum algorithms for Second-Order Cone Programming and Support Vector Machines**. *Quantum* **2021**, 5, 427. <https://doi.org/10.22331/q-2021-04-08-427>.
286. Jordan, S.P. **Fast Quantum Algorithm for Numerical Gradient Estimation**. *Phys. Rev. Lett.* **2005**, 95, 050501. <https://doi.org/10.1103/PhysRevLett.95.050501>.
287. Das Sarma, S.; Deng, D.L.; Duan, L.M. **Machine learning meets quantum physics**. *Physics Today* **2019**, 72, 48–54, [[https://pubs.aip.org/physicstoday/article-pdf/72/3/48/10121955/48\\_1\\_online.pdf](https://pubs.aip.org/physicstoday/article-pdf/72/3/48/10121955/48_1_online.pdf)]. <https://doi.org/10.1063/PT.3.4164>.
288. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. **Quantum machine learning**. *Nature* **2017**, 549, 195–202. <https://doi.org/10.1038/nature23474>.
289. Schuld, M.; Sinayskiy, I.; Petruccione, F. **An introduction to quantum machine learning**. *Contemporary Physics* **2014**, 56, 172–185. <https://doi.org/10.1080/00107514.2014.964942>.
290. Lamichhane, P.; Rawat, D.B. **Quantum Machine Learning: Recent Advances, Challenges, and Perspectives**. *IEEE Access* **2025**, 13, 94057–94105. <https://doi.org/10.1109/ACCESS.2025.3573244>.
291. Dunjko, V.; Briegel, H.J. **Machine learning & artificial intelligence in the quantum domain: a review of recent progress**. *Reports on Progress in Physics* **2018**, 81, 074001. <https://doi.org/10.1088/1361-6633/aab406>.
292. Schuld, M.; Killoran, N. **Is Quantum Advantage the Right Goal for Quantum Machine Learning?** *PRX Quantum* **2022**, 3, 030101. <https://doi.org/10.1103/PRXQuantum.3.030101>.
293. Lu, W.; Lu, Y.; Li, J.; Sigov, A.; Ratkin, L.; Ivanov, L.A. **Quantum machine learning: Classifications, challenges, and solutions**. *Journal of Industrial Information Integration* **2024**, 42, 100736. <https://doi.org/10.1016/j.jii.2024.100736>.
294. Wang, Y.; Liu, J. **A comprehensive review of quantum machine learning: from NISQ to fault tolerance**. *Reports on Progress in Physics* **2024**, 87, 116402. <https://doi.org/10.1088/1361-6633/ad7f69>.
295. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press, 2014.
296. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer-Verlag: Berlin, Heidelberg, 2006.
297. An, Q.; Rahman, S.; Zhou, J.; Kang, J.J. **A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges**. *Sensors* **2023**, 23. <https://doi.org/10.3390/s23094178>.
298. IBM. **What Is Supervised Learning?**, 2024. Accessed: 25-07-2025.
299. Lloyd, S.; Mohseni, M.; Rebentrost, P. **Quantum algorithms for supervised and unsupervised machine learning**. *arXiv* **2013**, [arXiv:quant-ph/1307.0411]. <https://doi.org/10.48550/ARXIV.1307.0411>.
300. Macaluso, A. **Quantum Supervised Learning**. *KI - Künstliche Intelligenz* **2024**, 38, 277–291. <https://doi.org/10.1007/s13218-024-00856-7>.
301. Li, W.; Deng, D.L. **Recent advances for quantum classifiers**. *Science China Physics, Mechanics & Astronomy*, 65, 220301 (2022) **2021**, 65, [arXiv:quant-ph/2108.13421]. <https://doi.org/10.1007/s11433-021-1793-6>.
302. Rebentrost, P.; Mohseni, M.; Lloyd, S. **Quantum Support Vector Machine for Big Data Classification**. *Physical Review Letters* **2014**, 113, 130503. <https://doi.org/10.1103/physrevlett.113.130503>.
303. Innan, N.; Khan, M.; Panda, B.; Bennai, M. **Enhancing quantum support vector machines through variational kernel training**. *Quantum Information Processing* **2023**, 22. <https://doi.org/10.1007/s11128-023-04138-3>.

304. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. [A training algorithm for optimal margin classifiers](#). In Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992, COLT92. <https://doi.org/10.1145/130385.130401>.
305. Suthaharan, S. [Support Vector Machine](#). In *Integrated Series in Information Systems*; Springer US, 2016; pp. 207–235. [https://doi.org/10.1007/978-1-4899-7641-3\\_9](https://doi.org/10.1007/978-1-4899-7641-3_9).
306. Platt, J. [Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines](#). Technical Report MSR-TR-98-14, Microsoft, 1998.
307. Mehrotra, S. [On the Implementation of a Primal-Dual Interior Point Method](#). *SIAM Journal on Optimization* **1992**, *2*, 575–601, [<https://doi.org/10.1137/0802028>]. <https://doi.org/10.1137/0802028>.
308. Hastie, T.; Tibshirani, R.; Friedman, J. [The Elements of Statistical Learning](#), 2nd ed.; Springer Series in Statistics, Springer New York Inc.: New York, NY, USA, 2009. <https://doi.org/10.1007/978-0-387-84858-7>.
309. Ben-Hur, A.; Weston, J., [A User's Guide to Support Vector Machines](#). In *Data Mining Techniques for the Life Sciences*; Humana Press: Totowa, NJ, 2010; pp. 223–239. [https://doi.org/10.1007/978-1-60327-241-4\\_13](https://doi.org/10.1007/978-1-60327-241-4_13).
310. Suykens, J.; Vandewalle, J. [Least Squares Support Vector Machine Classifiers](#). *Neural Processing Letters* **1999**, *9*, 293–300. <https://doi.org/10.1023/a:1018628609742>.
311. Liu, Y.; Arunachalam, S.; Temme, K. [A rigorous and robust quantum speed-up in supervised machine learning](#). *Nature Physics* **2021**, *17*, 1013–1017. <https://doi.org/10.1038/s41567-021-01287-z>.
312. Glick, J.R.; Gujarati, T.P.; Córcoles, A.D.; Kim, Y.; Kandala, A.; Gambetta, J.M.; Temme, K. [Covariant quantum kernels for data with group structure](#). *Nature Physics* **2024**. <https://doi.org/10.1038/s41567-023-02340-9>.
313. Shin, S.; Sweke, R.; Jeong, H. [New perspectives on quantum kernels through the lens of entangled tensor kernels](#), 2025, [[arXiv:quant-ph/2503.20683](https://arxiv.org/abs/2503.20683)].
314. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. [Supervised learning with quantum-enhanced feature spaces](#). *Nature* **2019**, *567*, 209–212. <https://doi.org/10.1038/s41586-019-0980-2>.
315. Neven, H.; Denchev, V.S.; Rose, G.; Macready, W.G. [Training a Large Scale Classifier with the Quantum Adiabatic Algorithm](#). *arXiv* **2009**, [[arXiv:quant-ph/0912.0779](https://arxiv.org/abs/0912.0779)]. <https://doi.org/10.48550/ARXIV.0912.0779>.
316. Pudenz, K.L.; Lidar, D.A. [Quantum adiabatic machine learning](#). *Quantum Information Processing* **2012**, *12*, 2027–2070. <https://doi.org/10.1007/s11128-012-0506-4>.
317. Shi, Y. [Entropy lower bounds for quantum decision tree complexity](#). *Information Processing Letters* **2002**, *81*, 23–27. [https://doi.org/10.1016/s0020-0190\(01\)00191-0](https://doi.org/10.1016/s0020-0190(01)00191-0).
318. Lu, S.; Braunstein, S.L. [Quantum decision tree classifier](#). *Quantum Information Processing* **2013**, *13*, 757–770. <https://doi.org/10.1007/s11128-013-0687-5>.
319. Heese, R.; Bickel, P.; Niederle, A.E. [Representation of binary classification trees with binary features by quantum circuits](#). *Quantum* **2022**, *6*, 676. <https://doi.org/10.22331/q-2022-03-30-676>.
320. Nishimura, H., [A Survey: SWAP Test and Its Applications to Quantum Complexity Theory](#). In *Algorithmic Foundations for Social Advancement*; Springer Nature Singapore, 2025; pp. 243–261. [https://doi.org/10.1007/978-981-96-0668-9\\_16](https://doi.org/10.1007/978-981-96-0668-9_16).
321. Gentinetta, G.; Thomsen, A.; Sutter, D.; Woerner, S. [The complexity of quantum support vector machines](#). *Quantum* **2024**, *8*, 1225. <https://doi.org/10.22331/q-2024-01-11-1225>.
322. Wang, X.; Du, Y.; Luo, Y.; Tao, D. [Towards understanding the power of quantum kernels in the NISQ era](#). *Quantum* **2021**, *5*, 531. <https://doi.org/10.22331/q-2021-08-30-531>.
323. Zalivako, I.V.; Gircha, A.I.; Kiktenko, E.O.; Nikolaeva, A.S.; Drozhzhin, D.A.; Borisenko, A.S.; Korolkov, A.E.; Semenina, N.V.; Galstyan, K.P.; Kamenskikh, P.A.; et al. [Supervised binary classification of small-scale digit images and weighted graphs with a trapped-ion quantum processor](#). *Phys. Rev. A* **2025**, *111*, 052436. <https://doi.org/10.1103/PhysRevA.111.052436>.
324. Suzuki, T.; Hasebe, T.; Miyazaki, T. [Quantum support vector machines for classification and regression on a trapped-ion quantum computer](#), 2024, [[arXiv:quant-ph/2307.02091](https://arxiv.org/abs/2307.02091)].
325. He, H.; Xiao, Y. [Probabilistic Quantum SVM Training on Ising Machine](#), 2025, [[arXiv:cs.LG/2503.16363](https://arxiv.org/abs/2503.16363)].
326. Basheer, A.; Afham, A.; Goyal, S.K. [Quantum k-nearest neighbors algorithm](#), 2021, [[arXiv:quant-ph/2003.09187](https://arxiv.org/abs/2003.09187)].
327. Wiebe, N.; Kapoor, A.; Svore, K. [Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning](#), 2014, [[arXiv:quant-ph/1401.2142](https://arxiv.org/abs/1401.2142)].
328. Peterson, L.E. [K-nearest neighbor](#). *Scholarpedia* **2009**, *4*, 1883. revision #137311, <https://doi.org/10.4249/scholarpedia.1883>.
329. IBM. [What is the k-nearest neighbors \(KNN\) algorithm?](#) Last accessed: 24.03.2025.

330. Jiang, L.; Cai, Z.; Wang, D.; Jiang, S. *Survey of Improving K-Nearest-Neighbor for Classification*. In Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007, Vol. 1, pp. 679–683. <https://doi.org/10.1109/FSKD.2007.552>.
331. Cunningham, P.; Delany, S.J. *k-Nearest Neighbour Classifiers - A Tutorial*. *ACM Comput. Surv.* **2021**, *54*. <https://doi.org/10.1145/3459665>.
332. Taunk, K.; De, S.; Verma, S.; Swetapadma, A. *A Brief Review of Nearest Neighbor Algorithm for Learning and Classification*. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1255–1260. <https://doi.org/10.1109/ICCS45141.2019.9065747>.
333. Cover, T.; Hart, P. *Nearest neighbor pattern classification*. *IEEE Transactions on Information Theory* **1967**, *13*, 21–27. <https://doi.org/10.1109/TIT.1967.1053964>.
334. Murphy, K. *Machine Learning: A Probabilistic Perspective*; Adaptive Computation and Machine Learning series, MIT Press, 2012.
335. Basheer, A.; Afham, A.; Goyal, S.K. *Quantum k-nearest neighbors algorithm*, 2021, [arXiv:quant-ph/2003.09187].
336. Zardini, E.; Blanzieri, E.; Pastorello, D. *A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation*. *Quantum Machine Intelligence* **2024**, *6*. <https://doi.org/10.1007/s42484-024-00155-2>.
337. Giovannetti, V.; Lloyd, S.; Maccone, L. *Quantum Random Access Memory*. *Phys. Rev. Lett.* **2008**, *100*, 160501. <https://doi.org/10.1103/PhysRevLett.100.160501>.
338. Phalak, K.; Chatterjee, A.; Ghosh, S. *Quantum Random Access Memory for Dummies*. *Sensors* **2023**, *23*, 7462, [arXiv:quant-ph/2305.01178]. <https://doi.org/10.3390/s23177462>.
339. Schuld, M.; Petruccione, F. *Supervised Learning with Quantum Computers*, 1st ed.; Springer Publishing Company, Incorporated, 2018.
340. IBM. *Data encoding*. Last accessed: 24.03.2025.
341. Jaques, S.; Rattew, A.G. *QRAM: A Survey and Critique*. *arXiv* **2023**, [arXiv:quant-ph/2305.10310]. <https://doi.org/10.48550/ARXIV.2305.10310>.
342. Giovannetti, V.; Lloyd, S.; Maccone, L. *Architectures for a quantum random access memory*. *Phys. Rev. A* **2008**, *78*, 052310. <https://doi.org/10.1103/PhysRevA.78.052310>.
343. Buhrman, H.; Cleve, R.; Watrous, J.; de Wolf, R. *Quantum fingerprinting*. *Phys. Rev. Lett.* **2001**, *87*, 167902. <https://doi.org/10.1103/PhysRevLett.87.167902>.
344. Guerrero-Estrada, A.Y.; Quezada, L.F.; Sun, G.H. *Benchmarking quantum versions of the kNN algorithm with a metric based on amplitude-encoded features*. *Scientific Reports* **2024**, *14*, 16697. <https://doi.org/10.1038/s41598-024-67392-0>.
345. Baldazzi, A.; Leone, N.; Sanna, M.; Azzini, S.; Pavesi, L. *A linear photonic swap test circuit for quantum kernel estimation*. *Quantum Science and Technology* **2024**, *9*, 045053. <https://doi.org/10.1088/2058-9565/ad7be7>.
346. Herman, D.; Googin, C.; Liu, X.; Galda, A.; Safro, I.; Sun, Y.; Pistoia, M.; Alexeev, Y. *A Survey of Quantum Computing for Finance*, 2022. <https://doi.org/10.48550/ARXIV.2201.02773>.
347. Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. *Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications*. *Journal of Big Data* **2024**, *11*, 113. <https://doi.org/10.1186/s40537-024-00973-y>.
348. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. *An introduction to decision tree modeling*. *Journal of Chemometrics* **2004**, *18*, 275–285, <https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.873>. <https://doi.org/10.1002/cem.873>.
349. Yang, F.J. *An Extended Idea about Decision Trees*. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 349–354. <https://doi.org/10.1109/CSCI49370.2019.00068>.
350. Younes, L. *Introduction to Machine Learning*, 2024, [arXiv:stat.ML/2409.02668].
351. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Chapman and Hall/CRC, 1984. <https://doi.org/10.1201/9781315139470>.
352. Shannon, C.E. *A mathematical theory of communication*. *The Bell System Technical Journal* **1948**, *27*, 623–656. <https://doi.org/10.1002/j.1538-7305.1948.tb00917.x>.
353. Nowozin, S. *Improved Information Gain Estimates for Decision Tree Induction*, 2012, [arXiv:cs.LG/1206.4620].

354. Murthy, S.; Salzberg, S. [Decision Tree Induction: How Effective is the Greedy Heuristic?](#) In Proceedings of the KDD 1995 - Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining; Fayyad, U.; Uthurusamy, R., Eds. AAAI Press, 1995, pp. 222–227.
355. van der Linden, J.G.M.; Vos, D.; de Weerd, M.M.; Verwer, S.; Demirović, E. [Optimal or Greedy Decision Trees? Revisiting their Objectives, Tuning, and Performance](#), 2025, [arXiv:cs.LG/2409.12788].
356. Khadiev, K.; Mannapov, I.; Safina, L. [The Quantum Version Of Classification Decision Tree Constructing Algorithm C5.0](#), 2019, [arXiv:cs.LG/1907.06840].
357. Ambainis, A. [Understanding Quantum Algorithms via Query Complexity](#), 2017, [arXiv:quant-ph/1712.06349].
358. Sharma, D.; Singh, P.; Kumar, A. [Quantum-inspired attribute selection algorithms](#). *Quantum Sci. Technol.* **2025**, *10*, 015036, [arXiv:quant-ph/2310.18243]. <https://doi.org/10.1088/2058-9565/ad934d>.
359. Koren, M.; Peretz, O. [A quantum procedure for estimating information gain in Boolean classification task](#). *Quantum Machine Intelligence* **2024**, *6*, 16. <https://doi.org/10.1007/s42484-024-00151-6>.
360. Ahuja, A.; Kapoor, S. [A Quantum Algorithm for finding the Maximum](#), 1999, [arXiv:quant-ph/quant-ph/9911082].
361. Kumar, N.; Yalovetzky, R.; Li, C.; Minssen, P.; Pistoia, M. [Des-q: a quantum algorithm to provably speedup retraining of decision trees](#). *Quantum* **2025**, *9*, 1588. <https://doi.org/10.22331/q-2025-01-13-1588>.
362. Wang, H.; Gupta, G. [Superfast Selection for Decision Tree Algorithms](#), 2024, [arXiv:cs.LG/2405.20622].
363. Zhang, G.; Gionis, A. [Regularized impurity reduction: accurate decision trees with complexity guarantees](#). *Data Mining and Knowledge Discovery* **2023**, *37*, 434–475. <https://doi.org/10.1007/s10618-022-00884-7>.
364. Cornelissen, A.; Mande, N.S.; Patro, S. [Improved Quantum Query Upper Bounds Based on Classical Decision Trees](#). *Quantum* **2025**, *9*, 1777. <https://doi.org/10.22331/q-2025-06-23-1777>.
365. Pandya, R.; Pandya, J. [C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning](#). *International Journal of Computer Applications* **2015**, *117*.
366. Kerenidis, I.; Landman, J.; Luongo, A.; Prakash, A. [q-means: A quantum algorithm for unsupervised machine learning](#). In Proceedings of the Advances in Neural Information Processing Systems; Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; Garnett, R., Eds. Curran Associates, Inc., 2019, Vol. 32.
367. Li, Z.; Terashi, K. [Quantum decision trees with information entropy](#), 2025, [arXiv:quant-ph/2502.11412].
368. Rad, D.; Cuc, L.D.; Croitoru, G.; Gomo, B.C.; Mazuru, L.; Bilti, R.S.; Rusu, S.; Sinaci, M.; Barbu, F.S. [Modeling Investment Decisions Through Decision Tree Regression—A Behavioral Finance Theory Approach](#). *Electronics* **2025**, *14*. <https://doi.org/10.3390/electronics14081505>.
369. Grudniewicz, J.; Ślepaczuk, R. [Application of machine learning in algorithmic investment strategies on global stock markets](#). *Research in International Business and Finance* **2023**, *66*, 102052. <https://doi.org/10.1016/j.ribaf.2023.102052>.
370. Tu, J.; Wu, Z. [Inherently interpretable machine learning for credit scoring: Optimal classification tree with hyperplane splits](#). *European Journal of Operational Research* **2025**, *322*, 647–664. <https://doi.org/10.1016/j.ejor.2024.10.046>.
371. Luo, G.; Arshad, M.A.; Luo, G. [Decision Trees for Strategic Choice of Augmenting Management Intuition with Machine Learning](#). *Symmetry* **2025**, *17*. <https://doi.org/10.3390/sym17070976>.
372. Černevičienė, J.; Kabašinskas, A. [Explainable artificial intelligence \(XAI\) in finance: a systematic literature review](#). *Artificial Intelligence Review* **2024**, *57*, 216. <https://doi.org/10.1007/s10462-024-10854-8>.
373. Bücker, M.; Szepannek, G.; Gosiewska, A.; Biecek, P. [Transparency, auditability, and explainability of machine learning models in credit scoring](#). *Journal of the Operational Research Society* **2022**, *73*, 70–90, [https://doi.org/10.1080/01605682.2021.1922098]. <https://doi.org/10.1080/01605682.2021.1922098>.
374. Forensic Risk Alliance. [Explainable AI: Building Confidence and Compliance](#), 2024. Accessed: 11-08-2025.
375. Montanaro, A. [Almost all decision trees do not allow significant quantum speed-up](#), 2012, [arXiv:quant-ph/1209.4781].
376. Shi, Y. [Entropy lower bounds of quantum decision tree complexity](#), 2000, [arXiv:quant-ph/quant-ph/0008095].
377. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An introduction to statistical learning*; Springer, 2021.
378. Altman, N.S. [An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression](#). *The American Statistician* **1992**, *46*, 175–185, [https://www.tandfonline.com/doi/pdf/10.1080/00031305.1992.10475879]. <https://doi.org/10.1080/00031305.1992.10475879>.
379. Seber, G.; Lee, A. *Linear regression analysis. 2nd edit.*; Wiley, 2003.

380. Strang, G. *Introduction to Linear Algebra*, sixth ed.; Wellesley-Cambridge Press: Wellesley, MA, 2023.
381. Aster, R.C.; Borchers, B.; Thurber, C.H. *Parameter Estimation and Inverse Problems (Third Edition)*, third edition ed.; Elsevier, 2019.
382. Hoerl, A.E.; Kennard, R.W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **1970**, *12*, 55–67, [<https://www.tandfonline.com/doi/pdf/10.1080/00401706.1970.10488634>]. <https://doi.org/10.1080/00401706.1970.10488634>.
383. Liu, S.; Dobriban, E. *Ridge Regression: Structure, Cross-Validation, and Sketching*, 2020, [[arXiv:math.ST/1910.02373](https://arxiv.org/abs/1910.02373)].
384. Golub, G.H.; Van Loan, C.F. *Matrix Computations - 4th Edition*; Johns Hopkins University Press: Philadelphia, PA, 2013; [<https://epubs.siam.org/doi/pdf/10.1137/1.9781421407944>]. <https://doi.org/10.1137/1.9781421407944>.
385. Song, Z.; Yin, J.; Zhang, R. Revisiting Quantum Algorithms for Linear Regressions: Quadratic Speedups without Data-Dependent Parameters, 2023, [[arXiv:quant-ph/2311.14823](https://arxiv.org/abs/2311.14823)].
386. Chakraborty, S.; Morolia, A.; Peduri, A. Quantum Regularized Least Squares. *Quantum* **2023**, *7*, 988. <https://doi.org/10.22331/q-2023-04-27-988>.
387. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. <https://doi.org/10.1103/PhysRevLett.122.040504>.
388. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press, 2004.
389. Welling, M. *Kernel Ridge Regression*. Lecture notes, University of Toronto, 2007.
390. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, 2006.
391. Schnabel, J.; Roth, M. Quantum kernel methods under scrutiny: a benchmarking study. *Quantum Machine Intelligence* **2025**, *7*. <https://doi.org/10.1007/s42484-025-00273-5>.
392. Schuld, M. Supervised quantum machine learning models are kernel methods, 2021, [[arXiv:quant-ph/2101.11020](https://arxiv.org/abs/2101.11020)].
393. Li, T.; Chakrabarti, S.; Wu, X. Sublinear quantum algorithms for training linear and kernel-based classifiers, 2019, [[arXiv:quant-ph/1904.02276](https://arxiv.org/abs/1904.02276)].
394. Huang, H.Y.; Broughton, M.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; McClean, J.R. Power of data in quantum machine learning. *Nature Communications* **2021**, *12*. <https://doi.org/10.1038/s41467-021-22539-9>.
395. Sabarad, V.; Varma, V.; Mahesh, T.S. Experimental Machine Learning with Classical and Quantum Data via NMR Quantum Kernels, 2025, [[arXiv:quant-ph/2412.09557](https://arxiv.org/abs/2412.09557)].
396. Miyabe, S.; Quanz, B.; Shimada, N.; Mitra, A.; Yamamoto, T.; Rastunkov, V.; Alevras, D.; Metcalf, M.; King, D.J.M.; Mamouei, M.; et al. Quantum Multiple Kernel Learning in Financial Classification Tasks, 2023, [[arXiv:quant-ph/2312.00260](https://arxiv.org/abs/2312.00260)].
397. Zhou, X.; Yu, J.; Tan, J.; Jiang, T. Quantum kernel estimation-based quantum support vector regression. *Quantum Information Processing* **2024**, *23*, 29. <https://doi.org/10.1007/s11128-023-04231-7>.
398. Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information* **2019**, *5*. <https://doi.org/10.1038/s41534-019-0157-8>.
399. Thanasilp, S.; Wang, S.; Cerezo, M.; Holmes, Z. Exponential concentration in quantum kernel methods. *Nature Communications* **2024**, *15*, 5200. <https://doi.org/10.1038/s41467-024-49287-w>.
400. Shiota, T.; Ishihara, K.; Mizukami, W. Universal neural network potentials as descriptors: Towards scalable chemical property prediction using quantum and classical computers. *Digital Discovery* **2024**, *3*, 1714–1728, [[arXiv:quant-ph/2402.18433](https://arxiv.org/abs/2402.18433)]. <https://doi.org/10.1039/D4DD00098F>.
401. Gurney, K. *An Introduction to Neural Networks*; An Introduction to Neural Networks, Taylor & Francis, 1997.
402. Schuld, M.; Sinayskiy, I.; Petruccione, F. The quest for a Quantum Neural Network. *Quantum Information Processing* **2014**, *13*, 2567–2586. <https://doi.org/10.1007/s11128-014-0809-8>.
403. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.
404. Abbas, A.; Sutter, D.; Zoufal, C.; Lucchi, A.; Figalli, A.; Woerner, S. The power of quantum neural networks. *Nature Computational Science* **2021**, *1*, 403–409. <https://doi.org/10.1038/s43588-021-00084-1>.
405. Altaisky, M.V. Quantum neural network, 2001, [[arXiv:quant-ph/0107012](https://arxiv.org/abs/quant-ph/0107012)].
406. Schuld, M.; Sweke, R.; Meyer, J.J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **2021**, *103*, 032430. <https://doi.org/10.1103/PhysRevA.103.032430>.
407. Ding, X.; Song, Z.; Xu, J.; Hou, Y.; Yang, T.; Shan, Z. Scalable parameterized quantum circuits classifier. *Scientific Reports* **2024**, *14*, 15886. <https://doi.org/10.1038/s41598-024-66394-2>.

408. Schuld, M.; Petruccione, F. *Machine Learning with Quantum Computers*; Quantum Science and Technology, Springer International Publishing, 2021.
409. Beer, K.; Bondarenko, D.; Farrelly, T.; Osborne, T.J.; Salzmann, R.; Scheiermann, D.; Wolf, R. [Training deep quantum neural networks](https://doi.org/10.1038/s41467-020-14454-2). *Nature Communications* **2020**, *11*, 808. <https://doi.org/10.1038/s41467-020-14454-2>.
410. Crooks, G.E. [Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition](https://arxiv.org/abs/1905.13311), 2019, [arXiv:quant-ph/1905.13311].
411. Paquet, E.; Soleymani, F. [QuantumLeap: Hybrid quantum neural network for financial predictions](https://doi.org/10.1016/j.eswa.2022.116583). *Expert Syst. Appl.* **2022**, *195*. <https://doi.org/10.1016/j.eswa.2022.116583>.
412. Choudhary, P.K.; Innan, N.; Shafique, M.; Singh, R. [HQNN-FSP: A Hybrid Classical-Quantum Neural Network for Regression-Based Financial Stock Market Prediction](https://arxiv.org/abs/2503.15403). *arXiv preprint arXiv:2503.15403* **2025**.
413. Tudisco, A.; Volpe, D.; Ranieri, G.; Curato, G.; Ricossa, D.; Graziano, M.; Corbelleto, D. [Evaluating the Computational Advantages of the Variational Quantum Circuit Model in Financial Fraud Detection](https://doi.org/10.1109/ACCESS.2024.3432312). *IEEE Access* **2024**, *12*, 102918–102940. <https://doi.org/10.1109/ACCESS.2024.3432312>.
414. Innan, N.; Marchisio, A.; Bennai, M.; Shafique, M. [QFNN-FFD: Quantum Federated Neural Network for Financial Fraud Detection](https://arxiv.org/abs/2404.02595), 2024, [arXiv:quant-ph/2404.02595].
415. Bowles, J.; Ahmed, S.; Schuld, M. [Better than classical? The subtle art of benchmarking quantum machine learning models](https://arxiv.org/abs/2403.07059), 2024, [arXiv:quant-ph/2403.07059].
416. Gonon, L.; Jacquier, A. [Universal Approximation Theorem and Error Bounds for Quantum Neural Networks and Quantum Reservoirs](https://doi.org/10.1109/tnnls.2025.3552223). *IEEE Transactions on Neural Networks and Learning Systems* **2025**, p. 1–14. <https://doi.org/10.1109/tnnls.2025.3552223>.
417. Kashif, M.; Marchisio, A.; Shafique, M. [Computational Advantage in Hybrid Quantum Neural Networks: Myth or Reality?](https://arxiv.org/abs/2412.04991), 2025, [arXiv:quant-ph/2412.04991].
418. Kashif, M.; Al-Kuwari, S. [ResQnets: a residual approach for mitigating barren plateaus in quantum neural networks](https://doi.org/10.1140/epjqt/s40507-023-00216-8). *EPJ Quantum Technology* **2024**, *11*. <https://doi.org/10.1140/epjqt/s40507-023-00216-8>.
419. Krenn, M.; Landgraf, J.; Foesel, T.; Marquardt, F. [Artificial intelligence and machine learning for quantum technologies](https://doi.org/10.1103/PhysRevA.107.010101). *Phys. Rev. A* **2023**, *107*, 010101. <https://doi.org/10.1103/PhysRevA.107.010101>.
420. Landman, J. [Quantum Algorithms for Unsupervised Machine Learning and Neural Networks](https://doi.org/10.48550/ARXIV.2111.03598), 2021. <https://doi.org/10.48550/ARXIV.2111.03598>.
421. Aïmeur, E.; Brassard, G.; Gambs, S. [Quantum speed-up for unsupervised learning](https://doi.org/10.1007/s10994-012-5316-5). *Mach. Learn.* **2013**, *90*, 261–287. <https://doi.org/10.1007/s10994-012-5316-5>.
422. Wang, Y.J.; Yang, X.; Ju, C.; Zhang, Y.; Zhang, J.; Xu, Q.; Wang, Y.; Gao, X.; Cao, X.; Ma, Y.; et al. [Quantum Computing in Community Detection for Anti-Fraud Applications](https://doi.org/10.3390/e26121026). *Entropy* **2024**, *26*. <https://doi.org/10.3390/e26121026>.
423. Lloyd, S. [Least squares quantization in PCM](https://doi.org/10.1109/TIT.1982.1056489). *IEEE Transactions on Information Theory* **1982**, *28*, 129–137. <https://doi.org/10.1109/TIT.1982.1056489>.
424. Arthur, D.; Vassilvitskii, S. [k-means++: the advantages of careful seeding](https://doi.org/10.1145/114466). In Proceedings of the Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, USA, 2007; SODA '07, p. 1027–1035.
425. Strohmer, T. [K-Means Clustering](https://www.eecs.berkeley.edu/~strohmer/teaching/EECS-293A/lectures/lecture10/k-means-clustering.pdf). Lecture slides, University of California, Davis, 2017.
426. Poggiali, A.; Berti, A.; Bernasconi, A.; Del Corso, G.M.; Guidotti, R. [Quantum clustering with k-Means: A hybrid approach](https://doi.org/10.1016/j.tcs.2024.114466). *Theor. Comput. Sci.* **2024**, *992*, 114466, [arXiv:quant-ph/2212.06691]. <https://doi.org/10.1016/j.tcs.2024.114466>.
427. Zubair, M.; Iqbal, A.; Shil, A.; Chowdhury, M.; Moni, M.A.; Sarker, I. [An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling](https://doi.org/10.1007/s40745-022-00428-2). *Annals of Data Science* **2022**, *11*, 1525–1544. <https://doi.org/10.1007/s40745-022-00428-2>.
428. Woodruff, D.P.; Zhong, P.; Zhou, S. [Near-Optimal k-Clustering in the Sliding Window Model](https://arxiv.org/abs/2311.00642), 2023, [arXiv:cs.DS/2311.00642].
429. Drineas, P.; Frieze, A.M.; Kannan, R.; Vempala, S.S.; Vinay, V. [Clustering Large Graphs via the Singular Value Decomposition](https://doi.org/10.1162/089976600560000). *Machine Learning* **2004**, *56*, 9–33.
430. Shi, X.; Shang, Y.; Guo, C. [Quantum inspired K-means algorithm using matrix product states](https://arxiv.org/abs/2006.06164), 2020, [arXiv:physics.comp-ph/2006.06164].
431. Park, D.; Petruccione, F.; Rhee, J. [Circuit-Based Quantum Random Access Memory for Classical Data](https://doi.org/10.1038/s41598-019-40439-3). *Scientific reports* **2019**, *9*. Publisher Copyright: © 2019, The Author(s)., <https://doi.org/10.1038/s41598-019-40439-3>.

432. Abreu, D.; Moura, D.; Esteve Rothenberg, C.; Abelém, A. [QuantumNetSec: Quantum Machine Learning for Network Security](#). *International Journal of Network Management* **2025**, *35*, e70018, [<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.70018>]. e70018 nem.70018, <https://doi.org/10.1002/nem.70018>.
433. Li, Y. [Quantum Machine Learning Based on K-Means to Optimize Cross-Border Intelligent Payment Supervision](#). In Proceedings of the Proceedings of the 2024 4th International Conference on Big Data, Artificial Intelligence and Risk Management, New York, NY, USA, 2025; ICBAR '24, p. 676–681. <https://doi.org/10.1145/3718751.3718859>.
434. Tsai, Y.C.; Chen, S.Y.C. [Quantum Feature Optimization for Enhanced Clustering of Blockchain Transaction Data](#), 2025, [[arXiv:cs.LG/2505.16672](https://arxiv.org/abs/2505.16672)].
435. Ng, A.; Jordan, M.; Weiss, Y. [On Spectral Clustering: Analysis and an algorithm](#). In Proceedings of the Advances in Neural Information Processing Systems; Dietterich, T.; Becker, S.; Ghahramani, Z., Eds. MIT Press, 2001, Vol. 14.
436. Shi, J.; Malik, J. [Normalized cuts and image segmentation](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2000**, *22*, 888–905. <https://doi.org/10.1109/34.868688>.
437. Yan, D.; Huang, L.; Jordan, M.I. [Fast approximate spectral clustering](#). In Proceedings of the Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2009; KDD '09, p. 907–916. <https://doi.org/10.1145/1557019.1557118>.
438. Kerenidis, I.; Landman, J. [Quantum spectral clustering](#). *Phys. Rev. A* **2021**, *103*, 042415. <https://doi.org/10.1103/PhysRevA.103.042415>.
439. Volya, D.; Mishra, P. [Quantum Spectral Clustering of Mixed Graphs](#). In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), 2021, pp. 463–468. <https://doi.org/10.1109/DAC18074.2021.9586308>.
440. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. [Generative Adversarial Networks: An Overview](#). *IEEE Signal Processing Magazine* **2018**, *35*, 53–65. <https://doi.org/10.1109/msp.2017.2765202>.
441. Coyle, B.; Mills, D.; Danos, V.; Kashefi, E. [The Born supremacy: quantum advantage and training of an Ising Born machine](#). *npj Quantum Information* **2020**, *6*. <https://doi.org/10.1038/s41534-020-00288-9>.
442. Kullback, S.; Leibler, R.A. [On Information and Sufficiency](#). *The Annals of Mathematical Statistics* **1951**, *22*, 79–86. <https://doi.org/10.1214/aoms/1177729694>.
443. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. [Parameterized quantum circuits as machine learning models](#). *Quantum Science and Technology* **2019**, *4*, 043001. <https://doi.org/10.1088/2058-9565/ab4eb5>.
444. Zoufal, C.; Lucchi, A.; Woerner, S. [Variational quantum Boltzmann machines](#). *Quantum Machine Intelligence* **2021**, *3*. <https://doi.org/10.1007/s42484-020-00033-7>.
445. Lloyd, S.; Weedbrook, C. [Quantum Generative Adversarial Learning](#). *Phys. Rev. Lett.* **2018**, *121*, 040502. <https://doi.org/10.1103/PhysRevLett.121.040502>.
446. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. [A learning algorithm for boltzmann machines](#). *Cognitive Science* **1985**, *9*, 147–169. [https://doi.org/10.1016/S0364-0213\(85\)80012-4](https://doi.org/10.1016/S0364-0213(85)80012-4).
447. Aggarwal, C.C. *Neural Networks and Deep Learning*, second ed.; Springer Cham, 2023; p. 497.
448. Hopfield, J.J. [Neural Networks and Physical Systems with Emergent Collective Computational Abilities](#). *Proceedings of the National Academy of Sciences of the United States of America* **1982**, *79*, 2554–2558.
449. Rumelhart, D.E.; McClelland, J.L., [Learning and Relearning in Boltzmann Machines](#). In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*; MIT Press, 1987; Vol. 1, pp. 282–317.
450. Haykin, S. *Neural Networks and Learning Machines*; Number v. 10 in Neural networks and learning machines, Pearson, 2009.
451. Aarts, E.; Korst, J. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*; John Wiley & Sons, Inc., 1989.
452. Carreira-Perpiñán, M.A.; Hinton, G. [On Contrastive Divergence Learning](#). In Proceedings of the Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics; Cowell, R.G.; Ghahramani, Z., Eds. PMLR, 06–08 Jan 2005, Vol. R5, *Proceedings of Machine Learning Research*, pp. 33–40. Reissued by PMLR on 30 March 2021.
453. Rossi, R.J. *Mathematical Statistics: An Introduction to Likelihood Based Inference*; John Wiley & Sons, Ltd, 2018.
454. Amin, M.H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. [Quantum Boltzmann Machine](#). *Phys. Rev. X* **2018**, *8*, 021050. <https://doi.org/10.1103/PhysRevX.8.021050>.
455. Hebb, D. *The Organization of Behavior: A Neuropsychological Theory*, first ed.; Psychology Press, 2002.

456. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum deep learning. *Quantum Info. Comput.* **2016**, *16*, 541–587.
457. Patel, D.; Koch, D.; Patel, S.; Wilde, M.M. Quantum Boltzmann machine learning of ground-state energies, 2024, [arXiv:quant-ph/2410.12935].
458. Berner, N.; Fortuin, V.; Landman, J. Quantum Bayesian Neural Networks, 2021, [arXiv:quant-ph/2107.09599].
459. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN, 2017, [arXiv:stat.ML/1701.07875].
460. Dallaire-Demers, P.L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324. <https://doi.org/10.1103/PhysRevA.98.012324>.
461. Situ, H.; He, Z.; Wang, Y.; Li, L.; Zheng, S. Quantum generative adversarial network for generating discrete distribution. *Information Sciences* **2020**, *538*, 193–208. <https://doi.org/10.1016/j.ins.2020.05.127>.
462. Schuld, M.; Bergholm, V.; Gogolin, C.; Izaac, J.; Killoran, N. Evaluating analytic gradients on quantum hardware. *Physical Review A* **2019**, *99*, 032331. <https://doi.org/10.1103/PhysRevA.99.032331>.
463. Wierichs, D.; Izaac, J.; Wang, C.; Lin, C.Y.Y. General parameter-shift rules for quantum gradients. *Quantum* **2022**, *6*, 677. <https://doi.org/10.22331/q-2022-03-30-677>.
464. Anoshin, M.; Saginalieva, A.; Mansell, C.; Zhiganov, D.; Shete, V.; Pflitsch, M.; Melnikov, A. Hybrid Quantum Cycle Generative Adversarial Network for Small Molecule Generation. *IEEE Transactions on Quantum Engineering* **2024**, *5*, 1–14. <https://doi.org/10.1109/TQE.2024.3414264>.
465. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications* **2021**, *12*. <https://doi.org/10.1038/s41467-021-21728-w>.
466. Wang, S.; Fontana, E.; Cerezo, M.; Sharma, K.; Sone, A.; Cincio, L.; Coles, P.J. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications* **2021**, *12*. <https://doi.org/10.1038/s41467-021-27045-6>.
467. Pajuhanfard, M.; Kiani, R.; Sheng, V.S. Survey of Quantum Generative Adversarial Networks (QGAN) to Generate Images. *Mathematics* **2024**, *12*. <https://doi.org/10.3390/math12233852>.
468. Liu, J.G.; Wang, L. Differentiable learning of quantum circuit Born machines. *Physical Review A* **2018**, *98*. <https://doi.org/10.1103/physreva.98.062324>.
469. Low, G.H.; Yoder, T.J.; Chuang, I.L. Quantum inference on Bayesian networks. *Physical Review A* **2014**, *89*. <https://doi.org/10.1103/physreva.89.062315>.
470. Borujeni, S.E.; Nguyen, N.H.; Nannapaneni, S.; Behrman, E.C.; Steck, J.E. Experimental evaluation of quantum Bayesian networks on IBM QX hardware. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2020, pp. 372–378. <https://doi.org/10.1109/qce492.97.2020.00053>.
471. Moreira, C.; Wichert, A. Quantum-Like Bayesian Networks for Modeling Decision Making. *Frontiers in Psychology* **2016**, *7*. <https://doi.org/10.3389/fpsyg.2016.00011>.
472. Wichert, A.; Moreira, C.; Bruza, P. Balanced Quantum-Like Bayesian Networks. *Entropy* **2020**, *22*, 170. <https://doi.org/10.3390/e22020170>.
473. F.R.S., K.P. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1* **1901**, *2*, 559–572.
474. Hotelling, H. Analysis of a Complex of Statistical Variables Into Principal Components. *Journal of Educational Psychology*, **1933**, *24*. <https://doi.org/http://dx.doi.org/10.1037/h0071325>.
475. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. *Numerische Mathematik* **1970**, *14*, 403–420. <https://doi.org/10.1007/BF02163027>.
476. Tenenbaum, J.B.; de Silva, V.; Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **2000**, *290*, 2319–2323, [<https://www.science.org/doi/pdf/10.1126/science.290.5500.2319>]. <https://doi.org/10.1126/science.290.5500.2319>.
477. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nature Physics* **2014**, *10*, 631–633. <https://doi.org/10.1038/nphys3029>.
478. Tang, E. Dequantizing algorithms to understand quantum advantage in machine learning. *Nature Reviews Physics* **2022**, *4*, 692–693. <https://doi.org/10.1038/s42254-022-00511-w>.
479. Li, Z.; Chai, Z.; Guo, Y.; Ji, W.; Wang, M.; Shi, F.; Wang, Y.; Lloyd, S.; Du, J. Resonant quantum principal component analysis. *Science Advances* **2021**, *7*. <https://doi.org/10.1126/sciadv.abg2589>.
480. Wang, Y.; Luo, Y. Resource-efficient quantum principal component analysis. *Quantum Science and Technology* **2024**, *9*, 035031. <https://doi.org/10.1088/2058-9565/ad466c>.
481. Nghiem, N.A. New Quantum Algorithm for Principal Component Analysis, 2025, [arXiv:quant-ph/2501.07891].

482. Hastings, M.B. Classical and Quantum Algorithms for Tensor Principal Component Analysis. *Quantum* **2020**, *4*, 237. <https://doi.org/10.22331/q-2020-02-27-237>.
483. Hoyle, D.C.; Rattray, M. PCA learning for sparse high-dimensional data. *Europhysics Letters* **2003**, *62*, 117. <https://doi.org/10.1209/epl/i2003-00370-1>.
484. Wu, J.; Fu, H.; Zhu, M.; Zhang, H.; Xie, W.; Li, X.Y. Quantum circuit autoencoder. *Physical Review A* **2024**, *109*. <https://doi.org/10.1103/physreva.109.032623>.
485. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press, Cambridge, MA, 2018.
486. Ng, A. *CS229 Lecture Notes: Machine Learning*, 2023. Stanford University, Accessed: 2025-07-05.
487. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. *Reinforcement Learning: A Survey*, 1996, [arXiv:cs.AI/cs/9605103].
488. Ladosz, P.; Weng, L.; Kim, M.; Oh, H. Exploration in deep reinforcement learning: A survey. *Information Fusion* **2022**, *85*, 1–22. <https://doi.org/10.1016/j.inffus.2022.03.003>.
489. Howard, R.A. *Dynamic Programming and Markov Processes*; MIT Press, 1960.
490. Bertsekas, D.; Tsitsiklis, J. *Neuro-Dynamic Programming*; Vol. 27, Athena Scientific, 1996. [https://doi.org/10.1007/978-0-387-74759-0\\_440](https://doi.org/10.1007/978-0-387-74759-0_440).
491. Bellman, R.; Bellman, R.; Corporation, R. *Dynamic Programming*; Rand Corporation research study, Princeton University Press, 1957.
492. Murray, P.; Wood, B.; Buehler, H.; Wiese, M.; Pakkanen, M. Deep Hedging: Continuous Reinforcement Learning for Hedging of General Portfolios across Multiple Risk Aversions. In Proceedings of the Proceedings of the Third ACM International Conference on AI in Finance, New York, NY, USA, 2022; ICAIF '22, p. 361–368. <https://doi.org/10.1145/3533271.3561731>.
493. Buehler, H.; Gonon, L.; Teichmann, J.; Wood, B.; Mohan, B.; Kochems, J. Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning. *SSRN Electronic Journal* **2019**. <https://doi.org/10.2139/ssrn.3355706>.
494. Cherrat, E.A.; Raj, S.; Kerenidis, I.; Shekhar, A.; Wood, B.; Dee, J.; Chakrabarti, S.; Chen, R.; Herman, D.; Hu, S.; et al. Quantum Deep Hedging. *Quantum* **2023**, *7*, 1191. <https://doi.org/10.22331/q-2023-11-29-1191>.
495. Skolik, A.; Jerbi, S.; Dunjko, V. Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning. *Quantum* **2022**, *6*, 720. <https://doi.org/10.22331/q-2022-05-24-720>.
496. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. <https://doi.org/10.1038/nature14236>.
497. van Hasselt, H.; Doron, Y.; Strub, F.; Hessel, M.; Sonnerat, N.; Modayil, J. *Deep Reinforcement Learning and the Deadly Triad*, 2018, [arXiv:cs.AI/1812.02648].
498. Jerbi, S.; Gyurik, C.; Marshall, S.C.; Briegel, H.J.; Dunjko, V. Parametrized quantum policies for reinforcement learning, 2021, [arXiv:quant-ph/2103.05577].
499. Dong, D.; Chen, C.; Li, H.; Tarn, T.J. Quantum Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2008**, *38*, 1207–1220. <https://doi.org/10.1109/tsmcb.2008.925743>.
500. Paparo, G.D.; Dunjko, V.; Makmal, A.; Martin-Delgado, M.A.; Briegel, H.J. Quantum Speedup for Active Learning Agents. *Physical Review X* **2014**, *4*. <https://doi.org/10.1103/physrevx.4.031002>.
501. Meyer, N.; Ufrecht, C.; Yammine, G.; Kontes, G.; Mutschler, C.; Scherer, D.D. Benchmarking Quantum Reinforcement Learning, 2025, [arXiv:quant-ph/2501.15893].
502. Chen, S.Y.C. Quantum Deep Q-Learning with Distributed Prioritized Experience Replay. In Proceedings of the 2023 International Conference on Quantum Computing and Engineering, 4 2023, [arXiv:quant-ph/2304.09648]. <https://doi.org/10.1109/QCE57702.2023.10180>.
503. Lockwood, O.; Si, M. Reinforcement Learning with Quantum Variational Circuit. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* **2020**, *16*, 245–251. <https://doi.org/10.1609/aiide.v16i1.7437>.
504. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. <https://doi.org/10.22331/q-2018-08-06-79>.
505. Cerezo, M.; Verdon, G.; Huang, H.Y.; Cincio, L.; Coles, P.J. Challenges and opportunities in quantum machine learning. *Nature Computational Science* **2022**, *2*, 567–576. <https://doi.org/10.1038/s43588-022-00311-3>.
506. Hohenfeld, H.; Heimann, D.; Wiebe, F.; Kirchner, F. Quantum Deep Reinforcement Learning for Robot Navigation Tasks. *IEEE Access* **2024**, *12*, 87217–87236. <https://doi.org/10.1109/ACCESS.2024.3417808>.

507. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. [Policy gradient methods for reinforcement learning with function approximation](#). In Proceedings of the Proceedings of the 13th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 1999; NIPS'99, p. 1057–1063.
508. Williams, R.J. [Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning](#). *Mach. Learn.* **1992**, *8*, 229–256. <https://doi.org/10.1007/BF00992696>.
509. Dawid, A.; Arnold, J.; Requena, B.; Gresch, A.; Płodzień, M.; Donatella, K.; Nicoli, K.A.; Stornati, P.; Koch, R.; Büttner, M.; et al. *Machine Learning in Quantum Sciences*; Cambridge University Press, 2025.
510. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. [Proximal Policy Optimization Algorithms](#), 2017, [[arXiv:cs.LG/1707.06347](https://arxiv.org/abs/1707.06347)].
511. Arjona-Medina, J.A.; Gillhofer, M.; Widrich, M.; Unterthiner, T.; Brandstetter, J.; Hochreiter, S. [RUDDER: Return Decomposition for Delayed Rewards](#), 2019, [[arXiv:cs.LG/1806.07857](https://arxiv.org/abs/1806.07857)].
512. Pascanu, R.; Bengio, Y. [Revisiting Natural Gradient for Deep Networks](#). *CoRR* **2013**, *abs/1301.3584*.
513. Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J.I. [Data re-uploading for a universal quantum classifier](#). *Quantum* **2020**, *4*, 226. <https://doi.org/10.22331/q-2020-02-06-226>.
514. Chen, S.Y.C. [An Introduction to Quantum Reinforcement Learning \(QRL\)](#), 2024, [[arXiv:quant-ph/2409.05846](https://arxiv.org/abs/2409.05846)].
515. Sequeira, A.; Santos, L.P.; Barbosa, L.S. [Trainability issues in quantum policy gradients](#), 2024, [[arXiv:quant-ph/2406.09614](https://arxiv.org/abs/2406.09614)].
516. Meyer, N.; Ufrecht, C.; Periyasamy, M.; Scherer, D.D.; Plinge, A.; Mutschler, C. [A Survey on Quantum Reinforcement Learning](#), 2024, [[arXiv:quant-ph/2211.03464](https://arxiv.org/abs/2211.03464)].
517. Jerbi, S.; Cornelissen, A.; Ozols, M.; Dunjko, V. [Quantum Policy Gradient Algorithms](#), 2023. <https://doi.org/10.4230/LIPICS.TQC.2023.13>.
518. Dunjko, V.; Liu, Y.K.; Wu, X.; Taylor, J.M. [Exponential improvements for quantum-accessible reinforcement learning](#), 2018, [[arXiv:quant-ph/1710.11160](https://arxiv.org/abs/1710.11160)].
519. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*; Chapman and Hall/CRC, 2007. <https://doi.org/10.1201/9781420010756>.
520. Sahu, S.K.; Mazumdar, K. [State-of-the-art analysis of quantum cryptography: applications and future prospects](#). *Frontiers in Physics* **2024**, *12*. <https://doi.org/10.3389/fphy.2024.1456491>.
521. Thornhill, J. [Time to get serious about the dangers of quantum computing](#), 2023.
522. Raheman, F. [The Q-Day Dilemma and the Quantum Supremacy/Advantage Conjecture](#), 2022. <https://doi.org/10.21203/rs.3.rs-2331935/v1>.
523. Rivest, R.L.; Shamir, A.; Adleman, L. [A method for obtaining digital signatures and public-key cryptosystems](#). *Commun. ACM* **1978**, *21*, 120–126. <https://doi.org/10.1145/359340.359342>.
524. Diffie, W.; Hellman, M. [New directions in cryptography](#). *IEEE Transactions on Information Theory* **1976**, *22*, 644–654. <https://doi.org/10.1109/TIT.1976.1055638>.
525. Merkle, R.C. [Secure communications over insecure channels](#). *Communications of the ACM* **1978**, *21*, 294–299. <https://doi.org/10.1145/359460.359473>.
526. Imam, R.; Areeb, Q.M.; Alturki, A.; Anwer, F. [Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status](#). *IEEE Access* **2021**, *9*, 155949–155976. <https://doi.org/10.1109/access.2021.3129224>.
527. Paillier, P. [Impossibility Proofs for RSA Signatures in the Standard Model](#). In *Topics in Cryptology – CT-RSA 2007*; Springer Berlin Heidelberg, 2006; pp. 31–48. [https://doi.org/10.1007/11967668\\_3](https://doi.org/10.1007/11967668_3).
528. Kleinjung, T.; Aoki, K.; Franke, J.; Lenstra, A.K.; Thomé, E.; Bos, J.W.; Gaudry, P.; Kruppa, A.; Montgomery, P.L.; Osvik, D.A.; et al. [Factorization of a 768-Bit RSA Modulus](#). In Proceedings of the Advances in Cryptology – CRYPTO 2010; Rabin, T., Ed., Berlin, Heidelberg, 2010; pp. 333–350.
529. Dritsas, E.; Trigka, M.; Mylonas, P. [Performance and Security Analysis of the Diffie-Hellman Key Exchange Protocol](#). In Proceedings of the 2024 19th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP). IEEE, 2024, pp. 166–171. <https://doi.org/10.1109/smap63474.2024.00039>.
530. Sarkar, A.; Guha Roy, D.; Datta, P. [An Overview of the Discrete Logarithm Problem in Cryptography](#). In *Proceedings of Third International Conference on Advanced Computing and Applications*; Springer Nature Singapore, 2024; pp. 129–143. [https://doi.org/10.1007/978-981-97-4799-3\\_10](https://doi.org/10.1007/978-981-97-4799-3_10).
531. KONOMA, C. [The Computational Difficulty of Solving Cryptographic Primitive Problems Related to the Discrete Logarithm Problem](#). *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **2005**, *E88-A*, 81–88. <https://doi.org/10.1093/ietfec/e88-a.1.81>.

532. Kute, S.; Desai, C.; Jadhav, M. **Analysis of RSA and Shor's algorithm for cryptography: A quantum perspective**. In Proceedings of the ANNUAL SYMPOSIUM ON APPLIED AND INNOVATION TECHNOLOGICAL ENVIRONMENT 2023 (ASAITE2023): Smart Technology based on Revolution Industry 4.0 and Society 5.0. AIP Publishing, 2024, Vol. 3222, p. 040004. <https://doi.org/10.1063/5.0227773>.
533. Irwan, N.F.I.B.A.; Zawawi, M.N.A.; Thabit, R.A.A.B. **Investigating the Impact of Grover's Algorithm on AES S-Box**. In Proceedings of the 2024 16th International Conference on Knowledge and System Engineering (KSE). IEEE, 2024, pp. 379–385. <https://doi.org/10.1109/KSE63888.2024.11063483>.
534. Olaoye, G. **Quantum Cryptanalysis: Breaking Classical Encryption with Shor's and Grover's Algorithms**, 2025. <https://doi.org/10.22541/au.174431280.02434905/v1>.
535. Hu, F.; Lamata, L.; Sanz, M.; Chen, X.; Chen, X.; Wang, C.; Solano, E. **Quantum computing cryptography: Finding cryptographic Boolean functions with quantum annealing by a 2000 qubit D-wave quantum computer**. *Physics Letters A* **2020**, *384*, 126214. <https://doi.org/10.1016/j.physleta.2019.126214>.
536. Zhang, A.; Feng, X. **Applications of Quantum Annealing in Cryptography**, 2022, [arXiv:cs.CR/2211.10076].
537. Wang, F. **The Hidden Subgroup Problem**, 2010. arXiv:1008.0010 [quant-ph].
538. Fowler, A.G.; Devitt, S.J.; Hollenberg, L.C.L. **Implementation of Shor's Algorithm on a Linear Nearest Neighbour Qubit Array**. *Quant. Info. Comput.* **4**, 237-251 (2004) **2004**, [arXiv:quant-ph/quant-ph/0402196]. <https://doi.org/10.48550/ARXIV.QUANT-PH/0402196>.
539. Albuainain, A.; Alansari, J.; Alrashidi, S.; Alqahtani, W.; Alshaya, J.; Nagy, N. **Experimental Implementation of Shor's Quantum Algorithm to Break RSA**. In Proceedings of the 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2022, pp. 748–752. <https://doi.org/10.1109/cicn56167.2022.10008287>.
540. Kumar, M.; Mondal, B. **Study on Implementation of Shor's Factorization Algorithm on Quantum Computer**. *SN Computer Science* **2024**, *5*. <https://doi.org/10.1007/s42979-024-02771-y>.
541. Gidney, C.; Ekerå, M. **How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits**. *Quantum* **2021**, *5*, 433. <https://doi.org/10.22331/q-2021-04-15-433>.
542. Gidney, C. **How to factor 2048 bit RSA integers with less than a million noisy qubits**. *arXiv* **2025**, [arXiv:quant-ph/2505.15917]. <https://doi.org/10.48550/ARXIV.2505.15917>.
543. Ivezic, M. **Q-Day Revisited – RSA-2048 broken by 2030**, 2025.
544. Sevilla, J.; Riedel, C.J. **Forecasting timelines of quantum computing**, 2020, [arXiv:quant-ph/2009.05045].
545. Jiang, S.; Britt, K.A.; McCaskey, A.J.; Humble, T.S.; Kais, S. **Quantum Annealing for Prime Factorization**, 2018, [arXiv:quant-ph/1804.02733].
546. Hong, C.; Pei, Z.; Wang, Q.; Yang, S.; Yu, J.; Wang, C. **Quantum attack on RSA by D-Wave Advantage: a first break of 80-bit RSA**. *Science China Information Sciences* **2025**, *68*. <https://doi.org/10.1007/s11432-024-4163-6>.
547. Zhang, A.; Feng, X. **Applications of Quantum Annealing in Cryptography**, 2022, [arXiv:cs.CR/2211.10076].
548. Ding, J.; Spallitta, G.; Sebastiani, R. **Effective Prime Factorization via Quantum Annealing by Modular Locally-structured Embedding**, 2023, [arXiv:quant-ph/2310.17574].
549. Mosca, M. *Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*; Springer International Publishing, 2014. <https://doi.org/10.1007/978-3-319-11659-4>.
550. Dam, D.T.; Tran, T.H.; Hoang, V.P.; Pham, C.K.; Hoang, T.T. **A Survey of Post-Quantum Cryptography: Start of a New Race**. *Cryptography* **2023**, *7*, 40. <https://doi.org/10.3390/cryptography7030040>.
551. Bavdekar, R.; Jayant Chopde, E.; Agrawal, A.; Bhatia, A.; Tiwari, K. **Post Quantum Cryptography: A Review of Techniques, Challenges and Standardizations**. In Proceedings of the 2023 International Conference on Information Networking (ICOIN). IEEE, 2023, pp. 146–151. <https://doi.org/10.1109/icoin56518.2023.10048976>.
552. Lohmiller, N.; Kaniewski, S.; Menth, M.; Heer, T. **A Survey of Post-Quantum Cryptography Migration in Vehicles**. *IEEE Access* **2025**, *13*, 10160–10176. <https://doi.org/10.1109/access.2025.3528562>.
553. Boutin, C. **NIST Announces First Four Quantum-Resistant Cryptographic Algorithms**. *NIST* **2022**.
554. Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.K.; Miller, C.; Moody, D.; et al. **Status report on the third round of the NIST Post-Quantum Cryptography Standardization process**. Technical Report NIST IR 8413, National Institute of Standards and Technology (U.S.), Gaithersburg, MD, 2022. <https://doi.org/10.6028/NIST.IR.8413>.
555. **Announcing Issuance of Federal Information Processing Standards (FIPS) FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS 204, Module-Lattice-Based Digital Signature Standard, and FIPS 205, Stateless Hash-Based Digital Signature Standard**, 2024.

556. Beck, B.; Benjamin, D.; O'Brien, D.; Adrian, D. [Advancing Our Amazing Bet on Asymmetric Cryptography](#), 2024.
557. [iMessage with PQ3: The new state of the art in quantum-secure messaging at scale - Apple Security Research](#), 2024.
558. Ajtai, M.; Dwork, C. [A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence](#), 1996.
559. Overbeck, R.; Sendrier, N. [Code-based cryptography](#). In *Post-Quantum Cryptography*; Bernstein, D.J.; Buchmann, J.; Dahmen, E., Eds.; Springer: Berlin, Heidelberg, 2009; pp. 95–145. [https://doi.org/10.1007/978-3-540-88702-7\\_4](https://doi.org/10.1007/978-3-540-88702-7_4).
560. Moody, D. [The 2nd Round of the NIST PQC Standardization Process](#), 2019.
561. Green, M. [Hash-based Signatures: An illustrated Primer](#), 2018.
562. Relyea, R. [Post-quantum cryptography: Hash-based signatures](#), 2022. publisher: Red Hat.
563. National Cyber Security Centre (NCSC). [Cyber chiefs unveil new roadmap for post-quantum cryptography migration](#). Technical report, National Cyber Security Centre (NCSC), UK, 2025. Accessed: 2025-08-14.
564. SIKE – Supersingular Isogeny Key Encapsulation. [SIKE – Supersingular Isogeny Key Encapsulation](#). SIKE homepage, n.d. Accessed: 2025-08-14.
565. Costello, C.; Longa, P.; Naehrig, M.; Renes, J.; Virdia, F., [Improved Classical Cryptanalysis of SIKE in Practice](#). In *Public-Key Cryptography – PKC 2020*; Springer International Publishing, 2020; pp. 505–534. [https://doi.org/10.1007/978-3-030-45388-6\\_18](https://doi.org/10.1007/978-3-030-45388-6_18).
566. Alagic, G.; Bros, M.; Ciadoux, P.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.K.; Miller, C.; et al. [Status report on the fourth round of the NIST post-quantum cryptography standardization process](#), 2025. <https://doi.org/10.6028/nist.ir.8545>.
567. Castryck, W.; Decru, T. [An efficient key recovery attack on SIDH](#), 2022. Publication info: Published by the IACR in EUROCRYPT 2023.
568. Bennett, C.H.; Brassard, G. [Quantum cryptography: Public key distribution and coin tossing](#). *Theoretical Computer Science* **1984**, 560, 7–11. <https://doi.org/10.1016/j.tcs.2014.05.025>.
569. Wolf, R. [Quantum Key Distribution: An Introduction with Exercises](#); Springer International Publishing, 2021. <https://doi.org/10.1007/978-3-030-73991-1>.
570. Wootters, W.K.; Zurek, W.H. [A single quantum cannot be cloned](#). *Nature* **1982**, 299, 802–803.
571. Bennett, C.H.; Brassard, G.; Robert, J.M. [Privacy Amplification by Public Discussion](#). *SIAM Journal on Computing* **1988**, 17, 210–229. <https://doi.org/10.1137/0217014>.
572. Ekert, A.K. [Quantum cryptography based on Bell's theorem](#). *Physical Review Letters* **1991**, 67, 661–663. <https://doi.org/10.1103/physrevlett.67.661>.
573. Nurhadi, A.I.; Syambas, N.R. [Quantum Key Distribution \(QKD\) Protocols: A Survey](#). In Proceedings of the 2018 4th International Conference on Wireless and Telematics (ICWT). IEEE, 2018, pp. 1–5. <https://doi.org/10.1109/icwt.2018.8527822>.
574. Cao, Y.; Zhao, Y.; Wang, Q.; Zhang, J.; Ng, S.X.; Hanzo, L. [The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet](#). *IEEE Communications Surveys & Tutorials* **2022**, 24, 839–894. <https://doi.org/10.1109/comst.2022.3144219>.
575. Dervisevic, E.; Tankovic, A.; Fazel, E.; Kompella, R.; Fazio, P.; Voznak, M.; Mehic, M. [Quantum Key Distribution Networks - Key Management: A Survey](#). *ACM Computing Surveys* **2025**, 57, 1–36. <https://doi.org/10.1145/3730575>.
576. Padavic-Callaghan, K. [China's city-wide quantum network](#). *New Scientist* **2023**, 259, 13. [https://doi.org/10.1016/S0262-4079\(23\)01771-2](https://doi.org/10.1016/S0262-4079(23)01771-2).
577. Wiesner, S. [Conjugate coding](#). *ACM SIGACT News* **1983**, 15, 78–88.
578. Jiang, Y.F.; Kent, A.; Pitalúa-García, D.; Yao, X.; Chen, X.; Huang, J.; Cowperthwaite, G.; Zheng, Q.; Li, H.; You, L.; et al. [Experimental practical quantum tokens with transaction time advantage](#). *arXiv preprint* **2024**, [arXiv:quant-ph/2408.13063]. <https://doi.org/10.48550/ARXIV.2408.13063>.
579. Ehara, Y.; Tada, M. [How to generate transparent random numbers using blockchain](#). In Proceedings of the 2018 International Symposium on Information Theory and Its Applications (ISITA). IEEE, 2018, pp. 169–173. <https://doi.org/10.23919/isita.2018.8664239>.
580. Bartoletti, M.; Pompianu, L., [An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns](#). In *Financial Cryptography and Data Security*; Springer International Publishing, 2017; pp. 494–509. [https://doi.org/10.1007/978-3-319-70278-0\\_31](https://doi.org/10.1007/978-3-319-70278-0_31).

581. Chatterjee, K.; Goharshady, A.K.; Pourdamghani, A. **Probabilistic Smart Contracts: Secure Randomness on the Blockchain**. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, 2019, pp. 403–412. <https://doi.org/10.1109/bloc.2019.8751326>.
582. L'Ecuyer, P. **Random Number Generation**. Lecture notes / Technical report Chapter or Report, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, n.d. Accessed: 2025-08-14.
583. Komarov, A. **Independent Functions or How to Create the Random Number Generator**, 2019. publisher: Serokell Labs.
584. L'Ecuyer, P. **Random Number Generation**. In *Handbook of Computational Statistics*; Springer Berlin Heidelberg, 2011; pp. 35–71. [https://doi.org/10.1007/978-3-642-21551-3\\_3](https://doi.org/10.1007/978-3-642-21551-3_3).
585. Blum, M.; Micali, S. **How to generate cryptographically strong sequences of pseudo random bits**; Association for Computing Machinery, 2019. <https://doi.org/10.1145/3335741.3335751>.
586. L'Ecuyer, P.; Simard, R. **TestU01: A C library for empirical testing of random number generators**. *ACM Transactions on Mathematical Software* **2007**, *33*, 1–40. <https://doi.org/10.1145/1268776.1268777>.
587. Pironio, S.; Acín, A.; Massar, S.; de la Giroday, A.B.; Matsukevich, D.N.; Maunz, P.; Olmschenk, S.; Hayes, D.; Luo, L.; Manning, T.A.; et al. **Random numbers certified by Bell's theorem**. *Nature* **2010**, *464*, 1021–1024. <https://doi.org/10.1038/nature09008>.
588. Aaronson, S.; Hung, S.H. **Certified Randomness from Quantum Supremacy**. In Proceedings of the Proceedings of the 55th Annual ACM Symposium on Theory of Computing. ACM, 2023, STOC '23, pp. 933–944. <https://doi.org/10.1145/3564246.3585145>.
589. Acín, A.; Masanes, L. **Certified randomness in quantum physics**. *Nature* **2016**, *540*, 213–219. <https://doi.org/10.1038/nature20119>.
590. Liu, M.; Shaydulin, R.; Niroula, P.; DeCross, M.; Hung, S.H.; Kon, W.Y.; Cervero-Martín, E.; Chakraborty, K.; Amer, O.; Aaronson, S.; et al. **Certified randomness using a trapped-ion quantum processor**. *Nature* **2025**, *640*, 343–348. <https://doi.org/10.1038/s41586-025-08737-1>.
591. Gheorghiu, A.; Kapourniotis, T.; Kashefi, E. **Verification of Quantum Computation: An Overview of Existing Approaches**. *Theory of Computing Systems* **2018**, *63*, 715–808. <https://doi.org/10.1007/s00224-018-9872-3>.
592. Carrasco, J.; Elben, A.; Kokail, C.; Kraus, B.; Zoller, P. **Theoretical and Experimental Perspectives of Quantum Verification**. *PRX Quantum* **2021**, *2*, 010102. <https://doi.org/10.1103/prxquantum.2.010102>.
593. Kapourniotis, T.; Kashefi, E.; Leichtle, D.; Music, L.; Ollivier, H. **Unifying quantum verification and error-detection: theory and tools for optimisations**. *Quantum Science and Technology* **2024**, *9*, 035036. <https://doi.org/10.1088/2058-9565/ad466d>.
594. Helm, T.; Low, A.; Townson, J. **UK FinTech State of the Nation**. Technical report, Department for International Trade, 2019.
595. Ramamurthy, S.; Sironi, P. **2025 Global Outlook for Banking and Financial Markets**. Technical report, IBM, 2025.
596. Youssef, W.A.B.; Mansour, N. **Finance in the Digital Age: The Challenges and Opportunities**. In Proceedings of the Technology: Toward Business Sustainability; Alareeni, B.; Hamdan, A., Eds., Cham, 2024; pp. 45–59.
597. International, K. **Voices on 2030: Financial Services reinvented**. Technical report, KPMG, 2022.
598. Bueno, L.A.; Sigahi, T.F.; Rampasso, I.S.; Leal Filho, W.; Anholon, R. **Impacts of digitization on operational efficiency in the banking sector: Thematic analysis and research agenda proposal**. *International Journal of Information Management Data Insights* **2024**, *4*, 100230. <https://doi.org/10.1016/j.jjime.2024.100230>.
599. Bodie, Z.; Kane, A.; Marcus, A.J. *Investments*, 13th ed.; McGraw-Hill Education, 2023.
600. Fama, E.F. **Efficient Capital Markets: A Review of Theory and Empirical Work**. *The Journal of Finance* **1970**, *25*, 383–417.
601. Hull, J. *Options, futures, and other derivatives*, eleventh edition, global edition ed.; Pearson4060 1 Online-Ressource (880 Seiten): Harlow, England, 2022. Description based on publisher supplied metadata and other sources.
602. Orús, R.; Mugel, S.; Lizaso, E. **Quantum computing for finance: Overview and prospects**. *Reviews in Physics* **2019**, *4*, 100028.
603. Föllmer, H.; Schied, A. *Stochastic finance*, fourth revised and extended edition ed.; De Gruyter graduate, De Gruyter: Berlin, 2016. Literaturverzeichnis: Seite 576-586.
604. Gu, S.; Kelly, B.; Xiu, D. **Empirical Asset Pricing via Machine Learning**. *The Review of Financial Studies* **2020**, *33*, 2223–2273, [<https://academic.oup.com/rfs/article-pdf/33/5/2223/33209812/hhaa009.pdf>]. <https://doi.org/10.1093/rfs/hhaa009>.

605. Vaheb, H. *Asset Price Forecasting using Recurrent Neural Networks*, 2020. <https://doi.org/10.48550/ARXIV.2010.06417>.
606. Bausch, J. *Recurrent Quantum Neural Networks*. In *Proceedings of the Advances in Neural Information Processing Systems*; Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; Lin, H., Eds. Curran Associates, Inc., 2020, Vol. 33, pp. 1368–1379.
607. Assouel, A.; Jacquier, A.; Kondratyev, A. *A Quantum Generative Adversarial Network for distributions*, 2021. <https://doi.org/10.48550/ARXIV.2110.02742>.
608. Bao, J.; Rebstrost, P. *Fundamental theorem for quantum asset pricing*. *arXiv* **2022**, [arXiv:quant-ph/2212.13815]. <https://doi.org/10.48550/ARXIV.2212.13815>.
609. Orrell, D. *Quantum economics and finance*, third edition ed.; Panda Ohana Publishing: New York City, NY, 2022.
610. Accardi, L.; Boukas, A. *The Quantum Black-Scholes Equation*. *GJPAM*, vol. 2, no. 2, pp. 155-170 (2006) **2007**, [arXiv:q-fin.PR/0706.1300]. <https://doi.org/10.48550/ARXIV.0706.1300>.
611. Insights, B.R. *Derivatives Market Size, Share, Growth, Trends, Global Industry Analysis*, 2025. Last accessed: 20-02-2025.
612. Wang, G.; Kan, A. *Option pricing under stochastic volatility on a quantum computer*. *Quantum* **2024**, 8, 1504. <https://doi.org/10.22331/q-2024-10-23-1504>.
613. Chen, J.; Li, Y.; Neufeld, A. *Quantum Monte Carlo algorithm for solving Black-Scholes PDEs for high-dimensional option pricing in finance and its complexity analysis*, 2025, [arXiv:quant-ph/2301.09241].
614. Fontanela, F.; Jacquier, A.; Oumgari, M. *Short Communication: A Quantum Algorithm for Linear PDEs Arising in Finance*. *SIAM Journal on Financial Mathematics* **2021**, 12, SC98–SC114. <https://doi.org/10.1137/21m1397878>.
615. FIA. *ETD Volume - January 2025*, 2025. Last accessed: 20-02-2025.
616. SEC, U.S. *Investor Bulletin: An Introduction to Options*, 2015. Last accessed: 20-02-2025.
617. Chen, J. *What Are Options? Types, Spreads, Example, and Risk*, 2024. Last accessed: 20-02-2025.
618. Mikosch, T. *Elementary Stochastic Calculus, with Finance in View*; WORLD SCIENTIFIC, 1998; [<https://worldscientific.com/doi/pdf/10.1142/3856>]. <https://doi.org/10.1142/3856>.
619. Rebstrost, P.; Gupt, B.; Bromley, T.R. *Quantum computational finance: Monte Carlo pricing of financial derivatives*. *Physical Review A* **2018**, 98, 022321. <https://doi.org/10.1103/physreva.98.022321>.
620. Stamatopoulos, N.; Egger, D.J.; Sun, Y.; Zoufal, C.; Iten, R.; Shen, N.; Woerner, S. *Option Pricing using Quantum Computers*. *Quantum* **2020**, 4, 291. <https://doi.org/10.22331/q-2020-07-06-291>.
621. Ramos-Calderer, S.; Pérez-Salinas, A.; García-Martín, D.; Bravo-Prieto, C.; Cortada, J.; Planagumà, J.; Latorre, J.I. *Quantum unary approach to option pricing*. *Physical Review A* **2021**, 103, 032414. <https://doi.org/10.1103/physreva.103.032414>.
622. Kaneko, K.; Miyamoto, K.; Takeda, N.; Yoshino, K. *Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance*. *Quantum Information Processing* **2021**, 20. <https://doi.org/10.1007/s11128-021-03127-8>.
623. Beck, C.; Hutzenthaler, M.; Jentzen, A.; Kuckuck, B. *An overview on deep learning-based approximation methods for partial differential equations*. *Discrete Contin. Dyn. Syst. Ser. B* **28** (2023), no. 6, 3697–3746 **2020**, 28, 3697–3746, [arXiv:math.NA/2012.12348]. <https://doi.org/10.3934/dcdsb.2022238>.
624. Brunton, S.L.; Kutz, J.N. *Promising directions of machine learning for partial differential equations*. *Nature Computational Science* **2024**, 4, 483–494. <https://doi.org/10.1038/s43588-024-00643-2>.
625. Joo, J.; Moon, H. *Quantum variational PDE solver with machine learning*. *arXiv* **2021**, [arXiv:quant-ph/2109.09216]. <https://doi.org/10.48550/ARXIV.2109.09216>.
626. Tavakoli, J.M. *Collateralized Debt Obligations and Structured Finance*; John Wiley & Sons, Ltd., 2004.
627. Tang, H.; Pal, A.; Wang, T.Y.; Qiao, L.F.; Gao, J.; Jin, X.M. *Quantum computation for pricing the collateralized debt obligations*. *Quantum Engineering* **2021**, 3, e84, [<https://onlinelibrary.wiley.com/doi/pdf/10.1002/que2.84>]. <https://doi.org/10.1002/que2.84>.
628. Li, D.X. *On Default Correlation*. *The Journal of Fixed Income* **2000**, 9, 43–54, [<https://jfi.pm-research.com/content/9/4/43.full.pdf>]. <https://doi.org/10.3905/jfi.2000.319253>.
629. Barndorff-Nielsen, O.E. *Normal Inverse Gaussian Distributions and Stochastic Volatility Modelling*. *Scandinavian Journal of Statistics* **1997**, 24, 1–13, [<https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9469.00045>]. <https://doi.org/10.1111/1467-9469.00045>.
630. Chen, J. *What Is a Swap?*, 2025. Accessed: 2025-08-29.

631. Baker, L.; Haynes, R.; Roberts, J.; Sharma, R.; Tuckman, B. Risk Transfer with Interest Rate Swaps. *Financial Markets, Institutions; Instruments* **2020**, *30*, 3–28. <https://doi.org/10.1111/fmii.12135>.
632. CFI Team. *Netting*, 2025. Accessed: 2025-08-29.
633. Schuldenzucker, S.; Seuken, S.; Battiston, S. The Computational Complexity of Financial Networks with Credit Default Swaps. *arXiv* **2017**, [arXiv:q-fin.RM/1710.01578]. <https://doi.org/10.48550/ARXIV.1710.01578>.
634. Rosenberg, G.; Adolphs, C.; Milne, A.; Lee, A. Swap netting using a quantum annealer, 2016.
635. Boyd, S.; Johansson, K.; Kahn, R.; Schiele, P.; Schmelzer, T. Markowitz Portfolio Construction at Seventy, 2024, [arXiv:q-fin.PM/2401.05080].
636. DeMiguel, V.; Garlappi, L.; Nogales, F.J.; Uppal, R. A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms. *Management Science* **2009**, *55*, 798–812.
637. Rosenberg, G.; Rounds, M. Long-Short Minimum Risk Parity Optimization Using a Quantum or Digital Annealer, 2018.
638. Benati, S.; Rizzi, R. A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem. *European Journal of Operational Research* **2007**, *176*, 423–434. <https://doi.org/10.1016/j.ejor.2005.07.020>.
639. Mansini, R.; Ogryczak, W.; Speranza, M.G. *Linear and Mixed Integer Programming for Portfolio Optimization*. Springer **2015**.
640. Poon, H.K. *Practical Portfolio Optimization with Metaheuristics: Pre-assignment Constraint and Margin Trading*, 2025, [arXiv:q-fin.PM/2503.15965].
641. Deng, W.; Polak, P.; Safikhani, A.; Shah, R. A Unified Framework for Fast Large-Scale Portfolio Optimization, 2023, [arXiv:q-fin.PM/2303.12751].
642. Skolpadungket, P.; Dahal, K.; Harnpornchai, N. Portfolio optimization using multi-objective genetic algorithms. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 516–523.
643. Owhadi-Kareshk, M.; Boulanger, P. Portfolio Optimization on Classical and Quantum Computers Using PortFawn, 2021. <https://doi.org/10.48550/ARXIV.2112.08998>.
644. Palmer, S.; Sahin, S.; Hernandez, R.; Mugel, S.; Orus, R. Quantum Portfolio Optimization with Investment Bands and Target Volatility, 2021. <https://doi.org/10.48550/ARXIV.2106.06735>.
645. Hodson, M.; Ruck, B.; Ong, H.; Garvin, D.; Dulman, S. Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz, publisher = arXiv, 2019. <https://doi.org/10.48550/ARXIV.1911.05296>.
646. Slate, N.; Matwiejew, E.; Marsh, S.; Wang, J.B. Quantum walk-based portfolio optimisation. *Quantum* **2021**, *5*, 513. <https://doi.org/10.22331/q-2021-07-28-513>.
647. Rosenberg, G.; Haghnegahdar, P.; Goddard, P.; Carr, P.; Wu, K.; de Prado, M.L. Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer. *IEEE Journal of Selected Topics in Signal Processing* **2016**, *10*, 1053–1060. <https://doi.org/10.1109/JSTSP.2016.2574703>.
648. Mugel, S.; Kuchkovsky, C.; Sánchez, E.; Fernández-Lorenzo, S.; Luis-Hita, J.; Lizaso, E.; Orús, R. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Phys. Rev. Research* **2022**, *4*, 013006. <https://doi.org/10.1103/PhysRevResearch.4.013006>.
649. Manouchehri, K.; Wang, J.B. Quantum Walks in an array of Quantum Dots, 2006. <https://doi.org/10.48550/ARXIV.QUANT-PH/0609088>.
650. Hadfield, S.; Wang, Z.; O’Gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* **2019**, *12*. <https://doi.org/10.3390/a12020034>.
651. Cornuejols, G.; Fisher, M.L.; Nemhauser, G.L. Exceptional Paper—Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. *Management Science* **1977**, *23*, 789–810, [https://doi.org/10.1287/mnsc.23.8.789]. <https://doi.org/10.1287/mnsc.23.8.789>.
652. Catalano, T.J. The Most Effective Hedging Strategies to Reduce Market Risk, 2024. Last accessed: 16-04-2025.
653. MARKOWITZ, H.M. Foundations of Portfolio Theory. *The Journal of Finance* **1991**, *46*, 469–477. <https://doi.org/10.1111/j.1540-6261.1991.tb02669.x>.
654. Kalra, A.; Qureshi, F.; Tisi, M. Portfolio Asset Identification Using Graph Algorithms on a Quantum Annealer. *SSRN Electronic Journal* **2018**. <https://doi.org/10.2139/ssrn.3333537>.
655. Loader, D. The structure of clearing and settlement. In *Clearing, Settlement and Custody*; Loader, D., Ed.; Butterworth-Heinemann: Oxford, 2002; pp. 1–18. <https://doi.org/10.1016/B978-075065484-5.50001-3>.
656. McGill, R.; Patel, N., *Clearing & Settlement*. In *Global Custody and Clearing Services*; Palgrave Macmillan UK, 2008; pp. 43–58. [https://doi.org/10.1057/9780230592445\\_4](https://doi.org/10.1057/9780230592445_4).

657. Bak0161ys, D.; Sakalauskas, L. *Modelling, Simulation and Optimisation of Interbank Settlements*. *Information Technology and Control* **2007**, *36*. Accessed: 2025-08-29.
658. Braine, L.; Egger, D.J.; Glick, J.; Woerner, S. *Quantum Algorithms for Mixed Binary Optimization Applied to Transaction Settlement*. *IEEE Transactions on Quantum Engineering* **2021**, *2*, 1–8. <https://doi.org/10.1109/tqe.2021.3063635>.
659. Huber, E.X.; Tan, B.Y.L.; Griffin, P.R.; Angelakis, D.G. *Exponential qubit reduction in optimization for financial transaction settlement*. *EPJ Quantum Technology* **2024**, *11*. <https://doi.org/10.1140/epjqt/s40507-024-00262-w>.
660. Ross, S. *Return, Risk and Arbitrage*; Vol. Vol. I, Rodney L. White Center for Financial Research, The Wharton School, 1977.
661. Dybvig, P.H.; Ross, S.A., *Arbitrage*. In *The New Palgrave Dictionary of Economics*; Palgrave Macmillan UK, 2008; pp. 1–12. [https://doi.org/10.1057/978-1-349-95121-5\\_449-2](https://doi.org/10.1057/978-1-349-95121-5_449-2).
662. Aquilina, M.; Budish, E.; O'Neill, P. *Quantifying the High-Frequency Trading 'Arms Race': A Simple New Methodology and Estimates*. Technical report, Financial Conduct Authority, 2020.
663. Borch, C.; Wosnitzer, R., Eds. *The Routledge handbook of critical finance studies*; Routledge international handbooks, Routledge, Taylor & Francis Group: New York, 2021.
664. Soon, W.; Ye, H.Q. *Currency arbitrage detection using a binary integer programming model*. In Proceedings of the 2007 IEEE International Conference on Industrial Engineering and Engineering Management, 2007, pp. 867–870. <https://doi.org/10.1109/IEEM.2007.4419314>.
665. Bellman, R. *On a routing problem*. *Quarterly of Applied Mathematics* **1958**, *16*, 87–90. <https://doi.org/10.1090/qam/102435>.
666. Floyd, R.W. *Algorithm 97: Shortest path*. *Commun. ACM* **1962**, *5*, 345. <https://doi.org/10.1145/367766.368168>.
667. Karp, R.M., *Reducibility among Combinatorial Problems*. In *Complexity of Computer Computations*; Springer US, 1972; p. 85–103. [https://doi.org/10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9).
668. Deshpande, S.; Das, E.R.; Mueller, F. *Currency Arbitrage Optimization using Quantum Annealing, QAOA and Constraint Mapping*, 2025. <https://doi.org/10.48550/ARXIV.2502.15742>.
669. Wu, W.; Wang, Y.; Yan, G.; Zhao, Y.; Zhang, B.; Yan, J. *On Reducing the Execution Latency of Superconducting Quantum Processors via Quantum Job Scheduling*. In Proceedings of the Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. ACM, 2024, ICCAD '24, p. 1–9. <https://doi.org/10.1145/3676536.3676678>.
670. Capponi, A.; (eds), C.A.L., *NLP in Finance*. In *Machine Learning and Data Sciences for Financial Markets: A Guide to Contemporary Practices*; Cambridge University Press, 2023; p. 563–592.
671. Boleda, G. *Distributional semantics and linguistic theory*. *Annual Review of Linguistics* **2020**, *6*, 213–234.
672. Stein, J.; Christ, I.; Kraus, N.; Mansky, M.B.; Müller, R.; Linnhoff-Popien, C. *Applying QNLP to Sentiment Analysis in Finance*. In Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), 2023, Vol. 02, pp. 20–25. <https://doi.org/10.1109/QCE57702.2023.10178>.
673. Steck, H.; Ekanadham, C.; Kallus, N. *Is Cosine-Similarity of Embeddings Really About Similarity?* In Proceedings of the Companion Proceedings of the ACM Web Conference 2024. ACM, 2024, WWW '24, p. 887–890. <https://doi.org/10.1145/3589335.3651526>.
674. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. *Efficient Estimation of Word Representations in Vector Space*, 2013, [arXiv:cs.CL/1301.3781].
675. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. *Distributed Representations of Words and Phrases and their Compositionality*. In Proceedings of the Advances in Neural Information Processing Systems; Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K., Eds. Curran Associates, Inc., 2013, Vol. 26.
676. Mikolov, T.; Yih, W.t.; Zweig, G. *Linguistic regularities in continuous space word representations*. In Proceedings of the Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies, 2013, pp. 746–751.
677. Widdows, D.; Aboumrada, W.; Kim, D.; Ray, S.; Mei, J. *Quantum Natural Language Processing*. *KI - Künstliche Intelligenz* **2024**, *38*, 293–310. <https://doi.org/10.1007/s13218-024-00861-w>.
678. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019, [arXiv:cs.CL/1810.04805].
679. Radford, A.; Narasimhan, K. *Improving Language Understanding by Generative Pre-Training*, 2018.

680. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *32*, 604–624. <https://doi.org/10.1109/TNNLS.2020.2979670>.
681. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019; pp. 4171–4186. <https://doi.org/10.18653/v1/N19-1423>.
682. Naseem, U.; Razzak, I.; Khan, S.K.; Prasad, M. A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*. <https://doi.org/10.1145/3434237>.
683. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds and Machines* **2020**, *110*.
684. Du, K.; Zhao, Y.; Mao, R.; Xing, F.; Cambria, E. Natural language processing in finance: A survey. *Information Fusion* **2025**, *115*, 102755. <https://doi.org/10.1016/j.inffus.2024.102755>.
685. OpenAI. Learning to reason with LLMs, 2024. [Accessed: 03.01.2025].
686. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP, 2019, [arXiv:cs.CL/1906.02243].
687. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Proceedings of the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, New York, NY, USA, 2021; FAccT '21, p. 610–623. <https://doi.org/10.1145/3442188.3445922>.
688. Sadrzadeh, M. High Level Quantum Structures in Linguistics and Multi-Agent Systems. In Proceedings of the AAAI Spring Symposia on Quantum Interactions (01/03/07); Bruza, P.; Lawless, W.; van Rijsbergen, C.J., Eds., 2006. Event Dates: March 2007.
689. Coecke, B.; Sadrzadeh, M.; Clark, S. Mathematical Foundations for a Compositional Distributional Model of Meaning, 2010, [arXiv:cs.CL/1003.4394]. <https://doi.org/10.48550/ARXIV.1003.4394>.
690. Coecke, B.; de Felice, G.; Meichanetzidis, K.; Toumi, A. Foundations for Near-Term Quantum Natural Language Processing, 2020. <https://doi.org/10.48550/ARXIV.2012.03755>.
691. Abramsky, S.; Coecke, B. Categorical quantum mechanics, 2008. <https://doi.org/10.48550/ARXIV.0808.1023>.
692. Coecke, B.; Duncan, R. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics* **2011**, *13*, 043016. <https://doi.org/10.1088/1367-2630/13/4/043016>.
693. Takaki, Y.; Mitarai, K.; Negoro, M.; Fujii, K.; Kitagawa, M. Learning temporal data with a variational quantum recurrent neural network. *Phys. Rev. A* **2021**, *103*, 052414. <https://doi.org/10.1103/PhysRevA.103.052414>.
694. de Felice, G.; Toumi, A.; Coecke, B. DisCoPy: Monoidal Categories in Python. *Electronic Proceedings in Theoretical Computer Science* **2021**, *333*, 183–197. <https://doi.org/10.4204/eptcs.333.13>.
695. Lorenz, R.; Pearson, A.; Meichanetzidis, K.; Kartsaklis, D.; Coecke, B. QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer, 2021. <https://doi.org/10.48550/ARXIV.2102.12846>.
696. Meichanetzidis, K.; Toumi, A.; de Felice, G.; Coecke, B. Grammar-aware sentence classification on quantum computers. *Quantum Machine Intelligence* **2023**, *5*. <https://doi.org/10.1007/s42484-023-00097-1>.
697. Meichanetzidis, K.; Gogioso, S.; de Felice, G.; Chiappori, N.; Toumi, A.; Coecke, B. Quantum Natural Language Processing on Near-Term Quantum Computers. *Electronic Proceedings in Theoretical Computer Science* **2021**, *340*, 213–229. <https://doi.org/10.4204/eptcs.340.11>.
698. Majumder, A.; Zimmerman, C.; Zhu, D.; Alexander, A.; Widdows, D. Near-term advances in quantum natural language processing. *Ann. Math. Artif. Intell.* **2024**, *92*, 1249–1272, [arXiv:cs.CL/2206.02171]. <https://doi.org/10.1007/s10472-024-09940-y>.
699. Duneau, T.; Bruhn, S.; Matos, G.; Laakkonen, T.; Saiti, K.; Pearson, A.; Meichanetzidis, K.; Coecke, B. Scalable and interpretable quantum natural language processing: an implementation on trapped ions, 2024, [arXiv:quant-ph/2409.08777].
700. Zeng, W.; Coecke, B. Quantum Algorithms for Compositional Natural Language Processing. *Electronic Proceedings in Theoretical Computer Science* **2016**, *221*, 67–75. <https://doi.org/10.4204/eptcs.221.8>.
701. Yang, X.; Zhu, J.; De Meo, P. A quantum-like zero-shot approach for sentiment analysis in finance. *Journal of Intelligent Information Systems* **2024**. <https://doi.org/10.1007/s10844-024-00912-6>.
702. Zhou, J. Quantum Finance: Exploring the Implications of Quantum Computing on Financial Models. *Computational Economics* **2025**.
703. Maurer, T.; Nelson, A. The global cyber threat to financial systems, 2021.

704. Company, M.. [The cyber clock is ticking: Derisking emerging technologies in financial services](#) | McKinsey, 2024.
705. Hull, J. [Risk management and financial institutions](#), sixth edition ed.; Wiley finance, Wiley: Hoboken, New Jersey, 2023. Includes index.
706. National Cyber Security Centre. [Cyber security risk management framework](#), 2023.
707. Wu, D.D.; Olson, D.L. [Computational simulation and risk analysis: An introduction of state of the art research](#). *Mathematical and Computer Modelling* **2013**, *58*, 1581–1587. Computational simulation and risk analysis, <https://doi.org/10.1016/j.mcm.2013.07.004>.
708. Evrin, V. [Risk Assessment and Analysis Methods: Qualitative and Quantitative](#), 2021.
709. Rumasukun, M.R.; Noch, M.Y. [Exploring Financial Risk Management: A Qualitative Study on Risk Identification, Evaluation, and Mitigation in Banking, Insurance, and Corporate Finance](#). *Jurnal Manajemen Bisnis* **2024**, *11*, 1068–1083. <https://doi.org/10.33096/jmb.v11i2.903>.
710. Woerner, S.; Egger, D.J. [Quantum risk analysis](#). *npj Quantum Information* **2019**, *5*. <https://doi.org/10.1038/s41534-019-0130-6>.
711. Laudagé, C.; Turkalj, I. [Quantum Risk Analysis: Beyond \(Conditional\) Value-at-Risk](#), 2025, [arXiv:quant-ph/2211.04456].
712. Stamatopoulos, N.; Clader, B.D.; Woerner, S.; Zeng, W.J. [Quantum Risk Analysis of Financial Derivatives](#), 2024, [arXiv:quant-ph/2404.10088].
713. Miyamoto, K. [Quantum algorithm for calculating risk contributions in a credit portfolio](#). *EPJ Quantum Technology* **2022**, *9*, 20.
714. Wilkens, S.; Moorhouse, J. [Quantum computing for financial risk measurement](#). *Quantum Information Processing* **2023**, *22*, 51. <https://doi.org/10.1007/s11128-022-03777-2>.
715. Bank for International Settlements. [MAR10 - Market Risk Terminologies](#), 2020. Accessed: 2025-04-26.
716. Basel Committee on Banking Supervision. [Revisions to the Basel II Market risk framework](#), 2009. Accessed: 2025-04-26.
717. Jorion, P. [Value at risk](#), 3. ed., [nachdr.] ed.; McGraw-Hill: New York, NY [u.a.], 2009. Literaturverz. S. 573 - 584.
718. Hong, L.J.; Hu, Z.; Liu, G. [Monte Carlo Methods for Value-at-Risk and Conditional Value-at-Risk: A Review](#). *ACM Trans. Model. Comput. Simul.* **2014**, *24*. <https://doi.org/10.1145/2661631>.
719. Egger, D.J.; García Gutiérrez, R.; Mestre, J.C.; Woerner, S. [Credit Risk Analysis Using Quantum Computers](#). *IEEE Transactions on Computers* **2021**, *70*, 2136–2145. <https://doi.org/10.1109/TC.2020.3038063>.
720. Saunders, A.; Allen, L. [Credit risk measurement](#), 2nd ed ed.; Number v.154 in Wiley Finance, John Wiley: New York, 2002. Includes bibliographical references (p. 258-275) and index.
721. Crouhy, M.; Galai, D.; Mark, R. [Risk management](#), [reprint] ed.; McGraw-Hill: New York [u.a.], 2009.
722. Thomas, L.C.; Edelman, D.; Crook, J.N. [Credit scoring and its applications](#), second edition ed.; Number 2 in Mathematics in industry, SIAM, Society for Industrial and Applied Mathematics: Philadelphia, 2017. Includes bibliographical references and index.
723. Teles, G.; Rodrigues, J.J.P.C.; Saleem, K.; Kozlov, S.; Rabêlo, R.A.L. [Machine learning and decision support system on credit scoring](#). *Neural Computing and Applications* **2019**, *32*, 9809–9826. <https://doi.org/10.1007/s00521-019-04537-7>.
724. Dastile, X.; Celik, T.; Potsane, M. [Statistical and machine learning models in credit scoring: A systematic literature survey](#). *Applied Soft Computing* **2020**, *91*, 106263. <https://doi.org/10.1016/j.asoc.2020.106263>.
725. Ala'raj, M.; Abbod, M.F.; Majdalawieh, M.; Jum'a, L. [A deep learning model for behavioural credit scoring in banks](#). *Neural Computing and Applications* **2022**, *34*, 5839–5866. <https://doi.org/10.1007/s00521-021-06695-z>.
726. Bluhm, C.; Overbeck, L.; Wagner, C. [Introduction to Credit Risk Modeling](#), second edition ed.; Chapman & Hall/CRC Financial Mathematics Series, CRC Press: Boca Raton, 2010.
727. Dri, E.; Aita, A.; Giusto, E.; Ricossa, D.; Corbelleto, D.; Montrucchio, B.; Ugoccioni, R. [A More General Quantum Credit Risk Analysis Framework](#). *Entropy* **2023**, *25*, 593. <https://doi.org/10.3390/e25040593>.
728. Sironi, A.; Resti, A. [Risk management and shareholders' value in banking](#); Wiley finance, Wiley: Chichester, West Sussex [England], 2007. Includes bibliographical references pages 759-770 and index.
729. Rockafellar, R.T.; Uryasev, S. [Optimization of conditional value-at-risk](#). *The Journal of Risk* **2000**, *2*, 21–41. <https://doi.org/10.21314/jor.2000.038>.
730. Glasserman, P. [Measuring Marginal Risk Contributions in Credit Portfolios](#). *SSRN Electronic Journal* **2005**. <https://doi.org/10.2139/ssrn.681227>.

731. Tasche, D. Capital Allocation to Business Units and Sub-Portfolios: the Euler Principle. *arXiv* **2007**, [arXiv:q-fin.PM/0708.2542]. <https://doi.org/10.48550/ARXIV.0708.2542>.
732. Rutkowski, M.; Tarca, S. Regulatory Capital Modelling for Credit Risk. *International Journal of Theoretical and Applied Finance* **2015**, *18*, 1550034. <https://doi.org/10.1142/s021902491550034x>.
733. World Bank Group. *Credit Scoring Approaches Guidelines*, 2019.
734. Dastile, X.; Çelik, T.; Potsane, M.M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Appl. Soft Comput.* **2020**, *91*, 106263.
735. Wang, Y.; Hu, Z.; Sanders, B.C.; Kais, S. Qudits and High-Dimensional Quantum Computing. *Frontiers in Physics* **2020**, *8*. <https://doi.org/10.3389/fphy.2020.589504>.
736. Milne, A.; Rounds, M.; Goddard, P. Optimal feature selection in credit scoring and classification using a quantum annealer, 2017.
737. Mancilla, J.; Sequeira, A.; Tagliani, T.; Llaneza, F.; Beiza, C. Empowering Credit Scoring Systems with Quantum-Enhanced Machine Learning. *arXiv preprint* **2024**, [arXiv:q-fin.RM/2404.00015]. <https://doi.org/10.48550/ARXIV.2404.00015>.
738. Schetakis, N.; Aghamalyan, D.; Boguslavsky, M.; Rees, A.; Rakotomalala, M.; Griffin, P.R. Quantum Machine Learning for Credit Scoring. *Mathematics* **2024**, *12*, 1391. <https://doi.org/10.3390/math12091391>.
739. Minati, R.; Hema, D. Quantum Powered Credit Risk Assessment: A Novel Approach using hybrid Quantum-Classical Deep Neural Network for Row-Type Dependent Predictive Analysis. *arXiv* **2025**, [arXiv:q-fin.CP/2502.07806]. <https://doi.org/10.48550/ARXIV.2502.07806>.
740. Chen, C.M.; Tso, G.K.F.; He, K. Quantum Optimized Cost Based Feature Selection and Credit Scoring for Mobile Micro-financing. *Computational Economics* **2023**, *63*, 919–950. <https://doi.org/10.1007/s10614-023-10365-8>.
741. Finance, U. *Annual Fraud Report*, 2024. Accessed: 2024-10-07.
742. Government, U. *Economic Crime and Corporate Transparency Act*, 2023.
743. Yndurain, E.; Woerner, S.; Egger, D. Exploring quantum computing use cases for financial services, 2019. Last accessed: 20-05-2022.
744. Berkson, J. Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association* **1944**, *39*, 357–365.
745. Balcan, M.F.; Sharma, D. Learning accurate and interpretable tree-based models, 2025, [arXiv:cs.LG/2405.15911].
746. Louppe, G. Understanding Random Forests: From Theory to Practice, 2015, [arXiv:stat.ML/1407.7502].
747. Moguerza, J.M.; Muñoz, A. Support Vector Machines with Applications. *Statistical Science* **2006**, *21*. <https://doi.org/10.1214/088342306000000493>.
748. Chen, Y.; Zhao, C.; Xu, Y.; Nie, C. Year-over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review, 2025, [arXiv:cs.LG/2502.00201].
749. Zhang, Z.; Zhou, X.; Zhang, X.; Wang, L.; Wang, P. A Model Based on Convolutional Neural Network for Online Transaction Fraud Detection. *Security and Communication Networks* **2018**, *2018*, 5680264, [https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/5680264]. <https://doi.org/10.1155/2018/5680264>.
750. Khalid, A.R.; Owoh, N.; Uthmani, O.; Ashawa, M.; Osamor, J.; Adejoh, J. Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach. *Big Data and Cognitive Computing* **2024**, *8*. <https://doi.org/10.3390/bdcc8010006>.
751. Grossi, M.; Ibrahim, N.; Radescu, V.; Loredò, R.; Voigt, K.; Altmann, C.V.; Rudnik, A. Mixed Quantum-Classical Method For Fraud Detection with Quantum Feature Selection, 2022, [arXiv:quant-ph/2208.07963].
752. Marini, F.; Muthu, K.; Singh, K.; Capozzi, M. How Deloitte Italy built a digital payments fraud detection solution using quantum machine learning and Amazon Braket, 2024.
753. Herr, D.; Obert, B.; Rosenkranz, M. Anomaly detection with variational quantum generative adversarial networks. *Quantum Science and Technology* **2021**, *6*, 045004. <https://doi.org/10.1088/2058-9565/ac0d4d>.
754. Islam, M.; Turkeli, S.; Ozaydin, F. A Survey of Quantum Generative Adversarial Networks: Architectures, Use Cases, and Real-World Implementations, 2025, [arXiv:quant-ph/2506.18002].
755. Ezugwu, A.E.; Ikotun, A.M.; Oyelade, O.O.; Abualigah, L.; Agushaka, J.O.; Eke, C.I.; Akinyelu, A.A. A Comprehensive Survey of Clustering Algorithms: State-of-the-Art Machine Learning Applications, Taxonomy, Challenges, and Future Research Prospects. *Eng. Appl. Artif. Intell.* **2022**, *110*. <https://doi.org/10.1016/j.engappai.2022.104743>.
756. *Implementing Technology Change*, 2021.
757. of Standards, N.I.; Technology. *Post-Quantum Cryptography*. Accessed: 2025-09-11.

758. Dahmen-Lhuissier, S. [Quantum-Safe Cryptography \(QSC\)](#).
759. Gross, G. [IETF launches post-quantum encryption working group](#), 2023. publisher: IETF.
760. Swayne, M. [Quantum Economics: Could a New Economic Paradigm Be Guided by Quantum Models?](#), 2024. Last accessed: 01-07-2025.
761. Orrell, D.; Houshmand, M. [Quantum Propensity in Economics](#). *Frontiers in Artificial Intelligence* **2022**, *4*. <https://doi.org/10.3389/frai.2021.772294>.
762. Trisetarso, A. [Quantum Computational Economics](#). *Procedia Computer Science* **2023**, *216*, 3. 7th International Conference on Computer Science and Computational Intelligence 2022, <https://doi.org/10.1016/j.procs.2022.12.103>.
763. Holtfort, T.; Horsch, A. [Quantum Economics: A Systematic Literature Review](#). *SocioEconomic Challenges* **2024**, pp. 62–77. [https://doi.org/10.61093/sec.8\(1\).62-77.2024](https://doi.org/10.61093/sec.8(1).62-77.2024).
764. Varian, H.R. *Computational Economics and Finance Modeling And Analysis With Mathematica*; Springer, 2011; p. 482.
765. Samuelson, L. [Game Theory in Economics and Beyond](#). *Journal of Economic Perspectives* **2016**, *30*, 107–130. <https://doi.org/10.1257/jep.30.4.107>.
766. Cristelli, M.; Pietronero, L.; Zaccaria, A. [Critical Overview of Agent-Based Models for Economics](#). *arXiv* **2011**, [arXiv:q-fin.TR/1101.1847]. <https://doi.org/10.48550/ARXIV.1101.1847>.
767. Axtell, R.L.; Farmer, J.D. [Agent-Based Modeling in Economics and Finance: Past, Present, and Future](#). *Journal of Economic Literature* **2025**, *63*, 197–287. <https://doi.org/10.1257/jel.20221319>.
768. Seuring, S. [A review of modeling approaches for sustainable supply chain management](#). *Decision Support Systems* **2013**, *54*, 1513–1520. <https://doi.org/10.1016/j.dss.2012.05.053>.
769. Aaronson, S. [Quantum copy-protection and quantum money](#). *Proceedings of the 24th Annual IEEE Conference on Computational Complexity* **2009**, pp. 229–242.
770. Lutomirski, A.; Aaronson, S.; Farhi, E.; Gosset, D.; Hassidim, A.; Kelner, J.; Shor, P. [Breaking and making quantum money: toward a new quantum cryptographic protocol](#). *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, 20-31, 978-7-302-21752-7 Tsinghua University Press **2009**, [arXiv:quant-ph/0912.3825]. <https://doi.org/10.48550/ARXIV.0912.3825>.
771. Aaronson, S.; Christiano, P. [Quantum money from hidden subspaces](#). In Proceedings of the Proceedings of the forty-fourth annual ACM symposium on Theory of computing. ACM, 2012, STOC'12, pp. 41–60. <https://doi.org/10.1145/2213977.2213983>.
772. Farhi, E.; Gosset, D.; Hassidim, A.; Lutomirski, A.; Shor, P. [Quantum money from knots](#). In Proceedings of the Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ACM, 2012, ITCS '12, pp. 276–289. <https://doi.org/10.1145/2090236.2090260>.
773. Zhandry, M. [Quantum Lightning Never Strikes the Same State Twice. Or: Quantum Money from Cryptographic Assumptions](#). *Journal of Cryptology* **2021**, *34*. <https://doi.org/10.1007/s00145-020-09372-x>.
774. Zhandry, M., [Quantum Lightning Never Strikes the Same State Twice](#). In *Advances in Cryptology – EURO-CRYPT 2019*; Springer International Publishing, 2019; pp. 408–438. [https://doi.org/10.1007/978-3-030-17659-4\\_14](https://doi.org/10.1007/978-3-030-17659-4_14).
775. Gavinsky, D. [Quantum Money with Classical Verification](#). In Proceedings of the 2012 IEEE 27th Conference on Computational Complexity. IEEE, 2012, pp. 42–52. <https://doi.org/10.1109/cc.2012.10>.
776. Mosca, M.; Stebila, D. [Quantum Coins](#). *Error-Correcting Codes, Finite Geometries and Cryptography. Contemporary Mathematics, volume 523, pages 35-47. American Mathematical Society, 2010* **2009**, [arXiv:quant-ph/0911.1295]. <https://doi.org/10.48550/ARXIV.0911.1295>.
777. Tokunaga, Y.; Okamoto, T.; Imoto, N. [Anonymous quantum cash](#). In Proceedings of the ERATO Conference on Quantum Information Science, 2003.
778. Ji, Z.; Liu, Y.K.; Song, F., [Pseudorandom Quantum States](#). In *Advances in Cryptology – CRYPTO 2018*; Springer International Publishing, 2018; pp. 126–152. [https://doi.org/10.1007/978-3-319-96878-0\\_5](https://doi.org/10.1007/978-3-319-96878-0_5).
779. Behera, A.; Sattath, O. [Almost Public Quantum Coins](#). *arXiv preprint* **2020**, [arXiv:quant-ph/2002.12438]. <https://doi.org/10.48550/ARXIV.2002.12438>.
780. Radian, R.; Sattath, O. [Semi-Quantum Money](#). *Cryptology ePrint Archive, Paper 2020/414*, 2020. <https://doi.org/10.1145/3318041.3355462>.
781. Amos, R.; Georgiou, M.; Kiayias, A.; Zhandry, M. [One-shot signatures and applications to hybrid quantum/classical authentication](#). In Proceedings of the Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. ACM, 2020, STOC '20, pp. 255–268. <https://doi.org/10.1145/3357713.3384304>.

782. Tapscott, D.; Tapscott, A. *The Blockchain Revolution*; Penguin USA, Inc. in New York, 2016.
783. Coladangelo, A.; Sattath, O. A Quantum Money Solution to the Blockchain Scalability Problem. *Quantum* **2020**, *4*, 297. <https://doi.org/10.22331/q-2020-07-16-297>.
784. Edwards, M.; Mashatan, A.; Ghose, S. A review of quantum and hybrid quantum/classical blockchain protocols. *Quantum Information Processing* **2020**, *19*. <https://doi.org/10.1007/s1128-020-02672-y>.
785. Bilyk, A.; Doliskani, J.; Gong, Z. Cryptanalysis of three quantum money schemes. *Quantum Information Processing* **2023**, *22*. <https://doi.org/10.1007/s1128-023-03919-0>.
786. Bozzio, M.; Orioux, A.; Trigo Vidarte, L.; Zaquine, I.; Kerenidis, I.; Diamanti, E. Experimental investigation of practical unforgeable quantum money. *npj Quantum Information* **2018**, *4*. <https://doi.org/10.1038/s41534-018-0058-2>.
787. Bozzio, M.; Diamanti, E.; Grosshans, F. Semi-device-independent quantum money with coherent states. *Physical Review A* **2019**, *99*, 022336. <https://doi.org/10.1103/physreva.99.022336>.
788. Kent, A.; Lowndes, D.; Pitalúa-García, D.; Rarity, J. Practical quantum tokens without quantum memories and experimental tests. *npj Quantum Information* **2022**, *8*. <https://doi.org/10.1038/s41534-022-00524-4>.
789. Liberto, D. *Economic Forecasting: Definition, Use of Indicators, and Example*, 2024. Accessed: 2025-08-29.
790. Hamilton, J.D. *Time series analysis*; Princeton University Press: Princeton, N.J, 1994. Includes bibliographical references and indexes.
791. Kirchgässner, G.; Wolters, J.; Hassler, U. *Introduction to Modern Time Series Analysis*, second edition ed.; SpringerLink, Springer: Berlin, Heidelberg, 2013.
792. Gohel, P.; Joshi, M. Quantum time series forecasting. In Proceedings of the Sixteenth International Conference on Machine Vision (ICMV 2023); Osten, W., Ed. SPIE, 2024, p. 39. <https://doi.org/10.1117/12.3023467>.
793. Granger, C.W.J.; Newbold, P.; Shell, K. *Forecasting Economic Time Series*, 2. Aufl. ed.; Economic theory, econometrics, and mathematical economics, Elsevier Reference Monographs: [s.l.], 2014. Description based upon print version of record.
794. Ryan Tibshirani. *ARIMA Models Lecture Notes*, 2023. Accessed: 2025-08-29.
795. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks. *Future Internet* **2023**, *15*, 255. <https://doi.org/10.3390/fi15080255>.
796. Francq, C.; Zakoian, J.M. *GARCH models*, second edition ed.; ProQuest Ebook Central, Wiley: Hoboken, NJ, 2019. Aus dem Französischen übersetzt.
797. Khodarahmi, M.; Maihami, V. A Review on Kalman Filter Models. *Archives of Computational Methods in Engineering* **2022**, *30*, 727–747. <https://doi.org/10.1007/s11831-022-09815-7>.
798. Daskin, A. A walk through of time series analysis on quantum computers. *arXiv* **2022**, [arXiv:quant-ph/2205.00986]. <https://doi.org/10.48550/ARXIV.2205.00986>.
799. Emmanoulopoulos, D.; Dimoska, S. Quantum Machine Learning in Finance: Time Series Forecasting. *arXiv preprint* **2022**, [arXiv:quant-ph/2202.00599]. <https://doi.org/10.48550/ARXIV.2202.00599>.
800. Hirth, A.; Droguett, E.L. State of Quantum RNNs for Time-Series Forecasting. In Proceedings of the 2025 Annual Reliability and Maintainability Symposium (RAMS). IEEE, 2025, pp. 1–6. <https://doi.org/10.1109/rams48127.2025.10935154>.
801. Jones, C.; Kraus, N.; Bhardwaj, P.; Adler, M.; Schrödl-Baumann, M.; Manrique, D.Z. Benchmarking Quantum Models for Time-Series Forecasting. In Proceedings of the 2024 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2024, pp. 22–27. <https://doi.org/10.1109/qce60285.2024.10246>.
802. Lu, Y.; Shen, M.; Wang, H.; Wang, X.; van Rechem, C.; Fu, T.; Wei, W. Machine Learning for Synthetic Data Generation: A Review, 2025, [arXiv:cs.LG/2302.04062].
803. Assefa, S.A.; Dervovic, D.; Mahfouz, M.; Tillman, R.E.; Reddy, P.; Veloso, M. Generating synthetic data in finance: opportunities, challenges and pitfalls. In Proceedings of the Proceedings of the First ACM International Conference on AI in Finance, New York, NY, USA, 2021; ICAIF '20. <https://doi.org/10.1145/3383455.3422554>.
804. Ramzan, F.; Sartori, C.; Consoli, S.; Reforgiato Recupero, D. Generative Adversarial Networks for Synthetic Data Generation in Finance: Evaluating Statistical Similarities and Quality Assessment. *AI* **2024**, *5*, 667–685. <https://doi.org/10.3390/ai5020035>.
805. Potluru, V.K.; Borrajo, D.; Coletta, A.; Dalmaso, N.; El-Laham, Y.; Fons, E.; Ghassemi, M.; Gopalakrishnan, S.; Gosai, V.; Kreačić, E.; et al. Synthetic Data Applications in Finance, 2024, [arXiv:cs.LG/2401.00081].
806. Figueira, A.; Vaz, B. Survey on Synthetic Data Generation, Evaluation Methods and GANs. *Mathematics* **2022**, *10*. <https://doi.org/10.3390/math10152733>.

807. Chen, J.; Hull, J.C.; Poulos, Z.; Rasul, H.; Veneris, A.G.; Wu, Y. *A Variational Autoencoder Approach to Conditional Generation of Possible Future Volatility Surfaces*. *SSRN Electronic Journal* **2023**.
808. Acciaio, B.; Eckstein, S.; Hou, S. *Time-Causal VAE: Robust Financial Time Series Generator*, 2024, [arXiv:cs.LG/2411.02947].
809. Cont, R.; Cucuringu, M.; Xu, R.; Zhang, C. *Tail-GAN: Learning to Simulate Tail Risk Scenarios*, 2025, [arXiv:q-fin.RM/2203.01664].
810. Chen, L. *Risk Management with Feature-Enriched Generative Adversarial Networks (FE-GAN)*, 2024, [arXiv:q-fin.RM/2411.15519].
811. Na, A.; Zhang, M.; Wan, J. *Computing Volatility Surfaces using Generative Adversarial Networks with Minimal Arbitrage Violations*, 2023, [arXiv:q-fin.CP/2304.13128].
812. Marti, G. *CORRGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks*. In Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 8459–8463. <https://doi.org/10.1109/ICASSP40776.2020.9053276>.
813. Milena Vuletić, F.P.; Cucuringu, M. *Fin-GAN: forecasting and classifying financial time series via generative adversarial networks*. *Quantitative Finance* **2024**, *24*, 175–199, [https://doi.org/10.1080/14697688.2023.2299466]. <https://doi.org/10.1080/14697688.2023.2299466>.
814. Guo, X.; Chen, Y. *Generative AI for Synthetic Data Generation: Methods, Challenges and the Future*, 2024, [arXiv:cs.LG/2403.04190].
815. Takahashi, S.; Chen, Y.; Tanaka-Ishii, K. *Modeling financial time-series with generative adversarial networks*. *Physica A: Statistical Mechanics and its Applications* **2019**, *527*, 121261. <https://doi.org/10.1016/j.physa.2019.121261>.
816. Caprioli, S.; Cagliero, E.; Crupi, R. *Quantifying Credit Portfolio sensitivity to asset correlations with interpretable generative neural networks*, 2023, [arXiv:q-fin.RM/2309.08652].
817. Vieloszynski, A.; Cherkaoui, S.; Ahmad, O.; Laprade, J.F.; Nahman-Lévesque, O.; Aaraba, A.; Wang, S. *LatentQGAN: A Hybrid QGAN with Classical Convolutional Autoencoder*, 2024, [arXiv:quant-ph/2409.14622].
818. Ganguly, S. *Implementing Quantum Generative Adversarial Network (qGAN) and QCBM in Finance*, 2023, [arXiv:quant-ph/2308.08448].
819. Coopmans, L.; Benedetti, M. *On the sample complexity of quantum Boltzmann machine learning*. *Communications Physics* **2024**, *7*. <https://doi.org/10.1038/s42005-024-01763-x>.
820. Salloum, H.; Salloum, A.; Mazzara, M.; Zzykov, S. *Quantum Annealing in Machine Learning: QBoost on D-Wave Quantum Annealer*. *Procedia Computer Science* **2024**, *246*, 3285–3293. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024), <https://doi.org/10.1016/j.procs.2024.09.311>.
821. Maheshwari, D.; Sierra-Sosa, D.; Garcia-Zapirain, B. *Variational Quantum Classifier for Binary Classification: Real vs Synthetic Dataset*. *IEEE Access* **2022**, *10*, 3705–3715. <https://doi.org/10.1109/ACCESS.2021.3139323>.
822. Pires, O.M.; Nooblath, M.Q.; Silva, Y.A.C.; da Silva, M.H.F.; Galvão, L.Q.; Albino, A.S. *Synthetic data generation with hybrid quantum-classical models for the financial sector*. *The European Physical Journal B* **2024**, *97*, 178.
823. Guryanova, L.S.; Gvozdytskyi, V.S.; Dymchenko, O.V.; Rudachenko, O.A. *Models of Forecasting in the Mechanism of Early Informing and Prevention of Financial Crises in Corporate Systems*. *Financial and credit activity problems of theory and practice* **2018**, *3*, 303–312. <https://doi.org/10.18371/fcaptop.v3i26.144280>.
824. Greenwood, R.; Hanson, S.G.; Shleifer, A.; Sørensen, J.A. *Predictable Financial Crises*, 2021.
825. Elliott, M.; Golub, B.; Jackson, M.O. *Financial Networks and Contagion*. *American Economic Review* **2014**, *104*, 3115–3153. <https://doi.org/10.1257/aer.104.10.3115>.
826. Ding, Y.; Gonzalez-Conde, J.; Lamata, L.; Martín-Guerrero, J.D.; Lizaso, E.; Mugel, S.; Chen, X.; Orús, R.; Solano, E.; Sanz, M. *Toward Prediction of Financial Crashes with a D-Wave Quantum Annealer*. *Entropy* **2023**, *25*, 323. <https://doi.org/10.3390/e25020323>.
827. Orús, R.; Mugel, S.; Lizaso, E. *Forecasting financial crashes with quantum computing*. *Phys. Rev. A* **2019**, *99*, 060301. <https://doi.org/10.1103/PhysRevA.99.060301>.
828. Fellner, M.; Ender, K.; ter Hoeven, R.; Lechner, W. *Parity Quantum Optimization: Benchmarks*, 2021. <https://doi.org/10.48550/ARXIV.2105.06240>.
829. National Quantum Computing Centre (NQCC). *Quantum Computing Scalability Conference 2025*, 2025. Accessed: 2025-05-07.
830. Aharonov, D.; Ben-Or, M. *Fault-Tolerant Quantum Computation With Constant Error Rate*, 1999. <https://doi.org/10.48550/ARXIV.QUANT-PH/9906129>.

831. Knill, E.; Laflamme, R.; Zurek, W.H. Resilient Quantum Computation. *Science* **1998**, *279*, 342–345, [<https://www.science.org/doi/pdf/10.1126/science.279.5349.342>]. <https://doi.org/10.1126/science.279.5349.342>.
832. Kitaev, A. Fault-tolerant quantum computation by anyons. *Annals of Physics* **2003**, *303*, 2–30. [https://doi.org/10.1016/S0003-4916\(02\)00018-0](https://doi.org/10.1016/S0003-4916(02)00018-0).
833. von Neumann, J., Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components. In *Automata Studies. (AM-34), Volume 34*; Princeton University Press, 2016; pp. 43–98. <https://doi.org/doi:10.1515/9781400882618-003>.
834. Ryan-Anderson, C.; Bohnet, J.; Lee, K.; Gresh, D.; Hankin, A.; Gaebler, J.; Francois, D.; Chernoguzov, A.; Lucchetti, D.; Brown, N.; et al. Realization of Real-Time Fault-Tolerant Quantum Error Correction. *Physical Review X* **2021**, *11*, 041058. <https://doi.org/10.1103/physrevx.11.041058>.
835. Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* **2023**, *614*, 676–681. <https://doi.org/10.1038/s41586-022-05434-1>.
836. Bluvstein, D.; Evered, S.J.; Geim, A.A.; Li, S.H.; Zhou, H.; Manovitz, T.; Ebadi, S.; Cain, M.; Kalinowski, M.; Hangleiter, D.; et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **2023**, *626*, 58–65. <https://doi.org/10.1038/s41586-023-06927-3>.
837. Liao, H.; Hartnett, G.S.; Kakkar, A.; Tan, A.; Hush, M.; Mundada, P.S.; Biercuk, M.J.; Baum, Y. Achieving computational gains with quantum error correction primitives: Generation of long-range entanglement enhanced by error detection. *arXiv* **2024**, [arXiv:quant-ph/2411.14638]. <https://doi.org/10.48550/ARXIV.2411.14638>.
838. Bravyi, S.; Cross, A.W.; Gambetta, J.M.; Maslov, D.; Rall, P.; Yoder, T.J. High-threshold and low-overhead fault-tolerant quantum memory. *Nature* **2024**, *627*, 778–782. <https://doi.org/10.1038/s41586-024-07107-7>.
839. Department for Science, Innovation and Technology. National Quantum Strategy Missions, 2023. Accessed: 2025-05-07.
840. Couteau, C. Quantum computing using photons. *The European Physical Journal A* **2025**, *61*. <https://doi.org/10.1140/epja/s10050-025-01517-5>.
841. Kaggle. Datasets, n.d. Retrieved May 7, 2025.
842. Davis, T.A.; Hu, Y. The university of Florida sparse matrix collection. *ACM Transactions on Mathematical Software* **2011**, *38*, 1–25. <https://doi.org/10.1145/2049662.2049663>.
843. Kolodziej, S.; Aznavah, M.; Bullock, M.; David, J.; Davis, T.; Henderson, M.; Hu, Y.; Sandstrom, R. The SuiteSparse Matrix Collection Website Interface. *Journal of Open Source Software* **2019**, *4*, 1244. <https://doi.org/10.21105/joss.01244>.
844. GQI, G.Q.I. Midstack Focus Report with Classiq, 2023.
845. GQI, G.Q.I. Midstack Focus Report with QuantroLOX, 2023.
846. GQI, G.Q.I. Midstack Focus Report with QEDMA, 2023.
847. Faro, I.; Sitdikov, I.; Valiñas, D.G.; Fernandez, F.J.M.; Codella, C.; Glick, J. Middleware for Quantum: An orchestration of hybrid quantum-classical systems. In Proceedings of the 2023 IEEE International Conference on Quantum Software (QSW). IEEE, 2023. <https://doi.org/10.1109/qsw59989.2023.00011>.
848. Ostrander, A.J. Quantum Algorithms for Differential Equations. phdthesis, University of Maryland, College Park, United States – Maryland, 2019.
849. Jordan, S.P.; Shutty, N.; Wootters, M.; Zalcman, A.; Schmidhuber, A.; King, R.; Isakov, S.V.; Khattar, T.; Babbush, R. Optimization by Decoded Quantum Interferometry, 2025, [arXiv:quant-ph/2408.08292].
850. Zeitgeist, Q. Exploring QRAM: The Daring Quest For Quantum Memory Persistence, 2024.
851. Physics World. On the path towards a quantum economy, 2025. Accessed: 2025-05-08.
852. Ciceri, A.; Cottrell, A.; Freeland, J.; Fry, D.; Hirai, H.; Intallura, P.; Kang, H.; Lee, C.K.; Mitra, A.; Ohno, K.; et al. Enhanced fill probability estimates in institutional algorithmic bond trading using statistical learning algorithms with quantum computers. *arXiv* **2025**, [arXiv:quant-ph/2509.17715]. <https://doi.org/10.48550/ARXIV.2509.17715>.
853. Agliardi, G.; Alevras, D.; Kumar, V.; Nardo, R.L.; Compostella, G.; Kumar, S.; Proissl, M.; Mehta, B. Portfolio construction using a sampling-based variational quantum scheme. *arXiv* **2025**, [arXiv:quant-ph/2508.13557]. <https://doi.org/10.48550/ARXIV.2508.13557>.
854. Markham, C.; Grassie, R. Quantum computing applications in financial services. Technical report, Financial Conduct Authority, 2025.

855. Hughes, A.C.; Srinivas, R.; Löschnauer, C.M.; Knaack, H.M.; Matt, R.; Ballance, C.J.; Malinowski, M.; Harty, T.P.; Sutherland, R.T. Trapped-ion two-qubit gates with >99.99% fidelity without ground-state cooling, 2025, [arXiv:quant-ph/2510.17286].
856. Abanin, D.A.; et al.. Observation of constructive interference at the edge of quantum ergodicity. *Nature* **2025**, *646*, 825–830. <https://doi.org/10.1038/s41586-025-09526-6>.
857. Foundation, T.N. Nobel Prize in Physics 2025 - Press Release, 2025. Last accessed: 27-10-2025.
858. Eisert, J.; Preskill, J. Mind the gaps: The fraught road to quantum advantage. *arXiv* **2025**, [arXiv:quant-ph/2510.19928]. <https://doi.org/10.48550/ARXIV.2510.19928>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.