

# Quantum computing for finance

Dylan Herman<sup>1</sup>✉, Cody Googin<sup>2</sup>, Xiaoyuan Liu<sup>3</sup>, Yue Sun<sup>1</sup>, Alexey Galda<sup>4</sup>, Ilya Safro<sup>5</sup>, Marco Pistoia<sup>1</sup> & Yuri Alexeev<sup>6</sup>

## Abstract

Quantum computers are expected to surpass the computational capabilities of classical computers and have a transformative impact on numerous industry sectors. We present a comprehensive summary of the state of the art of quantum computing for financial applications, with particular emphasis on stochastic modelling, optimization and machine learning. This Review is aimed at physicists, so it outlines the classical techniques used by the financial industry and discusses the potential advantages and limitations of quantum techniques. Finally, we look at the challenges that physicists could help tackle.

## Sections

Introduction

Stochastic modelling

Optimization

Machine learning

Outlook

<sup>1</sup>JPMorgan Chase, New York, NY, USA. <sup>2</sup>University of Chicago, Chicago, IL, USA. <sup>3</sup>Fujitsu Research of America, Inc., Sunnyvale, CA, USA. <sup>4</sup>Menten AI, San Francisco, CA, USA. <sup>5</sup>University of Delaware, Newark, DE, USA. <sup>6</sup>Argonne National Laboratory, Lemont, IL, USA. ✉e-mail: [dylan.a.herman@jpmorgan.com](mailto:dylan.a.herman@jpmorgan.com)

## Key points

- Quantum algorithms for stochastic modelling, optimization and machine learning are applicable to various financial problems.
- Quantum Monte Carlo integration and gradient estimation can provide quadratic speedup over classical methods, but more work is required to reduce the amount of quantum resources for early fault-tolerant feasibility and achieving an actual speedup.
- Financial optimization problems can be continuous (convex or non-convex), discrete or mixed, and thus quantum algorithms for these problems can be applied.
- The advantages and challenges of quantum machine learning for classical problems are also apparent in finance.

## Introduction

Financial institutions tackle a wide array of computationally challenging problems on a daily basis. These problems include forecasting (ranging from pricing and risk estimation to identifying anomalous transactions and customer preferences) and optimization (such as portfolio selection, finding optimal trading strategies and hedging). Owing to the advances in mathematical finance and computational techniques, with contributions from the financial industry and the scientific community, financial institutions have adopted a diverse set of problem-solving tools using stochastic modelling<sup>1</sup>, optimization algorithms and machine learning (ML) models for attacking both categories. In recent years, there has been increasing interest from both the academia and the industry in exploring whether quantum computing could be used to solve classically challenging problems. Researchers and industry practitioners have developed various quantum computing algorithms for each of the aforementioned problem-solving tools<sup>2</sup>. For example, the quantum algorithm for Monte Carlo integration (MCI) has shown promise for a quadratic speedup in convergence compared with the classical counterpart; this can greatly improve risk modelling, in which MCI<sup>3</sup> is an indispensable tool. In optimization and ML, recent studies have demonstrated the advantages of quantum algorithms for discrete optimization, dynamic programming, boosting and clustering, showing polynomially better complexity dependence on certain dimensions of the size of the problem than the state-of-the-art classical algorithms. In addition, various heuristic approaches (which often use a classical–quantum hybrid methodology involving variational quantum circuits) have been developed and heavily investigated. Preliminary results have shown indications of improvements provided by these heuristics over classical approaches. As most of these algorithms have general applicability to a broad category of problems, they also have similar potential in financial applications.

Notwithstanding the notable advances in quantum algorithms, there are still many challenges in achieving end-to-end quantum advantages on problems with commercially relevant specifications. In particular, a large amount of work needs to be carried out to reduce the resource requirements in certain components of the algorithms, such as quantum embedding of classical input data, readout of the quantum output and pre-processing and post-processing. Otherwise, the complexity of these components may significantly dispare the speedup obtained from other components of the algorithm. Moreover,

current quantum hardware is still in a nascent stage; existing noisy intermediate-scale quantum devices have low fidelity and a small number of qubits, which severely limits the size of problems that can be solved. As a result, executing quantum circuits on hardware often requires a significant optimization effort at both the circuit-design and compilation levels and relies on classical computing to find optimal initial conditions and for error mitigation. These challenges hint at research opportunities in addressing the bottlenecks of end-to-end quantum applications and the design of hardware-aware quantum algorithms. In addition, more opportunities may be discovered in exploiting the rich structures in financial problems when searching for a quantum speedup, introducing new heuristics for computationally hard problems and improving classical heuristic algorithms.

In this Review, we survey the state of the art of quantum algorithms for financial applications. We discuss important financial problems that are solved using techniques from stochastic modelling, optimization and ML. For each of these areas, we review the classical techniques used and discuss whether the associated quantum algorithms could be useful. We also discuss the associated implementation challenges and identify potential research directions. Table 1 is a high-level overview of the quantum algorithms covered in this Review. We note that similar tables summarizing quantum algorithms for finance from different perspectives are also present in previous reviews<sup>4,5</sup>.

There is an existing line of reviews of quantum computing for finance. For example, ref. 4 focuses on covering the hardware and algorithmic work done by IBM. The article in ref. 6 focuses on works done by the QC Ware team. The survey in ref. 5 highlights financial applications that make use of quantum annealers, while also providing a short overview of certain aspects in quantum ML and quantum MCI (QMCI) for finance. Reference<sup>7</sup> covers various quantum ML algorithms applicable to finance. There are also several problem-specific surveys such as the derivative pricing<sup>8</sup>, supply chain finance<sup>9</sup> and high-frequency trading<sup>10</sup>. In contrast to these works, this Review takes a more holistic view and discusses a wider variety of financial applications and more recent quantum algorithms. There has been significant progress in using quantum-inspired approaches for finance, and although a review of this line of work is beyond the scope of the current article, we do briefly mention quantum-inspired algorithms in the context of dequantized quantum algorithms in the ‘Machine learning’ section. We also direct the reader to existing surveys on other quantum-inspired methods such as tensor networks for ML<sup>11,12</sup>, which have been applied to finance<sup>13</sup>.

We assume some familiarity with quantum computing and applied mathematics; otherwise, we direct the reader to the standard introductory texts in quantum computing<sup>14,15</sup> and mathematical finance<sup>16,17</sup>.

## Stochastic modelling

Stochastic processes are commonly used to model phenomena in physical sciences, biology, epidemiology and finance. In the latter, stochastic modelling is often used to help make investment decisions, usually with the goal of maximizing returns and minimizing risks (see ref. 18 for an introduction). Quantities that are descriptive of the market condition, including stock prices, interest rates and their volatilities, are often modelled by stochastic processes and represented by random variables. The evolution of such stochastic processes is governed by stochastic differential equations (SDEs), and stochastic modelling aims to solve SDEs for the expectation value of a certain random variable of interest, such as the expected payoff of a financial derivative at a future time, which determines the fair value of the derivative.

**Table 1 | Quantum algorithms for financial problems**

Methodology	Applications	Algorithms	Challenges	Advantages	Applicability
Stochastic modelling	Pricing Risk modelling	Quantum Monte Carlo integration	Model loading	Quadratic speedup	Long term
		Quantum PDE solvers	Pre-processing and post-processing	Unclear	Long term
Optimization	Portfolio management Financing Resource allocation	Quantum-enhanced SCP solvers	QBLAS limitations	Unclear	Long term
		Quantum-enhanced MIP solvers	Oracle complexity	Quadratic speedup	Long term
		Quantum heuristics	Parameter tuning	Partial evidence	Near term
Machine learning	Forecasting Anomaly detection Recommendations	Quantum-enhanced ML	Data loading	Unclear	Long term
		Quantum-native ML	Parameter tuning and/or sampling complexity	Unclear	Near term

MIP, mixed integer programming; ML, machine learning; PDE, partial differential equation; QBLAS, quantum basic linear algebra subroutine; SCP, symmetric cone program.

Although analytical solutions for SDEs are available for a few simple cases, such as the geometric Brownian motion used in the Black–Scholes model for a European option<sup>19</sup>, the vast majority of financial models involves SDEs of more complex forms that can only be solved with numerical approaches. Moreover, even in cases in which the SDE can be analytically solved, the complexity of the payoff specifications of the derivative may make it difficult to find closed-form exact solutions for moments of the payoff beyond the simplest cases of European options and certain Asian options<sup>20</sup>. Such more complex payoff types include American options and barrier options. If the desired moments of the stochastic process can be expressed as a manageable differential equation, one numerical approach is to use high-precision non-randomized methods for approximately solving differential equations. The high precision comes at the cost of a strong dependence on the dimension of the stochastic process. Alternatively, one may use MCI to trade precision for a much weaker dependence on the dimension through the variance of the stochastic process. In fact, the integrals computed for financial problems typically involve stochastic processes whose variances do not grow exponentially with dimension as it can in the more general case. The duality between directly solving the partial differential equation (PDE) governing the evolution of moments and estimating it through sampling has been exemplified in the Feynman–Kac formula<sup>21,22</sup>. Both approaches are widely applicable even outside stochastic modelling and have sparked off developments in corresponding quantum methods. In the following sections, we overview financial applications that could benefit from quantum versions of numerical methods for estimating moments and other deterministic functions of stochastic processes.

## Quantum methods for Monte Carlo-based pricing and risk analysis

An important type of asset-pricing problem in finance is the pricing of derivatives. A derivative is a contract that derives its value from another source (such as a collection of assets or financial benchmarks) called the ‘underlying’, whose value is modelled by a stochastic process. The value of a derivative is typically calculated by simulating the dynamics of the underlying and computing the payoff accordingly. At each time step, the payoff function computes the cash flow to the owner of the contract, given the current and potentially historical states of the underlying assets. The payoff is a simple function that is determined

when the derivative contract is made. The fair-value price of the derivative is given by the expected future cash flows, or payoffs, discounted to the current date.

Classically, one could simulate the payoff stochastic process using a model and estimate the price of the derivative with sample means. This is the MCI approach<sup>3</sup>, which makes no assumptions about the underlying model and can handle high-dimensional financial problems. In accordance with Chebyshev’s inequality, the number of required samples from the model to achieve an estimation-error  $\varepsilon$  scales as  $O(\sigma^2/\varepsilon^2)$ , where  $\sigma^2$  is the variance of the payoff process. However, there exists a quantum algorithm, commonly known as QMCI, that can produce an  $\varepsilon$ -estimate of the price using  $O(\sigma/\varepsilon)$  quantum samples with a constant success probability<sup>23–27</sup>. Specifically, suppose with probability  $p(\omega)$ , the state of the underlying follows a trajectory  $\omega \in \Omega$  from an initial point in time to the maturity of the contract, and additionally suppose that  $f(\omega)$  is the payoff function scaled to be in  $[0, 1]$ , then the following operation gives a single quantum sample:

$$Q|0\rangle = \sum_{\omega \in \Omega} \sqrt{f(\omega)p(\omega)} |\omega\rangle |1\rangle + |\perp\rangle, \quad (1)$$

which is produced by a unitary operator  $Q$ , and  $|\perp\rangle$  is an unnormalized orthogonal garbage state produced owing to unitarity. The vanilla QMCI uses quantum amplitude estimation (QAE) to estimate the probability of observing the rightmost qubit in the  $|1\rangle$  state. A necessary condition for a practical quantum advantage is that  $Q$  is implementable in a time comparable to the time to generate a classical sample from  $p(\omega)$ . In addition, it has been noted that the overhead from current quantum error-correcting codes could prevent the quadratic speedup from being realizable<sup>28</sup>. The good news is that the payoff functions in most derivatives have simple forms such as piece-wise linear functions and therefore can be implemented using reversible-arithmetic techniques<sup>29</sup>. Furthermore, there are variants of QAE with reduced circuit depth<sup>30–32</sup>.

To implement  $Q$ , various techniques have been developed. Specifically, the Grover–Rudolph algorithm<sup>33</sup> and its approximate variant<sup>34</sup> aim to address loading distributions that are efficiently numerically integrable (for example, log-concave distributions). However, it has been shown that, when numerical methods, such as classical MCI, which we are trying to avoid, are used to integrate the distribution,

QMCI does not provide a quadratic speedup when using this state preparation technique<sup>35</sup>. For loading general  $L_1$  or  $L_2$  normalized functions, a number of black-box state preparation techniques have been proposed<sup>36–40</sup>. These algorithms start by preparing the target state with a success probability and then use amplitude amplification to boost that probability. Despite the general applicability of these algorithms, their inverse dependence on the filling ratio (the relevant norm of the function divided by the corresponding norm of the box bounding of that function) of the function being loaded makes these algorithms much less efficient when the target distribution is concentrated around a few values. An approximate method that loads degree- $d$  polynomial approximations of functions has been proposed in ref. 41, which has a similar dependence on the filling ratio. Alternatively, one could use data-driven methods, such as variational quantum algorithms<sup>42</sup>, which can be trained to load the path distribution<sup>43,44</sup>. Non-unitary distribution loading approaches have also been proposed<sup>45,46</sup>, although it is still an open question how these methods can be integrated into QMCI. For loading tabulated data, one has to resort to generic data loading techniques<sup>47–49</sup>, which in general scale linearly with the number of data points to be loaded.

Furthermore, rather than loading the full payoff distribution  $p(\omega)$  directly, it may be easier to load the joint distribution over path increments and use arithmetic to introduce any necessary correlations<sup>50</sup>. Nevertheless, this approach requires a number of qubits that scale linearly with the number of time steps, and quantum arithmetic operations can be expensive to perform. Therefore, it remains an open question whether there are alternative unitary procedures for loading the payoff distribution that avoid significant arithmetic and have sublinear scaling in the number of qubits. For scenarios in which the SDE can only be simulated approximately, that is, when using local-volatility models<sup>51</sup>, the discretization in time may introduce an additional multiplicative factor of  $O(1/\epsilon)$  to the overall complexity of the MCI algorithm<sup>52</sup>. To address this issue, the multilevel MCI<sup>53</sup> was proposed, which ensures that the overall algorithm still scales as single-time-step MCI up to a polylogarithmic factor under certain assumptions on the payoff function. This approach has been quantized in ref. 54 with similar guarantees and extended to other payoff functions.

There is a deterministic classical method known as quasi-MCI, which for low-dimensional problems obtains a similar error scaling as QMCI. However, this comes at the cost of an exponential dependence on the dimension. Interestingly, for reasons not fully understood, this exponential dependence does not appear in practice for some financial problems (in section 5.5 of ref. 3). Thus, a similar classical quadratic speedup can be obtained in some settings.

Because of the provable speedup in query complexity that QMCI provides for the widely used MCI method, it is not surprising that the community has started to perform application-specific resource analysis for the pricing of various types of derivatives, such as European<sup>55</sup> and Asian<sup>56</sup> options, autocallable and target accrual redemption forward<sup>50</sup>.

For some derivatives, evaluating the expected payoff may be affected by future decisions, which can significantly complicate the pricing process. An important example of such derivatives is an American option. An American call option gives the holder the right to buy the underlying asset at a fixed price  $K$  (strike) at any time between today ( $t = 0$ ) and the maturity date ( $t = T$ ). At each time point  $t \in [0, T]$ , the holder must decide whether to exercise the option by comparing the payoff  $\max(S_t - K, 0)$  if exercised today, where  $S_t$  is the price of the underlying asset at  $t$ , with the expected payoff if exercised later. American option pricing falls under the broader class of optimal stopping

problems. The exact solution of the value of an American option requires dynamic programming, because determining whether to exercise at  $t$  requires solving the subproblem of determining the future expected payoff from exercising after  $t$ , called a continuation value. For a more detailed discussion on dynamic programming, we refer the readers to the next section. In finance, American option pricing is typically solved approximately by using regression to predict continuation values and computing the stopping time by backtracking from the maturity point. This technique is called least-squares Monte Carlo<sup>57</sup>. The labels used for training the regression model are typically computed using MCI, and there are versions that use QMCI<sup>58</sup>.

Another important task often accompanying derivative pricing is the computation of sensitivities of the derivative price to model and market parameters, which is equivalent to computing gradients of the price with respect to these input parameters. Such gradients are often known as the ‘Greeks’. Greeks allow for systematically hedging against risks associated with holding a derivative contract under market movements and hence are a vital tool in risk management. Classically, Greeks can be computed by ‘bump-and-reprice’, which combines finite difference with MCI. Under certain continuity conditions of the payoff function with respect to the input parameter of interest, and when the MCI is performed using common random numbers, Greeks computed using bump-and-reprice can attain the same mean-squared error convergence in terms of the number of samples used in MCI (in chapter 7 of ref. 3). Consequently, the number of samples required to compute  $k$  Greeks with an  $\epsilon$  accuracy is  $O(k\sigma^2/\epsilon^2)$ . Using quantum gradient methods<sup>59</sup>, ref. 60 proposed a quantum acceleration of the classical bump-and-reprice method for computing Greeks, achieving a quadratic reduction in the number of samples required with respect to both  $k$  and  $\epsilon$ , resulting in a sampling complexity of  $O(\sqrt{k}\sigma/\epsilon)$ .

When computing Greeks, under certain smoothness conditions, one can move the differentiation operation inside the expectation and instead compute the sample average of path-wise derivatives, that is, derivatives of the random variable on a fixed realization of the stochastic process. Various derivative payoffs satisfy such conditions, and smoothing techniques can be applied to payoffs with singularities<sup>61</sup>. On a classical computer, path-wise derivatives can be computed analytically using automatic differentiation (AD)<sup>62</sup>, which scales with the logarithmic error dependence of arithmetic. In particular, the adjoint mode of AD allows for the computation of  $k$  gradients of a function with a cost (in terms of the number of function evaluations) independent of  $k$ , which provides a more efficient way of computing Greeks when the number of Greeks is large. On a quantum computer, AD may also be performed using reversible arithmetic. As noted in ref. 60, when AD is used, one could use QAE to speed up the error convergence. Furthermore, one could utilize neural networks combining AD with back-propagation to simultaneously learn the integral and its partial derivatives<sup>63</sup>. Although the advantage of using a neural network is unclear at the moment, if the neural network can learn both the price and the Greeks at once requiring only  $o(\sigma^2/\epsilon^2)$  training samples, it would be superior to bump-and-reprice. Unfortunately, current architectures for quantum neural networks (QNNs) pay a classical sampling cost when computing gradients, that is, using the parameter-shift rule<sup>64</sup>, which makes quantum advantage unlikely.

In addition to pricing and Greeks computation, MCI is also widely used in computing other risk metrics that involve the estimation of expected quantities from a stochastic process. Similar to derivative pricing, QMCI can be applied to these tasks leading to a potential quantum speedup. Specifically, quantum methodologies based on



QMCI have been demonstrated for the computation of value-at-risk<sup>65</sup> and credit risk<sup>66</sup>.

## Quantum methods for differential-equation-based pricing and risk analysis

As mentioned earlier, the expectation of certain random variables, whose evolution is described by SDEs, can be formulated as the solution to parabolic PDEs. This connection between SDEs and PDEs allows one to study stochastic processes using deterministic methods. One common numerical technique for solving PDEs is the finite-difference method, which approximately transforms the differential equation into a system of linear equations on discretized grid points<sup>67</sup>. In ref. 68, a finite-difference method-based approach combined with quantum linear systems algorithms (QLSAs) was used to solve the multi-asset Black–Scholes PDE. Although under various assumptions about the conditioning of the linear systems and data-access model, the quantum linear system can be solved with an exponential reduction in the dependence on the dimension, a sampling cost needs to be paid to readout the solution from the quantum state, making this technique reminiscent of classical and quantum MCI. Nevertheless, it may still be beneficial to choose PDE solving methods over MCI, as it has been shown that the former can have better convergence on certain errors caused by the discretization in time, for example, when pricing barrier options<sup>69</sup>.

For second-order linear PDEs, it is possible to transform the PDE into a form that resembles the Schrödinger equation and use Hamiltonian simulation techniques to simulate the dynamics generated by such PDEs. If the resulting evolution is non-unitary, a block encoding<sup>59</sup> may be necessary to convert it into a unitary evolution. It has been shown that if all eigenvalues of the evolution matrix are known, then an exponential speedup can be achieved in simulating the Hamiltonian dynamics compared with classical methods<sup>70–74</sup>. However, as the solution of the PDE is encoded in a quantum state, QMCI, or QAE, is still needed to obtain the desired quantity from the PDE solution, thus adding additional overhead.

Variational quantum simulation (VQS) can also be used to simulate the solution for certain PDEs as the evolution of a quantum state. Reference<sup>75</sup> showed that this could be done for the Black–Scholes PDE. Then ref. 76 extended the approach to the more general Feynman–Kac PDEs and variants and also discussed the potential applicability to the pricing of American options (that is, the Hamilton–Jacobi–Bellman equation) and models with stochastic volatility. Reference<sup>77</sup> used VQS to simulate the trinomial-tree model<sup>78</sup> for solving SDEs and proposed a methodology for computing the expectation values of random variables from the SDE.

The aforementioned algorithms encode the solution into a quantum state such that the computational basis states denote the discretized values of the spatial variable and the amplitudes are proportional to the corresponding values of the function. Reference<sup>79</sup>, however, proposed an alternative approach that uses a QNN as a parameterized ansatz for the PDE solution. The variational parameters are updated through gradient descent to satisfy the PDE. One thing to note about this approach<sup>79</sup> is that the input data are encoded into a product state. This potentially allows one to implement a more expressive version of the model using classical kernel methods<sup>80</sup>. However, it is believed that variational quantum models can introduce a regularization not accessible to kernel methods<sup>81</sup>, still making QNNs that encode the input data into a single product state potentially useful. Although VQS and QNNs are often less resource-intensive than QLSAs, without further

experimentation it remains unclear whether there is any quantum advantage from these heuristic techniques.

## Optimization

In this section, we discuss the potential of using quantum algorithms to help solve various optimization problems that are prevalent in finance. Most financial optimization problems are typically highly constrained linear or quadratic programs with continuous variables, discrete variables or both. Quantum computing provides various heuristics and algorithms for tackling such problems. In the sections that follow, we review the various categories of financial optimization problems and a multitude of quantum algorithms that are applicable to such problems; we discuss whether they could actually be useful.

### Quantum methods for continuous optimization

In continuous optimization, the problem decision variables are real-valued. Convex programming<sup>82</sup> is one area of continuous optimization for which there exist various efficient classical algorithms for structured problems<sup>83</sup>. Notable examples of structured convex problems<sup>84</sup> are symmetric cone programs (SCPs), such as linear programs (LPs), second-order cone programs (SOCPs) and semidefinite programs (SDPs), which frequently appear in financial applications. Financing or cash-flow management and arbitrage detection are examples of important financial linear optimization problems. In a short-term financing problem (in chapter 3 of ref. 85), the goal is to select cash flows – for example, assets with fixed return rates, or lines of credit – to match periodic quotas while maximizing assets. As the amount of cash flow or credit obligation is directly proportional to the amount held, the various problem constraints are linear.

Portfolio optimization is the process of selecting the best set of assets and their quantities from a pool of assets being considered according to some predefined objective. The objective can vary depending on the preference of an investor regarding financial risk and expected return. The modern portfolio theory<sup>86</sup> focuses on the trade-offs between risk and return to produce what is known as an efficient portfolio, which maximizes the expected return given a certain amount of risk. This trade-off relationship is represented by a curve known as the efficient frontier. The expected return and risk of a financial portfolio can often be modelled by looking at the mean and variance, respectively, of portfolio returns. The problem setup for portfolio optimization can be formulated as constrained utility maximization:

$$\max_{\mathbf{x} \in \mathcal{F}} \mathbf{x}^T \boldsymbol{\mu} - q \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} \quad (2)$$

over some feasible set of portfolios,  $\mathcal{F}$ . In portfolio optimization,  $\mathbf{x}$  is the vector of asset allocations,  $\boldsymbol{\Sigma}$  is the covariance or correlation matrix of the assets,  $\boldsymbol{\mu}$  is the vector of expected returns and  $q$  is a scaling factor that represents risk-averseness. The signs of the problem variables,  $\mathbf{x}$ , can also be used to indicate long or short positions. The problem is typically highly constrained. For example, a weight may be assigned to each asset class, such as stocks, bonds, futures and options, restricting the proportion of each in the portfolio. Assets within each class are then allocated according to their respective risks, returns, time to maturity and liquidity, during which additional constraints may apply. In this section, we consider cases in which the variables are continuous-valued, but  $\mathcal{F}$  can be non-convex. The discrete case is addressed in the following section. As most classical and quantum algorithms for continuous optimization target convex problems, specifically SCPs, we also mainly focus on that.

## Glossary

### Autocallable

A financial product that pays the holder a high return if the value of the underlying asset passes an upside barrier.

### Black–Scholes model

A mathematical model for the dynamics of a financial market containing derivative investment instruments.

### Bump-and-reprice

A method to estimate the sensitivity of the price of a financial derivative with respect to an underlying parameter by evaluating the price at different values of the parameter and taking the difference.

### Financial derivative

A financial contract that derives its value from the performance of an underlying entity.

### Hamilton–Jacobi–Bellman equation

An equation that gives a necessary and sufficient condition for optimality of a control with respect to a loss function.

### Martingale measure

A probability measure such that the conditional expectation of a random variable in a sequence given the value of a random variable prior in the sequence is equal to the value of this

prior random variable on which the expectation is conditioned.

### Option

A financial contract that gives the holder the right, but not the obligation, to buy or sell an underlying asset at an agreed-upon price and time frame.

### Target accrual redemption forward

A financial product that allows the holder to achieve a target rate (interest rate, exchange rate and so on) or rate range on a pre-defined schedule (for example, monthly) up to a limit on the maximum payout and under certain conditions on the extreme values of the rate observed in the market (spot rate). It achieves this goal by paying the holder a positive amount if the spot rate is higher than a target value and negative if lower, until the maximum amount of accrual has been reached or the spot rate hits certain upper and/or lower barriers.

### Vapnik–Chervonenkis (VC) dimension

A measure of the capacity of a set of functions that can be learnt by a statistical binary classification algorithm, defined as the cardinality of the largest set of data points that the algorithm can always learn a perfect classifier for an arbitrary labelling.

Considering that SCPs can be solved very efficiently classically, there is a small margin for quantum speedup in practice. Still, quantum reductions have been observed in worst-case running times in terms of the problem dimension owing to the existence of fast quantum algorithms for basic linear algebra subroutines (BLASs). However, a quantum BLAS (QBLAS)<sup>59,87</sup> comes with three notable caveats that make it challenging to compare quantum and classical worst-case complexities. First, a QBLAS operates on the spectrum of matrices and thus has polynomial dependence on the conditioning of the matrix, which classically typically appears only in iterative algorithms. Second, retrieving data from a quantum device requires sampling and consequently incurs a classical sampling cost that depends poorly on the desired error. Finally, the algorithms require access to the classical data in quantum superposition, which can only be done efficiently without quantum memory in certain cases when the input matrices are sparse.

Classical techniques for solving SCPs fall into two categories: those with better dependence on the problem size and those with better dependence on the error in the solution. The first category is mainly

dominated by the matrix multiplicative-weight (MMW) meta-algorithm<sup>88</sup>. It can be shown that the complexity for general SDP solving is<sup>89</sup>  $\tilde{O}(mns\gamma^4 + ns\gamma^7)$ , where  $m$  is the number of constraints,  $n$  is the number of variables,  $s$  is the sparsity and  $\gamma$  is a function of the desired error. The notation  $\tilde{O}(\cdot)$  ignores factors of the form  $O(\text{polylog}(\cdot))$  in the complexity. Currently, the best quantum version of MMW for general SDP solving, in ref. 90, has complexity  $\tilde{O}(s\sqrt{m}\gamma^4 + s\sqrt{n}\gamma^5)$ , in terms of number of gates and calls to an oracle that provides matrix elements. This has the optimal dependence on  $m$  and  $n$  one can hope for in general. However, when fine-tuned to the SDP relaxation of the MaxCut problem, for certain graphs, the classical MMW algorithm runs in  $\tilde{O}(m)$ . As MMW already depends poorly on the desired problem error and assumes sparse matrices, quantum MMW may not suffer from the caveats of QBLASs. Although MMW has not been extensively studied for solving SCPs in finance, the scalar multiplicative-weights update method<sup>91</sup>, which is a special case of MMW, has. The scalar multiplicative-weights method is applicable to online portfolio optimization<sup>92</sup> – an example of online convex optimization<sup>93</sup>. Online portfolio optimization, in which the algorithm adapts its portfolio selection as the market changes, has received attention from the quantum computing community<sup>94</sup>.

The scalar multiplicative-weights framework has also been used to produce sublinear algorithms for estimating Nash equilibrium of two-player zero-sum games<sup>95</sup>, a type of matrix game over simplices. Quantum algorithms for Gibbs sampling have been used to produce quadratic reductions in the dimension dependence for solving zero-sum games<sup>96,97</sup>. These algorithms have been applied to the task of estimating a martingale measure from market data and derivative pricing<sup>98</sup>. Furthermore, quantum amplitude amplification has been used to provide a quadratic reduction in dimension dependence over classical sublinear algorithms for a more general class of matrix games<sup>99</sup> solved with multiplicative weights and online mirror descent in a primal-dual fashion<sup>100</sup>, with applications to training kernel classifiers<sup>101</sup>. Online mirror descent is a generalization of the scalar multiplicative-weights framework.

The second category of algorithms, called the polynomial-time methods, consists of the interior-point methods (IPMs) and cutting planes. In theory, the fastest method in this category for solving general SDPs is based on cutting planes<sup>102</sup>, whereas IPMs are the fastest for LPs<sup>103</sup> and SOCPs<sup>104</sup>. References<sup>105,106</sup> use QBLASs, for example, QLSAs, to perform Newton's method, which is a common IPM subroutine. For all three conic programs, the explicit dependence on the problem dimension is superior to the fastest classical algorithms. Unfortunately, these quantum IPMs suffer from all three aforementioned caveats with QBLASs. Classical IPMs can be implemented in polynomial time with logarithmic dependence on the conditioning of the matrix, that is, through classical arithmetic. The classical sampling cost that the quantum algorithm pays effectively negates the good error dependence of classical IPMs and only solves the Newton system approximately. It was shown that this results in the quantum IPM only solving an approximation of the original optimization problem<sup>105</sup>. Proposals have been made to use variants of the standard classical IPM that may be better candidates for quantization<sup>107</sup> and combat some of the issues brought up. Still, resource estimates for simple portfolio optimization problems indicate a lack of advantage with quantum IPMs<sup>108</sup>.

Although not a polynomial-time algorithm, the simplex method is very fast in practice for solving LPs for which quantum algorithms for various subroutines have been proposed<sup>109</sup>. Also, QLSAs have been directly applied to portfolio optimization problems in the case with

linear equality constraints, in which the solution can be obtained in closed form by solving a linear system<sup>110,111</sup>. Besides suffering from the general issues with QBLASs, the overall benefits of these approaches remain indeterminate owing to their limited applicability.

Currently, it seems to still be an open question whether quantum computation can provide an advantage for structured convex programs in general and not just finance. However, for well-conditioned problems with sparseness or access to sufficient quantum memory, there may be speedups realizable on practical problems. However, these problems can already be solved fast with classical computation.

In the continuous, non-convex setting, it has been demonstrated that convex relaxations may exist for some financial problems, leading to comparable performance as global optimizers, but with significantly reduced computational cost. In particular, tax-aware portfolio optimization is a variant of the standard modern portfolio theory problem that accounts for tax liabilities. Although the tax penalty term in the cost function is non-convex, numerical evidence shows that a convex relaxation to an SOCP can result in a solution quality nearly identical to that from exact solvers<sup>112</sup>. Similarly, sparse portfolio optimization, which imposes a constraint on the number of nonzero allocations in the solution, admits an SOCP relaxation that provides fast and relatively accurate solutions for some asset classes<sup>113</sup>. For quantum algorithms, there has been work demonstrating speedups for certain continuous, non-convex landscapes<sup>114</sup> as well as developing and benchmarking quantum analogues of gradient descent<sup>115</sup>.

## Quantum methods for discrete optimization

Many optimization problems in finance require that the solutions take values from a discrete set, as opposed to a continuous spectrum. Examples include some of the most commonly seen optimization problems in finance, such as portfolio optimization also discussed in the previous section. In many portfolio optimization problems, the optimal solution from a convex optimization may suggest the allocation of fractional units of an asset, whereas market constraints often require that the positions on these assets are integers or multiples of a fixed increment. These financial use cases call for discrete optimization methods or integer programming, in which the variables to optimize are restricted to integers, and more generally, mixed integer programming (MIP), in which some of the variables are integers. For the unfamiliar reader, an introduction to discrete optimization can be found in ref. 116.

One approximate approach for solving MIP problems is to convert the MIP into a continuous optimization problem by relaxing the integer constraints and then round the solution to the nearest feasible values allowed by the integer constraints. Specifically, for a mixed integer linear programming problem, one may apply linear programming relaxation to remove all integer constraints. In the more general case, convex hull relaxation may be used in which the feasible set of solutions is replaced by the minimal convex set that contains the feasible set. In portfolio optimization problems, in which typical values of the asset positions are much larger than the minimum allowed increment, approximated optimal solutions through relaxation are often acceptable with minor modifications. This is usually the case with portfolios of stocks, in which the minimum holding size is one share and the holding sizes are usually at least two orders of magnitude larger. However, the fixed-income and derivatives markets usually require a much larger unit trading size, which makes the approximate solutions from the relaxed problem of unsatisfactory quality. Therefore, these problems often have to resort to discrete optimization techniques.

Branch-and-bound (B&B) methods form a powerful class of algorithms for solving hard optimization problems, such as MIPs, and can provide either a certificate of optimality or an optimality gap<sup>117</sup>. The optimality gap is the difference between the best-known solution and a known bound. B&B is the core algorithm in the majority of commercial solvers and is thus commonly adopted by financial institutions. At a high level, a B&B algorithm constructs a tree of subproblems, for example, convex relaxations in the case of MIP, whose solutions represent bounds on a possible solution to the non-relaxed problem one could hope to obtain. Subroutines based on heuristics are used for searching subproblems to solve, specifically leaves in a tree, and prune them when possible, as the tree can be exponentially large in problem size.

Quantum walk search (QWS)<sup>118</sup> is a powerful framework for speeding up classical search algorithms and has been applied to important optimization algorithms used in finance. It also worth noting that the overarching framework of quantum walks enables an at most quadratic reduction in the time to simulate symmetric Markov chains<sup>119</sup>, which is potentially applicable to simulating stochastic processes arising in finance. Reference<sup>120</sup> developed QWS-based techniques, which provide an almost quadratic speedup for tree search in terms of the size of the tree. However, this does not allow the integration of search heuristics or early stopping, both of which have been found to significantly boost the performance of classical algorithms on typical problem instances. Advances have been made in this direction to integrate depth-first search heuristics<sup>121</sup> and, more recently, a large class of practical heuristics<sup>122</sup>. However, similar to the case of quantum MCI for stochastic modelling, it is unclear whether an actual reduction in the time to solution could be achieved when considering the resources required to solve the subproblems.

Simulated annealing (SA)<sup>123</sup> is a widely used Markov chain Monte Carlo (MCMC) method<sup>124</sup> for combinatorial optimization. Quantum walks have been used to quadratically reduce the spectral-gap dependence of the asymptotic convergence time to an exact solution<sup>125–127</sup>. However, it is still unclear whether, in general, the dependence on all parameters with quantum can be made better than or even match the dependence they have with classical MCMC<sup>128</sup>. Furthermore, classical SA is typically used as a heuristic and not run until the Markov chains have converged to their stationary distributions<sup>129,130</sup>. Still, ref. 131 has investigated the use of quantum SA as a heuristic as well as designed gate-efficient constructions for implementing the quantum walk operator.

A framework<sup>132–134</sup> was developed on the basis of Grover's algorithm<sup>135</sup>, which is a special case of QWS that makes use of global problem information for unstructured optimization. This algorithm was subsequently generalized in ref. 89 to allow for arbitrary priors over the search space. The unstructured search framework requires the ability to query oracles for evaluating the function to optimize and the various constraints. Although providing a quadratic speedup in query complexity, over classical unstructured search, quantum unstructured search is not as resource-efficient as quantum-walk-based techniques, which rely on local, as opposed to global, information<sup>136</sup>. In the case of discrete optimization, uniform superpositions over the unconstrained search space can be prepared with low gate complexity, and for NP optimization problems the oracles for checking the constraints and evaluating the cost function can be implemented with an efficient gate complexity<sup>137</sup>.

Finally, there is also the short-path algorithm<sup>138,139</sup>, which makes use of techniques from quantum-walk literature, and takes advantage



of problem-specific information to achieve speedups that are superior to Grover-based algorithms for some discrete optimization problems.

In addition to the various quantized versions of classical algorithms, there are quantum heuristic algorithms<sup>140</sup>. These consist of quantum annealing<sup>141–143</sup> and variational quantum algorithms: the quantum approximate optimization algorithm (QAOA)<sup>144,145</sup>, variational quantum eigensolver (VQE)<sup>146,147</sup> and VQS<sup>148–150</sup>. In general, these methods can naturally handle unconstrained binary optimization problems, but have also been applied to continuous optimization problems<sup>151,152</sup>. It is possible to efficiently restrict the evolution of these quantum heuristics to respect binary-variable constraints<sup>145,153–156</sup>, at least when implemented on a universal digital quantum computer.

Variational quantum algorithms usually suffer from difficulties in tuning the variational parameters and hyperparameters, which by itself can be NP-hard<sup>157</sup>. In addition, for certain initializations, the gradients with respect to the parameters can vanish even when the model is far from convergence<sup>158,159</sup>, requiring exponentially many samples to estimate them. Still, the convergence of VQE has been demonstrated in the over-parameterized regime<sup>160</sup>, even with exponentially small gradients.

Although existing quantum annealing devices are large scale and do not need to meet the error-tolerance requirements of universal devices, the overall benefit of quantum annealing is yet to be determined. Quantum tunnelling can allow for penetrating tall, thin potential barriers<sup>161</sup>. However, it is not possible to know a priori whether the potential barriers that appear in a practical problem permit tunnelling. For highly constrained financial optimization problems with multiple types of variables, it is often costly to transform the problem into an unconstrained quadratic binary optimization problem, especially when the number of qubits is limited. Still, owing to the availability of relatively-large quantum annealers, the community has already begun experimenting with applications, such as portfolio optimization<sup>162–164</sup> and crash detection<sup>165</sup>.

Finally, in contrast to VQE and VQS, QAOA has been observed to have additional advantages, such as only using two parameters per layer and theoretical and numerical results showing the ability to transfer optimized parameters between problem instances<sup>166–168</sup>. Furthermore, in some instances, the parameter optimization can be performed efficiently classically using simulators of quantum circuits<sup>169–172</sup>. There is still a lot of work that needs to be done to better understand the performance of quantum heuristic algorithms, especially on financial optimization problems<sup>173</sup>.

## Quantum methods for dynamic programming

In some optimization problems, the information needed to make subsequent decisions is only revealed after intermediate decisions are made. Therefore, in such cases, decisions must be made in a sequential manner, and an optimal strategy must take into account both current and future decisions. Solving these decision-making problems often requires dynamic programming, in which a problem is solved recursively by reducing the main problem into a series of smaller subproblems that are easier to solve.

Dynamic programming problems<sup>174</sup> are also commonly seen in finance. In addition to the American option pricing problem mentioned in the first section, another important place where dynamic programming appears in finance is in the structuring of collateralized mortgage obligations (CMOs) (in chapter 15 of ref. 85). A CMO bundles a pool of mortgages and rearranges their cash flows into multiple ‘tranches’ that are paid in sequence. The issuer of a CMO is often interested in an optimal payment schedule (structure) of these tranches to minimize

the payment obligations to the CMO owners, which needs to be solved recursively as the optimal start and end times of the  $k$ th tranche depend on the optimal schedule of the preceding  $k - 1$  tranches. Although, to the best of our knowledge, there have not been demonstrations of quantum solutions to the structuring of CMOs and other similar financial problems, quantum algorithms have been proposed for reducing the exponential dependence of exponential-time dynamic programming for various NP-complete problems<sup>175</sup>.

## Machine learning

In this section, we discuss the potential of using quantum algorithms to help solve ML tasks that arise in various financial applications. The field of ML<sup>176,177</sup> has become a crucial part of various applications in the finance industry. Rich historical financial data and advances in ML make it possible, for example, to train sophisticated models to detect patterns in stock markets, find outliers and anomalies in financial transactions, automatically classify and categorize financial news and optimize portfolios<sup>7,178</sup>.

Quantum algorithms for ML can be further divided into methods for accelerating classical techniques, typically by applying QBLASs, and quantum-native algorithms, which attempt to harness the classical intractability of quantum simulation to build more expressive models. The former typically requires an error-corrected quantum computer, whereas it is argued that the latter may not and is more near term<sup>42</sup>. The methods for accelerating classical algorithms need to overcome the various limitations of quantum linear algebra addressed in the previous section. The most notable one being data loading, and unfortunately, most of these algorithms work in a setting with quantum memory<sup>179,180</sup>. It is currently unclear whether such a device can be constructed. Furthermore, various approaches only provide speedups when the data have low-rank approximations, and in this setting, the speedup is not exponential as it was once thought. However, there remain some quantum ML algorithms that do not rely on this low-rank assumption and may remain impervious to dequantization<sup>181</sup>. Further research is needed to determine whether an exponential speedup of classical ML problems is still possible with quantum computing.

Quantum-native models, such as QNNs, quantum circuit Born machine (QCBM) and quantum kernel methods, circumvent the issues of QBLASs and can be intractable to classically simulate; however, it is currently unclear whether they provide advantage on classical problems. Specifically, although these methods can be more expressive, this has only been shown to provide advantage for problems involving data generated from quantum processes<sup>182</sup>. In addition, initial numerical evidence suggests that this does not extend to classical problems<sup>183</sup>. Parameterized quantum methods, such as QNNs, can be challenging to train in general<sup>158</sup> and do not currently have back-propagation algorithms that are as efficient as those for classical NNs. Although, at the moment, quantum advantage on classical problems by using quantum-native methods appears to be unlikely, there is still significant algorithmic research that needs to be done to be certain. Furthermore, as quantum hardware advances, one will eventually be able to benchmark these heuristic algorithms on real-world problems.

In the following sections, we highlight quantum ML algorithms that could be applied to financial problems and hope to entice further research towards quantum advantage.

## Quantum methods for regression

Regression is the process of fitting a numeric function from the training data set. This process is often used to understand how the value changes



when the attributes vary, and it is a key tool for economic forecasting. For example, regression models can be used in asset pricing<sup>63,184</sup> and volatility forecasting<sup>185</sup>. Least squares is one of the most widely adopted regression models, and corresponding quantum algorithms have been proposed<sup>186–188</sup>. Moreover, quantum annealing has been used to solve least-squares regression when formulated as quadratic unconstrained binary optimization<sup>189</sup>. In addition to least squares, Gaussian process regression is another technique with applications in finance<sup>190</sup>, but it suffers from a slow classical runtime. Reference<sup>191</sup> proposed a QLSA-based algorithm that can obtain up to an exponential speedup for Gaussian process regression for certain sparse, high-rank kernels. Another study<sup>192</sup> used a QNN with a quantum feature map that encodes classical input data into a unitary with configurable parameters. This technique is dubbed as quantum circuit learning<sup>192</sup> and has been numerically shown to allow for a low-depth circuit, hence implying important application opportunities in quantum algorithms for finance.

## Quantum methods for classification

Classification is the process of placing objects into predefined groups. This type of process is also called pattern recognition. This area of ML can be used effectively in risk management and large data processing when the group information is of particular interest, for example, in credit-worthiness identification<sup>193</sup> and fraud detection<sup>194</sup>. There are many well-known classical classification algorithms, such as linear classification, support vector machine (SVM), nearest centroid and neural-network-based methods. Quantum algorithms could be used as subroutines to speed up existing classical algorithms. Alternatively, a quantum version of such algorithms could be developed. Both cases could potentially benefit financial applications. Sublinear quantum algorithms for training linear and kernel-based classifiers have been proposed<sup>195</sup>, which gives a quadratic improvement for training classifiers with constant margin. Reference<sup>196</sup> proposed the quantum least-squares SVM, which uses QLSA. Reference<sup>106</sup> proposed a quantum algorithm for SOCPs that was subsequently applied to the training of the classical  $\ell_1$  SVM, realizing a small polynomial speedup through numerical experiments. In addition, classical SVMs using quantum-enhanced feature spaces to construct quantum kernels have been proposed<sup>197</sup>. Quantum nearest-neighbour classification algorithms based on Euclidean distance<sup>198</sup> and Hamming distance<sup>199</sup> as the metric have also been investigated. Reference<sup>200</sup> proposed using a quantum  $k$  maxima-finding algorithm to find the  $k$ -nearest neighbours and use the fidelity and inner product as measures of similarity. Various types of QNNs have also been applied to classification tasks<sup>201–203</sup>. Quantum algorithms that improve the complexity of inference or training of a classical neural network have also been developed<sup>204,205</sup>.

## Quantum methods for boosting

Boosting algorithms<sup>206</sup> use queries to a weak learning algorithm, which often produces models that classify or regress marginally better than a random guess, to construct a strong learner that can achieve an arbitrarily low prediction error. Adaptive boosting (AdaBoost)<sup>207</sup> was the first boosting algorithm and can be viewed as an instance of the multiplicative-weights update method. One component that appears in the complexity of AdaBoost is a linear dependence on the Vapnik–Chervonenkis (VC) dimension of the weak learners. Reference<sup>208</sup> demonstrated a quantization of the AdaBoost algorithm, which promises a quadratic reduction in the dependence on the VC dimension. However, this is at the cost of a significantly worse dependence on the margin over

random guess. Reference<sup>209</sup> quantized the Smooth boost algorithm<sup>210</sup>, which guarantees a similar quadratic reduction in the VC dependence, but with better margin dependence than ref. 208. A major limitation for both approaches is that they need to have access to quantum examples, which requires preparing quantum states encoding distributions over the training data. Finally, there is also the QBoost framework<sup>211</sup>, which proposes to use quantum heuristics for discrete optimization to select a linear combination of weak classifiers and has been applied to financial-risk detection<sup>212</sup>.

One very popular form of boosting is called gradient boosting<sup>213</sup>, of which a high-performance instantiation has been dubbed extreme gradient boosting (XGBoost)<sup>214</sup>. Currently, there does not appear to be any quantization of gradient boosting. Given the obvious generality of boosting, these models have been applied to a wide variety of ML applications in finance. Particularly, boosting has been applied to forecasting, such as derivative pricing<sup>215</sup>, financial-crisis prediction<sup>216</sup>, credit-risk assessment<sup>217</sup> and volatility forecasting<sup>218</sup>.

## Quantum methods for clustering

Clustering, or cluster analysis, is an unsupervised ML task. It explores and discovers the grouping structure of the data. In finance, cluster analysis can be used to develop a trading approach that helps investors build a diversified portfolio<sup>219</sup>. It can also be used to analyse different stocks, such that the stocks with high correlations in returns fall into one basket<sup>220</sup>. Classically, there is the  $k$ -means clustering algorithm (also known as Lloyd's algorithm), and quantum computing can be leveraged to accelerate a single step of  $k$ -means<sup>221</sup>. Reference<sup>198</sup> showed that a step for  $k$ -means can be performed by using a number of queries that scales as  $O(M\sqrt{k\log(k)/\epsilon})$ , where  $M$  is the number of data points,  $N$  is the dimension of the vectors and  $\epsilon$  is the error. Although a direct classical method requires  $O(kMN)$ , the quantum solution is substantially better under plausible assumptions, such as that the number of queries made by the algorithm is independent of the number of features. Reference<sup>222</sup> proposed  $q$ -means, which is the quantum equivalent of a perturbed version of  $k$ -means called  $\delta$ - $k$ -means. The  $q$ -means algorithm has a running time that depends polylogarithmically on the number of data points. A noisy intermediate-scale quantum version of  $k$ -means clustering using quantum computing has been proposed<sup>223</sup>. Reference<sup>224</sup> discussed a quantum version of expectation maximization, a common tool used in unsupervised ML. Another version of  $k$ -means clustering has been developed<sup>225</sup> on the basis of a quantum expectation-maximization algorithm for Gaussian mixture models.

Another clustering method that has achieved great success classically is spectral clustering<sup>226</sup>. However, it suffers from a fast-growing running time of  $O(N^3)$ , where  $N$  is the number of points in the data set. Reference<sup>227</sup> used phase estimation and QAE for spectral clustering on quantum computers, and ref. 228 proposed another quantum algorithm to perform spectral clustering. Reference<sup>229</sup> developed quantum algorithms using the graph Laplacian for ML applications, including spectral  $k$ -means clustering. More discussions on unsupervised quantum ML techniques can be found in refs. 230,231. A technique for using quantum annealing for combinatorial clustering was described in ref. 232. Reference<sup>233</sup> proposed to reduce the clustering problem to an optimization problem and then solve it via a VQE combined with non-orthogonal qubit states.

## Quantum methods for generative learning

Unsupervised generative learning is at the forefront of deep learning research<sup>234</sup>. The goal of generative learning is to model the probability

distribution of observed data and generate new samples accordingly. One of the most promising aspects of achieving potential quantum advantage lies in the sampling advantage of quantum computers, especially considering that many applications in finance require generating samples from complex distributions. Therefore, further investigations of generative quantum ML for finance are needed.

The QCBM<sup>235</sup> directly exploits the inherent probabilistic interpretation of quantum wave functions and represents a probability distribution using a quantum pure state instead of the thermal distribution. Numerical simulations suggest that in the task of learning the distribution, QCBM at least matches the performance of the restricted Boltzmann machine and demonstrates superior performance as the model scales<sup>236</sup>. (A Boltzmann machine is an undirected probabilistic graphical model inspired by thermodynamics<sup>237</sup>.) QCBM was also used to model copulas<sup>238</sup>. Reference<sup>239</sup> proposed separating the training and sampling stages and showed numerically that probability distributions can be trained and sampled efficiently, whereas SDEs can act as differential constraints on such trainable quantum models.

Bayesian networks are probabilistic graphical models<sup>237</sup> representing random variables and conditional dependencies via a directed acyclic graph, in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable. Bayesian inference on this type of graphs has many applications, such as prediction, anomaly detection, diagnostics, reasoning and decision-making with uncertainty. Although exact inference is  $\#P$ -hard, a quadratic speedup in certain parameters can be obtained by using quantum techniques<sup>240</sup>. Mapping this problem to a quantum Bayesian network seems plausible as quantum mechanics naturally describes a probabilistic distribution. Reference<sup>241</sup> introduced the quantum Bayesian network as an analogue to classical Bayesian networks, and ref. 242 proposed a procedure to design a quantum circuit to represent a generic discrete Bayesian network. Potential applications in finance include portfolio simulation<sup>243</sup> and decision-making modelling<sup>244</sup>.

Classically, inference is usually performed by using MCMC methods to sample from the equilibrium distribution of the model (the Boltzmann distribution) (in chapter 16 of ref. 234). Because of the intractability of the partition function in the general Boltzmann machine, the graph structure is typically bipartite, resulting in a restricted Boltzmann machine<sup>245,246</sup>. Quantum Boltzmann machines have been implemented with quantum annealing<sup>247–249</sup>. In addition, a gate-based variational approach using variational quantum imaginary-time evolution has been designed<sup>250</sup>.

Generative adversarial networks (GANs) represent a powerful tool for classical ML: a generator tries to create statistics for data that mimic those of the real data set, whereas a discriminator tries to discriminate between the true and fake data. Reference<sup>251</sup> introduced the notion of quantum GANs, in which the data consist of either quantum states or classical data, and the generator and discriminator are equipped with quantum information processors. The authors showed that when the data consist of samples of measurements made on high-dimensional spaces, quantum adversarial networks may exhibit an exponential advantage over classical adversarial networks. Quantum GAN has been used to learn and load random distribution and can facilitate financial derivative pricing<sup>43</sup>.

## Quantum methods for feature extraction

Feature extraction techniques aim to preprocess raw training data to identify important components, transform the data to a more meaningful representation space or reduce dimension<sup>252</sup>. However,

sometimes, this can result in modifying the data in such a way that it is not human-interpretable, which makes it difficult to use such techniques in the highly regulated financial industry. Thus, for finance, it is important to find efficient, useful and explainable feature extraction methods. One particular simple algorithm is principal component analysis (PCA), which finds a low-dimensional representation of the data on a linear manifold. Various quantum versions of PCA have been proposed<sup>253–256</sup>, all of which encode principal components in quantum superposition. Similar to most techniques that use QBLASs, it is particularly difficult to obtain useful classical information from the quantum output. However, one could feed the result to other quantum-linear-algebra-based ML models.

Interest in topological data analysis<sup>257</sup>, specifically persistent homology, has surged in the quantum community<sup>258</sup>. In a practical setting, the potential speedup over classical algorithms is believed to be at most polynomial for dense clique complexes<sup>259</sup>. In addition, various quantum optimization heuristics could be used for combinatorial feature selection<sup>260–263</sup>, which means choosing a subset of the input features to use according to some measure of importance.

## Quantum methods for reinforcement learning

Reinforcement learning (RL) is an area of ML that considers how agents ought to take actions in an environment to maximize their reward<sup>264</sup>. It has been applied to many financial applications, including pricing<sup>265</sup> and hedging<sup>266</sup> of contingent claims, portfolio allocation<sup>267</sup>, automated trading under market frictions<sup>268,269</sup>, market making<sup>270</sup>, asset liability management<sup>271</sup> and optimization of tax consequences<sup>272</sup>. There has been a line of work investigating applying quantum algorithms to RL when one has access to state or actions spaces, for example, access to Markov decision processes through quantum oracles. Reference<sup>273</sup> proposed using Grover's algorithm to amplify the probability of observing actions that result in a positive reward. In addition, the authors showed that the approach makes a good trade-off between exploration and exploitation using the probability amplitude and can speed up learning through quantum parallelism. Additionally, ref. 274 explored ways to apply QMCI and gradient estimation to quantize the policy gradient method, given access to a Markov decision process through quantum oracles. Reference<sup>275</sup> demonstrated that the computational complexity of a particular model, projective simulation, can be quadratically reduced. In this scenario, the agent only requires quantum access to an internal memory that it builds by interacting with the environment.

Cutting-edge research in classical RL focuses on the approximate setting, in which deep neural networks are used to handle state and action spaces that would otherwise be intractable with tabular RL methods<sup>264</sup>. The benefits from quantum algorithms – for instance, with QNNs – in such a scenario appear to be unclear, similarly to when variational quantum models are used for supervised or unsupervised models. However, there has been some work in this direction<sup>276–282</sup>.

## Dequantized algorithms

The framework for dequantizing QBLAS-based algorithms started with the breakthrough result in Ref. 283. Dequantization results in a classical randomized algorithm that achieves a dimension dependence that is competitive with the best quantum algorithm when provided sampling access to a certain data structure containing low-rank data. These algorithms typically have very poor error dependence. Following Ref. 283, various sublinear classical algorithms have been developed for solving low-rank SDPs, performing PCA, clustering and more<sup>284</sup>. These quantum-inspired algorithms can potentially provide benefits

when high precision is not required and could be applied to financial applications<sup>285</sup>. Despite these algorithms refuting much of the originally claimed exponential quantum speedups for ML, they could potentially inspire useful algorithms for high-dimensional financial problems.

## Outlook

Quantum computers are expected to surpass the computational capabilities of classical computers and provide a speedup for real-world applications. As outlined in this Review, the financial industry deals with various computationally challenging problems for which many applicable quantum algorithms exist. Promising provable quantum advantages have been discovered in certain black-box settings, such as MCI. However, it is still unclear whether any of these proposed approaches for stochastic modelling, optimization and ML can be turned into an end-to-end quantum advantage on practical problems. In finance, computational time and accuracy can often directly translate to the profit and loss of the business for which the problems are being solved, inasmuch that any actual wall-clock speedup and associated model performance improvement that new forms of computing could bring can have a tremendous impact on the financial industry. For example, fast and accurate evaluation of the risk metrics in derivatives trading is crucial in effectively hedging the risks especially under volatile market conditions. Fraud detection is apparently another time-sensitive use case in finance, as early and accurate detection of fraudulent activities can avoid potentially significant monetary loss and reputational damage to a financial institute. As a result, the financial industry is perfectly positioned to be an early adopter and take full advantage of quantum computing in the field of computational finance.

We close with a discussion on existing quantum hardware and architecture challenges that, if solved, could benefit quantum algorithms for financial applications. As mentioned earlier, state preparation, specifically the loading of quantum states encoding classical probability distributions, is a common task required in quantum algorithms for stochastic modelling. One potential hardware feature that could reduce state preparation complexity is native multicontrolled gates, whose implementations have been proposed for various architectures including cold atoms<sup>286</sup>, trapped ions<sup>287</sup> and superconducting qubits<sup>288</sup>.

Similarly, variational quantum algorithms may also benefit from the access to native multiqubit gates. In particular, when solving higher-order combinatorial optimization problems (for example, problems on hypergraphs<sup>167</sup>), variational quantum optimization algorithms, such as QAOA, often require multibody interactions, which can be encoded as parameterized multiqubit entangling gates. Therefore, having native access to such multiqubit gates would help reduce the circuit complexity of these algorithms. In addition to multiqubit gates, a larger class of native two-qubit interactions could also enable one to more efficiently implement QAOA mixers for constrained problems, such as the Hamming-weight-preserving XY-mixers<sup>145,153</sup>. As mentioned in the 'Optimization' section, financial problems are typically highly constrained. Moreover, having access to various native entangling gates may also motivate the design and implementation of more efficient circuits for quantum ML<sup>182</sup>.

Coherent quantum arithmetic could also be beneficial to quantum algorithms for finance. Specifically, quantum algorithms for stochastic modelling and realizations of oracles for quantum optimization algorithms will most likely require a significant amount of reversible arithmetic. Consequently, qubit-count requirements for these algorithms could potentially be reduced with the development of efficient quantum floating-point arithmetic<sup>289</sup>.

Another hardware feature that could drastically improve the feasibility for quantum algorithms for finance is quantum memory. Most of the algorithms for continuous optimization and PDE solving can work with a classical-write, quantum-read memory<sup>90</sup>. However, it has been recently highlighted that existing quantum memory technologies have fundamental limitations that make the realization of low-cost and scalable quantum memory without active error correction highly challenging<sup>290</sup>. Therefore, new quantum memory architectures are likely needed to overcome these limitations.

Additionally, the operation clock rate of the current quantum hardware is much slower compared with classical computers; this is also a limiting factor in the practical usability of many quantum algorithms. For example, quantum variational algorithms for ML require a large number of high-quality circuit evaluations to estimate gradients through sampling. Therefore, improving the gate operation speed in quantum computers would reduce the overhead in quantum algorithms, hence potentially bringing the advantage over classical algorithms to a wider range of problem sizes as opposed to only in the asymptotic regime. It is commonly observed that qubits encoded in 'natural atoms', such as trapped ions, produce high-fidelity gate operations, but are typically slow. By contrast, 'artificial atoms' such as superconducting qubits are known to be fast, but low fidelity. Combining these aspects is critical to make effective use of near-term quantum devices for financial applications. Moreover, current quantum error correction techniques introduce a significant additional overhead that potentially negates certain quantum speedups for finance<sup>28</sup>. Thus, continued research into improving error correction and quantum architecture is also critical in realizing commercial applications of quantum computing in finance.

We hope that this Review has highlighted that in addition to challenges in hardware technologies there are still a lot of interesting quantum-algorithmic challenges to overcome to bring about a real-world quantum advantage. As quantum hardware advances, we expect to be able to benchmark more complex quantum algorithms, especially heuristic ones, on interesting problems.

Published online: 11 July 2023

## References

- Shreve, S. E. *Stochastic Calculus for Finance I* (Springer New York, 2004).
- Alexeev, Y. et al. Quantum computer systems for scientific discovery. *PRX Quantum* **2**, 017001 (2021).
- Glasserman, P. *Monte Carlo Methods in Financial Engineering*, Vol. 53 (Springer, 2004).
- Egger, D. J. et al. Quantum computing for finance: state-of-the-art and future prospects. *IEEE Trans. Quantum Eng.* **1**, 3101724 (2020).
- Orus, R., Mugel, S. & Lizaso, E. Quantum computing for finance: overview and prospects. *Rev. Phys.* **4**, 100028 (2019).
- Bouland, A., van Dam, W., Joorati, H., Kerenidis, I. & Prakash, A. Prospects and challenges of quantum finance. Preprint at <https://doi.org/10.21203/rs.3.rs-201106492> (2020).
- Pistoia, M. et al. Quantum machine learning for finance. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)* (IEEE/ACM, 2021).
- Gómez, A. et al. A survey on quantum computational finance for derivatives pricing and VaR. in *Archives of Computational Methods in Engineering* 1–27 (Springer, 2022).
- Griffin, P. & Sampat, R. Quantum computing for supply chain finance. In *2021 IEEE International Conference on Services Computing (SCC)* 456–459 (IEEE, 2021).
- Ganapathy, A. Quantum computing in high frequency trading and fraud detection. *Eng. Int.* **9**, 61–72 (2021).
- Wang, M., Pan, Y., Yang, X., Li, G. & Xu, Z. Tensor networks meet neural networks: a survey. Preprint at <https://doi.org/10.21203/rs.3.rs-230209019> (2023).
- Sengupta, R., Adhikary, S., Oseledets, I. & Biamonte, J. Tensor networks in machine learning. Preprint at <https://doi.org/10.21203/rs.3.rs-220702851> (2022).
- Patel, R. G. et al. Quantum-inspired tensor neural networks for option pricing. Preprint at <https://doi.org/10.21203/rs.3.rs-221214076> (2022).
- Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2010).



15. Kitaev, A. Y., Shen, A. H. & Vayalyi, M. N. *Classical and Quantum Computation* (American Mathematical Society, 2002).
16. Hull, J. C. *Options Futures and Other Derivatives* 11th edn (Pearson, 2021).
17. Wilmott, P. *Paul Wilmott on Quantitative Finance* 2nd edn (John Wiley & Sons, 2013).
18. Föllmer, H. & Schied, A. *Stochastic Finance: An Introduction in Discrete Time* (Walter de Gruyter, 2011).
19. Black, F. & Scholes, M. The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Vol. 1: Foundations of CCA and Equity Valuation* 3–21 (World Scientific, 2019).
20. Tse, W. M. *Closed-form Solutions for Fixed-strike Arithmetic Asian Options* (SSRN, 2018); <https://dx.doi.org/10.2139/ssrn.3176932>.
21. Kac, M. On distributions of certain Wiener functionals. *Trans. Am. Math. Soc.* **65**, 1–13 (1949).
22. Feynman, R. P. The principle of least action in quantum mechanics. In *Feynman's Thesis — A New Approach to Quantum Theory* 1–69 (World Scientific, 2005).
23. Heinrich, S. Quantum summation with an application to integration. *J. Complex* **18**, 1–50 (2002).
24. Brassard, G., Dupuis, F., Gambs, S. & Tapp, A. An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance. Preprint at <https://doi.org/10.1106.4267> (2011).
25. Montanaro, A. Quantum speedup of Monte Carlo methods. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **471**, 20150301 (2015).
26. Cornelissen, A. & Jerbi, S. Quantum algorithms for multivariate Monte Carlo estimation. Preprint at <https://doi.org/10.2107.03410> (2021).
27. Kothari, R. & O'Donnell, R. Mean estimation when you have the source code; or, quantum Monte Carlo methods. Preprint at <https://doi.org/10.2208.07544> (2022).
28. Babbush, R. et al. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum* **2**, 010103 (2021).
29. Huang, H.-Y., Bharti, K. & Rebentrost, P. Near-term quantum algorithms for linear systems of equations with regression loss functions. *New J. Phys.* **23**, 113021 (2021).
30. Suzuki, Y. et al. Amplitude estimation without phase estimation. *Quantum Inf. Process.* **19**, 1–17 (2020).
31. Grinko, D., Gacon, J., Zoufal, C. & Woerner, S. Iterative quantum amplitude estimation. *npj Quantum Inf.* **7**, 52 (2021).
32. Giurgica-Tiron, T., Kerenidis, I., Labib, F., Prakash, A. & Zeng, W. Low depth algorithms for quantum amplitude estimation. *Quantum* **6**, 745 (2022).
33. Grover, L. & Rudolph, T. Creating superpositions that correspond to efficiently integrable probability distributions. Preprint at <https://doi.org/10.208112> (2002).
34. Marin-Sanchez, G., Gonzalez-Conde, J. & Sanz, M. Quantum algorithms for approximate function loading. Preprint at <https://doi.org/10.2111.07933> (2021).
35. Herbert, S. No quantum speedup with Grover–Rudolph state preparation for quantum Monte Carlo integration. *Phys. Rev. E* **103**, 063302 (2021).
36. Grover, L. K. Synthesis of quantum superpositions by quantum computation. *Phys. Rev. Lett.* **85**, 1334–1337 (2000).
37. Sanders, Y. R., Low, G. H., Scherer, A. & Berry, D. W. Black-box quantum state preparation without arithmetic. *Phys. Rev. Lett.* **122**, 020502 (2019).
38. Wang, S. et al. Fast black-box quantum state preparation based on linear combination of unitaries. *Quantum Inf. Process.* **20**, 270 (2021).
39. Wang, S. et al. Inverse-coefficient black-box quantum state preparation. *New J. Phys.* **24**, 103004 (2022).
40. Bausch, J. Fast black-box quantum state preparation. *Quantum* **6**, 773 (2022).
41. McArdle, S., Gilyén, A. & Berta, M. Quantum state preparation without coherent arithmetic. Preprint at <https://doi.org/10.2210.14892> (2022).
42. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
43. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Inf.* **5**, 103 (2019).
44. Nakaji, K. et al. Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Phys. Rev. Res.* **4**, 023136 (2022).
45. Rattew, A. G., Sun, Y., Minssen, P. & Pistoia, M. The efficient preparation of normal distributions in quantum registers. *Quantum* **5**, 609 (2021).
46. Rattew, A. G. & Koczor, B. Preparing arbitrary continuous functions in quantum registers with logarithmic complexity. Preprint at <https://doi.org/10.2205.00519> (2022).
47. Zhang, X.-M., Yung, M.-H. & Yuan, X. Low-depth quantum state preparation. *Phys. Rev. Res.* **3**, 043200 (2021).
48. Möttönen, M., Vartiainen, J. J., Bergholm, V. & Salomaa, M. M. Transformation of quantum states using uniformly controlled rotations. *Quantum Info. Comput.* **5**, 467–473 (2005).
49. Araujo, I. F., Park, D. K., Petruccione, F. & da Silva, A. J. A divide-and-conquer algorithm for quantum state preparation. *Sci. Rep.* **11**, 6329 (2021).
50. Chakrabarti, S. et al. A threshold for quantum advantage in derivative pricing. *Quantum* **5**, 463 (2021).
51. Dupire, B. Pricing with a smile. *Risk* **7**, 18–20 (1994).
52. Duffie, D. & Glynn, P. Efficient Monte Carlo simulation of security prices. *Ann. Appl. Probab.* **5**, 897–905 (1995).
53. Giles, M. B. Multilevel Monte Carlo path simulation. *Oper. Res.* **56**, 607–617 (2008).
54. An, D. et al. Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. *Quantum* **5**, 481 (2021).
55. Ramos-Calderer, S. et al. Quantum unary approach to option pricing. *Phys. Rev. A* **103**, 032414 (2021).
56. Stamatopoulos, N. et al. Option pricing using quantum computers. *Quantum* **4**, 291 (2020).
57. Longstaff, F. A. & Schwartz, E. S. Valuing American options by simulation: a simple least-squares approach. *Rev. Financ. Stud.* **14**, 113–147 (2001).
58. Doriguello, J. A. F., Luongo, A., Bao, J., Rebentrost, P. & Santha, M. Quantum algorithm for stochastic optimal stopping problems with applications in finance. In *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, Vol. 232 of *Leibniz International Proceedings in Informatics (LIPIs)* (eds Le Gall, F. & Morimae, T.) 2:1–2:24 (Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022); <https://drops.dagstuhl.de/opus/volltexte/2022/16509>.
59. 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 51st Annual ACM SIGACT Symposium on Theory of Computing (ACM, 2019)*.
60. Stamatopoulos, N., Mazzola, G., Woerner, S. & Zeng, W. J. Towards quantum advantage in financial market risk using quantum gradient algorithms. *Quantum* **6**, 770 (2022).
61. Capriotti, L. *Fast Greeks by Algorithmic Differentiation* (SSRN, 2010); <https://dx.doi.org/10.2139/ssrn.1619626>.
62. Giles, M. & Glasserman, P. Smoking adjoints: fast Monte Carlo greeks. *Risk* **19**, 88–92 (2006).
63. Huge, B. & Savine, A. Differential machine learning. Preprint at <https://doi.org/10.2105.02347> (2020).
64. Wierichs, D., Izaac, J., Wang, C. & Lin, C. Y.-Y. General parameter-shift rules for quantum gradients. *Quantum* **6**, 677 (2022).
65. Woerner, S. & Egger, D. J. Quantum risk analysis. *npj Quantum Inf.* **5**, 15 (2019).
66. Egger, D. J., Gutierrez, R. G., Mestre, J. & Woerner, S. Credit risk analysis using quantum computers. *IEEE Trans. Comput.* **70**, 2136–2145 (2021).
67. Grossmann, C., Roos, H.-G. & Stynes, M. *Numerical Treatment of Partial Differential Equations* Vol. 154 (Springer, 2007).
68. Miyamoto, K. & Kubo, K. Pricing multi-asset derivatives by finite-difference method on a quantum computer. *IEEE Trans. Quantum Eng.* **3**, 3100225 (2022).
69. Broadie, M., Glasserman, P. & Kou, S. A continuity correction for discrete barrier options. *Math. Financ.* **7**, 325–349 (1997).
70. Linden, N., Montanaro, A. & Shao, C. Quantum vs. classical algorithms for solving the heat equation. In *Communications in Mathematical Physics* 1–41 (Springer, 2022).
71. Gonzalez-Conde, J., Rodriguez-Rozas, A., Solano, E. & Sanz, M. Simulating option price dynamics with exponential quantum speedup. Preprint at <https://doi.org/10.2101.04023> (2021).
72. Jin, S. & Liu, N. Quantum algorithms for computing observables of nonlinear partial differential equations. Preprint at <https://doi.org/10.2202.07834> (2022).
73. Jin, S., Liu, N. & Yu, Y. Quantum simulation of partial differential equations via schrodingerisation. Preprint at <https://doi.org/10.2212.13969> (2022).
74. Jin, S., Liu, N. & Yu, Y. Quantum simulation of partial differential equations via schrodingerisation: technical details. Preprint at <https://doi.org/10.2212.14703> (2022).
75. Fontanela, F., Jacquier, A. & Oumgari, M. A quantum algorithm for linear PDEs arising in finance. *SIAM J. Financ. Math.* **12**, SC98–SC114 (2021).
76. Alghassi, H. et al. A variational quantum algorithm for the Feynman–Kac formula. *Quantum* **6**, 730 (2022).
77. Kubo, K., Nakagawa, Y. O., Endo, S. & Nagayama, S. Variational quantum simulations of stochastic differential equations. *Phys. Rev. A* **103**, 052425 (2021).
78. Boyle, P. P. Option valuation using a three jump process. *Int. Opt. J.* **3**, 7–12 (1986).
79. Kyriienko, O., Paine, A. E. & Elfving, V. E. Solving nonlinear differential equations with differentiable quantum circuits. *Phys. Rev. A* **103**, 052416 (2021).
80. Schuld, M. & Petruccione, F. In *Quantum Models as Kernel Methods* 217–245 (Springer International Publishing, 2021); [https://doi.org/10.1007/978-3-030-83098-4\\_6](https://doi.org/10.1007/978-3-030-83098-4_6).
81. Jerbi, S. et al. Quantum machine learning beyond kernel methods. Preprint at <https://doi.org/10.2110.13162> (2021).
82. Nesterov, Y. Introductory lectures on convex programming volume I: basic course. *Lect. Notes* **3**, 5 (1998).
83. Nemirovski, A. Lectures on modern convex optimization. In *Society for Industrial and Applied Mathematics (SIAM) (Citeseer)*, 2001).
84. Panik, M. J. *Fundamentals of Convex Analysis* (Springer, 1993).
85. Cornuéjols, G., Peña, J. & Tütüncü, R. *Optimization Methods in Finance* 2nd edn (Cambridge Univ. Press, 2018).
86. Markowitz, H. Portfolio selection. *J. Finance* **7**, 77–91 (1952).
87. Chakraborty, S., Gilyén, A. & Jeffery, S. The power of block-encoded matrix powers: improved regression techniques via faster hamiltonian simulation. In *International Colloquium on Automata, Languages and Programming (ICALP, 2019)*.
88. Arora, S. & Kale, S. A combinatorial, primal-dual approach to semidefinite programs. *J. ACM* **63**, 12 (2016).
89. van Apeldoorn, J., Gilyén, A., Gribling, S. & de Wolf, R. Quantum SDP-solvers: better upper and lower bounds. *Quantum* **4**, 230 (2020).
90. van Apeldoorn, J. & Gilyén, A. Improvements in quantum SDP-solving with applications. In *ICALP (Schloss Dagstuhl — Leibniz-Zentrum fuer Informatik, 2019)*.
91. Arora, S., Hazan, E. & Kale, S. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.* **8**, 121–164 (2012).
92. Hazan, E. & Kale, S. An online portfolio selection algorithm with regret logarithmic in price variation. *Math. Financ.* **25**, 288–310 (2015).



93. Hazan, E. et al. Introduction to online convex optimization. *Found. Trends Optim.* **2**, 157–325 (2016).
94. Lim, D. & Rebentrost, P. A quantum online portfolio optimization algorithm. Preprint at <https://doi.org/10.2208.14749> (2022).
95. Grigoriadis, M. D. & Khachiyan, L. G. A sublinear-time randomized approximation algorithm for matrix games. *Oper. Res. Lett.* **18**, 53–58 (1995).
96. van Apeldoorn, J. & Gilyén, A. Quantum algorithms for zero-sum games. Preprint at <https://doi.org/10.1104.03180> (2019).
97. Boulund, A., Getachew, Y., Jin, Y., Sidford, A. & Tian, K. Quantum speedups for zero-sum games via improved dynamic Gibbs sampling. Preprint at <https://doi.org/10.1104.03763> (2023).
98. Rebentrost, P., Luongo, A., Bosch, S. & Lloyd, S. Quantum computational finance: martingale asset pricing for incomplete markets. Preprint at <https://doi.org/10.1104.08867> (2022).
99. Li, T., Wang, C., Chakrabarti, S. & Wu, X. Sublinear classical and quantum algorithms for general matrix games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, 8465–8473 (AAAI, 2021).
100. Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Found. Trends Machine Learn.* **4**, 107–194 (2012).
101. Li, T., Chakrabarti, S. & Wu, X. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning* 3815–3824 (PMLR, 2019).
102. Lee, Y. T., Sidford, A. & Wong, S. C.-W. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* 1049–1065 (IEEE, 2015).
103. Cohen, M. B., Lee, Y. T. & Song, Z. Solving linear programs in the current matrix multiplication time. *J. ACM* <https://doi.org/10.1145/3424305> (2021).
104. Monteiro, R. & Tsuchiya, T. Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. *Math. Program.* **88**, 61–83 (2000).
105. Kerenidis, I. & Prakash, A. A quantum interior point method for LPs and SDPs. *ACM Trans. Quantum Comput.* **1**, 5 (2020).
106. Kerenidis, I., Prakash, A. & Szilágyi, D. Quantum algorithms for second-order cone programming and support vector machines. *Quantum* **5**, 427 (2021).
107. Augustino, B., Nannicini, G., Terlaky, T. & Zuluaga, L. F. Quantum interior point methods for semidefinite optimization. Preprint at <https://doi.org/10.1104.06025> (2021).
108. Dalzell, A. M. et al. End-to-end resource analysis for quantum interior point methods and portfolio optimization. Preprint at <https://doi.org/10.1104.12489> (2022).
109. Nannicini, G. Fast quantum subroutines for the simplex method. In *Integer Programming and Combinatorial Optimization* (eds Singh, M. & Williamson, D. P.) 311–325 (Springer International Publishing, 2021).
110. Rebentrost, P. & Lloyd, S. Quantum computational finance: quantum algorithm for portfolio optimization. Preprint at <https://doi.org/10.1104.03975> (2018).
111. Yalovetzky, R., Minssen, P., Herman, D. & Pistoia, M. NISQ-HHL: portfolio optimization for near-term quantum hardware. Preprint at <https://doi.org/10.1104.15958> (2021).
112. Moehle, N., Kochenderfer, M. J., Boyd, S. & Ang, A. Tax-aware portfolio construction via convex optimization. *J. Optim. Theory Appl.* **189**, 364–383 (2021).
113. Bertsimas, D. & Cory-Wright, R. A scalable algorithm for sparse portfolio selection. *INFORMS J. Comput.* **34**, 1305–1840 (2022).
114. Liu, Y., Su, W. J. & Li, T. On quantum speedups for nonconvex optimization via quantum tunneling walks. Preprint at <https://doi.org/10.1104.14501> (2022).
115. Leng, J., Hickman, E., Li, J. & Wu, X. Quantum hamiltonian descent. Preprint at <https://doi.org/10.1104.01471> (2023).
116. Wolsey, L. A. & Nemhauser, G. L. *Integer and Combinatorial Optimization*, Vol. 55 (John Wiley & Sons, 1999).
117. Morrison, D. R., Jacobson, S. H., Sauppe, J. J. & Sewell, E. C. Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning. *Discrete Optim.* **19**, 79–102 (2016).
118. Apers, S., Gilyén, A. & Jeffery, S. A unified framework of quantum walk search. In *Symposium on Theoretical Aspects of Computer Science* (Dagstuhl Publishing, 2021).
119. Apers, S. & Sarlette, A. Quantum fast-forwarding: Markov chains and graph property testing. Preprint at <https://doi.org/10.1104.02321> (2018).
120. Montanaro, A. Quantum speedup of branch-and-bound algorithms. *Phys. Rev. Res.* **2**, 013056 (2020).
121. Ambainis, A. & Kokainis, M. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, 989–1002 (Association for Computing Machinery, 2017); <https://doi.org/10.1145/3055399.3055444>.
122. Chakrabarti, S., Minssen, P., Yalovetzky, R. & Pistoia, M. Universal quantum speedup for branch-and-bound, branch-and-cut, and tree-search algorithms. Preprint at <https://doi.org/10.1104.03210> (2022).
123. Romeo, F. & Sangiovanni-Vincentelli, A. A theoretical framework for simulated annealing. *Algorithmica* **6**, 302–345 (1991).
124. Levin, D. A. & Peres, Y. *Markov Chains and Mixing Times*, Vol. 107 (American Mathematical Society, 2017).
125. Somma, R. D., Boixo, S., Barnum, H. & Knill, E. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.* **101**, 130504 (2008).
126. Wocjan, P. & Abeyesinghe, A. Speedup via quantum sampling. *Phys. Rev. A* **78**, 042336 (2008).
127. Harrow, A. W. & Wei, A. Y. Adaptive quantum simulated annealing for Bayesian inference and estimating partition functions. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* 193–212 (Society for Industrial and Applied Mathematics, 2020); <https://doi.org/10.1137/1.9781611975994.12>.
128. Montanaro, A. & Pallister, S. Quantum algorithms and the finite element method. *Phys. Rev. A* **93**, 032324 (2016).
129. Henderson, D., Jacobson, S. H. & Johnson, A. W. *The Theory and Practice of Simulated Annealing* 287–319 (Springer US, 2003); [https://doi.org/10.1007/0-306-48056-5\\_10](https://doi.org/10.1007/0-306-48056-5_10).
130. Li, Y., Protopopescu, V. A., Arnold, N., Zhang, X. & Gorin, A. Hybrid parallel tempering and simulated annealing method. *Appl. Math. Comput.* **212**, 216–228 (2009).
131. Lemieux, J., Heim, B., Poulin, D., Svore, K. & Troyer, M. Efficient quantum Walk circuits for Metropolis–Hastings algorithm. *Quantum* **4**, 287 (2020).
132. Dürr, C. & Høyer, P. A quantum algorithm for finding the minimum. Preprint at <https://doi.org/10.1104.9607014> (1996).
133. Bulger, D., Baritomp, W. P. & Wood, G. R. Implementing pure adaptive search with Grover’s quantum algorithm. *J. Optim. Theory Appl.* **116**, 517–529 (2003).
134. Gilliam, A., Woerner, S. & Goniçulea, C. Grover adaptive search for constrained polynomial binary optimization. *Quantum* **5**, 428 (2021).
135. Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’96 212–219 (Association for Computing Machinery, 1996).
136. Ambainis, A. Quantum walk algorithm for element distinctness. *SIAM J. Comput.* **37**, 210–239 (2007).
137. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
138. Hastings, M. B. A short path quantum algorithm for exact optimization. *Quantum* **2**, 78 (2018).
139. Dalzell, A. M., Pancotti, N., Campbell, E. T. & Brandão, F. G. S. L. Mind the gap: achieving a super-Grover quantum speedup by jumping to the end. Preprint at <https://doi.org/10.1104.01513> (2022).
140. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
141. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**, 5355–5363 (1998).
142. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. Preprint at <https://doi.org/10.1104.0001106> (2000).
143. Hegade, N. N. et al. Shortcuts to adiabaticity in digitized adiabatic quantum computing. *Phys. Rev. Appl.* **15**, 024038 (2021).
144. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at <https://doi.org/10.1104.1411.0028> (2014).
145. Hadfield, S. et al. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34 (2019).
146. Peruzzo, A. et al. A variational eigensolver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
147. Liu, X. et al. Layer VQE: a variational approach for combinatorial optimization on noisy quantum computers. *IEEE Trans. Quantum Eng.* **3**, 3100920 (2022).
148. Yuan, X., Endo, S., Zhao, Q., Li, Y. & Benjamin, S. C. Theory of variational quantum simulation. *Quantum* **3**, 191 (2019).
149. McArdle, S. et al. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Inf.* **5**, 75 (2019).
150. Benedetti, M., Fiorentini, M. & Lubasch, M. Hardware-efficient variational quantum algorithms for time evolution. *Phys. Rev. Res.* **3**, 033083 (2021).
151. Bermejo, P. & Orús, R. Variational quantum continuous optimization: a cornerstone of quantum mathematical analysis. Preprint at <https://doi.org/10.1104.03136> (2022).
152. Fernández-Lorenzo, S., Porras, D. & García-Ripoll, J. J. Hybrid quantum–classical optimization with cardinality constraints and applications to finance. *Quantum Sci. Technol.* **6**, 034010 (2021).
153. Niroula, P. et al. Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer. *Sci. Rep.* **12**, 17171 (2022).
154. Herman, D. et al. Portfolio optimization via quantum Zeno dynamics on a quantum processor. Preprint at <https://doi.org/10.1104.2209.15024> (2022).
155. Tangpanitanon, J. et al. Hybrid quantum-classical algorithms for loan collection optimization with loan loss provisions. Preprint at <https://doi.org/10.1104.15870> (2021).
156. Drieb-Schön, M., Ender, K., Javanmard, Y. & Lechner, W. Parity quantum optimization: encoding constraints. *Quantum* **7**, 951 (2023).
157. Bittel, L. & Kliesch, M. Training variational quantum algorithms is NP-hard. *Phys. Rev. Lett.* **127**, 120502 (2021).
158. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).
159. Larocca, M. et al. Diagnosing barren plateaus with tools from quantum optimal control. *Quantum* **6**, 824 (2022).
160. You, X., Chakrabarti, S. & Wu, X. A convergence theory for over-parameterized variational quantum eigensolvers. In *26th Conference on Quantum Information Processing* (TQC, 2023).
161. Denchev, V. S. et al. What is the computational value of finite-range tunneling? *Phys. Rev. X* **6**, 031015 (2016).
162. Mugel, S. et al. Hybrid quantum investment optimization with minimal holding period. *Sci. Rep.* **11**, 19587 (2021).

163. Mugel, S. et al. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Phys. Rev. Res.* **4**, 013006 (2022).
164. Palmer, S. et al. Financial index tracking via quantum computing with cardinality constraints. Preprint at <https://doi.org/arXiv:2208.11380> (2022).
165. Orús, R., Mugel, S. & Lizaso, E. Forecasting financial crashes with quantum computing. *Phys. Rev. A* **99**, 060301 (2019).
166. Akshay, V., Rabinovich, D., Campos, E. & Biamonte, J. Parameter concentrations in quantum approximate optimization. *Phys. Rev. A* **104**, L010401 (2021).
167. Basso, J., Farhi, E., Marwaha, K., Villalonga, B. & Zhou, L. *The Quantum Approximate Optimization Algorithm at High Depth for Maxcut on Large-girth Regular Graphs and the Sherrington–Kirkpatrick Model* (Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022); <https://drops.dagstuhl.de/opus/volltexte/2022/16514/>.
168. Sureshbabu, S. H. et al. Parameter setting in quantum approximate optimization of weighted problems. Preprint at <https://doi.org/arXiv:2305.15201> (2023).
169. Lykov, D. et al. Performance evaluation and acceleration of the QTensor quantum circuit simulator on GPUs. In 2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS) (IEEE, 2021); <https://doi.org/10.1109/qcs54837.2021.00007>.
170. Lykov, D. & Alexeev, Y. Importance of diagonal gates in tensor network simulations. Preprint at <https://doi.org/arXiv:2106.15740> (2021).
171. Lykov, D., Schutski, R., Galda, A., Vinokur, V. & Alexeev, Y. Tensor network quantum simulator with step-dependent parallelization. Preprint at <https://doi.org/arXiv:2012.02430> (2020).
172. Lykov, D. et al. Performance evaluation and acceleration of the Q Tensor quantum circuit simulator on GPUs. 2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS) (IEEE/ACM, 2021); <https://doi.org/10.1109/qcs54837.2021.00007>.
173. He, Z. et al. Alignment between initial state and mixer improves QAOA performance for constrained portfolio optimization. Preprint at <https://doi.org/arXiv:2305.03857> (2023).
174. Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. *Introduction to Algorithms* (MIT Press, 2009).
175. Ambainis, A. et al. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* 1783–1793 (ACM-SIAM, 2019); <https://epubs.siam.org/doi/abs/10.1137/1.9781611975482.107>.
176. Mohri, M., Rostamizadeh, A. & Talwalkar, A. *Foundations of Machine Learning* (MIT Press, 2018).
177. Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* Vol. 2 (Springer, 2009).
178. Herman, D. et al. A survey of quantum computing for finance. Preprint at <https://doi.org/arXiv:2201.02773> (2022).
179. Giovannetti, V., Lloyd, S. & Maccone, L. Quantum random access memory. *Phys. Rev. Lett.* **100**, 160501 (2008).
180. Aaronson, S. Read the fine print. *Nat. Phys.* **11**, 291–293 (2015).
181. Tang, E. Dequantizing algorithms to understand quantum advantage in machine learning. *Nat. Rev. Phys.* **4**, 692–693 (2022).
182. Huang, H.-Y. et al. Power of data in quantum machine learning. *Nat. Commun.* **12**, 2631 (2021).
183. Slattery, L. et al. Numerical evidence against advantage with quantum fidelity kernels on classical data. Preprint at <https://doi.org/arXiv:2211.16551> (2022).
184. Gu, S., Kelly, B. & Xiu, D. Empirical asset pricing via machine learning. *Rev. Financ. Stud.* **33**, 2223–2273 (2020).
185. Ghysels, E., Santa-Clara, P. & Valkanov, R. Predicting volatility: getting the most out of return data sampled at different frequencies. *J. Econom.* **131**, 59–95 (2006).
186. Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **109**, 050505 (2012).
187. Wang, G. Quantum algorithm for linear regression. *Phys. Rev. A* **96**, 012335 (2017).
188. Kerenidis, I. & Prakash, A. Quantum gradient descent for linear systems and least squares. *Phys. Rev. A* **101** (2020).
189. Date, P. & Potok, T. Adiabatic quantum linear regression. *Sci. Rep.* **11**, 21905 (2021).
190. Han, J., Zhang, X.-P. & Wang, F. Gaussian process regression stochastic volatility model for financial time series. *IEEE J. Sel. Top. Signal Process.* **10**, 1015–1028 (2016).
191. Zhao, Z., Fitzsimons, J. K. & Fitzsimons, J. F. Quantum-assisted Gaussian process regression. *Phys. Rev. A* **99**, 052331 (2019).
192. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
193. Abdou, H. A., Tsafack, M. D. D., Ntim, C. G. & Baker, R. D. Predicting creditworthiness in retail banking with limited scoring data. *Knowl. Based Syst.* **103**, 89–103 (2016).
194. Awoyemi, J. O., Adetunmbi, A. O. & Oluwadare, S. A. Credit card fraud detection using machine learning techniques: a comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCN)* 1–9 (ICCN, 2017).
195. Li, T., Chakrabarti, S. & Wu, X. *Sublinear Quantum Algorithms for Training Linear and Kernel-Based Classifiers* (ICML, 2019).
196. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **113**, 130503 (2014).
197. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
198. Wiebe, N., Kapoor, A. & Svore, K. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Inf. Comput.* **15**, 316–356 (2015).
199. Ruan, Y., Xue, X., Liu, H., Tan, J. & Li, X. Quantum algorithm for K-nearest neighbors classification based on the metric of Hamming distance. *Int. J. Theor. Phys.* **56**, 3496–3507 (2017).
200. Basheer, A., Afham, A. & Goyal, S. K. Quantum k-nearest neighbors algorithm. Preprint at <https://doi.org/arXiv:2003.09187> (2020).
201. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at <https://doi.org/arXiv:1802.06002> (2018).
202. Killoran, N. et al. Continuous-variable quantum neural networks. *Phys. Rev. Res.* **1**, 033063 (2019).
203. Henderson, M., Shakya, S., Pradhan, S. & Cook, T. Quantum convolutional neural networks: powering image recognition with quantum circuits. *Quantum Mach. Intell.* **2**, 2 (2020).
204. Allcock, J., Hsieh, C.-Y., Kerenidis, I. & Zhang, S. Quantum algorithms for feedforward neural networks. *ACM Trans. Quantum Comput.* **1**, 6 (2020).
205. Kerenidis, I., Landman, J. & Mathur, N. Classical and quantum algorithms for orthogonal neural networks. Preprint at <https://doi.org/arXiv:2106.07198> (2021).
206. Schapire, R. E. & Freund, Y. Boosting: foundations and algorithms. *Kybernetes* **42**, 164–166 (2013).
207. Schapire, R. E. The strength of weak learnability. *Machine Learn.* **5**, 197–227 (1990).
208. Arunachalam, S. & Maity, R. Quantum boosting. In *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research* (eds Daumé, H. III & Singh, A.) 377–387 (PMLR, 2020); <https://proceedings.mlr.press/v119/arunachalam20a.html>.
209. Izdebski, A. & de Wolf, R. Improved quantum boosting. Preprint at <https://doi.org/arXiv:2009.08360> (2020).
210. Servedio, R. A. Smooth boosting and learning with malicious noise. *J. Machine Learn. Res.* **4**, 633–648 (2003).
211. Neven, H., Denchev, V. S., Rose, G. & Marendy, W. G. Qboost: large scale classifier training with adiabatic quantum optimization. In *Asian Conference on Machine Learning* 333–348 (PMLR, 2012).
212. Leclerc, L. et al. Financial risk management on a neutral atom quantum processor. Preprint at <https://doi.org/arXiv:2212.03223> (2022).
213. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
214. Chen, T. & Guestrin, C. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016); <https://doi.org/10.1145/2939672.2939785>.
215. Davis, J., Devos, L., Reyniers, S. & Schoutens, W. Gradient boosting for quantitative finance. *J. Comput. Finance* **24**, 1–40 (2020).
216. Carmona, P., Climent, F. & Mompalao, A. Predicting failure in the U.S. banking sector: an extreme gradient boosting approach. *Int. Rev. Econ. Finance* **61**, 304–323 (2019).
217. Chang, Y.-C., Chang, K.-H. & Wu, G.-J. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Appl. Soft Comput.* **73**, 914–920 (2018).
218. Gavrishchaka, V. V. Boosting-based frameworks in financial modeling: application to symbolic volatility forecasting. In *Econometric Analysis of Financial and Economic Time Series* (Emerald Group Publishing Limited, 2006).
219. León, D. et al. Clustering algorithms for risk-adjusted portfolio construction. *Proc. Computer Sci.* **108**, 1334–1343 (2017).
220. Tola, V., Lillo, F., Gallegati, M. & Mantegna, R. N. Cluster analysis for portfolio optimization. *J. Econ. Dyn. Control* **32**, 235–258 (2008).
221. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. Preprint at <https://doi.org/arXiv:1307.0411> (2013).
222. Kerenidis, I., Landman, J., Luongo, A. & Prakash, A. q-Means: a quantum algorithm for unsupervised machine learning. Preprint at <https://doi.org/arXiv:1812.03584> (2018).
223. Khan, S. U., Awan, A. J. & Vall-Llosera, G. K-means clustering on noisy intermediate scale quantum computers. Preprint at <https://doi.org/arXiv:1909.12183> (2019).
224. Kerenidis, I., Luongo, A. & Prakash, A. Quantum expectation-maximization for Gaussian mixture models. In *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research* (eds Daumé, H. III & Singh, A.) 5187–5197 (PMLR, 2020); <https://proceedings.mlr.press/v119/kerenidis20a.html>.
225. Miyahara, H., Aihara, K. & Lechner, W. Quantum expectation-maximization algorithm. *Phys. Rev. A* **101**, 012326 (2020).
226. Ng, A. Y., Jordan, M. I. & Weiss, Y. On spectral clustering: analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* 849–856 (ACM, 2001).
227. Daskin, A. Quantum spectral clustering through a biased phase estimation algorithm. *TWMS J. Appl. Eng. Math.* **10**, 24–33 (2017).
228. Kerenidis, I. & Landman, J. Quantum spectral clustering. *Phys. Rev. A* **103**, 042415 (2021).
229. Apers, S. & de Wolf, R. Quantum speedup for graph sparsification, cut approximation and Laplacian solving. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)* 637–648 (IEEE, 2020).
230. Aïmeur, E., Brassard, G. & Gambs, S. Quantum clustering algorithms. In *Proceedings of the 24th International Conference on Machine Learning* (ACM, 2007); <https://icml.cc/impls/conferences/2007/proceedings/papers/518.pdf>.
231. Aïmeur, E., Brassard, G. & Gambs, S. Quantum speed-up for unsupervised learning. *Mach. Learn.* **20**, 261–287 (2013).

232. Kumar, V., Bass, G., Tomlin, C. & Dulny, J. Quantum annealing for combinatorial clustering. *Quantum Inf. Process.* **17**, 39 (2018).
233. Bermejo, P. & Orús, R. Variational quantum and quantum-inspired clustering. Preprint at <https://arxiv.org/abs/2206.09893> (2022).
234. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
235. Benedetti, M. et al. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf.* **5**, 45 (2019).
236. Coyle, B. et al. Quantum versus classical generative modelling in finance. *Quantum Sci. Technol.* **6**, 024013 (2021).
237. Koller, D. & Friedman, N. *Probabilistic Graphical Models: Principles and Techniques* (The MIT Press, 2009).
238. Zhu, E. Y. et al. Generative quantum learning of joint probability distribution functions. Preprint at <https://doi.org/ArXiv:2109.06315> (2021).
239. Kyriienko, O., Paine, A. E. & Elfving, V. E. Protocols for trainable and differentiable quantum generative modelling. Preprint at <https://doi.org/ArXiv:2202.08253> (2022).
240. Low, G. H., Yoder, T. J. & Chuang, I. L. Quantum inference on Bayesian networks. *Phys. Rev. A* **89**, 062315 (2014).
241. Tucci, R. R. Quantum Bayesian nets. *Int. J. Mod. Phys. B* **09**, 295–337 (1995).
242. Borujeni, S. E., Nannapaneni, S., Nguyen, N. H., Behrman, E. C. & Steck, J. E. Quantum circuit representation of Bayesian networks. *Exp. Syst. Appl.* **176**, 114768 (2021).
243. Klepac, G. The Schrödinger equation as inspiration for a client portfolio simulation hybrid system based on dynamic Bayesian networks and the REFI model. In *Quantum Inspired Computational Intelligence* Ch. 12, 391–416 (Morgan Kaufmann, 2017).
244. Moreira, C. & Wichert, A. Quantum-like Bayesian networks for modeling decision making. *Front. Psychol.* **7**, 11 (2016).
245. Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **14**, 1771–1800 (2002).
246. Salakhutdinov, R. & Hinton, G. Deep Boltzmann machines. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, Vol. 5, 448–455 (PMLR, 2009).
247. Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning. *Phys. Rev. A* **94**, 022308 (2016).
248. Dixit, V., Selvarajan, R., Alam, M. A., Humble, T. S. & Kais, S. Training restricted Boltzmann machines with a d-wave quantum annealer. *Front. Phys.* **9**, 589626 (2021).
249. Amin, M. H., Andriyash, E., Rolfe, J., Kulchitsky, B. & Melko, R. Quantum Boltzmann machine. *Phys. Rev. X* **8**, 021050 (2018).
250. Zoufal, C., Lucchi, A. & Woerner, S. Variational quantum Boltzmann machines. *Quantum Mach. Intell.* **3**, 7 (2021).
251. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**, 040502 (2018).
252. Guyon, I., Gunn, S., Nikravesh, M. & Zadeh, L. A. *Feature Extraction: Foundations and Applications* Vol. 207 (Springer, 2008).
253. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014).
254. Yu, C., Gao, F. & Lin, S. Quantum data compression by principal component analysis. *Quantum Inf. Process.* **18**, 249 (2019).
255. Lin, J., Bao, W.-S., Zhang, S., Li, T. & Wang, X. An improved quantum principal component analysis algorithm based on the quantum singular threshold method. *Phys. Lett. A* **383**, 2862–2868 (2019).
256. Martin, A. et al. Toward pricing financial derivatives with an IBM quantum computer. *Phys. Rev. Res.* **3**, 013167 (2021).
257. Carlsson, G. Topological methods for data modelling. *Nat. Rev. Phys.* **2**, 697–708 (2020).
258. Lloyd, S., Garnerone, S. & Zanardi, P. Quantum algorithms for topological and geometric analysis of data. *Nat. Commun.* **7**, 10138 (2016).
259. McArdle, S., Gilyén, A. & Berta, M. A streamlined quantum algorithm for topological data analysis with exponentially fewer qubits. Preprint at <https://doi.org/ArXiv:2209.12887> (2022).
260. Grossi, M. et al. Mixed quantum-classical method for fraud detection with quantum feature selection. *IEEE Trans. Quantum Eng.* **3**, 3102812 (2022).
261. Dacrema, M. F. et al. Towards feature selection for ranking and classification exploiting quantum annealers. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 2022).
262. Zoufal, C. et al. Variational quantum algorithm for unconstrained black box binary optimization: application to feature selection. *Quantum* **7**, 909 (2023).
263. Mücke, S., Heese, R., Müller, S., Wolter, M. & Piatkowski, N. Feature selection on quantum computers. *Quantum Mach. Intell.* **5**, 11 (2023).
264. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
265. Halperin, I. QLBS: Q-learner in the Black-Scholes-Merton worlds. *J. Deriv.* **28**, 99–122 (2020).
266. Buehler, H., Gonon, L., Teichmann, J. & Wood, B. Deep hedging. *Quant. Finance* **19**, 1271–1291 (2019).
267. Benhamou, E., Saltiel, D., Ohana, J. J., Atif, J. & Laraki, R. Deep reinforcement learning (DRL) for portfolio allocation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 527–531 (Springer, 2020).
268. Deng, Y., Bao, F., Kong, Y., Ren, Z. & Dai, Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 653–664 (2016).
269. Zhang, Z., Zohren, S. & Roberts, S. Deep reinforcement learning for trading. *J. Financ. Data Sci.* **2**, 25–40 (2020).
270. Spooner, T., Fearnley, J., Savani, R. & Koukorinis, A. Market making via reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* 434–442 (International Foundation for Autonomous Agents and Multiagent Systems, 2018).
271. Abe, N. et al. Optimizing debt collections using constrained reinforcement learning. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 75–84 (ACM SIGKDD, 2010).
272. Kolm, P. N. & Ritter, G. Modern perspectives on reinforcement learning in finance (September 6, 2019). *J. Mach. Learn. Finance* <https://dx.doi.org/10.2139/ssrn.3449401> (2020).
273. Dong, D., Chen, C., Li, H. & Tarn, T.-J. Quantum reinforcement learning. *IEEE Trans. Systems Man Cybern. Part B (Cybern.)* **38**, 1207–1220 (2008).
274. Cornelissen, A. *Quantum Gradient Estimation and Its Application to Quantum Reinforcement Learning*. Master's thesis, Delft University of Technology (2018).
275. Paparo, G. D., Dunjko, V., Makmal, A., Martin-Delgado, M. A. & Briegel, H. J. Quantum speedup for active learning agents. *Phys. Rev. X* **4**, 031002 (2014).
276. Chen, S. Y.-C. et al. Variational quantum circuits for deep reinforcement learning. *IEEE Access* **8**, 141007–141024 (2020).
277. Chen, S. Y.-C., Huang, C.-M., Hsing, C.-W., Goan, H.-S. & Kao, Y.-J. Variational quantum reinforcement learning via evolutionary optimization. *Mach. Learn. Sci. Tech.* **3**, 015025 (2021).
278. Lockwood, O. & Si, M. Reinforcement learning with quantum variational circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* Vol. 16, 245–251 (AAAI Press, 2020); <https://doi.org/10.1609/aiide.v16i1.7437>
279. Jerbi, S., Trenkwalder, L. M., Poulsen Nautrup, H., Briegel, H. J. & Dunjko, V. Quantum enhancements for deep reinforcement learning in large spaces. *PRX Quantum* **2**, 010328 (2021).
280. Jerbi, S., Gyurik, C., Marshall, S., Briegel, H. J. & Dunjko, V. Parametrized quantum policies for reinforcement learning. *Adv. Neural Inf. Process. Syst.* **34**, 28362–28375 (2021).
281. Crawford, D., Levit, A., Ghadermarzy, N., Oberoi, J. S. & Ronagh, P. Reinforcement learning using quantum Boltzmann machines. *Quantum Inf. Comput.* **18**, 51–74 (2018).
282. Cherrat, E. A. et al. Quantum deep hedging. Preprint at <https://doi.org/ArXiv:2303.16585> (2023).
283. Tang, E. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, 2019).
284. Chia, N.-H. et al. 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* (2020).
285. Arrazola, J. M., Delgado, A., Bardhan, B. R. & Lloyd, S. Quantum-inspired algorithms in practice. *Quantum* **4**, 307 (2020).
286. Isenhowe, L., Saffman, M. & Mølmer, K. Multibit  $C_{\text{NOT}}$  quantum gates via Rydberg blockade. *Quantum Inf. Process.* **10**, 755–770 (2011).
287. Goel, N. & Freericks, J. K. Native multiqubit Toffoli gates on ion trap quantum computers. *Quantum Sci. Technol.* **6**, 044010 (2021).
288. Roy, T. et al. Programmable superconducting processor with native three-qubit gates. *Phys. Rev. Appl.* **14**, 014072 (2020).
289. Haener, T., Soeken, M., Roetteler, M. & Svore, K. M. Quantum circuits for floating-point arithmetic. In *Reversible Computation: 10th International Conference, RC 2018* 162–174 (Springer, 2018).
290. Jaques, S. & Rattew, A. G. QRAM: a survey and critique. Preprint at <https://doi.org/ArXiv:2305.10310> (2023).

## Acknowledgements

The authors appreciate support from the Chicago Quantum Exchange. Y.A. acknowledges support from the Office of Science, US Department of Energy, under contract DE-AC02-06CH11357 at Argonne National Laboratory. D.H., Y.S. and M.P. appreciate the insightful discussions they had with their colleagues from the Global Technology Applied Research centre at JPMC.

## Author contributions

D.H., C.G., X.L. and Y.S. contributed equally to all aspects of the article. All authors contributed to the writing of the manuscript.

## Competing interests

The author declare no competing interests.

## Additional information

**Peer review information** *Nature Reviews Physics* thanks the anonymous reviewers for their contribution to the peer review of this work.

**Disclaimer** This paper was prepared for informational purposes with contributions from the Global Technology Applied Research Center of JPMorgan Chase & Co. This paper is

---

not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accepts any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy or reliability of the information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© Springer Nature Limited 2023