NISQ quantum computing: Why it persists and what can we do with it?

Shaik Asaaduddin Khwaja
(Supervised by Professor Mike Payne)

April 2024

Abstract

Quantum computing has been gaining popularity over the past few years because of its potential to outperform classical computers. Despite significant theoretical progress over past decades, physical implementations are still unable to implement the algorithms theorized to bring about blockbuster speedups over classical computers. The field is currently said to be in the *noisy intermediate quantum* (NISQ) era - where quantum computers are noisy and cannot be scaled up significantly. While work carries on towards moving the field into the *fault-tolerant* era by overcoming these challenges, there is scepticism about the utility of quantum computers. In this review, we provide an overview of quantum computing before discussing why the NISQ era persists, and what the most promising applications of NISQ quantum computers are.

1 Introduction

With over $2.35 billion invested into quantum technology start-ups in 2022, quantum computing has come a long way since its value in simulation was proposed by Richard Feynman in 1981 [1, 2]. Theoretical breakthroughs over the last 40 years like Shor's factoring algorithm have led to immense excitement about its potential to outperform classical computers and thus provide a *quantum speedup* [3].

Understanding for which applications such speedups exist, if they do, is still an area of active research [4] complicated by environmental noise-induced errors [5]. Theoretically, error rates can reduced via *quantum error correction* (QEC) [6] but current quantum computers (QCs) are too small to practically implement this [7]. These devices, limited in their qubit count ($\sim 10^2$ qubits) and still significantly prone to errors, characterize the *Noisy Intermediate-Scale Quantum* (NISQ) era [5].

The ultimate goal is to build *fault-tolerant* QCs, that possess high qubit counts ($\sim 10^6$ qubits), low error rates, and can run the algorithms theorized to achieve practical speedups [7, 8]. Despite growing commercial interest [1], the chasm between NISQ and fault-tolerant QCs has resulted in scepticism about its utility [9–11]. To understand how significant the limitations of NISQ QCs are, existing literature [7, 8, 12–15] is reviewed to summarise its most promising applications. The difficulties of reaching fault-tolerant computation are also briefly discussed.

2 Overview of Quantum Computing

2.1 Theoretical Background

Classical computers work by storing information that is encoded into classical bits, which are manipulated by logical gates [16]. When connected by wires to allow information flow, computation can be performed.

Analogous to these, are quantum bits (qubits), gates, and circuits [16]. Qubits are two-level quantum systems that exist in 2D Hilbert space [17]. They are expressed using the basis vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

which forms the computational basis such that a general qubit state is expressed as [17]

$$\begin{gathered} |\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad \text{where} \quad \alpha, \beta \in \mathbb{C}, \\ \text{Normalisation} : |\alpha|^2 + |\beta|^2 = 1 \end{gathered} \tag{2}$$

The familiar fundamental postulates of quantum mechanics apply here [16]. A qubit can be altered using quantum gates where single-qubit gates, that act on a single qubit, can be described by a 2×2 matrix U with the restriction that U is unitary [16]. In general, an n qubit quantum circuit consists of gates (see Figure 1) being applied to evolve a well-described multi-qubit initial state (e.g. $|0\rangle^{\otimes n} = |0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle$) before the expectation value of each qubit is measured in the computational basis to obtain the result of the computation [17]. The initial state is repeatedly prepared and measured, with the average of all measurements quoted as the result [17].

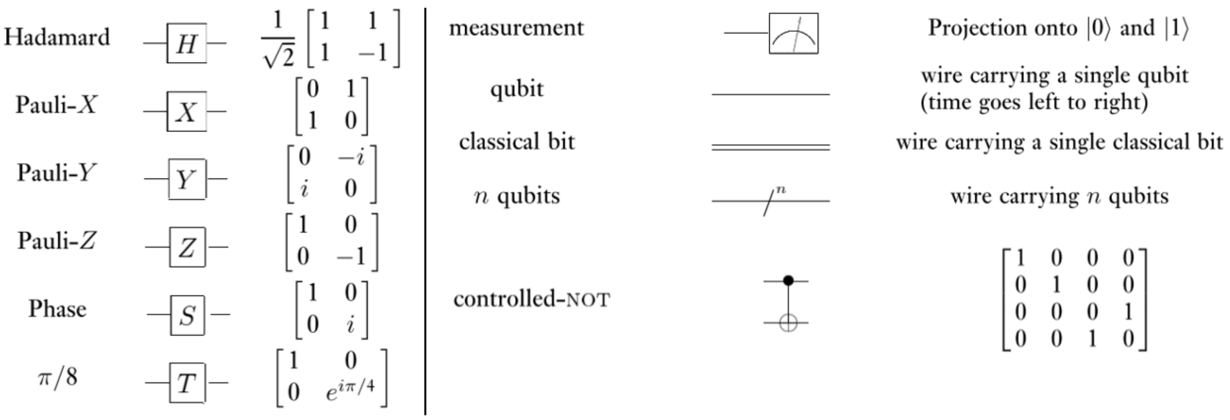

Hadamard
H
$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X
X
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y
Y
$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z
Z
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase
S
$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$
T
$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
measurement
Projection onto $|0\rangle$ and $|1\rangle$
qubit
wire carrying a single qubit (time goes left to right)
classical bit
wire carrying a single classical bit
n qubits
n
wire carrying n qubits
controlled-NOT
$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Figure 1: Common gates and symbols used in circuits. The CNOT gate flips the bottom qubit if the top qubit is $|1\rangle$. Figure adapted from Page xxx - xxxi of Ref [16]

This concludes a description of the canonical quantum circuit model of computation [17], but there exist other equivalent models of computation such as adiabatic quantum computing [18] that fall outside this review's scope.

A key difference between classical and quantum computing is the $O(\exp(N))$ scaling of the classical bits required to encode the information in an N body quantum system compared to $O(N)$ qubits for a quantum computer[2, 14]. This does not automatically guarantee that QCs are more powerful than classical computers, but that they could potentially solve hard problems on classical computers more easily [5]. Over the last 30 years, numerous algorithms (see Table 1) have been theorized to provide speedups over classical methods, with Shor's factoring algorithm [19] gaining fame for its potential to disrupt public key cryptography.

Algorithm	Description	Speedup over classical algorithms
Shor's Factoring Algorithm [20]	Determines the prime factors of an n bit integer	Superpolynomial
Grover's Searching Algorithm [20]	Given a black box function, with N inputs, all outputs are 0 except for a winning input, w that gives 1 as an output. The algorithm finds w.	Polynomial
Quantum Phase Estimation [21]	Phase estimation is an important sub-routine in other algorithms. It enables the eigenphases of an operator to be estimated using a quantum speedup arising from running a quantum Fourier transform.	NA

Table 1: Notable quantum algorithms theorised to provide significant practical speedups

2.2 Decoherence and Error Correction

QCs can be physically implemented via a range of technologies such as superconducting circuits, trapped ions, and photons [22]. QCs are not isolated from their surroundings, which can lead to pure states qubits becoming correlated with the environment, resulting in them not being able to interfere and perform computation [23]. Each implementation has to contend with this decoherence, while externally controlling and allowing interference between qubits [5, 7].

Thus, errors in gate operations (quantified by their error rates) are an inherent part of quantum computing, which need to be actively managed[24]. The no-cloning theorem prevents the duplication of quantum states $|\psi\rangle$ i.e the operation in Equation 3 [24]

$$U_{\text{duplicate}}\left(|\psi\rangle \otimes |0\rangle\right) \rightarrow |\psi\rangle \otimes |\psi\rangle \tag{3}$$

Hence, error-correction codes used in classical computers cannot be used in QCs [24] and quantum error correction (QEC) was developed [6]. QEC works by redundantly encoding x *logical* qubits by entangling them over y physical qubits [17, 24] (where $y > x$). Logical qubits, unlike physical qubits, are representative of the theoretical state in Equation 2 [17].

It can then be shown, using the *threshold theorem*, that if gate error rates are below a constant threshold, arbitrarily large-scale quantum computation can be achieved under "physically reasonable assumptions" about the type of noise present [16]. Using the surface code, the most extensively researched QEC code, a threshold of 1% is obtained entailing $y/x \sim 10^3 - 10^4$ [17]. This requires a fault-tolerant QC to have $\sim 10^6$ physical qubits, in stark contrast with the $\sim 10^3$ physical qubits in the largest QCs now [25], to obtain a speedup over classical computers with Shor's factoring algorithms [17]. Therefore, the NISQ era persists (see Figure 2.2).

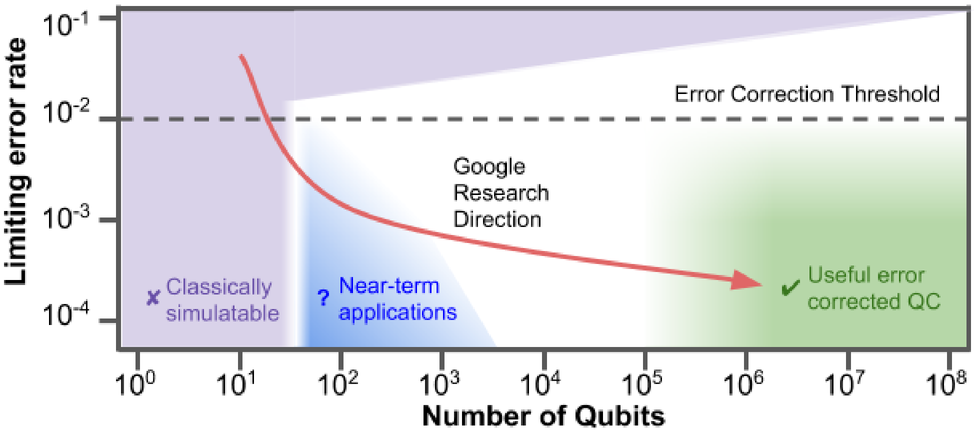


Figure 2: A plot reproduced from Google Research's blog [26] demonstrating where practical error correction lies.

3 NISQ algorithms

Having looked at the limits of NISQ quantum computers, we now consider what we can do with the algorithms we can run on them.

3.1 Constraints on NISQ algorithms

The limited capabilities of NISQ computers hinder the realisation of the theorized speedups in Table 1. Therefore we limit our discussion to 'NISQ algorithms' capable of running on "near term" QCs without error correction [7]. While there is some subjectivity about what "near-term" constitutes, we refer to the operational definition proposed by Ref [8]. This limits NISQ algorithms to those that [8]:

- work within current QC's qubit counts,
- have low-depth circuits (less than 100 gates),
- minimise usage of complicated gates (e.g. multi-qubit controlled gates),
- and are suitable for noise reduction through error mitigation.

Our discussion here focuses on variational quantum algorithms (VQAs) since most applications in recent reviews of NISQ algorithms [7, 8] utilized VQAs.

3.2 Variational Quantum Algorithms (VQAs)

3.2.1 Description of VQAs

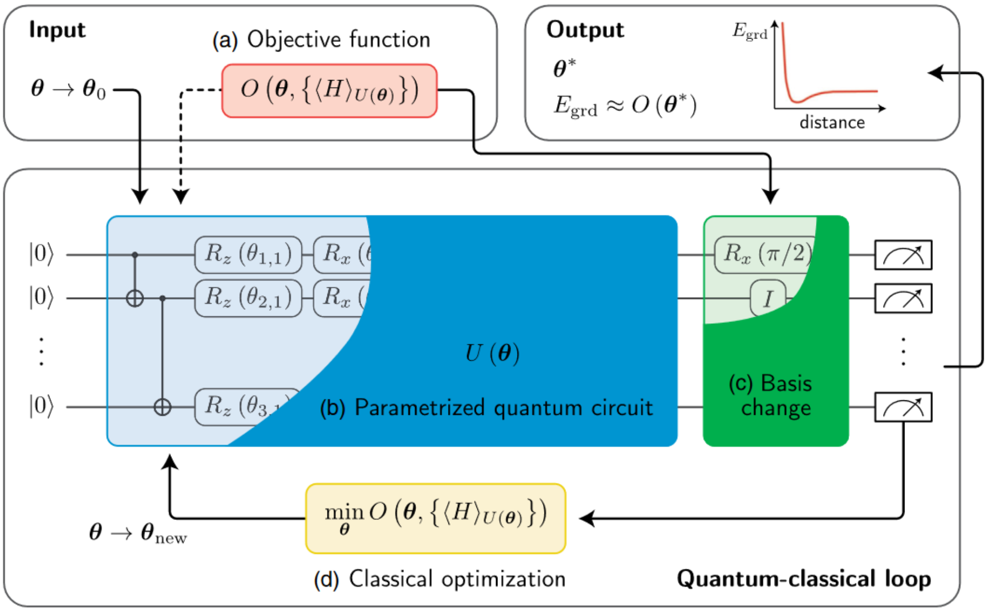
First described by [27], Variational Quantum Algorithms (VQAs) are hybrid quantum-classical algorithms, that involve variationally updating parameters to solve simulation and optimisation problems [12]. As described in Figure 3, VQAs consist of [7]:

(a) A cost (or objective function) $\mathcal{O}(\boldsymbol{\theta})$ encodes the chosen problem's solution. It is commonly defined in terms of the expectation value of an operator (typically a Hamiltonian H), parameterised by a set of parameters $\boldsymbol{\theta}$

(b) The parameterized quantum circuit (PQC) (or ansatz) $U(\boldsymbol{\theta})$: a unitary operation parameterised by $\boldsymbol{\theta}$, which are updated to minimise $\mathcal{O}(\boldsymbol{\theta})$. $U(\boldsymbol{\theta})$ acts on an initial n qubit state $|\Psi_0\rangle = |0\rangle^{\otimes n}$ to form $|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|\Psi_0\rangle$

(c) A measurement protocol to evaluate $\langle H\rangle_{U(\boldsymbol{\theta})} = \langle\Psi(\boldsymbol{\theta})|H|\Psi(\boldsymbol{\theta})\rangle$ in the computational basis

(d) A classical optimiser used to update the θ from the input θ_0 to θ_{new} in each iteration of the quantum-classical loop to minimise $\mathcal{O}$ i.e obtain $\min_{\boldsymbol{\theta}} O\left(\boldsymbol{\theta}, \left\{\langle H\rangle_{U(\boldsymbol{\theta})}\right\}\right)$



Input
$\boldsymbol{\theta} \to \boldsymbol{\theta}_0$
(a) Objective function
$O\left(\boldsymbol{\theta}, \left\{\langle H\rangle_{U(\boldsymbol{\theta})}\right\}\right)$
Output
$\boldsymbol{\theta}^*$
$E_{\text{grd}} \approx O\left(\boldsymbol{\theta}^*\right)$
E_{grd}
distance
$|0\rangle$
$|0\rangle$
$|0\rangle$
$R_z\left(\theta_{1,1}\right)$
$R_x\left(\theta\right.$
$R_z\left(\theta_{2,1}\right)$
$R_x\left(\right.$
$R_z\left(\theta_{3,1}\right)$
$U\left(\boldsymbol{\theta}\right)$
(b) Parametrized quantum circuit
$R_x\left(\pi/2\right)$
I
(c) Basis change
$\min_{\boldsymbol{\theta}} O\left(\boldsymbol{\theta}, \left\{\langle H\rangle_{U(\boldsymbol{\theta})}\right\}\right)$
$\boldsymbol{\theta} \to \boldsymbol{\theta}_{\text{new}}$
(d) Classical optimization
Quantum-classical loop

Figure 3: The quantum-classical loop in the VQA circuit runs until $\min_{\boldsymbol{\theta}} O\left(\boldsymbol{\theta}, \left\{\langle H\rangle_{U(\boldsymbol{\theta})}\right\}\right)$ is obtained. Figure reproduced from Fig 2 of Ref [7]

VQAs describe a general process, customisable by specifying the cost function, the ansatz, and the optimisation process.

When encoding the problem into $\mathcal{O}$, there are a few options. One could encode the problem (even if it is a non-physical one) into a Hamiltonian (or any valid operator) so that finding its ground state energy produces a solution [7]. This requires decomposing the Hamiltonian (or operator) into a set of operators measurable by QC in the computational basis (such as linear combinations of Pauli matrix tensor products) in the measurement step [7]. Another choice involves defining the solution as a target state $|\psi\rangle$ and optimising $|\Psi(\boldsymbol{\theta})\rangle$ with respect to the target state [7]. The cost function used here is the *fidelity* between $|\psi\rangle$ and $|\Psi(\boldsymbol{\theta})\rangle$ given by

$$F\left(\Psi, \Psi_{U(\boldsymbol{\theta})}\right) \equiv \left|\left\langle \Psi \mid \Psi_{U(\boldsymbol{\theta})}\right\rangle\right|^2 \tag{4}$$

With optimisation, there are two possible approaches: gradient-free methods and gradient descent methods [12]. Gradient-free approaches include the techniques like Nelder-Mead method that fall outside the scope of this review [12]. Gradient descent involves finding a local minimum for a given $C(\boldsymbol{\theta})$ with respect to parameters $\boldsymbol{\theta}$ by computing the direction of steepest descent $\nabla_\theta C(\boldsymbol{\theta})$ to iteratively update $\boldsymbol{\theta}$ and minimise $\nabla_\theta C(\boldsymbol{\theta})$ via Equation 5.

$$\boldsymbol{\theta} \to \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) \tag{5}$$

where η is the step size [12].

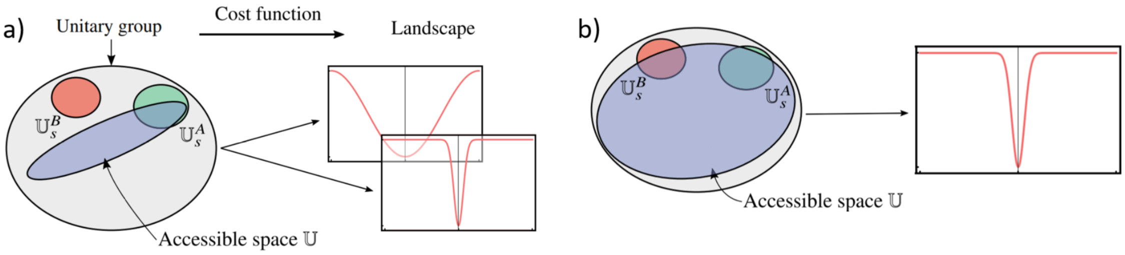


Figure 4: Considering problems A and B, that have solutions in $\mathbb{U}_s^a$ and $\mathbb{U}_s^b$ a) demonstrates a low accessibility PQC that cannot access solutions to B, but can have steep or flat cost gradients. b) shows a high accessibility PQC that can access solutions to A and B, but leads to mostly flat cost gradients. Figure adapted from Fig 1 in Ref [28].

The choice of PQC $U(\boldsymbol{\theta})$ heavily influences the VQA's effectiveness, where highly expressible and trainable PQCs are preferred [28]. The expressibility can be understood by considering that for a set PQC $U(\boldsymbol{\theta})$, an ensemble of unitary operations $\mathbb{U} = \left\{U^{(1)}, U^{(2)}, \ldots, U^{(y)}\right\}$ are generated for different $\boldsymbol{\theta}$: $\left\{\boldsymbol{\theta}^{(1)}, \ldots \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(y)}\right\}$ [28]. Considering Figure 4, the expressibility quantifies the extent to which $\mathbb{U}$ covers the space of all possible same-degree unitaries [28]. We want high expressibility so that the unitaries representing solutions in $\mathbb{U}_s^a$ and $\mathbb{U}_s^b$ are accessible to $U(\boldsymbol{\theta})$ as it is optimised[28]. At the same time, we want a well *trainable* PQC that has large gradients in the cost-function landscape to allow $U(\boldsymbol{\theta})$ to converge to the unitary that represents the solution [28]. Some common PQCs include "problem-inspired" ansatze which are constructed by considering specific aspects of a problem and "hardware-efficient" ansatzes which are designed with the hardware specifications of NISQ devices in mind [12].

3.2.2 The Utility of VQAs

The use of the VQAs is complicated by multiple factors. Firstly, it is subject to difficulties generally encountered in optimisation like the 'barren-plateu' problem [8]. When using a random class of PQCs an exponential flattening of the cost landscape is seen due to the exponential decay of the gradient of $\langle H\rangle_{U(\boldsymbol{\theta})}$ [7, 29]. Resultantly, the optimisation process becomes exponentially harder [8]. It is expected that this would affect gradient descent methods, but gradient-free methods are also affected due to an exponential drop in cost function differences, which the method bases its decisions on [30]. Barren plateus can arise due to an "excess of entanglement", hardware noise present in the system, and the choice of cost function [7]. Furthermore, there is ambiguity involved in choosing a PQC [8] and a trade-off between expressibility and trainability (see Figure 4 [28])

Lastly, complications can arise due to real and imaginary parts present in $\langle H\rangle_{U(\boldsymbol{\theta})}$. This requires additional Hadamard tests in the measurement step and the inclusion of an ancillary qubit, which adds to the complexity of the circuit [8].

However, VQAs also offer much promise. VQAs can be implemented on any type of quantum computer (enabling the performance benchmarking of NISQ computers), can leverage hardware advantages such as lower gate errors, and are capable of variationally mitigating quantum errors [31–33].

The applicability of VQAs to a wide range of problems [12] makes it a potentially valuable part of most future QC applications.

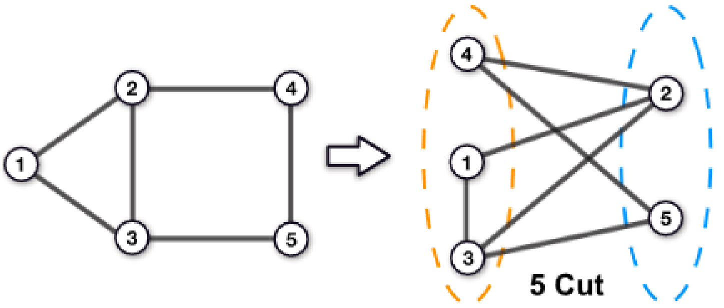
3.2.3 Variational Quantum Eigensolver (VQE)

The VQE is a specific instance of the VQA used to determine the ground state E_0 of a Hamiltonian H [27, 34]. This defines the cost function to be minimised as

$$C(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle \tag{6}$$

where for PQC $U(\boldsymbol{\theta})$ and initial state $|\psi_0\rangle$, $|\psi(\theta)\rangle = U(\theta)\,|\psi_0\rangle$ [34]. It follows from the Rayleigh-Ritz that minimising Equation 6, yields $C(\boldsymbol{\theta}) \geq E_g$ [34, 35]. Equality holds if $|\psi(\theta)\rangle$ is the ground state of the system [34, 35]. This describes the original VQE, which can be modified and expanded to find exited states, the Hamiltonian, non-equilibrium steady states, and energy derivatives [12]. This has applications in a wide range of fields such as many-body physics and quantum chemistry [14, 17, 36] (See Section 4.1).

3.2.4 Quantum Approximate Optimisation Algorithm



$x_1..x_5$	Val	$x_1..x_5$	Val
00000	0	01000	3
00001	2	01001	**5**
00010	2	01010	3
00011	2	01011	3
00100	3	01100	4
00101	3	01101	4
00110	5	01110	4
00111	3	01111	3

Figure 5: An overview of the MaxCut problem for a 5-node graph, with solution = 5. Possible division of nodes are represented by bitstrings. Figure reproduced from Fig 33 in Ref. [21]

First proposed by [37], the Quantum Approximate Optimisation Algorithm has promise in demonstrating a practical quantum speed for combinatorial optimisation problems [12]. In particular, the Max-Cut problem has garnered lots of attention in the field [8]. The problem involves partitioning nodes of a graph with n nodes and m edges into 2 different sets A and B so that the number of edges cut by the partition is maximized (as shown in Figure 5) [12]. The division of nodes can be represented using a bitstring $z = z_1, \ldots, z_n$ where each bit represents a node and the node belongs to A if $z_i = 1$ and B if $z_i = 0$ [12, 37]. This can be understood using the VQA framework by denoting $C(z)$, the number of edges cut, as an objective function to maximise [12]. Writing the bit-string as a product of computational basis states $|z\rangle$, the objective function is

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) = \sum_{\text{edge }(j,k)}^{m} \frac{1}{2}\left(1 - \sigma_z^j \sigma_z^k\right)|z\rangle, \tag{7}$$

where j, k are indices representing nodes connecteed to edge α and σ_z is the Pauli Z gate [12]. The PQC is then written as [8]

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \underbrace{\left(e^{-iH_c\beta_p} e^{-iH_m\gamma_p}\right) \ldots \left(e^{-iH_c\beta_1} e^{-iH_m\gamma_1}\right)}_{p\text{ times}} |+\rangle^{\otimes n} \tag{8}$$

where the mixing problem Hamiltonian $H_C = \sum_{\text{edge }(j,k)}^{m} \frac{1}{2}\left(1 - \sigma_z^j \sigma_z^k\right)$ and the mixing Hamiltonian $H_M = \sum_{i=1}^{n} \sigma_x^i$ are alternatively applied [12]. $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ form the parameters that are optimised and $|+\rangle^{\otimes n}$ is the input state [12]. A larger p yields better approximations, with $p \to \infty$ yielding the exact solution [8]. The NISQ constraint of low-depth circuits limits the QAOA implementations to low p values and hence only approximate solutions, but there have been demonstrable advantages over classical methods (under certain reasonable assumptions) in literature [8]

We briefly mention other NISQ algorithms in use today, that appear to have less promise for practical use than VQAs.

Boson sampling (BS) and Gaussian boson sampling (GBS) were both suggested as tasks in photonic QCs on which quantum computers could demonstrate an advantage over classical computers [7, 8]. Both processes differ in their inputs, with GBS improving on difficulties in BS, but involve calculating some property of a matrix that is computationally difficult on a classical computer [8]. Despite being used to showcase how quantum computers could outperform classical computers [38], GBS has limited practical applications [39].

Refs [7, 8, 40] also refer to a class of algorithms called QAMs to find the ground state energy and simulate time-evolution for a given Hamiltonian [8]. These methods appear to avoid feedback looks and barren plateus, but have limited mention in literature.

4 Applications of NISQ algorithms

Having considered the capabilities of the most promising NISQ algorithms, we now consider how they might be applied in 3 categories: simulation, optimisation, and machine learning.

4.1 Quantum Simulation

QCs efficiently implement two aspects of computation that classical algorithms seek to circumvent [14]:

- keeping track of the complete many-body wavefunction $|\psi\rangle$
- time-evolving the wavefunction by "general matrix-vector multiplication"

Simulation problems are either static problems involving measuring the expectation value of an observable, or dynamic problems concerning wavefunction time-evolution [8, 17]. Classical approximation techniques begin to fail for static calculations when describing strongly correlated systems [41], estimating quantities of fermionic states [14], and determining the wavefunction (and energy) of larger molecules [14]. In the case of dynamic calculations, classical techniques are only applicable for "very small" systems and short time-evolutions [14].

There exist many strategies to plug some of these gaps, but these involve the quantum phase estimation algorithm (see Table 1) which cannot be implemented within the constraints of NISQ devices [8]. In NISQ devices limited success has been seen using VQEs (see Section 3.2.3). The use of VQEs has been demonstrated experimentally on popular experimental setups such as superconducting, trapped-ions, and photonic devices [17]. It also displays limited error suppression and has been used to find energies of small systems with shallow circuits [17, 42]. However, calculations involving such systems are already achievable on classical computers, and there is no guarantee of a quantum advantage over a classical computer by a VQE [8].

Achieving a quantum advantage would require longer circuits which are likely to be too noisy to implement even after considering error mitigation methods[17] but it might be possible with the choice of a suitable ansatz [33]. VQEs also suffer from the same limitations involving optimisation and scaling of measurement counts that affect all VQAs. Lastly, while obtaining energy calculations is easy, extracting the full wavefunction can be more difficult than classical computation of the solution itself [43].

VQAs developed to simulate the time-evolution of many-body systems have similarly been shown to work well for small systems, even avoiding barren plateaus [44], but there's no evidence suggesting this works for larger systems [8].

In effect, most methods available for simulation currently offer interesting proof of principles but cannot demonstrate a quantum advantage over classical computers with certainty.

4.2 Machine Learning

Machine Learning (ML), a process by which algorithms and data are leveraged to identify and categorise concealed patterns in large datasets, has been of immense practical value to society [8, 15]. ML can be expressed within the framework of quantum mechanics, to yield quantum machine learning (QML), which is believed to yield a quantum speedup on NISQ devices [15]. While there is no clear definition of what QML includes, the tasks it can perform are summarised in Figure 6. Just as classical ML consists of a variety of models (see Table 2), so QML has analogous paradigms, with supervised learning being particularly promising due to its noise-resistance, broad applicability, and potential for quantum speedup [15]. Therefore we focus on a key feature of supervised (as well as other) learning schemes: Quantum Neural Networks (QNNs) [15]. Section 4.3 discusses NISQ applications relevant to sub-routines in ML such as solving linear equations.

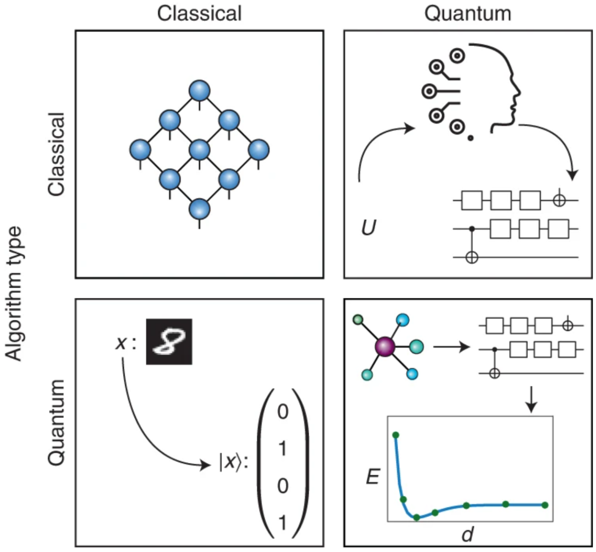


Figure 6: QML use-cases can be broken down into four categories[15]. Top left: The use of quantum-inspired algorithms such as tensor networks for analysis of classical data [15]. Top right: Data from a unitary time evolution is compiled into a circuit via classical methods [15]. Bottom right: the classification of molecular ground-state information on a quantum computer [15]. Bottom left: classifying handwritten numbers by mapping them to quantum numbers to be analysed on a quantum computer. Figure reproduced from Fig 1 of Ref [15]

Classical neural networks can be ”embedded” into PQCs [15] and generalised to receive and analyse quantum inputs [45]. QNNs are therefore a subset of VQAs and beholden to the same problems of barren-plateaus and trainability discussed in Section 3.2.[46]. The scarcity of ”truly quantum” datasets adds to the difficulties in QML since QML models utilising quantum data are more likely to achieve a quantum speedup [15].

While classical datasets are widely available and are used in benchmarking of QML architectures, encoding classical information into quantum information via the process of *embedding* raises other issues [15]. Embedding techniques have conflicting requirements: the encoded quantum states should have an inner product that is difficult to classically simulate, but it should also contain features useful to the problem (e.g. the states of a classification task should be in distinct sub-spaces of Hilbert space) [15]. Most QML methods that suggest near-term compatibility also ignore hardware noise in their analysis, yet noise contributes to barren-plateaus and limits circuit depth [15].

QNNs apply to supervised, unsupervised, and reinforcement learning [15] making them useful across a range of disciplines. A key proposed speedup is searching for quantum error correcting codes (see Section 2.2) since searching Hilbert spaces for QEC subspaces is an inherent benefit of running QNNs on quantum data [15]. Yet this has been difficult to study using only "small-scale numerical simulations"[15].

Current approaches have aimed to implement QNNs with better expressibility (see Section 3.2) than classical neural networks but whether an advantage is to be gained as the size of the problems scale is ambiguous [43].

Machine Learning Models	Description
Supervised Learning	This involves training an algorithm to perform a task with a complete set of labeled data. Labeled data has the successful outcome of the task tagged to it. E.g. images in a labeled dataset for a classification task would have the images tagged with a category
Unsupervised Learning	When labeled data is not at hand, an algorithm (like a neural network) can be run on the data to autonomously identify patterns in the data
Semi-Supervised Learning	Uses a dataset containing both labeled and unlabeled data for training. This is beneficial when labeling data takes too long, and autonomous pattern identification does not identify features relevant to the task.
Reinforcement Learning	In this case, no dataset is used. The model is trained to achieve the optimal solution via repeated trial and error.
Deep Learning	Deep learning models are a type of neural networks, that aim to simulate how the human brain works. This allows for sophisticated models to be built, that can be trained without much direction.

Table 2: A description of classical machine learning models[45, 47–49]

4.3 Combinatorial Optimisation and other mathematical methods

Current literature also suggests that there are quantum speedups in finding optimal solutions to combinatorial problems (i.e. combinatorial optimisation) and solving linear systems.

While we have seen proof of concept NISQ implementations of QAOA to solve these problems[8, 50, 51], there has not been any demonstrable speed up of QAOA [7].

A quantum speedup has also been theoretically postulated for particular numerical solvers[8]. In particular, the Harrow-Hassidim-Lloyd (HHL) algorithm [52] is expected to show an exponential speedup in inverting a matrix (under certain conditions) over classical solvers of the linear system problem (solving for x in $Ax = B$). HHL has been demonstrated for an 8×8 matrix, but cannot be scaled up on NISQ devices [8]. VQAs have also been proposed to solve the linear system problem, but are bound to the issues discussed earlier [8].

There have also been proof-of-principle demonstrations of VQAs, QAOA and quantum-assisted methods for problems such as non-linear differential equations and factoring, but only for simplified systems with computation significantly impacted by noise [8].

5 Conclusions and Outlook

In conclusion, this review presented a brief introduction to quantum computing, error correction as well as, NISQ algorithms and their applications.

VQAs, which leverage the strengths of classical and quantum methods, show promise of demonstrating a quantum speedup over classical methods in optimisation, simulation, and machine learning, but their utility is severely curtailed by noise and decoherence. Other methods such as GBS have limited practical utility.

But this is not surprising. When he coined the term NISQ, John Preskill emphasised that realising QCs' full potential will not be straightforward [5]. NISQ was never the end-goal, and the progress made during this era could potentially help make fault-tolerance a reality [8]. Recent experiments have shown drastic improvements in error rates [53, 54], which suggests that this could happen sooner rather than later.

References

[1] Bogobowicz, M. *et al.* Record investments in quantum technology — McKinsey. URL https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-technology-sees-record-investments-progress-on-talent-gap.

[2] Feynman, R. P. Simulating physics with computers **21**, 467–488. URL https://doi.org/10.1007/BF02650179.

[3] Biamonte, J. *et al.* Quantum machine learning **549**, 195–202. URL https://www.nature.com/articles/nature23474.

[4] Gamble, S. Quantum Computing: What It Is, Why We Want It, and How We're Trying to Get It. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium* (National Academies Press (US)). URL https://www.ncbi.nlm.nih.gov/books/NBK538701/.

[5] Preskill, J. Quantum Computing in the NISQ era and beyond **2**, 79. URL https://doi.org/10.22331/q-2018-08-06-79. 1801.00862.

[6] Shor, P. W. Scheme for reducing decoherence in quantum computer memory **52**, R2493–R2496. URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[7] Bharti, K. *et al.* Noisy intermediate-scale quantum algorithms **94**, 015004. URL https://link.aps.org/doi/10.1103/RevModPhys.94.015004.

[8] Lau, J. W. Z., Lim, K. H., Shrotriya, H. & Kwek, L. C. NISQ computing: Where are we and where do we go? **32**, 27. URL https://doi.org/10.1007/s43673-022-00058-z.

[9] Kalai, G. Quantum Computers: A Brief Assessment of Progress in the Past Decade. URL https://gilkalai.wordpress.com/2022/05/26/waging-war-on-quantum/.

[10] Gourianov, N. The quantum computing bubble URL https://www.ft.com/content/6d2e34ab-f9fd-4041-8a96-91802bab7765.

[11] Dyakonov, M. The Case Against Quantum Computing - IEEE Spectrum. URL https://spectrum.ieee.org/the-case-against-quantum-computing.

[12] Huang, H.-L. *et al.* Near-term quantum computing techniques: Variational quantum algorithms, error mitigation, circuit compilation, benchmarking and classical simulation **66**, 250302. URL https://doi.org/10.1007/s11433-022-2057-y.

[13] García, D. P., Cruz-Benito, J. & García-Peñalvo, F. J. Systematic Literature Review: Quantum Machine Learning and its applications. URL http://arxiv.org/abs/2201.04093. 2201.04093.

[14] Cao, Y. *et al.* Quantum Chemistry in the Age of Quantum Computing **119**, 10856–10915. URL http://arxiv.org/abs/1812.09976. 1812.09976.

[15] Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L. & Coles, P. J. Challenges and opportunities in quantum machine learning **2**, 567–576. URL https://www.nature.com/articles/s43588-022-00311-3.

[16] Nielsen, M. A. & Chuang, I. L. Quantum Computation and Quantum Information: 10th Anniversary Edition. URL https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE.

[17] McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. & Yuan, X. Quantum computational chemistry **92**.

[18] Aharonov, D. *et al.* Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation **50**, 755–787. URL `https://epubs.siam.org/doi/10.1137/080734479`.

[19] Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer **26**, 1484–1509. URL `https://doi.org/10.1137/S0097539795293172`.

[20] Jordan, S. Quantum Algorithm Zoo. URL `https://quantumalgorithmzoo.org/`.

[21] J., A. *et al.* Quantum Algorithm Implementations for Beginners **3**, 18:1–18:92. URL `https://dl.acm.org/doi/10.1145/3517340`.

[22] Chae, E., Choi, J. & Kim, J. An elementary review on basic principles and developments of qubits for quantum computing **11**, 11. URL `https://doi.org/10.1186/s40580-024-00418-5`.

[23] Chuang, I. L., Laflamme, R., Shor, P. W. & Zurek, W. H. Quantum Computers, Factoring, and Decoherence **270**, 1633–1635. URL `https://www.science.org/doi/10.1126/science.270.5242.1633`.

[24] Roffe, J. Quantum Error Correction: An Introductory Guide **60**, 226–245. URL `http://arxiv.org/abs/1907.11157`. `1907.11157`.

[25] IBM Quantum System Two: The era of quantum utility is here — IBM Quantum Computing Blog. URL `https://www.ibm.com/quantum/blog/quantum-roadmap-2033`.

[26] A Preview of Bristlecone, Google's New Quantum Processor. URL `http://research.google/blog/a-preview-of-bristlecone-googles-new-quantum-processor/`.

[27] Peruzzo, A. *et al.* A variational eigenvalue solver on a photonic quantum processor **5**, 4213. URL `https://www.nature.com/articles/ncomms5213`.

[28] Holmes, Z., Sharma, K., Cerezo, M. & Coles, P. J. Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus **3**, 010313. URL `https://link.aps.org/doi/10.1103/PRXQuantum.3.010313`.

[29] McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes **9**, 4812. URL `https://www.nature.com/articles/s41467-018-07090-4`.

[30] Arrasmith, A., Cerezo, M., Czarnik, P., Cincio, L. & Coles, P. J. Effect of barren plateaus on gradient-free optimization **5**, 558. URL `https://quantum-journal.org/papers/q-2021-10-05-558/`.

[31] O'Malley, P. J. J. *et al.* Scalable Quantum Simulation of Molecular Energies **6**, 031007. URL `https://link.aps.org/doi/10.1103/PhysRevX.6.031007`.

[32] McClean, J. R., Kimchi-Schwartz, M. E., Carter, J. & de Jong, W. A. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states **95**, 042308. URL `https://link.aps.org/doi/10.1103/PhysRevA.95.042308`.

[33] McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms **18**, 023023. URL `https://dx.doi.org/10.1088/1367-2630/18/2/023023`.

[34] Cerezo, M. *et al.* Variational quantum algorithms **3**, 625–644. URL `https://www.nature.com/articles/s42254-021-00348-9`.

[35] Gould, S. H. Variational Methods for Eigenvalue Problems: An Introduction to the Weinstein Method of Intermediate Problems (Second Edition). In *Variational Methods for Eigenvalue Problems*, 57–84 (University of Toronto Press). URL `https://www.degruyter.com/document/doi/10.3138/9781487596002/html`.

[36] Pal, S., Bhattacharya, M., Lee, S.-S. & Chakraborty, C. Quantum Computing in the Next-Generation Computational Biology Landscape: From Protein Folding to Molecular Dynamics URL https://doi.org/10.1007/s12033-023-00765-4.

[37] Farhi, E., Goldstone, J. & Gutmann, S. A Quantum Approximate Optimization Algorithm. URL http://arxiv.org/abs/1411.4028. 1411.4028.

[38] Zhong, H.-S. *et al.* Quantum computational advantage using photons **370**, 1460–1463. URL https://www.science.org/doi/10.1126/science.abe8770.

[39] Bromley, T. R. *et al.* Applications of near-term photonic quantum computers: Software and algorithms **5**, 034010. URL https://dx.doi.org/10.1088/2058-9565/ab8504.

[40] Bharti, K. & Haug, T. Iterative quantum-assisted eigensolver **104**, L050401. URL https://link.aps.org/doi/10.1103/PhysRevA.104.L050401.

[41] Bulik, I. W., Henderson, T. M. & Scuseria, G. E. Can Single-Reference Coupled Cluster Theory Describe Static Correlation? **11**, 3171–3179. URL https://doi.org/10.1021/acs.jctc.5b00422.

[42] Kandala, A. *et al.* Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets **549**, 242–246. URL http://arxiv.org/abs/1704.05018. 1704.05018.

[43] Outeiral, C. *et al.* The prospects of quantum computing in computational molecular biology **11**, e1481. URL http://arxiv.org/abs/2005.12792. 2005.12792.

[44] Lee, C. K., Patil, P., Zhang, S. & Hsieh, C. Y. Neural-network variational quantum algorithm for simulating many-body dynamics **3**, 023095. URL https://link.aps.org/doi/10.1103/PhysRevResearch.3.023095.

[45] Wan, K. H., Dahlsten, O., Kristjánsson, H., Gardner, R. & Kim, M. S. Quantum generalisation of feedforward neural networks **3**, 36. URL http://arxiv.org/abs/1612.01045. 1612.01045.

[46] Abbas, A. *et al.* The power of quantum neural networks **1**, 403–409. URL https://www.nature.com/articles/s43588-021-00084-1.

[47] What Is Machine Learning (ML)? — IBM. URL https://www.ibm.com/topics/machine-learning.

[48] Wordpress, U. What Is Machine Learning (ML)? URL https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/.

[49] Salian, I. NVIDIA Blog: Supervised Vs. Unsupervised Learning. URL https://blogs.nvidia.com/blog/supervised-unsupervised-learning/.

[50] Oh, Y.-H. *et al.* Solving Multi-Coloring Combinatorial Optimization Problems Using Hybrid Quantum Algorithms. URL http://arxiv.org/abs/1911.00595. 1911.00595.

[51] Braine, L., Egger, D. J., Glick, J. & Woerner, S. Quantum Algorithms for Mixed Binary Optimization applied to Transaction Settlement **2**, 1–8. URL http://arxiv.org/abs/1910.05788. 1910.05788.

[52] Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum Algorithm for Linear Systems of Equations **103**, 150502. URL https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[53] da Silva, M. P. *et al.* Demonstration of logical qubits and repeated error correction with better-than-physical error rates. URL http://arxiv.org/abs/2404.02280. 2404.02280.

[54] Bluvstein, D. *et al.* Logical quantum processor based on reconfigurable atom arrays **626**, 58–65. URL https://www.nature.com/articles/s41586-023-06927-3.