This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.DOI

Quantum Computing: Circuits, Algorithms, and Applications

MUHAMMAD ALI SHAFIQUE[1], ARSLAN MUNIR[2], and IMRAN LATIF[3]
[1]Department of Electrical and Computer Engineering, Kansas State University, Manhattan, Kansas, 66506, USA (e-mail: alishafique@ksu.edu)
[2]Department of Computer Science, Kansas State University, Manhattan, Kansas, 66506, USA (e-mail: amunir@ksu.edu)
[3]Brookhaven National Laboratory, U.S. Department of Energy, Upton, NY 11973-5000, USA (e-mail: ilatif@bnl.gov)

Corresponding author: Arslan Munir (e-mail: amunir@ksu.edu).

ABSTRACT Quantum computing, a transformative field that emerged from quantum mechanics and computer science, has gained immense attention for its potential to revolutionize computation. This paper aims to address the fundamentals of quantum computing and provide a comprehensive guide for both novices and experts in the field of quantum computing. Beginning with the foundational principles of quantum computing, we introduce readers to the fundamental concepts of qubits, superposition, entanglement, interference, and noise. We explore quantum hardware, quantum gates, and basic quantum circuits. This study offers insight into the current phase of quantum computing, including the noisy intermediate-scale quantum (NISQ) era and its potential for solving real-world problems. Furthermore, we discuss the development of quantum algorithms and their applications, with a focus on famous algorithms like Shor's algorithm and Grover's algorithm. We also touch upon quantum computing's impact on various industries, such as cryptography, optimization, machine learning, and material science. By the end of this paper, readers will have a solid understanding of quantum computing's principles, applications, and the steps involved in developing quantum circuits. Our goal is to provide a valuable resource for those eager to embark on their quantum computing journey and for researchers looking to stay updated on this rapidly evolving field.

INDEX TERMS quantum computing, entanglement, interference, quantum circuits, quantum algorithms, quantum applications

I. INTRODUCTION

QUANTUM computing technology uses different approaches to solve certain computational problems, demonstrating greater efficiency compared to classical computing systems. Recent experimental outcomes are remarkable, hinting at the possibility of quantum computers becoming commercially available in the near future [1]–[4]. A prominent example of quantum computing's ability lies in Shor's algorithm, renowned for its capability to factor large numbers efficiently [5]. This algorithm in 1994 marked a pivotal step in the advancement of quantum computing by enabling the determination of prime factors of large numbers by quantum computers [6], and has also caused consternation in cryptography field as many public key cryptography algorithms rely on the difficulty of factoring large numbers by classical computers. A distinct difference in computational power of classical versus quantum computers can be illustrated with the evaluation of the time required to crack encryption schemes like Rivest–Shamir–Adleman (RSA) that rely on the difficulty of finding prime factors of large numbers [7]. While traditional computers would require billions of years for such a task, quantum computers could potentially solve it in a short time [8], [9].

Quantum computers share some components with classical computers, such as registers, gates, and memory elements. However, their underlying physical structures are fundamentally distinct and unique. Quantum computations unfold within quantum registers, where qubits can exist in the state of superposition and entanglement. These unique characteristics make quantum computers fundamentally different from traditional classical computers.

Another distinct difference of quantum computing versus classical computing is computational units such as bits. Bits in classical computing are restricted to zero or one whereas quantum computing employ units (qubits) that are capable of existing in states of zero, one, or any intermediate value [10]–[12]. This unique attribute grants quantum computers the remarkable ability to simultaneously follow multiple computational paths within a single calculation, which is not possible by classical computers without repeated iterations.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

A. QUANTUM COMPUTING INTRODUCTION AND HISTORY

Quantum computing, in contrast to classical computing, is a relatively recent development. Its origins can be traced back to the late 1970s when it initially appeared in science fiction, subsequently attracting significant attention from the media. It was in 1981 that Richard Feynman is credited with pioneering the concept of a quantum computer. He proposed the idea that quantum computers could efficiently simulate quantum systems that could avoid the exponential resource requirements for classical computers. Classical computers encounter substantial difficulties when attempting to simulate quantum systems. Feynman, along with visionaries like Yuri Manin and Paul Benioff, recognized the vast potential of quantum computers in the realm of complicated computing problems. In 1985, David Deutsch formalized the concept of a quantum computer, marking a significant milestone in the field of quantum computing. Furthermore, he distinguished between quantum simulators and programmable quantum devices.

In subsequent years, significant achievements were made in the field of quantum computing, revealing its potential to surpass classical counterparts in terms of computational efficiency. It became increasingly clear that quantum computers could offer solutions for specific computing problems efficiently. Notably, Simon and Shor made remarkable contributions by developing algorithms that demonstrated speed enhancements for particular problem sets, including the field of prime factorization and cryptography. Seth Lloyd further enriched the supremacy of quantum computers by introducing an algorithm for simulating a wide range of quantum systems on quantum computers.

In summary, quantum computing is getting better with time and has the potential to solve certain computing problems more efficiently than classical computers. This is shown by various quantum algorithms, which highlight how powerful quantum computing can be.

B. NOISY INTERMEDIATE-SCALE QUANTUM (NISQ)

From the beginning, there has been a doubt whether a quantum computer could surpass the capabilities of a classical computer. Many of these doubts originate from concerns about the complexity of quantum computer design and and difficulty of controlling quantum computation devices [13]–[15]. These concerns are primarily related to the concept of decoherence, where quantum systems interact with their environment and lose their quantum properties (superposition, entanglement, and, interference) over time affecting the outcome of quantum circuits. A controlled quantum environment has led to debates about achieving reliable quantum computers [16]. It also suggests that quantum computers can outperform classical if certain conditions are met [17].

While early noisy quantum computers have been used to implement algorithms such as Shor's, Grover's, and Deutsch–Jozsa's, the prevailing high error rates and noise prevent the scaling of these algorithms [18]–[21]. In order to achieve fault-tolerant quantum computation, substantial improvements are required in quantum computers to control and protect the qubits sufficiently for reliable algorithms. These improvements can be made with hardware modifications or the use of error-correcting codes. Shor introduced Quantum Error Correction (QEC) in 1995, showing that information from one logical qubit can be encoded onto multiple physical qubits, protecting it from errors [22]. Shor's work demonstrated the possibility of executing quantum computations reliably with noisy quantum hardware [23].

Further research has revealed that noise and quantum scaling relate to each other. If errors and noise are below a certain threshold, it's theoretically possible to scale up quantum computers to larger sizes [24], [25]. Many types of error-correcting techniques have been developed [26]–[28], but studies indicate that millions of physical qubits are needed to achieve useful quantum computers [29]. Despite this, various algorithms have claimed quantum supremacy, showcasing computations on quantum devices likely surpass classical computers' capabilities in a reasonable time frame. The research works of IBM, Xanadu, and Google's Quantum AI team are prominent accomplishments in the field of quantum computing [30]–[33]. While these achievements are significant, they have limitations in scaling up quantum computations due to noise and errors.

The term NISQ stands for "Noisy Intermediate-Scale Quantum." It refers to a phase in quantum computing where quantum computers are not yet completely error-corrected but are large enough to perform computations beyond classical computers' capabilities. NISQ devices are characterized by the presence of errors due to noise, but they are sufficiently reliable for solving certain problems more efficiently than classical computers [34]. This phase represents a transitional period in the advancement of quantum computing technology and quantum supremacy.

The remainder of this paper is organized in the following manner. Section II provides the fundamental concepts of qubits, superposition, entanglement, interference, and noise. Section III explores the building blocks of quantum circuits such as quantum gates and measurement. In Section IV, various types of quantum circuits are explained with numerical examples. Section V discusses quantum algorithms which include famous Shor's algorithm and Grover's algorithm. Section VI elaborates the utilization of quantum computing in machine learning. Popular quantum simulators and their features are explored in Section VII. Quantum hardware and its deployment requirements are discussed in Section VIII. Section IX discusses quantum computing applications for various fields, such as cryptography, optimization, machine learning, and finance. Lastly, Section X concludes the article by summarizing the fundamental concepts, algorithms and applications of quantum computing. It also motivates the researchers to apply this rapidly evolving technology in different fields.

II. QUANTUM COMPUTING FUNDAMENTALS

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

A. QUBITS

In classical computing, a bit is analogous to a binary light switch, capable of assuming only two discrete states: 0 or 1, without any intermediary values. In contrast, for quantum computing, a quantum bit (qubit) operates more like a dimmer switch. It possesses not just the 0 and 1 states but also the ability to exist in an intermediate state, which is a linear combination of the 0 and 1 states, weighted by specific coefficients. These coefficients are used to calculate the probability of measuring either the 0 or 1 state when measured.

1) Bra-ket notation

Qubit is a quantum computing particle that has a wave-like nature with wavefunction $\psi(x)$ that satisfies the Schrödinger equation. Theoretically, this wavefunction exists in an infinite dimensional Hilbert dual space [35]. Therefore, the state vector representing this wavefunction in Hilbert space requires an infinite dimensional vector notation. This infinite dimensional vector state of the qubit in Hilbert dual space is shown using Dirac's bra-ket notation, which was created by Paul Dirac in 1939 [36]. However, it can also be a finite-dimensional vector having two states, on/off or spin-up/spin-down, which can be shown in two-dimensional Hilbert space. In this notation, two-dimensional state vectors $|1\rangle$ (read ket one) and $|0\rangle$ (read ket zero) are used for qubit.

$$|0\rangle = 1|0\rangle + 0|1\rangle \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{1}$$

$$|1\rangle = 0|0\rangle + 1|1\rangle \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

In the equation (1), ket zero shows that the qubit is at an off or spin-down state. Here, the first element represents the probability amplitude of off or spin down, and the second element shows the probability amplitude of on or spin up. Probability amplitude can be a complex value and it is used to compute the probabilities of vector states. Additionally, in Dirac's notation, the bra is a complex conjugate transpose of a ket. For example, $\langle\phi|$ (read bra of ϕ) is a complex conjugate transpose vector of ket ψ. The inner product of these two vectors $\langle\phi|\psi\rangle$ is a scalar value [37]. The symbol "$|\rangle$" denotes a column vector, and is known as a "ket". The "bra" ($\langle|$) form is a row vector and it is shown below:

$$\langle 0| = 1\langle 0| + 0\langle 1| \rightarrow \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3}$$

$$\langle 1| = 0\langle 0| + 1\langle 1| \rightarrow \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{4}$$

The ket notation is widely used in quantum computing as bra-ket representation of the qubit. The following two states $(|0\rangle + |1\rangle)/\sqrt{2}$ and $(|0\rangle - |1\rangle)/\sqrt{2}$ are also commonly used in quantum calculations and these are sometimes written as $|+\rangle$ and $|-\rangle$, respectively.

A single qubit is also called a two-level quantum system because it is a linear combination of two state basis, 0 and 1. Below is the common form of a single qubit in bra-ket notation.

$$|v\rangle = v_0|0\rangle + v_1|1\rangle = v_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \tag{5}$$

Here v_o and v_1 are complex coefficients to measure probability amplitudes. The probability and the phase of each computational state basis for a qubit can be computed as follows:

For state basis $|0\rangle$ with complex coefficient $v_o = x + i * y$

$$probability\ amplitude = |v_o| \tag{6}$$

$$|v_o| = \sqrt{(x + i \cdot y) * (x - i \cdot y)} = \sqrt{x^2 + y^2} \tag{7}$$

$$probability = |v_o|^2 \tag{8}$$

$$phase(\text{rad}) = \tan^{-1} \frac{y}{x} \tag{9}$$

$$phase\ (\text{degree}) = phase\ (\text{rad}) * (180/\pi) \tag{10}$$

The probability amplitude is used to calculate the probability of each state basis of the qubit which helps in the measurement of the qubit state. Similarly, phase is used for quantifying *interference*. The concepts of measurement and interference are explained in the following sections of the paper. If the complex coefficients are normalized, then they represent the probability of the qubit for 0 and 1 state

$$|v_o|^2 + |v_1|^2 = 1 \tag{11}$$

This is known as the **normalization constraint** since all two-level systems must obey this quality to function as a qubit.

For two or multiple qubits, the tensor product (or Kronecker product) is used to compute the resultant states of the quantum system. The tensor product is denoted by the symbol $\otimes$. Let us consider two qubits $|a\rangle$ and $|b\rangle$ as

$$|a\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \quad \text{and} \quad |b\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \tag{12}$$

The tensor product of the two qubits is

$$|x\rangle = |a\rangle \otimes |b\rangle = |ab\rangle \tag{13}$$

$$|x\rangle = \begin{bmatrix} a_0 * \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ a_1 * \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{14}$$

$$|x\rangle = a_0 b_0|00\rangle + a_0 b_1|01\rangle + a_1 b_0|10\rangle + a_1 b_1|11\rangle \tag{15}$$

$$|x\rangle = x_0|00\rangle + x_1|01\rangle + x_2|10\rangle + x_3|11\rangle \tag{16}$$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

and the **normalization constraint rule** for the two qubits will be the same as follows:

$$|a_0b_0|^2 + |a_0b_1|^2 + |a_1b_0|^2 + |a_1b_1|^2 = 1 \tag{17}$$

Similarly for 3-qubits, if $|c\rangle = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$ then the tensor product of the three qubits is

$$|y\rangle = |ab\rangle \otimes |c\rangle = |abc\rangle \tag{18}$$

$$|y\rangle = \begin{bmatrix} a_0b_0 * \begin{bmatrix} c0 \\ c1 \end{bmatrix} \\ a_0b_1 * \begin{bmatrix} c0 \\ c1 \end{bmatrix} \\ a_1b_0 * \begin{bmatrix} c0 \\ c1 \end{bmatrix} \\ a_1b_1 * \begin{bmatrix} c0 \\ c1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0b_oc_0 \\ a_0b_oc_1 \\ a_0b_1c_0 \\ a_0b_1c_1 \\ a_1b_oc_0 \\ a_1b_oc_1 \\ a_1b_1c_0 \\ a_1b_1c_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \tag{19}$$

The same method will be used to combine n qubits, and normalization constraint rules for n-qubits will be given as in equation (20).

$$\sum |v_i|^2 = 1 \tag{20}$$

If we have n qubits, we will need to keep track of 2^n complex probability amplitudes. As we can see, these vectors grow exponentially with the number of qubits. This is the reason quantum computers with large numbers of qubits are so difficult to simulate in classical computers. A modern laptop can easily simulate a general quantum state of around 20 qubits, but simulating 100 qubits is too difficult even for the largest supercomputers.

2) Bloch sphere notation

The Bloch sphere is a mathematical representation of a given quantum state of a qubit, with which researchers can pinpoint and manipulate various such states within the sphere to their advantage. Three qubits $|1\rangle, |-\rangle$ and, $|\gamma\rangle$ are shown in Bloch sphere representation in Figure 1.

B. QUANTUM SUPERPOSITION

In classical computing, a bit possesses a binary nature, exclusively adopting either a state of 1 or 0. Correspondingly, in a 2-bit classical system, only one state can exist at a given time among four distinct states that is 00, 01, 10, and 11. This conceptual framework can be extended to n-bit classical systems with 2^n states but only one state exists at a given time representing the state of the classical system.

Conversely, in quantum computing, a single quantum bit (qubit) can exist in the state of 0, 1, or any linear combination of these states as shown in Figure 2. This phenomenon is called superposition which enables qubits to exist in a combination of the states. Upon measurement, the superposition collapses, and the final outcome is determined depending on the probability distribution of the qubit states. Quantum superposition is the ability of a qubit to be in multiple states simultaneously until it is measured.

C. QUANTUM ENTANGLEMENT

In classical computers, the state of a bit can vary independently, that is, the state of a bit is not influenced by the state of another bit. However, in quantum computing, the probability of a qubit state can be affected by the change of another qubit state probability. This phenomenon is called entanglement [38]. In quantum circuits, entanglement is created through quantum gates by performing specific operations on the qubits that result in inseparable states of qubits as shown in equations (21), (22), (23), and (24). Regardless of the physical distance between the entangled qubits, a change in one qubit state probability can change the probability distribution of all qubits in the entangled quantum system [39].

Quantum entanglement is a phenomenon that occurs when two or more particles become correlated in such a way that the state of one qubit is dependent on the state of the other qubit, regardless of the distance between them. If the state of one qubit changes in the entangled system, then the states of all other qubits will be affected. There are specific states in 2-qubit systems, which are called Bell's states or EPR (Einstein–Podolsky–Rosen) pairs, which exhibit entangled properties and cannot be written in separable states as given below:

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle) \tag{21}$$

$$|\phi^{'}\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle - |1\rangle|1\rangle) \tag{22}$$

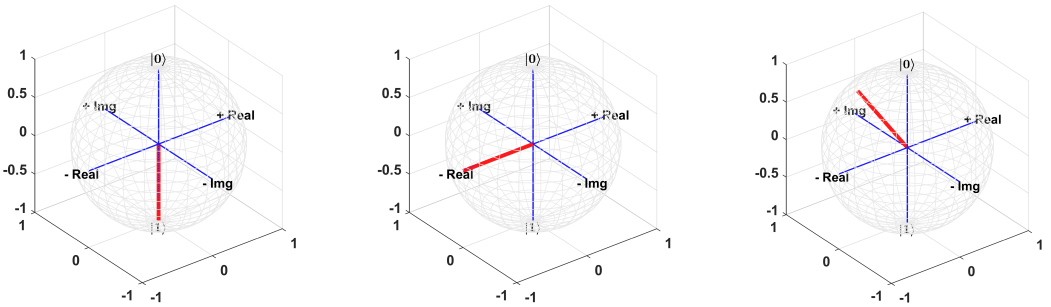
$$|\phi^{''}\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle) \tag{23}$$

$$|\phi^{'''}\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle - |1\rangle|0\rangle) \tag{24}$$

In the 2-qubit system, as shown in Figure 3a, each of the qubits is in a superposition state but qubits are not entangled. Therefore, the probabilities of all superposed qubit states are independent of each other. When these qubits are entangled as shown in Figure 3b, then the change in the probability of one qubit affects the probabilities of the entangled qubits. The blue color in Figure 3b shows that the two qubits are not independent particles. They are entangled and their states are dependent on each other. This entanglement results in the

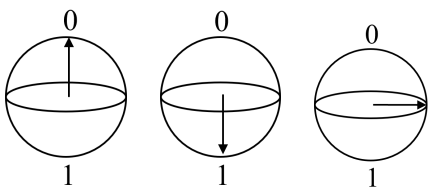
This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955



(a) $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (b) $|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ (c) $|\gamma\rangle = \begin{bmatrix} 0.8137 + 0.0i \\ 0.0 + 0.5812i \end{bmatrix}$

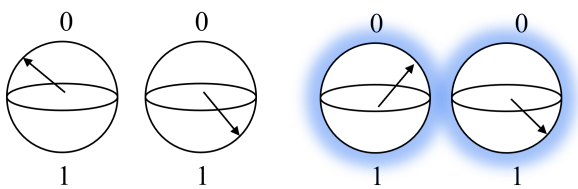
FIGURE 1: Bloch sphere representation of three different qubits.



(a) (b) (c)

FIGURE 2: Representation of qubit with the state of 0 in (a), 1 in (b), and superposed states in (c)

change of probability distribution of the state of the entangled quantum system, even if the entangled qubits are far away from each other.



(a) (b)

FIGURE 3: Two qubits in non-entangled (a) and entangled quantum states (b)

D. QUANTUM INTERFERENCE

Qubit is represented with bra-ket notation or Bloch sphere but this is just a mathematical representation of the qubit state. In reality, the qubit has a wave-like nature that is described by a quantum wavefunction satisfying the Schrödinger equation as shown in Figure 4. A wavefunction is a mathematical description of the quantum state that consists of complex probability amplitudes, and the corresponding probabilities of quantum system states.

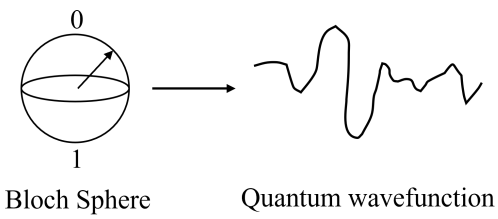


FIGURE 4: Wave-like nature of the qubit

When we have multiple qubits, their wavefunctions are added together to give an overall wavefunction describing the resultant states of a quantum system. This adding process of wavefunctions is called interference. It is a fundamental phenomenon that arises from the wave-like nature of quantum particles, such as electrons or photons and it distinguishes quantum systems from classical systems.

In quantum computing, when two quantum wavefunctions overlap, they can interfere with each other constructively or destructively. This results in a change in the resultant wavefunction of the quantum system that affects the probability distribution of its quantum states as shown in Figure 5.

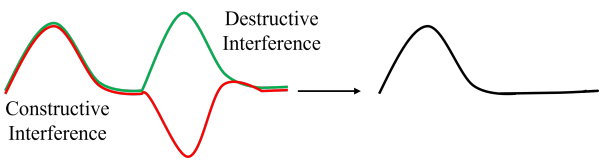


FIGURE 5: Quantum interference in two quantum wavefunctions

Interference can also be a challenge in quantum computing due to the phenomenon of decoherence. Decoherence is the loss of quantum coherence, which is the property of a quantum state to maintain its superposition and entanglement, due to the interactions with environment and thermalization. This loss of coherence leads to a breakdown of interference effects and making quantum computation error-prone. Quantum er-

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

ror correction techniques are used to mitigate the impact of the external environment and preserve the delicate quantum interference necessary for quantum computation.

Overall, interference is a foundational concept in quantum computing, allowing quantum systems to perform computations by updating the probability distributions of the quantum states. This concept solves certain problems in ways that are not achievable using classical computing methods.

E. QUANTUM NOISE

Quantum noise refers to the uncertainty and fluctuations that arise in quantum systems due to the probabilistic nature of quantum mechanics. It is a challenge in quantum systems even at low temperatures.

In classical systems, noise is often associated with random variations in signals or disturbances caused by external factors. When a quantum system is in a superposition state, its outcome upon measurement is not deterministic but is determined by the probability distribution of the quantum states. Noise and error can affect the outcome due to the quantum system's interaction with the external environment. It can lead to loss of quantum properties (superposition, entanglement, and, interference) over time affecting the outcome of quantum circuits.

Quantum noise has several manifestations in quantum systems, and it can impact various aspects of quantum computing. Some common examples of quantum noise include:

- Measurement Noise: When measuring a quantum system, the act of measurement can cause a quantum system to lose the quantum superposition and collapse the quantum state into one of its states, introducing uncertainty in the outcome due to the probabilistic nature of the measurement process.
- Decoherence: Interactions with the environment can cause quantum systems to lose quantum superposition, entanglement, and interference, affecting the performance of quantum algorithms.

Quantum noise poses a significant challenge for quantum computing. To address this challenge, researchers have been working on quantum error correction techniques, which are essential for preserving the quantum states against the detrimental effects of measurement noise and decoherence.

III. BUILDING BLOCKS OF QUANTUM CIRCUITS

A. HOW YOU PERFORM COMPUTATION WITH QUBITS

Practically all classical computers work in a similar way. They use a group of bits to store information in a binary form within memory. The state of these bits can be altered using logical gates like AND, OR, NOT, and NAND gates.

Quantum computing utilizes a model similar to classical computing known as the gate model or circuit model as shown in Figure 6. Within this framework, a set of qubits is used and their states can be superposed and entangled with each other. Multiple gates are available for conducting operations on these qubits, thereby altering the probability distribution of states within a quantum circuit. Quantum algorithms are constructed by applying a sequence of gates to qubits in a specific order. Ultimately, the measurement is executed at the end of the quantum circuit, collapsing the superposition states and revealing the quantum circuit's state based on the probability distribution of the states.

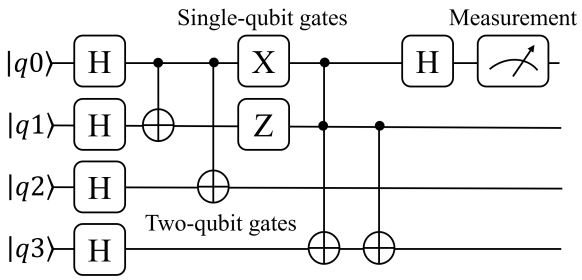


FIGURE 6: Quantum circuit example

B. QUANTUM GATES

Quantum gates are basic building blocks in quantum computing that manipulate the state of the qubits in quantum circuits. They are analogous to classical logic gates in classical computing but leverage the principles of quantum mechanics for processing quantum information. Quantum gates are summarized in Table 1 with symbol and matrix representation.

1) Identity Gate

The identity gate is a single-qubit gate. It does not modify the state of a qubit; therefore, it is represented by an identity matrix as shown in equation (25).

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \rightarrow \quad \boxed{\text{I}} \tag{25}$$

The identity gate is helpful during mathematical computation. This gate is used to compute the resultant gate matrix in multi-qubit circuits. The identity matrix is an involutory matrix, meaning that the identity matrix is equal to its inverse matrix as shown in equation (26).

$$I = I^{-1} \quad I^2 = I \tag{26}$$

2) Single-Qubit Pauli Gates

The Pauli gates $(X, Y, \text{and } Z)$ are based on Pauli matrices $(\sigma_x, \sigma_y, \sigma_z)$, which are useful to modify the state or phase of the qubit. Pauli gates are single-qubit gates and they rotate the state of the qubit around the x, y, and z axes of the Bloch sphere by π radians.

- Pauli-X gate is the quantum equivalent of the NOT gate for classical computers. It maps $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. It is also called a qubit-flip gate.
- Pauli-Y maps $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ to $-i|0\rangle$.
- Pauli-Z maps $|1\rangle$ to $-|1\rangle$ and leaves the state $|0\rangle$ unchanged. Due to this nature, Pauli Z is also called a phase-flip gate.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

These matrices are usually represented as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \rightarrow \quad \text{X} \tag{27}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \rightarrow \quad \text{Y} \tag{28}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \rightarrow \quad \text{Z} \tag{29}$$

The Pauli matrices are anti-commute as shown in equation (30). Two matrices A and B are considered to be anti-commute if AB = -BA.

$$ZX = -XZ = iY \tag{30}$$

The Pauli matrices are traceless matrices meaning the sum of the eigenvalues of Pauli matrices is zero.

$$eig(X) = eig(Y) = eig(Z) = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{31}$$

The Pauli matrices are also involutory, meaning that Pauli matrices are equal to their inverse matrices as shown in equation (32), and the square of the Pauli matrices is the identity matrix as shown in equation (33),.

$$X = X^{-1} \quad Y = Y^{-1} \quad Z = Z^{-1} \tag{32}$$

$$X^2 = Y^2 = Z^2 = I^2 \tag{33}$$

3) Two-Qubit Controlled Gates

Controlled gates are two-qubit gates, where the first qubit acts as a control and the second qubit is a target qubit. The controlled-NOT gate (CNOT) also known as controlled-X gate (CX), operating on two qubits, performs the NOT operation on the second qubit (target qubit) only when the first qubit (control qubit) is $|1\rangle$ otherwise leaves the second qubit unchanged. It is represented by the matrix CNOT.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \rightarrow \quad \text{X} \tag{34}$$

Controlled gates can be generalized using the universal matrix U which is a single-qubit matrix.

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} \tag{35}$$

The controlled-U gate acting on two qubit can be defined as CU.

$$CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \tag{36}$$

where U is one of the Pauli matrices that is X, Y, and Z. Based on the respective Pauli matrices, we can have "controlled-X", "controlled-Y", or "controlled-Z" gates. The shortened symbols of these controlled gates are CX, CY, and CZ respectively.

Controlled-Y gate (CY) performs the Y operation on the second qubit only when the first qubit is $|1\rangle$, and otherwise leaves it unchanged.

$$CY = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix} \quad \rightarrow \quad \text{Y} \tag{37}$$

Controlled-Z gate (CZ) performs the Z operation (phase flip) on the second qubit only when the first qubit is $|1\rangle$, and otherwise leaves it unchanged.

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \rightarrow \quad \text{Z} \tag{38}$$

The Controlled gates are Hermitian matrices as shown in equation (39). A square matrix (complex or real) is said to be a Hermitian matrix if it is equal to its own conjugate transpose matrix.

$$CX = \overline{CX^T} \quad CY = \overline{CY^T} \quad CZ = \overline{CZ^T} \tag{39}$$

The Controlled gates are also involutory, meaning that Controlled gates matrices are equal to their inverse matrices as shown in equation (40), and the square of a Controlled gates matrices is the identity matrix as shown in equation (41).

$$CX = CX^{-1} \quad CY = CY^{-1} \quad CZ = CZ^{-1} \tag{40}$$

$$CX^2 = CY^2 = CZ^2 = I^2 \tag{41}$$

4) Single-Qubit Phase Shift Gates

The phase shift gates are quantum gates that introduce a phase shift to the quantum state of a qubit. These are one-qubit gates and these are used to alter the phase of the qubit's state without changing its probability amplitudes. They map the states $|0\rangle \rightarrow |0\rangle$ and $|1\rangle \rightarrow e^{i\varphi}|1\rangle$. The probability of measuring a $|0\rangle$ or $|1\rangle$ does not change after applying this gate. Generally, the phase shift gate is generally represented by the matrix P:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{42}$$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

where φ is the phase shift with the period 2π. Some common examples of phase shift gates are,

- T gate where $\varphi = \frac{\pi}{4}$

$$T = P(\frac{\pi}{4}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \quad \rightarrow \quad \text{T} \tag{43}$$

- S gate, though S notation is sometimes used for SWAP gate where $\varphi = \frac{\pi}{2}$

$$S = P(\frac{\pi}{2}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} \quad \rightarrow \quad \text{S} \tag{44}$$

- The Pauli-Z gate where $\varphi = \pi$

$$Z = P(\pi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{bmatrix} \quad \rightarrow \quad \text{Z} \tag{45}$$

5) Single-Qubit Hadamard Gate

The Hadamard or Walsh-Hadamard gate, named after Jacques Hadamard and Joseph L. Walsh, is applied on a single qubit. It creates superposition states with an equal probability distribution for a given qubit. It maps the state of the qubit as follows:

$$|0\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{46}$$

$$|1\rangle \rightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{47}$$

The Hadamard gate is represented by a matrix H:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \rightarrow \quad \text{H} \tag{48}$$

The Hadamard matrix is a traceless and involutory matrix as shown in equation (49) and equation (50) respectively.

$$eig(H) = \begin{bmatrix} -\sqrt{2} \\ \sqrt{2} \end{bmatrix} \tag{49}$$

$$H = H^{-1} \quad H^2 = I \tag{50}$$

6) Two-Qubit Swap Gate

A swap gate performs a swap operation between two qubits. It is a fundamental quantum gate used in quantum computing with the primary purpose of exchanging the states of two qubits.

$$SWAP|\text{qubit1}, \text{qubit2}\rangle = |\text{qubit2}, \text{qubit1}\rangle \tag{51}$$

It is represented by the matrix:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \rightarrow \tag{52}$$

The SWAP matrix is an involutory matrix as shown in equation (53).

$$SWAP = SWAP^{-1} \quad SWAP^2 = I \tag{53}$$

7) Three-Qubit Toffoli (CCNOT) Gate

The Toffoli gate is named after Tommaso Toffoli and is also known as CCNOT gate (Controlled-Controlled-NOT gate) or Deutsch gate D$(\pi/2)$. The Toffoli gate is like a CNOT gate with two control qubits and one target qubit. The target qubit will be inverted if the first and second qubits are in $|1\rangle$ state. It is represented by the matrix CCNOT as given below

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow \tag{54}$$

The CCNOT matrix is an involutory matrix as shown in equation (55).

$$CCNOT = CCNOT^{-1} \quad CCNOT^2 = I \tag{55}$$

C. MEASUREMENT

In quantum computing, measurement is a fundamental operation that provides a way to extract information from a quantum system. When a quantum measurement is performed on a qubit, it yields a classical result. This measurement outcome collapses the superposition states of the qubit and provides either 0 or 1 based on the probability distribution of the qubit states. Measurement provides information about the quantum state of the qubit at the time of measurement. Measurements convert multiple (superposition) probabilistic states to one absolute state and collapse other states and superpositions. This is known as the measurement problem of quantum mechanics.

IV. BASIC QUANTUM CIRCUITS

A quantum circuit is a series of qubits and gates. Qubits can be in superposed or entangled states. Gates are used to change the state of qubits. Several gates perform different operations that are summarized in Table 1. Quantum gates are represented in matrix form, and the qubit's states are denoted in vector notation. The overall state of the quantum system can be calculated using the matrix product between the gate's matrices and qubit vectors. Common examples of quantum circuits are summarized in Figure 7.

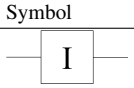
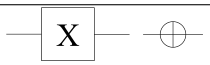
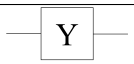
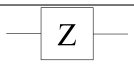
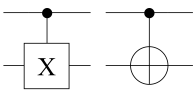
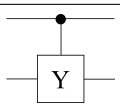
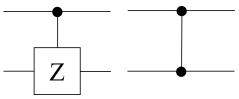
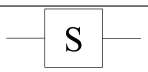
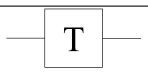
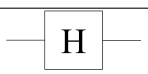
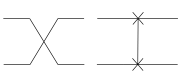
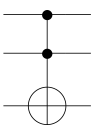
A. SINGLE-QUBIT GATE ON SINGLE QUBIT

In a single-qubit gate operated on a single qubit, the order of the gate matrix will be 2×2 which is equal to the total number

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

TABLE 1: Common Quantum Logic Gates

Common Quantum Logic Gates			
Operator Gate	Number of Qubit Gate	Symbol	Matrix
Identity gate	One-qubit		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Pauli-X (X)	One-qubit		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	One-qubit		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	One-qubit		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{bmatrix}$
Controlled-X (CNOT,CX)	Two-qubit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled-Y (CY)	Two-qubit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix}$
Controlled-Z (CZ)	Two-qubit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
Phase Gate-S	One-qubit		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}$
Phase Gate-T ($\pi/8$)	One-qubit		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$
Hadamard (H)	One-qubit		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
SWAP Gate	Two-qubit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli Gate (CCNOT, CCX, TOFF)	Three-qubit		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

of states for the 1-qubit quantum system, that is, two. The output state of the quantum system can be calculated using the matrix product of the gate matrix and the vector notation of the qubit. NOT gate and Hadamard gate are examples of single gate applied on a single qubit as shown in Figures 7a and 7b, respectively.

B. MULTIPLE-QUBIT GATE ON MULTIPLE QUBITS

In a multiple qubits gate operated on multiple qubits, the order of the gate matrix depends on the number of qubits it operates. For a multiple qubits gate applied on two qubits, the order of the gate matrix would be 4×4 which is equal to the total number of states for a 2-qubit quantum system. Similarly, for a multiple qubits gate operated on three qubits, the order of the gate matrix would be 8×8 which is equal to the total number of states for the 3-qubit quantum system and so on. The resultant input quantum state for multiple qubits can be calculated using the Kronecker product (or tensor product) of all qubits of the quantum system as shown in equation (58). For two or multiple qubits, the tensor product is used to find the resultant quantum states. The tensor product is denoted by the symbol $\otimes$.

Let us consider qubits $|a\rangle$ and $|b\rangle$:

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

$$|a\rangle = \begin{bmatrix} a_o \\ a_1 \end{bmatrix} \quad \text{and} \quad |b\rangle = \begin{bmatrix} b_o \\ b_1 \end{bmatrix} \tag{56}$$

The resultant input quantum state for multiple qubits can be calculated using the tensor product of all qubits of the quantum system.

$$|\alpha\rangle = |a\rangle \otimes |b\rangle \tag{57}$$

$$|\alpha\rangle = |ab\rangle = \begin{bmatrix} a_o * \begin{bmatrix} b_o \\ b_1 \end{bmatrix} \\ \\ a_1 * \begin{bmatrix} b_o \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_o b_o \\ a_o b_1 \\ a_1 b_o \\ a_1 b_1 \end{bmatrix} \tag{58}$$

$$|\alpha\rangle = a_o b_o|00\rangle + a_o b_1|01\rangle + a_1 b_o|10\rangle + a_1 b_1|11\rangle \tag{59}$$

$$|\alpha\rangle = \alpha_o|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle \tag{60}$$

The output state of the quantum system can be calculated using the matrix product of the gate matrix and vector notation of the input quantum state of the gate. SWAP gate and CNOT gate are examples of multiple-qubit gates applied on multiple qubits shown in Figures 7c and 7d respectively.

C. LOWER ORDER QUBIT GATE ON HIGHER # QUBITS

In this case, the order of the gate matrix is less than the total states of the quantum system. In such a scenario, two methods are possible. In the first method, the lower order qubit gates $g0$ and $g1$ are applied to the qubits individually. The output state $|C\rangle$ of the quantum system is computed using the Kronecker product of the output vectors $|c0\rangle$ and $|c1\rangle$ which are calculated from each lower order qubit gate as shown in equation (63).

$$|c0\rangle = g0 \times |a\rangle \tag{61}$$

$$|c1\rangle = g1 \times |b\rangle \tag{62}$$

$$|C\rangle = |c0\rangle \otimes |c1\rangle \tag{63}$$

The second method is to compute the resultant input state $|v\rangle$ of the quantum system using the Kronecker product of the qubits, that is:

$$|v\rangle = |a\rangle \otimes |b\rangle \tag{64}$$

$$G = g0 \otimes g1 \tag{65}$$

$$|C\rangle = G \times |v\rangle \tag{66}$$

The resultant quantum gate G can also be calculated using the Kronecker product of lower order qubit gates as shown in equation (65). The output quantum state can be found using the matrix product of the input quantum state and the corresponding gate matrix as shown in equation (66).

In Figure 7e, 1-qubit Hadamard and NOT gates are applied on a 2-qubit quantum system. The output state of the quantum system is computed using the second method as explained above. If the number of single-qubit gates are less than the total number of qubits, then identity gate can be used for the computation of the resultant multiple-qubit gate.

D. ENTANGLED CIRCUITS

In classical systems, only one state exists at a time but in quantum computing, all states can exist simultaneously. This combination of the states is called superposition in quantum systems. These superposed states continue if there is no external influence. However, when the superposed states are measured externally, the superposition collapses and the final output state is recorded based on the probability distribution of the qubit states. Superposed states of the qubit can also be entangled states or not. It can be understood with an example. Let $|v\rangle$ represents a 2-qubit quantum system as

$$|v\rangle = a_0 b_0|00\rangle + a_0 b_1|01\rangle + a_1 b_0|10\rangle + a_1 b_1|11\rangle \tag{67}$$

$$|v\rangle = |a\rangle \otimes |b\rangle = v_0|00\rangle + v_1|01\rangle + v_2|10\rangle + v_3|11\rangle \tag{68}$$

where $|v_0|^2$,$|v_1|^2$,$|v_2|^2$, and $|v_3|^2$ are the probabilities of $|v\rangle$ qubit states.
If $|v\rangle$ can be written as

$$|v\rangle = (a_0|0\rangle + a_1|1\rangle) \times (b_0|0\rangle + b_1|1\rangle) = |a\rangle \otimes |b\rangle \tag{69}$$

then $|v\rangle$ is not entangled because $|v\rangle$ can be written as separable states of qubits $|a\rangle$ and $|b\rangle$. The probabilities of qubit $|a\rangle$ and $|b\rangle$ states are independent of each other. However, if $|v\rangle$ cannot be written into separable states then these are entangled states. Figure 7f shows an example of an entangled quantum circuit with calculations and probabilities amplitude plot.

Non-Entangled System Example:
Let us take an example of a quantum system that is not entangled and see the effect of measurement on the probabilities of 2-qubit quantum states:

$$|\phi 1\rangle = \frac{1}{\sqrt{2}}(|0_a\rangle|0_b\rangle + |0_a\rangle|1_b\rangle) \tag{70}$$

the probability of $|a\rangle$ as $|0_a\rangle$ before measuring $|b\rangle$ is

$$prob(|0_a\rangle) = (\frac{1}{\sqrt{2}})^2 + (\frac{1}{\sqrt{2}})^2 = 1 \tag{71}$$

Suppose if we measure the state of $|b\rangle$ as $|1_b\rangle$ then the superposed states of $|\phi 1\rangle$ will be collapsed and we end up with

$$|\phi 1\rangle = |0_a\rangle|1_b\rangle \tag{72}$$

Now the probability of $|a\rangle$ as $|0_a\rangle$ after measuring $|b\rangle$ is

$$prob(|0_a\rangle) = (1)^2 = 1 \tag{73}$$

The probability of $|a\rangle$ states have not been affected by the change in $|b\rangle$ state therefore it is a non-entangled quantum system.

Entangled System Example:

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

Now let us take an example of a quantum system that is entangled and see the effect of measurement on the probabilities of 2-qubit quantum states:

$$|\phi 2\rangle = \frac{1}{\sqrt{2}}(|0_a\rangle|0_b\rangle + |1_a\rangle|1_b\rangle) \tag{74}$$

The probability of $|a\rangle$ as $|0_a\rangle$ before measuring $|b\rangle$ is

$$prob(|0_a\rangle) = (\frac{1}{\sqrt{2}})^2 = 0.5 \tag{75}$$

Suppose if we measure the state of $|b\rangle$ as $|1_b\rangle$ then the superposed states of $|\phi 2\rangle$ will be collapsed and we end up with

$$|\phi 2\rangle = |1_a\rangle|1_b\rangle \tag{76}$$

Now the probability of $|a\rangle$ as $|0_a\rangle$ after measuring $|b\rangle$ is

$$prob(|0_a\rangle) = 0 \tag{77}$$

Here, the probability of $|a\rangle$ state has been changed by the change in $|b\rangle$ state, therefore, it is an entangled quantum system.

V. QUANTUM ALGORITHMS

A. DEUTSCH-JOZSA ALGORITHM

The Deutsch-Jozsa algorithm is a quantum algorithm that solves the Deutsch-Jozsa problem, which involves determining whether a given Boolean function is balanced or constant. The algorithm can solve this problem with just one step, providing a significant speedup compared to classical algorithms. It was proposed by David Deutsch and Richard Jozsa in 1992 and is one of the early examples of a quantum algorithm that provides a significant speedup over classical algorithms. The algorithm generalizes the Deutsch algorithm to handle multiple degrees of freedom and can be derived from the quantum Fourier transform algorithm [40]. Additionally, the algorithm has been used as an educational experiment to demonstrate qubit fundamental concepts and algorithmic challenges in quantum science and technology [41]. Experimental implementations of the Deutsch-Jozsa algorithm have also been performed on IBM's quantum computer, showcasing its efficiency compared to classical techniques [42].

B. BERNSTEIN-VAZIRANI ALGORITHM

The Bernstein-Vazirani algorithm is a quantum algorithm designed to find a hidden binary string in a function. It uses quantum principles to extract information about the hidden string and determine it with fewer queries than classical algorithms, even in the presence of noise and imperfect equipment [43]. As the number of bits in the secret string increases, the probability of correctly guessing the string becomes less dependent on the type of disorder and more reliant on the center and spread of the disorder [44]. The classical algorithm, on the other hand, becomes inefficient for long strings, even in a noiseless scenario [45]. Overall, the Bernstein-Vazirani algorithm outperforms classical algorithms in most cases. This algorithm has potential applications in cryptography and demonstrates the advantages of quantum computing for certain problems.

C. SIMON'S ALGORITHM

Simon's algorithm is a quantum algorithm for obtaining the period of a vectorial Boolean function with polynomial time complexity. The algorithm achieves exponential speedup over classical algorithms. It has applications in quantum cryptanalysis and cryptography. Simon's algorithm has been utilized to study the autocorrelation spectrum and Walsh spectrum of Boolean functions [45]. It has also been applied in the design of a lightweight encryption algorithm called SIMON-GCM for IoT security, which combines the SIMON cipher block and Galois/Counter Mode (GCM) [46]. Additionally, this algorithm has been analyzed for different use cases in cryptanalysis [47].

D. SHOR'S ALGORITHM

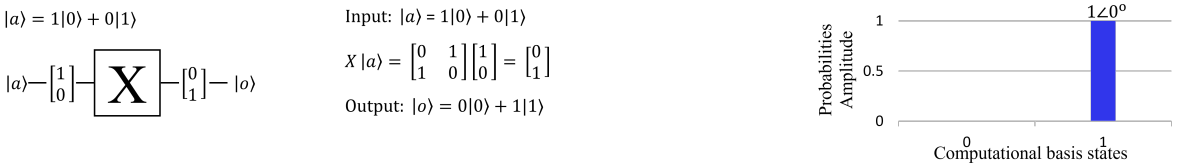
One of the most compelling quantum algorithm is Shor's algorithm [5], which is a quantum algorithm for finding the prime factors of an integer. It was developed in 1994 by the American mathematician Peter Shor [48]. It has the potential to break widely used encryption schemes, such as RSA, which rely on the difficulty of factoring large numbers. It leverages the quantum properties of superposition and entanglement to perform the factorization process exponentially faster than the best-known classical algorithms. In the integer factorization problem, given an integer $N = p \times q$ for some prime numbers p and q, the main goal is to find the prime factors p and q. The traditional trial division method has a time complexity of about $O(\sqrt{N})$ and the best classical algorithm is the general number field sieve (GNFS), which has a sub-exponential time complexity. For GNFS, the time complexity is roughly $O(exp((64/9)^{1/3} \times (logN)^{1/3} \times (log{*}logN)^{2/3}))$ [49]. Shor's quantum algorithm solves this problem substantially faster, in time $O(logN)^3)$. It's important to note that the implementation of Shor's algorithm becomes more complex for large numbers. In practice, Shor's algorithm demonstrates the potential of quantum computing to break widely used encryption methods, emphasizing the need for post-quantum cryptography techniques.

E. GROVER'S ALGORITHM

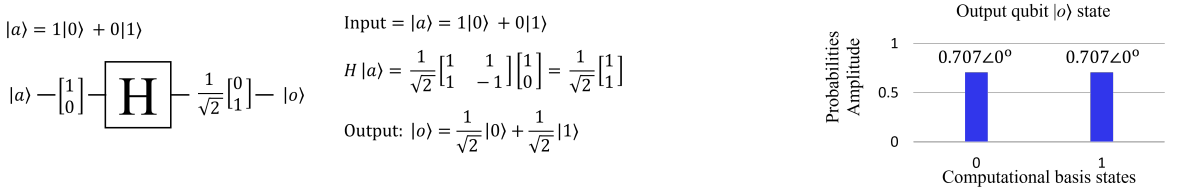
Grover's algorithm is a quantum algorithm designed to find a specific value within unsorted (or unstructured) databases or lists exponentially faster than classical algorithms. It provides a quadratic speedup, allowing it to search the target item in approximately $O(\sqrt{N})$ steps instead of the $O(N)$ queries required classically where N is the number of items in the database. Grover's algorithm is a fundamental demonstration of quantum computing's potential for solving certain search and optimization problems faster than classical computers. It was proposed by Lov Grover in 1996 [50]. Since classical algorithms for NP-complete problems (nondeterministic polynomial time problems) require exponentially many steps, and

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

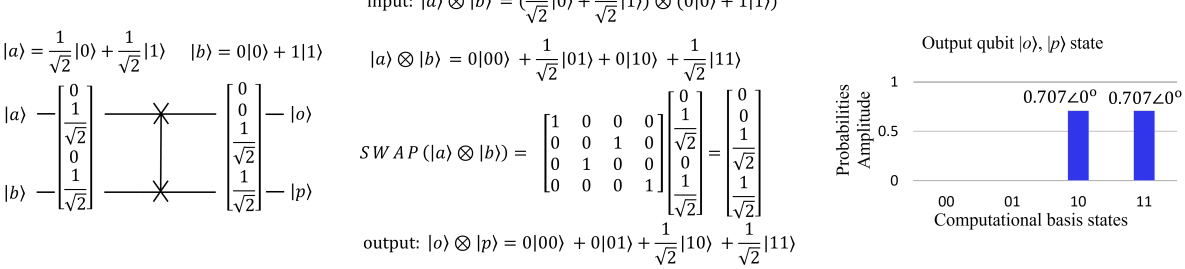
This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955



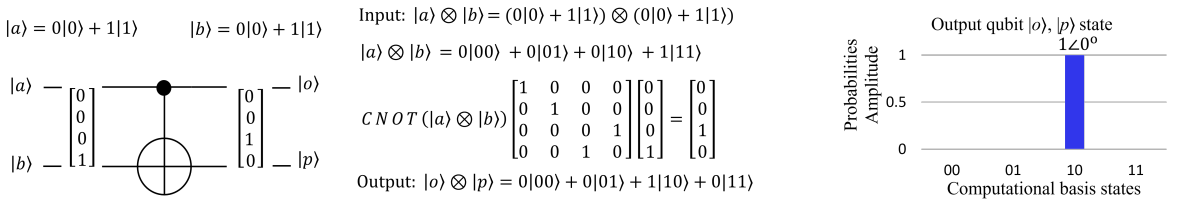
(a) Single-Qubit NOT gate on one qubit example



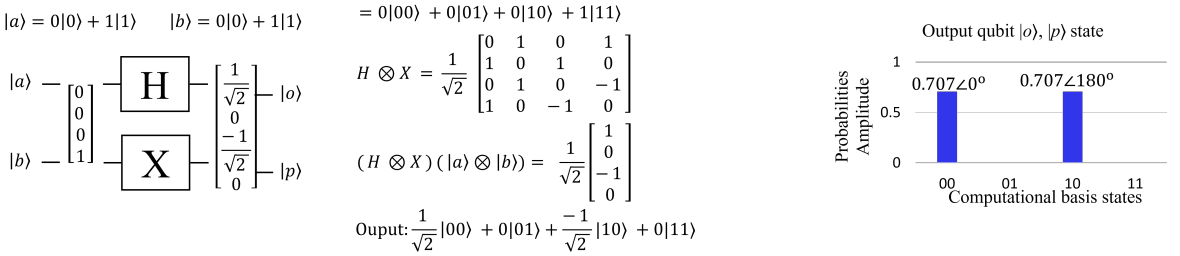
(b) Single-Qubit Hadamard gate on one qubit example



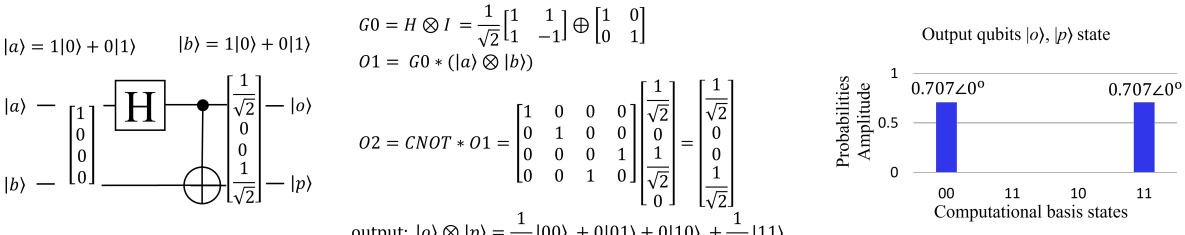
(c) Multiple-Qubit SWAP gate on two qubits example



(d) Multiple-Qubit Controlled-X gate on two qubits example



(e) Lower order gate matrices applied on higher order qubit states



(f) Two Qubits entanglement example

FIGURE 7: Basic circuits of quantum systems.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

Grover's algorithm provides at most a quadratic speedup; this suggests that Grover's algorithm by itself will not provide polynomial-time solutions for NP-complete problems [51].

VI. DESIGNING HYBRID CLASSICAL-QUANTUM MACHINE LEARNING (ML) MODELS

Machine learning (ML) involves creating models or algorithms that are capable of learning patterns within datasets. ML includes regression, classification, segmentation, object detection, etc. After learning from a dataset, these models become capable of predicting outcomes for unseen data. Originally, analytical approaches were used in Machine learning [52]; however, recently, heuristic-based methods have become more dominant due to the abundance of data and computational resources [53]. These methods have achieved substantial success in the field of deep learning [54], [55]. Deep learning methods develop data representation in the form of high-dimensional vectors using parameterized layers. Optimization techniques are used to fine-tune these parameters and fit the deep learning model.

Concurrently, academia and industry have witnessed a growing interest in quantum computing [34]. Quantum computing utilizes quantum mechanics principles such as superposition, entanglement, and quantum parallelism to improve the performance of classical machine learning algorithms. Quantum computers exhibit these novel behaviors that are often challenging to simulate using classical computers. These novel behaviors of quantum computers make them capable of solving certain problems more efficiently as compared to classical systems. Quantum computers have the potential to accelerate tasks such as optimization [56] and cryptanalysis [57].

A. FIRST GENERATION OF QUANTUM MACHINE LEARNING

Quantum computing is capable of enhancing the machine learning design process. A pivotal development leading to the utilization of quantum computers in machine learning is their ability to speed up linear algebra operations exponentially as state space grows. These QML algorithms, based on quantum-accelerated linear algebra techniques, form the first generation of QML algorithms, addressing the set of supervised and unsupervised learning tasks [11], [58]–[60]. These algorithms offer solutions that are faster compared to their classical counterparts for certain problems. For example, the quantum technique for solving systems of linear equations [61] is utilized for classification problems such as perceptron or linear regression training.

B. SECOND GENERATION OF QUANTUM MACHINE LEARNING

With the advent of NISQ processors, the second generation of QML has emerged [62]–[64]. Differing from the first generation, this new trend in QML relies on heuristic methods due to the increased computational capabilities of quantum hardware particularly in the field of deep learning [65]. These novel quantum algorithms utilize parameterized quantum circuits, often referred to as Parameterized Quantum Circuits (PQCs) or Quantum Neural Networks (QNNs) [66] in deep learning applications. Similar to classical deep learning, QNN parameters are optimized with respect to a cost function through black-box optimization heuristics [67] or gradient-based methods [68]. This optimization aims to facilitate the learning of data representations.

C. HYBRID MACHINE LEARNING

Quantum processors are still fairly small and noisy, therefore to improve machine learning performance effectively, NISQ processors will need to work with classical co-processors in hybrid mode. Scalability, training time, and model accuracy are the key factors in the field of deep learning [69] and these are the limitations of classical deep learning methods. These issues can also be addressed with the combination of classical and quantum processors. The abstract model of the classical-quantum deep neural network is shown in Figure 8.

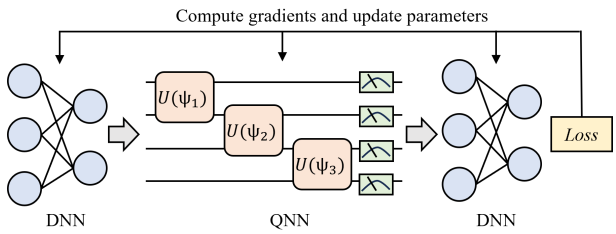


FIGURE 8: Abstract structure for hybrid classical-quantum model. ψ represents quantum model (QNN) parameters

Liu et al. have proposed a hybrid quantum-classical convolutional neural network (QCCNN), similar to convolutional neural networks (CNNs) in classical deep learning but adapted to quantum computing to improve the process of feature mapping [70]. Another study [71] has proposed a hybrid quantum-classical graph convolutional network (QGCNN) for learning high-energy physics data. The proposed framework has shown an advantage over classical multilayer perceptron and convolutional neural networks in the aspect of the number of parameters with comparable accuracy. This paper Liang et al. [72] have proposed the idea of a hybrid quantum–classical neural network with deep residual learning (Res-HQCNN). They have designed a residual block structure with a quantum neural network and the corresponding training algorithm.

VII. QUANTUM SIMULATORS

Quantum simulators are used to test and debug quantum circuits with realistic noise models. Following are five notable quantum simulators:

A. QISKIT AER

Qiskit Aer [73] is a high-performance quantum computing simulator provided by IBM. It provides interfaces to run quantum circuits with various noise models. It also includes

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

state vector and density matrix simulators for realistic simulations. It can also utilize graphical processing units (GPUs) to improve simulation performance.

B. QUTIP (QUANTUM TOOLBOX IN PYTHON)

QuTiP [74] is an open-source quantum computing framework for Python. It is designed for simulating the dynamics of open quantum systems. This framework depends on the Scipy, Cython, and Numpy packages. In addition, Matplotlib is used for graphical output. QuTiP is user-friendly and free of any licensing fees, therefore, it is considered suitable for learning quantum computing in the classroom.

C. CIRQ

Cirq [75] is an open-source interface designed for programming quantum computers. It is a Python software library developed for creating and simulating quantum circuits on quantum processors. Cirq also includes a simulator to run quantum algorithms for testing and debugging.

D. PROJECTQ

ProjectQ [76] is an open-source programming interface for quantum computers. It supports compilation framework for various types of quantum hardware such as IBM Quantum Experience chip, Azure Quantum, AWS Braket, AQT devices, or ion-trap quantum (IonQ) devices. ProjectQ also includes a simulator for testing and debugging quantum algorithms.

E. PYQUIL

pyQuil [77] is a Python library for writing quantum programs using Quil, the quantum instruction language developed at Rigetti Computing, Inc. It comes with a quantum virtual machine (QVM) for simulating and testing quantum circuits.

VIII. QUANTUM COMPUTING DEPLOYMENT REQUIREMENTS

Quantum computing promises to outperform classical computing by solving certain problems. Though still in the development phase, there are many approaches to quantum computing namely superconducting, photonic, trapped ion, neutral atoms, and quantum dots. Superconducting quantum computing is the widely used type of quantum supercomputing. Due to highly customized end use, and the limitations of the resources such as Quantum hardware, quantum experts believe that the design, delivery, and deployment of this type of computing can be challenging. The topology that is the most studied and vetted of the superconducting circuit approaches is the nearest-neighbor cavity coupled transmon qubits [78].

Quantum computing circuits operate at superconducting temperatures. The qubits are sensitive to environmental interference such as electromagnetic and thermal fluctuations; hence the system needs to be maintained at near absolute zero temperature. Depending on the type of quantum computing, a temperature of 10-20mK is required using a state-of-the-art dilution refrigerator, the heart of quantum computing infrastructure.

Qubits are created with superconducting materials. At extremely low (near absolute zero) temperatures, the superconducting circuits act as superconducting material as they carry current at low electrical resistivity. A Josephson junction is an assembly of two weakly coupled superconductors with a thin layer of insulator. At superconducting temperature, electric current flows through the Josephson junction [79] with no applied voltage. To generate a qubit, a linear capacitor made of a superconducting material and an insulator is tied together to superconducting wires with Josephson junction in the conducting loop to form an artificial atom which holds the qubit. At near absolute zero temperature, the qubit has no thermal energy in the surroundings, so it stays in a stable state. The qubits are then entangled using a microwave pulse with a range depending upon the strength of the magnetic field of the qubit or its resonance frequency. The quantum computer setup consists of logical and physical qubits. Physical qubits are the actual quantum bits implemented in quantum hardware. They are the primary building blocks of quantum computers, typically found in quantum systems like superconducting circuits, photons, or trapped ions. Physical qubits are susceptible to errors and noise due to the external environment. Several physical qubits or the actual quantum hardware are grouped together to form a single logical qubit. Logical qubits are a higher-level concept that represents quantum information that is protected against noise and errors. The logical qubits are used for error and noise detection.

The superconducting quantum computing cooling setup mainly consists of the following components and is shown in Figure 9,

- Dilution refrigerator (Cryostat with pulse tube cooler)
- Gas handling system with gas/liquid mixture pumps, valves and controls
- Liquid Nitrogen Dewars
- ^{3}He and ^{4}He storage tanks
- Cryo compressor
- Cryostat control panel

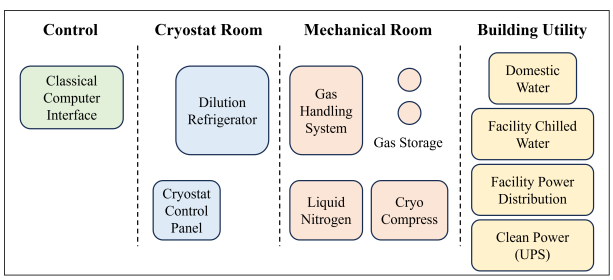


FIGURE 9: Quantum computing cooling infrastructure layout.

The quantum qubits need to be stored at near absolute zero temperatures for which a mixture of ^{3}He and ^{4}He along

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

with nitrogen is used in the dilution refrigerator [80]. Liquid nitrogen is used for 80K cooling and the He mixture is to achieve 2K cooling. The liquid nitrogen is also used to cool the ^{3}He return from the system and to remove any air impurities. Due to its sensitive nature, the presence of any AC and DC magnetic fields may be checked in the room housing the cryostat prior to its installation. Helium and its isotope used in the cryogenic cooling process may potentially leak and pose issues with the depletion of oxygen in space. A helium leak detector with *oxygen deficiency hazard alarm sensors* is recommended to be installed in quantum computing lab space.

The gas handling system (GHS) is an essential element of a quantum cooling system. It comprises a cabinet that contains the ^{3}He and ^{4}He mixture reservoir, pumps for circulation, valves, chilled water, liquid nitrogen connection, and controls (mostly pneumatic). The operation of GHS is considerably noisy, so it is recommended to place it in separate rooms alongside the cryostat compressors. Also, vibration isolators are recommended under the Dilution refrigeration assembly. The ambient space of the quantum setup as shown in Figure 9 may be maintained at 68-70F dry bulb and 30-50% relative humidity levels.

Unlike conventional high-performance computing (HPC) using graphics processing units (GPUs) and central processing units (CPUs) with thermal design power (TDP) of more than 1000W, a superconducting quantum system relies on low power consumption. For operational reliability of the quantum computer, clean power from a UPS source is generally recommended. The nominal voltage ranges are 120V-480V. Depending on the size of the qubit system, the Cryo compressor may need a dedicated chiller to provide lower-temperature cooling water generally at a higher delta $>$ 30F across the compressor.

The requirements for connectivity are generally scalable and flexible as the need and user community grows over time. A microwave amplifier, signal synthesizers, a multi-channel arbitrary pulse sequencer, 1 gigabyte of DDR3 SDRAM, and a gigabit Ethernet interface for high-speed data upload are generally required for a qubit setup.

There are many moving components in the cryogenics of the Quantum Infrastructure. Most of the components do not require frequent maintenance, however, proactive preventative maintenance can ensure the longevity of the entire quantum computing system. The parts are manufactured by multiple vendors; therefore assembly, integration, and commissioning of the entire system can be challenging. The quantum computing operators need to learn and obtain hands-on experience to ensure the optimum functioning of the setup.

IX. APPLICATIONS OF QUANTUM COMPUTING

Quantum computing has the potential to transform many areas including astrophysics, aerospace, pharmaceuticals, drug discovery, artificial intelligence, cybersecurity, and secure communication. It has the potential to perform certain tasks exponentially faster than classical computing. Quantum computing can be used to enhance algorithms and discover structures and patterns efficiently. In this section, we will dive into the applications of quantum computing in more detail. Quantum computing applications are summarized in Table 2.

A. CRYPTOGRAPHY

Cryptography involves the study of secure communication, and quantum computing has the potential to undermine numerous encryption techniques employed in classical cryptography, including RSA and elliptic curve cryptography. However, quantum computing can also be used to create new cryptographic protocols that are resistant to quantum attacks. For example, it can be used for secure key distribution over quantum channels, which have advantages over classical channels in terms of detecting eavesdropping [81]. Additionally, Quantum cryptography schemes can protect wireless sensor networks from attacks by hackers [92]. Quantum computing has emerged as a compelling complement to classical technologies for applications in security and communications [93]. In a nutshell, quantum computing offers promising solutions for cryptanalysis as well as for enhancing the security and efficiency of cryptographic systems.

B. OPTIMIZATION

Optimization is the process of finding the best solution to a given problem from a set of solutions. Quantum computers can efficiently solve certain optimization problems that are complicated tasks for classical computers. Quantum algorithms such as Grover search, quantum phase estimation, quantum annealing, quantum approximate optimization algorithm, and variational quantum eigensolver can be used to solve optimization problems [94]. Additionally, quantum computing can be applied in flight path optimization within the aerospace engineering domain, where quantum computing can tackle computational challenges and improve performance over classical algorithms [95].

C. CHEMISTRY

Quantum computing has numerous applications in pharmaceuticals and drug discovery. It can be used for generative chemistry and quantum chemistry simulations [96]. By using quantum computing, drug companies can potentially save time and money by accelerating the drug discovery process, leading to a more efficient and productive pharmaceutical industry [97]. Quantum computing can also reduce costs and time in drug development by decreasing the number of necessary biochemical experiments [98]. Quantum computing has been used in protein structure prediction, molecular docking, and quantum simulation [99]. Although current quantum devices are still susceptible to external noise and error, but hybrid quantum-classical techniques are well suited for drug discovery and development.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

TABLE 2: Summary of quantum applications with algorithm/model used.

Task	Algorithm/ Model	Study	Used for
Secure Key Distribution	Quantum channels	Rana et al. [81]	Secure communication and detection of eavesdropping.
Classification	Quantum Variational Classifier	Adhikary et al. [82]	Fisher's Iris, Sonar, and Wisconsin's Breast Cancer (WBC) datasets.
	Quantum inspired Nearest Mean Classifier	Sergioli et al. [83]	Idiopathic pulmonary fibrosis (IPF) dataset.
	Quantum Neural Network	Xiong et al. [84]	Image classification using MNIST digits and CIFAR-10 datasets.
Regression	Quantum Neural Network	Ngo et al. [85]	Li-ion battery degradation dataset.
	Continuous Variable - Quantum Neural Network (CV-QNN)	Kanimozhi et al. [86]	Surface plasmon resonance sensor dataset.
Reinforcement Learning	Hybrid-Quantum Proximal Policy Optimization	Rainjonneau et al. [87]	Satellite path planning and task decision using multiple satellites dataset.
	Hybrid Quantum Deep Q Network	Bar et al. [88]	Synthetic robot navigation dataset.
Dimension Reduction	Quantum PCA	Dri et al. [89]	Synthetic financial interest rate dataset.
Image Generation	Quantum–Classical GAN (QGANs)	Tsang et al. [90]	High-resolution image generation using MNIST digits and Fashion datasets.
Contrastive Learning	Quantum SimCLR Model	Jaderberg et al. [91]	Image processing using CIFAR-10 dataset.

D. MACHINE LEARNING

Quantum computing has shown potential to improve machine learning model accuracy. Quantum machine learning algorithms such as quantum neural networks, have been applied in the context of image classification for iris, sonar, breast cancer, idiopathic pulmonary fibros, CIFAR-10 and MNIST datasets [82]–[84]. These quantum models have shown benefits over classical models, including reduced training time and improved model accuracy. While there is still a need for further development of quantum hardware, quantum machine learning holds great potential for improving the efficiency of classical machine learning methods.

E. ENERGY

Quantum computing has various applications in the field of energy. One area of application is in power and energy systems, where quantum computing algorithms like Grover's algorithm can be applied to solve problems more efficiently than traditional algorithms. Another application area is battery profiling, where quantum computing can be used to model the degradation profile of the battery [85]. Additionally, quantum computing can contribute to simulating linear and nonlinear dynamics in fusion energy science applications, which can help in understanding wave-particle interactions and nonlinear plasma dynamics [100].

F. FINANCE

Quantum computing has numerous applications in finance. It can be used for financial interest rate modeling [89]. It has also been applied to develop financial models, such as churn prediction and credit risk assessment, where it has demonstrated better performance compared to traditional methods [101]. Quantum computing offers significant benefits in terms of computational speed and accuracy, making it a valuable tool in the finance field [102].

X. CONCLUSIONS AND FUTURE OF QUANTUM COMPUTING

This paper has offered a thorough exploration of the field of quantum computing, making its complexities more understandable. We have covered the fundamental concepts of quantum mechanics and the evolving quantum computing landscape with practical applications. This paper has discussed noisy intermediate-scale quantum (NISQ) along with quantum computing fundamentals, such as qubits' notations, quantum superposition, quantum entanglement, quantum interference, and quantum noise. We have elaborated building blocks of quantum systems, such as quantum gates, quantum measurements, and basic quantum circuits. We have provided an overview of various quantum algorithms, such as the Deutsch-Jozsa algorithm, Bernstein-Vazirani algorithm, Shor's algorithm, and Grover's algorithm. We have discussed the design of hybrid classical-quantum machine learning algorithms. We have further discussed the deployment requirements of superconducting quantum computing, the most prevalent form of quantum supercomputing.

Quantum computing has the ability to solve problems that are beyond the capabilities of any classical computer, a capability referred to as *quantum supremacy*. Achieving true quantum supremacy would be a major milestone in the field of quantum computing. However, with such powerful technology, it also raises issues such as privacy and security. Researchers will need to consider these issues as quantum computing technology grows.

In conclusion, we recognize the remarkable progress in quantum computing and its potential to address complicated problems with efficiency more than classical computers. This tutorial is intended to be a valuable resource for those beginning their journey into quantum computing and for researchers to stay informed about this exciting field. The future of quantum computing is promising, and we can

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

expect to see significant breakthroughs in scientific research, engineering, and technology in the coming years.

ACKNOWLEDGMENTS

This study acknowledges the University of Engineering and Technology (UET), Lahore for the support and affiliation with the author Muhammad Ali Shafique.

REFERENCES

[1] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell et al., "Superconducting quantum circuits at the surface code threshold for fault tolerance," Nature, vol. 508, no. 7497, pp. 500–503, 2014.

[2] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," Nature, vol. 549, no. 7671, pp. 195–202, 2017.

[3] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," Nature, vol. 536, no. 7614, pp. 63–66, 2016.

[4] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. Girvin, L. Jiang et al., "Extending the lifetime of a quantum bit with error correction in superconducting circuits," Nature, vol. 536, no. 7617, pp. 441–445, 2016.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM review, vol. 41, no. 2, pp. 303–332, 1999.

[6] L. G. Sandor Imre, Advanced quantum communications: an engineering approach. Wiley-IEEE Press, 2013.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[8] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. foundations of computer science." Conference Publications, 1997.

[9] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," 2004.

[10] T. Albash and D. A. Lidar, "Adiabatic quantum computation," Reviews of Modern Physics, vol. 90, no. 1, p. 015002, 2018.

[11] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," arXiv preprint arXiv:1307.0411, 2013.

[12] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, "Defining and detecting quantum speedup," science, vol. 345, no. 6195, pp. 420–424, 2014.

[13] I. L. Chuang, R. Laflamme, P. W. Shor, and W. H. Zurek, "Quantum computers, factoring, and decoherence," Science, vol. 270, no. 5242, pp. 1633–1635, 1995.

[14] W. G. Unruh, "Maintaining coherence in quantum computers," Physical Review A, vol. 51, no. 2, p. 992, 1995.

[15] B. Georgeot and D. L. Shepelyansky, "Quantum chaos border for quantum computing," Physical Review E, vol. 62, no. 3, p. 3504, 2000.

[16] G. Kalai, "The argument against quantum computers," Quantum, probability, logic: The work and influence of Itamar Pitowsky, pp. 399–422, 2020.

[17] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, "Focus beyond quadratic speedups for error-corrected quantum advantage," PRX Quantum, vol. 2, no. 1, p. 010103, 2021.

[18] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance," Nature, vol. 414, no. 6866, pp. 883–887, 2001.

[19] S. Gulde, M. Riebe, G. P. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt, "Implementation of the deutsch–jozsa algorithm on an ion-trap quantum computer," Nature, vol. 421, no. 6918, pp. 48–50, 2003.

[20] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin et al., "Demonstration of two-qubit algorithms with a superconducting quantum processor," Nature, vol. 460, no. 7252, pp. 240–244, 2009.

[21] T. Van der Sar, Z. Wang, M. Blok, H. Bernien, T. Taminiau, D. Toyli, D. Lidar, D. Awschalom, R. Hanson, and V. Dobrovitski, "Decoherence-protected quantum gates for a hybrid solid-state spin register," Nature, vol. 484, no. 7392, pp. 82–86, 2012.

[22] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," Physical review A, vol. 52, no. 4, p. R2493, 1995.

[23] P. Shor, "Proceedings of 37th conference on foundations of computer science, burlington, 1996," 1996.

[24] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation," Science, vol. 279, no. 5349, pp. 342–345, 1998.

[25] J. Preskill, "Reliable quantum computers," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1969, pp. 385–410, 1998.

[26] F. Gaitan, Quantum error correction and fault tolerant quantum computing. CRC Press, 2008.

[27] D. A. Lidar and T. A. Brun, Quantum error correction. Cambridge university press, 2013.

[28] G. G. La Guardia, "Quantum error correction," Quantum science and technology. Cham: Springer, 2020.

[29] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Physical Review A, vol. 86, no. 3, p. 032324, 2012.

[30] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, no. 7779, pp. 505–510, 2019.

[31] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu et al., "Quantum computational advantage using photons," Science, vol. 370, no. 6523, pp. 1460–1463, 2020.

[32] J. Chow, O. Dial, and J. Gambetta, "Ibm quantum breaks the 100-qubit processor barrier," IBM Research Blog, vol. 2, 2021.

[33] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins et al., "Quantum computational advantage with a programmable photonic processor," Nature, vol. 606, no. 7912, pp. 75–81, 2022.

[34] J. Preskill, "Quantum computing in the nisq era and beyond," Quantum, vol. 2, p. 79, 2018.

[35] B. Schumacher, "Quantum coding," Phys. Rev. A, vol. 51, pp. 2738–2747, Apr 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.51.2738

[36] P. A. M. Dirac, "A new notation for quantum mechanics," Mathematical Proceedings of the Cambridge Philosophical Society, vol. 35, no. 3, p. 416–418, 1939.

[37] J. Solem and L. Biedenharn, "Understanding geometrical phases in quantum mechanics: An elementary example," Foundations of physics, vol. 23, no. 2, pp. 185–195, 1993.

[38] V. Vedral and M. B. Plenio, "Basics of quantum computation," Progress in Quantum Electronics, vol. 22, no. 1, pp. 1–39, jan 1998. [Online]. Available: https://doi.org/10.1016%2Fs0079-6727%2898%2900004-4

[39] A. Steane, "Quantum computing," Reports on Progress in Physics, vol. 61, no. 2, pp. 117–173, feb 1998. [Online]. Available: https://doi.org/10.1088%2F0034-4885%2F61%2F2%2F002

[40] P. Aradyamath, N. Naghabhushana, and R. Ujjinimatad, "Quantum computing concepts with deutsch jozsa algorithm," JOIV: International Journal on Informatics Visualization, vol. 3, no. 1, pp. 59–68, 2019.

[41] R. De, R. Moberly, C. Beery, J. Juybari, and K. Sundqvist, "Multi-qubit size-hopping deutsch-jozsa algorithm with qubit reordering for secure quantum key distribution," in 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2021, pp. 473–474.

[42] A. N. Oliveira, E. V. de Oliveira, A. C. Santos, and C. J. Villas-Bôas, "Quantum algorithms in ibmq experience: Deutsch-jozsa algorithm," arXiv preprint arXiv:2109.07910, 2021.

[43] A. Gupta, P. Ghosh, K. Sen, and U. Sen, "Effects of noise on performance of bernstein-vazirani algorithm," arXiv preprint arXiv:2305.19745, 2023.

[44] P. Fernández and M. A. Martin-Delgado, "Homomorphic encription of the k= 2 bernstein-vazirani algorithm," arXiv preprint arXiv:2303.17426, 2023.

[45] X. Zhou, D. Qiu, and L. Lou, "Distributed exact quantum algorithms for bernstein-vazirani and search problems," arXiv preprint arXiv:2303.10670, 2023.

[46] X. Cheng, Y. Xu, K. Wang, Y. Zhang, B. Li, and Z. Zhang, "Lightweight and flexible hardware implementation of authenticated encryption algorithm simon-galois/counter mode," International Journal of Circuit Theory and Applications.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

[47] X. Bonnetain, "Tight bounds for simon's algorithm," in Progress in Cryptology–LATINCRYPT 2021: 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6–8, 2021, Proceedings 7. Springer, 2021, pp. 3–23.

[48] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in Proceedings 35th annual symposium on foundations of computer science. Ieee, 1994, pp. 124–134.

[49] J. P. Buhler, H. W. Lenstra, and C. Pomerance, "Factoring integers with the number field sieve," in The development of the number field sieve. Springer, 1993, pp. 50–94.

[50] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212–219.

[51] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information. Cambridge university press, 2010.

[52] K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.

[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[54] A. Krizhevsky, "Advances in neural information processing systems," (No Title), p. 1097, 2012.

[55] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[56] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," arXiv preprint arXiv:1411.4028, 2014.

[57] P. Shor, "35th annual symposium on foundations of computer science, 1994 proceedings," 1994.

[58] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," Nature Physics, vol. 10, no. 9, pp. 631–633, 2014.

[59] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," Physical review letters, vol. 113, no. 13, p. 130503, 2014.

[60] I. Kerenidis and A. Prakash, "Quantum recommendation systems," arXiv preprint arXiv:1603.08675, 2016.

[61] B. Duan, J. Yuan, C.-H. Yu, J. Huang, and C.-Y. Hsieh, "A survey on hhl algorithm: From theory to application in quantum machine learning," Physics Letters A, vol. 384, no. 24, p. 126595, 2020.

[62] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," Physical Review X, vol. 10, no. 2, p. 021067, 2020.

[63] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, "Variational ansatz-based quantum simulation of imaginary time evolution," npj Quantum Information, vol. 5, no. 1, p. 75, 2019.

[64] Z. Jiang, J. McClean, R. Babbush, and H. Neven, "Majorana loop stabilizer codes for error mitigation in fermionic quantum simulations," Physical Review Applied, vol. 12, no. 6, p. 064041, 2019.

[65] M. Mohseni, P. Read, H. Neven, S. Boixo, V. Denchev, R. Babbush, A. Fowler, V. Smelyanskiy, and J. Martinis, "Commercialize quantum technologies in five years," Nature, vol. 543, no. 7644, pp. 171–174, 2017.

[66] H. Chen, L. Wossnig, S. Severini, H. Neven, and M. Mohseni, "Universal discriminative quantum neural networks," Quantum Machine Intelligence, vol. 3, pp. 1–11, 2021.

[67] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, "Learning to learn with quantum neural networks via classical neural networks," arXiv preprint arXiv:1907.05415, 2019.

[68] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, "Stochastic gradient descent for hybrid quantum-classical optimization," Quantum, vol. 4, p. 314, 2020.

[69] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa et al., "Tensorflow quantum: A software framework for quantum machine learning," arXiv preprint arXiv:2003.02989, 2020.

[70] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, "Hybrid quantum-classical convolutional neural networks," Science China Physics, Mechanics & Astronomy, vol. 64, no. 9, p. 290311, 2021.

[71] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Hybrid quantum-classical graph convolutional network," arXiv preprint arXiv:2101.06189, 2021.

[72] Y. Liang, W. Peng, Z.-J. Zheng, O. Silvén, and G. Zhao, "A hybrid quantum–classical neural network with deep residual learning," Neural Networks, vol. 143, pp. 133–147, 2021.

[73] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.

[74] J. J. P.D. Nation. Qutip. [Online]. Available: https://qutip.org/

[75] C. Developers, "Cirq," Zenodo, 2023.

[76] Projectq - an open source software framework for quantum computing. [Online]. Available: https://github.com/ProjectQ-Framework/ProjectQ

[77] R. S. Smith, M. J. Curtis, and W. J. Zeng, "A practical quantum instruction set architecture," 2016.

[78] X. Jin, K. Cicak, Z. Parrott, S. Kotler, F. Lecocq, J. Teufel, J. Aumentado, E. Kapit, and R. Simmonds, "Versatile parametric coupling between two statically decoupled transmon qubits," arXiv preprint arXiv:2305.02907, 2023.

[79] R. Newrock, C. Lobb, U. Geigenmüller, and M. Octavio, "The two-dimensional physics of josephson-junction arrays," SOLID STATE PHYSICS-NEW YORK-ACADEMIC PRESS-, vol. 54, pp. 266–512, 2000.

[80] H. Dang, R. Zha, J. Tan, T. Zhang, J. Li, N. Li, B. Zhao, Y. Zhao, H. Tan, and R. Xue, "Investigations on a 3.3 k four-stage stirling-type pulse tube cryocooler. part b: Experimental verifications," Cryogenics, vol. 105, p. 103015, 2020.

[81] H. Rana and N. Verma, "Enhanced quantum key distribution using hybrid channels and natural random numbers," arXiv preprint arXiv:2007.14298, 2020.

[82] S. Adhikary, S. Dangwal, and D. Bhowmik, "Supervised learning with a quantum classifier using multi-level systems," Quantum Information Processing, vol. 19, pp. 1–12, 2020.

[83] G. Sergioli, G. Russo, E. Santucci, A. Stefano, S. E. Torrisi, S. Palmucci, C. Vancheri, and R. Giuntini, "Quantum-inspired minimum distance classification in a biomedical context," International Journal of Quantum Information, vol. 16, no. 08, p. 1840011, 2018.

[84] H. Xiong, X. Duan, Y. Yu, J. Zhang, and H. Yin, "Image classification based on quantum machine learning," in 2023 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), 2023, pp. 891–895.

[85] A. P. Ngo, N. Le, H. T. Nguyen, A. Eroglu, and D. T. Nguyen, "A quantum neural network regression for modeling lithium-ion battery capacity degradation," in 2023 IEEE Green Technologies Conference (GreenTech), 2023, pp. 164–168.

[86] T. Kanimozhi, S. Sridevi, M. Valliammai, J. Mohanraj, and V. Kumar, "Quantum regression model for the prediction of surface plasmon resonance sensor behaviour," in 2022 Workshop on Recent Advances in Photonics (WRAP). IEEE, 2022, pp. 1–2.

[87] S. Rainjonneau, I. Tokarev, S. Iudin, S. Rayaprolu, K. Pinto, D. Lemtiuzhnikova, M. Koblan, E. Barashov, M. Kordzanganeh, M. Pflitsch, and A. Melnikov, "Quantum algorithms applied to satellite mission planning for earth observation," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 7062–7075, 2023.

[88] N. F. Bar, H. Yetis, and M. Karakose, "An approach based on quantum reinforcement learning for navigation problems," in 2022 International Conference on Data Analytics for Business and Industry (ICDABI), 2022, pp. 593–597.

[89] E. Dri, A. Aita, T. Fioravanti, G. Franco, E. Giusto, G. Ranieri, D. Corbelletto, and B. Montrucchio, "Towards an end-to-end approach for quantum principal component analysis," in 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), vol. 02, 2023, pp. 1–6.

[90] S. L. Tsang, M. T. West, S. M. Erfani, and M. Usman, "Hybrid quantum–classical generative adversarial network for high-resolution image generation," IEEE Transactions on Quantum Engineering, vol. 4, pp. 1–19, 2023.

[91] B. Jaderberg, L. W. Anderson, W. Xie, S. Albanie, M. Kiffner, and D. Jaksch, "Quantum self-supervised learning," Quantum Science and Technology, vol. 7, no. 3, p. 035005, 2022.

[92] A. Montanaro, "Quantum algorithms: an overview," npj Quantum Information, vol. 2, no. 1, pp. 1–8, 2016.

[93] V. Hassija, V. Chamola, A. Goyal, S. S. Kanhere, and N. Guizani, "Forthcoming applications of quantum computing: peeking into the future," IET Quantum Communication, vol. 1, no. 2, pp. 35–41, 2020.

[94] Y. Wang, J. E. Kim, and K. Suresh, "Opportunities and challenges of quantum computing for engineering optimization," Journal of Computing and Information Science in Engineering, vol. 23, no. 6, 2023.

[95] H. Makhanov, K. Setia, J. Liu, V. Gomez-Gonzalez, and G. Jenaro-Rabadan, "Quantum computing applications for flight trajectory optimization," arXiv preprint arXiv:2304.14445, 2023.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

[96] A. Pyrkov, A. Aliper, D. Bezrukov, Y.-C. Lin, D. Polykovskiy, P. Kamya, F. Ren, and A. Zhavoronkov, "Quantum computing for near-term applications in generative chemistry and drug discovery," Drug Discovery Today, p. 103675, 2023.

[97] V. Mahesh and S. Shijo, "Accelerating drug discovery with quantum computing," Evolution and Applications of Quantum Computing, pp. 175–181, 2023.

[98] P.-H. Wang, J.-H. Chen, Y.-Y. Yang, C. Lee, and Y. J. Tseng, "Recent advances in quantum computing for drug discovery and development," IEEE Nanotechnology Magazine, 2023.

[99] M. Avramouli, I. K. Savvas, A. Vasilaki, and G. Garani, "Unlocking the potential of quantum machine learning to advance drug discovery," Electronics, vol. 12, no. 11, p. 2402, 2023.

[100] H. M. Gray and K. Terashi, "Quantum computing applications in future colliders," Frontiers in Physics, vol. 10, p. 864823, 2022.

[101] K. A. Tychola, T. Kalampokas, and G. A. Papakostas, "Quantum machine learning—an overview," Electronics, vol. 12, no. 11, p. 2379, 2023.

[102] Y.-J. Chang, M.-F. Sie, S.-W. Liao, and C.-R. Chang, "The prospects of quantum computing for quantitative finance and beyond," IEEE Nanotechnology Magazine, 2023.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3362955

ALI SHAFIQUE is a Ph.D. candidate in the Department of Computer Science at Kansas State University. He also holds the position of research assistant in the ISCAAS laboratory. Ali Shafique earned his Bachelor of Science degree from the University of Engineering and Technology, Lahore, where he completed his studies between 2011 and 2015. He obtained his Master's degree from the same University from 2016 to 2017. Shafique has developed a keen interest in the field of efficient Machine Learning in resource-constrained systems, He is committed to conducting research in this area with the aim of advancing the current knowledge and understanding of these fields.

ARSLAN MUNIR (M'09, SM'17) is currently an Associate Professor in the Department of Computer Science at Kansas State University. He was a postdoctoral research associate in the Electrical and Computer Engineering (ECE) department at Rice University, Houston, Texas, USA from May 2012 to June 2014. He received his M.A.Sc. in ECE from the University of British Columbia (UBC), Vancouver, Canada, in 2007 and his Ph.D. in ECE from the University of Florida (UF), Gainesville, Florida, USA, in 2012. From 2007 to 2008, he worked as a software development engineer at Mentor Graphics Corporation in the Embedded Systems Division. Munir's current research interests include embedded and cyber-physical systems, secure and trustworthy systems, parallel computing, artificial intelligence, and computer vision. Munir received many academic awards including the doctoral fellowship from Natural Sciences and Engineering Research Council (NSERC) of Canada. He earned gold medals for best performance in electrical engineering, gold medals and academic roll of honor for securing rank one in pre-engineering provincial examinations (out of approximately 300,000 candidates). He is a Senior Member of IEEE.

IMRAN LATIF is currently serving as the Chief Operations Officer at the U.S. Department of Energy, Office of Science, at Brookhaven National Laboratory where he leads the infrastructure operations of high-performance computing centers supporting global cutting-edge research collaborations in physics, life sciences, quantum and computational sciences, and artificial intelligence. Mr. Imran has over 20 years of leadership experience delivering large-scale engineering, construction, and sustainability projects with great success for top companies including IBM, Microsoft, Google, Wall Street District, and many others. Mr. Imran is also a keynote speaker and has delivered expert talks at various tech and entrepreneur conferences worldwide, including DCD, BICSI, Super Computing, HEPIX, and CHEP. He completed his masters in engineering from The City University of New York. He holds professional engineering licenses in the states of New York and Texas along with several professional certifications including CEM, PMP, DCEP, CDCDP and LEED. He is also a mentor for the U.S. Federal Government-funded internship programs where he works with students from Ivy league universities on leading-edge scientific research projects. Mr. Imran is presently spearheading global research initiatives with leading academia and big tech companies on data center energy conservation. He is the founder and CEO of New York-based engineering design and consulting firm providing professional services to Fortune 500 clients within retail, healthcare, telecom, energy, and industrial sectors.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4