

Networked Quantum Services[†]

Laszlo Gyongyosi* and Sandor Imre

Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, 3 Muegyetem rkp., H-1111 Budapest, Hungary;
imre@hit.bme.hu

[†] Invited paper

* Correspondence author: gyongyosi@hit.bme.hu

Received date: 6 January 2025; Accepted date: 25 February 2025; Published online: 18 March 2025

Abstract: The intense growth of quantum computation and communication allows the development of advanced solutions and services. Networked quantum services are provided for the users via quantum computers and quantum networking. Here, we review the fundamental concepts and recent achievements of networked quantum services. We present a comprehensive study of the state of the art, the different technologies, platforms, and applications. We analyze the implementation basis and identify key challenges.

Keywords: quantum computing; quantum networking; quantum services; survey

1. Introduction

One of the main drivers of today's scientific and technological development is the constantly growing computing power. However, we often run into the limitations of current computing technology. Keeping Moore's law alive is not possible by relying on classical physics. Although quantum computers open up new horizons, their full development will take a long time. Therefore, it is of utmost importance to combine the capabilities of machines with a limited number of qubits.

Quantum networked devices are systems that leverage quantum mechanics to establish communication networks, which can be used for applications involving security and privacy, remote sensing, or distributed computational capabilities, including the quantum cloud, quantum artificial intelligence, and solving computationally hard problems by means of limited-qubit computers.

Quantum sensing vastly improves the accuracy of how we measure and interact with the world around us by sensing changes in electric and magnetic fields. Collecting data at the atomic level means extracting information from individual atoms instead of from the huge collections of atoms, as usual in classical systems. This allows quantum sensors to make our devices exponentially more accurate. Devices that use quantum sensing are also not subject to the same physical constraints as conventional sensors, allowing for exceptional reliability with less vulnerability to the signal jamming and other electromagnetic interference that is increasingly common with today's light- and sound-based data sensors. Quantum sensors can be applied to build devices—among others—for faster, more accurate, more reliable geolocation, more detailed and accurate medical diagnostic images with fewer potential side effects for patients, reliable detection, imaging, and mapping of underground environments and deeper, more active sensing of gravitational changes and tectonic shifts that can forewarn or trigger avalanches, earthquakes, volcanic eruptions.

Distributed quantum computing represents a new computing paradigm in which quantum processing tasks are spread across multiple quantum devices or nodes, thus allowing these devices to work collaboratively on complex problems. In distributed quantum computing, qubits are managed across different locations or devices, which enables parallel processing of quantum algorithms. This approach can harness the combined power of multiple quantum systems. As the workload is distributed, systems can share computational resources, which is especially useful when individual quantum devices have limited qubit counts or processing capabilities. Entanglement plays a crucial role both in well-known efficient communication protocols [1–3] and algorithms [4–6]. Distributed quantum computing relies on entanglement between qubits located in different devices; this allows for quantum information to be shared and manipulated across the network, thereby enhancing computational power and efficiency.

The impact of quantum networks on real-world scenarios relies on the utilization of advanced quantum phenomena, such as quantum superposition and quantum entanglement, which prove useful in achieving performance better than classical networks. Quantum networks use quantum channels [7] to transmit the superposed or entangled quantum states (i.e., qubits in practical scenarios) from a sender user to a receiver user. The quantum channels can preserve the state of the qubits over long distances, allowing the development of efficient quantum networking services.

In the physical layer, the quantum channels can be implemented by different physical systems, such as optical fibers of current telecommunication systems, wireless optical channels, atoms, and superconducting technology. The quantum channels also allow high-speed communications similar to classical networking, but with significantly extended possibilities in data communications (quantum teleportation, superdense coding, distributed quantum computing, entanglement-based communication protocols) and advanced security via QKD (quantum key distribution) and quantum cryptographic protocols compared to classical networking. These fundamentally new possibilities make it possible to overcome several issues and drawbacks of classical telecommunications. In real-world scenarios, quantum channels and networking protocols are integrated into the existing classical telecommunication infrastructure. Classical channels are used to provide side information for the quantum operations of quantum networks. These data include controlling information, synchronization, signaling and timing for performing quantum operations. A distributed quantum protocol aims to perform some communication steps between the network users or complete a computational task in a distributed manner. The distant quantum computers and devices are connected via a set of intermediate quantum repeaters. The quantum repeaters establish a set of entangled connections with different entanglement levels between the users. The network structure integrates optical fiber and wireless optical channels between the ground-to-ground quantum repeaters and free-space optical channels for ground-to-satellite communications.

However, in order to build a network from our currently available quantum devices, we still have to solve many technological problems. In order for these tools to work together, additional challenges must be faced.

1.1. Novel Contributions

The novel contributions of our review article are as follows.

- We provide a comprehensive review of networked quantum services, with a detailed overview of the different technologies, platforms, and applications.
- We study the core concepts of networked quantum services. We review recent advances in networked quantum services including networked quantum computers, networked quantum machine learning, networked quantum software, quantum programming languages, quantum cloud and quantum application programming interfaces.
- We present a comprehensive review of the state of the art of networked quantum services in a compact form and using well-structured, easy-to-access tables.
- We study the implementation basis of networked quantum services and identify key challenges and open problems for future research.

1.2. Structure of the Survey

The structure of this survey is as follows. Section 2 summarizes the related surveys. In Section 3, we comprehensively review the fundamental methods and recent research on networked quantum services. We study the networked quantum devices, the architecture of networked quantum computers, and networked quantum machine learning. Section 4 focuses on the quantum software of networked quantum services, the quantum programming languages, quantum application programming interfaces (APIs), and standardization approaches. Section 5 reviews the implementation basis. Finally, Section 6 concludes the survey, and outlines open problems and future research directions.

2. Related Work

Recent surveys related to quantum computing, quantum communications, and quantum software tools can be found in the literature. Memon et al. [8] reviewed the potential and current challenges of quantum computing along with important questions of quantum hardware development. Yang et al. [9] summarized the current issues in quantum computing and communications focusing on quantum machine learning, quantum communications and quantum internet, quantum neural networks, and quantum cryptography. Sahu et al. [10] reviewed the different quantum development tools and kits for quantum hardware and software. The survey of Singh et al. [11] also studied the available tools and technologies for quantum computing. Gyongyosi et al. [12] reviewed the results of quantum computing

technology. The survey of Gill et al. [13] studies the taxonomy and provides a systematic review on quantum computing.

Chae et al. [14] summarized the recent developments of qubits for quantum computing, while Bochkarev et al. [15] studied the application of quantum computing for discrete optimization. The survey of Li et al. [16] focused on quantum optimization and quantum machine learning. Kussyk et al. [17] discussed the methods of quantum circuit compilation for quantum computers of the NISQ (noisy intermediate scale quantum) era, along with the utilization of artificial intelligence methods in compilation. Upama et al. [18] overviewed the different quantum programming languages, quantum simulators, and quantum compilers. Massoli et al. [19] focused on quantum computing and quantum neural networks. The work of Ramezani et al. [20] reviewed the different quantum machine learning algorithms. The survey of Peral-Garcia et al. [21] presented a comprehensive overview and literature review on quantum machine learning. Tensor networks allow an efficient solution to simulate general quantum systems on classical computers and to study high-dimensional numerical problems. Garcia et al. [22] studied the computational applications of tensor network simulations. The survey of Ayral et al. [23] studied the connections between quantum many-body systems and quantum processors.

Moguel et al. [24] reviewed the current state of quantum software engineering and service-oriented quantum computing. Serrano et al. [25] studied quantum software components and platforms along with the quality assessment requirements for quantum software. The survey of Khan et al. [26] focused on the different architectures, frameworks, and tools for quantum software engineering. Jimenez-Navajas et al. [27] reviewed the actual status of quantum software development, the current trends, and requirements. The survey of Garhwal et al. [28] focused primarily on quantum programming languages. Dwivedi et al. [29] reviewed quantum software engineering and the quantum software development lifecycle.

Cuomo et al. [30] and Caleffi et al. [31] overviewed the current status of distributed quantum computing and quantum networking. Barrala et al. [32] presented a comprehensive overview of distributed quantum computing including the different quantum communication protocols and quantum networking aspects. The survey of Boschero et al. [33] studied the different applications and challenges of distributed quantum computing. Jones et al. [34] reviewed the utilization of distributed quantum computing for chemical applications. The survey of Pira et al. [35] reviewed the current state of the art in distributed quantum neural networks. Nguyen et al. [36] studied quantum cloud computing with the concepts, architecture models, software, and resource management. In [37], Moguel et al. reviewed the development and deployment of quantum services. The authors propose a comparison between different quantum computing service providers using a case study via empirical tests.

In [38], Phillipson studied the applications of quantum computing in telecommunications. The survey of Baseri et al. [39] focused on the security aspects of quantum networking, while Dutta et al. [40] reviewed the recent trends and challenges of quantum communications. Li et al. [41] overviewed the fundamentals of entanglement-assisted quantum networks, while Mehic et al. [42] and Popa et al. [43] reviewed the applications of quantum cryptography in telecommunications. The survey of Abane et al. [44] proposed a comprehensive overview on the different entanglement routing methods in quantum networks. Gyongyosi et al. [45] overviewed the fundamental attributes of the quantum internet, while in [7] Gyongyosi et al. presented a comprehensive overview on quantum channels. Wehner et al. [46] studied the current stage of the quantum internet along with the requirements of a reliable large-scale global quantum communication network. Li et al. [47] studied the protocols of the quantum internet.

Currently, there are no surveys that specifically address networked quantum services. The studies discussed above also did not focus exhaustively on the related technologies of networked quantum services.

2.1. Comparative Analysis

For a comparative analysis, the primary scope of the related works is summarized in Table 1.

Table 1. Comparison of the primary scope of the surveys.

Survey	Primary Scope
Abane et al. [44]	Quantum internet.
Ayral et al. [23]	Quantum computing.
Barrala et al. [32]	Distributed quantum computing.
Baseri et al. [39]	Quantum communication.
Bochkarev et al. [15]	Quantum computation.
Boschero et al. [33]	Distributed quantum computing.

Table 1 Continued.

Caleffi et al. [31]	Distributed quantum computing.
Chae et al. [14]	Quantum computation.
Cuomo et al. [30]	Distributed quantum computing.
Dutta et al. [40]	Quantum communication.
Dwivedi et al. [29]	Quantum development tools.
Garcia et al. [22]	Quantum computing, quantum machine learning.
Garhwal et al. [28]	Quantum programming languages.
Gill et al. [13]	Quantum computing.
Gyongyosi et al. [12]	Quantum computing.
Gyongyosi et al. [7]	Quantum communication.
Gyongyosi et al. [45]	Quantum internet.
Jimnez-Navajas et al. [27]	Quantum programming languages, quantum development tools.
Jones et al. [34]	Distributed quantum computing.
Khan et al. [26]	Quantum programming languages, quantum development tools.
Kusyk et al. [17]	Quantum computation, quantum machine learning.
Li et al. [47]	Quantum internet.
Li et al. [41]	Quantum communication.
Li et al. [16]	Quantum computation, quantum machine learning.
Massoli et al. [19]	Quantum computing, quantum machine learning.
Mehic et al. [42]	Quantum cryptography, quantum communication.
Memon et al. [8]	Quantum computation.
Moguel et al. [37]	Development and deployment of quantum services.
Moguel et al. [24]	Quantum programming languages, quantum development tools.
Nguyen et al. [36]	Quantum cloud computing.
Peral-Garcia et al. [21]	Quantum machine learning.
Phillipson [38]	Quantum computing, quantum communication.
Pira et al. [35]	Distributed quantum neural networks.
Popa et al. [43]	Quantum cryptography, quantum communication.
Ramezani et al. [20]	Quantum machine learning.
Sahu et al. [10]	Quantum computation, quantum development tools.
Serrano et al. [25]	Quantum programming languages, quantum development tools.
Singh et al. [11]	Quantum computation, quantum programming languages.
Upama et al. [18]	Quantum programming languages, quantum simulators.
Wehner et al. [46]	Quantum internet.
Yang et al. [9]	Quantum computation, quantum communication, quantum machine learning.

3. Architecture for Networked Quantum Services

Networked quantum services represent a novel approach to significantly increase computational power compared to the computational capabilities of classical distributed services. The main idea behind networked services is the efficient utilization of quantum resources, such as quantum superposition and entanglement in distributed computations. The services could also use efficient quantum communications between the users and classical communications for the distribution of side information between the quantum nodes.

In practical scenarios, the quantum advantage of networked quantum services over classical distributed computations can be improved through the unconditionally secure quantum communications between the parties. Quantum entanglement applications in various protocols significantly extend the possibilities of classical networking. The fundamentals of quantum networking and secure quantum protocols, like QKD, provide the background for the advanced security of networked quantum services. The high-speed and high-fidelity quantum channels provide reliable high-speed quantum data transmission for distributed quantum computations. The scalability of networked services allows the use of quantum devices with a few qubits and quantum circuits for implementing efficient quantum computations in a distributed manner.

The quantum internet [45] provides a large-scale quantum network to implement networked quantum services in experiments. The quantum internet utilizes quantum channels (mainly optical fibers or free-space optical channels), quantum repeaters, quantum devices with quantum memory, and quantum computers to provide scalable high-speed networked services with unconditional security. The building blocks of the quantum internet extend the advanced properties of quantum communications to realize distributed services using quantum computer clusters (clusters of quantum devices with few quantum resources) and long-distance quantum communications.

Table 2 demonstrates the superior of networked quantum services to classical distributed computing with some practical applications of distributed quantum computing.

Table 2. Superiority of networked quantum services to classical distributed computing.

Reference	Result	Superiority
Ang et al. [48]	Multinode quantum computing architecture.	Fast distributed computations.
Barz et al. [49], Ruiting et al. [50], Mantri et al. [51]	Demonstration of blind quantum computing.	Enhanced privacy.
Cirac et al. [52]	Distributed phase estimation problem over noisy channels.	Fast distributed estimation of the phase of eigenvalues of unitary operators.
Collins et al. [53]	Nonlocal content of quantum operations.	Local implementation of non-local quantum gates in a distributed quantum computer.
Eisert et al. [54]	Resource-optimized protocols for non-local quantum gates.	Local implementation of non-local quantum gates in a distributed quantum computer.
Gidney et al. [55]	Factoring RSA integers via distributed Shor algorithm.	Exponential speedup in distributed integer factorization and in discrete logarithm problems.
Gyongyosi et al. [56]	Distributed multiple access QKD.	Enhanced security.
Gyongyosi et al. [57]	Distributed resource allocation.	Improved resource prioritization and balancing.
Gyongyosi et al. [58]	Distributed problem-solving.	Fast distributed computations for optimization problems.
Li et al. [59]	Distributed Deutsch-Jozsa algorithm.	Exponential speedup over distributed deterministic classical computers.
Neumann et al. [60]	Distributed quantum phase estimation.	Fast distributed estimation of the phase of eigenvalues of unitary operators.
Nguyen et al. [36]	Quantum cloud computing.	Improved security, faster distributed computations.
Shi et al. [61]	Quantum message passing interface.	Fast distributed computations for statistical and optimization problems, constraint-satisfaction and graph isomorphism problems.
Tan et al. [62]	Distributed Simon’s quantum algorithm.	Exponential speedup over distributed probabilistic classical computers.
Van Meter et al. [63]	Arithmetic on a distributed quantum computer.	Exponential speedup in distributed integer factorization and in discrete logarithm problems.
Xiao et al. [64]	Distributed quantum-classical factoring algorithm.	Exponential speedup in distributed integer factorization and in discrete logarithm problems.
Zhang et al. [65], Degen et al. [66]	Distributed quantum sensing.	Improved accuracy in distributed environment sensing and data acquisition.
Zhou et al. [67]	Distributed Bernstein-Vazirani algorithm.	Efficient distributed solution of black-box problems.
Zhou et al. [68]	Distributed quantum searching algorithm.	Quadratic speedup in searching of elements in unstructured databases.

The core concept of networked quantum services is to use several smaller quantum processing units (QPUs) in parallel for the processing of input quantum data, and then the output of the QPUs is combined via post-processing. In general, each QPU can contain a given number of smaller quantum circuits with limited quantum resources. The use of multiple QPUs allows a scalable architectural, since the connection of smaller distant QPUs in a distributed network allows greater processing capabilities. Networked quantum services can be classified according to their available quantum resources and communication channels. In all the different concepts, classical communication is always available between the distant QPUs for sharing side-information and for post-processing. However, quantum communication is not always available between the QPUs. As we will detail later, these circumstances lead to different conceptions and models of networked quantum services.

The concept of the quantum internet offers the most favorable architectural conditions for networked quantum services. In a quantum internet setting, the distant QPUs have access to quantum communication channels and quantum

resources for entanglement generation and distribution. The entanglement distribution procedure uses quantum repeaters between the distant QPUs and different protocols such as entanglement swapping. Quantum entanglement between the distant QPUs allows implementing different quantum communication protocols, such as quantum teleportation, quantum key distribution, superdense coding, etc. A hierarchical quantum internet architecture [69] for networked quantum services in compliance with the RFC 9340 [70] standard is depicted in Figure 1. The bottom layer integrates the QPUs and the quantum repeaters. The quantum devices communicate with a local domain controller via a quantum network and a classical network. This layer is also responsible for performing quantum operations, and it is referred to as the infrastructure plane. The middle and top layers are referred to as the control plane. The middle layer contains local domain controllers, which handle the entanglement generation and distribution between the QPUs. The top layer consists of a central controller, which uses a classical network for communication with the local domain controllers. The central unit controls the error correction¹ and the collection of global network information and is responsible for a single quantum local area network (Q-LAN). Throughout our manuscript the classical and quantum communications are depicted by dashed and solid-lined arrows in the figures.

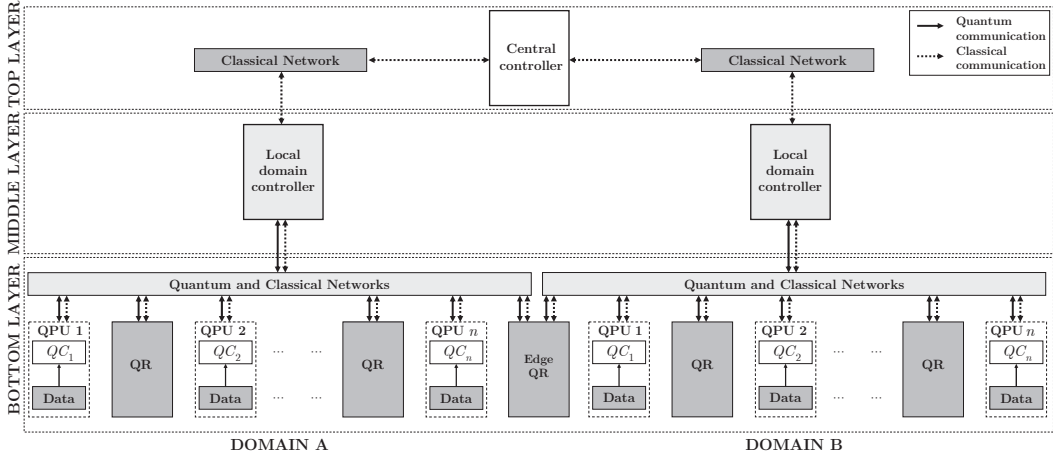


Figure 1. A hierarchical quantum internet architecture for networked quantum services in compliance with the RFC 9340 standard [69,70]. The n QPUs, QPU i , $i = 1, \dots, n$, process in parallel using their local input quantum data (Data) and local quantum circuits QC_i . The outputs of the QPUs are combined via post-processing. A given domain integrates n QPUs, $n - 1$ intermediate quantum repeaters (QR), and a local domain controller unit. The edge quantum repeater (Edge QR) logically separates the domains (Domains A and B). Classical and quantum communications are depicted by dashed and solid-lined arrows.

Since the general structure of a quantum internet is currently under development, the implementation of networked quantum services cannot assume that quantum entanglement between distant QPUs is available. Furthermore, the quantum circuits of the QPUs in the NISQ era are small circuits with a reduced number of qubits, called *shallow* circuits [75], in which the circuit depth increases no faster than logarithmically in the number of qubits. Shallow quantum circuits have a crucial significance in the NISQ era due to the limitations of current quantum hardware, such as the lack of error correction, limited resources, and high levels of disturbance and noise.

3.1. Networked Quantum Devices

The various technologies [10,76,77] that are available for the implementation of quantum computations (such as photonic [78], superconducting [79,80], silicon-spins [81], trapped neutral atom [82], giant atoms [83], trapped ion [84], topological [85], nitrogen-vacancy [86], nuclear magnetic resonance-based [87,88] implementations) require a generalized framework and guidelines for the development of hardware for quantum computation. The aim of the layer architecture is to serve as an abstract hardware-independent standard for scalable fault-tolerant quantum hardware [89].

¹ On a detailed description of quantum error correction, fault tolerance, and quantum memory limitations, we suggest references [71–74].

The layer architecture of a general (gate-model) quantum device is as follows:

- *Physical Layer*: a hardware layer for physical quantum states, physical operations, quantum hardware, and physical environment.
- *Virtual Layer*: conversion of physical systems to virtual primitives: virtual quantum states, virtual quantum gates, virtual quantum registers, and virtual measurement.
- *Error Correction Layer*: error correction and fault-tolerance methods, conversion of virtual primitives to logical primitives: logical quantum states, logical gates, logical quantum registers, and logical measurement.
- *Logical Layer*: logical abstractions for a universal quantum computer using fault-tolerant logical structures.
- *Application Layer*: a software layer for interface with the user for data input and execution of quantum algorithms.

Similar to the traditional OSI (Open Systems Interconnection) layer architecture, a layer architecture can also be defined for distributed quantum computation and networking with the aim of a hardware-independent standardization framework [32–34,90,91].

The general layer architecture of distributed quantum computation consists of the following layers:

- *Physical Layer*: a hardware layer for physical mechanisms to connect distant nodes (quantum entanglement generation and distribution, entanglement swapping, entanglement purification, quantum teleportation, specific operations [91]).
- *Network Layer*: a hardware layer for definition of procedures for establishing communications between distant quantum nodes.
- *Development Layer*: a software layer for mechanisms that allow the applications to be distributed and executed on the physically distributed quantum network (qubit mapping methods, partitioning, compilation, machine code generation, optimization [90]).
- *Application Layer*: a software layer for user interface for data input and execution of distributed quantum algorithms.

The layer structure of a quantum computer (gate-model quantum device) and the layered model for a distributed quantum computation are shown in Figure 2.

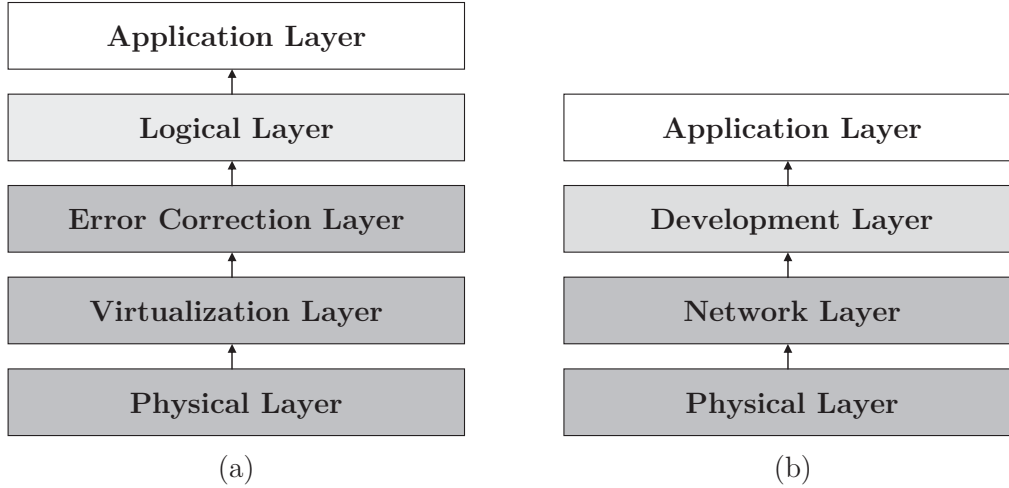


Figure 2. Layer architectures. (a) Layer architecture of a quantum computer (general gate-model quantum device). (b) A general layer architecture for distributed quantum computation.

While the layer structure can be referred to as a general layer architecture, we note that other layer approaches are also available for distributed quantum computation [32,41,45,70,92]. The main attributes of the different approaches to layer architectures are summarized in Table 3.

Table 3. Different layer architectures for distributed quantum computation.

Approach	No. of layers	Layer architecture
Quantum networking via bipartite entanglement [93,94]	5	Physical Layer, Link Layer, Network Layer, Transport Layer, Application Layer.
Quantum networking via multipartite entanglement [95]	4	Physical Layer, Connectivity Layer, Link Layer, Network Layer.
Quantum recursive network architecture [96,97]	5	Physical Layer, Link Layer, Remote State Composition Layer, Error Management Layer, Application Layer.
Multinode quantum computing architecture [48]	6	Physical Layer, Distillation Layer, Data Link Layer, Network Layer, Transport Layer, Application Layer.

3.2. Networked Quantum Computers

Quantum computing utilizes the fundamental concepts of quantum mechanics for computing, such as superposition, entanglement, and interference [80,98–102]. As the development of quantum computers evolve [13,80,98,99,101,103–119], the relevance of quantum computations has become more interpretable for problem-solving tasks [80,120–123]. Quantum computers could definitely help solve difficult computational problems [12,98,124–130].

As a first step, Shor’s factorization algorithm [131] has been implemented via a small NMR (nuclear magnetic resonance) device in 2001 [87]. In 2010, the quantum annealing computers have been announced by D-Wave,² with limited application possibilities. The quantum computing research at Google has been started in 2013.³ In 2016, they developed a simulated annealing algorithm [132] and also simulated the energy levels of the hydrogen molecule [133]. In 2019, Google announced a 53-qubit gate-model superconducting quantum computer [80], which, in theory, can solve optimization problems in only 3 min that would take a classical supercomputer 10,000 years. A theoretical framework of the experiments of Google’s 53-qubit quantum computer has been defined in [100]. Quantum computing research at IBM was started in 2016⁴ via the IBM quantum experience program. IBM has made it possible for users to utilize the IBM Quantum Cloud Services⁵ to run algorithms on the IBM quantum processor from the users’ computers. Between 2016 and 2019, IBM developed 5-qubit and 27-qubit quantum processors, while in 2020, they announced the 65-qubit IBM Quantum Hummingbird processor, a 127-qubit processor in 2021, and a 433-qubit processor in 2022. The 1121-qubit processor has been released in 2023.⁶ In 2020, Amazon launched the Amazon Braket Cloud Quantum Computing Service⁷ to provide quantum-based web services.

In the NISQ era [98,130], gate-model quantum computers have particular relevance since these architectures can be implemented in near-term settings [13,80,98–101,103–111,113–119,134–137]. In a gate-model quantum computer, the computational steps are realized via unitary gates. The gates are associated with a gate parameter value, while the computational problem fed into the quantum computer identifies an objective function [101,110–112,136,137]. For objective function examples, see [101,103,106,107,136,137]. The aim of problem solving is to maximize the objective function value via several iteration steps. Each iteration step includes the application of unitary gates, as well as a measurement of the resulting quantum states. From the measurement results, an averaged value can be determined to estimate the actual objective function value [101]. Many hybrid variational quantum algorithms also use gate-model circuits and classical objective functions [138–142], similar to the quantum approximate optimization algorithm (QAOA) [101,110].

The NISQ quantum devices are noisy, the decoherence and the imperfections of gates and measurements could easily destroy any possible quantum speedup on these devices. The lack of error-correction prevents *scaling*. Fault-tolerance requires precise physical control over the qubits (two-qubit gates with 99.99% fidelity) [8,98]. Another crucial problem of quantum computing is scalability. As the number of qubits and the number of quantum gates increase, so do the errors and noise. *Networked quantum computers* allow us to overcome the limitations of the current NISQ quantum hardware. In a networked setting, the quantum computations are realized by clusters of quantum nodes in a distributed manner. The aim of a distributed quantum computer architecture is to use multiple smaller quantum computers (equipped with a small number of qubits and quantum gates) instead of a single large quantum computer

² D-Wave. URL: <https://www.dwavesys.com>

³ Google Quantum AI. URL: <https://quantumai.google/>

⁴ A new way of thinking: The IBM quantum experience. URL: <http://www.research.ibm.com/quantum>

⁵ IBM Cloud, IBM. URL: <https://www.ibm.com/cloud>

⁶ IBM Quantum Roadmap. URL: <https://research.ibm.com/blog/ibm-quantum-roadmap>

⁷ Amazon Braket. URL: <https://aws.amazon.com/braket/>

to perform the same computational task. The total number of qubits and gates of the networked architecture can exceed the number of qubits and gates of the large quantum computer, allowing efficient scalability with low error rates and noise levels. The interconnection between the quantum nodes requires control mechanisms in the hardware and software layers.

In a networked quantum computer architecture, the remote quantum processing units (QPUs) use classical and quantum channels to communicate with the classical network and quantum network. The quantum channel can be an optical link, free space optical channel, or other physical implementation (see also Section 5). In a large quantum computer, the levels of error and noise are high, and the coherence times are limited. In a distributed architecture, the QPUs are equipped with significantly fewer qubits and quantum gates, which reduces the errors and noise, and overcomes the problem of limited coherence times. It allows using simplified, low-complexity error-correction techniques in the nodes, and decreases the noise that arises from the imperfections of the quantum gates. Both attributes are crucial in an NISQ setting, where the quantum hardware and quantum resources are limited.

We discuss networked quantum computers using the following categorization:

- *Multichip*: an input computational problem is divided into several smaller quantum circuits for parallel processing. The quantum circuits can work independently without the need for direct communication between them. The smaller quantum circuits fit in one QPU, no circuit decomposition is needed. The quantum communication between the quantum circuits is implementation-specific, and classical communication is always available. If only classical communication is available, it is also called *embarrassing parallelism* [32] (Table 4).
- *Circuit distribution*: a large quantum circuit requires a decomposition into smaller quantum circuits. The smaller circuits are mapped to distant QPUs (nodes) using quantum communications, quantum resources, and classical communications. Quantum and classical communication is available between the nodes (Table 5).
- *Circuit splitting*: a large quantum circuit is split into smaller quantum circuits between distant quantum nodes. Quantum communication is not available, classical communication is available (Table 6).

Table 4. Multichip approaches.

Attribute	Description and related works
Multichip	Smaller quantum circuits are executed in parallel, outputs of the quantum circuits are combined via post-processing [32]. Workload distribution : the QPUs are scheduled for the computation of a subset of an input problem [138,146–149]. Offloading : execution of programs with quantum tasks that are offloaded to a given QPU [150–154]. Mapping : quantum circuits are physically mapped to the QPUs [155–157].
Quantum and classical communication	Quantum communication is implementation-specific, and classical communication is available.
Applications	Phase estimation [158], amplitude estimation [159], quantum searching [160,161], multi-programming of quantum computers [162–166].
NISQ compatibility	Partial compatibility: multichip approaches with no entanglement between the circuits, due to the current quantum hardware and quantum resource limitations.

The distributed architecture of networked quantum services allows the upscaling of the computational power of smaller quantum devices to a high computational capacity. In the multichip approach, several smaller quantum chips are used in parallel to realize a scalable quantum system. In the circuit distribution and splitting methods, a large quantum circuit of a large quantum computer is split into several smaller subcircuits and implemented in the distant quantum nodes. The splitting of the QC can be made in different ways: horizontally or vertically, in general. Note, circuit splitting can refer to gate-cutting [143] or wire-cutting [144], both can be shown to be equivalent [145]. The main attributes of the multichip approach with the related works are summarized in Table 4.

The details of the circuit distribution method with the related works are given in Table 5.

Table 5. Circuit distribution approaches.

Attribute	Description and related works
Circuit distribution	Partitioning: the large quantum circuit is mapped onto a graph, and the optimal partitioning of the graph has to be found with minimal teleportation or distributed (non-local) gates between the nodes [32,167]. Partitioning methods for quantum circuits: deep reinforcement learning (DRL) [168], Karlsruhe hypergraph partitioning [167,169–176], Kernighan-Lin partitioning [177–181], Fiduccia-Mattheyses algorithm [182,183], tree-based directed acyclic graph (TDAG) [184], relaxed-overall extreme exchange (rOEE) [185,186], Hungarian qubit assignment (HQA) [187], quadratic unconstrained binary optimization (QUBO) [188]. Distribution of partitions: via entanglement generation and quantum communications between the QPUs. Partition mapping to QPUs: the subcircuit is mapped to the physical structure in the QPU via a local map. Primary method: Fine Grained Partitioning (FGP) [186,189].
Quantum and classical communication	Quantum communication is available, classical communication is available.
Applications	Distributed quantum gates [63], estimating the mean of numbers [190], distributed Simon’s algorithm [62], distributed quantum phase estimation [52,60,61], distributed Deutsch-Jozsa algorithm [59], distributed Bernstein-Vazirani algorithm [67], distributed quantum searching [68,191], distributed quantum Fourier-transform [192], distributed integer factoring [55,64,193].
NISQ compatibility	Partial compatibility, due to the current limitations of quantum hardware and quantum resources.

Relevant attributes of the circuit splitting method with the related works are included in Table 6.

Table 6. Circuit splitting approaches.

Attribute	Description and related works
Circuit splitting	Horizontal splitting: a quantum circuit is split horizontally between distant nodes [143–145]. Horizontal, incoherent splitting: the quantum circuit is simulated by a sequence of local circuits followed by a quantum measurement of the qubits [143,144,194–199]. Horizontal, coherent splitting: the quantum circuit is simulated by a sequence of local circuits with no quantum measurement on the qubits to preserve quantum information [143,200–202]. Horizontal, combined incoherent and coherent splitting: combination of incoherent and coherent splitting [203]. Vertical splitting: a quantum circuit is split vertically between distant nodes.
Quantum and classical communication	Quantum communication is not available, classical communication is available. Horizontal split requires classical communication between the nodes [199–201,203,204], vertical split requires quantum tomography [196]. Utilization of quantum datasets [200].
Applications	Simulation of large quantum circuits [144], optimal quantum circuit cuts to clustered Hamiltonian simulation [205], combinatorial optimization [198,206], solution of chemical problems [195], complex problem solving via small quantum circuits [207], high-dimensional quantum machine learning [208].
NISQ compatibility	Partial compatibility: horizontal, incoherent splitting due to the current limitations of quantum hardware.

The multichip, circuit distribution and circuit splitting architectures are depicted in Figure 3.

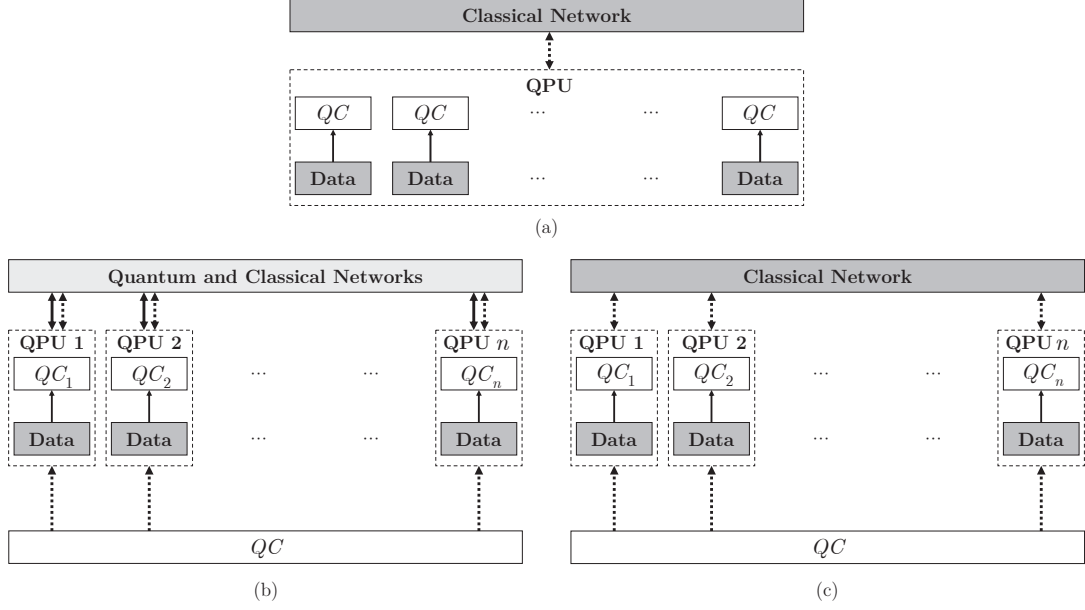


Figure 3. Distributed quantum computer architectures. **(a)** Multichip: the n smaller quantum circuits (QC) fit in a quantum processing unit (QPU), quantum communication is implementation-specific, classical communication is always available via Classical Network. **(b)** Circuit distribution: a large quantum circuit (QC) is distributed among the n QPUs such that each node realizes a smaller subcircuit of it with relation $\sum_{i=1}^n w_i QC_i = QC$, where QC_i is the local circuit of the i -th QPU, while w_i is a real weighting coefficient. Quantum communication is available (via Quantum Network), classical communication is available (via Classical Network). **(c)** Circuit splitting approach: a large quantum circuit (QC) is split among the n QPUs, such that each node realizes a smaller subcircuit with relation $\sum_{i=1}^n w_i QC_i = QC$, no quantum communication is available, classical communication is available.

A distributed CNOT gate between two distant QPUs is shown in Figure 4.

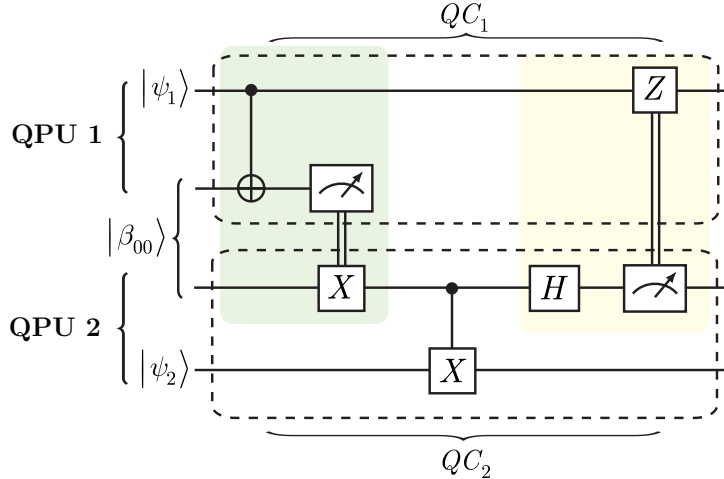


Figure 4. A distributed CNOT gate realized by two QPUs between distant quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$. Local system $|\psi_1\rangle$ is at QPU 1, while $|\psi_2\rangle$ is at QPU 2. QPU 1 has the quantum circuit QC_1 (dashed line frame) and QPU 2 has QC_2 (dashed line frame). The quantum nodes also share a Bell state $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The QPUs use their local quantum circuits and classical communication (doubled lines) to realize the CNOT gate between $|\psi_1\rangle$ and $|\psi_2\rangle$ in a distributed manner. The green-shaded box is the cat-entangler sequence [192], and the yellow-shaded box is the cat-disentangler sequence. Related implementations: [209–213].

Recent implementations of multichip and circuit splitting approaches are given in Table 7.

Table 7. Recent distributed quantum computing approaches and implementations.

Approach	Description and related works
Multichip	Architecture for multicore quantum computers with double full-stack communication [90].
	Quantum data networking for distributed quantum computing [214].
	Multi-qubit generation, development of multichip quantum computing platform [215].
	Scalable multichip quantum architecture using hybrid wireless/quantum-coherent network [216].
	Multichip multidimensional quantum network with entanglement retrievability [217]. Modular superconducting-qubit architecture with a multichip tunable coupler [218]. Variational quantum algorithms [219–224].
Circuit distribution	Arithmetic on a distributed-memory quantum multicomputer [63].
	Scalable distributed gate-model quantum computers for distributed problem-solving [58].
	Architectures for multinode superconducting quantum computers [48].
	Modular quantum compilation framework for distributed quantum computing [225].
	Factoring 2048 bit RSA integers using 20 million noisy qubits [55].
	Distributed quantum-classical hybrid factoring algorithm [64].
	Imperfect distributed quantum phase estimation [60].
	Implementation for a quantum message passing interface [61].
	Distributed quantum algorithm for Simon’s problem [62].
	Distributed quantum algorithms for Deutsch-Jozsa problem [59].
Circuit splitting	Distributed Bernstein-Vazirani algorithm [67].
	Distributed Grover’s algorithm [68].
	Scalable quantum computing infrastructure using superconducting electronics [226].
	Distributed quantum computation with circuit splitting [198].
	Quantum circuit cutting with maximum-likelihood tomography [196].
	A smart quantum circuit cutting method [227].
	Hypergraphic partitioning of quantum circuits for distributed quantum computing [183].
	Fast quantum circuit cutting with randomized measurements [199].
	Clifford-based circuit cutting for quantum simulation [228].
	Scalable emulation of quantum algorithms on high-performance computers [229]. Dimensionality reduction via circuit splitting for quantum reinforcement learning [230].

3.3. Networked Quantum Machine Learning

In this section, we are focusing on distributed quantum neural networks (QNN). Quantum neural networks are also referred to as parameterized quantum circuits (PQC) or variational quantum circuits (VQC) [130,140,231–237].

The most promising approach for quantum machine learning in the NISQ setting is to use variational quantum algorithms (VQAs) [103,129,238]: this includes classical hardware to overcome the limitations of the current quantum hardware. VQAs are useful for machine learning problems [103,129,238] and combinatorial optimization [239,240]. A VQA consists of a feature map, an architecture (ansatz), loss function, and an optimization method [234,241]. These attributes make them a strong candidate for achieving a quantum advantage in the near term. The feature map encodes the classical input data into quantum states [234,242]. The algorithms use an optimizer on classical hardware to train a PQC, which is used to find the quantum state containing the solution to the problem [234,243]. However, the optimal choice of the optimizer, the structure of the PQC, and the hyperparameters are problem-specific and have a major impact on the performance of a VQA.

A schematic model of a VQA is depicted in Figure 5.

In general, quantum circuits are sequences of unitaries each of which depends on a few parameters. Due to the lack of error correction in the NISQ era, the number of gates that can be applied is limited by the fidelities of the quantum gates, and the tolerable error is low, which is a challenge for the implementation of gate-model quantum neural networks in near-term devices [21,112,129,238]. Depending on whether classical or quantum data is used, and a classical or quantum algorithm, the quantum machine learning architectures can be classified into different groups [35,244,245]. These groups with their main attributes and the related works are summarized in Table 8.

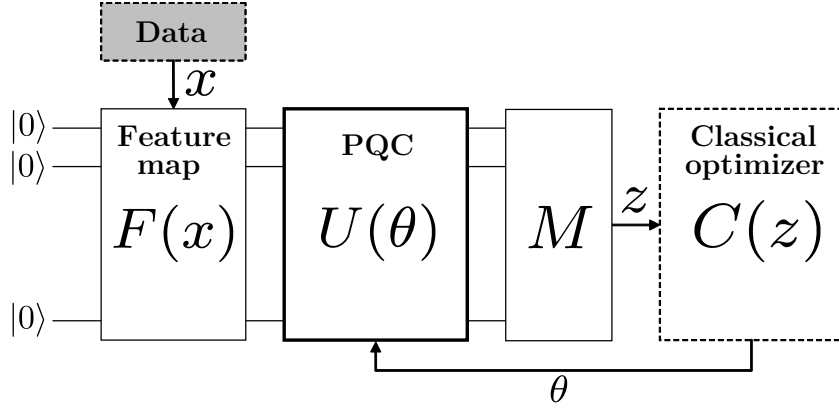


Figure 5. Schematic model of a VQA. The classical data x is mapped onto n qubits via feature map $F(x)$, and fed into the parameterized quantum circuit $U(\theta)$ (PQC), where θ is the parameter of unitary U . The output of $U(\theta)$ is measured via measurement M in the computational basis, which results in a classical bit string $z \in \{0, 1\}^n$ of length n . This string is fed to a classical computer to evaluate a cost function $C(z)$. Depending on the value of $C(z)$, the parameter θ is updated via a classical optimizer and propagated back to $U(\theta)$. The circuit is re-run multiple times until a convergence or other stopping criterion is fulfilled.

Table 8. Classes of quantum machine learning based on the data type (classical or quantum) and the algorithm (classical or quantum).

Data	Algorithm	
	Classical	Quantum
Classical	Classical-Classical: classical data and classical machine learning algorithms inspired by quantum mechanics [246].	quantum-classical: classical data encoded into quantum states and processed by quantum processors for quantum speedups [247–250], NISQ applications [112,251].
Quantum	Quantum-Classical: quantum data augmented by classical computations [252–259].	Quantum-Quantum: quantum data on quantum processors [260,261].

Major challenges in the implementation of VQAs are trainability, mainly by the occurrence of the *barren plateau* [234,262–264] and *narrow gorge* [265], efficiently calculable cost functions and expectation values, *multiple local minima* in the loss function [266], high precision measurements [234], and ensuring high accuracy on near-term quantum processors with high levels of noise and errors. The hyperparameters of VQAs are the configuration parameters, such as those for the optimizer or the PQC [234,243,267,268]. It may be assumed that a further quantum advantage can be reached in terms of space efficiency, performance, and expressiveness of learning models in the NISQ era. Translational and exploratory approaches are studied in [269]. Different types of quantum neural networks have been defined in the literature, such as orthogonal, convolutional, feed-forward, self-supervised quantum neural networks, variational depth quantum circuits, quantum Boltzmann machines, along with the applications of linear regressions or quantum amplitude estimation. For a comprehensive review of quantum machine learning applications, see [21].

In a distributed quantum neural network, the training is realized in a parallelized way between distant nodes. The parallelization can be made via *data splitting* (Table 9) or *circuit splitting* (Table 6). In data splitting approaches, the dataset is split across the quantum nodes such that the actual model is loaded into each node (term “model” refers to a PQC in this context). In circuit splitting, the model (a PQC) is split across the distant nodes such that each node has access to the full dataset.

The main attributes of data splitting approaches (encoding, data storage, and use of classical side information) in distributed quantum neural networks are summarized in Table 9.

Table 9. Data splitting approaches in distributed quantum neural networks.

Attribute	Description and related works
Data encoding	Encoding of a classical dataset into quantum states. Basis encoding: the data is encoded in a computational basis state. The dataset is encoded as superposition of the computational basis states [270–272]. Amplitude encoding: uses amplitudes of the quantum state for dataset encoding [273,274]. Angle encoding: tensor product encoding at the single-qubit level without entanglement within feature vectors [275–278]. Hamiltonian encoding: encoding is made at the Hamiltonian level mostly by two-qubit entangling gates [139,242,279,280].
Data storage	Quantum random access memory [248–250].
Quantum and classical communication	Quantum communication is available, quantum simulators are available, quantum AI software packages [274], classical communication is available [281,282].
Applications	Classification problems [112,251,283], quantum data compression [284], quantum Boltzmann machines [285], feature mapping [139,242,279,280,286], machine learning problems [275–278].
NISQ compatibility	Partial compatibility: basis encoding, angle encoding, and Hamiltonian encoding, due to the current limitations of quantum hardware.

The data splitting and circuit splitting methods in a distributed quantum neural network setting are compared in Figure 6.

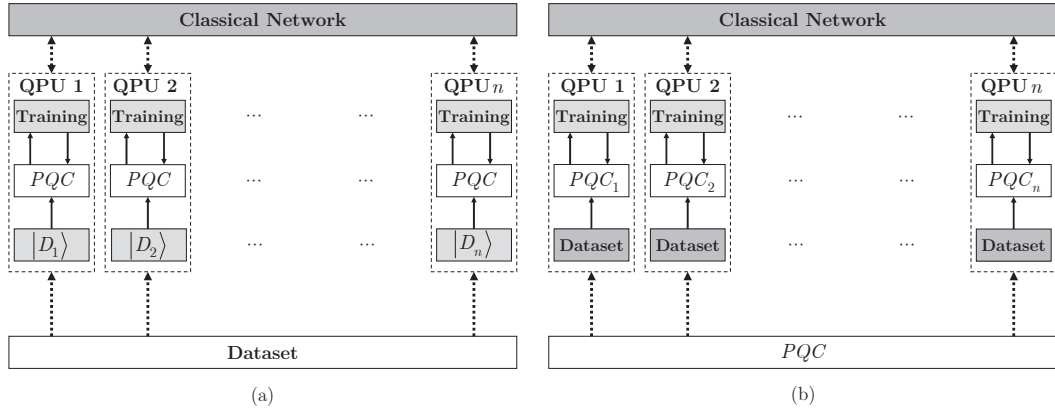


Figure 6. Data splitting and circuit splitting in distributed quantum neural networks. **(a)** Data splitting. The full dataset (Dataset) is split into n parts, denoted by $|D_1\rangle, \dots, |D_n\rangle$, across the n quantum nodes (QPUs), while satisfying $\sum_{i=1}^n |D_i\rangle = \text{Dataset}$. The model (PQC) is loaded into each QPU. The QPUs process and train in parallel, each QPU has access to a classical network (Classical Network). **(b)** Circuit splitting. The input PQC is split into n parts, $PQC_1, \dots, PQC_n$, and loaded into the n QPUs along with the full quantum dataset (Dataset); $\sum_{i=1}^n c_i PQC_i = PQC$, where PQC_i is the local PQC of the i -th QPU, while c_i is a real weighting coefficient. The QPUs process and train in parallel; each QPU has access only to a classical network in general.

While data splitting is a general strategy for data parallelization in quantum neural networks, other methods have also been investigated. *Data reduction* (or coreset technique) is a different method that can also be used in quantum neural networks to optimize the amount of data shared between the quantum nodes. In data reduction approaches, an approximation of the dataset is distributed between the nodes to improve the parallel processing and to decrease the size of the dataset. In parameter distribution, classical side information is distributed between the nodes in a parallelized way for optimization purposes. Table 10 summarizes the main attributes of data reduction and data distribution in quantum neural networks.

Table 10. Data reduction and data distribution approaches in distributed quantum neural networks.

Approach	Description and related works
Data reduction (coreset)	Approximation of an original dataset. Application in variational algorithms [287], and in hybrid quantum-classical architectures [287,288].
Data distribution	Parameters are distributed as classical information between the quantum nodes in parallel for optimization [251,289].
Quantum and classical communication	Quantum communication is available, classical communication is available.
NISQ compatibility	Full compatibility.

In [290], a modified distributed learning scheme called quantum distributed optimization (QUDIO) has been proposed for variational quantum algorithms. The model consists of a distributed optimization scheme whose back ends support both quantum devices and various quantum simulators to further improve the capabilities of the VQAs. The proposed method collaborates with multiple quantum machines or simulators to complete learning tasks. The model is depicted in Figure 7.

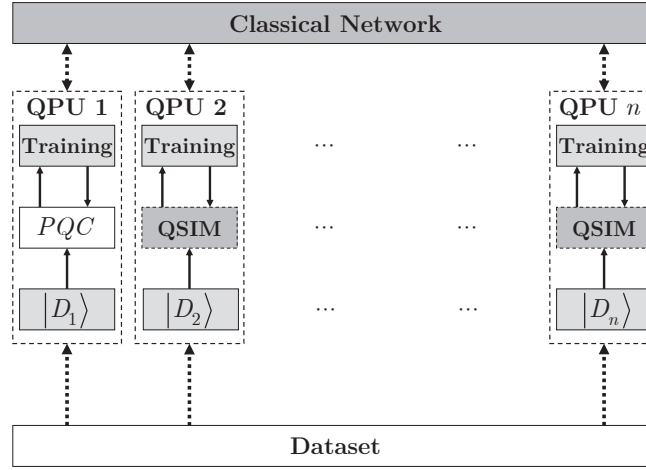


Figure 7. A QUDIO model for VQAs. The nodes are trained in parallel using a PQC or a quantum simulator (QSIM). A classical central server communicates with all local quantum nodes to synchronize the trainable parameters of the nodes.

3.4. Blind Quantum Computation

In blind quantum computations [291] multiple users (untrusted devices) perform computations and analysis on the joint data without sharing or revealing their own data with each other. The calculations are hidden from the users and the encrypted input algorithm is received by a server from a trusted client. The data of the users are fed into local quantum devices and a distributed quantum computation is established between them via a particular distributed quantum algorithm [33,91]. Classical communication is generally allowed between the quantum devices. A particular user has no knowledge about the other users' data, nor about the actual distributed quantum algorithm; however, some part of a hidden algorithm and the joint state can be learned by the parties, and the server can also have partial information about some parameters of the calculations. Due to these security issues, another approach for the realization of quantum blind computation is the use of a trusted secure quantum cloud [36,292,293]. If the users are connected via a secure quantum cloud, their local quantum devices cannot communicate with each other directly, only via the secure quantum cloud. On the experimental realization of blind quantum computing, see [49–51].

3.5. Quantum Cloud

Quantum cloud services allow the users to run quantum algorithms on quantum computers. These cloud services can be accessed via classical computers allowing the users to have access to quantum devices and resources. Several different cloud-based platforms have been developed with different resource capabilities, quantum computing models, and supported software development kits [36,37,294–296]. In general, the different quantum cloud platforms

provide commercial quantum computing as a main service. For a comprehensive review of quantum cloud computing, see [36].

The main attributes of recent quantum cloud platforms with the related works are summarized in Table 11. *Superconducting technology*: IBM Quantum⁸, Google Quantum⁹, Rigetti¹⁰, OCO: Oxford Quantum Circuits¹¹, QuTech¹². *Trapped ion technology*: IonQ¹³, Quantinuum¹⁴. *Neutral atom technology*: Pasqal¹⁵, QuEra¹⁶. *Quantum annealing technology*: D-Wave¹⁷. *Solid-state spins technology*: QuTech.

3.6. Remote Sensing and Interferometry

Remote quantum sensing uses the fundamentals of quantum mechanics for highly sensitive measurements in a distributed manner via an entangled network of distributed quantum sensors [65,66]. Applications of remote sensing include atomic quantum clocks, quantum sensors for geology, biomedical applications [317] and navigation, nanoscale vortex imaging using a cryogenic quantum magnetometer [318], quantum-enhanced accelerometers and gyroscopes, magnetic field detection, and gravitational wave detection.

Quantum interferometry can be also realized in a distributed way using multiple local quantum interferometers at distant quantum computers [33,319]. The local quantum computers process the results of the interferometers and use a distributed quantum algorithm to extract information to improve the accuracy and to decrease the noise level.

3.7. Further Works

On the use of QAOA in the NISQ setting, see [101,110,320–322], on variational quantum eigensolvers, see [323–326], and on pseudorandom quantum circuit-based solutions for quantum advantage, see [327–329]. The complexity of NISQ has been derived in [330], while the structure of shallow quantum circuits has been studied in [75,331]. For details on the methods of loading classical information into quantum computers, see [332].

For the fundamental details of parameterized quantum circuits, we suggest [130,140,231–237]. On the interoperability of quantum neural networks and quantum computations, and on the attributes of the different training procedures of quantum neural networks, we suggest [149,251,333–336]. A quantum-classical hybrid algorithm for determining eigenstate energies in quantum chemistry is proposed in [337]. On resource-efficient quantum algorithms for quantum chemistry, see [338]. For an enhanced feature encoding and classification on distributed quantum hardware, see [286]. Machine learning algorithms for near-term quantum computers can also be found in [242,339–342]. For the different learning architectures of quantum neural networks, see [235,343–348]. For further works on the barren plateau phenomenon in quantum neural networks, see [233,263,264,349–351]. The implementation aspects of quantum neural networks on current quantum hardware architectures for different problems are studied in [231,352–359].

4. Networked Quantum Software, Programming Languages, and SDKs

Networked quantum services are supported by different quantum software tools. Several different quantum programming languages are available for writing and running networked quantum services, with different functionalities for preparing the quantum gates, quantum circuits, and quantum algorithms [11,28,360,361]. The quantum software development kits (SDKs) are integrated frameworks with several different tools, libraries, and simulators to develop quantum circuits [27,362]. Quantum software libraries (SLs) are also available with functions and tools for development [363]. The study of Zhao [364] reviews quantum software engineering, while the study of Murillo et al. [365] proposes a roadmap and reviews the challenges of quantum software engineering.

In a networked quantum computing environment, the quantum software tools and quantum APIs allow to provide different networked quantum services to the users. These services include quantum computing as a service (QCaaS) solutions (such as quantum cloud computing, see Table 11), quantum encryption and quantum cryptographic services

⁸ IBM Quantum Computing Services. URL: <https://quantum-computing.ibm.com/>

⁹ Google Quantum AI. URL: <https://quantumai.google/>

¹⁰ Rigetti Computing. URL: <https://qcs.rigetti.com>

¹¹ Oxford Quantum Circuits. URL: <https://oqc.tech/>

¹² Quantum Inspire, De Voorhoe. URL: <https://www.quantum-inspire.com/>

¹³ IonQ. IonQ Cloud Service. URL: <https://ionq.com/>

¹⁴ Quantinuum Cloud Service, Quantinuum. URL: <https://www.quantinuum.com/>

¹⁵ Pasqal: Programmable Atomic Arrays. URL: <https://www.pasqal.com/>

¹⁶ QuEra, Quantum Computing with Neutral Atoms (2024). URL: <https://www.quera.com/>

¹⁷ D-Wave. URL: <https://www.dwavesys.com>

Table 11. Recent quantum cloud platforms.

Platform	Computing Model	Quantum hardware vendor	Description and related works
IBM Cloud	Gate-model, quantum simulation	IBM Quantum (1121 qubits) [297]	Remote access to the IBM quantum computing hardware, serverless model [298], supported quantum software: Qiskit.
Google Cloud	Gate-model, quantum simulation	Google (54 qubits) [299], IonQ (36 qubits) [300]	Remote access to Google’s quantum computing hardware [299], supported quantum software: Cirq.
Microsoft Azure Quantum	Gate-model, quantum simulation	Quantinuum (32 qubits) [301], Rigetti (84 qubits) [294], IonQ (36 qubits) [300], Pasqal (100 qubits) [302]	Remote access to a diverse portfolio of current quantum hardware [303], supported quantum software: Q#, Qiskit, Cirq.
Amazon Braket	Gate-model, quantum simulation	Rigetti (84 qubits) [294], OCQ (32 qubits), IonQ (36 qubits) [300], QuEra (256 qubits) [304]	Access to different types of quantum computers, simulators and quantum-classical algorithms, serverless model [305,306], supported quantum software: Braket, Qiskit, PennyLane.
PlanQK	Gate-model, quantum annealing, quantum simulation	IBM Quantum [297], Amazon Braket [305,306], Azure Quantum [303]	Remote running of quantum tasks and algorithms, provides access to major quantum backends and simulators, serverless model [307,308], supported quantum software: Qiskit, PennyLane.
QuantumPath	Gate-model, quantum annealing, quantum simulation	IBM Quantum [297], Amazon Braket [305,306], D-Wave (5000 annealing qubits), QuTech (5 qubits) [309]	Industry-ready hybrid quantum-classical solutions [310,311], supported software: Qiskit, Ocean, Braket, Q#.
Strangeworks	Gate-model, quantum annealing, quantum simulation	IBM Quantum [297], Amazon Braket [305,306], Azure Quantum [303]	Hybrid quantum-classical solutions, serverless model [312], supported quantum software: Qiskit, Braket, Forest.
QFaaS	Gate-model, quantum annealing, quantum simulation	IBM Quantum [297], Strangeworks [312]	A function-as-a-service framework for quantum computing, open-source, serverless model [313,314], supported quantum software: Qiskit, Cirq, Q#.
1Qloud	Quantum simulation	1Qbit [315]	Hybrid quantum-classical solutions [315], supported quantum software: 1Qbit.
QEMIST	Quantum simulation	1Qbit [315]	Hybrid quantum-classical solutions [316], supported quantum software: OpenQEMIST.

(such as the QKD services of ID Quantique, MagicQ, etc.), and quantum AI services (AI services of QC Ware, Xanadu, CQC, etc.).

The limitations of quantum software tools in a networked quantum computing environment are related to the No-cloning theorem (we cannot copy the program states) and the problem of measurement (a measurement destroys the program states). These issues can be handled via quantum simulators [9,195] in the stage of quantum software development. Quantum simulators are available in the different quantum SDKs (see Tables 12 and 13), while the various quantum simulators and online AI platforms can be connected to serve as an integrated platform for quantum programming and quantum software development [366]. The quantum programs then can be tested on real quantum devices with several repetitions and measurement rounds to get a probability distribution of the results.

Quantum programming languages allow the development of quantum operating systems, quantum firmware, and quantum toolkits [367] for distributed quantum computations. Quantum software handle distributed quantum computations via error correction protocols and routing protocols, and provide an infrastructure for the various quantum hardware platforms. The networking protocols provided by quantum programming languages allow to adapt and

improve the network performance via the enhancement of transmission rates, and the quality of service (QoS) of distributed quantum computations.

A challenge of integrating quantum and classical computing workflows is the cooperation of quantum and classical interfaces and programs. A possible solution to the problem is the utilization of paradigms of service-oriented computing (SOC) in quantum computing. SOC-based approaches allow quantum and classical developers and programmers to collaboratively create hybrid quantum-classical applications in an efficient and scalable way. The works of Kumara et al. [368] and Moguel et al. [369] focus on this topic. Another challenge is the handling of decoherence and noise in quantum hardware platforms, since disturbances can result in different probability distributions on different quantum hardware. The quantum simulators integrated into the different quantum programming frameworks and quantum SDKs can also result in different classical outputs which makes it difficult to integrate the results.

Relevant attributes of recent quantum software are included in Table 12.

Table 12. Quantum software for networked quantum services.

Quantum Software	Description and related works
CutQC	Software package for circuit splitting [197].
Interlin-q	Software package for the development of distributed quantum algorithms [166].
PennyLane	Quantum software for simulations and experiments on current NISQ quantum devices, by Xanadu [370].
Qiskit	Quantum software for simulations and experiments on current NISQ quantum devices, by IBM [363,371].
QuantumCircuitOpt	Software for the implementation of mathematical optimization and algorithms for decomposing arbitrary unitary gates into a sequence of hardware-native gates [372].
QuPanda	Software for creating and executing complex quantum circuits and algorithms, by Origin Quantum [373].
QuNetSim	Software for real time quantum networks simulation [374].
Qurzon	A compiler that uses CutQC and other tools [375].
ScaleQC	A software tool for circuit splitting [376].
SuperSim	A software tool for circuit splitting [377].
TorchQuantum	A software for merging quantum computing with deep learning, by MIT [378].
Tensorflow Quantum	Distributed quantum machine learning [379], distributed training of quantum neural networks [349,380], by Google and NASA Ames.

The attributes of recent quantum programming languages, quantum SDKs, and quantum SLs for networked quantum services are summarized in Table 13.

Table 13. Quantum programming languages, quantum SDKs, and quantum SLs for networked quantum services.

Approach	Description and related works
Programming language	Qiskit: an integrated programming language and quantum SDK for quantum simulations and quantum algorithms, by IBM [363,371,381–383].
	PyQuil: a Python library for quantum programming using Quil, the quantum instruction language developed by Rigetti Computing [384].
	Q#: programming language for quantum computing, by Microsoft. Integrated support for different program languages and quantum development kit [385].
	Quipper: an embedded, scalable functional programming language for quantum computing [386,387], by Microsoft and the University of Oxford.
	Ocean: quantum SDK for quantum annealing algorithms, by D-Wave [388,389].
Quantum SDK	Forest: quantum SDK for quantum computing using the quantum services of Rigetti [390,391].
	Microsoft QSDK: quantum SDK for quantum computing using Microsoft quantum services [392,393].
	Strawberry Fields: a cross-platform Python library for simulating and executing programs on the quantum photonic hardware of Xanadu [394].
	ProjectQ: a Python-based open source framework for quantum computing [395].

Table 13 Continued.

Quantum SL	Cirq: a quantum software library by Google for the development of quantum circuits and noise modeling. Cirq provides abstractions for NISQ quantum computers [396]. OpenFermion: a quantum software library and electronic structure package for quantum computers [397]. QuTiP: an open-source Python quantum software library for the dynamics of open quantum systems [398].
------------	---

4.1. Quantum API

A quantum API is a software interface that allows interaction with quantum computers in applications. These APIs also connect the different programming languages and the quantum hardware. The quantum computing vendors provide various APIs (IBM Quantum Experience API, Amazon Braket API, D-Wave API, etc.), therefore a standardization of access to quantum services is an important problem regarding quantum APIs [37,399–401]. A solution to the standardization problem and for the optimization of accessing quantum services could be the use of quantum API gateways [399,402]. Quantum API gateways integrate the multiple quantum vendors and offer the best quantum hardware available for a particular task based on the input parameters received from the users' applications.

A solution to the problem of service accessing the different platforms is an extension of the quantum API gateway, application of code generators, and a workflow to automate the development and deployment of the services [369,403]. It allows to deploy quantum code without specific knowledge of the quantum platforms, which can simplify the development of quantum applications and workflow integration.

An aim of the utilization of programming tools and middleware units is to allow access to quantum services using a standardized format for all quantum providers. A standard format for requesting services and handling service responses also allows developers to manage the communication workflow in an abstract-level, independent from the actual physical quantum hardware and the constraint parameters of the different quantum providers. An approach to standardize the process of defining quantum services using the OpenAPI specification has been proposed in [37]. The method can generate source code of quantum services from an API specification and a quantum circuit. For a detailed discussion on the subject of standardization, see the works of Moguel et al. [24,37].

The different standardization approaches of service access, service development, and programming are summarized in Table 14.

Table 14. Standardization approaches for networked quantum services.

Approach	Description and related works
Quantum API Gateway	A machine learning-based middleware for the integration of different quantum vendors for quantum service access [37,399–402,404].
Open API Quantum	A vendor-independent description format for quantum services [405, 406].
Qiskit Runtime	API for hybrid quantum-classical computing [37].
Universal Quantum Intermediate	API for hybrid quantum-classical computing [407].
Compilation protocol	Simultaneous execution of quantum circuits on NISQ systems [408].
Quantum simulators	Connection of different quantum simulators and online AI platforms [366].
Cloud computing with load balancing	Quantum-based security approach for cloud computing with load balancing [409].
Communication with quantum providers	Quantum secure communication between service provider and subscriber identity module [410].
Quantum service prioritization	Integration of different quantum hardware and quantum compilers [411].
Quality of quantum services	Improving the quality of quantum services via additional layer [412, 413].
Programming of quantum services	Python-based programming in the quantum cloud for quantum services [414].

Table 14 Continued.

Development and deployment	Tools for the development of quantum services [369], and hybrid quantum-classical services [403].
Provenance system	A provenance method for the selection of a quantum computer [415].
Hybrid quantum applications	Orchestration of quantum and classical services in hybrid systems [416].

5. Implementation Basis

In this section, we review the concepts that are essential for the operation of networked quantum devices, as well as the main types of devices and requirements.

5.1. Quantum Channels

Any two communicating nodes in the system are connected by a quantum channel, a communication medium that is influenced by nature in the form of noise [417]. The theoretical modeling of quantum channels has a long history. Many articles have related the theoretical maximum of the transmission capacity of the channels to their capacity; for understandable reasons, the capacity for classical information has primarily been considered, since in the initial point-to-point systems, the end users were human beings who could only interpret this type of information. However, even in this case, a very diverse range of questions arise; for example, how much classic information can be safely transferred over the channel, and by how much is the capacity increased if we share entangled pairs between the nodes before communication (i.e., when investigating entanglement-assisted capacity)?

The situation becomes completely different when networked quantum devices communicate with each other. This case involves the transmission of quantum information, which is much more sensitive to noise effects, meaning that error correction codes that enable noise control play an even more prominent role. Discussion of this topic is further complicated by the fact that the definitions of the various quantum capacities are not comprehensive, and in some cases there is even a lack of consensus on the quantity describing the information to be maximized.

The capacity of the quantum channel is a guiding quantity, as it maximizes the amount of information that can be transferred. In a practical implementation, error correction codes/algorithms compete for the best approach to capacity [418].

In a similar way to the planning and deployment of classical communication networks, it is also necessary in the quantum case to refine the theoretical channel models based on measurements, to bring them closer to reality.

According to current trends in quantum communication, there are two parallel directions for future development. Optical fiber-based systems will use existing classical optical communication links that are widely installed worldwide; however, these have a different effect on classical photon packets and the information encoded in the polarization of photons. In a real-world implementation, additional challenges arise if classical and quantum communication travels on the same optical fiber, as artificial interference also inevitably appears in addition to the noise caused by nature.

Quantum free space communication primarily involves the exchange of information with GEO/LEO (geostationary equatorial orbit/low Earth orbit) satellites. As the Earth's atmosphere is relatively thin, its effect on photons is significantly weaker than that of optical fibers, meaning that longer distances can be bridged. This type of communication is also asymmetric, meaning that it is primarily suitable for communication in the satellite-to-Earth direction.

Finally, the role of quantum channel simulators should be emphasized. These are sophisticated tools that are designed to model and analyze the behavior of quantum channels while accounting for various physical effects such as noise, decoherence, and loss. Quantum channel simulators play a crucial role in understanding how quantum states evolve as they are transmitted through the channels, and can help in the development of quantum communication protocols, error correction techniques, and other quantum technologies.

5.2. Quantum Devices, Quantum Repeaters, and Quantum Computers

When bridging large distances in classical networks, devices may be needed between the endpoints to mitigate losses and distortions in the communication channel. This task is performed by devices called repeaters, which amplify incoming weak signals and send them on. Amplification involves a kind of duplicating (copying) of the incoming weak signal. However, in quantum networks, the No-cloning theorem makes it impossible to copy any incoming arbitrary quantum state without error.

Quantum repeaters are essential components of quantum communication networks and are designed to extend the range of quantum communication by overcoming the limitations of direct transmission of quantum states [419].

They are used to facilitate long-distance quantum communication by addressing issues such as loss and decoherence, which occur in optical fibers or free-space communication.

Quantum long-distance communication uses a combination of two protocols in order to circumvent the No-cloning theorem. Firstly, using entanglement swapping, entangled pairs are created between endpoints located at great distances from each other. Following this, they undergo teleportation, in which quantum states can be transmitted between the parties by communication over a classical channel. The quantum repeaters play an active role in the first phase, as they create entangled pairs between themselves and then perform an operation (Bell measurement) in a synchronized manner, which creates an entangled pair between the endpoints. From the point of view of implementation, the repeater function relies on an extremely precise classical synchronization mechanism between the endpoints. In addition, the resulting entangled quantum states must be preserved for a sufficient time so that they are available for teleportation; in other words, the decoherence time of the quantum memory function must also be sufficiently long.

Although quantum processors, or computers that have been proven to obey quantum phenomena, have been appearing regularly since 2018 and their physical numbers of qubits are increasing exponentially (2^n) every year, meaning that the state space of manageable problems is growing super-exponentially (2^{2^n}), there is still competition between different technologies.

Quantum computers can be categorized into several types based on their underlying technology and operational principles, as follows:

- *Superconducting Qubit Computers:* These quantum computers use superconducting circuits to create qubits. Superconducting qubits are among the most widely researched and developed types, with companies like IBM and Google leading the way. They operate at extremely low temperatures to maintain quantum coherence.
- *Trapped Ion Quantum Computers:* This type uses ions trapped in electromagnetic fields as qubits. Laser beams manipulate the states of the ions. Trapped ion systems, like those developed by IonQ and Honeywell, have demonstrated high fidelity in quantum operations and scalability.
- *Topological Quantum Computers:* Topological qubits are based on anyons, which are exotic particles that exist in two-dimensional spaces. Theoretical models suggest that topological qubits could be more robust against errors due to their topological nature. Research is ongoing in this area, with Microsoft being a prominent player.
- *Photonic Quantum Computers:* These systems use photons (light particles) as qubits. Photonic quantum computers can operate at room temperature and utilize optical components like beam splitters and waveguides. They are being explored for their potential in quantum communication and networking.
- *Quantum Dot Computers:* Quantum dots are semiconductor particles that can confine electrons and create qubits. These systems leverage the properties of quantum dots to perform quantum computations, and researchers are exploring their potential for scalability and integration with existing semiconductor technology.
- *Neutral Atom Quantum Computers:* Neutral atoms are trapped and manipulated using optical tweezers or magnetic fields. This approach allows for scalable qubit arrays and is being researched for its potential in quantum simulation and computing.
- *Adiabatic Quantum Computers:* These systems use adiabatic processes to evolve a quantum state from an easy-to-prepare initial state to a hard-to-solve problem state. D-Wave Systems is known for developing adiabatic quantum computers primarily focused on optimization problems.

The wide variety of quantum computers that will be available in the future may limit their networked operation, since only machines in which there is an interface between the qubits performing the calculations and the photons carrying the information will be able to be connected to the network. Another requirement is that quantum computers must be able to preserve quantum stored information for at least as long as the period over which communication between the machines takes place. That is, the coherence time of quantum memories also restricts the range of machines that are capable of cooperation. Here, we draw attention to the possibility that the information storage qubits of quantum memories may be based on a different physical medium. In other words, a convention may be necessary between the qubits that perform calculations, store information, and deliver information; these processes suffer from noise, making the role of the error correction codes even more valuable.

Based on their important roles in quantum communication, we can identify the most promising physical implementations of quantum memory as follows [420]:

- *Atomic Ensembles:* These systems use collections of atoms or ions to store quantum information. The quantum states of the atoms can be manipulated using light fields, which allows for the storage and retrieval of quantum states. Techniques such as electromagnetically induced transparency (EIT) are often employed in these systems.
- *Solid-State Systems:* Quantum memories can also be realized using solid-state materials, such as quantum dots or nitrogen-vacancy (NV) centers in diamond. These systems can provide robust storage capabilities and are often integrated with photonic devices for efficient state transfer.
- *Photonic Quantum Memories:* These memories utilize photons as carriers of quantum information. Photonic quantum memories can be implemented using nonlinear optical processes, which allows for the temporary storage of photonic quantum states.
- *Cryogenic Quantum Memories:* Some quantum memory systems operate at cryogenic temperatures to reduce thermal noise and enhance coherence times. These systems often utilize superconducting qubits or other low-temperature phenomena.

In summary, it can be concluded that all quantum devices intended for use in network operation must meet three requirements: *precise synchronization*, *efficient error-correction coding*, and *memory with adequate coherence time*. These functions do not necessarily require different physical implementations, and combining them can increase efficiency.

5.3. Entanglement-Assisted Networked Computing

In networked quantum services, a particular quantum state can be affected by noise from various sources such as quantum gate errors, channel losses, and environment noise. The F fidelity of the state, $0 \leq F \leq 1$, can be used to quantify how much a particular state has been affected by these noise sources [70,91]. The generation of Bell pairs, or the process of entanglement swapping are both noisy operations, and as a corollary it degrades the fidelity of a quantum state. Higher-fidelity Bell states can be generated from lower-fidelity pairs via the process of entanglement purification (or entanglement distillation). Quantum error correction (QEC) can also be utilized in entanglement distillation with lower communication costs [421], but with higher demands on the fidelity of the input quantum state. The resulting output fidelity, F' , of the entanglement purification at an input state fidelity F_0 is $F' = F_0^2/p_S$, with relation $F' > F_0$ if $F_0 > 0.5$, where p_S is the success probability of entanglement purification, $p_S = F_0^2 + (1 - F_0)^2$. The utilization of Bell pairs in networked quantum services depends on a F_{thr} fidelity threshold, below which the entangled states cannot be used in practice. Different applications can have different requirements on F_{thr} . In networked quantum services, a Quality of Service (QoS) parameter can be the end-to-end Bell pairs per second (BPPS) and the required fidelity of the Bell pairs [46,91,422].

Scalability of networked services requires the extension of quantum systems over long-distances while preserving quantum coherence. The current hardware limitations of quantum devices such as gate fidelities and error rates make the problem of scaling even more difficult [58,423–425]. The handling of external disturbances, unwanted interactions, and decoherence in quantum systems are the main challenges in scaling networked quantum services. To achieve scalability over long distances in networked quantum services, the quantum repeater architecture of the quantum internet can be utilized with advanced quantum networking protocols [45]. An integration of physical layer developments (entanglement swapping, entanglement purification, multipartite entanglement, quantum error correction, quantum memories, controlling) with higher layer advancements (entanglement routing, delay minimization techniques, dynamic path selection, network intelligence, error management, connectivity management, etc.) provides an effective solution for the scalability of practical networked quantum services.

5.4. Quantum Error Correction, Fault-Tolerance, Quantum Memories

The aim of quantum error correction is to preserve quantum information decoherence and errors. By using quantum error correction codes (such as surface codes, error syndromes, code concatenation) the accuracy and reliability of quantum information processing can be enhanced significantly. The tools of quantum error correction allow us to realize fault-tolerant quantum computations in experiments. According to the threshold theorem [426] (or quantum fault-tolerance theorem), a quantum computer with a physical error rate below a certain threshold can suppress the logical error rate to arbitrarily low levels via the application of quantum error correction. This means that quantum computers can be made fault-tolerant.

Due to the efficient error correction techniques, the percentage of error of quantum gates is decreasing over the years [13,427,428]. The current ε error rate (%) of quantum gates is in the range of $\varepsilon \approx 0.01$, which represents

approximately two orders of magnitude improvement in the last few years. Table 15 summarizes the ε gate error rates of quantum processors of different quantum vendors.

Table 15. Gate error rates of quantum processors.

Quantum processor	Error rate (%)	Release date
IBM Quantum Eagle	~ 0.9	2019
Google Sycamore	~ 0.6	2020
Rigetti Aspen-9	~ 0.5	2021
IonQ Harmony	~ 0.3	2022
IBM Quantum Hummingbird	~ 0.2	2023
Google Quantum Bristlecone	~ 0.15	2023
Rigetti Aspen-12	~ 0.1	2024
IonQ Symphony	~ 0.01	2024

An advantage of networked quantum services is the *enhanced fault-tolerance* and *improved scalability* due to the distributed architecture [31]. For a detailed discussion on quantum error correction, fault tolerance, and quantum memory limitations, we suggest references [71–74].

In networked quantum services, quantum error correction can be implemented with a high cost due to the conditions on physical resources (high number of physical qubits with high fidelity). According to a current roadmap [70] of quantum networks, quantum error correction will be used only in later generations of quantum networks, while current networked quantum services can use entanglement swapping and quantum teleportation. Using Bell states with small-scale error correction such as entanglement distillation represents an implementable alternative of high-cost quantum error correction schemes in practical scenarios.

Quantum networks can be categorized into Types I–III, based on their loss and error tolerance schemes [70,429], as it is summarized in Table 16.

Table 16. Error correction and management in quantum networks.

Network type	Loss tolerance scheme	Error tolerance scheme	Delay	Cost
Type I.	Heralded entanglement generation with bidirectional classical side-information	Entanglement distillation with bidirectional classical side-information	High	Polynomial scaling with total distance
Type II.	Heralded entanglement generation with bidirectional classical side-information	Entanglement distillation with unidirectional classical side-information, or QEC with no classical side-information	Moderate	Polylogarithmic scaling with total distance
Type III.	QEC with no classical side-information	QEC with no classical side-information	Low	Polylogarithmic scaling with total distance

Type I quantum networks use heralded entanglement generation, the entangled states are probabilistically generated and the distribution range is extended via entanglement swapping. The term “heralded” refers to that the entanglement generation process runs until the two non-neighboring stations do not receive an ACK (acknowledge) message from the intermediary station, which confirms that the entanglement swapping operation was successful. High-fidelity entanglement is produced via entanglement purification. These networks use two-way classical communications which reduces significantly the achievable communication rates.

From Type II quantum networks, the entanglement purification procedure can be replaced by QEC. It improves the rate of communication since it can reduce the required amount of classical communications. However, QEC also requires higher state fidelities and additional quantum circuits that make the scheme more complex.

Type III quantum networks use QEC to handle the loss and operational errors without classical communications. These networks can use direct unidirectional communication, but with an improved complexity due to the use of QEC for loss and error tolerance. Type III networks can directly transmit QEC-encoded qubits to adjacent quantum nodes, and the Bell pairs can be created without heralding or the use of classical side-information [70]; however, they require high-fidelity quantum gates and longer coherence times.

Table 17 summarizes the applications of different QEC codes [427] in Type II-III networks.

Table 17. Error correction in Type II-III quantum networks.

Network type	QEC code	Application
Type II.	Repetition code, Shor code, Calderbank-Shor-Steane (CSS) code	Correction of operational errors.
Type III.	Surface code, Gottesman-Kitaev-Preskill (GKP) code	Correction of photon loss and operational errors.

Regarding quantum memories, a critical problem in achieving persistent storage is the isolation of a quantum system from the environment. Due to the decoherence, the environment adds an uncontrollable noise into the system, which also degrades the fidelity of the state. The lifetime (coherence time) of a quantum memory is implementation-specific; however, the highest achievable values currently in a quantum network hardware are in the range of few seconds to few minutes (see Table 18). The coherence time values of quantum memories are continuously increasing as the different technologies evolve. Networked quantum services also have to be able to manage short coherence times. A possible solution to this problem is the reduction of the latency on critical paths of the network [70]. Details of recent quantum memory implementations with their highest coherence time values are given in Table 18.

Table 18. Quantum memory implementations and their lifetime (coherence time) values.

Quantum memory	Coherence time
Single ion qubit [430]	~1 hour
Single trapped ion qubit [431]	~10 min
Ten-qubit solid-state spin register [432]	~1 min
Single electron spin coupled to a multi-qubit nuclear-spin environment [433]	~1 sec
Superconducting cavity qubit [434]	~Tens of milliseconds

6. Conclusion and Outlook

The advances in quantum computation and communications provide the ground for novel developments such as networked quantum services, which allow users to access quantum resources and efficient solutions in quantum networking scenarios and in hybrid quantum-classical architectures. In a similar way to classical systems, the networked operation of quantum devices will bring qualitative advances, representing the third quantum revolution. This paper has reviewed the topic of the networking of quantum devices, which requires significant theoretical and technological progress. In our article, we have presented the gate-model concept and its potential applications in various fields of the quantum world. We have also pointed out the technological requirements that networked devices must meet.

From the point of view of future developments, significant challenges remain from both the theoretical and practical/technological points of view. Networked communication is currently at the experimental, test network TRL (technology readiness level) stage. In the future, highly scalable versions of these solutions will be needed, and the overall capabilities of network devices must also be enhanced. The importance of ensuring interoperability between devices from different manufacturers cannot be overemphasized, and to achieve this, it will also be necessary to harmonize the parallel standardization efforts.

Further open problems related to networked quantum services are the improvement of the accessibility of quantum services, the automatic generation and deployment of quantum services, as well as novel service engineering methods for service developers to simplify the development and deployment of quantum services. Another important direction for future research is the API standardization and middleware development to access quantum services of different service providers in a simplified and standardized way, and the development of a standardized architecture stack. The interoperability of classical and networked quantum services is another important subject, along with the development of hybrid high-performance quantum-classical networks. The integration of quantum key distribution and quantum cryptographic primitives into networked quantum services and quantum-classical architectures is another direction for research. Regarding quantum software of networked services, an important task is the development of a standardized programming language for the services. A current challenge for quantum software is the application of distributed algorithms and compilers. Most of the hardware challenges of networked quantum services coincidence with the challenges of the NISQ era devices. These problems are centered around decoherence and noise reduction, limited

capabilities and lack of error correction. An open research problem is the efficient implementation of networked quantum services via actual noisy quantum devices and shallow quantum circuits. By using intermediate quantum repeaters, the distant quantum devices can be coupled. However, the quantum hardware and quantum resources are limited by the capabilities of the current quantum devices, which cannot provide high performance generation and distribution of entanglement between the distant nodes. Another challenge is the handling and management of the noise of the quantum links and the reduction of communication overhead between current quantum devices. An important research problem is the integration of different enabling technologies to quantum devices to improve the performance of quantum networking, and the establishment of a scalable architecture of networked quantum services via the quantum internet.

This survey provided an overview of the concepts and attributes of networked quantum services. We studied the architectural background, the different applications, the quantum software and quantum programming languages, and the approaches to the standardization of quantum APIs and service accessing. The survey presented a comprehensive review of the state of the art of networked quantum services in a compact form. With this research, we expect to contribute to the further development of the field of networked quantum services.

Funding

This research was supported by the Ministry of Culture and Innovation and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004).

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. C. H. Bennett, S. J. Wiesner (1992). "Communication via 1- and 2-particle operators on Einstein-Podolsky-Rosen states". *Physical Review Letters*, 69, 2881–2884. <https://doi.org/10.1103/PhysRevLett.69.2881>.
2. C. H. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, W. K. Wootters (1993). "Teleporting an unknown quantum state via dual classic and Einstein-Podolsky-Rosen channels". *Physical Review Letters*, 70, 1895–1899. <https://doi.org/10.1103/PhysRevLett.70.1895>.
3. S. Imre and M. Berces (2021). "Entanglement-based competition resolution in distributed systems". *IEEE Access*, 9, 10253–10262. <https://doi.org/10.1109/ACCESS.2021.3050271>.
4. P. W. Shor (2002). "Introduction to quantum algorithms". *AMS PSAPM*, 58, 143–159. <https://doi.org/10.48550/arXiv.quant-ph/0005003>.
5. R. Jozsa (1998). "Quantum algorithms and the Fourier transform". *Proceedings of the Royal Society of London. Series A*, 454, 323–337. <https://doi.org/10.1098/rspa.1998.0163>.
6. S. Imre (2007). "Quantum existence testing and its application for finding extreme values in unsorted databases". *IEEE Transactions on Computers*, 56: 5, 706–710. <https://doi.org/10.1109/TC.2007.1032>.
7. L. Gyongyosi, S. Imre, H. V. Nguyen (2018). "A survey on quantum channel capacities". *IEEE Communications Surveys and Tutorials*, 20: 2. <https://doi.org/10.1109/COMST.2017.2786748>.
8. Q. A. Memon, M. Al Ahmad and M. Pecht, M (2024). "Quantum computing: Navigating the future of computation, challenges, and technological breakthroughs". *Quantum Reports*, 6, 627663. <https://doi.org/10.3390/quantum6040039>.
9. Z. Yang, M. Zolanvari, and R. Jain (2023). "A survey of important issues in quantum computing and communications". *IEEE Communications and Surveys Tutorials*, 25: 2, 1059–1094. <https://doi.org/10.1109/COMST.2023.3254481>.
10. H. Sahu, H. P. Gupta (2023). "Quantum computing toolkit from nuts and bolts to sack of tools". *arXiv:2302.08884*. <https://doi.org/10.48550/arXiv.2302.08884>.
11. P. Singh, R. Dasgupta, A. Singh et al. (2024). "A survey on available tools and technologies enabling quantum computing". *IEEE Access*, 57974–57991. <https://doi.org/10.1109/ACCESS.2024.3388005>.
12. L. Gyongyosi and S. Imre (2019). "A survey on quantum computing technology". *Computer Science Review*, 31, 51–71. <https://doi.org/10.1016/j.cosrev.2018.11.002>.
13. S. S. Gill et al (2021). "Quantum computing: a taxonomy, systematic review and future directions". *Software: Practice and Experience*. <https://doi.org/10.1002/spe.3039>.
14. E. Chae, J. Choi, J. Kim (2024). "An elementary review on basic principles and developments of qubits for quantum computing". *Nano Convergence*, 11, 11. <https://doi.org/10.1186/s40580-024-00418-5>.

15. A. Bochkarev, R. Heese, S. Jager et al (2024). "Quantum computing for discrete optimization: a highlight of three technologies". *arXiv:2409.01373*. <https://doi.org/10.48550/arXiv.2409.01373>.
16. Y. Li, M. Tian, G. Liu, C. Peng, and L. Jiao (2020). "Quantum optimization and quantum learning: A survey". *IEEE Access*, 8, 23 568–23 593. <https://doi.org/10.1109/ACCESS.2020.2970105>.
17. J. Kusyk, S. M. Saeed, and M. U. Uyar (2021). "Survey on quantum circuit compilation for noisy intermediate-scale quantum computers: Artificial intelligence to heuristics". *IEEE Transactions on Quantum Engineering*, 2, 1–16. <https://doi.org/10.1109/TQE.2021.3068355>.
18. P. Upama, M. J. Hossain Faruk, M. Nazim, M. Masum, H. Shahriar, G. Uddin, S. Barzanjeh, A. Rahman, and S. Ahamed (2022). "Evolution of quantum computing: a systematic survey on the use of quantum computing tools". *2022 IEEE Computers, Software, and Applications Conference*. <https://doi.org/10.1109/COMPSAC54236.2022.00096>.
19. F. V. Massoli, L. Vadicamo, G. Amato, and F. Falchi (2022). "A leap among quantum computing and quantum neural networks: A survey". *ACM Computing Surveys*, 55, 5. <https://doi.org/10.1145/3529756>.
20. S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi (2020). "Machine learning algorithms in quantum computing: A survey". *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. <https://doi.org/10.1109/IJCNN48605.2020.9207714>.
21. D. Peral-Garcia et al (2024). "Systematic literature review: Quantum machine learning and its applications". *Computer Science Review*, <https://doi.org/10.1016/j.cosrev.2024.100619>.
22. M. D. Garcia, A. M. Romero (2024). "Survey on computational applications of tensor network simulations". *IEEE Access*, DOI: 10.1109/ACCESS.2024.3519676. <https://doi.org/10.1109/ACCESS.2024.3519676>.
23. T. Ayril, P. Besserve, D. Lacroix, E. A. R. Guzman (2023). "Quantum computing with and for many-body physics". *The European Physical Journal A*, 59, 227. <https://doi.org/10.1140/epja/s10050-023-01141-1>.
24. E. Moguel, J. R. and V. Garcia-Alonso (2022). "Quantum service-oriented computing: current landscape and challenges". *Software Quality Journal*, 30, 9831002. <https://doi.org/10.1007/s11219-022-09589-y>.
25. M. A. Serrano, J. A. Cruz-Lemus, R.-P. Castillo and M. Piattini (2022). "Quantum software components and platforms: overview and quality assessment". *ACM Computing Surveys*, 55: 8 1–31. <https://doi.org/10.1145/3548679>.
26. A. A. Khan, A. Ahmad, M. Waseem, P. Liang, M. Fahmideh, T. Mikkonen, and P. Abrahamsson (2023). "Software architecture for quantum computing systems - A systematic review". *Journal of Systems and Software*, 201, 111682. <https://doi.org/10.1016/j.jss.2023.111682>.
27. L. Jimnez-Navajas, F. Bhler, F. Leymann et al (2024). "Quantum software development: a survey". *Quantum Information and Computation*, 24: 7–8, 0609–0642. <https://doi.org/10.26421/qic24.7-8-4>.
28. S. Garhwal, M. Ghorani, and A. Ahmad (2021). "Quantum programming language: A systematic review of research topic and top cited languages". *Archives of Computational Methods in Engineering*, 28, 89–310. <https://doi.org/10.1007/s11831-019-09372-6>.
29. K. Dwivedi, M. Haghparast, T. Mikkonen (2024). "Quantum software engineering and quantum software development lifecycle: a survey". *Cluster Computing*, 27, 7127–7145. <https://doi.org/10.1007/s10586-024-04362-1>.
30. D. Cuomo, M. Caleffi (2020). "Towards a distributed quantum computing ecosystem". *IET Quantum Communication*, 1: 1, 38. <https://doi.org/10.1049/iet-qtc.2020.0002>.
31. M. Caleffi, M. Amoretti, D. Ferrari et al (2024). "Distributed quantum computing: a survey". *Computer Networks*, 254, 110672. <https://doi.org/10.1016/j.comnet.2024.110672>.
32. D. Barrala, F. J. Cardamab, G. Diaz et al (2024). "Review of distributed quantum computing. From single QPU to high performance quantum computing". *arXiv:2404.01265v1*. <https://doi.org/10.48550/arXiv.2404.01265>.
33. J. C. Boschero, N. M. P. Neumann, W. van der Schoot, T. Sijpesteijn, R. Wezeman (2024). "Distributed quantum computing: applications and challenges". *arXiv:2410.00609*. <https://doi.org/10.48550/arXiv.2410.00609>.
34. G. M. Jones, H.-A. Jacobsen (2024). "Distributed quantum computing for chemical applications". *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. <https://doi.org/10.1109/QCE60285.2024.10270>.
35. L. Pira and C. Ferrie (2023). "An invitation to distributed quantum neural networks". *Quantum Machine Intelligence*, <https://doi.org/10.1007/s42484-023-00114-3>.
36. H. T. Nguyen, P. Krishnan, D. Krishnaswamy et al (2024). "Quantum cloud computing: a review, open problems, and future directions". *arXiv:2404.11420*. <https://doi.org/10.48550/arXiv.2404.11420>.

37. E. Moguel, J. Garcia-Alonso, J. M. Murillo (2024). “Development and deployment of quantum services”, in I. Exman et al. (eds), *Quantum Software*, Springer. https://doi.org/10.1007/978-3-031-64136-7_8.
38. F. Phillipson (2023). “Quantum computing in telecommunication-a survey”. *Mathematics*, 11: 15, 3423. <https://doi.org/10.3390/math11153423>.
39. Y. Baseri, V. Chouhan, A. Ghorbani (2024). “Cybersecurity in the quantum era: Assessing the impact of quantum computing on infrastructure”. *arXiv:2404.10659*. <https://doi.org/10.48550/arXiv.2404.10659>.
40. H. Dutta, A. K. Bhuyan (2024). “Quantum communication: from fundamentals to recent trends, challenges and open problems”. *arXiv:2406.04492*. <https://doi.org/10.48550/arXiv.2406.04492>.
41. Z. Li, K. Xue, J. Li, L. Chen, R. Li, Z. Wang, et al (2023). “Entanglement-assisted quantum networks: Mechanics, enabling technologies, challenges, and research directions”. *IEEE Communications Surveys and Tutorials*, 25: 4, 2133–2189. <https://doi.org/10.1109/COMST.2023.3294240>.
42. M. Mehic et al (2023). “Quantum cryptography in 5G networks: A comprehensive overview”. *IEEE Communications Surveys and Tutorials*, 26: 1, <https://doi.org/10.1109/COMST.2023.3309051>.
43. A. B. Popa, P. G. Popescu (2024). “The future of QKD networks”. *arXiv:2407.00877*. <https://doi.org/10.48550/arXiv.2407.00877>.
44. A. Abane, M. Cubeddu, V. S. Mai, A. Battou (2025). “Entanglement routing in quantum networks: a comprehensive survey”. *IEEE Transactions on Quantum Engineering*, 1–36. <https://doi.org/10.1109/TQE.2025.3541123>.
45. L. Gyongyosi, S. Imre (2022). “Advances in the quantum internet”. *Communications of the ACM*, 65: 8, 52–63. <https://doi.org/10.1145/3524455>.
46. S. Wehner, D. Elkouss, R. Hanson (2018). “Quantum internet: A vision for the road ahead”. *Science*, 362, 6412. <https://doi.org/10.1126/science.aam9288>.
47. Y. Li. et al (2024). “A survey of quantum internet protocols from a layered perspective”. *IEEE Communications Surveys and Tutorials*, 26: 3. <https://doi.org/10.1109/COMST.2024.3361662>.
48. J. Ang, G. Carini, Y. Chen, I. Chuang et al (2024). “Architectures for multinode superconducting quantum computers”. *ACM Transactions on Quantum Computing*, 5: 3, 1–59. <https://doi.org/10.1145/3674151>.
49. S. Barz et al (2012). “Demonstration of blind quantum computing”. *Science*, 335: 6066, 303–308. <https://doi.org/10.1126/science.1214707>.
50. S. Ruiting et al (2021). “Verifiable multiparty universal blind quantum computing in distributed network”’s. *Chinese Journal of Electronics*, 30: 4, 712–718. <https://dx.doi.org/10.1049/cje.2021.05.013>.
51. A. Mantri, C. Perez-Delgado and J. Fitzsimons (2013). “Optimal blind quantum computation”. *Physical Review Letters*, 111: 23, 230502. <https://doi.org/10.1103/PhysRevLett.111.230502>.
52. J. I. Cirac, A. K. Ekert, S. F. Huelga, C. Macchiavello (1999). “Distributed quantum computation over noisy channels”. *Physical Review A*, 59, 4249–4254. <https://doi.org/10.1103/PhysRevA.59.4249>.
53. D. Collins, N. Linden, S. Popescu (2001). “Nonlocal content of quantum operations”. *Physical Review A*, 64, 7. <https://doi.org/10.1103/PhysRevA.64.032302>.
54. J. Eisert, K. Jacobs, P. Papadopoulos, M. B. Plenio (2000). “Optimal local implementation of nonlocal quantum gates”. *Physical Review A*, 62: 5. <https://doi.org/10.1103/PhysRevA.62.052317>.
55. C. Gidney, M. Ekerä (2021). “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. *Quantum*, 5. <https://doi.org/10.22331/q-2021-04-15-433>.
56. L. Gyongyosi, S. Imre (2018). “Multiple access multicarrier continuous-variable quantum key distribution”. *Chaos, Solitons and Fractals*, 114, 491–505. <https://doi.org/10.1016/j.chaos.2018.07.006>.
57. L. Gyongyosi, S. Imre (2020). “Resource prioritization and balancing for the quantum internet”. *Scientific Reports*, [https://doi.org/DOI: 10.1038/s41598-020-78960-5](https://doi.org/DOI:10.1038/s41598-020-78960-5).
58. L. Gyongyosi, S. Imre (2021). “Scalable distributed gate-model quantum computers”. *Scientific Reports*, 11: 5172. <https://doi.org/10.1038/s41598-020-76728-5>.
59. H. Li, D. Qiu, L. Luo (2023). “Distributed exact quantum algorithms for Deutsch-Jozsa problem”. *arXiv.2303.10663*. <https://doi.org/10.48550/arXiv.2303.10663>.
60. N. M. Neumann, R. van Houte, T. Attema (2020). “Imperfect distributed quantum phase estimation”. *Computational Science-ICCS*, 2020: 12142 of LNCS, Springer, pp. 605–615. https://doi.org/10.1007/978-3-030-50433-5_46.
61. Y. Shi, T. Nguyen, S. Stein, T. Stavenger, M. Warner, M. Roetteler et al (2023). “A reference implementation for a quantum message passing interface”. *Proc. of the SC23 Workshops of The Int. Conf.*

- on *High Performance Computing, Network, Storage, and Analysis*, ACM, New York, pp. 1420–1425. <https://doi.org/10.1145/3624062.3624212>.
62. J. Tan, L. Xiao, D. Qiu, L. Luo, P. Mateus (2022). “Distributed quantum algorithm for Simon’s problem”. *Physical Review A*, 106: 3. <https://doi.org/10.1103/PhysRevA.106.032417>.
 63. R. Van Meter, W. J. Munro, K. Nemoto, K. M. Itoh (2008). “Arithmetic on a distributed-memory quantum multicomputer”. *ACM Journal on Emerging Technologies in Computing Systems*, 3: 4. <https://doi.org/10.1145/1324177.1324179>.
 64. L. Xiao, D. Qiu, L. Luo, P. Mateus (2023). “Distributed quantum-classical hybrid Shor’s algorithm”. *arXiv:2304.12100*. <https://doi.org/10.48550/arXiv.2304.12100>.
 65. Z. Zhang and Q. Zhuang (2021). “Distributed quantum sensing”. *Quantum Science and Technology*, 6: 4, 043001. <https://doi.org/10.1088/2058-9565/abd4c3>.
 66. C. L. Degen, F. Reinhard, and P. Cappellaro (2017). “Quantum sensing”. *Reviews of Modern Physics*, 89: 3, 035002. <https://doi.org/10.1103/RevModPhys.89.035002>.
 67. X. Zhou, D. Qiu, L. Luo (2023). “Distributed Bernstein-Vazirani algorithm”. *Physica A: Statistical Mechanics and its Applications*, 629, 129209. <https://doi.org/10.1016/j.physa.2023.129209>.
 68. X. Zhou, D. Qiu, L. Luo (2023). “Distributed exact Grover’s algorithm”. *Frontiers of Physics*, 18: 5, 1–25. <https://doi.org/10.1007/s11467-023-1327-x>.
 69. B. He, D. Zhang, S. W. Loke, S. Lin, L. Lu (2024). “Building a hierarchical architecture and communication model for the quantum internet”. *IEEE Journal on Selected Areas in Communications*, 42, 7. <https://doi.org/10.1109/JSAC.2024.3380103>.
 70. W. Kozłowski, S. Wehner, R. V. Meter, B. Rijsman, A. S. Cacciapuoti, M. Caleffi et al. (2023). *RFC 9340: Architectural Principles for a Quantum Internet*, Internet Research Task Force (IRTF). <https://doi.org/10.17487/RFC9340>.
 71. J. Roffe (2019). “Quantum error correction: an introductory guide”. *Contemporary Physics*, 60, 3. <https://doi.org/10.1080/00107514.2019.1667078>.
 72. A. Paler, S. J. Devitt (2015). “An introduction to fault-tolerant quantum computing”. *DAC’15 Proceedings of the 52nd Annual Design Automation Conference*, 60. <https://doi.org/10.1145/2744769.2747911>.
 73. K. Heshami, D. G. England, P. C. Humphreys (2016). “Quantum memories: emerging applications and recent advances”. *Journal of Modern Optics*, 63, S3, S42–S65. <https://doi.org/10.1080/09500340.2016.1148212>.
 74. B. M. Terhal (2015). “Quantum error correction for quantum memories”. *Reviews of Modern Physics*, 87, 307. <https://doi.org/10.1103/RevModPhys.87.307>.
 75. S. Bravyi, D. Gosset, and R. König (2018). “Quantum advantage with shallow circuits”. *Science*, 362, 308. <https://doi.org/10.1126/science.aar3106>.
 76. J. D. Hidary (2021). *Quantum Computing: an Applied Approach*. Springer, ISBN-10: 3030832732.
 77. T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien (2010). “Quantum computers”. *Nature*, 464: 7285, 4553. <https://doi.org/10.1038/nature08812>.
 78. M. AbuGhanem (2024). “Photonic quantum computers”. *arXiv:2409.08229*. <https://doi.org/10.48550/arXiv.2409.08229>.
 79. M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver (2020). “Superconducting qubits: Current state of play”. *Annual Review of Condensed Matter Physics*, 11, 369395. <https://doi.org/10.1146/annurev-conmatphys-031119-050605>.
 80. F. Arute et al (2019). “Quantum supremacy using a programmable superconducting processor”. *Nature*, 574. <https://doi.org/10.1038/s41586-019-1666-5>.
 81. R. Maurand, X. Jehl, D. Kotekar-Patil, A. Corna, H. Bohuslavskyi, R. Lavieville, L. Hutin, S. Barraud, M. Vinet, M. Sanquer et al (2016). “A cmos silicon spin qubit”. *Nature Communications*, 7: 1, 13575. <https://doi.org/10.1038/ncomms13575>.
 82. L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak (2020). “Quantum computing with neutral atoms”. *Quantum*, 4: 327. <https://doi.org/10.22331/q-2020-09-21-327>.
 83. R.-Y. Gong, Z.-Y. He, C.-H. Yu, G.-F. Zhang, F. Nori, Z.-L. Xiang (2024). “Tunable quantum router with giant atoms, implementing quantum gates, teleportation, non-reciprocity, and circulators”. *arXiv:2411.19307*. <https://doi.org/10.48550/arXiv.2411.19307>.
 84. C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage (2019). “Trapped-ion quantum computing: Progress and challenges”. *Applied Physics Reviews*, 6: 2, 021314. <https://doi.org/10.1063/1.5088164>.

85. C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. D. Sarma (2008). “Non-abelian anyons and topological quantum computation”. *Reviews of Modern Physics*, 80: 3, 1083. <https://doi.org/10.1103/RevModPhys.80.1083>.
86. S. Pezzagna, J. Meijer (2021). “Quantum computer based on color centers in diamond”. *Applied Physics Reviews*, 8, 011308. <https://doi.org/10.1063/5.0007444>.
87. L. M. K. Vandersypen et al (2001). “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance”. *Nature*, 414, 883–887. <https://doi.org/10.1038/414883a>.
88. L. M. K. Vandersypen, I. L. Chuang (2005). “NMR techniques for quantum control and computation”. *Reviews of Modern Physics*, 76, 1037. <https://doi.org/10.1103/RevModPhys.76.1037>.
89. N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, Y. Yamamoto (2012). “Layered architecture for quantum computing”. *Physical Review X*, 2, 031007. <https://doi.org/10.1103/PhysRevX.2.031007>.
90. S. Rodrigo, S. Abadal, E. Alarcon et al (2021). “On double full-stack communication-enabled architectures for multicore quantum computers”. *IEEE Micro*, 41, 4856. <https://doi.org/10.1109/MM.2021.3092706>.
91. R. Van Meter (2014). *Quantum Networking*. John Wiley and Sons Ltd.
92. J. Illiano, M. Caleffi, A. Manzalini, A. S. Cacciapuoti (2022). “Quantum Internet protocol stack: A comprehensive survey”. *Computer Networks*, 213, 109092. <https://doi.org/10.1016/j.comnet.2022.109092>.
93. Z. Li, K. Xue, J. Li, N. Yu, J. Liu, D. S. L. Wei et al (2021). “Building a large scale and wide-area quantum Internet based on an OSI-alike model”. *China Communications*, 18: 10, 1–14. <https://doi.org/10.23919/JCC.2021.10.001>.
94. A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpundineddek, M. Pompili et al (2019). “A link layer protocol for quantum networks”, in *Proceedings of the ACM Special Interest Group on Data Communication*, ACM, New York, 159–173. <https://doi.org/10.1145/3341302.3342070>.
95. A. Pirker, W. Dur (2019). “A quantum network stack and protocols for reliable entanglement-based networks”. *New Journal of Physics*, 21, 3. <https://doi.org/10.1088/1367-2630/ab05f7>.
96. R. V. Meter, J. Touch (2013). “Designing quantum repeater networks. *IEEE Communications Magazine*, 51: 8, 64–71. <https://doi.org/10.1109/MCOM.2013.6576340>.
97. R. Van Meter, R. Satoh, N. Benchasattabuse, K. Teramoto, T. Matsuo, M. Hajdusek et al. (2022). “A quantum Internet architecture”. *2022 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, 341–352. <https://doi.org/10.1109/QCE53715.2022.00055>.
98. J. Preskill (2018). “Quantum Computing in the NISQ era and beyond”. *Quantum*, 2, 79. <https://doi.org/10.22331/q-2018-08-06-79>.
99. A. W. Harrow and A. Montanaro (2017). “Quantum computational supremacy”. *Nature*, 549, 203–209. <https://doi.org/10.1038/nature23458>.
100. S. Aaronson and L. Chen (2017). “Complexity-theoretic foundations of quantum supremacy experiments”, in *Proceedings of the 32nd Computational Complexity Conference, CCC ’17*, pp. 22:1–22:67. <https://doi.org/10.4230/LIPIcs.CCC.2017.22>.
101. E. Farhi, J. Goldstone, S. Gutmann and H. Neven (2017). “Quantum algorithms for fixed qubit architectures”. *arXiv:1703.06199v1*. <https://doi.org/10.48550/arXiv.1703.06199>.
102. M. Mastriani (2024). “Explaining the results of quantum mechanics via entanglement closed loop”. *Optical and Quantum Electronics*, 56, 801. <https://doi.org/10.1007/s11082-024-06393-9>.
103. Y. Alexeev et al (2019). “Quantum computer systems for scientific discovery”. *PRX Quantum*, 2, 017001. <https://doi.org/10.1103/PRXQuantum.2.017001>.
104. M. Loncar et al (2019). “Development of quantum interconnects for next-generation information technologies”. *PRX Quantum*, 2, 017002. <https://doi.org/10.1103/PRXQuantum.2.017002>.
105. B. Foxen et al (2020). “Demonstrating a continuous set of two-qubit gates for near-term quantum algorithms”. *Physical Review Letters*, 125, 120504. <https://doi.org/10.1103/PhysRevLett.125.120504>.
106. A. Ajagekar, T. Humble, and F. You (2020). “Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems”. *Computers and Chemical Engineering*, 132, 106630. <https://doi.org/10.1016/j.compchemeng.2019.106630>.
107. A. Ajagekar and F. You (2019). “Quantum computing for energy systems optimization: Challenges and opportunities”. *Energy*, 179, 76–89. <https://doi.org/10.1016/j.energy.2019.04.186>.

108. M. Harrigan et al (2020). “Quantum approximate optimization of non-planar using eigen energies graph problems on a planar superconducting processor”. *Nature Physics*, 17, 332–336. <https://doi.org/10.1038/s41567-020-01105-y>.
109. N. Rubin et al (2020). “Hartree-Fock on a superconducting qubit quantum computer”. *Science*, 69: 6507, 1084–1089. <https://doi.org/10.1126/science.abb9811>.
110. E. Farhi, J. Goldstone and S. Gutmann (2014). “A quantum approximate optimization algorithm”. *arXiv:1411.4028v1*. <https://doi.org/10.48550/arXiv.1411.4028>.
111. E. Farhi, J. Goldstone, S. Gutmann and L. Zhou, L (2019). “The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size”. *Quantum*, 6, 759. <https://doi.org/10.22331/q-2022-07-07-759>.
112. E. Farhi and H. Neven (2018). “Classification with quantum neural networks on near term processors”. *arXiv:1802.06002v1*. <https://doi.org/10.48550/arXiv.1802.06002>.
113. I. Sax et al (2020). “Approximate approximation on a quantum annealer”. *CF '20: Proceedings of the 17th ACM International Conference on Computing Frontiers*, pp. 108–117. <https://doi.org/10.1145/3387902.3392635>.
114. K. A. Brown and T. Roser (2020). “Towards storage rings as quantum computers”. *Physical Review Accelerators and Beams*, 23, 054701. <https://doi.org/10.1103/PhysRevAccelBeams.24.049901>.
115. J. Bae, P. M. Alsing, D. Ahn et al (2020). “Quantum circuit optimization using quantum Karnaugh map”. *Scientific Reports*, 10, 15651, <https://doi.org/10.1038/s41598-020-72469-7>.
116. S. Li et al (2020). “Programmable unitary operations for orbital angular momentum encoded states”. *National Science Open*, 1: 2, 20220019. <https://doi.org/10.1360/nso/20220019>.
117. S. Bugu, F. Ozaydin and T. Koderä (2020). “Surpassing the classical limit in magic square game with distant quantum dots coupled to optical cavities”. *Scientific Reports*, DOI: <https://doi.org/10.1038/s41598-020-79295-x>.
118. A. Teplukhin, B. Kendrick and D. Babikov (2020). “Solving complex eigenvalue problems on a quantum annealer with applications to quantum scattering resonances”. *Physical Chemistry Chemical Physics*. <https://doi.org/10.1039/D0CP04272B>.
119. S. Lloyd (2018). “Quantum approximate optimization is computationally universal”. *arXiv:1812.11075*. <https://doi.org/10.48550/arXiv.1812.11075>.
120. Q. Zhu. et al (2022). “Quantum computational advantage via 60-qubit 24-cycle random circuit sampling”. *Science Bulletin*, 67: 3, 240–245. <https://doi.org/10.1016/j.scib.2021.10.017>.
121. S. Aaronson, A. Arkhipov (2013). “The computational complexity of linear optics”. *Theory of Computing*, 9: 4, 143–252. <https://doi.org/10.1145/1993636.1993682>.
122. H-S. Zhong et al (2020). “Quantum computational advantage using photons”. *Science*, 370: 6523, 1460–1463. <https://doi.org/10.1126/science.abe8770>.
123. L. S. Madsen et al (2022). “Quantum computational advantage with a programmable photonic processor”. *Nature*, 606: 7912, 75–81. <https://doi.org/10.1038/s41586-022-04725-x>.
124. M. Reiher, N. Wiebe, K. M. Svore et al (2017). “Elucidating reaction mechanisms on quantum computers”. *Proceedings of the National Academy of Sciences*, 114, 7555–7560. <https://doi.org/10.1073/pnas.1619152114>.
125. S. Mandra and H. G. Katzgraber (2018). “A deceptive step towards quantum speedup detection”. *Quantum Science Technology*, 3, 04LT. <https://doi.org/10.1088/2058-9565/aac8b2>.
126. V. Moret-Bonillo (2015). “Can artificial intelligence benefit from quantum computing?” *Progress in Artificial Intelligence*, 3, 89–105. <https://doi.org/10.1007/s13748-014-0059-0>.
127. T. Yamakawa and M. Zhandry (2022). “Verifiable quantum advantage without structure”, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, 69–74. <https://doi.org/10.1145/3658665>.
128. T. F. Ronnow, Z. Wang, J. Job et al (2014). “Defining and detecting quantum speedup”. *Science*, 345, 420–424. <https://doi.org/10.1126/science.1252319>.
129. S. Aaronson (2015). “Quantum machine learning algorithms: Read the fine print”, *Nature Physics*, 11, 291–293. <https://doi.org/10.1038/nphys3272>.
130. K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al (2022). “Noisy intermediate-scale quantum (NISQ) algorithms”. *Reviews of Modern Physics*, 94, 015004. <https://doi.org/10.1103/RevModPhys.94.015004>.
131. P. W. Shor (1994). “Algorithms for quantum computation: discrete logarithms and factoring”, in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. <https://doi.org/10.1109/SFCS.1994.365700>.

132. V. S. Denchev, S. Boixo, S. V. Isakov et al (2016). “What is the computational value of finite-range tunneling?” *Physical Review X*, 6, 1–17. <https://doi.org/10.1103/PhysRevX.6.031015>.
133. P. J. J. O’Malley et al (2016). “Scalable quantum simulation of molecular energies”. *Physical Review X*, 6, 031007. <https://doi.org/10.1103/PhysRevX.6.031007>.
134. F. Arute et al (2020). “Observation of separated dynamics of charge and spin in the Fermi-Hubbard model”. *arXiv:2010.07965*. <https://doi.org/10.48550/arXiv.2010.07965>.
135. I. Aleiner et al (2020). “Accurately computing electronic properties of materials using eigenenergies”. *arXiv:2012.00921*. <https://doi.org/10.1038/s41586-021-03576-2>.
136. E. Farhi, D. Gamarnik and S. Gutmann (2020). “The quantum approximate optimization algorithm needs to see the whole graph: a typical case”. *arXiv:2004.09002v1*. <https://doi.org/10.48550/arXiv.2004.09002>.
137. E. Farhi, D. Gamarnik and S. Gutmann (2020). “The quantum approximate optimization algorithm needs to see the whole graph: worst case examples”. *arXiv:2005.08747*. <https://doi.org/10.48550/arXiv.2005.08747>.
138. A. Peruzzo et al (2014). “A variational eigenvalue solver on a photonic quantum processor”. *Nature Communications*, 5, 4213. <https://doi.org/10.1038/ncomms5213>.
139. D. Wecker, M. B. Hastings, and M. Troyer (2015). “Progress towards practical quantum variational algorithms”. *Physical Review A*, 92, 042303. <https://doi.org/10.1103/PhysRevA.92.042303>.
140. J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik (2016). “The theory of variational hybrid quantum-classical algorithms”. *New Journal of Physics*, 18. <https://doi.org/10.1088/1367-2630/18/2/023023>.
141. X. Yuan et al (2019). “Theory of variational quantum simulation”. *Quantum*, 3, 191. <https://doi.org/10.22331/q-2019-10-07-191>.
142. A. Kardashin et al (2021). “Benchmarking variational quantum simulation against an exact solution”, in *023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Bellevue, WA, USA, 518–523. <https://doi.org/10.1109/QCE57702.2023.00065>.
143. K. Mitarai and K. Fujii (2021). “Constructing a virtual two-qubit gate by sampling single-qubit operations”. *New Journal of Physics*, 23. <https://doi.org/10.1088/1367-2630/abd7bc>.
144. T. Peng, A. W. Harrow, M. Ozols, and X. Wu (2020). “Simulating large quantum circuits on a small quantum computer”. *Physical Review Letters*, 125, 150504. <https://doi.org/10.1103/PhysRevLett.125.150504>.
145. L. Brenner, C. Piveteau, D. Sutter (2023). “Optimal wire cutting with classical communication”. *arXiv:2302.03366*. <https://doi.org/10.48550/arXiv.2302.03366>.
146. P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, et al. (2019). “Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families”. *arXiv:1907.13623*.
147. J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, et al (2022). “The variational quantum eigensolver: A review of methods and best practices”. *Physics Reports*, 986, 1128. <https://doi.org/10.1016/j.physrep.2022.08.003>.
148. Y. Wang, L. M. Sager-Smith, D. A. Mazziotti (2023). “Quantum simulation of bosons with the contracted quantum eigensolver”. *New Journal of Physics*, 25, 10. <https://doi.org/10.1088/1367-2630/acf9c3>.
149. M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran (2019). “Evaluating analytic gradients on quantum hardware”. *Physical Review A*, 99. <https://doi.org/10.1103/PhysRevA.99.032331>.
150. A. J. McCaskey, D. I. Lyakh, E. F. Dumitrescu, S. S. Powers, T. S. Humble (2019). “XACC: A system-level software infrastructure for heterogeneous quantum-classical computing”. *Quantum Science and Technology*, 5: 2. <https://doi.org/10.1088/2058-9565/ab6bf6>.
151. A. J. McCaskey, T. Nguyen, A. Santana, D. Claudino, T. Kharazi, H. Finkel (2021). “Extending C++ for heterogeneous quantum-classical computing”. *ACM Transactions on Quantum Computing*, 2: 2, 136. <https://doi.org/10.1145/3462670>.
152. J. K. Lee, O. T. Brown, M. Bull, M. Ruefenacht, J. Doerfert, M. Klemm et al. (2023). “Quantum task offloading with the openmp api”. *arXiv:2311.03210*. <https://doi.org/10.48550/arXiv.2311.03210>.
153. J. Gambetta (2022). *Quantum-centric Supercomputing: The Next Wave of Computing*. <https://www.ibm.com/quantum/blog/next-wave-quantum-centric-supercomputing>.
154. J. Gambetta (2023). *The Hardware and Software for the Era of Quantum Utility Is Here*. <https://www.ibm.com/quantum/blog/quantum-roadmap-2033>.
155. L. Liu, X. Dou (2021). “QuCloud: A new qubit mapping mechanism for multiprogramming quantum computing in cloud environment”, in *IEEE Int. Symposium on High-performance Computer Architecture (HPCA)*, IEEE, pp. 167178. <https://doi.org/10.1109/HPCA51647.2021.00024>.

156. L. Liu, X. Dou, QuCloud+ (2024). “A holistic qubit mapping scheme for single/multi-programming on 2D/3D NISQ quantum computers”. *ACM Transactions on Architecture and Code Optimization*, 21: 1, 127. <https://doi.org/10.1145/3631525>.
157. M. Bandic, C. G. Almudever, S. Feld (2023). “Interaction graph-based characterization of quantum benchmarks for improving quantum circuit mapping techniques”. *Quantum Machine Intelligence*, 5, 130. <https://doi.org/10.1007/s42484-023-00124-1>.
158. K. Li, D. Qiu, L. Li, S. Zheng, Z. Rong (2017). “Application of distributed semi-quantum computing model in phase estimation”. *Information Processing Letters*, 120, 23–29. <https://doi.org/10.1016/j.ipl.2016.12.002>.
159. T. Tanaka, Y. Suzuki, S. Uno, R. Raymond, T. Onodera, N. Yamamoto (2021). “Amplitude estimation via maximum likelihood on noisy quantum computer”. *Quantum Information Processing*, 20, 1–29. <https://doi.org/10.1007/s11128-021-03215-9>.
160. K. Zhang, P. Rao, K. Yu, H. Lim, V. Korepin (2021). “Implementation of efficient quantum search algorithms on NISQ computers”. *Quantum Information Processing*, 20, 233. <https://doi.org/10.1007/s11128-021-03165-2>.
161. G. Park, K. Zhang, K. Yu, V. Korepin (2023). “Quantum multi-programming for Grover’s search”. *Quantum Information Processing*, 22, 1. <https://doi.org/10.1007/s11128-022-03793-2>.
162. P. Das, S. S. Tannu, P. J. Nair, M. Qureshi (2019). “A case for multiprogramming quantum computers”, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 291–303. <https://doi.org/10.1145/3352460.3358287>.
163. S. Niu, A. Todri-Sanial (2023). “Enabling multi-programming mechanism for quantum computing in the NISQ era”. *Quantum*, 7. <https://doi.org/10.22331/q-2023-02-16-925>.
164. A. Ash-Saki, M. Alam, S. Ghosh (2020). “Analysis of crosstalk in NISQ devices and security implications in multi-programming regime”, in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 2530. <https://doi.org/10.1145/3370748.3406570>.
165. Y. Ohkura (2021). *Crosstalk-aware NISQ Multi-programming*. Bachelor’s thesis, Faculty Policy Manage., Keio Univ., Tokyo, Japan (2021).
166. R. Parekh, A. Ricciardi, A. Darwish, and S. DiAdamo (2021). “Quantum algorithms and simulation for parallel and distributed quantum computing”, in *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pp. 9–19. <https://doi.ieeecomputersociety.org/10.1109/QCS54837.2021.00005>.
167. P. Andres-Martinez, C. Heunen (2019). “Automated distribution of quantum circuits via hypergraph partitioning”. *Physical Review A*, 100: 3, 032308. <https://doi.org/10.1103/PhysRevA.100.032308>.
168. A. Pastor, P. Escofet, S. B. Rached, E. Alarcon, P. Barlet-Ros, S. Abadal (2024). “Circuit partitioning for multi-core quantum architectures with deep reinforcement learning”. *arXiv.2401.17976*. <https://doi.org/10.48550/arXiv.2401.17976>.
169. Y. Akhremtsev, T. Heuer, P. Sanders, S. Schlag (2017). “Engineering a direct k-way hypergraph partitioning algorithm”. *Proceedings of the Meeting on Algorithm Engineering and Experiments (ALENEX)*, SIAM, pp. 28–42. <https://doi.org/10.1137/1.9781611974768.3>.
170. S. Schlag, T. Heuer, L. Gottesburen, Y. Akhremtsev, C. Schulz, P. Sanders (2023). “High-quality hypergraph partitioning”. *ACM Journal of Experimental Algorithmics*, 27, 1–39. <https://doi.org/10.1145/3529090>.
171. J.-Y. Wu, K. Matsui, T. Forrer, A. Soeda, P. Andres-Martinez, D. Mills et al (2023). “Entanglement-efficient bipartite-distributed quantum computing”. *Quantum*, 7, 1196. <https://doi.org/10.22331/q-2023-12-05-1196>.
172. P. Andres-Martinez, T. Forrer, D. Mills, J.-Y. Wu, L. Henaut, K. Yamamoto et al (2023). “Distributing circuits over heterogeneous, modular quantum computing network architectures”. *Quantum Science and Technology*, 9, 4. <https://doi.org/10.1088/2058-9565/ad6734>.
173. R. G. Sundaram, H. Gupta, C. R. Ramakrishnan (2021). “Efficient distribution of quantum circuits”, in *35th Int. Symp. on Distributed Computing (DISC)*, 209 of Leibniz International Proceedings in Informatics (LIPIcs). <https://doi.org/10.4230/LIPIcs.DISC.2021.41>.
174. R. G. Sundaram, H. Gupta, C. R. Ramakrishnan (2022). “Distribution of quantum circuits over general quantum networks”, in *2022 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, pp. 415–425. <https://doi.ieeecomputersociety.org/10.1109/QCE53715.2022.00063>.
175. R. G. Sundaram, H. Gupta (2023). “Distributing quantum circuits using teleportations”, in *2023 IEEE Int. Conf. on Quantum Software (QSW)*, pp. 186–192. <https://doi.ieeecomputersociety.org/10.1109/QSW59989.2023.00030>.

176. Z. Davarzani, M. Zomorodi-Moghadam, M. Houshmand, M. Nouri-Baygi (2020). "A dynamic programming approach for distributing quantum circuits by bipartite graphs". *Quantum Information Processing*, 19, 9. <https://doi.org/10.1007/s11128-020-02871-7>.
177. B. W. Kernighan, S. Lin (1970). "An efficient heuristic procedure for partitioning graphs". *The Bell System Technical Journal*, 49: 2, 291–307. <https://doi.org/10.1002/j.1538-7305.1970.tb01770.x>.
178. M. Zomorodi-Moghadam, M. Houshmand, M. Houshmand (2018). "Optimizing teleportation cost in distributed quantum circuits". *International Journal of Theoretical Physics*, 57: 3, 848–861. <https://doi.org/10.1007/s10773-017-3618-x>.
179. M. Houshmand, Z. Mohammadi, M. Zomorodi-Moghadam, M. Houshmand (2020). "An evolutionary approach to optimizing teleportation cost in distributed quantum computation". *International Journal of Theoretical Physics*, 59: 4. <https://doi.org/10.1007/s10773-020-04409-0>.
180. O. Daei, K. Navi, M. Zomorodi-Moghadam (2020). "Optimized quantum circuit partitioning". *International Journal of Theoretical Physics*, 59, 12. <https://doi.org/3804-3820>.
181. E. Nikahd, N. Mohammadzadeh, M. Sedighi, M. S. Zamani (2021). "Automated window-based partitioning of quantum circuits". *Physica Scripta*, 96: 3, 035102. [10.1088/1402-4896/abd57c](https://doi.org/10.1088/1402-4896/abd57c)
182. C. M. Fiduccia, R. M. Mattheyses (1982). "A linear-time heuristic for improving network partitions". *Papers on Twenty-five Years of Electronic Design Automation*, pp. 175–181. <https://doi.org/10.1109/DAC.1982.1585498>.
183. W. Cambiucci, R. Silveira, W. Ruggiero (2023). "Hypergraphic partitioning of quantum circuits for distributed quantum computing", in *2023 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 268–269. <https://doi.org/10.1109/QCE57702.2023.10237>.
184. J. Clark, T. Humble, H. Thapliyal (2023). "TDAG: Tree-based directed acyclic graph partitioning for quantum circuits", in *Proceedings of the Great Lakes Symposium on VLSI 2023*, GLSVLSI 23, ACM, New York, USA, p. 587–592. <https://doi.org/10.1145/3583781.3590234>.
185. T. Park, C. Y. Lee (1995). "Algorithms for partitioning a graph". *Computers and Industrial Engineering*, 28: 4, 899909. [https://doi.org/10.1016/0360-8352\(95\)00003-J](https://doi.org/10.1016/0360-8352(95)00003-J).
186. J. M. Baker, C. Duckering, A. Hoover, F. T. Chong (2020). "Time-sliced quantum circuit partitioning for modular architectures", in *17th ACM Int. Conf. on Computing Frontiers 2020 Proceedings*, ACM, pp. 98–107. <https://doi.org/10.1145/3387902.3392617>.
187. P. Escofet, A. Ovide, C. G. Almudever, E. Alarcon, S. Abadal (2023). "Hungarian qubit assignment for optimized mapping of quantum circuits on multi-core architectures". *IEEE Computer Architecture Letters*. <https://doi.org/10.1109/LCA.2023.3318857>.
188. M. Bandic, L. Prielinger, J. Nublein, A. Ovide, S. Rodrigo, S. Abadal et al (2023). "Mapping quantum circuits to modular architectures with QUBO". *2023 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*, 1, IEEE, pp. 790801. <https://doi.org/10.1109/QCE57702.2023.00094>.
189. A. Ovide, S. Rodrigo, M. Bandic, H. Van Someren, S. Feld, S. Abadal et al. (2023). "Mapping quantum algorithms to multi-core quantum computing architectures". *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, IEEE, pp. 1–5. <https://doi.org/10.1109/ISCAS46773.2023.10181589>.
190. L. K. Grover (1997). "Quantum telecomputation". *arXiv.quant-ph/9704012*. <https://doi.org/10.48550/arXiv.quant-ph/9704012>.
191. M. Gupta, A. Pathak (2007). "A scheme for distributed quantum search through simultaneous state transfer mechanism". *Annalen der Physik*, 16: 12, 791797. <https://doi.org/10.1002/andp.200710265>.
192. A. Yimsiriwattana and S. J. Lomonaco (2004). "Distributed quantum computing: A distributed Shor algorithm", in *Quantum Information and Computation II*, 5436. International Society for Optics and Photonics, pp. 360372. DOI: <https://doi.org/10.1117/12.546504>.
193. M. Eker, J. Hastad (2017). "Quantum algorithms for computing short discrete logarithms and factoring RSA integers", in T. Lange, T. Takagi (Eds.), *Post-Quantum Cryptography*, no. 10346 in LNCS, Springer, pp. 347–363. https://doi.org/10.1007/978-3-319-59879-6_20.
194. Z.-Y. Chen, Q. Zhou, C. Xue, X. Yang, G.-C. Guo, and G.-P. Guo (2018). "64-Qubit Quantum Circuit Simulation". *Science Bulletin*, 63, 964. <https://doi.org/10.1016/j.scib.2018.06.007>.
195. A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon (2022). "Doubling the size of quantum simulators by entanglement forging". *PRX Quantum*, 3. <https://doi.org/10.1103/PRXQuantum.3.010309>.

196. M. A. Perlin, Z. H. Saleem, M. Suchara, and J. C. Osborn (2021). “Quantum circuit cutting with maximum likelihood tomography”. *npj Quantum Information*, 7: 1, 64. <https://doi.org/10.1038/s41534-021-00390-6>.
197. W. Tang, T. Tomesh, M. Suchara, J. Larson, and M. Martonosi (2021). “CutQC: using small Quantum computers for large Quantum circuit evaluations”, in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21)*, 473–486. <https://doi.org/10.1145/3445814.3446758>.
198. Z. H. Saleem, T. Tomesh, M. A. Perlin, P. Gokhale, and M. Suchara (2021). “Divide and conquer for combinatorial optimization and distributed quantum computation”. *arXiv:2107.07532*. <https://doi.org/10.48550/arXiv.2107.07532>.
199. A. Lowe et al (2023). “Fast quantum circuit cutting with randomized measurements”. *Quantum*, 7, 934. <https://doi.org/10.22331/q-2023-03-02-934>.
200. S. C. Marshall, C. Gyurik, and V. Dunjko (2022). “High dimensional quantum machine learning with small quantum computers”. *arXiv:2203.13739*. <https://doi.org/10.22331/q-2023-08-09-1078>.
201. C. Piveteau and D. Sutter (2023). “Circuit knitting with classical communication”. *IEEE Transactions on Information Theory*, DOI: 10.1109/TIT.2023.3310797. <https://doi.org/10.1109/TIT.2023.3310797>.
202. C. Tuysuz et al (2021). “Classical splitting of parametrized quantum circuits”. *Quantum Machine Intelligence*, 5, 34. <https://doi.org/10.1007/s42484-023-00118-z>.
203. K. Mitarai and K. Fujii (2021). “Overhead for simulating a non-local channel with local channels by quasiprobability sampling”. *Quantum*, 5, 388. <https://doi.org/10.22331/q-2021-01-28-388>.
204. S. Bravyi, G. Smith, and J. A. Smolin (2016). “Trading classical and quantum computational resources”. *Physical Review X*, 6. <https://doi.org/10.1103/PhysRevX.6.021043>.
205. A. W. Harrow, A. Lowe (2025). “Optimal quantum circuit cuts with application to clustered Hamiltonian simulation”. *PRX Quantum*, 6, 010316. <https://doi.org/10.1103/PRXQuantum.6.010316>.
206. Z. Zhou, Y. Du, X. Tian, D. Tao (2023). “QAOA-in-QAOA: Solving large-scale max-cut problems on small quantum machines”. *Physical Review Applied*, 19: 2, 024027. <https://doi.org/10.1103/PhysRevApplied.19.024027>.
207. K. Fujii, K. Mizuta, H. Ueda, K. Mitarai, W. Mizukami, Y. O. Nakagawa (2022). “Deep variational quantum eigensolver: A divide-and-conquer method for solving a larger problem with smaller size quantum computers”. *PRX Quantum*, 3. <https://doi.org/10.1103/PRXQuantum.3.010346>.
208. S. C. Marshall, C. Gyurik, V. Dunjko (2023). “High dimensional quantum machine learning with small quantum computers”. *Quantum*, 7. <https://doi.org/10.22331/q-2023-08-09-1078>.
209. D. Gottesman, I. L. Chuang (1999). “Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations”. *Nature*, 402, 6760, 390393. <https://doi.org/10.1038/46503>.
210. Y. F. Huang, X. F. Ren, Y. S. Zhang, L. M. Duan, G. C. Guo (2004). “Experimental teleportation of a quantum controlled-NOT gate”. *Physical Review Letters*, 93. <https://doi.org/10.1103/PhysRevLett.93.240501>.
211. K. S. Chou, J. Z. Blumoff, C. S. Wang, P. C. Reinhold, C. J. Axline, Y. Y. Gao, et al. (2018). “Deterministic teleportation of a quantum gate between two logical qubits”. *Nature*, 561: 7723, 368373. <https://doi.org/10.1038/s41586-018-0470-y>.
212. Y. Wan, D. Kienzler, S. Erickson, K. Mayer, T. Tan, J. Wu, et al (2019). “Quantum gate teleportation between separated qubits in a trapped-ion processor”. *Science*, 364, 875878. <https://doi.org/10.1126/science.aaw9415>.
213. S. Daiss, S. Langenfeld, S. Welte, E. Distant, P. Thomas, L. Hartung, et al (2021). “A quantum-logic gate between distant quantum-network modules”. *Science*, 371: 6529, 614617. <https://doi.org/10.1126/science.abe3150>.
214. C. Qiao, Y. Zhao, G. Zhao, H. Xu (2022). “Quantum data networking for distributed quantum computing: opportunities and challenges”, in *Proceedings of the IEEE Conference on Computer Communications Workshop*, New York, NY, pp. 16. <https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798138>.
215. Z. Cho, Y. Son, H. Jeong et al (2022). “A new approach to quantum computing multi-qubit generation and development of quantum computing platform with magnetic resonance imaging techniques”. *arXiv:2206.05932*. <https://doi.org/10.48550/arXiv.2206.05932>.
216. E. Alarcon, S. Abadal, F. Sebastiano et al (2023). “Scalable multi-chip quantum architectures enabled by cryogenic hybrid wireless/quantum-coherent network-in-package”, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Monterey, CA, USA, pp. 15. <https://doi.org/10.1109/ISCAS46773.2023.10181857>.

217. Y. Zheng, C. Zhai, D. Liu et al (2023). “Multichip multidimensional quantum networks with entanglement retrievability”. *Science*, 381, 221226. <https://doi.org/10.1126/science.adg9210>.
218. M. Field, A. Chen, B. Scharmann et al (2024). “Modular superconducting-qubit architecture with a multichip tunable coupler”. *Physical Review Applied*, 21, 054063. <https://doi.org/10.1103/PhysRevApplied.21.054063>.
219. J. Stokes, J. Izaac, N. Killoran, G. Carleo (2020). “Quantum natural gradient”. *Quantum*, 4, 269. <https://doi.org/10.22331/q-2020-05-25-269>.
220. M. S. Alvarez-Alvarado, F. E. Alban-Chacon, E. A. Lamilla-Rubio et al (2021). “Three novel quantum inspired swarm optimization algorithms using different bounded potential fields”. *Scientific Reports*, 11: 1, 11655. <https://doi.org/10.1038/s41598-021-90847-7>.
221. D. Wierichs, J. Izaac, C. Wang, C. Y.-Y. Lin (2022). “General parameter-shift rules for quantum gradients”. *Quantum*, 6, 677. <https://doi.org/10.22331/q-2022-03-30-677>.
222. E. R. Anschuetz, B. T. Kiani (2022). “Quantum variational algorithms are swamped with traps”. *Nature Communications*, 13: 1, 7760. <https://doi.org/10.1038/s41467-022-35364-5>.
223. D. Failde, J. D. Viqueira, M. Mussa Juane, A. Gomez (2023). “Using differential evolution to avoid local minima in variational quantum algorithms. *Scientific Reports*, 13: 1 16230. <https://doi.org/10.1038/s41598-023-43404-3>.
224. J. D. Viqueira, D. Failde, M. M. Juane, A. Gomez, D. Mera (2025). “Density matrix emulation of quantum recurrent neural networks for multivariate time series prediction”. *Machine Learning: Science and Technology*, 6, 015023. <https://doi.org/10.1088/2632-2153/ad9431>
225. D. Ferrari, S. Carretta, M. Amoretti (2023). “A modular quantum compilation framework for distributed quantum computing”. *IEEE Transactions on Quantum Engineering*, 4, 2500213. <https://doi.org/10.1109/TQE.2023.3303935>.
226. O. Mukhanov, B. L. T. Plourde, A. Opremcak et al (2019). “Scalable quantum computing infrastructure based on superconducting electronics”, in *Proceedings of the IEEE International Electron Devices Meeting*, San Francisco, CA, USA, pp. 31.2.131.2.4. <https://doi.org/10.1109/IEDM19573.2019.8993634>.
227. B. Baheri, Q. Guan, S. Xu, V. Chaudhary (2022). “SQCC: Smart quantum circuit cutting”, in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops*, Lyon, France, pp. 614615. <https://doi.org/10.1109/IPDPSW55747.2022.00104>.
228. K. Smith, M. Perlin, P. Gokhale et al (2023). “Clifford-based circuit cutting for quantum simulation”, in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando, FL, USA, pp. 113. <https://doi.org/10.1145/3579371.3589352>.
229. E. El-Araby, N. Mahmud, M. J. Jeng et al (2023). “Towards complete and scalable emulation of quantum algorithms on high-performance reconfigurable computers”. *IEEE Transactions on Computers*, 72, 23502364. <https://doi.org/10.1109/TC.2023.3248276>.
230. E. Andres, M. P. Cuellar, G. Navarro (2023). “Efficient dimensionality reduction strategies for quantum reinforcement learning”. *IEEE Access*, 11, 104534104553. <https://doi.org/10.1109/ACCESS.2023.3318173>.
231. M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini (2019). “Parameterized quantum circuits as machine learning models”. *Quantum Science and Technology*, 4. <https://doi.org/10.1088/2058-9565/ab4eb5>.
232. M. Schuld, I. Sinayskiy, and F. Petruccione (2014). “The quest for a quantum neural network”. *Quantum Information Processing*, 13. <https://doi.org/10.1007/s11128-014-0809-8>.
233. K. Beer, D. Bondarenko, T. Farrelly, T. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf. (2020). “Training deep quantum neural networks”. *Nature Communications*, 11, 808. <https://doi.org/10.1038/s41467-020-14454-2>.
234. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al (2021). “Variational quantum algorithms”. *Nature Reviews Physics*, 3, 625. <https://doi.org/10.1038/s42254-021-00348-9>.
235. S. Mangini, F. Tacchino, D. Gerace, D. Bajoni, and C. Macchiavello (2021). “Quantum computing models for artificial neural networks”. *Europhysics Letters*, 134, 10002. <https://doi.org/10.1209/0295-5075/134/10002>.
236. A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta (2017). “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. *Nature*, 549, 242. <https://doi.org/10.1038/nature23879>.
237. R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Faehrmann, B. Meynard-Piganeau, and J. Eisert (2020). “Stochastic gradient descent for hybrid quantum-classical optimization”. *Quantum*, 4, 314. <https://doi.org/10.22331/q-2020-08-31-314>.

238. J. Biamonte et al (2017). “Quantum machine learning”. *Nature*, 549, 195–202. <https://doi.org/10.1038/nature23474>.
239. Z.-C. Yang et al (2017). “Optimizing variational quantum algorithms using Pontryagin’s minimum principle”. *Physical Review X*, 7, 021027. <https://doi.org/10.1103/PhysRevX.7.021027>.
240. Y.-H. Oh et al (2019). “Solving multi-coloring combinatorial optimization problems using hybrid quantum algorithms”. *arXiv:1911.00595*. <https://doi.org/10.48550/arXiv.1911.00595>.
241. J. Biamonte (2021). “Universal variational quantum computation”. *Physical Review A*, 103: 3, L030401. <https://doi.org/10.1103/PhysRevA.103.L030401>.
242. V. Havlicek et al (2019). “Supervised learning with quantum-enhanced feature spaces”. *Nature*, 567: 7747, 209–212. <https://doi.org/10.1038/s41586-019-0980-2>.
243. A. Manzano et al (2023). “Parametrized quantum circuits and their approximation capacities in the context of quantum machine learning”. *arXiv:2307.14792*. <https://doi.org/10.48550/arXiv.2307.14792>.
244. V. Dunjko, J. M. Taylor, and H. J. Briegel (2016). “Quantum-enhanced machine learning”. *Physical Review Letters*, 117. <https://doi.org/10.1103/PhysRevLett.117.130501>.
245. L. Pira (2024). *The Fundamentals of Quantum Neural Networks*. PhD Dissertation. University of Technology Sydney.
246. E. Tang (2019). “A quantum-inspired classical algorithm for recommendation systems”, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. <https://doi.org/10.1145/3313276.3316310>.
247. S. Lloyd, M. Mohseni, and P. Rebentrost (2013). “Quantum algorithms for supervised and unsupervised machine learning”. *arXiv:1307.0411*. <https://doi.org/10.48550/arXiv.1307.0411>.
248. V. Giovannetti, S. Lloyd, and L. Maccone (2008). “Architectures for a quantum random access memory”. *Physical Review A*, 78. <https://doi.org/10.1103/PhysRevA.78.052310>.
249. V. Giovannetti, S. Lloyd, and L. Maccone (2008). “Quantum random access memory”. *Physical Review Letters*, 100. <https://doi.org/10.1103/PhysRevLett.100.160501>.
250. S. Arunachalam, V. Gheorghiu, T. Jochym-O’Connor, M. Mosca, and P. V. Srinivasan (2015). “On the robustness of bucket brigade quantum RAM”. *New Journal of Physics*, 17, 123010. <https://doi.org/10.1088/1367-2630/17/12/123010>.
251. L. Gyongyosi and S. Imre (2019). “Training optimization for gate-model quantum neural networks”. *Scientific Reports*, DOI: 10.1038/s41598-019-48892-w. <https://doi.org/10.1038/s41598-019-48892-w>.
252. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova (2019). “Machine learning and the physical sciences”. *Reviews of Modern Physics*, 91. <https://doi.org/10.1103/RevModPhys.91.045002>.
253. A. Dawid, J. Arnold, B. Requena, A. Gresch, M. P. lodzien, K. Donatella, K. A. Nicoli, P. Stornati, R. Koch, M. Buttner, et al (2022). “Modern applications of machine learning in quantum sciences”. *arXiv:2204.04198*. <https://doi.org/10.48550/arXiv.2204.04198>.
254. M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta (2018). “Reinforcement learning in different phases of quantum control”. *Physical Review X*, 8. <https://doi.org/10.1103/PhysRevX.8.031086>.
255. M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven (2019). “Universal quantum control through deep reinforcement learning”. *npj Quantum Information*, 5, 1. <https://doi.org/10.1038/s41534-019-0141-3>.
256. H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis (2019). “Optimizing quantum error correction codes with reinforcement learning”. *Quantum*, 3, 215. <https://doi.org/10.22331/q-2019-12-16-215>.
257. G. Torlai and R. G. Melko (2017). “A neural decoder for topological codes”. *Physical Review Letters*, 119, 030501. <https://doi.org/10.1103/PhysRevLett.119.030501>.
258. G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo (2018). “Many-body quantum state tomography with neural networks”. *Nature Physics*, 14. <https://doi.org/10.1038/s41567-018-0048-5>.
259. Q. Xu and S. Xu (2018). “Neural network state estimation for full quantum state tomography. *arXiv:1811.06654*. <https://doi.org/10.1103/PhysRevA.106.012409>.
260. G. Sentis, A. Monras, R. Munoz Tapia, J. Calsamiglia, and E. Bagan (2019). “Unsupervised classification of quantum data”. *Physical Review X*, 9, 041029. <https://doi.org/10.1103/PhysRevX.9.041029>.
261. N. Liu and P. Rebentrost (2018). “Quantum machine learning for quantum anomaly detection”. *Physical Review A*, 97, 042315. <https://doi.org/10.1103/PhysRevA.97.042315>.
262. S. Wang et al (2021). “Noise-induced barren plateaus in variational quantum algorithms”. *Nature Communications*, 12: 1, 6961. <https://doi.org/10.1038/s41467-021-27045-6>.

263. J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (2018). “Barren plateaus in quantum neural network training landscapes”. *Nature Communications*, 9. <https://doi.org/10.1038/s41467-018-07090-4>.
264. M. Cerezo, A. Sone, T. Volko, L. Cincio, and P. J. Coles (2021). “Cost function dependent barren plateaus in shallow parametrized quantum circuits”. *Nature Communications*, 12: 1. <https://doi.org/10.1038/s41467-021-21728-w>.
265. A. Arrasmith et al (2022). “Equivalence of quantum barren plateaus to cost concentration and narrow gorges”. *Quantum Science and Technology*, 7: 4, 045015. <https://doi.org/10.1088/2058-9565/ac7d06>.
266. M. Cerezo et al (2022). “Challenges and opportunities in quantum machine learning”. *Nature Computational Science*, 2: 9, 567–576. <https://doi.org/10.1038/s43588-022-00311-3>.
267. S. Herbst (2024). “Beyond 0s and 1s: Exploring the impact of noise, data encoding, and hyperparameter optimization in quantum machine learning”. *Diploma Thesis*, Technische Universitat Wien.
268. D. Suchan (2024). “Hyperparameter tuning for quantum machine learning”. *Diploma Thesis*, Technische Universitat Wien.
269. L. Alchieri et al (2021). “An introduction to quantum machine learning: from quantum logic to quantum deep learning”. *Quantum Machine Intelligence*, 3: 2, 28. <https://doi.org/10.1007/s42484-021-00056-8>.
270. N. Wiebe (2020). “Key questions for the quantum machine learner to ask themselves”. *New Journal of Physics*, 22, 091001. <https://doi.org/10.1088/1367-2630/abac39>
271. D. Ventura and T. Martinez (2000). “Quantum associative memory”. *Information Sciences*, 126, 273. [https://doi.org/10.1016/S0020-0255\(99\)00101-2](https://doi.org/10.1016/S0020-0255(99)00101-2).
272. R. LaRose and B. Coyle (2020). “Robust data encodings for quantum classifiers”. *Physical Review A*, 102. <https://doi.org/10.1103/PhysRevA.102.032420>.
273. M. Schuld and F. Petruccione (2018). *Supervised Learning with Quantum Computers*, Springer, DOI: 10.1007/978-3-319-96424-9.
274. H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean (2021). “Power of data in quantum machine learning”. *Nature Communications*, 12. <https://doi.org/10.1038/s41467-021-22539-9>.
275. M. Schuld and N. Killoran (2019). “Quantum machine learning in feature Hilbert spaces”. *Physical Review Letters*, 122, 040504. <https://doi.org/10.1103/PhysRevLett.122.040504>.
276. A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib (2021). “Layerwise learning for quantum neural networks”. *Quantum Machine Intelligence*, 3, 1. <https://doi.org/10.1007/s42484-020-00036-4>.
277. T. Haug, C. N. Self, and M. S. Kim (2023). “Quantum machine learning of large datasets using randomized measurements”. *Machine Learning: Science and Technology*, 4 015005. <https://doi.org/10.1088/2632-2153/acb0b4>.
278. Y. Du, Y. Qian, and D. Tao (2021). “Accelerating variational quantum algorithms with multiple quantum processors”. *arXiv:2106.12819*. <https://doi.org/10.48550/arXiv.2106.12819>.
279. C. Cade, L. Mineh, A. Montanaro, and S. Stanisic (2020). “Strategies for solving the Fermi-Hubbard model on near-term quantum computers”. *Physical Review B*, 102. <https://doi.org/10.1103/PhysRevB.102.235122>.
280. R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen (2020). “Exploring entanglement and optimization within the Hamiltonian Variational Ansatz”. *PRX Quantum*, 1. <https://doi.org/10.1103/PRXQuantum.1.020319>.
281. M. Weigold, J. Barzen, F. Leymann, and M. Salm (2020). “Data encoding patterns for quantum computing”, in *Proceedings of the 27th Conference on Pattern Languages of Programs*, <https://doi.org/10.5555/3511065.3511068>.
282. M. Weigold, J. Barzen, F. Leymann, and M. Salm (2021). “Expanding data encoding patterns for quantum algorithms”, in *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, pp. 95–101. <https://doi.org/10.1109/ICSA-C52384.2021.00025>.
283. M. Schuld, M. Fingerhuth and F. Petruccione (2017). “Implementing a distance-based classifier with a quantum interference circuit”. *Europhysics Letters*, 119, 60002. <https://doi.org/10.1209/0295-5075/119/60002>
284. J. Romero, J.P. Olson, and A. Aspuru-Guzik (2017). “Quantum autoencoders for efficient compression of quantum data”. *Quantum Science and Technology*, 2, 045001. <https://doi.org/10.1088/2058-9565/aa8072>.
285. M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko (2018). “Quantum Boltzmann Machine”. *Physical Review X*, 8, 021050. <https://doi.org/10.1103/PhysRevX.8.021050>.

286. R. Moretti, A. Giachero, V. Radescu, M. Grossi (2024). “Enhanced feature encoding and classification on distributed quantum hardware”. *arXiv:2412.01664*. <https://doi.org/10.48550/arXiv.2412.01664>.
287. T. Tomesh, P. Gokhale, E. R. Anschuetz, and F. T. Chong (2021). “Coreset clustering on small quantum computers”. *Electronics*, 10. <https://doi.org/10.3390/electronics10141690>.
288. A. W. Harrow (2020). “Small quantum computers and large classical data sets”. *arXiv:2004.00026*. <https://doi.org/10.48550/arXiv.2004.00026>.
289. C. N. Self, K. E. Khosla, A. W. R. Smith, F. Sauvage, P. D. Haynes, J. Knolle, F. Mintert, and M. S. Kim (2021). “Variational quantum algorithm with information sharing”. *npj Quantum Information*, 7. <https://doi.org/10.1038/s41534-021-00452-9>.
290. Y. Du, Y. Qian, X. Wu, D. Tao (2022). “A distributed learning scheme for variational quantum algorithms”. *IEEE Transactions on Quantum Engineering*, 3, 116. <https://doi.org/10.1109/TQE.2022.3175267>.
291. J. F. Fitzsimons (2017). “Private quantum computation: an introduction to blind quantum computing and related protocols”. *npj Quantum Information*, 3, 23. <https://doi.org/10.1038/s41534-017-0025-3>
292. R. Kaewpuang, M. Xu, D. T. Hoang et al (2023). “Elastic entangled pair and qubit resource management in quantum cloud computing”. *arXiv:2307.13185*. <https://doi.org/10.48550/arXiv.2307.13185>.
293. M. A. Serrano, L. E. Sanchez, A. Santos-Olmo et al (2023). “Minimizing incident response time in realworld scenarios using quantum computing”. *Software Quality Journal*, 32, 163192. <https://doi.org/10.1007/s11219-023-09632-6>.
294. P. J. Karalekas, T. Ryan, R. S. Smith (2020). “A quantum-classical cloud platform optimized for variational hybrid algorithms”. *Quantum Science and Technology*, 5, 2. <https://doi.org/10.1088/2058-9565/ab7559>.
295. Y. Ding and A. Javadi-Abhari. (2022). “Quantum and post-Moore’s law computing”. *IEEE Internet Computing*, 26: 1, 56. <https://doi.org/10.1109/MIC.2021.3133675>.
296. D. Subhi and L. Bacsardi (2023). “Using quantum nodes connected via the quantum cloud to perform IOT quantum network”. *Condensed Matter*, 8: 1, p. 24. <https://doi.org/10.3390/condmat8010024>.
297. IBM Quantum Computing Services, IBM (2023). <https://quantum-computing.ibm.com/>
298. IBM Cloud, IBM (2024). <https://www.ibm.com/cloud>
299. Google Cloud, Google (2024). Google Quantum AI. <https://quantumai.google/>
300. IonQ (2022). IonQ Cloud Service. <https://ionq.com/>
301. Quantinuum Cloud Service, Quantinuum (2022). <https://www.quantinuum.com/>.
302. Pasqal: Programmable Atomic Arrays (2024). <https://www.pasqal.com/>.
303. Azure Quantum, Microsoft (2023). <https://azure.microsoft.com/en-us/services/quantum>
304. QuEra, Quantum Computing with Neutral Atoms (2024). <https://www.quera.com/>
305. C. Gonzalez (2021). “Cloud based QC with Amazon Braket”. *Digitale Welt*, 5: 2, 14–17. <https://doi.org/10.1007/s42354-021-0330-z>.
306. Amazon Braket, Amazon (2024). <https://aws.amazon.com/braket/>
307. PlanQK (2020). <https://www.planqk.de/>
308. C. Linnhoff-Popien (2020). “PlanQK: Quantum computing meets artificial intelligence: How to make an ambitious idea reality”. *Digitale Welt*, 4: 2, 28–35. <https://doi.org/10.1007/s42354-020-0257-9>.
309. Quantum Inspire, De Voorhoeve (2024). <https://www.quantum-inspire.com/>
310. QuantumPath. <https://www.quantumpath.es/>
311. J. L. Hevia, G. Peterssen, M. Piattini (2021). “QuantumPath: A quantum software development platform”. *Software: Practice and Experience*, 1–14. <https://doi.org/10.1002/spe.3064>.
312. Strangeworks Quantum Computing Platform, Strangeworks (2022). <https://strangeworks.com/>
313. QFaaS, Github (2022). <https://github.com/Cloudslab/qfaas>
314. H. T. Nguyen, M. Usman, and R. Buyya (2024). “QFaaS: A serverless function-as-a-service framework for quantum computing”. *Future Generation Computer Systems*, 154, 281–300. <https://doi.org/10.1016/j.future.2024.01.018>.
315. 1Qloud Optimization Platform, 1QBit (2024). <https://1qbit.com/1qloud/>
316. Qemist Cloud, Good Chemistry Co (2024). <https://goodchemistry.com/qemist-cloud/>
317. N. Aslam, H. Zhou, E. K. Urbach, M. J. Turner, R. L. Walsworth, M. D. Lukin, and H. Park (2023). “Quantum sensors for biomedical applications. *Nature Reviews Physics*, 5: 3, 157–169. <https://doi.org/10.1038/s42254-023-00558-3>.

318. L. Thiel, D. Rohner, M. Ganzhorn, P. Appel, E. Neu, B. Muller, R. Kleiner, D. Koelle, and P. Maletinsky (2016). “Quantitative nanoscale vortex imaging using a cryogenic quantum magnetometer”. *Nature Nanotechnology*, 11: 8, 677–681. <https://doi.org/10.1038/nnano.2016.63>.
319. H. M. J. Boffin, G. Hussain, J.-P. Berger, and L. Schmidtbreick (2016). “Astronomy at high angular resolution”. *Astrophysics and Space Science Library*, 439. <https://doi.org/10.1007/978-3-319-39739-9>.
320. Y. Zhang et al (2021). “QED driven QAOA for networkflow optimization”. *Quantum*, 5: 510. <https://doi.org/10.22331/q-2021-07-27-510>.
321. B. Chevalier, W. Roga, M. Takeoka (2024). “Compressed sensing enhanced by a quantum approximate optimization algorithm”. *Physical Review A*, 110, 062410. <https://doi.org/10.1103/PhysRevA.110.062410>.
322. J. Li, M. Alam, and S. Ghosh (2023). “Large-scale quantum approximate optimization via divide-and-conquer”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42: 6, 18521860. <https://doi.org/10.1109/TCAD.2022.3212196>.
323. G. D. Kahanamoku-Meyer et al (2022). “Classically verifiable quantum advantage from a computational Bell test”. *Nature Physics*, 18: 8, 918–924. <https://doi.org/10.1038/s41567-022-01643-7>.
324. Z. Brakerski et al (2021). “A cryptographic test of quantumness and certifiable randomness from a single quantum device”. *Journal of the ACM (JACM)*, 68: 5, 1–47. <https://doi.org/10.1145/3441309>.
325. H. R. Grimsley et al (2019). “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. *Nature Communications*, 10: 1, 3007. <https://doi.org/10.1038/s41467-019-10988-2>.
326. Y.-F. Niu, S. Zhang, C. Ding, W.-S. Bao, and H.-L. Huang (2023). “Parameter-parallel distributed variational quantum algorithm”. *SciPost Physics*, 14, 5. <https://doi.org/10.21468/SciPostPhys.14.5.132>.
327. D. Bluvstein et al (2024). “Logical quantum processor based on reconfigurable atom arrays”. *Nature*, 626: 7997, 58–65. <https://doi.org/10.1038/s41586-023-06927-3>.
328. J. Bartusek et al (2023). “Obfuscation of pseudo-deterministic quantum circuits”, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 1567–1578. <https://doi.org/10.1145/3564246.3585179>.
329. J. Rajakumar et al (2024). “Polynomial-time classical simulation of noisy IQP circuits with constant depth”. *arXiv:2403.14607*. <https://doi.org/10.48550/arXiv.2403.14607>.
330. S. Chen, J. Cotler, H.-Y. Huang, and J. Li (2023). “The complexity of NISQ”. *Nature Communications*, 14, 6001. <https://doi.org/10.1038/s41467-023-41217-6>.
331. V. Dunjko and H. J. Briegel (2018). “Machine learning and artificial intelligence in the quantum domain”. *Reports on Progress in Physics*, 81, 074001. <https://doi.org/10.1088/1361-6633/aab406>.
332. J. A. Cortese, Timothy M. Braje (2018). “Loading classical data into a quantum computer”. *arXiv:1803.01958*. <https://doi.org/10.48550/arXiv.1803.01958>.
333. Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik (2017). “Quantum neuron: An elementary building block for machine learning on quantum computers”. *arXiv:1711.11240*. <https://doi.org/10.48550/arXiv.1711.11240>.
334. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii (2018). “Quantum circuit learning”. *Physical Review A*, 98. <https://doi.org/10.1103/PhysRevA.98.032309>.
335. J. Allcock, C.-Y. Hsieh, I. Kerenidis, and S. Zhang (2020). “Quantum algorithms for feedforward neural networks”. *ACM Transactions on Quantum Computing*, 1. <https://doi.org/10.1145/3411466>.
336. M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe (2020). “Circuit-centric quantum classifiers”. *Physical Review A*, 101. <https://doi.org/10.1103/PhysRevA.101.032308>.
337. Q.-X. Xie, Y. Zhao (2024). “A novel quantum-classical hybrid algorithm for determining eigenstate energies in quantum chemistry”. *arXiv:2406.00296*. <https://doi.org/10.48550/arXiv.2406.00296>.
338. C. C. Hian (2024). *Resource-Efficient Quantum Algorithms for Quantum Chemistry*. PhD Dissertation, National University of Singapore.
339. J. Bausch (2018). “Classifying data using near-term quantum devices”. *International Journal of Quantum Information*, 16, 8. <https://doi.org/10.1142/S0219749918400014>.
340. M. L. Wall et al (2021). “Generative machine learning with tensor networks: Benchmarks on near-term quantum computers”. *Physical Review Research*, 3: 2. <https://doi.org/10.1103/PhysRevResearch.3.023010>.
341. B. Coecke et al (2020). “Foundations for near-term quantum natural language processing”. *arXiv:2012.03755*. <https://doi.org/10.48550/arXiv.2012.03755>.
342. M. Alam et al (2022). “QNet: A scalable and noise-resilient quantum neural network architecture for noisy intermediate-scale quantum computers”. *Frontiers in Physics*, 9. <https://doi.org/10.3389/fphy.2021.755139>.

343. S. Lloyd and C. Weedbrook (2018). “Quantum generative adversarial learning”. *Physical Review Letters*, 121. <https://doi.org/10.1103/PhysRevLett.121.040502>.
344. P.-L. Dallaire-Demers and N. Killoran (2018). “Quantum generative adversarial networks”. *Physical Review A*, 98, 012324. <https://doi.org/10.1103/PhysRevA.98.012324>.
345. C. Zoufal, A. Lucchi, and S. Woerner (2019). “Quantum generative adversarial networks for learning and loading random distributions”. *npj Quantum Information*, 5. <https://doi.org/10.1038/s41534-019-0223-2>.
346. M. Henderson, S. Shakyia, S. Pradhan, and T. Cook (2019). “Quantum convolutional neural networks: powering image recognition with quantum circuits”. *arXiv:1904.04767*. <https://doi.org/10.48550/arXiv.1904.04767>.
347. J. Bausch (2020). “Recurrent quantum neural networks”, in H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (eds), *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 33, pp. 1368–1379. <https://doi.org/10.48550/arXiv.2006.14619>.
348. I. Kerenidis, J. Landman, and A. Prakash (2020). “Quantum algorithms for deep convolutional neural networks”. *arXiv:1911.01117*. <https://doi.org/10.48550/arXiv.1911.01117>.
349. I. Cong, S. Choi, and M. D. Lukin (2019). “Quantum convolutional neural networks”. *Nature Physics*, 15, 1273. <https://doi.org/10.1038/s41567-019-0648-8>.
350. E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti (2019). “An initialization strategy for addressing barren plateaus in parametrized quantum circuits”. *Quantum*, 3. <https://doi.org/10.22331/q-2019-12-09-214>.
351. A. Pesah, M. Cerezo, S. Wang, T. Volk, A. T. Sornborger, and P. J. Coles (2021). “Absence of barren plateaus in quantum convolutional neural networks”. *Physical Review X*, 11, 041011. <https://doi.org/10.1103/PhysRevX.11.041011>.
352. Y. Ding, L. Lamata, M. Sanz, X. Chen, and E. Solano (2019). “Experimental implementation of a quantum autoencoder via quantum adders”. *Advanced Quantum Technologies*, 2, 1800065. <https://doi.org/10.1002/qute.201800065>.
353. A. Rocchetto, S. Aaronson, S. Severini, G. Carvacho, D. Poderini, I. Agresti, M. Bentivegna, and F. Sciarrino (2019). “Experimental learning of quantum states”. *Science Advances*, 5, eaau1946. <https://doi.org/10.1126/sciadv.aau1946>.
354. B. Coyle, D. Mills, V. Danos, and E. Kashe (2020). “The born supremacy: Quantum advantage and training of an Ising Born Machine”. *npj Quantum Information*, 6, 1. <https://doi.org/10.1038/s41534-020-00288-9>.
355. M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz (2019). “A generative modeling approach for benchmarking and training shallow quantum circuits”. *npj Quantum Information*, 5, 1. <https://doi.org/10.1038/s41534-019-0157-8>.
356. F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni (2019). “An artificial neuron implemented on an actual quantum processor”. *npj Quantum Information*, 5, 1. <https://doi.org/10.1038/s41534-019-0140-4>.
357. E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini (2018). “Hierarchical quantum classifiers”. *npj Quantum Information*, 4, 1. <https://doi.org/10.1038/s41534-018-0116-9>.
358. D. Riste, M. P. Da Silva, C. A. Ryan, A. W. Cross, A. D. Corcoles, J. A. Smolin, J. M. Gambetta, J. M. Chow, and B. R. Johnson (2017). “Demonstration of quantum advantage in machine learning”. *npj Quantum Information*, 3, 1. <https://doi.org/10.1038/s41534-017-0017-3>.
359. J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, et al (2017). “Unsupervised machine learning on a hybrid quantum computer”. *arXiv:1712.05771*. <https://doi.org/10.48550/arXiv.1712.05771>.
360. B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, M. Troyer, and K. Svore (2020). “Quantum programming languages”. *Nature Reviews Physics*, 2, 12, 709–722. <https://doi.org/10.1038/s42254-020-00245-7>.
361. A. Tavakoli, A. Pozas-Kerstjens, P. Brown, M. Araujo (2024). “Semidefinite programming relaxations for quantum correlations”. *Reviews of Modern Physics*, 96, 045006. <https://doi.org/10.1103/RevModPhys.96.045006>.
362. R. LaRose (2019). “Overview and comparison of gate level quantum software platforms”. *Quantum*, 3, 130. <https://doi.org/10.22331/q-2019-03-25-130>.
363. M. Fingerhuth, T. Babej, and P. Wittek (2018). “Open source software in quantum computing”. *PloS One*, 13, 12, e0208561. <https://doi.org/10.1371/journal.pone.0208561>.

364. J. Zhao (2020). “Quantum software engineering: landscapes and horizons”. *arXiv:2007.07047*. <https://doi.org/10.48550/arXiv.2007.07047>.
365. J. M. Murillo et al. (2024). “Quantum software engineering: roadmap and challenges ahead”. *ACM Transactions on Software Engineering and Methodology*. <https://doi.org/10.1145/3712002>.
366. J. Ray. (2022). “China at the nexus of AI and quantum computing”. *Chinese Power and Artificial Intelligence: Perspectives and Challenges*, 155–172.
367. M. Ying, Y. Feng, R. Duan et al (2012). “Quantum programming: From theories to implementations”. *Chinese Science Bulletin*, 57, 1903–1909. <https://doi.org/10.1007/s11434-012-5147-6>.
368. I. Kumara, W. J. Van Den Heuvel, D. A. Tamburri (2021). “Qsoc: Quantum service-oriented computing”, in *Symposium and Summer School on Service-Oriented Computing*, 5263. https://doi.org/10.1007/978-3-030-87568-8_3.
369. J. Romero-Alvarez, J. Alvarado-Valiente, E. Moguel, J. Garcia-Alonso (2023). “Quantum web services: Development and deployment”, in *Web Engineering*. Springer, Alicante, Spain, pp. 421–423. <https://doi.org/10.1007/978-3-031-34444-239>.
370. V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash Narayanan, A. Asadi, et al (2018). “PennyLane: Automatic differentiation of hybrid quantum-classical computations”. *arXiv:1811.04968*. <https://doi.org/10.48550/arXiv.1811.04968>.
371. G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernandez, J. Carballo-Franquis, A. Chen, C.-F. Chen, et al (2019). “Qiskit: An open-source framework for quantum computing”. <https://doi.org/10.5281/zenodo.2562111>.
372. H. Nagarajan, O. Lockwood and C. Coffrin (2021). “QuantumCircuitOpt: An open-source framework for provably optimal quantum circuit design”. *arXiv:2111.11674*. <https://doi.org/10.48550/arXiv.2111.11674>.
373. QuPanda: an open source quantum computing framework (2018). “Origin quantum”, <https://github.com/OriginQ/QPanda-2>
374. S. Diadamo, J. Notzel, B. Zanger, and M. M. Bese (2021). “Distributed quantum computing and network control for accelerated VQE”. *IEEE Transactions on Quantum Engineering*, 2, 1. <https://doi.org/10.1109/TQE.2021.3057908>.
375. T. Chatterjee, A. Das, S. I. Mohtashim, A. Saha, A. Chakrabarti (2022). “Qurzon: A prototype for a divide and conquer-based quantum compiler for distributed quantum systems”. *SN Computer Science*, 3, 4. <https://doi.org/10.1007/s42979-022-01207-9>.
376. W. Tang, M. Martonosi, ScaleQC (2022). “A scalable framework for hybrid computation on quantum and classical processors”. *arXiv:2207.00933*. <https://doi.org/10.48550/arXiv.2207.00933>.
377. K. N. Smith, M. A. Perlin, P. Gokhale, P. Frederick, D. Owusu-Antwi, R. Rines et al. (2023). “Clifford-based circuit cutting for quantum simulation”. *Proc. of the 50th Annual Int. Symp. on Computer Architecture (ISCA'23)*, ACM, New York, pp. 113. <https://doi.org/10.1145/3579371.3589352>.
378. “TorchQuantum: Fast library for quantum computing in PyTorch”. (2022). <https://hanruiwanghw.wixsite.com/torchquantum>
379. 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 (2020). “TensorFlow quantum: A software framework for quantum machine learning”. *arXiv:2003.02989*. <https://doi.org/10.48550/arXiv.2003.02989>.
380. C. Xing and M. Broughton (2021). “Training with multiple workers using tensor flow quantum”. <https://blog.tensorflow.org>.
381. N. N. Hegade, N. L. Kortikar, B. Das, B. K. Behera, and P. K. Panigrahi (2017). “Experimental demonstration of quantum tunneling in IBM quantum computer”. *arXiv:1712.07326*. <https://doi.org/10.48550/arXiv.1712.07326>.
382. H. Norlen (2020). *Quantum Computing in Practice with Qiskit and IBM Quantum Experience: Practical Recipes for Quantum Computer Coding at the Gate and Algorithm Level with Python*. Packt Publishing Ltd.
383. D. Fortunato, J. Campos, and R. Abreu (2022). “Mutation testing of quantum programs: A case study with qiskit”. *IEEE Transactions on Quantum Engineering*, 3, 1–17. <https://doi.org/10.1109/TQE.2022.3195061>.
384. Pyquil, Rigetti Computing (2017). <https://www.rigetti.com/applications/pyquil>
385. K. Singhal, K. Hietala, S. Marshall, and R. Rand (2022). “Q# as a quantum algorithmic language”. *arXiv:2206.03532*. <https://doi.org/10.4204/EPTCS.394.10>.
386. The Quipper Language (2013). <https://www.mathstat.dal.ca/~selinger/quipper/>

387. J. M. Smith, N. J. Ross, P. Selinger, and B. Valiron (2014). “Quipper: Concrete resource estimation in quantum algorithms”. *arXiv:1412.0625*. <https://doi.org/10.48550/arXiv.1412.0625>.
388. Dwave Ocean (2018). <https://docs.ocean.dwavesys.com/en/stable/>
389. D. Willsch, M. Willsch, C. D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson, and K. Michielsen (2022). “Benchmarking advantage and Dwave 2000q quantum annealers with exact cover problems”. *Quantum Information Processing*, 21: 4, 141. <https://doi.org/10.1007/s11128-022-03476-y>.
390. R. Forest (2020). <https://qcs.rigetti.com/sdk-downloads>
391. J. Olivares-Sanchez, J. Casanova, E. Solano, and L. Lamata (2020). “Measurement-based adaptation protocol with quantum reinforcement learning in a Rigetti quantum computer”. *Quantum Reports*, 2: 2, 293–304. <https://doi.org/10.3390/quantum2020019>.
392. Microsoft Quantum SDK, Microsoft (2022). <https://learn.microsoft.com/en-us/azure/quantum/qsharp-overview>
393. J. Hooyberghs (2022). *Introducing Microsoft Quantum Computing for Developers: Using the Quantum Development Kit and Q#*, Apress, ISBN-10: 1484272455.
394. Strawberry Fields, Xanadu (2018). <https://strawberryfields.ai/>
395. ProjectQ: An open source software framework for quantum computing (2022). <https://github.com/ProjectQ-Framework/ProjectQ>
396. Cirq: An open source framework for programming quantum computers. <https://quantumai.google/cirq>
397. J. R. McClean, N. C. Rubin, K. J. Sung, I. D. Kivlichan, X. Bonet-Monroig, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby et al (2020). “Openfermion: the electronic structure package for quantum computers”. *Quantum Science and Technology*, 5: 3, 034014. <https://doi.org/10.1088/2058-9565/ab8ebc>.
398. J. R. Johansson, P. D. Nation, and F. Nori (2012). “Qutip: An open-source Python framework for the dynamics of open quantum systems”. *Computer Physics Communications*, 183: 8, 1760–1772. <https://doi.org/10.1016/j.cpc.2012.02.021>.
399. J. Garcia-Alonso, J. Rojo, D. Valencia, E. Moguel, J. Berrocal, and J. M. Murillo (2021). “Quantum software as a service through a quantum API gateway”. *IEEE Internet Computing*, 26: 1, 34–41. <https://doi.org/10.1109/MIC.2021.3132688>.
400. T. L. Patti, J. Kossaifi, S. F. Yelin, and A. Anandkumar (2021). “Tensorly-quantum: Quantum machine learning with tensor methods”. *arXiv:2112.10239*. <https://doi.org/10.48550/arXiv.2112.10239>.
401. C. F. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski (2020). “Detecting multiple communities using quantum annealing on the D-wave system”. *Plos One*, 15: 2, e0227538. <https://doi.org/10.1371/journal.pone.0227538>.
402. J. Romero-Alvarez, J. Alvarado-Valiente, E. Moguel et al (2023). “Leveraging API specifications for scaffolding quantum applications”, in *International Workshop on Quantum Software Engineering and Technology(QCE23)*. IEEE Quantum Week 2023. <https://doi.org/10.1109/QCE57702.2023.10208>.
403. I. Kumara, W.-J. Van Den Heuvel, D. A. Tamburri (2021). “Qsoc: Quantum service-oriented computing”, in *Symposium and Summer School on Service-Oriented Computing*, Springer, pp. 52–63. https://doi.org/10.1007/978-3-030-87568-8_3.
404. J. Alvarado-Valiente, J. Romero-Alvarez, E. Moguel, J. Garcia-Alonso, J. M. Murillo (2023). “Technological diversity of quantum computing providers: a comparative study and a proposal for API Gateway integration”. *Software Quality Journal*, 32, 5373. <https://doi.org/10.1007/s11219-023-09633-5>.
405. S. Schwichtenberg, C. Gerth, G. Engels (2017). “From Open API to semantic specifications and code adapters” in *2017 IEEE International Conference on Web Services (ICWS)*, pp. 484–491. <https://doi.org/10.1109/ICWS.2017.56>.
406. J. Romero-Alvarez, J. Alvarado-Valiente, J. Casco-Seco et al. (2023). “Developing high-level abstractions for creating quantum services: Open API and Async API”, in *Symposium and Summer School On Service-Oriented Computing (SummerSOC 2023)*, Crete, Greece. https://doi.org/10.1007/978-3-031-64136-7_8.
407. B. Heim (2021). “Universal quantum intermediate representation”. *APS March Meeting Abstracts*, 2021, 34-009.
408. Y. Ohkura, T. Satoh, R. Van Meter (2022). “Simultaneous execution of quantum circuits on current and near-future NISQ systems”. *IEEE Transactions on Quantum Engineering*, 3. <https://doi.org/10.1109/TQE.2022.3164716>.

409. M. Kaliyanandi, J. Murugan, S. K. Subburaj et al (2023). "Design and development of novel security approach designed for cloud computing with load balancing", in *Advances in Intelligent Applications and Innovative Approach*, 2760, 050005. <https://doi.org/10.1063/5.0126814>.
410. E. Karacan, A. Karakaya, S. Akleyek (2022). "Quantum secure communication between service provider and SIM". *IEEE Access*, 10, 69135-69146. <https://doi.org/10.1109/ACCESS.2022.3186306>.
411. M. Salm, J. Barzen, F. Leymann, B. Weder (2022). "Prioritization of compiled quantum circuits for different quantum computers", in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 1258-1265. <https://doi.org/10.1109/SANER53432.2022.00150>.
412. J. Alvarado-Valiente, J. Romero-Alvarez, D. Arias et al (2023). "Improving the quality of quantum services generation process: Controlling errors and noise", in *Hybrid Artificial Intelligent Systems*. Springer, Salamanca, Spain, pp. 180–191. https://doi.org/10.1007/978-3-031-40725-3_16.
413. J. Alvarado-Valiente, J. Romero-Alvarez, A. Diaz et al (2023). "Quantum services generation and deployment process: A quality-oriented approach", in *Quality of Information and Communications Technology*. Springer, Aveiro, Portugal, pp. 200–214. https://doi.org/10.1007/978-3-031-43703-8_15.
414. V. Silva (2018). *Practical Quantum Computing for Developers*, Programming Quantum Rigs in the Cloud Using Python, Quantum Assembly Language and IBM QExperience. Apress, USA.
415. B. Weder, J. Barzen, F. Leymann et al (2021). "Qprov: A provenance system for quantum computing". *IET Quantum Communication*, 2: 4, 171–181. <https://doi.org/10.1049/qtc2.12012>.
416. B. Weder, J. Barzen, J. F. Leymann, M. Zimmermann (2021). "Hybrid quantum applications need two orchestrations in superposition: A software architecture perspective", in *2021 IEEE International Conference on Web Services (ICWS)*, pp. 1–13. <https://doi.org/10.1109/ICWS53863.2021.00015>.
417. S. Imre, L. Gyongyosi (2013). *Advanced Quantum Communications: An Engineering Approach*. Wiley-IEEE Press, 488 pp. ISBN: 978-1-118-00236-0.
418. W. Cai, X. Mu, W. Wang et al. (2024). "Protecting entanglement between logical qubits via quantum error correction". *Nature Physics*, 20, 1022–1026. <https://doi.org/10.1038/s41567-024-02446-8>.
419. N. Benchasattabuse, M. Hajdusek and R. Van Meter (2024). "Engineering challenges in all-photon quantum repeaters". *IEEE Network*. <https://doi.org/10.1109/MNET.2024.3411802>.
420. A. K. Nayak, E. Chitambar, L. R. Varshney (2024). "Reliable quantum memories with unreliable components". *Physical Review A*, 110, 032423. <https://doi.org/10.1103/PhysRevA.110.032423>.
421. W. Dur, H.J. Briegel (2007). "Entanglement purification and quantum error correction". *Reports on Progress in Physics*, 70: 8, 1381–1424. DOI: 10.1088/0034-4885/70/8/R03.
422. C. Wang, A. Rahman, R. Li, M. Aelmans, K. Chakraborty (2024). "Application Scenarios for the Quantum Internet". *RFC 9583 Internet-Draft*. <https://datatracker.ietf.org/doc/rfc9583/>.
423. D. J. Reilly (2019). "Challenges in scaling-up the control interface of a quantum computer". *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, pp. 31.7.1–31.7.6. <https://doi.org/10.1109/IEDM19573.2019.8993497>.
424. A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland (2012). "Surface codes: Towards practical large-scale quantum computation". *Physical Review A*, 86, 32324. <https://doi.org/10.1103/PhysRevA.86.032324>.
425. T. S. Metodi, D. D. Thaker, A. W. Cross et al. "A quantum logic array microarchitecture: scalable quantum data movement and computation" in *Proc. 38th Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO)*, p. 12. <https://doi.org/10.1109/MICRO.2005.9>
426. D. Aharonov, M. Ben-Or (1999). "Fault-tolerant quantum computation with constant error rate". *arXiv:quant-ph/9906129*. <https://doi.org/10.48550/arXiv.quant-ph/9906129>.
427. Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien (2023). "Quantum error mitigation". *Reviews of Modern Physics*, 95, 045005. <https://link.aps.org/doi/10.1103/RevModPhys.95.045005>.
428. S. M. Attia, S. Q. G. Haddad, H. K. R. Al-Zaidi et al (2024). "Quantum computing impact on traditional computer architecture models". *Radioelectronics. Nanosystems. Information Technologies*, 16: 5, 691704. <http://dx.doi.org/10.17725/j.rensit.2024.16.691>.
429. S. Muralidharan, L. Li, J. Kim et al (2016). "Optimal architectures for long distance quantum communication". *Scientific Reports*, 6, 20463. <https://doi.org/10.1038/srep20463>.
430. P. Wang, C. Y. Luan, M. Qiao et al. et al (2021). "Single ion qubit with estimated coherence time exceeding one hour". *Nature Communications*, 12, 233. <https://doi.org/10.1038/s41467-020-20330-w>.

431. Y. Wang, M. Um, J. Zhang et al (2017). “Single-qubit quantum memory exceeding ten-minute coherence time”. *Nature Photon*, 11, 646–650. <https://doi.org/10.1038/s41566-017-0007-1>.
432. C.E. Bradley, J. Randall, M.H. Abobeih et al (2019). “A ten-qubit solid-state spin register with quantum memory up to one minute”. *Physical Review X*, 9, 031045. <https://doi.org/10.1103/PhysRevX.9.031045>.
433. M. H. Abobeih, J. Cramer, M. A. Bakker et al (2018). “One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment”. *Nature Communications*, 9, 2552. <https://doi.org/10.1038/s41467-018-04916-z>.
434. O. Milul, B. Guttel, U. Goldblatt, S. Hazanov et al (2023). “Superconducting cavity qubit with tens of milliseconds single-photon coherence time”. *PRX Quantum*, 4, 030336. <https://doi.org/10.1103/PRXQuantum.4.030336>.