EPJ Quantum Technology

RESEARCH

Open Access

Circuit-based vs. measurement-based quantum computing: a comparative analysis, layered metrics, and decision flow for approach selection

Harashta Tatimma Larasati[1,2] and Byung-Soo Choi[1*]

*Correspondence:
bschoi@pknu.ac.kr
[1]Department of Scientific Computing, Pukyong National University, Busan, 48513, Republic of Korea
Full list of author information is available at the end of the article

Abstract

Circuit-based quantum computing (CBQC), also known as the gate model, is currently regarded as the leading paradigm for performing quantum computation and has garnered widespread interest due to its relatively better-established implementation and close intuitive resemblance to classical computation. However, an alternative model that solely utilizes single-qubit measurements on entangled states to perform computation, namely the measurement-based quantum computing (MBQC), has started to attract further research interest as it is deemed more naturally suited for certain quantum hardware architectures, though its actual implementation remains largely underexplored. Recently, there have been preliminary efforts to characterize the specific regimes in which MBQC may potentially be more advantageous compared to CBQC. Nevertheless, a more systematic assessment is necessary to assess the suitability of each model across a broader range of more practical cases. In this work, we present a detailed comparative analysis of CBQC and MBQC, beginning with reviewing the theoretical foundations of each model before highlighting its distinguishing features that set the two paradigms apart. Furthermore, building on previous work, we propose a generalized regime analysis that more accurately captures the conditions under which a given quantum algorithm is more favorably executed in MBQC or CBQC. Subsequently, we identify critical parameters that likely determine whether a quantum algorithm is better suited to CBQC or MBQC, categorizing detailed key factors and performance metrics into layers of quantum computation. Next, we propose a decision flow that, given specific hardware constraints and algorithm requirements, guides the selection to the most appropriate model, discussing suitability in two representative worked examples of quantum algorithm applications: Quantum Fourier Transform (QFT) and Variational Quantum Algorithm (VQE). Last, we discuss the potential to combine both models as a hybrid approach to leverage the best of both worlds and explore its benefits for practical quantum computing applications.

Keywords: Circuit-based quantum computing; Comparative analysis; Decision flow; Measurement-based quantum computing; Review

1 Introduction

The rapid advancement of quantum computing research in the past two decades has largely been dominated by the circuit-based quantum computing (CBQC) paradigm. Building upon a direct generalization of classical circuit models, CBQC employs sequences of unitary operations (quantum gates) applied to qubits, and has become the de facto standard for both theoretical formulations and physical hardware realizations [1]. Numerous quantum algorithms, including Shor's factoring algorithm [2] and Grover's search algorithm [3], have been developed and implemented within this framework, driving the growth of (mostly) CBQC-based hardware platforms such as superconducting qubits and trapped ions.

However, measurement-based quantum computing (MBQC), a fundamentally different paradigm introduced by Raussendorf and Briegel in the early 2000s [4], has increasingly gained attention in recent years. Unlike CBQC, which relies on sequential gate operations, MBQC performs computation through a sequence of adaptive single-qubit measurements on a highly entangled initial state, typically a cluster state. While historically underexplored compared to its circuit-based counterpart, MBQC has demonstrated unique advantages in areas such as fault tolerance, distributed quantum computing, and resource optimization for specific classes of algorithms [5, 6].

Despite the growing interest in MBQC and the emergence of hybrid models that aim to integrate features from both paradigms, the number of comprehensive studies directly comparing CBQC and MBQC remains limited. Existing reviews and surveys have primarily focused on individual models or specific application areas, leaving a significant gap in literature regarding their systematic comparison across operational, resource, and architectural dimensions.

In this paper, we present a comprehensive comparative analysis of CBQC and MBQC, carefully examining their computational structures, resource requirements, fault-tolerance schemes, and algorithmic suitability. We argue that while CBQC maintains its predominance, MBQC possesses distinct merits that render it more advantageous in certain scenarios, particularly in distributed and fault-tolerant quantum architectures. It is thus imperative to evaluate these paradigms side-by-side to identify cases where one model may be preferential over the other.

Departing from the work of Lee et al. [6], which initially proposed a threshold model for determining algorithmic suitability between CBQC and MBQC based on select criteria, we extend this line of inquiry by identifying additional parameters from both theoretical and technical viewpoints. This includes considerations of circuit depth, entanglement structure, resource state preparation, measurement adaptivity, and fault-tolerance overhead.

Furthermore, we construct a mapping of several well-known quantum algorithms to their more suitable computational model, considering factors such as resource efficiency, error mitigation capabilities, and hardware compatibility. This mapping is intended to serve as a practical guide for selecting appropriate quantum computing frameworks based on algorithmic characteristics and implementation contexts.

Lastly, we discuss the potential of hybrid quantum computing architectures that seek to leverage the complementary strengths of CBQC and MBQC. Notable examples include fusion-based quantum computing [7], which integrates MBQC's resource-efficient measurement schemes with modular photonic systems, and recent CBQC-based schemes augmented by MBQC-inspired measurement layers for state preparation and error suppression [8]. To date, however, no study has explicitly demonstrated MBQC models infused with CBQC subroutines despite the direction demands further investigation as quantum hardware capabilities mature.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical foundations and historical development of both CBQC and MBQC. Subsequently, Sect. 3 presents a comparative analysis highlighting the distinguishing features that set the two paradigms apart, ranging from their conceptual differences, resource implications, fault tolerance and error correction characteristics, hardware suitability, to scalability. In Sect. 4], prior work's regime analyses on the algorithmic favorability of CBQC versus MBQC are first reviewed, followed by the introduction of our proposed generalized regime analysis that extends beyond fixed-ratio assumptions. Next, Sect. 5 elaborates on our layered framework, which identifies and organizes key factors and performance metrics across various layers of quantum computation, followed by discussions in Sect. 7. Finally, Sect. 8 concludes the paper.

2 Theoretical foundations

2.1 Evolution of quantum computing paradigms

Evolution of CBQC: from early gate model to modern implementations The establishment of quantum computing paradigms traces back to the late 1970s and early 1980s, when researchers began to explore the feasibility of computational processes governed by quantum mechanical principles. The seminal work of David Deutsch [9] introduced the concept of a universal quantum computer, formalizing the circuit-based quantum computing (CBQC) model as a direct generalization of classical Boolean circuits. This framework utilizes quantum bits (qubits) as information carriers, manipulated by a sequence of unitary quantum gates interspersed with measurements.

CBQC quickly established itself as the primary computational model for quantum algorithms, largely due to its conceptual compatibility with classical computation and the emergence of landmark algorithms such as Shor's factoring algorithm [2] and Grover's search algorithm [3]. Over subsequent decades, significant advancements in quantum hardware (e.g., superconducting qubits, trapped ions, and photonic systems) further solidified CBQC's prominence as the most widely adopted paradigm for quantum computation.

Emergence of MBQC: from one-way quantum computer to distributed architectures While CBQC remained the dominant framework, a novel computational model emerged in the early 2000s. Introduced by Raussendorf and Briegel [4], measurement-based quantum computing (MBQC) —also referred to as one-way quantum computing —proposed a radically different approach wherein computation is driven by a sequence of adaptive, single-qubit measurements performed on a highly entangled initial resource state, typically a cluster state.

In MBQC, the unitary evolution of qubits is effectively replaced by classical feedforward control over measurement outcomes, offering unique flexibility in distributed quantum networks and photonic systems [5]. The model's intrinsic separation between entanglement generation and measurement-driven computation renders it particularly advantageous for scenarios involving modular architectures, fault tolerance, and scalable photonic-based quantum systems.

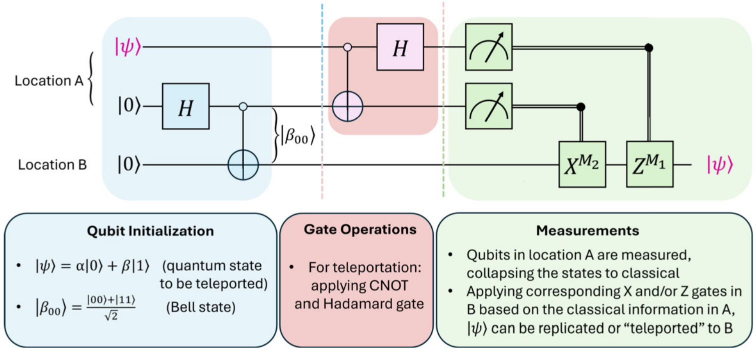

Figure 1 Example of a gate teleportation circuit in the quantum circuit (CBQC) paradigm [12]

The past decade has witnessed a gradual increase in MBQC-related research, particularly in contexts such as fault-tolerant quantum computation [10], distributed quantum computing [11], and photonic quantum processors [7]. Notably, the recent exploration of hybrid models integrating CBQC and MBQC components has highlighted the complementary strengths of these paradigms, motivating the need for comparative analyses such as the present work.

2.2 Circuit-based quantum computation (CBQC)

The circuit-based quantum computation (CBQC) paradigm, also known as the gate-model QC, is predicated on the sequential application of quantum gates—unitary operations acting on one or more qubits—structured into a quantum circuit, as illustrated in Fig. 1. A typical CBQC process involves three primary stages: qubit initialization, gate operations, and final measurement. Quantum information is manipulated via standard gate sets, such as the Clifford+T gate set, known for its universality and relevance in fault-tolerant quantum computing [1].

CBQC's operational model aligns closely with classical digital circuits, facilitating intuitive algorithm design and circuit analysis. Resource estimation in CBQC typically involves metrics such as circuit depth, circuit width, and gate count, often with emphasis on costly non-Clifford gates like the T gate. Recent developments in fault-tolerant architectures and surface code implementations have further refined CBQC's practicality for near-term and future quantum devices.

While CBQC offers a clear, well-established computational model and extensive hardware support, it can encounter challenges in deep circuit implementations due to coherence time limitations and gate error accumulation, especially on near-term noisy intermediate-scale quantum (NISQ) hardware.

2.3 Measurement-based quantum computation (MBQC)

Measurement-based quantum computation (MBQC) offers an alternative framework wherein the computational process is driven by single-qubit measurements performed on a pre-entangled cluster state or graph state. Introduced by Raussendorf and Briegel [4], MBQC delineates the processes of entanglement generation and computational evolution, allowing for flexible, adaptive computation steered by classical control based on prior measurement outcomes.

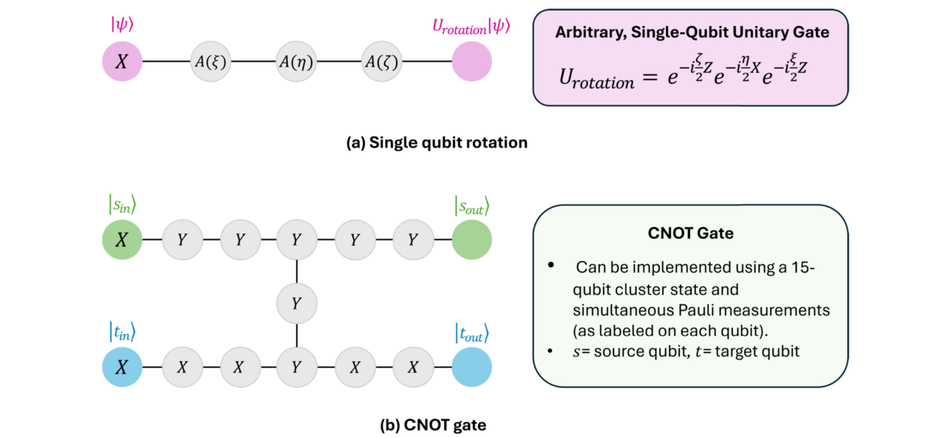

Figure 2 Example of (a) a single qubit rotation and (b) a CNOT gate in the MBQC paradigm [12]

The MBQC protocol begins by preparing a highly entangled cluster state, followed by a carefully designed sequence of adaptive, single-qubit measurements in specific bases. The choice of measurement basis and the order of operations determine the realized quantum algorithm, with classical feedforward control used to adjust subsequent measurements based on earlier results, as shown in Fig. 2. This model replaces the direct gate-sequence structure of CBQC with a measurement-driven flow (c.f. Fig. 3), offering advantages in scenarios where entanglement resources can be generated efficiently or distributed across a network.

Key benefits of MBQC include its suitability for distributed quantum computing, photonic systems, and fault-tolerant architectures leveraging topological cluster states [10]. MBQC's inherent decoupling of state preparation and computational progression also enables more robust modularization, particularly valuable in long-distance quantum communication and scalable photonic implementations.

3 Distinguishing features of circuit-based and measurement-based quantum computation

While both CBQC and MBQC serve as universal models for quantum information processing, they are built upon distinct conceptual frameworks, which result in differences in perceiving resource complexity. In addition, this also gives rise to variation in fault-tolerance mechanisms, quantum hardware suitability, and scalability. In this section, we detail the distinguishing features that characterize each model to provide a foundation for the analysis presented in the subsequent sections. In detail, the comparison includes conceptual differences, resource complexity implications, fault tolerance and error correction aspects, quantum hardware suitability, and scalability considerations. For each part, a summary table is provided for clarity.

3.1 Conceptual differences

Here, we discuss the conceptual differences between CBQC and MBQC in several aspects: computational model, the role of entanglement and measurement, how it performs (gate) operations, and their processing of quantum information, which is also summarized in Table 1.

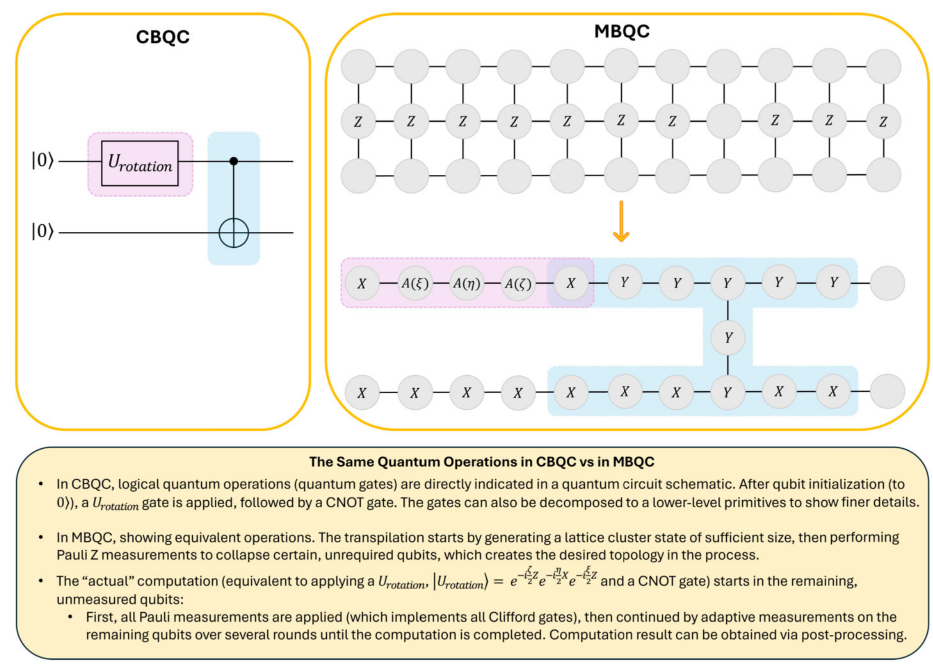

Figure 3 Example of a transpilation of the same quantum operations from CBQC to MBQC [12]

Table 1 Comparison of Conceptual Differences of CBQC and MBQC, compiled from [13, 14]

Aspect	CBQC	MBQC
Computational Model	Sequential gate application	Adaptive measurement on cluster states
Entanglement	Generated dynamically when required	Pre-prepared, fundamental
Operations	Unitary gates	Measurement patterns
Measurement	At final step (or optionally, mid-circuit)	Throughout computation
Information Processing	Unitary evolution, deterministic	Projective measurement, probabilistic

3.1.1 Computational model

In CBQC, the computation is performed by initializing a set of qubits on a quantum register of a quantum circuit, applying a sequence of unitary operations (i.e., quantum gates), then measuring the qubits to extract the corresponding classical information. The computation proceeds through discrete time steps, with each gate transforming the quantum state according to a prescribed quantum circuit layout [1]. This model closely follows classical logic circuits, thus deemed more intuitive, which contributes to its prominence. Coupled with, arguably, much higher support from the industry such as IBM and Google, it has shown relatively significant progress and has been the most widely adopted framework for practical quantum computing implementations [15].

In contrast, MBQC realizes computation by first preparing a highly entangled resource state, typically a two-dimensional cluster state. Then, the computation is then performed solely through a sequence of adaptive, single-qubit measurements. The logic of the algorithm (i.e., the quantum gate equivalent of CBQC) is encoded in the pattern and basis of these measurements, with classical processing (i.e., feed-forward) used to adapt subsequent measurement choices based on previous outcomes [4]. This approach shifts the computational complexity from dynamic gate application to the initial entanglement structure and measurement sequence. Hence, in MBQC, measurement is the core of the computational process, rather than unitary evolution as in CBQC, in which measurement is only performed at the end (or optionally, also in the middle) to obtain the computation result.

3.1.2 Role of entanglement

In CBQC, entanglement is generated dynamically during computation by applying multi-qubit gates (e.g., CNOT gate or Toffoli gate) when necessary. Though the utilization of entanglement is typically essential for most quantum algorithms, not all quantum subroutines require entanglement at every stage.

In contrast, entanglement is, in fact, a prerequisite for MBQC. In detail, the entire resource state must be prepared in a highly entangled configuration before the computation even begins. Then, the computation's power derives from this initial entanglement, which is then "consumed" as measurements are performed [16].

3.1.3 Role of measurement

For CBQC, measurement typically occurs only at the end of the computation (though depending on the desired algorithm and technique, mid-circuit measurement is possible [17]) to extract the results as a classical outcome. In MBQC, measurement is the central computational primitive, performed adaptively throughout the process, with outcomes used to determine the bases for subsequent measurements. This necessitates a fast classical processing so that the mechanism does not be a bottleneck in the computation.

3.1.4 Operations

CBQC implements operations (i.e., quantum gates) directly as physical operations applied to the qubits. This is done via single-qubit gates (e.g., Hadamard, Pauli gates, T-gate) and/or two-qubit gates[1] (e.g., CNOT and CZ gate). These gates are applied in a specified order, and their physical realization depends on the underlying hardware.

On the other hand, MBQC realizes quantum gates through sequences of single-qubit measurements on the resource state. The effect of a measurement depends on the measurement basis and the structure of the entangled state. By carefully choosing measurement patterns and adapting them based on previous outcomes, MBQC can implement any quantum gate. Clifford gates correspond to fixed measurement patterns of X, Y, or Z (thus, can be performed simultaneously rather than sequentially like in CBQC), while non-Clifford gates require adaptive strategies, i.e., sequential based on the result of previous measurement, which is essentially implemented by destructive measurements [10].

3.1.5 Processing of quantum information

CBQC processes quantum information via explicit gate operations, with quantum states evolving unitarily except at the final measurement. Hence, the computation in CBQC, which is performed through unitary evolution of quantum gates, is deterministic. In contrast, MBQC processes information through the propagation of measurement-induced collapses across the entangled resource [13]. Note that as MBQC relies on projective,

[1] Or via higher multi-qubit gates such as Toffoli/CCNOT gate, but which will eventually be decomposed to a series of one and two-qubit gates.

adaptive measurements, its computation is probabilistic, with undesired randomness introduced to the computation [18]. To alleviate this randomness inherent in quantum measurement, a classical feedforward is employed to ensure correct logical operation despite stochastic outcomes. Hence, classical aspects are also as essential in the workings of MBQC [18, 19].

3.2 Resource complexity implications

Although both circuit-based quantum computing (CBQC) and measurement-based quantum computing (MBQC) are universal and, in principle, polynomially equivalent in computational power [4, 20], since the workings of these two paradigms fundamentally differ, there are implications on how we perceive resources in each model, as summarized in Table 2. Note that the categorizations presented in the table represent the primary characteristics and more widely employed method in the existing literature rather than a strict or exclusive distinctions. In practice, and depending on the maturity and future direction of technologies, many techniques and optimization strategies can be adapted across both paradigms. For instance, graphical approaches such as ZX-calculus-based optimizations which was initially associated with MBQC have now also been applied in CBQC.

3.2.1 What constitutes primary resources?

In the circuit-based model, the primary resources are the number of physical qubits and the circuit depth, which is defined as the number of sequential layers of quantum gates required to implement a given algorithm. Each qubit is typically initialized at the start of the computation and remains active throughout, participating in various gate operations until the final measurement. Circuit depth is particularly important because it is directly related to the time the quantum system must maintain coherence; longer circuits are more susceptible to decoherence and noise, which degrade computational fidelity. In addition to qubit count and depth, other hardware-dependent factors such as gate fidelity, qubit connectivity, and error rates also play crucial roles in determining the overall resource cost and feasibility of running a quantum algorithm on a given device [21].

In contrast, MBQC introduces a fundamentally different perspective on resource allocation. Here, the key resource is the entangled resource state, most commonly a cluster state, which is prepared before computation begins. The size and geometry of this cluster state, often in the form of a two-dimensional lattice of qubits, determine the spatial resources required. Notably, all qubits in the resource state are prepared and entangled upfront, and computation is subsequently performed by a sequence of single-qubit measurements. The temporal resource in MBQC is the number of measurement layers, which, depending on the algorithm and the structure of the cluster state, can sometimes be reduced to a single round if sufficient spatial resources are available [4, 5]. This trade-off between spatial and temporal resources is a defining feature of MBQC: by increasing the number of qubits in the initial resource state, it is possible to parallelize measurements and thus reduce the effective computation time.

Another important distinction arises in the interpretation and management of entanglement. In CBQC, entanglement is generated dynamically as needed through the application of multi-qubit gates, and while it is a hallmark of quantum computation, it is not always present at every stage of the circuit. In MBQC, however, the entangled resource state is the very substrate upon which computation is enacted. The entire computational

Table 2 Summary of Resource Complexity and Related Aspects in CBQC and MBQC, compiled from [4, 14, 20, 21, 23, 25, 28, 29]

Aspect	CBQC	MBQC
Primary Resources	- Qubit count (active during computation) - Circuit depth (sequential gate layers) - Gate count	- Size of entangled resource state (cluster state) - Number of measurement layers - Measurement and classical overhead
Entanglement	Generated dynamically as needed by applying multi-qubit gates	Prepared upfront as the substrate for computation, Computation "consumes" entanglement
Temporal Resource	Circuit depth (number of sequential gate layers), Coherence time is a limiting factor	Number of sequential measurement layers Can be traded for spatial resources (more qubits)
Spatial Resource	Number of physical qubits required for the algorithm	Number of qubits in the cluster/resource state (often larger than logical qubit count)
Operations	Unitary gates applied in sequence, Clifford and non-Clifford gates realized directly	Computation via single-qubit measurement patterns Clifford gates by fixed patterns, non-Clifford by adaptive measurements
Measurement	Typically performed at the end of computation, Intermediate measurements less common	Central to computation Performed throughout, with adaptive bases and classical feedforward
Classical Processing	Minimal (mostly for final readout and error correction)	Essential for adaptive measurement and feedforward, can become a bottleneck
Overheads	- Gate synthesis - Error correction - Hardware connectivity constraints	- Resource state preparation - Measurement overhead - Classical-quantum interfacing - Qubit overhead
Optimization Focus	- Depth and qubit minimization - Gate reordering and cancellation - Hardware-aware compilation	- Cluster state size minimization - Measurement order and adaptivity - Cluster/qubit reuse - ZX-calculus techniques
Fault Tolerance	Surface/concatenated codes Magic state distillation for non-Clifford gates	Naturally suited to 3D cluster states Topological codes integrated into resource state
Scalability	Mature experimentally; Scaling limited by error rates, connectivity, and error correction overhead	Conceptually scalable Cluster states can be distributed for distributed QC

process can be viewed as a controlled "consumption" of entanglement, with each measurement irreversibly collapsing part of the resource state and propagating quantum information through the lattice [4].

Both models have additional overheads that must be considered in practical implementations. In CBQC, these include the overhead associated with gate synthesis (especially for non-native gates), error correction, and the need for high-fidelity operations. MBQC, on the other hand, incurs overhead in the preparation of large, high-fidelity cluster states, as well as the classical processing required for adaptive measurement and feedforward control. The latter is particularly significant, as the outcome of each measurement may determine the basis for subsequent measurements, necessitating fast and reliable classical-quantum interfacing [1, 5, 6].

In summary, while CBQC and MBQC are theoretically equivalent in computational power, their resource landscapes are shaped by the different ways in which they allocate and utilize qubits, entanglement, and time. CBQC emphasizes active qubit management and sequential gate application, with depth and coherence time as critical bottlenecks. MBQC, by contrast, frontloads resource investment into the preparation of a large entangled state, enabling potential gains in parallelism and fault tolerance, but at the cost of increased spatial and classical processing overheads.

3.2.2 Gates vs. measurement patterns

The operational paradigm of CBQC is centered on the direct application of unitary gates to quantum states. Each gate is a physical operation, and the sequence of gates defines the computation. Universal quantum computation is achieved by combining single-qubit gates (such as the Hadamard or phase gate) with a two-qubit entangling gate (such as CNOT). Clifford gates (Hadamard, CNOT, S) are relatively straightforward to implement and can often be executed with high fidelity. However, non-Clifford gates (such as the T gate) are more challenging, especially in the context of fault-tolerant quantum computing, where they typically require ancillary qubits and resource-intensive procedures like magic state distillation [22, 23].

In MBQC, the notion of a "gate" is replaced by the concept of a measurement pattern. Computation proceeds by measuring qubits in the cluster state in specific bases, with the choice of basis and the order of measurements encoding the algorithm. Clifford operations can be realized with fixed measurement patterns, while non-Clifford operations require adaptive measurements, where the basis for each measurement depends on the outcomes of previous measurements. This adaptivity introduces a layer of classical processing and feedforward, which is critical for universality but can also become a bottleneck if not efficiently managed [23, 24].

Fault-tolerant quantum computation (FTQC) further accentuates the differences between the two models. In CBQC, FTQC is typically implemented using surface codes or concatenated codes, with logical gates realized through code deformation or magic state injection. In MBQC, FTQC is naturally suited to three-dimensional cluster states, where logical qubits and gates are encoded in the topology of the resource state. This topological approach offers inherent protection against certain types of errors and can simplify the implementation of logical operations [1, 22].

3.2.3 Optimization strategies

Optimization strategies in CBQC are well-established and focus primarily on minimizing circuit depth and qubit count. Techniques such as gate commutation, cancellation, and advanced compilation are routinely employed to reduce resource usage and mitigate error accumulation. Hardware-aware optimization, which takes into account the connectivity and native gate set of the quantum processor, is also a key area of ongoing research [25].

In MBQC, optimization is a newer but rapidly developing field. Strategies include minimizing the size of the cluster state, reusing portions of the resource state (qubit recycling), and optimizing the order and adaptivity of measurements to reduce classical processing overhead. The ZX-calculus, a graphical language for quantum circuits, has emerged as a powerful tool for optimizing both CBQC and MBQC representations, enabling transformations that can bridge the two models and yield more resource-efficient implementations [26, 27]. While MBQC-specific optimization is less mature, recent work has shown that careful design of measurement patterns and resource states can lead to significant reductions in spatial and temporal overhead, especially for algorithms amenable to parallelization [6].

In conclusion, resource complexity in CBQC and MBQC is shaped by their respective operational paradigms. CBQC's resources are dominated by qubit count and circuit depth, with a strong emphasis on minimizing decoherence and error rates. MBQC, in contrast, requires up-front investment in large entangled states and efficient classical-quantum interfacing, but offers unique opportunities for parallelism, fault tolerance, and distributed computation. The choice of model (and the optimization of resources within each) depends on the specific requirements of the algorithm, the capabilities of the available hardware, and the predominant goals of the computation.

3.3 Fault tolerance and error correction

Fault tolerance (FT) and quantum error correction (QEC) are indispensable for scalable quantum computation, as physical qubits are inherently noisy and susceptible to decoherence and operational errors. Both CBQC and MBQC must address these challenges, but the strategies and natural fit for FT and QEC differ significantly between the models.

In the circuit-based paradigm, FT is typically achieved through the use of quantum error-correcting codes (QECCs), such as the surface code or concatenated code families [29].These codes encode logical qubits into multiple physical qubits, allowing for the detection and correction of errors via syndrome measurements and recovery operations. Logical gates are implemented either transversally (when possible) or via code deformation techniques. A particular challenge for CBQC is the implementation of non-Clifford gates, such as the T gate, which cannot be performed transversally in most codes. Instead, these gates are realized using resource-intensive procedures like magic state distillation, which introduces significant overhead in both qubit count and circuit depth [22, 23].

MBQC, on the other hand, is naturally suited to fault-tolerant computation due to its intrinsic reliance on entangled resource states. The most prominent approach to FT in MBQC is the use of three-dimensional (3D) cluster states, where logical information is encoded topologically [10]. In this framework, errors correspond to topological defects, and logical operations are implemented by braiding or manipulating these defects within the cluster state. The measurement pattern itself can be designed to incorporate error correction, making QEC an integral part of the computational process rather than a separate overhead. This topological approach offers high thresholds for error rates and can be more robust to certain types of noise, although it places demanding requirements on the preparation and stability of large, high-fidelity cluster states [30].

Despite these conceptual advantages, MBQC's practical realization of FT and QEC remains less mature than in CBQC, primarily due to the technical challenges associated with generating and maintaining large-scale cluster states with sufficient fidelity and connectivity. Nonetheless, the topological nature of MBQC's FT schemes continues to attract significant research interest, especially as hardware platforms capable of generating such states improve.

In addition, note that it is also worth mentioning on the suitability of both MBQC and CBQC models for the current Noisy Intermediate-Scale Quantum (NISQ) era, which is

Table 3 Fault Tolerance, Error Correction, and NISQ Suitability of CBQC vs. MBQC

Aspect	CBQC	MBQC
FTQC Realization	- Surface/concatenated codes - Logical gates via code deformation or magic state distillation	- 3D cluster states - Logical qubits encoded topologically - Operations via defect manipulation
QEC Integration	- Active correction cycles - Syndrome measurement and recovery	- Integrated into measurement pattern and resource state topology
Clifford/Non-Clifford	- Clifford: transversal or code deformation, - Non-Clifford: magic state distillation	- Clifford: fixed measurement patterns - Non-Clifford: adaptive measurement and feedforward
Practical Status	- Demonstrated in experiments, more mature	- Conceptually promising, under active development
NISQ Suitability	- Widely used in NISQ devices - Aligns with current hardware and shallow circuits - Optimization focuses on minimizing depth and gate count	- Conceptually offers parallelism and error mitigation via resource states, but limited by cluster state preparation and adaptive measurement challenges on current NISQ hardware

characterized by quantum devices with limited qubit counts and significant noise, as discussed below.

CBQC in the NISQ Era. Circuit-based quantum computing is currently the dominant model implemented on most NISQ devices, such as superconducting qubits and trapped ions. Its gate-based structure aligns well with available hardware control techniques, and many near-term quantum algorithms (e.g., Variational Quantum Eigensolver, Quantum Approximate Optimization Algorithm) have been developed specifically for CBQC on NISQ hardware. However, NISQ devices suffer from noise and decoherence that limit the circuit depth and size, restricting reliable execution to relatively shallow circuits with a small number of qubits [21].Optimization efforts in CBQC focus heavily on reducing circuit depth and gate count to mitigate noise impact, but the inherent sequential nature of gate application constrains parallelism and scalability on noisy hardware [31].

MBQC in the NISQ Era. Measurement-based quantum computing, while less mature experimentally, offers conceptual advantages that could be beneficial in the NISQ regime. Since MBQC performs computation via adaptive single-qubit measurements on a pre-prepared entangled resource state, it can, in principle, exploit parallelism by performing many measurements simultaneously, potentially reducing temporal overhead and exposure to decoherence [4, 13]. Moreover, the upfront preparation of cluster states may allow for error mitigation strategies tailored to the resource state itself. However, the preparation of high-fidelity cluster states at scale remains a significant technical challenge on current NISQ hardware. Additionally, the classical feedforward required for adaptive measurements introduces latency and complexity, which can be limiting given the short coherence times of NISQ devices [32, 33]. Table 3 summarizes the fault tolerance, error correction, and NISQ suitability of CBQC and MBQC.

3.4 Quantum hardware suitability

The suitability of quantum hardware for implementing CBQC or MBQC is determined by various factors, including the physical characteristics of the platform, the ease of realizing the required operations, and the alignment between the model's requirements and the hardware's native capabilities [32, 34].

Table 4 Quantum Hardware Suitability for CBQC and MBQC

Hardware Platform	Suitability for CBQC	Suitability for MBQC
Superconducting	High; mature and widely used	Possible, but less explored
Trapped Ions	High; mature	Possible, but less explored
Photonics	Moderate (circuit model)	Highly suitable, Cluster states and measurements easy
Neutral Atoms	Emerging	Potential for large-scale cluster state generation

CBQC is currently the dominant model for most leading quantum computing platforms, including superconducting qubits, trapped ions, and neutral atom arrays. These platforms naturally support the initialization, coherent manipulation, and measurement of individual qubits, as well as the application of a universal set of quantum gates. The circuit model's sequential, gate-based approach maps straightforwardly onto these systems, and extensive engineering advances have led to significant improvements in gate fidelity, connectivity, and error rates [21], resulting in its most advanced experimental progress compared to other platforms [35]. The suitability of several quantum hardware platforms based on the recent progress for implementing each computing model is summarized in Table 4.

MBQC, by contrast, is particularly well-suited to photonic quantum computing architectures. In photonics, it is often easier to generate large-scale entangled states (such as cluster states) using linear optics and probabilistic entangling operations, and single-qubit measurements are naturally implemented using beam splitters, phase shifters, and photodetectors [5, 33]. The measurement-driven nature of MBQC aligns well with the operational strengths of photonic systems, where maintaining long-term coherence is challenging but rapid, high-fidelity measurements are feasible. Neutral atom and trapped ion systems are also exploring MBQC, leveraging their ability to create and manipulate entangled states, though practical demonstrations remain limited compared to the circuit model [33].

The choice of hardware model is not always exclusive; hybrid approaches are being explored, where elements of both CBQC and MBQC are combined to exploit the strengths of each. For example, measurement-infused circuit models that incorporate MBQC into traditional circuit-based Variational Quantum Eigensolver (VQE) [8], as well as fusion-based photonic architectures, i.e., paradigm that uses finite-sized entangled states and destructive entangling measurements to achieve fault-tolerance [7], seek to leverage the resource efficiency and error resilience of MBQC within the broader framework of circuit-based control.

3.5 Scalability considerations

Scalability is the ultimate test for any quantum computing model, as practical quantum advantage requires systems that can handle large numbers of qubits and operations while maintaining high fidelity. The scalability prospects of CBQC and MBQC are shaped by various factors, including their respective resource requirements for experimental or large scale, error correction, as well as suitability with the distributed computing paradigm and modular architectures, as summarized in Table 5.

CBQC has demonstrated superior scalability in current experiments, driven by the maturity of hardware platforms and the extensive development of error correction and control techniques. Superconducting and trapped ion systems, for example, have scaled to

Table 5 Scalability Considerations in CBQC and MBQC

Scalability Aspect	CBQC	MBQC
Experimental Scale	Larger, more mature >100 qubits demonstrated	Limited, but conceptually scalable Large cluster states possible in principle
DQC Suitability	Challenging, requires entanglement distribution and synchronization	Natural fit Cluster states can be distributed, Computation performed via measurements
Modular Architectures	Emerging Requires complex routing and control	Well-suited Cluster states can be fused or connected modularly
Error Correction	Overhead increases rapidly with scale	Topological codes integrated into resource state Potentially more scalable

over 100 qubits [36], with ongoing improvements in gate fidelity and error correction protocols. In fact, recently, IBM announced its 1121-qubit processor titled Condor [37, 38]. However, further scaling is constrained by the overhead of quantum error correction, the need for high connectivity, and the challenge of maintaining coherence across large, densely connected systems [39, 40].

MBQC, while less advanced experimentally, offers inherent advantages for scalability, particularly in distributed and modular quantum computing. The cluster state resource model naturally lends itself to the preparation and manipulation of large entangled states, which can be distributed across multiple physical nodes. Computation can then proceed via local measurements and classical communication, enabling a form of distributed quantum computing (DQC) that is more challenging to realize in the circuit model. The spatial structure of cluster states also facilitates modular architectures, where independent resource states can be fused or connected to build larger computational systems. Moreover, MBQC's topological error correction schemes are well-suited to large-scale systems, offering robustness against local errors and noise [19, 41].

Despite these theoretical advantages, MBQC's scalability is currently limited by the technical difficulty of preparing large, high-fidelity cluster states and reliably performing adaptive measurements. As photonic and atomic platforms improve, however, MBQC is expected to become increasingly competitive, especially for applications that benefit from parallelism, distribution, or topological protection [6, 19, 41].

4 Drawing (and extending) the line: CBQC or MBQC-favorable?

Although both CBQC and MBQC are fundamentally universal models, they exhibit performance trade-offs, different operational requirements, and architectural compatibilities that may render each paradigm favorable under different circumstances. This section discusses the conditions under which one paradigm may be more advantageous than the other, based on prior analyses as well as the extension introduced in this work.

4.1 Prior analysis: a review of previous works

As MBQC gains more attention as an emerging quantum computing platform and is now explored further for actual and practical use, the question of which platform is best suited for running certain quantum algorithms has also started to be investigated. To our knowledge, one of the first attempts to systematically compare the performance of CBQC and MBQC was presented by Lee et al. [6] in 2022. Though their work focuses on MBQC for

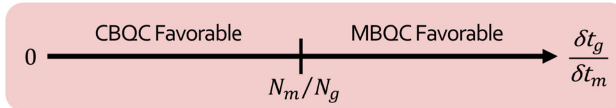

Figure 4 Illustration of MBQC/CBQC-favorable regimes determined by the hardware-dependent parameter $\delta t_g/\delta t_m$ as proposed in [6]

quantum simulation of fermionic systems, the authors first featured an initial analytical framework for determining the regime at which quantum algorithms might exhibit a performance advantage when executed in MBQC compared to in CBQC. In summary, the authors argued that MBQC may yield a runtime advantage compared to CBQC with their methods under the condition that a large number of qubits are available for use, the time taken for an accurate single-qubit measurement can be made small enough to avoid decoherence of the resource state, and the entangling gates are performed in parallel mostly at the beginning [6].

In detail, their analysis argues that the runtime of CBQC and MBQC can be estimated using a straightforward model as follows. Let M denote the number of sequential evolution steps (e.g., Trotter steps). For each step, N_g denotes the number of circuit-model gates required per step, while N_m denotes the number of single-qubit measurements performed in the corresponding MBQC measurement round. Introducing characteristic durations δt_g and δt_m for a circuit-model gate and an accurate single-qubit measurement, respectively, the total runtimes are approximated as in Equation (1).

$$T_{\mathrm{CBQC}} \sim M\,N_g\,\delta t_g, \qquad T_{\mathrm{MBQC}} \sim M\,N_m\,\delta t_m. \tag{1}$$

Then, MBQC becomes favorable in the cases as stated in Equation (2).

$$\frac{\delta t_g}{\delta t_m} > \frac{N_m}{N_g}. \tag{2}$$

In the concrete constructions considered by Lee et al. [6], which focus on local Hamiltonian simulation, a method known as the graph-state compactification [42] allows the ratio N_m/N_g to be fixed to unity. Consequently, the regime boundary reduces to a single hardware-dependent threshold $\delta t_g/\delta t_m = 1$, separating CBQC and MBQC-favorable regions. This situation is illustrated schematically in Fig. 4, which reproduces the regime boundary considered in their work.

This analysis was then featured and discussed by Qin et al. [32], who briefly expand the discussion to multi-qubit operations. However, their analysis was relatively limited to high-level architectural parameters and did not extend into detailed resource trade-offs or hardware-specific considerations.

4.2 Our proposal: generalized regime analysis

In this paper, we depart from the parameters firstly analyzed by Lee et al. [6], expanding their work to fit broader cases. In particular, we argue that their horizontal regime boundary is a consequence of the specific algorithmic structure and compilation strategy

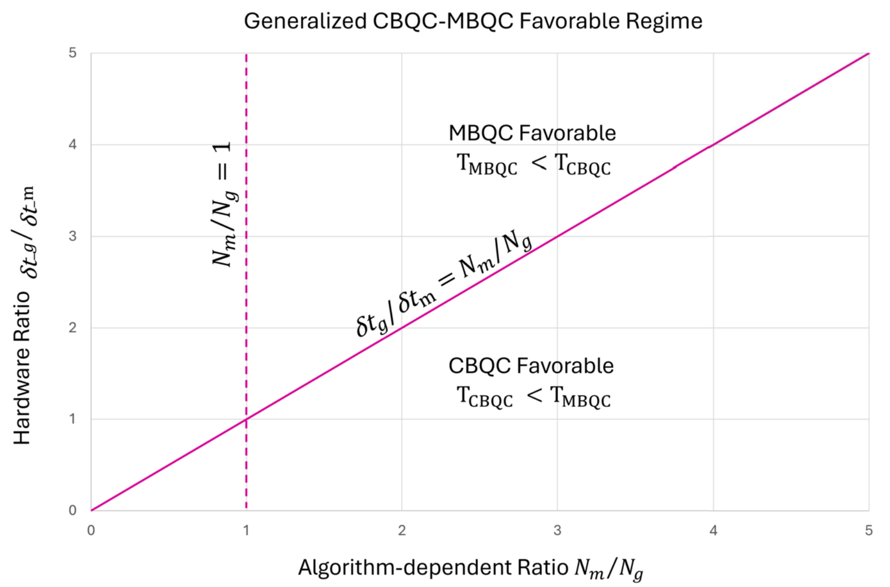

Figure 5 Generalized CBQC vs. MBQC-favorable regimes obtained by allowing the algorithm-dependent ratio N_m/N_g to vary. The diagonal boundary follows from equating the runtime proxies in Equation (1). The horizontal regime considered by Lee et al. [6] corresponds to the special case $N_m/N_g = 1$

employed in [6] rather than a universal property of MBQC. Thus, here, we first generalize the regime analysis by allowing the algorithm-dependent ratio N_m/N_g to vary, so as to capture a broader class of quantum algorithms and compilation strategies.

While the regime analysis of Lee et al. provides valuable insight, its applicability is restricted to algorithmic settings in which the ratio N_m/N_g can be fixed by construction. In more general quantum algorithms, including variational algorithms, quantum Fourier transform subroutines, and distributed or modular executions, the effective numbers of gate layers and measurement layers are strongly algorithm and compilation-dependent. In such settings, we argue that the ratio N_m/N_g should be treated as a variable rather than a constant. Retaining the runtime model in Equation (1), but allowing N_m/N_g to vary, the regime condition in Equation (2) defines a continuous boundary in the $(N_m/N_g,\ \delta t_g/\delta t_m)$ parameter space.

Taking these concerns into account, we propose a generalized regime map as shown schematically in Fig. 5. In this representation, the horizontal boundary covered by Lee et al. [6] (c.f. Fig. 4 corresponds to the special case $N_m/N_g = 1$, i.e., a horizontal slice of the more general diagram. In contrast, Fig. 5 explicitly captures how algorithm structure (through N_m/N_g) and hardware characteristics (through $\delta t_g/\delta t_m$) jointly determine whether CBQC or MBQC is favored. This generalized regime analysis forms the basis for the layered metrics and decision flow introduced in the following sections, where additional factors such as circuit depth, measurement depth, fault-tolerance overhead, and classical feedforward latency are incorporated.

Next, we also cover lower-level aspects for a more thorough analysis to better reflect the condition of running specific quantum algorithms on each quantum computing platforms. Specifically, in the succeeding sections, we identify and devise the key factors and performance metrics for both MBQC and CBQC in Sect. 5, then present a decision flow to help assist in model selection (i.e., CBQC or MBQC), with worked examples of two representative quantum algorithm subroutines in Sect. 6.2.

5 Identifying key factors and performance metrics: a layered approach

A comprehensive comparison between CBQC and MBQC would benefit from not only a qualitative assessment of their distinguishing features but also a systematic mapping of the key factors and performance metrics that determine their practical suitability for different quantum algorithms and hardware regimes. In this section, we adopt a layered approach, drawing inspiration from established frameworks such as those by Jones et al. [43] to categorize and analyze parameters across the quantum computing stack. In particular, for the stack category, we mainly refer to (and slightly modify) the quantum computing layers appeared in [44], which conceptually aligns with the layered architecture presented in Fig. 1 of Gheorghiu and Mosca's quantum resource estimation paper [45]. In this section, we present our analysis of key factors and performance metrics of both computing models, categorizing them as a layered approach, from top to bottom: Algorithms, Logical, Fault Tolerance, and Hardware, for clarity and a more systematic view.

5.1 Layered analysis of key factors and performance metrics

The challenge of optimizing quantum computation, whether in the circuit-based (CBQC) or measurement-based (MBQC) paradigm, benefits from a structured analysis as a layered approach. This methodology enables independent reasoning and targeted optimization at each level, from abstract algorithm design to the physical realities of hardware implementation. By decomposing the quantum stack into algorithmic, logical, fault-tolerant, and hardware layers, we can clarify how resource requirements and performance bottlenecks propagate, interact, and ultimately determine the feasibility and efficiency of quantum computation. The resulting categorization of key factors and performance metrics is summarized in Table 7.

5.1.1 Algorithm layer

At the algorithmic layer, the focus is on the high-level structure of quantum algorithms and their intrinsic computational demands, independent of any specific compilation or execution model. This is the stage at which quantum parallelism, circuit width, and depth sensitivity are first determined. For example, algorithms such as VQE, QAOA, and quantum simulation kernels define not only the logical sequence of operations but also the maximum number of qubits that must be simultaneously manipulated, i.e., the circuit width prior to compilation. This metric sets a lower bound on the hardware resources required for implementation and directly constrains the degree of parallelism that can be exploited [46].

At this layer, algorithms may also be intrinsically deterministic or probabilistic in their output structure. For instance, Shor's algorithm and phase estimation succeed with constant probability per run and rely on repetition and classical post-processing as part of their algorithmic definition. Accordingly, "success probability" at the algorithmic layer refers to intrinsic algorithmic success defined by the computational task itself, rather than to execution-level failure mechanisms that arise from, e.g., noise, postselection, or non-deterministic operations, which are addressed at lower layers.

Furthermore, depth sensitivity (here, as similarly discussed in this context in [47, 48], refer to the extent in which an algorithm's correctness, fidelity, or practical feasibility depends on the length of its sequential execution depth rather than circuit width or gate count) and parallelizability further shape an algorithm's suitability for near-term quantum computing (NISQ), in which shallow circuits and high parallelism are advantageous. In addition, some algorithms permit structural transformations that significantly alter resource scaling under specific execution models. For example, certain Boolean functions admit constant-depth implementations only when adaptive strategies are allowed, whereas non-adaptive realizations may incur exponential overheads in space or repetition [6]. In this sense, the algorithmic layer provides the conceptual blueprint that constrains all subsequent compilation and execution choices.

5.1.2 Logical layer

The logical layer translates the abstract algorithm into an explicit quantum circuit or measurement pattern, assuming idealized, error-free conditions. At this stage, compilation choices determine the logical gate set (e.g., Clifford+T, CZ+H, etc.), as well as the ratio of Clifford to non-Clifford operations, and the associated gate-synthesis costs [25]. Logical gate count, circuit depth, qubit count, and ancilla requirements is usually considered as essential performance metrics.

In CBQC, this layer captures how algorithmic structure is mapped onto sequences of quantum gates, including the impact of connectivity constraints and the resulting SWAP overhead. A wider circuit allows for more parallel gate execution, reducing depth but potentially increasing the number of SWAP gates if hardware connectivity is limited [49]. In MBQC, the logical layer instead specifies the cluster state's geometry and the sequence of measurements required to realize the desired computation. Properties such as flow existence (e.g., gflow, Pauli flow, or related variants), measurement adaptivity, and by-product-operator handling can be considered as the key structural factors that determine whether the computation proceeds deterministically or requires postselection. The complexity of the measurement pattern, the number of adaptive measurement layers, and the classical feedforward latency become critical performance metrics [20]. Notably, MBQC can sometimes achieve significant reductions in logical depth by leveraging parallel measurements, but this often comes at the cost of increased initial resource state size and classical processing overhead. Additionally, a larger cluster state enables greater parallelism in measurements, but the preparation of such states can be resource-intensive, and adaptivity may introduce additional classical control complexity [4, 5].

Note that at this layer, execution success is better understood as a qualitative property of the logical implementation (e.g., deterministic versus postselected execution or the presence of probabilistic primitives such as heralded operations) as opposed to a scalar performance metric, as it fundamentally alters how logical depth and resource counts translate into execution cost. Quantitative consequences of such nondeterminism, such as expected repetition factors or effective depth inflation due to repetition, are therefore treated as logical-layer performance metrics rather than algorithmic ones.

5.1.3 Fault tolerance layer

The fault tolerance layer addresses the encoding of logical circuits into error-corrected, fault-tolerant primitives, which is essential for scalable quantum computation. At this level, logical circuits or operations is mapped onto encoded qubits using quantum error correction codes, such as surface codes or color codes. The choice of code family, code distance, and syndrome-extraction cadence [50] determines the overhead and efficacy of error correction [29]. Also, the temporal structure of fault-tolerant operations, including the degree to which error-correction cycles and logical gates can be executed in parallel, further influences overall performance.

In CBQC, non-Clifford gates (such as T gates) typically require magic-state distillation, which can dominate the resource consumption in large-scale computations [51]. The logical-to-physical qubit ratio, logical error rate, and distillation cost are key metrics that reflect the efficiency of the fault-tolerant encoding.

Circuit width plays a dual role at this layer: increasing width raises the physical-qubit overhead required for encoding, while also enabling greater parallelism that can reduce logical execution time and mitigate decoherence, provided the error-correction scheme supports sufficient throughput.

Meanwhile, in MBQC, recent advances have demonstrated that certain cluster state constructions, e.g., color-code-based cluster states, can implement universal, fault-tolerant computation with potentially lower overhead for some logical gates, although state distillation remains a major challenge for arbitrary gates [52]. Distributed and networked MBQC architectures further complicate the picture, as they must contend with both circuit-level and network-induced noise, but can benefit from modularity and parallelism [53].

5.1.4 Hardware layer

At the hardware layer, the logical, error-corrected circuit is mapped onto a specific physical quantum hardware platform, subject to the constraints on e.g., qubit count, connectivity, gate fidelities, measurement fidelity, and coherence times. The native gate set, qubit topology (e.g., nearest-neighbor, all-to-all), and measurement fidelity all play decisive roles in determining the achievable circuit width and depth [21]. For instance, a hardware with limited connectivity may require additional SWAP gates to implement multi-qubit operations, increasing both gate count and exposure to noise.

In MBQC, the hardware layer must also support the efficient generation of large, high-fidelity cluster states and the rapid, programmable measurement of individual qubits. Time-domain multiplexing and photonic architectures have emerged as promising approaches for scaling MBQC, enabling the preparation of large cluster states and high-throughput measurements [54]. However, the preparation and stabilization of such states remain significant technical challenges, particularly as the system scales.

The interplay between circuit width, gate count, and decoherence is most acute at this layer. A circuit with width matched to the available hardware can exploit maximum parallelism, reducing execution time and decoherence risk. Conversely, if the circuit width exceeds hardware capacity, operations must be serialized, increasing depth and the likelihood of errors. Aggregate metrics such as quantum volume and system throughput have been developed to capture this balance, providing holistic benchmarks for hardware performance [55].

6 Decision flow for CBQC-MBQC model selection

6.1 Decision flow

Choosing between the circuit-based quantum computing (CBQC) model and measurement-based quantum computing (MBQC) for a given algorithm or workload depends on interacting constraints across the algorithmic, logical, fault-tolerance, and hardware layers identified in Sect. 5.1. We outline a decision flow as a structured entry point for model selection (i.e., to determine which model is better suited in a given scenario) instead of a universal rule-set, since trade-offs are often platform and compiler-dependent. Below, we highlight the major considerations:

1. **NISQ Regime Checks**
    - *Noise Robustness and Error Mitigation:* If MBQC is more robust to noise and supports more effective error mitigation than CBQC, select MBQC. Conversely, if CBQC is superior in these respects, select CBQC.
    - *Transpilation and Compilation Efficiency:* If CBQC allows for significantly more effective noise-adaptive transpilation and circuit optimization, select CBQC.
    - *Hybrid Quantum-Classical Integration:* If MBQC supports more efficient hybrid integration (e.g., rapid measurement-feedback cycles), select MBQC.
    - *Device Variability and Qubit Mapping:* If device variability is high and CBQC offers greater flexibility in mapping to reliable qubits, select CBQC.
2. **FTQC Regime Checks**
    - *Error Correction Overhead:* If CBQC has significantly lower error correction overhead, select CBQC. If MBQC is superior in this regard (e.g., via topological codes), select MBQC.
3. **Universal Checks (Both Regimes)**
    - *Circuit Depth vs. Coherence and Latency:* If the estimated CBQC circuit runtime (circuit depth $\times$ two-qubit gate time) exceeds a critical fraction of the hardware's coherence time, and MBQC's measurement plus feedforward time is lower, select MBQC.
    - *Variational/Adaptive Structure and Measurement Adaptivity:* If the algorithm is variational or requires adaptive measurements, select MBQC.
    - *Clifford-Dominated and Highly Parallel Circuits:* If the circuit is Clifford-dominated and highly parallel, select MBQC.
    - *Runtime and Qubit-Width Trade-Off:* If MBQC runtime is less than CBQC runtime and MBQC does not require more qubits, select MBQC.
    - *Hardware Connectivity and MBQC Gadget/Resource Support:* If hardware connectivity is not all-to-all and MBQC-specific gadgets or resource states are available, select MBQC.
    - *Resource State Preparation Efficiency:* If MBQC resource state preparation is inefficient, select CBQC.
    - *Feedforward and Classical Control Latency:* If classical feedforward or control latency is high, select CBQC.
4. **Fallback:** If no clear advantage is found for MBQC, default to CBQC, reflecting its maturity and broader hardware support.

Rationale and Literature Context. In the NISQ regime, the fragility of quantum information and the absence of full error correction demand a focus on noise robustness, error mitigation, and the practicalities of hybrid quantum-classical integration. In the FTQC

Table 6 Ranked key factors for CBQC vs. MBQC model selection, with layer and regime applicability

Rank	Factor	Layer	Regime	Justification
1	Circuit depth vs. coherence and latency	Algorithm /Hardware	Both	- Ensures computation completes before decoherence dominates - MBQC can leverage parallelism and fast measurements if CBQC depth is too high
2	Noise sensitivity, error rates, and error mitigation capability	Hardware /Logical	NISQ	- NISQ devices are dominated by noise - Models robust to noise and supporting error mitigation are favored
3	Parallelism and measurement latency	Algorithm /Hardware	Both	BQC benefits from high parallelism and fast measurement/feedforward, reducing runtime and logical depth.
4	Device variability and qubit mapping	Hardware	NISQ	Ability to exploit the most reliable qubits and adapt to device-specific noise profiles is critical in NISQ.
5	Clifford/non-Clifford ratio	Algorithm /Logical	Both	- MBQC is efficient for Clifford-dominated circuits - Non-Clifford gates add overhead, especially for CBQC.
6	Connectivity and MBQC gadget/resource support	Hardware	Both	MBQC can mitigate limited connectivity if cluster state gadgets are available, reducing SWAP overhead.
7	Variational/adaptive structure and measurement adaptivity	Algorithm	Both	MBQC naturally supports variational and adaptive algorithms, often enabling more efficient implementations.
8	Ancilla and resource state overhead	Logical /Hardware	Both	High ancilla or resource state overhead may favor one model depending on hardware capabilities and preparation efficiency.
9	Feedforward and classical control latency	Logical /Hardware	Both	High classical latency can bottleneck MBQC, negating its advantages in adaptivity and parallelism.
10	Error correction overhead	Fault Tolerance	FTQC	Lower error correction overhead in one model is decisive for scalable, fault-tolerant computation.
11	Transpilation and compilation efficiency	Logical /Hardware	NISQ	Aggressive, noise-adaptive compilation can significantly improve NISQ performance.
12	Hybrid quantum-classical integration	Algorithm /Logical	NISQ	Efficient support for hybrid algorithms and rapid feedback cycles is key in NISQ.

regime, the resource scaling of error correction becomes the decisive constraint, with the model offering lower overhead emerging as the natural choice for scalable, reliable computation. Universal factors (e.g., circuit depth, parallelism, hardware connectivity), remain relevant, shaping the practical feasibility of any quantum algorithm regardless of regime.

Ranked Key Factors and Performance Metrics. Moreover, considering the level of importance, the layered key factors and performance metrics in the previous subsection is ranked as presented in Table 6.

6.2 Use-case examples: applying the decision flow

While the layered metrics in Table 7 and the decision flow in Sect. 6 provide a systematic framework for comparing CBQC and MBQC, their practical value ultimately depends on whether they can be instantiated for concrete algorithmic workloads under realistic hardware constraints. In this subsection, we therefore present two use-case examples: the Quantum Fourier Transform (QFT) and Variational Quantum Eigensolver (VQE), that connect algorithm structure, logical compilation characteristics, and hardware-dependent execution times. These use cases are deemed representative due to their applicability and frequent use for evaluating proposed methods in various research. Notably, QFT (and its usual counterpart such as adder or the modular exponentiation circuit for Shor's algorithm) is a canonical subroutine for various methods employing quantum arithmetic-based techniques whereas VQE is prominently used for evaluating quantum machine learning (QML)-based proposals.

Note that these examples are intentionally analytical and parameterized to illustrate how the dominant resource constraints identified in the layered analysis manifest in practice, and how they guide the selection between CBQC and MBQC instead of for device-level benchmarks or performance claims for specific platforms. In addition, throughout the scenarios, as a proof-of-concept, we keep it high-level by taking into account only the relatively dominant parameters (i.e., the ones with leading effects) while leaving out finer-grained details (e.g., detailed scheduling, calibration drift, and full fault-tolerance cycle overheads).

Here, we first describe the assumptions that we employ for the use-cases. For both examples, we employ simplified execution-time estimates that we believe can reflect the dominant sequential contributions to the runtime of a quantum algorithm, shown in Equation (3):

$$T_{\mathrm{CBQC}} \sim N_g\, t_{2q}, \qquad T_{\mathrm{MBQC}} \sim N_m\,(t_m + t_{\mathrm{ff}}), \tag{3}$$

where N_g denotes the effective number of sequential two-qubit gate layers after compilation in the circuit model, N_m denotes the effective number of sequential measurement layers imposed by the measurement pattern and adaptivity in MBQC, t_{2q} is the characteristic two-qubit gate time, t_m is the single-qubit measurement time, and t_{ff} captures classical feedforward latency when adaptive measurements are required.

Equation (3) assumes deterministic logical execution in both models. When postselection or probabilistic primitives are present, the expected repetition factor identified at the logical layer in Table 7 then should be incorporated multiplicatively into the total execution time. Note that the distinction is essential, but does not alter the qualitative conclusions of the examples considered below.

Additionally, for the purpose of the following use-case illustrations, we adopt representative hardware parameters that aligns with current experimental results and directly enter the simplified execution-time estimates in Equation (3).

For superconducting circuit-based platforms, characteristic two-qubit gate times t_{2q} are typically reported in the range of a few hundred nanoseconds. For example, Krinner et al. [56] report surface-code cycle times of approximately 1.1 µs that include multiple two-qubit operations and readout, while recent large-scale superconducting demonstrations on Google report individual two-qubit gate durations on the order of 10^2 ns. We therefore take $t_{2q} \sim 200 - 400$ ns as a conservative representative range [56, 57]. Single-qubit

Table 7 Identified Key Factors and Performance Metrics of CBQC and MBQC across Quantum Computing Layers

Layer	Description	Key Factors	Performance Metrics
Algorithm	• High-level quantum-algorithm design; defines computational workload and parallelism. Representative families: • VQE / QAOA • Phase-estimation QFT-based (e.g., Shor's, quantum chemistry) • Quantum-simulation kernels (Hubbard, Fermi–Hubbard, Kitaev chain) • Search/optimization oracle (Grover, max-cut)	• Intrinsically deterministic vs probabilistic output structure • Depth sensitivity • Parallelizability • Ancilla requirement	• Intrinsic algorithmic success probability • Required repetition for constant success
Logical	Architecture-agnostic quantum circuit translation of a quantum algorithm, assuming ideal condition without error correction or mitigation.	• Clifford vs Non-Clifford ratio • Logical gate set (Clifford+T, CZ+H, …) • Gate-synthesis cost • By-product-operator handling • Deterministic vs postselected execution • Requirement for adaptivity (feedforward) • Probabilistic primitives (e.g., heralded operations) • Flow existence (Gflow, Pauli flow, Shadow Pauli flow) (MBQC) • Measurement adaptivity	• Logical circuit depth • Logical qubit count • Logical gate count • Total gate count • Ancilla count • Connectivity overhead (SWAP count) (CBQC) • T-count, T-depth (CBQC) • Measurement-layer count (MBQC) • Feed-forward latency • Classical feed-forward cycles (MBQC) • Execution-induced repetition factor • Effective depth inflation due to repetition
Fault Tolerance	Encoding of logical operations into fault-tolerant primitives: • Full QEC (FTQC era) • Error mitigation or none (NISQ era)	• Code family (surface, color, LDPC, Raussendorf lattice) • Code distance • Syndrome-extraction cadence (rate) • Magic-state / Pauli-gadget factories • Lattice surgery vs braiding • Temporal structure of FT operations (serial vs parallel)	• Logical-to-physical qubit ratio • Logical error rate • FT-gate overhead (space/time) • Distillation cost • Cycle time per error-correction round
Hardware	Mapping of error-corrected /mitigated circuits onto actual hardware platforms, e.g.: • Superconducting • Ion-trap • Photonic • Neutral-atom	• Native gate set fidelities • Qubit topology/connectivity (nearest-neighbour, fixed-degree, all-to-all) • Coherence time • Measurement fidelity • Measurement latency • Control-loop bandwidth • Distributed-entanglement links*	• Physical qubit count • Physical gate error rate • Effective coherence budget vs execution time • Measurement latency • Parallel-gate bandwidth • Inter-module entanglement rate*

measurement times on such platforms are typically in the sub-microsecond to microsecond regime. Here, we adopt $t_m \sim 0.5 - 1.5$ µs, consistent with reported dispersive readout latencies in contemporary superconducting devices [56, 57]. Coherence times T_1, T_2

for superconducting qubits are commonly reported in the range of several tens to a few hundred microseconds, thus the representative interval $T_{1,2} \sim 50 - 200$ µs used in the discussion [56, 57]. In addition, the architectures considered are assumed to have planar, nearest-neighbor connectivity, as is standard for surface-code-oriented superconducting layouts.

For photonic measurement-based platforms, the relevant temporal parameters are single-qubit measurement time t_m and classical feedforward latency t_{ff}, rather than intrinsic qubit coherence. Large-scale continuous-variable cluster-state experiments report high-rate homodyne measurements with effective per-mode measurement times in the tens of nanoseconds regime. Thus, we take $t_m \sim 20 - 50$ ns as representative [58, 59]. Also, classical feedforward latencies reported in photonic MBQC experiments and architectures range from tens of nanoseconds to the microsecond scale depending on electronics and system integration. Accordingly, we adopt a broad indicative range $t_{\mathrm{ff}} \sim 10 - 10^3$ ns [58, 60]. While photonic platforms are not limited by intrinsic qubit coherence times in the same sense as solid-state systems, their performance is instead constrained by optical loss, detector efficiency, and stability of resource-state generation [58]. Finally, effective all-to-all logical connectivity is assumed via the structure of the cluster state, following standard MBQC constructions [61].

6.2.1 Example 1: *quantum Fourier transform* (*QFT*)

We first consider an n-qubit quantum Fourier transform (QFT), representative of phase-estimation-type subroutines that appear in algorithms such as Shor's algorithm and quantum simulation. QFT's characteristics are also beneficial to show the relation between gate count and achievable parallelism across different computational models.

At the algorithmic and logical layers, an ideal QFT contains $\Theta(n^2)$ two-qubit controlled-phase gates. In the circuit model, these gates must be arranged into sequences with an effective depth depends strongly on hardware connectivity and compilation strategy. For instance, an ideal, all-to-all connectivity that considers SWAP operations as free will have a much less resource requirements compared to a more realistic scenarios in which qubit placement is fixed, such as in a CBQC, superconducting qubit-based circuit.

CBQC realization. On superconducting architectures with planar, nearest-neighbor connectivity, routing and scheduling constraints introduce additional SWAP operations that inflate the effective sequential depth. We therefore write

$$N_g \approx \alpha\, n^2, \tag{4}$$

where $\alpha \geq 1$ signifies compilation and connectivity-induced depth inflation. Using Equation (3), the dominant execution-time contribution becomes

$$T_{\mathrm{CBQC}} \approx N_g\, t_{2q}. \tag{5}$$

A necessary feasibility condition is that T_{CBQC} remains well below a platform-dependent fraction of the coherence time T_2. As n increases or connectivity constraints worsen, N_g grows rapidly, and circuit depth becomes the dominant bottleneck.

MBQC realization. In MBQC, the same logical transformation is implemented via a measurement pattern on a pre-prepared entangled resource state. Crucially, many of the logical dependencies in QFT can be resolved through parallel measurements, with the remaining sequential structure determined by the measurement flow. As a result, the number of sequential measurement layers scales approximately linearly as:

$$N_m \approx \beta\, n, \tag{6}$$

where β depends on the chosen decomposition and flow structure. The dominant temporal contribution is then

$$T_{\mathrm{MBQC}} \approx N_m\,(t_m + t_{\mathrm{ff}}). \tag{7}$$

On photonic MBQC platforms employing time-domain multiplexed cluster states, measurement and feedforward operations can be performed at high rates, . . . shifting the dominant constraint from intrinsic qubit coherence to single-qubit measurement time, classical feedforward latency, and resource-state preparation overhead.

Implication for model selection. This example directly illustrates the highest-ranked factor in the decision flow: when routing-induced depth inflation causes T_{CBQC} to approach the coherence budget, while MBQC sustains sufficiently low measurement and feedforward latency with feasible resource-state preparation, MBQC becomes the more favorable execution model. Conversely, if measurement latency or resource-state preparation dominates, CBQC remains preferable despite its higher two-qubit gate count.

6.2.2 Example 2: *variational quantum eigensolver* (*VQE*)

As a contrasting case, we consider the variational quantum eigensolver (VQE), a prototypical NISQ workload characterized by shallow parameterized circuits executed repeatedly to estimate expectation values. The objective function takes the form

$$E(\boldsymbol{\theta}) = \sum_{j=1}^{M} c_j \langle P_j \rangle_{\boldsymbol{\theta}}, \tag{8}$$

where $\{P_j\}$ are Pauli operators and statistical precision requires repeated circuit executions.

Runtime structure. Unlike QFT, VQE as a parameterized circuit is primarily limited by repetition count instead of single-run coherent depth. The dominant total execution time scales as

$$T_{\mathrm{VQE}} \approx N_{\mathrm{exec}}\, t_{\mathrm{cycle}}, \tag{9}$$

where N_{exec} is the total number of circuit executions required across all Pauli terms and optimization steps, and t_{cycle} includes state preparation, measurement, and classical processing.

CBQC vs. MBQC interpretation. In CBQC, ansatz depth can typically be tuned to remain below coherence limits, making measurement throughput and device stability across many repetitions the dominant constraints. In MBQC, VQE corresponds to repeatedly preparing suitable fragments of a resource state and executing high-throughput measurement patterns. As a result, MBQC is favorable only if resource-state preparation and measurement pipelines can sustain the large number of independent circuit executions required to estimate Pauli expectation values to the desired statistical accuracy. Otherwise, the overhead of repeated state generation outweighs the benefits of measurement parallelism.

Implication for model selection. VQE thus occupies a regime in which the decision flow is governed primarily by measurement latency, classical control bandwidth, and hybrid quantum–classical integration, rather than by coherent circuit depth. In such settings, neither CBQC nor MBQC is universally superior. Instead, the more suitable model is determined by which platform can sustain higher throughput across repeated executions under realistic stability constraints.

7 Discussions

The comparative framework and decision flow presented above provide a principled starting point for selecting between circuit-based (CBQC) and measurement-based (MBQC) quantum computing paradigms. However, there are several notes to consider to place these results in proper context.

7.1 On the decision flow and metrics

First, it is important to emphasize that the proposed decision flow and the ranking of key factors are not exhaustive nor immutable. They are constructed from the current state of the art in quantum hardware, algorithmic techniques, and benchmarking methodologies. As the field rapidly evolves, new hardware architectures, error mitigation strategies, or algorithmic breakthroughs may shift the relative importance of these factors. For instance, advances in cluster state generation or classical feedforward control could tip the practical balance in favor of MBQC for a broader class of algorithms and devices.

In addition, the metrics and thresholds employed, e.g., circuit depth relative to coherence time, or the practical overheads of error correction, should be viewed as starting point or guidelines rather than rigid criteria. In real-world scenarios, the optimal choice may depend on implementation constraints or trade-offs, including hardware-specific characteristics (e.g., qubit connectivity, calibration stability), the maturity of software toolchains, and the specific performance targets of the application (e.g., fidelity, runtime, resource constraints).

7.2 Potential hybrid CBQC-MBQC models

Historically, circuit-based quantum computing (CBQC) and measurement-based quantum computing (MBQC) have been treated as fundamentally distinct paradigms, leading researchers to concentrate primarily on one model at a time. CBQC, benefiting from a longer history of theoretical development and the availability of algorithmic primitives such as gate decomposition and quantum error-correcting codes, has attracted most of the community's attention and resources. In contrast, MBQC remained largely at the proof-of-concept and small-scale experimental stage, owing to the less mature technical aspect and significant overhead associated with the preparation and stabilization of large cluster states.

Recent advances, however, suggest that CBQC and MBQC may be bridged by hybrid architectures that exploit the strengths of both approaches. One example is the incorporation of measurement-based subroutines into otherwise gate-based circuits. Chan et al. [8] demonstrated how adaptively measured cluster fragments can implement non-Clifford rotations within a variational quantum eigensolver, reducing gate depth and improving noise resilience on near-term devices [8]. Such *measurement-infused circuits* leverage offline generation of entanglement resources, followed by online adaptive measurements, to implement complex rotations without long sequences of T-gates and distillation.

Conversely, one may envisage *circuit-infused* MBQC protocols, where small, deterministic gate blocks—compiled via standard CBQC techniques—are interleaved with measurement layers on a cluster state. Nonetheless, to our knowledge, this approach has not been clearly discussed in literature, while still may be an interesting future research direction.

Note that another emerging approach, called the fusion-based quantum computing (FBQC), has also been proposed. In detail, it combines finite-sized resource states with destructive entangling measurements to achieve fault tolerance without relying exclusively on either CBQC or MBQC [7]. While FBQC is not strictly a CBQC–MBQC integration, its operational principles(i.e., namely, generation of small graph-state fragments and subsequent fusion operations) share conceptual similarities with hybrid measurement-infused circuits.

In the future, not only each model's suitability for certain quantum algorithms need to be further explored. A hybrid approach to leverage the strengths of both model may be instrumental in guiding experimental implementations on superconducting, trapped-ion, and photonic platforms.

7.3 Limitations and future directions

Several limitations of the present analysis should be acknowledged. The decision flow, while comprehensive, is based on a finite set of metrics and does not capture all possible hardware or algorithmic idiosyncrasies. For instance, the practical realization of large-scale cluster states for MBQC remains a significant technical challenge, and the classical feedforward required for adaptive measurements can introduce nontrivial latency. Conversely, the overheads of error correction and magic state distillation in CBQC may become prohibitive as systems scale, motivating further research into alternative fault-tolerant schemes.

The continued co-evolution of hardware, algorithms, and error mitigation techniques may even strengthen each model's applicability, or on the other hand, may blur the boundaries between CBQC and MBQC. The development of modular, distributed, and hybrid quantum architectures, along with advances in quantum networking and photonic integration, may open new opportunities for MBQC-inspired computation even in platforms traditionally dominated by the circuit model.

8 Conclusions

In this study, we have presented a systematic comparison between circuit-based quantum computing (CBQC) and measurement-based quantum computing (MBQC), extending prior works to assess each paradigm's suitability for executing quantum algorithms under a range of practical scenarios. We first revisited the theoretical foundations and historical development of both models, then examined the features that distinguish one from the other. Moving beyond prior analyses, we introduced a generalized regime framework to better capture broader scenarios between algorithm structure and hardware-dependent parameters, moving beyond fixed algorithmic assumptions. Within this framework, we identified the key parameters that help characterize an algorithm's favorability for CBQC or MBQC, organizing these criteria and performance metrics into hierarchical layers of the quantum computing stack. Leveraging these layered metrics, we then proposed a decision-flow framework as a guideline for selecting the most appropriate computational model in both the NISQ and fully fault-tolerant (FTQC) era along with two representative worked examples of QFT and VQE. Finally, we discussed the potential for a hybrid CBQC–MBQC approaches, highlighting how combining gate-based and measurement-based elements may enable more flexible and resource-efficient execution strategies for practical quantum computing applications.

Acknowledgements
The authors would like to thank the anonymous reviewers for their feedback, which significantly improves the quality of the manuscript. The first author would also like to thank Muhammad Firdaus and SAH for the support.

Author contributions
Conceptualization, Investigation: HTL, BSC. Writing, Formal analysis: HTL, BSC. Supervision, Funding acquisition: BSC.

Funding information
This research was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government Ministry of Science and ICT (MSIT) under Grant 2020K1A3A1A78087782, in part by the Pukyong National University Industry-University Cooperation Research Fund in 2023 (202312370001), and in part by the Pukyong National University, South Korea, Industry-University Cooperation Foundation's 2025 Post-Doc Support Project.

Data availability
No datasets were generated or analysed during the current study.

Materials availability
Not applicable

Code availability
Not applicable

Declarations

Ethics approval and consent to participate
Not applicable

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Scientific Computing, Pukyong National University, Busan, 48513, Republic of Korea. [2]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, 40116, West Java, Indonesia.

Received: 1 August 2025 Accepted: 16 February 2026 Published online: 27 February 2026

References

1. Nielsen MA, Chuang IL. Quantum computation and quantum information: 10th anniversary edition. 10th ed. Cambridge University Press; 2011.
2. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. 1994. p. 124–34.
3. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on theory of computing. 1996. p. 212–9.
4. Raussendorf R, Briegel HJ. A one-way quantum computer. Phys Rev Lett. 2001;86(22):5188.
5. Briegel HJ, Browne DE, Dür W, Raussendorf R, Nest M. Measurement-based quantum computation. Nat Phys. 2009;5(1):19–26.
6. Lee W-R, Qin Z, Raussendorf R, Sela E, Scarola V. Measurement-based time evolution for quantum simulation of fermionic systems. Phys Rev Res. 2022;4(3):032013.
7. Bartolucci S, Birchall P, Bombin H, Cable H, Dawson C, Gimeno-Segovia M, Johnston E, Kieling K, Nickerson N, Pant M, et al. Fusion-based quantum computation. Nat Commun. 2023;14(1):912.
8. Chan A, Shi Z, Dellantonio L, Dür W, Muschik CA. Measurement-based infused circuits for variational quantum eigensolvers. Phys Rev Lett. 2024;132(24):240601.
9. Deutsch D. Quantum theory, the Church–Turing principle and the universal quantum computer. Proc R Soc Lond Ser A, Math Phys Sci. 1985;400(1818):97–117.
10. Raussendorf R, Harrington J, Goyal K. A fault-tolerant one-way quantum computer. Ann Phys. 2006;321(9):2242–70.
11. Danos V, d'Hondt E, Kashefi E, Panangaden P. Distributed measurement-based quantum computation. Electron Notes Theor Comput Sci. 2007;170:73–94.

12. Vijayan MK, Paler A, Gavriel J, Myers CR, Rohde PP, Devitt SJ. Compilation of algorithm-specific graph states for quantum circuits. Quantum Sci Technol. 2024;9(2):025005.
13. Briegel HJ, Browne DE, Dur W, Raussendorf R, Nest M. Measurement-based quantum computation. Nat Phys. 2009;5(1):19–26. https://doi.org/10.1038/nphys1157.
14. Jozsa R. An introduction to measurement based quantum computation. In: Quantum information processing-from theory to experiment. NATO science series, III: computer and systems sciences. vol. 199. 2006. p. 137–58.
15. Alberts GJ, Rol MA, Last T, Broer BW, Bultink CC, Rijlaarsdam MS, Van Hauwermeiren AE. Accelerating quantum computer developments. EPJ Quantum Technol. 2021;8(1):18.
16. Benjamin SC, Lovett BW, Smith JM. Prospects for measurement-based quantum computing with solid state spins. Laser Photonics Rev. 2009;3(6):556–74.
17. DeCross M, Chertkov E, Kohagen M, Foss-Feig M. Qubit-reuse compilation with mid-circuit measurement and reset. Phys Rev X. 2023;13(4):041057.
18. Kissinger A, Wetering J. Universal mbqc with generalised parity-phase interactions and Pauli measurements. Quantum. 2019;3:134.
19. Zhang H, Wu A, Wang Y, Li G, Shapourian H, Shabani A, Ding Y. Oneq: a compilation framework for photonic one-way quantum computation. In: Proceedings of the 50th annual international symposium on computer architecture. 2023. p. 1–14.
20. Nielsen MA. Cluster-state quantum computation. Rep Math Phys. 2006;57(1):147–61.
21. Preskill J. Quantum computing in the nisq era and beyond. Quantum. 2018;2:79.
22. Ding Y, Holmes A, Javadi-Abhari A, Franklin D, Martonosi M, Chong F. Magic-state functional units: mapping and scheduling multi-level distillation circuits for fault-tolerant quantum architectures. In: 2018 51st annual IEEE/ACM international symposium on microarchitecture (MICRO). IEEE; 2018. p. 828–40.
23. Nielsen MA, Dawson CM. Fault-tolerant quantum computation with cluster states. Phys Rev A, At Mol Opt Phys. 2005;71(4):042323.
24. Miyazaki J, Hajdušek M, Murao M. Analysis of the trade-off between spatial and temporal resources for measurement-based quantum computation. Phys Rev A. 2015;91(5):052302.
25. Amy M, Maslov D, Mosca M, Roetteler M. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans Comput-Aided Des Integr Circuits Syst. 2013;32(6):818–30.
26. Coecke B, Duncan R. Interacting quantum observables: categorical algebra and diagrammatics. New J Phys. 2011;13(4):043016.
27. Kissinger A, Wetering J. Pyzx: large scale automated diagrammatic reasoning. In: Coecke B, Leifer M, editors. Proceedings 16th international conference on quantum physics and logic, Chapman University, Orange, CA, USA, 10-14 June 2019. vol. 318. 2020. p. 229–41. https://doi.org/10.4204/EPTCS.318.14.
28. Raussendorf R, Browne DE, Briegel HJ. Measurement-based quantum computation on cluster states. Phys Rev A. 2003;68(2):022312. https://doi.org/10.1103/physreva.68.022312.
29. Fowler AG, Mariantoni M, Martinis JM, Cleland AN. Surface codes: towards practical large-scale quantum computation. Phys Rev A, At Mol Opt Phys. 2012;86(3):032324.
30. Montfort Y, De Bone S, Elkouss D. Fault-tolerant structures for measurement-based quantum computation on a network. Quantum. 2025;9:1723.
31. Karuppasamy K, Puram V, Johnson S, Thomas JP. A comprehensive review of quantum circuit optimization: current trends and future directions. Quantum Rep. 2025;7(1):2.
32. Qin Z, Li X, Zhou Y, Zhang S, Li R, Du C, Xiao Z. Emerging applications of measurement-based quantum computing. Quantum Inf Process. 2024;23(12):389.
33. Kashif M, Al-Kuwari S. Physical realization of measurement based quantum computation. IEEE Access. 2023;11:90105–30.
34. Scott JR, Balram KC. Timing constraints imposed by classical digital control systems on photonic implementations of measurement-based quantum computing. IEEE Trans Quantum Eng. 2022;3:1–20.
35. Singh P, Dasgupta R, Singh A, Pandey H, Hassija V, Chamola V, Sikdar B. A survey on available tools and technologies enabling quantum computing. IEEE Access. 2024;12:57974–91.
36. Sanders BC. Superconducting quantum computing beyond 100 qubits. Physics. 2025;18:45.
37. Castelvecchi D. Ibm releases first-ever 1,000-qubit quantum chip. Nature. 2023;624(7991):238–238.
38. AbuGhanem M. Ibm quantum computers: evolution, performance, and future directions. J Supercomput. 2025;81(5):687.
39. Lall D, Agarwal A, Zhang W, Lindoy L, Lindström T, Webster S, Hall S, Chancellor N, Wallden P, Garcia-Patron R, et al. A review and collection of metrics and benchmarks for quantum computers: definitions, methodologies and software. 2025. arXiv preprint arXiv:2502.06717.
40. Zhu C, Wu X, Yang Z, Wang J, Wu A, Zheng S, Wang X. Quantum compiler design for qubit mapping and routing: a cross-architectural survey of superconducting, trapped-ion, and neutral atom systems. 2025. arXiv preprint arXiv:2505.16891.
41. Fujii K. Quantum computation with topological codes: from qubit to topological fault-tolerance. vol. 8. Singapore: Springer; 2015.
42. Hein M, Eisert J, Briegel HJ. Multiparty entanglement in graph states. Phys Rev A, At Mol Opt Phys. 2004;69(6):062311.
43. Jones NC, Van Meter R, Fowler AG, McMahon PL, Kim J, Ladd TD, Yamamoto Y. Layered architecture for quantum computing. Phys Rev X. 2012;2(3):031007.
44. Gheorghiu V. Quantum resource estimation. In: ETSI/IQC quantum-safe workshop, London, United Kingdom. 2017. ETSI and Institute for Quantum Computing. Invited talk. Available at https://docbox.etsi.org/Workshop/2017/201709_ETSI_IQC_QUANTUMSAFE/TECHNICAL_TRACK/S03_THREATS/GHEORGHIU_IQC.pdf, Page 2.
45. Gheorghiu V, Mosca M. Quantum resource estimation for large scale quantum algorithms. Future Gener Comput Syst. 2025;162:107480.
46. Babbush R, McClean JR, Wecker D, Aspuru-Guzik A, Wiebe N, Childs AM. Low-depth quantum simulation of materials. Phys Rev X. 2018;8(1):011044.
47. Long C, Huang M, Ye X, Futamura Y, Sakurai T. Hybrid quantum-classical-quantum convolutional neural networks. Sci Rep. 2025;15(1):31780.

48. Deshpande A, Niroula P, Shtanko O, Gorshkov AV, Fefferman B, Gullans MJ. Tight bounds on the convergence of noisy random circuits to the uniform distribution. PRX Quantum. 2022;3(4):040329.
49. Murali P, Linke NM, Martonosi M, Abhari AJ, Nguyen NH, Alderete CH. Full-stack, real-system quantum computer studies: architectural comparisons and design insights. New York: ACM; 2019. p. 527–40.
50. LeBlond T, Dean C, Watkins G, Bennink R. Realistic cost to execute practical quantum circuits using direct Clifford+t lattice surgery compilation. ACM Trans Quantum Comput. 2024;5(4):1–28.
51. Bravyi S, Kitaev A. Universal quantum computation with ideal Clifford gates and noisy ancillas. Phys Rev A. 2005;71(2):022316.
52. Bombin H, et al. Universal hardware-efficient topological measurement-based quantum computation via color-code-based cluster states. Phys Rev Res. 2022;4(1):013010.
53. Nickerson NH, Li Y, Benjamin SC. Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links. Nat Commun. 2014;4:1756.
54. Asavanant W, Yokoyama S, Shiozawa Y, et al. Time-domain-multiplexed measurement-based quantum operations with 25-mhz clock frequency. Phys Rev Appl. 2021;16(3):034005.
55. Cross AW, Bishop LS, Sheldon S, Nation PD, Gambetta JM. Validating quantum computers using randomized model circuits. Phys Rev A. 2019;100(3):032328.
56. Krinner S, Lacroix N, Remm A, Di Paolo A, Genois E, Leroux C, Hellings C, Lazar S, Swiadek F, Herrmann J, et al. Realizing repeated quantum error correction in a distance-three surface code. Nature. 2022;605(7911):669–74.
57. AI GQ. Suppressing quantum errors by scaling a surface code logical qubit. Nature. 2023;614(7949):676–81.
58. Asavanant W, Shiozawa Y, Yokoyama S, Charoensombutamon B, Emura H, Alexander RN, Takeda S, Yoshikawa J-i, Menicucci NC, Yonezawa H, et al. Generation of time-domain-multiplexed two-dimensional cluster state. Science. 2019;366(6463):373–6.
59. Zhong H-S, Wang H, Deng Y-H, Chen M-C, Peng L-C, Luo Y-H, Qin J, Wu D, Ding X, Hu Y, et al. Quantum computational advantage using photons. Science. 2020;370(6523):1460–3.
60. Menicucci NC. Fault-tolerant measurement-based quantum computing with continuous-variable cluster states. Phys Rev Lett. 2014;112(12):120504.
61. Menicucci NC, Van Loock P, Gu M, Weedbrook C, Ralph TC, Nielsen MA. Universal quantum computation with continuous-variable cluster states. Phys Rev Lett. 2006;97(11):110501.

Publisher’s note