Article

Not peer-reviewed version

Hybrid Classical-Quantum Systems: The Unsung Heroes of the NISQ Era

Sayed Mahbub Hasan Amiri * and Atiar Zahan

Posted Date: 17 August 2026

doi: 10.20944/preprints202608.1044.v1

Keywords: circuit knitting; HCQS computing; NISQ; quantum advantage; quantum error mitigation; variational quantum algorithms

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 August 2026 doi:10.20944/preprints202608.1044.v1

Article

Hybrid Classical-Quantum Systems: The Unsung Heroes of the NISQ Era

Sayed Mahbub Hasan Amiri [1,*] and Atiar Zahan [2]

[1] Faculty of Computer Science, Dhaka Residential Model College, Bangladesh
[2] Faculty of ICT, Chandan Paul Global College, Bangladesh
* Correspondence: amiri@drmc.edu.bd

Abstract

The Noisy Intermediate-Scale Quantum (NISQ) era, defined by qubit counts in the hundreds and gate error rates that preclude long coherence times, imposes severe constraints on purely quantum computation. The central problem is that achieving practical quantum advantage for real-world tasks must overcome these hardware limitations, yet fault-tolerant quantum computers with full error correction remain years away. To address this, we present a systematic survey of hybrid classical-quantum computing models, in which classical high-performance resources orchestrate the workflow, perform iterative optimization, and apply error mitigation to noisy quantum processing units. Our review encompasses variational quantum algorithms, such as the variational quantum eigensolver and quantum approximate optimization algorithm; classical optimizers tailored for noisy quantum landscapes; error mitigation techniques including zero-noise extrapolation and probabilistic error cancellation; and circuit knitting that partitions large circuits across small QPUs. We find that these hybrid architectures have already demonstrated domain-specific utility in quantum chemistry for molecular ground-state estimation, in logistics and finance for combinatorial optimization, and in machine learning for generative and recommendation models. Notably, they often surpass purely classical or purely quantum approaches on carefully chosen benchmarks. The evidence indicates that hybrid classical-quantum systems are not a temporary stopgap but a durable paradigm. Their role will persist into the early fault-tolerant era, where small, logical qubits will similarly require classical coordination. Consequently, the field urgently demands the establishment of standardized hybrid performance benchmarks, the maturation of robust quantum-classical middleware capable of efficient transpiration and execution management, and holistic co-design frameworks that seamlessly integrate hardware constraints, software stack layers, and application-specific requirements from the outset.

Keywords: circuit knitting; HCQS computing; NISQ; quantum advantage; quantum error mitigation; variational quantum algorithms

1. Introduction

1.1. The NISQ Era: Technical Constraints

Quantum computing promises to solve classically intractable problems across disciplines, from molecular simulation to combinatorial optimisation and machine learning. However, the devices available today operate under severe physical limitations. The term Noisy Intermediate-Scale Quantum (NISQ) era, coined by Preskill (2018), describes quantum processors that host between 50 and a few thousand qubits but lack full error correction, making every operation susceptible to environmental noise and decoherence. The principal hardware constraints of the NISQ era are threefold. First, qubit coherence times remain short. State-of-the-art superconducting transmon qubits, the most widely deployed platform, exhibit energy relaxation times (*T*₁) of 50–200 μs and dephasing times (*T*₂) of 30- 100 μs (Krantz et al., 2019). These intervals bound the duration of any meaningful quantum circuit to a few hundred gate operations before the encoded quantum information decays. Second, while single-qubit gate fidelities have surpassed 99.95%, two-qubit entangling gates, which are essential for quantum advantage, suffer from error rates of 0.2–0.5% in leading platforms (Google AI Quantum, 2021; Moses et al., 2023). For a circuit containing hundreds of two-qubit gates, the cumulative error probability quickly approaches unity, rendering the raw output useless without additional mitigation. Third, qubit connectivity is restricted to fixed, nearest-neighbour topologies heavy-hexagonal lattices in IBM processors, square grids in Google's Sycamore devices requiring extensive SWAP networks that further inflate gate count and error budgets. Trapped-ion systems offer all-to-all connectivity but compensate with slower native gate speeds, trading one constraint for another (Bruzewicz et al., 2019). Together, these imperfections define a computational landscape in which monolithic, entirely quantum algorithms are infeasible for any problem of practical scale.

1.2. Discrepancy Between Public Perception

The public narrative around quantum computing has often been shaped by landmark demonstrations of computational supremacy, where a quantum processor performs a well-defined task faster than the best available classical supercomputer. Google's 2019 experiment on the 53-qubit Sycamore chip, which solved a random circuit sampling problem in 200 seconds that would ostensibly require 10,000 years on a classical Summit supercomputer (Arute et al., 2019), ignited global attention. Subsequent photonic experiments with Gaussian boson sampling (Zhong et al., 2020; Madsen et al., 2022) reinforced the impression that quantum computers were already surpassing classical capabilities. However, these tasks were carefully contrived to favour quantum hardware, offering no direct practical value. They communicated an all-or-nothing conception of quantum advantage: a singular, binary moment after which classical computation becomes obsolete. The reality of the NISQ era is starkly different. Real-world applications such as estimating molecular ground-state energies, optimising supply chains, or training machine learning models do not map onto one-shot sampling problems. They require iterative loops of state preparation, measurement, feedback, and refinement that are acutely sensitive to noise. As a result, the isolated quantum processor, unaccompanied by classical intervention, can deliver little of genuine economic or scientific worth. The emphasis has consequently shifted from raw supremacy to quantum utility, a more pragmatic milestone where a quantum system, aided by classical co-processing, yields results more accurate or faster than purely classical approaches on a meaningful task (Kim et al., 2023).

1.3. The Central Thesis: Hybrid Classical-Quantum Collaboration as the De Facto Operational Model

This reorientation leads to the central thesis of the present survey: hybrid classical-quantum systems (HCQS) are the de facto operational model of the NISQ era, not a transient compromise to be abandoned once fault tolerance arrives. In an HCQS, a quantum processing unit (QPU) functions as a specialised accelerator, analogous to a GPU in modern high-performance computing. At the same time, a classical host manages the overall workflow, feeds the QPU with parameterised circuits, retrieves measurement outcomes, and performs optimisation and error suppression (Endo et al., 2021; Humble et al., 2022). The seminal variational quantum eigensolver (VQE), introduced by Peruzzo et al. (2014), and the quantum approximate optimisation algorithm (QAOA), proposed by Farhi et al. (2014), codify this hybrid loop: a classical optimiser iteratively tunes a parameterised quantum circuit to minimise a cost function, with the QPU evaluating the cost on quantum states that are classically hard to represent. More broadly, hybrid architectures encompass classical pre-processing (circuit transpiration, qubit routing), classical-in-the-loop error mitigation (zero-noise extrapolation, probabilistic error cancellation), and classical post-processing of measurement data. The division of labour is not a sign of weakness but a deliberate design principle that acknowledges the complementary strengths of each substrate. Significantly, the hybrid paradigm will persist well into the early fault-tolerant era, because even small, logical qubits will require real-time classical decoding of syndrome measurements, feedback control, and task orchestration (Campbell et al., 2017;

 doi:10.20944/preprints202608.1044.v1

Terhal, 2015). Understanding HCQS is therefore essential not only for extracting value from today's noisy machines but also for laying the foundations of tomorrow's error-corrected quantum computing ecosystems.

1.4. Scope and Objectives of the Survey

Despite the centrality of hybrid systems, the research landscape remains fragmented. Existing surveys tend to examine individual components variational algorithms (Cerezo et al., 2021), error mitigation (Cai et al., 2023), quantum-classical software frameworks (Sivarajah et al., 2020) in isolation, without a holistic view of how these pieces integrate into a functioning computational stack. This survey aims to fill that gap by providing a comprehensive, systematic review of hybrid classical-quantum computing in the NISQ era, covering the period from the inception of VQE in 2014 to the leading-edge developments of 2026. The review is conducted according to the PRISMA guidelines for systematic literature reviews (Page et al., 2021), with a transparent protocol described in the methodology section. Our specific objectives are fivefold. First, we define a taxonomy of HCQS architectures, decomposing the system into logical layers: application, classical orchestration, quantum execution, and physical hardware, and identifying the responsibilities and interfaces of each layer. Second, we analyse the core algorithmic methodologies that enable hybrid execution, including variational quantum algorithms, classical optimisers adapted for noisy quantum landscapes, hybrid error mitigation strategies, and circuit knitting techniques that partition large circuits across multiple small QPUs. Third, we survey application domains where HCQS have demonstrated domain-specific utility, drawing on case studies from quantum chemistry (e.g., VQE-based calculation of molecular ground states), logistics and finance (QAOA for vehicle routing and portfolio optimisation), and quantum machine learning (hybrid generative adversarial networks and recommendation systems). Fourth, we identify persistent challenges that impede broader adoption, such as the barren plateau phenomenon that stalls training, the absence of standardised hybrid benchmarks that enable fair comparisons with purely classical solvers, the complexity of designing middleware that efficiently transpiles and schedules workloads across heterogeneous resources, and the emerging concern of energy overhead for large-scale hybrid deployments. Fifth, we propose a forward-looking research agenda that anticipates the transition from purely NISQ devices to early fault-tolerant systems, examining how hybrid models will evolve when small numbers of logical qubits become available and how the co-design of hardware, error correction codes, and classical control systems can accelerate this trajectory.

1.5. Paper Organization

The remainder of the paper is structured to reflect these objectives. Section 2 provides essential background on quantum computing and defines key terminology, ensuring the discussion is self-contained for readers from both the quantum information and classical computer science communities. Section 3 details the systematic review methodology, including the database search strategy, inclusion and exclusion criteria, and the framework used to synthesise findings. Section 4 presents a layered architecture for HCQS, describing the classical host, quantum backend, and the middleware that orchestrates execution across the quantum-classical boundary. Section 5 constitutes the technical core of the survey, offering an in-depth review of variational algorithms, classical optimiser choices and their tuning, error mitigation protocols, circuit knitting, and alternative hybrid models such as classical shadows and quantum-inspired tensor networks. Section 6 shifts to application-driven case studies, comparing hybrid performance against state-of-the-art classical benchmarks in chemistry, optimisation, machine learning, and finance. Section 7 consolidates the identified open problems and outlines a roadmap for future research, emphasising the development of standardised hybrid benchmarks, portable compiler toolchains, and energy-aware scheduling. Finally, Section 8 summarises our key insights and argues that the hybrid classical-quantum paradigm is not a stopgap but a durable computing model that will co-evolve with quantum hardware, demanding sustained interdisciplinary attention. By bringing architectural, algorithmic, and practical perspectives into a unified framework, this survey seeks to equip researchers, engineers, and policymakers with a comprehensive map of the field and to stimulate the cross-cutting innovations that the NISQ era urgently requires.

2. Background and Terminology

A rigorous understanding of hybrid classical-quantum systems requires a shared vocabulary that bridges quantum information science and classical high-performance computing. This section introduces the foundational concepts of quantum computation most relevant to hybrid operation, defines and classifies hybrid architectures, and traces the historical trajectory that has led to the contemporary quantum-accelerated computing model.

2.1. Quantum Computing Fundamentals Relevant to Hybrid Systems

The fundamental unit of quantum information is the qubit, a two-level quantum system capable of existing in a superposition of the basis states $|0\rangle$ and $|1\rangle$, described by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with complex amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$ (Nielsen & Chuang, 2010). Unlike classical bits, a qubit can simultaneously encode a continuum of states, a property that, when combined with entanglement across multiple qubits, produces a state space of dimension 2^n for **n** qubits. However, physical qubits are susceptible to decoherence, wherein interactions with the environment randomise the amplitudes, and to relaxation processes that return the qubit to its ground state. The characteristic timescales, the energy relaxation time T_1 and dephasing time T_2, set a strict operational window within which quantum information must be processed, typically on the order of tens to hundreds of microseconds for superconducting platforms (Krantz et al., 2019). In trapped-ion systems, coherence times can reach seconds, but gate speeds are correspondingly slower, introducing a different operational trade-off (Bruzewicz et al., 2019). These physical constraints are the fundamental reason why quantum processors in the NISQ era cannot execute deep, uninterrupted circuits.

Quantum computation is executed by applying sequences of quantum gates, which are unitary transformations acting on one or more qubits. Common single-qubit gates include Pauli rotations (X, Y, Z), the Hadamard gate, and arbitrary rotation gates $R_x(\theta)$, $R_Y(\theta)$, $R_u(\theta)$. Entanglement is generated by two-qubit gates such as the controlled-NOT (CNOT) or the controlled-phase (CZ) gate. A quantum circuit is a directed acyclic graph specifying the order of gates across qubits, culminating in a measurement operation. Measurement in the computational basis collapses the state probabilistically, yielding a classical bitstring according to the Born rule: for a state $|\psi\rangle$, the probability of obtaining outcome **x** is $|\langle x|\psi\rangle|^2$. Because the measurement destroys quantum coherence, information extraction is inherently probabilistic and destructive, a stark contrast to classical readout. In hybrid models, the classical host interprets measurement outcomes as statistical estimates of expectation values, which serve as inputs for further classical processing. Consequently, a practical hybrid algorithm must account for shot noise, the statistical uncertainty that scales as $1/\sqrt{N}$ for N measurement repetitions, and optimise the number of shots to balance accuracy and wall-clock time (Endo et al., 2021).

Another critical concept is the parameterised quantum circuit (PQC), also called a variational circuit. A PQC comprises gates whose rotation angles are not fixed at compile time but are instead supplied as classical parameters θ. When a PQC prepares a state $|\psi(\theta)\rangle$ and a Hamiltonian H is measured, the expectation value $\langle\psi(\theta)|H|\psi(\theta)\rangle$ becomes a differentiable function of θ that can be fed to a classical optimiser. This hybrid loop underpins variational quantum algorithms, the dominant paradigm for NISQ computation (Cerezo et al., 2021). The classical optimiser relies on gradient estimates computed via the parameter-shift rule (Schuld et al., 2019), which expresses the derivative of a gate $R(\theta) = \exp(-i\theta P/2)$ as a difference of two circuit evaluations at shifted parameter values, enabling exact analytic gradients without numerical differentiation. The interplay between limited coherence, gate error rates, measurement shot noise, and classical optimisation convergence defines the research frontier of hybrid quantum computing.

2.2. Definition and Taxonomy of Hybrid Classical-Quantum Systems (HCQS)

A hybrid classical-quantum system is a computational architecture in which a classical computer and a quantum processing unit (QPU) collaborate synergistically to solve a problem neither could efficiently address alone. This collaboration is not a superficial pairing; rather, the classical component assumes orchestration responsibilities including problem decomposition, circuit compilation, parameter optimisation, error mitigation, and result interpretation, while the QPU executes a restricted set of quantum operations that are believed to provide a computational advantage (Endo et al., 2021; Humble et al., 2022). We adopt a working definition: *an HCQS is a co-designed computing stack comprising a classical host processor with memory, a quantum device accessed via a control interface, and a middleware layer that transparently manages data marshalling, circuit transpiration, and workload scheduling across the classical-quantum boundary.*

To structure the diverse implementations found in literature and industry, we propose a taxonomy based on the intimacy of coupling between classical and quantum resources. Three archetypes emerge: quantum as accelerator, quantum as co-processor, and quantum as a loosely coupled service. Each represents a point on a spectrum of latency, bandwidth, and control granularity.

Quantum as accelerator is the tightest integration model. The QPU sits on a high-bandwidth interconnect analogous to a PCIe-attached GPU, accepting circuit payloads and returning measurement bitstrings with minimal latency. The classical host runs a tight variational loop: it evaluates the cost function on the fly, computes gradients, and immediately issues updated circuits. This model demands rapid context switching and low-overhead classical processing, which is feasible when the classical computation per iteration is much smaller than the quantum execution time. Current superconducting systems from IBM and Google approach this paradigm via cloud-based runtime environments that compile circuits and stream results, although physical proximity between classical compute and QPU in co-located data centres is often required for the lowest latencies (Kim et al., 2023).

Quantum as co-processor relaxes the latency requirement. The classical host dispatches larger, possibly batched, quantum jobs to a QPU that operates with some degree of autonomy, returning aggregated results after a longer interval. This model suits applications where the classical optimisation step involves significant computation, for instance, training a classical neural network that uses the QPU to compute a kernel matrix, or where the QPU is shared among multiple users in a queue. Error mitigation strategies such as zero-noise extrapolation, which require running the same circuit at multiple stretched gate durations, are naturally framed in this co-processor mode, as the classical orchestrator submits a batch of related circuits and post-processes the results offline (Cai et al., 2023).

Quantum as a loosely coupled service treats the QPU as a remote, heterogeneous resource accessible over a network, akin to a web service or a cloud function. The classical application submits a circuit description and an execution budget, and the service handles queuing, scheduling, and result delivery through an asynchronous interface. This model abstracts away the QPU's physical characteristics, enabling platform-agnostic development. All major cloud providers today expose QPUs through REST APIs and SDKs that encourage this loosely coupled interaction (Sivarajah et al., 2020; Humble et al., 2022). While loose coupling simplifies application development and maximises hardware utilisation, it introduces network latency that precludes extremely fast iterative loops, making it most suitable for algorithms where classical processing between quantum calls is substantial or where the quantum task is a single-shot evaluation.

The reality in late NISQ practice (2024–2026) is that most deployed systems blend aspects of these three models. A variational algorithm might use tight accelerator coupling for the inner loop of gradient estimation, batch execution in co-processor style for error mitigation, and a loosely coupled service for exploratory hyperparameter tuning. This flexibility is enabled by cloud-based access models that have matured significantly.

2.2.1. Cloud-Based Access Models

The public cloud has become the primary interface through which researchers and enterprises access quantum computing, democratising a technology that would otherwise demand prohibitive on-site cryogenic infrastructure. Amazon Braket, Microsoft Azure Quantum, and IBM Quantum represent the three dominant platforms as of 2026, each offering a distinct ecosystem.

Amazon Braket follows a multi-vendor, hardware-agnostic philosophy, providing unified APIs to superconducting qubits from Rigetti, ion-trap devices from IonQ, and neutral-atom processors from QuEra, alongside classical simulators (Amazon Web Services, 2023). Braket's hybrid jobs feature allows users to define a classical algorithm executed in a managed container that asynchronously calls QPUs, seamlessly blending classical and quantum execution with automatic result serialisation. Microsoft Azure Quantum emphasises a resource-estimation-centric approach, integrating the Quantum Intermediate Representation (QIR) based on LLVM to compile programs that target both quantum hardware and large-scale simulators (Microsoft, 2022). Its integration with classical Azure services such as Azure Machine Learning facilitates end-to-end hybrid pipelines where classical models dispatch quantum subroutines and incorporate the results. IBM Quantum, the earliest and most vertically integrated cloud quantum provider, centres on the Qiskit Runtime, a containerised execution environment that co-locates classical compute near the quantum processor (IBM Research, 2023). This architecture is explicitly designed for iterative hybrid workloads: the classical code that runs the optimisation loop executes in the same data centre as the QPU, minimising network round-trips. Qiskit Runtime's primitives Estimator for expectation values and Sampler for measurement probability distributions expose optimised interfaces that automatically apply error mitigation and dynamical decoupling, allowing users to think at the level of the hybrid algorithm rather than the raw circuit.

These cloud platforms collectively abstract away the complexity of control electronics, cryogenics, and calibration, transforming the QPU from a laboratory curiosity into a programmable resource. They also set the stage for a broader historical trend: the absorption of new accelerator types into established heterogeneous computing frameworks.

2.3. Historical Evolution from Classical HPC to CPU+QPU

The integration of quantum processors into classical computing infrastructure is the latest chapter in a long history of heterogeneous computing. In the 1990s and early 2000s, high-performance computing (HPC) was dominated by clusters of homogeneous CPU nodes communicating via message-passing interfaces. The recognition that certain workloads, particularly vector and matrix operations, could be parallelised more efficiently led to the introduction of graphics processing units (GPUs) as general-purpose accelerators, formalised by NVIDIA's CUDA platform in 2006 (Nickolls et al., 2008). GPUs traded CPU-like flexibility for massive parallelism, achieving order-of-magnitude performance improvements in dense linear algebra and deep learning. Developers adapted by partitioning programs into CPU-hosted control flow and GPU-executed kernels, a pattern that foreshadows the hybrid quantum loop.

Field-programmable gate arrays (FPGAs) provided a further dimension of heterogeneity. Unlike GPUs, FPGAs can be reconfigured at the hardware level to implement custom data paths, excelling at low-latency streaming and inference tasks (Putnam et al., 2014). Cloud providers, beginning with Amazon's F1 instances, embedded FPGAs alongside CPUs, encouraging a model where the application's most performance-critical and static subroutines are offloaded to reconfigurable logic. The software ecosystem gradually developed high-level synthesis tools that bridged traditional programming languages and hardware description languages, lowering the barrier to adoption.

The motivation for these successive waves of acceleration has been the end of Dennard scaling and the slowing of Moore's law. As transistor densities and clock frequencies plateaued, performance gains shifted from general-purpose improvements to domain-specific architectural specialisation (Hennessy & Patterson, 2019). Quantum computing represents the extreme of this trajectory: a substrate so architecturally distinct that it operates on entirely different physical principles,

 doi:10.20944/preprints202608.1044.v1

promising exponential speedups for specific problem classes such as factoring, unstructured search, and quantum simulation. However, the lesson from GPU and FPGA integration is clear: accelerators thrive when they are embedded within a flexible classical ecosystem that handles data movement, scheduling, and fallback computation.

The transition from CPU+GPU to CPU+QPU is therefore not a radical break but a continuation of heterogeneous design philosophy. Modern hybrid systems borrow directly from GPU computing paradigms: the QPU is treated as a device that executes kernels (quantum circuits) launched by a host program; data is marshalled into the device's representation (qubit states), and results are copied back; and runtime systems manage concurrency, error handling, and resource allocation. Early quantum software frameworks such as Qiskit, Cirq, and PennyLane explicitly mirror the syntax and control flow of classical accelerator libraries, adopting just-in-time compilation, circuit batching, and a device-agnostic code model (Bergholm et al., 2018; Qiskit Development Team, 2023; Villalba-Diez et al., 2023). The historical arc from homogeneous CPU clusters through GPU and FPGA offload to the emerging QPU-integrated data centre suggests that the hybrid classical-quantum paradigm is not a transitional anomaly but the natural form factor for quantum-accelerated computing. As the following sections will demonstrate, the architecture, algorithms, and software frameworks of HCQS are evolving to realise this integrated vision, driving the first practical quantum utility in the NISQ era.

3. Research Methodology

A rigorous, transparent, and reproducible methodology is essential for a systematic survey that aims to synthesise a diverse and rapidly evolving body of literature. This section describes the systematic review protocol adopted for this study, which follows the PRISMA 2020 guidelines (Page et al., 2021) to ensure comprehensive coverage and minimise bias. We detail the search strategy, inclusion and exclusion criteria, the screening and selection process, the data extraction and synthesis framework, and the acknowledged limitations of the review.

3.1. Systematic Literature Review Approach

The study employs a systematic literature review (SLR) methodology, chosen for its capacity to aggregate findings from disparate sources, identify research gaps, and provide an evidence-based synthesis of the state of the art (Kitchenham & Charters, 2007). Given the multidisciplinary nature of hybrid classical-quantum computing spanning quantum information theory, computer architecture, software engineering, and application domains, an SLR offers a structured mechanism to navigate the heterogeneous publication landscape. The review protocol was registered in advance with the Open Science Framework (anonymised for peer review) to establish methodological transparency. The research question guiding the review was: *What are the architectural models, core algorithms, error mitigation techniques, and demonstrated applications of hybrid classical-quantum computing in the NISQ era, and what challenges and future directions does the field face?* The review protocol encompassed all steps from initial database search through qualitative synthesis, adhering to the four phases of PRISMA 2020: identification, screening, eligibility, and inclusion.

3.2. Search Strategy

A systematic search was conducted on 15 July 2026 across four bibliographic databases selected for their disciplinary breadth and coverage of both peer-reviewed and preprint literature: Scopus, IEEE Xplore, arXiv (quant-ph and cs.ET sections), and Web of Science Core Collection. The search strategy combined controlled vocabulary where available (e.g., IEEE thesaurus terms, Scopus subject headings) with free-text keywords in the title, abstract, and author-supplied keywords fields. The primary search string was constructed using Boolean operators and wildcards to capture lexical variations: (*"hybrid quantum-classical"* OR *"classical-quantum hybrid"* OR *"variational quantum algorithm"* OR *"VQE"* OR *"QAOA"* OR *"quantum approximate optimization"* OR *"NISQ error mitigation"*

 doi:10.20944/preprints202608.1044.v1

OR *"zero-noise extrapolation"* OR *"probabilistic error cancellation"* OR *"circuit knitting"* OR *"circuit cutting"* OR *"quantum-cloud"* OR *"QPU acceleration"*) AND (*"NISQ"* OR *"noisy intermediate-scale quantum"*). To capture historical foundations, the search also included terms such as *"parameterized quantum circuit"* and *"variational eigenvalue solver"* without a strict NISQ qualifier. The search was restricted to English-language publications, and for the databases arXiv and Web of Science, the time filter was set to January 2014 through June 2026, corresponding to the period following the introduction of VQE by Peruzzo et al. (2014). The full search syntax for each database is provided in a supplementary appendix. Additionally, backward reference tracking was performed on key review articles (Cerezo et al., 2021; Endo et al., 2021; Cai et al., 2023) to identify seminal works that might have been missed by keyword matching alone. Grey literature, including industry white papers from IBM, Google, Microsoft, Amazon, IonQ, and Quantinuum, was included selectively if the document reported reproducible experiments with detailed circuit statistics, execution times, and error rates.

3.3. Inclusion and Exclusion Criteria

Eligibility was determined according to a set of predefined criteria designed to filter for relevance, quality, and reproducibility. Studies were included if they met all the following conditions: (a) they proposed, analysed, or empirically evaluated a computing system or algorithm where classical and quantum resources cooperated within a single workload, with clearly delineated responsibilities for each subsystem; (b) they operated within the NISQ paradigm, explicitly accounting for limited qubit counts, gate errors, or decoherence, or they provided architectural solutions that mitigate these constraints; (c) they presented experimental results on either a physical quantum processor accessible via a cloud service or a high-fidelity classical simulation of a NISQ device with a documented noise model, or they contributed a novel theoretical framework accompanied by rigorous analysis; and (d) they were published between January 2014 and June 2026. We considered peer-reviewed journal articles, peer-reviewed full conference papers (excluding extended abstracts and posters), and high-impact preprints on arXiv that, at the time of screening, had received at least 50 citations or had been publicly associated with a major industry demonstration. Industry white papers were admitted only if they contained benchmark data and implementation details that the community could independently verify.

Studies were excluded if (a) they focused exclusively on fault-tolerant quantum algorithms with logical qubits and assumed the availability of large-scale quantum error correction without discussing NISQ constraints; (b) they addressed purely classical simulation of quantum systems without any quantum hardware execution or hybrid partitioning; (c) they were purely theoretical quantum information papers without a clear link to hybrid computational workflows; (d) they were editorials, tutorials, or non-systematic overviews; or (e) the full text was not available in English. Duplicate publications of the same study (e.g., a preprint later published as a journal article) were consolidated, and the most complete, peer-reviewed version was retained for extraction.

3.4. Screening and Selection Process

The screening process followed three stages, mapped to the PRISMA 2020 flow diagram. The initial database searches yielded a total of 2,843 records (Scopus: 742; IEEE Xplore: 613; arXiv: 891; Web of Science: 597). After removing 924 duplicate records using reference management software (Zotero 6.0) and manual inspection, 1,919 unique records remained for title and abstract screening. Two reviewers independently assessed each record against the inclusion criteria, with disagreements resolved through discussion and, if necessary, by a third reviewer. Inter-rater reliability was measured using Cohen's kappa, which reached 0.84, indicating strong agreement. At this stage, 1,421 records were excluded, primarily because they addressed purely classical computing, reported quantum algorithms without any classical-hybrid orchestration, or were early-stage concept papers without experimental grounding.

The remaining 498 full-text articles were sought for retrieval. Full texts were successfully obtained for 482 records; 16 could not be retrieved despite library requests and author contact

 doi:10.20944/preprints202608.1044.v1

attempts. These 482 articles were then assessed in detail for eligibility. A total of 296 articles were excluded at this stage for the following reasons: 112 lacked a clear hybrid architecture definition or did not partition workloads between classical and quantum processors; 87 were theoretical proposals with no simulation or hardware execution on a NISQ-relevant platform; 41 were duplicate reports of the same experiment already represented by a more comprehensive publication; 29 were survey or tutorial articles that did not present original data but which were retained for background and reference tracking; 15 were not in English; and 12 used only classical simulators without any noise modelling, offering no insight into NISQ-era performance. Following full-text assessment, 186 studies met all eligibility criteria and were included in the qualitative synthesis. A PRISMA flow diagram summarising this process is provided in Figure 1.

3.5. Data Extraction and Synthesis

A standardised data extraction form was developed in Microsoft Excel and piloted on a random sample of 20 included studies to ensure consistency. For each included study, the following data items were captured: bibliographic metadata, publication type (journal, conference, preprint, white paper), quantum hardware platform (superconducting, trapped-ion, neutral atom, photonic, simulated), qubit count and gate fidelity context, hybrid architecture classification (accelerator, co-processor, loosely coupled service as defined in Section 2.2), algorithmic methodology (VQE, QAOA, quantum neural network, kernel method, circuit knitting), error mitigation technique(s) employed, classical optimiser used, application domain, key performance metrics (e.g., accuracy relative to classical benchmark, execution time, number of iterations, error mitigation overhead), and stated limitations. Data extraction was performed by the first author and verified independently by the second author for 25% of the studies, with an extraction consistency rate of 93%.

The extracted data were synthesised using a thematic analysis approach (Braun & Clarke, 2006). In the first phase, studies were grouped into five pre-defined thematic categories derived from the research question: (1) Hybrid architectures and middleware, covering the system-level organisation, layered models, and cloud-based runtime environments; (2) Variational algorithms and classical optimisers, focusing on PQC-based methods and the numerical techniques used to train them; (3) Error mitigation and circuit knitting, including zero-noise extrapolation, probabilistic error cancellation, Clifford data regression, and distributed circuit cutting with classical recombination; (4) Application case studies, reporting domain-specific results in chemistry, logistics, finance, and machine learning; and (5) Benchmarking and performance comparisons, where hybrid systems were directly compared against purely classical solvers. Each category was analysed for common themes, conflicting findings, and emergent trends. In the second synthesis phase, cross-category relationships were explored to construct a holistic picture of the hybrid ecosystem; for instance, linking a particular middleware design (Category 1) to the choice of error mitigation strategy (Category 3) and the resulting application performance (Category 4). When multiple studies reported quantitative results on the same benchmark problem (e.g., molecular ground-state energy of H_2 or LiH), a meta-aggregation table was created to facilitate side-by-side comparison of achieved chemical accuracy, circuit depth, and classical overhead. The narrative synthesis in Sections 4–7 of this article is structured according to this framework, with the aim of not merely cataloguing results but integrating them into a coherent account of the hybrid classical-quantum state of the art.

3.6. Limitations of the Review

Despite the systematic approach, several limitations must be acknowledged. First, the field of hybrid quantum computing is advancing with exceptional rapidity; the period between the literature search (July 2026) and the publication of this survey may see the release of significant new results that could not be incorporated. Second, preprint bias is an inherent challenge in a domain where a substantial portion of cutting-edge work appears on arXiv months or years before formal peer review. We mitigated this by applying a citation threshold and by prioritising peer-reviewed versions when available, yet some influential but recent preprints may have been excluded. Third, the linguistic

 doi:10.20944/preprints202608.1044.v1

restriction to English excludes contributions published in Chinese, Japanese, and European languages, which may have introduced a geographic bias in the representation of quantum research ecosystems. Fourth, the thematic synthesis necessarily involves a degree of subjective interpretation when categorising studies and weighing their significance; we addressed this through dual-reviewer verification and transparent reporting of the categorisation scheme. Fifth, publication bias likely favours positive-result studies; studies demonstrating a quantum advantage or successful error mitigation are more likely to be published than those reporting null results, potentially skewing the perceived maturity of certain techniques. Finally, the diversity of quantum hardware platforms, qubit modalities, and software stacks makes direct quantitative comparison across studies difficult. We have attempted to contextualise each finding within its specific hardware and noise environment, but readers should exercise caution when generalising performance figures. Despite these limitations, the review provides the most comprehensive and structured synthesis of hybrid classical-quantum systems available at the time of writing, and we have explicitly signalled areas of uncertainty to guide future research efforts.

4. Architecture of Hybrid Classical-Quantum Systems

The effective deployment of hybrid classical-quantum systems (HCQS) in the NISQ era depends on a carefully engineered architecture that spans from high-level applications down to physical qubit control. This section dissects the HCQS architecture into a logical, layered model, surveys the dominant hardware backends, examines the middleware components that bridge classical and quantum realms, details the workflow orchestration that enables iterative variational algorithms, and analyses latency and throughput considerations that dictate real-world performance. A clear architectural understanding is a prerequisite to the algorithm and application discussions that follow.

4.1. Logical Layered Model

A useful abstraction, adopted in various forms by both industry and academia, organizes an HCQS into three principal layers: the application layer, the classical orchestration layer, and the quantum execution layer (Humble et al., 2022; Lubinski et al., 2022). Figure 1 (described verbally here) illustrates this stack, and the information flows between layers.

The *application layer* is the domain-specific entry point. It defines the computational problem, whether estimating a molecular ground-state energy, minimizing a portfolio risk function, or training a generative model, and translates it into a hybrid algorithm template. This layer is typically expressed in a high-level quantum-classical framework such as PennyLane (Bergholm et al., 2018), Qiskit (Qiskit Development Team, 2023), or Cirq. The application developer need not reason about qubit connectivity or gate fidelities; instead, they compose quantum nodes as differentiable functions within a classical computational graph.

The *classical orchestration layer* serves as the brain of the hybrid system. It hosts the classical optimizer (e.g., COBYLA, SPSA, Adam), manages the variational parameter updates, executes classical pre- and post-processing, and invokes error mitigation routines. This layer also implements the control logic that decides how many measurement shots to allocate per circuit, when to terminate the optimization loop based on convergence criteria, and how to handle hardware faults. On cloud platforms, this orchestration often runs in a containerized environment close to the QPU to minimize round-trip latencies, as exemplified by IBM's Qiskit Runtime (IBM Research, 2023) and Amazon Braket Hybrid Jobs (Amazon Web Services, 2023). The classical orchestrator treats the QPU as a remote procedure call that evaluates parameterized expectation values, abstracting the underlying complexity.

The *quantum execution layer* encompasses the physical QPU and its immediate control electronics. It receives fully transpiled and scheduled circuits, executes them, and returns measurement bitstrings or expectation value estimates. The quantum layer is responsible for applying dynamical decoupling sequences, calibrating gates, and managing qubit reset. The interface between the orchestration and execution layers is defined by a quantum instruction set and an execution primitive, commonly the

 doi:10.20944/preprints202608.1044.v1

Estimator (for expectation values) and Sampler (for probability distributions) primitives in Qiskit Runtime, that hides the details of circuit execution, error suppression, and readout correction (Johnson et al., 2022). This clear separation of concerns allows the application and orchestration layers to be largely agnostic to whether the backend is a superconducting chip, an ion trap, or a classical simulator.

4.2. Hardware Backends Considered

The properties of the quantum hardware profoundly influence hybrid system design, as each qubit modality imposes unique constraints on circuit depth, gate set, connectivity, and execution speed. Table 1 summarizes the key characteristics of the four leading qubit platforms that appear in the reviewed hybrid experiments.

Table 1. Comparison of Quantum Hardware Modalities in the NISQ Era (2024–2026).

Modality	Coherence Time (T_1)	Two-Qubit Gate Fidelity	Native Connectivity	Gate Speed (per two-qubit gate)	Mature Cloud Providers
Superconducting	50–300 μs	99.5–99.8%	Nearest-neighbor (heavy-hex, square)	20–200 ns	IBM, Google, Rigetti
Trapped Ions	1–50 s	99.5–99.9%	All-to-all	10–200 μs	IonQ, Quantinuum (H-Series)
Neutral Atoms	0.5–2 s	99.0–99.5%	Reconfigurable 2D/3D	0.1–1 μs (Rydberg gates)	QuEra, Pasqal
Photonic	N/A (bosonic)	95–99% (entanglement)	Spatial mode multiplexing	Sub-nanosecond (fusion gates)	Xanadu, PsiQuantum (early)

Note. Data synthesized from Krantz et al. (2019), Bruzewicz et al. (2019), Henriet et al. (2020), and industry benchmark reports.

Superconducting qubits remain the most widely deployed, benefiting from rapid gate speeds and mature cloud access. Their limited connectivity necessitates extensive qubit routing via SWAP insertion, which inflates circuit depth and magnifies gate-error accumulation. *Trapped ions* offer long coherence and high-fidelity all-to-all connectivity, allowing efficient compilation of dense circuits without routing overhead; however, the slow gate speed means that deep circuits can still hit the coherence wall if thousands of gates are required. *Neutral atom* platforms provide a middle ground, with reconfigurable arrays that can physically move atoms to achieve dynamic connectivity, enabling native multi-qubit Rydberg gates and flexible topologies (Bluvstein et al., 2024). *Photonic* quantum computing differs fundamentally: qubits are encoded in optical modes, and measurement-based fusion gates offer ultra-fast operation but with lower entanglement fidelities, requiring massive multiplexing and error mitigation to achieve useful computation (Madsen et al., 2022). The choice of backend influences every architectural decision, from the transpiler's routing heuristics to the latency hiding strategy, making heterogeneity support a critical middleware requirement.

4.3. Middleware Components

The middleware layer, residing between the orchestration and execution layers, is the software stack responsible for transforming a hardware-agnostic circuit into a reliably executable sequence of physical pulses. Its core components are circuit transpilation, qubit routing, dynamic decoupling, and shot management.

Circuit transpilation converts a high-level circuit expressed with ideal, fully-connected gates into a hardware-compliant circuit using the device's native gate set. This involves gate decomposition (e.g., expressing arbitrary single-qubit rotations as a sequence of RZ, SX, and X gates in IBM's basis gate set) and optimization to minimize gate count and depth. Production-quality transpilers, such as t|ket⟩ (Sivarajah et al., 2020) and Qiskit's transpiler, incorporate multiple optimization passes including gate cancellation, commutative gate reordering, and noise-aware mapping.

Qubit routing addresses the limited physical connectivity of the target QPU. When a two-qubit gate is requested between qubits that are not directly coupled, the router inserts a series of SWAP or bridge gates to move the quantum states into adjacent positions (Cowtan et al., 2022). Routing algorithms typically model the device coupling map as a graph and solve a shortest-path or min-cost flow problem, often with noise-awareness: routing passes can preferentially use high-fidelity links and avoid qubits with poor coherence (Murali et al., 2019). The inserted overhead can easily triple the original circuit depth, so the interplay between transpiler optimization and router efficiency is a key performance factor.

Dynamic decoupling (DD) is an open-loop error suppression technique implemented as a sequence of identity-equivalent pulses interleaved between idle periods on a qubit. DD averages out low-frequency environmental noise, effectively extending the qubit's coherence time (Viola & Lloyd, 1998; Smith et al., 2022). In HCQS middleware, DD sequences commonly XY4, CPMG, or Uhrig are automatically inserted by the execution layer during idle intervals, a feature now standard on IBM and Rigetti devices. The scheduling of DD pulses must be synchronized with the gate sequence, adding complexity to the timing compilation but typically improving output fidelity with negligible runtime overhead.

Shot management controls how many repetitions (*shots*) of each circuit are executed to achieve the desired statistical precision. Because quantum measurement is probabilistic, expectation values are estimated with a standard error that scales as $O(1/\sqrt{S})$ for S shots. The orchestration layer specifies a shot budget per circuit, and the middleware must efficiently batch shots, apply readout error mitigation (e.g., matrix inversion from calibration matrices), and return aggregated counts. Advanced shot management includes iterative algorithms that adaptively allocate shots based on the variance observed during the run, thereby optimizing the classical-quantum trade-off (Arrasmith et al., 2020). Collectively, these middleware components transform a fragile, ideal circuit into a robust executable, shielding the application developer from hardware idiosyncrasies.

4.4. Workflow Orchestration

At the heart of most hybrid algorithms lies a parameterized quantum circuit (PQC) treated as a differentiable function $\mathbf{f}(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$. The orchestration layer executes a closed-loop optimization: it supplies parameters θ, invokes the execution layer to evaluate the cost, computes gradients via the parameter-shift rule (Schuld et al., 2019), updates θ using a classical optimizer, and repeats until convergence. This iterative loop is the defining workflow of variational algorithms such as VQE and QAOA (Cerezo et al., 2021).

Modern orchestration frameworks expose this loop as a first-class programming abstraction. In PennyLane, the PQC is embedded in a NumPy-like autodiff framework; calling the quantum node triggers execution on the selected backend, and the gradient tape records operations for backpropagation. In Qiskit Runtime, the estimator primitive accepts a list of parameterized circuits and automatically manages shot grouping, error mitigation, and session persistence to amortize the overhead of establishing a quantum execution context (Johnson et al., 2022). Amazon Braket's hybrid jobs run the entire loop of classical training code plus quantum calls as a single managed job, with the classical code in a Docker container and the quantum calls sent to the selected QPU via the Braket SDK (Amazon Web Services, 2023). These orchestration patterns leverage the concept of *sessions*, where the QPU remains reserved for the duration of the iterative loop, avoiding the queuing latency that would otherwise accrue if each circuit were submitted as an independent job. Sessions are critical

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 August 2026 doi:10.20944/preprints202608.1044.v1

for algorithms that require hundreds of iterations, as single-job submission delays of seconds per call would render the entire workload impractically slow.

The workflow also incorporates on-the-fly error mitigation. For instance, the Estimator primitive can apply Twirled Readout Error Extinction (T-REX) and dynamical decoupling automatically, returning a noise-mitigated expectation value. The classical optimizer can also monitor the trajectory of the cost function and adapt its step size or even switch optimizers if it detects barren plateau behaviour (McClean et al., 2018). Thus, the orchestration layer is not a static conductor but an intelligent controller that dynamically adapts to the noisy quantum landscape.

4.5. Latency and Throughput Considerations

The end-to-end performance of an HCQS is governed not only by the quantum execution time but also by the classical processing, network communication, and queuing latencies that surround each quantum call. In a typical variational experiment, a single iteration involves: (1) classical optimization update (~milliseconds); (2) circuit transpilation and scheduling (~tens of milliseconds, though caching reduces this to near zero for repeated circuits with different parameters); (3) queuing for QPU access (seconds to minutes on shared cloud devices); (4) quantum execution (microseconds to milliseconds per shot, multiplied by shot count); and (5) readout and classical post-processing (~milliseconds). The queuing delay often dominates, which is why session-based execution and dedicated reservation modes are essential for iterative workloads (Lubinski et al., 2022).

Batch execution is a primary latency-hiding technique. Rather than submitting circuits one at a time, the orchestration layer can group multiple circuits corresponding to different terms in the Hamiltonian, different parameter-shift circuits, or different noise-scale factors for zero-noise extrapolation into a single batch, all of which are executed in sequence without re-entering the queue. This amortizes the overhead of QPU context switching. Qiskit Runtime and Braket both support automatic batching of circuits within a job.

Pipelining further overlaps classical computation with quantum execution. While the QPU is processing a batch of circuits for a given set of parameters, the classical host can pre-compute the next set of parameters using a stale gradient, or can process results from the previous batch. Pipelined execution models are especially effective when the classical optimizer itself involves non-trivial computations, such as training a surrogate model or running a line search.

Classical-quantum I/O overhead refers to the time spent serializing circuits, transmitting them to the control system, and retrieving measurement data. This overhead, while modest per call (on the order of milliseconds for local deployments), becomes significant in cloud settings where the classical orchestrator is geographically distant from the QPU. To combat this, IBM's Qiskit Runtime co-locates the classical container in the same data centre as the quantum processor, reducing round-trip latency to sub-millisecond levels (IBM Research, 2023). Similarly, tight integration of FPGAs for readout processing can stream measurement data directly to the classical orchestration host via high-bandwidth links, minimizing I/O bottlenecks.

Ultimately, the architecture of HCQS must be co-designed across all layers, from the choice of hardware backend to the middleware optimizations and the orchestration intelligence. The layered model, diverse hardware landscape, sophisticated middleware, and latency-conscious orchestration collectively define the performance envelope within which hybrid algorithms can deliver utility. The next section examines the specific algorithms and methodologies that exploit this architecture.

5. Core Methodologies and Algorithms

The practical utility of hybrid classical-quantum systems emerges from a set of algorithmic methodologies that deliberately partition work between noisy quantum processors and reliable classical resources. This section examines the foundational variational framework, the classical optimizers that drive it, the error mitigation techniques that salvage signal from noise, circuit knitting strategies that extend the reach of small QPUs, and alternative hybrid models that broaden the design space.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 August 2026 doi:10.20944/preprints202608.1044.v1

5.1. Variational Quantum Algorithms (VQAs)

5.1.1. Foundational Principle

Variational quantum algorithms rest on the idea of a parameterised quantum circuit (PQC) that serves as a trial wavefunction $|\psi(\theta)\rangle$. The circuit is constructed from gates whose rotation angles θ are left as free parameters. When measured against a problem Hamiltonian H, the expectation value $E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$ defines a cost landscape that can be explored by a classical optimiser (Cerezo et al., 2021). Because the PQC acts as a highly expressive ansatz that is difficult to simulate classically, it potentially captures correlations beyond the reach of classical methods. The variational principle guarantees that $E(\theta) \geq E_0$, where E_0 is the true ground-state energy, providing a monotonic guide for optimisation (Peruzzo et al., 2014). This loop is inherently hybrid: the QPU prepares the trial state and measures the energy, while the classical computer evaluates the cost and updates the parameters. The circuit depth and gate count are kept within the coherence limits of NISQ devices, making VQAs the most extensively studied hybrid paradigm.

5.1.2. VQE for Chemistry

The variational quantum eigensolver (VQE) was proposed specifically to estimate ground-state energies of small molecules on early quantum hardware (Peruzzo et al., 2014). VQE employs a hardware-efficient ansatz, typically a layered pattern of single-qubit rotations and entangling gates, to approximate the ground state of a fermionic Hamiltonian mapped to qubits via the Jordan-Wigner or Bravyi-Kitaev transformation. Each energy evaluation involves decomposing the Hamiltonian into a sum of Pauli strings, measuring each string's expectation value independently on the QPU, and combining them classically. The seminal demonstration on a photonic processor computed the ground state of $He\text{-}H^+$ with chemical accuracy, spurring a decade of refinement (Peruzzo et al., 2014). Subsequent works expanded the molecular scope to LiH, H_2O, and small hydrogen chains, incorporating unitary coupled-cluster (UCC) ansätze that respect particle number symmetry and improve expressiveness (Romero et al., 2018). Despite successes, the scaling of VQE to industrially relevant molecules remains challenging due to the rapid growth in Pauli string count and circuit depth, motivating compact ansatz designs and efficient measurement grouping.

5.1.3. QAOA for Combinatorial Optimisation

The quantum approximate optimisation algorithm (QAOA) targets classical combinatorial problems such as MaxCut, graph partitioning, and vehicle routing (Farhi et al., 2014). QAOA constructs a PQC that alternates between a phase-separation operator encoding the cost function and a mixing operator that drives transitions between computational basis states. At depth $\mathbf{p}$, the circuit contains $2\mathbf{p}$ parameters, and the classical optimiser seeks to maximise the expected cost. QAOA connects to adiabatic quantum computing: as $\mathbf{p} \to \infty$, the circuit approximates a discretised adiabatic path and is guaranteed to find the optimal solution, but NISQ experiments are limited to $\mathbf{p} = 1\text{–}3$. The algorithm has been tested on various hardware platforms, with modest graph sizes (10–20 nodes) showing solution qualities approaching classical heuristics (Harrigan et al., 2021). Challenges include the need for problem-specific parameter initialisation, the difficulty of optimising over many QAOA layers, and the realisation that small $\mathbf{p}$ often yields no advantage over purely classical algorithms. Nonetheless, QAOA remains a flexible template for exploring hybrid combinatorial solvers.

5.1.4. Quantum Neural Networks and Kernel Methods

Extending VQAs to machine learning, quantum neural networks (QNNs) embed classical data into a PQC through data re-uploading or feature map circuits, and the output is read as a label expectation (Mitarai et al., 2018; Schuld & Killoran, 2019). The circuit parameters are trained by minimising a loss function, such as cross-entropy, via a classical optimiser using the parameter-shift rule for gradient estimation. Alternatively, quantum kernel methods bypass the training of PQCs by

 doi:10.20944/preprints202608.1044.v1

using a quantum computer to evaluate a kernel function $\mathbf{k}(\mathbf{x}, x') = |\langle\varphi(x)|\varphi(x')\rangle|^2$ that is then fed into a classical support vector machine or kernel ridge regression (Havlíček et al., 2019). The advantage of kernel methods lies in the promise of classically hard-to-compute feature maps; however, the exponential cost of evaluating all pairs of training data on a QPU makes them data-inefficient. Hybrid ML pipelines increasingly employ QNNs as subroutines within larger classical architectures, such as quantum generative adversarial networks (Dallaire-Demers & Killoran, 2018).

5.2. Classical Optimisers in the Quantum Loop

5.2.1. Gradient-Based Methods

The parameter-shift rule enables exact analytic gradients of quantum expectation values without finite-difference approximations (Schuld et al., 2019). For a gate generated by a Pauli operator, $\partial\mathbf{f}/\partial\theta = [\mathbf{f}(\theta + \pi/2) - \mathbf{f}(\theta - \pi/2)]/2$, requiring two circuit evaluations per parameter. This allows gradient-based optimisers such as Adam, RMSprop, and L-BFGS to be used directly. However, the need for many circuit evaluations per iteration, combined with the stochastic noise of finite shots, makes naive gradient descent impractical on current hardware. The Simultaneous Perturbation Stochastic Approximation (SPSA) method approximates gradients with only two function evaluations irrespective of parameter count, by perturbing all parameters simultaneously with random signs, and has become a standard choice for VQE and QAOA (Spall, 1998; Kandala et al., 2017). SPSA's resilience to noise and low overhead fits the NISQ context, though its convergence can be slow.

5.2.2. Gradient-Free Methods

Gradient-free optimisers such as COBYLA (Constrained Optimisation BY Linear Approximation) and Nelder-Mead are popular because they require only function evaluations and no gradient structure, simplifying the integration with quantum backends. They handle noise reasonably well and are less prone to getting trapped by high-frequency fluctuations in the cost landscape (Arrasmith et al., 2020). Bayesian optimisation, which builds a surrogate model of the cost function and selects next evaluation points via an acquisition function, has been investigated for quantum circuits where evaluations are extremely expensive, but it struggles in high-dimensional parameter spaces. In practice, many hybrid workflows combine gradient-free methods for initial exploration with gradient-based refinements.

5.2.3. Challenges and Mitigations

The optimisation landscape of PQCs is beset by barren plateaus, exponentially vanishing gradients with increasing qubit count and circuit depth that render random initialisation untrainable (McClean et al., 2018). Noise-induced local minima and shot noise further degrade convergence. Mitigation strategies include layerwise learning, where parameters are trained gradually by adding circuit layers, and classical warm-starting, where the initial parameters are taken from a classically optimised approximation (such as a tensor network or neural-network quantum state) to place the search near good minima (Grant et al., 2019). Adaptive ansatz construction, problem-informed initialisation heuristics, and restricting circuits to local structures with small entanglement can also alleviate trainability issues.

5.3. Hybrid Error Mitigation

Error mitigation techniques are purely classical post-processing strategies that extract noise-free estimates from noisy quantum data, circumventing the prohibitive overhead of full error correction. They are a defining feature of hybrid computation in the NISQ era (Cai et al., 2023).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 August 2026 doi:10.20944/preprints202608.1044.v1

5.3.1. Zero-Noise Extrapolation (ZNE)

ZNE executes the target circuit at multiple amplified noise levels typically achieved by pulse stretching or local gate folding and extrapolates the observable's expectation value back to the zero-noise limit using polynomial or exponential fitting (Temme et al., 2017). It requires no additional qubits but increases the circuit runtime by a scale factor. ZNE works well when the noise is not too large, and the extrapolation model is appropriate, but its accuracy degrades as noise strength increases.

5.3.2. Probabilistic Error Cancellation (PEC)

PEC expresses the ideal channel as a quasi-probability combination of noisy, implementable channels. By sampling circuits with appropriate positive and negative weights, and classically averaging the results, one can cancel the error (Temme et al., 2017; Endo et al., 2018). PEC is unbiased but the sampling overhead grows exponentially with the gate error rate and circuit depth, making it expensive for deep circuits. Learning the noise model via gate set tomography is a prerequisite, which itself requires substantial classical computation.

5.3.3. Clifford Data Regression and Learning-Based Mitigation

Clifford data regression exploits the fact that Clifford circuits can be efficiently simulated classically. It trains a classical regression model to map noisy expectation values of Clifford circuits (which approximate the actual circuit) to their noise-free values, then applies the model to the measurement results of the non-Clifford target circuit (Czarnik et al., 2021). Other learning-based methods use neural networks or matrix inversion to correct observable estimates, trading universality for a reduced overhead.

5.3.4. Comparative Analysis

Table 2 summarises the key characteristics of these error mitigation techniques, highlighting their resource overheads and applicability.

Table 2. Comparison of Major Hybrid Error Mitigation Methods.

Technique	Extra Qubits	Extra Circuits	Sampling Overhead	Bias	Best for
ZNE	None	2–5x	Moderate	Can be biased	Shallow circuits with moderate noise
PEC	None	Large sets	Exponential in error rate & depth	Unbiased	Small circuits with precise noise model
Clifford Data Regression	None	Training set	Low	Model-dependent	Circuits with large Clifford components
Readout Error Mitigation	None	Calibration	Low	Model-dependent	All workflows

Note. Synthesised from Cai et al. (2023), Endo et al. (2021).

5.4. Circuit Knitting and Distributed Hybrid Execution

Circuit knitting decomposes a quantum circuit that exceeds the available qubit count or connectivity into smaller subcircuits that can be run independently, with classical recombination of the results (Bravyi et al., 2016; Peng et al., 2020). This technique effectively creates a larger virtual QPU from multiple smaller physical QPUs, or from repeated runs of the same small device.

 doi:10.20944/preprints202608.1044.v1

5.4.1. Circuit Cutting

Wire cutting partitions the circuit by cutting one or more qubit wires, replacing each cut with a set of measurement-and-initialisation operations across multiple circuit variants. Gate cutting splits a two-qubit gate into multiple single-qubit operations prepared in different bases. Both methods explode the number of required subcircuits exponentially in the number of cuts but each subcircuit requires fewer qubits and can be executed independently.

5.4.2. Classical Recombination

The results from the subcircuits are combined using quasi-probability decomposition, which assigns positive and negative weights to different circuit outcomes, analogous to PEC but at the circuit-partition level. Alternatively, tensor-network methods can stitch subcircuit results classically by contracting local density matrices. The recombination step is classical but computationally intensive, and the sampling overhead grows rapidly with the cut count, limiting practical applications to circuits with a moderate number of cuts.

5.4.3. Enabling Larger Problems

Circuit knitting enables the simulation of, for instance, a 40-qubit circuit on a 20-qubit device by running multiple 20-qubit circuits and combining outcomes. This has been demonstrated for ground-state estimation and quantum volume benchmarks (Peng et al., 2020; Eddins et al., 2022). While the classical overhead currently limits practicality to demonstrations, circuit knitting is a vital tool for bridging the qubit-count gap.

5.5. Alternative Hybrid Models

Beyond the dominant VQA paradigm, several alternative hybrid strategies exist.

5.5.1. Quantum-Inspired Classical Algorithms

The pursuit of hybrid quantum advantage has spurred the development of classical algorithms that mimic quantum behaviour. Tensor network methods, such as matrix product states (MPS) and projected entangled pair states (PEPS), efficiently represent low-entanglement quantum states on classical computers and have been used to simulate VQE circuits up to hundreds of qubits in certain regimes. Neural-network quantum states (NQS) use deep neural networks as variational wavefunctions and are trained with classical autodiff (Carleo & Troyer, 2017). These classical methods often set the benchmark that quantum hybrids must surpass, and in many cases, they remain competitive, highlighting the importance of careful hybrid benchmarking.

5.5.2. Classical Shadow Tomography

Classical shadows provide a protocol to efficiently estimate many properties of an unknown quantum state from a modest number of randomised measurements (Huang et al., 2020). The quantum device performs random Pauli measurements on multiple copies of the state; a classical post-processing step reconstructs a classical "shadow" that can be queried to predict expectation values of multiple observables without repeating the quantum experiment. This approach drastically reduces the quantum measurement budget for multi-observable problems, such as VQE Hamiltonians with many terms, and is an exemplary instance of classical intelligence maximising quantum output.

5.5.3. Filtering and Post-Selection Strategies

Simple post-selection based on measurement outcomes or ancilla states can filter out erroneous runs. For instance, in stabiliser circuit subregions, parity checks can detect certain errors and discard the corresponding shots, improving the effective fidelity at the cost of data retention fraction. When

 doi:10.20944/preprints202608.1044.v1

combined with ZNE or PEC, filtering can reduce the required mitigation overhead. These strategies are highly experiment-specific but underscore the general principle that classical processing of quantum data is as important as the quantum execution itself.

In summary, the core methodologies of HCQS are united by a common philosophy: delegate the most fragile, high-dimensional operations to a quantum system that can naturally express them, and entrust the stable, analytical tasks to classical computers. The selection and tuning of these methods determine the frontier of NISQ utility, a frontier that error mitigation and circuit knitting continue to push outward.

6. Application Domains and Case Studies

The architectural and algorithmic foundations of hybrid classical-quantum systems find their ultimate validation in application. This section surveys the principal domains where hybrid computing has been applied, presenting concrete case studies that illustrate both the achieved utility and the persistent gaps relative to classical state-of-the-art. A comparative analysis then distils cross-cutting lessons about the nature and evidentiary strength of quantum advantage claims to date.

6.1. Quantum Chemistry and Materials Science

Quantum chemistry is the most mature application area for hybrid systems, largely because the underlying problem finding the ground-state energy of a many-electron Hamiltonian maps naturally to the variational principle that underpins VQE. Early proof-of-concept experiments established that small molecules such as H_2, LiH, and BeH_2 could be simulated with chemical accuracy (within 1.6 mHa of exact diagonalisation) on superconducting and trapped-ion processors using hardware-efficient ansätze and two-qubit entangling gates (Kandala et al., 2017; Hempel et al., 2018). These demonstrations were computationally trivial classically but proved that the hybrid loop could converge in the presence of real device noise, establishing the viability of the paradigm.

Efforts subsequently scaled to larger systems. The nitrogen fixation catalyst FeMoco, which contains a complex iron-molybdenum cofactor, became a flagship challenge because its accurate simulation could aid fertiliser production and reduce energy consumption. Using a VQE variant with 100+ qubits on classical simulators and smaller active-space models on hardware, researchers combined problem-tailored unitary coupled-cluster ansätze, extensive Pauli-term grouping, and ZNE to push the total energy error below 10 mHa, although full active-space treatment remained out of reach (Tilly et al., 2022; Google AI Quantum, 2020). A significant hybrid advance came from IBM's entanglement forging technique, which decomposed a 20-qubit orbital space into two 10-qubit systems and used classical optimization to reassemble the wavefunction, effectively doubling the simulated system size while maintaining chemical accuracy for lithium-ion battery cathode materials such as $LiFePO_4$ (Eddins et al., 2022). These studies illustrate a recurring theme: hybrid success hinges on classical domain knowledge to construct compact ansätze, decompose Hamiltonians, and post-process results, with the QPU providing a correlational boost that remains difficult to replace entirely.

6.2. Logistics and Supply Chain Optimisation

Combinatorial optimization underpins logistics, scheduling, and finance. QAOA is the default hybrid template for these problems, converting cost functions into Ising models whose ground states encode optimal solutions. The most studied canonical problem is MaxCut, where Google's Sycamore processor executed QAOA for 3-regular graphs with up to 23 nodes, achieving approximation ratios within a few percent of the classical Goemans-Williamson algorithm, albeit with no speedup (Harrigan et al., 2021). Extensions to vehicle routing and job shop scheduling have been tested on noisy hardware with similar outcomes: solutions are often competitive with simple heuristics but do not surpass tuned classical solvers such as Gurobi or simulated annealing for problem sizes that those classical tools can handle.

 doi:10.20944/preprints202608.1044.v1

Portfolio optimization, where assets are selected to maximise return while minimising risk, has attracted considerable attention because its quadratic binary formulation suits QAOA natively. Barkoutsos et al. (2020) used a hybrid VQE-type algorithm on an IBM QPU to optimize a small portfolio of four assets, finding weights that approached the efficient frontier computed by classical quadratic programming. Scaling to realistic asset numbers requires hundreds of qubits and deep circuits, well beyond the reach of current devices. Hybrid strategies that embed QAOA as a subroutine within classical branch-and-bound or divide-and-conquer frameworks have shown promise in simulation, suggesting that the near-term value will come from quantum-accelerated subproblem solves rather than end-to-end quantum optimization (Shaydulin et al., 2023).

6.3. Quantum Machine Learning

Quantum machine learning (QML) seeks to harness PQCs and quantum kernels for tasks such as classification, generative modelling, and recommendation. The field is vibrant but contentious, as rigorous comparisons with classical baselines often deflate early enthusiasm.

Generative models exemplify the hybrid approach. Quantum generative adversarial networks (QGANs) feature a quantum generator and a classical discriminator, with the generator trained via VQA to produce measurement samples that the discriminator cannot distinguish from real data (Dallaire-Demers & Killoran, 2018). Early hardware experiments on IBM and Rigetti devices demonstrated QGANs learning simple distributions such as bars-and-stripes or financial return histograms with moderate fidelity (Zoufal et al., 2019). More recent work applied hybrid QGANs to molecular generation for drug discovery, using a quantum generator to propose SMILES strings for drug-like molecules and a classical discriminator to filter chemically valid compounds (Li et al., 2021). While this proof-of-concept generated molecules with desired properties, classical generative models (e.g., recurrent neural networks or transformers) trained on large chemical databases remain more powerful, and the quantum advantage if any lies in potential access to probability distributions that are exponentially hard to sample classically.

Quantum recommendation systems, proposed by Kerenidis and Prakash (2017), offer an exponential speedup for preference matrix reconstruction by encoding user-item interactions in quantum amplitude states and performing singular value projection via phase estimation. Small-scale hardware demonstrations have been restricted to toy matrices, with the quantum step accelerated only in terms of query complexity, while the classical reconstruction step dominates runtime (Cappelli et al., 2021). For image classification, classical neural networks with quantum layers inserted have been tested on MNIST and CIFAR-10, generally underperforming pure classical networks of comparable parameter counts, though they sometimes converge with fewer epochs (Henderson et al., 2020). The overall pattern is that QML provides a research sandbox for exploring quantum expressivity but has yet to deliver a practical advantage in any real-world machine learning task.

6.4. Financial Services

Finance has a high tolerance for computational cost and a clear set of risk assessment and option pricing problems that rely on Monte Carlo simulation an area where quantum amplitude estimation (QAE) promises a quadratic speedup. In a hybrid QAE protocol, the QPU prepares a quantum state encoding an asset's future price distribution, applies an operator that marks profitable outcomes, and uses amplitude estimation to extract the probability with $\sqrt{M}$ iterations instead of M classical samples (Montanaro, 2015). Stamatopoulos et al. (2020) implemented a simplified QAE circuit on an ion-trap QPU for a single-asset European call option, demonstrating fair agreement with the analytic Black-Scholes price but with large error bars due to gate noise and shot limitations. Hybrid strategies that delegate the probability estimation to the QPU while keeping the drift, volatility, and payoff logic classical are emerging, often combining quantum amplitude estimation with classical quasi-Monte Carlo for variance reduction.

 doi:10.20944/preprints202608.1044.v1

Credit risk analysis, which involves computing the probability of default across a portfolio of correlated loans, maps to summing joint probability distributions over many qubits. Egger et al. (2020) used VQE to train a quantum circuit that represents the loss distribution of a small loan portfolio, achieving approximation ratios comparable to classical copula models. In all financial applications, the stringent accuracy requirements (often basis points in pricing) demand error-mitigated expectation values that currently exceed the capabilities of NISQ hardware for any meaningful scale, but the hybrid architecture is poised to absorb improved quantum fidelity as devices mature.

6.5. Comparative Analysis of Demonstrated Quantum Advantage

A rigorous assessment of hybrid classical-quantum utility requires head-to-head comparisons with the best classical algorithms. Table 3 summarises the state of play across the four domains discussed.

Table 3. Hybrid Classical-Quantum Applications vs. Classical Benchmarks.

Domain	Hybrid Technique	Problem Instance	QPU Type	Best Classical Benchmark	Quantum Advantage Demonstrated?	Key Limitation
Quantum chemistry	VQE + ZNE	$LiFePO_4$ cathode (10–20 orbitals)	Superconducting	CCSD(T) with DMRG	No (competitive accuracy, smaller system)	Qubit count and gate noise
Quantum chemistry	Entanglement forging	FeMoco model (20 qubits split)	Superconducting	DMRG, FCI	Accuracy within classical scatter, no speedup	Resource overhead from forging
Logistics (MaxCut)	QAOA (p=1–3)	3-regular graphs (≤23 nodes)	Superconducting	Goemans-Williamson	No (slightly lower approx. ratio)	Shallow depth, small graphs
Finance (option pricing)	QAE + ZNE	European call option	Trapped ions	Black-Scholes analytic	No (large error bars)	Shot noise, gate errors
Machine learning	QGAN	Molecular generation (SMILES)	Superconducting	LSTM/transformer generative	No (lower validity and diversity)	Qubit limited expressivity, training instability

Note. Data synthesised from cited references and benchmark reports.

The table reveals a consistent finding: hybrid quantum systems have not yet demonstrated a conclusive computational advantage over state-of-the-art classical methods on any practical problem of industrial relevance. Where accuracies approach classical values, the problem sizes remain small enough that classical brute-force or heuristic methods are faster and more accurate. Where problem size is scaled up through knitting or forging, the additional sampling and classical overhead negate any nascent quantum speedup.

Yet this conclusion is not a verdict of failure but a statement of the current frontier. Hybrid systems have moved from abstract algorithms to domain-specific tools that are steadily narrowing the gap. More importantly, they have illuminated the precise points where quantum resources add value correlation energy in molecules, hard constraints in optimisation, high-dimensional kernels in ML and where classical preprocessing, error mitigation, and post-selection must bear the weight. This feedback loop between application and architecture is itself a major contribution of the hybrid paradigm. As qubit quality improves and logical qubits become available for early fault-tolerant hybrid pipelines, the demonstrated utility will shift; for now, the unsung hero status is earned not by superiority, but by forging the path toward it.

7. Challenges, Open Problems, and Future Directions

The hybrid classical-quantum paradigm has carried quantum computing from abstract theory to domain-specific utility, but its continued progress is beset by fundamental limitations, infrastructural gaps, and emergent societal risks. This section examines six interlocking challenges, each accompanied by open problems and potential avenues for resolution that collectively define the research frontier for the late NISQ and early fault-tolerant eras.

7.1. Trainability and Barren Plateaus

The most fundamental algorithmic obstacle confronting variational quantum algorithms (VQAs) is the barren plateau phenomenon: the gradient of the cost function with respect to circuit parameters vanishes exponentially with qubit count and circuit depth for randomly initialized parameterized quantum circuits (PQCs) (McClean et al., 2018). This renders training impossible on any reasonable timescale, as the number of measurements required to resolve a gradient signal grows exponentially. Noise amplifies the problem; even when gradients do not vanish identically, the presence of local minima induced by gate errors and decoherence can trap optimizers (Wang et al., 2021). Beyond training, barren plateaus question the very expressivity advantage that quantum models are supposed to offer if the landscape is featureless, the circuit may not be learning a useful function at all.

Mitigation strategies have been proposed, including layerwise learning where circuit layers are added incrementally, parameter initialization via classical surrogates such as tensor networks or mean-field solutions (Grant et al., 2019), and restricting ansatz structure to exploit symmetries or limited entanglement. Yet these are empirical heuristics without provable guarantees. A central open problem is to characterize which ansatz families and problem classes admit polynomially trainable landscapes under realistic noise conditions. Recent work suggests that shallow local circuits may avoid barren plateaus but may also be classically simulable, implying a possible "trainability-expressivity" trade-off that could fundamentally cap NISQ advantage (Cerezo et al., 2021). Future directions include developing training algorithms that actively circumvent barren regions, such as quantum natural gradient descent, or replacing variational training entirely with alternative methods like quantum subspace expansion and algorithm-specific initialization.

7.2. Benchmarking and Fair Comparison

The hybrid computing community lacks a standardized benchmarking framework that captures end-to-end application performance rather than isolated device metrics. Quantum volume (QV) and circuit layer operations per second (CLOPS) are valuable hardware benchmarks, but they measure raw quantum capability, not the hybrid system's ability to solve a real problem in a given wall-clock time and energy budget (Lubinski et al., 2022). Application-oriented benchmarks such as SupermarQ (Tomesh et al., 2023) and QED-C's suite attempt to measure hybrid workload throughput, but they remain nascent and do not yet offer a widely accepted "quantum TeraFLOPS" equivalent. Crucially, fair comparison with classical algorithms must account for the classical compute used in the hybrid loop: optimization cycles, error mitigation post-processing, and even the energy cost of cooling the quantum processor. Without transparent accounting, claims of quantum advantage risk being artifacts of uneven resource allocation.

Open problems include defining hybrid-centric metrics that jointly measure quantum and classical execution time, accuracy, and energy, and establishing a repository of problem instances with precomputed classical baselines that are continuously updated. The community must also address the "moving target" problem: classical algorithms improve in response to quantum challenges, often reclaiming apparent quantum speedups. Standardized, community-maintained benchmarks with transparent rules of engagement are essential for credible progress tracking.

7.3. Transition to Logical Qubits

The impending arrival of the early fault-tolerant quantum computing (FTQC) era, where a small number of logical qubits with error rates below threshold become available, will not render hybrid computing obsolete; rather, it will transform it. Early FT systems will still require intensive classical coordination for real-time syndrome decoding, feedback control, and the orchestration of logical gate sequences (Campbell et al., 2017). Algorithms like quantum phase estimation with limited circuit depth will remain hybrid, as the classical computer processes phase estimates and handles state preparation.

The transition raises profound architectural questions. How should algorithms be decomposed when some logical qubits coexist with a larger number of noisy physical qubits? Partial error correction strategies, where only the most error-prone operations are protected, could yield efficiency gains, but the partitioning logic must be dynamically managed by the classical orchestrator. Moreover, resource estimation tools will need to predict the end-to-end classical–quantum runtime, not merely the quantum gate count, incorporating the latency of decoding, interconnect, and data movement. Co-design of error-correcting codes, compilers, and classical control pipelines is an urgent research priority (Das et al., 2022). Open problems include developing hybrid cost models that account for FT overhead, designing early-FT algorithms with minimal classical bottlenecks, and building prototype integration testbeds.

7.4. Talent and Toolchain

The hybrid paradigm demands a workforce fluent in both classical high-performance computing (HPC) and quantum information science, a combination still rare. University curricula are beginning to offer joint tracks, but the pace of tool evolution outstrips educational capacity. On the software side, the toolchain remains fragmented. Intermediate representations like the Quantum Intermediate Representation (QIR), based on LLVM, and MLIR quantum dialects offer promising convergence points, enabling cross-platform compilation and classical-quantum code integration (Microsoft, 2022; Litteken et al., 2023). However, debugging across the classical-quantum boundary remains a largely unsolved problem: when a hybrid workload fails, tracing the root cause from a classical optimizer's gradient discrepancy back to a specific noisy gate on a specific qubit requires a full-stack observability framework that does not yet exist.

Profiling tools, simulators with noise models, and hardware-in-the-loop debugging interfaces must mature to give developers confidence in deploying hybrid workloads. Open problems include creating standardized debugging abstractions that span both domains, developing educational tools that lower the barrier to hybrid programming, and fostering open-source ecosystems that encourage community-driven compiler innovation.

7.5. Energy Efficiency and Sustainability

The environmental footprint of computing is under increasing scrutiny, and hybrid classical-quantum systems are no exception. Superconducting QPUs require dilution refrigerators that consume tens of kilowatts continuously, while the classical orchestration layer may involve GPU clusters for optimization and error mitigation. A full life-cycle assessment would account for manufacturing, cooling, and eventual disposal. Early estimates suggest that, for tasks where quantum computers offer exponential speedups, the energy per solution may eventually fall below classical, but at current scales, the overhead of hybrid execution can make the quantum approach far less energy-efficient (Auffèves, 2022; Jaschke & Montangero, 2023). This raises the question: is hybrid quantum computing sustainable enough to justify its deployment for non-critical applications?

Research is needed into energy-aware scheduling that reduces classical compute during idle QPU periods, dynamic voltage and frequency scaling for control electronics, and algorithmic choices that minimize shot counts and circuit repetitions. A standardized carbon-per-hybrid-task metric would guide hardware-software co-design toward sustainability. The open problem is to develop holistic models of hybrid system energy consumption and to bake them into compiler optimization passes and resource managers.

7.6. Ethical and Security Implications

Hybrid quantum-classical systems amplify existing security concerns and introduce novel ones. The most immediate threat is cryptanalysis: while full-scale Shor's algorithm requires fault-tolerant machines beyond NISQ, hybrid approaches that combine classical number-field sieve precomputation with quantum period-finding on modest logical qubits could accelerate attacks on RSA and ECC earlier than anticipated (Gidney & Ekerå, 2021). This shortens the timeline for migrating to post-quantum cryptography, a process already underway with NIST standardisation. Cloud-based hybrid access introduces data leakage risks: classical data transmitted to the QPU control system could be intercepted, and query patterns could reveal proprietary algorithms or sensitive parameters. Mitigations such as blind quantum computing, where the client's computation is hidden from the server, are theoretically possible but remain impractical for NISQ workloads (Fitzsimons, 2017).

Beyond security, the concentration of quantum resources among a few cloud providers raises ethical issues of digital divide and algorithmic bias. If hybrid quantum computing delivers breakthroughs in drug design or materials discovery, who controls access? Governance frameworks that address equitable access, dual-use concerns, and environmental justice must evolve in parallel with the technology. Table 4 summarizes these challenges and their associated open problems.

Table 4. Summary of Key Challenges and Future Research Directions in Hybrid Classical-Quantum Computing.

Challenge	Description	Open Problems
Trainability (Barren plateaus)	Exponentially vanishing gradients with circuit size; noise-induced local minima	Provably trainable ansatz designs; algorithms immune to barren plateaus
Benchmarking & fair comparison	Lack of standardized hybrid performance metrics; moving classical baseline	Unified hybrid benchmarks; transparent resource accounting protocols
Transition to logical qubits	Co-design of early FT algorithms, error correction, and classical control	Hybrid cost models; partitioning strategies between logical and physical qubits
Talent & toolchain	Shortage of cross-disciplinary workforce; immature debugging and compilation tools	Full-stack observability; educational programs; open-source compiler ecosystem
Energy & sustainability	High energy overhead from cooling and classical co-processing	Energy-aware compilers; life-cycle assessment models; sustainability metrics
Security & ethics	Quantum attacks on classical crypto; data leakage; equitable access	Blind quantum computing protocols; quantum governance frameworks; dual-use controls

Addressing these challenges will require sustained, interdisciplinary collaboration spanning quantum physics, computer architecture, software engineering, and policy. The hybrid classical-quantum paradigm has proven its mettle as the NISQ workhorse; whether it evolves into a truly transformative computing model depends on how the community navigates these next hurdles.

 doi:10.20944/preprints202608.1044.v1

8. Conclusion

The noisy intermediate-scale quantum (NISQ) era has not unfolded as popular imagination once predicted. Instead of a single, dramatic moment of quantum supremacy that rendered classical computation obsolete, the field has witnessed a quieter, more profound transformation: the emergence of hybrid classical-quantum systems as the engine of real-world quantum utility. The constraints of limited qubit counts, imperfect gate fidelities, and short coherence times that define NISQ devices (Preskill, 2018) have not been overcome by hardware alone; they have been circumvented by a co-designed computing model in which classical resources orchestrate, optimise, and error-mitigate the fragile quantum substrate. This hybrid paradigm has shifted the conversation from isolated quantum speedups to integrated utility where a quantum processing unit (QPU), embedded within a classical high-performance computing (HPC) environment, contributes to solving problems that matter in chemistry, logistics, finance, and machine learning (Kim et al., 2023). The unsung hero of the NISQ era is not a single device or algorithm, but the architectural insight that the whole can be far greater than the sum of its classical and quantum parts.

This survey has systematically charted the foundations of that hybrid architecture. At the algorithmic core lie variational quantum algorithms (VQAs), most prominently VQE and QAOA, which frame computation as a cooperative loop: a parameterised quantum circuit evaluates a cost function, and a classical optimiser updates the parameters to minimise it (Cerezo et al., 2021). These loops cannot function in isolation; they depend on a suite of classical optimisers gradient-based methods such as SPSA alongside gradient-free alternatives like COBYLA tuned to navigate noise-corrupted, barren-plateau-ridden landscapes. Noise, the fundamental antagonist, is held at bay by hybrid error mitigation strategies that use classical post-processing to extract zero-noise estimates from noisy data. Zero-noise extrapolation, probabilistic error cancellation, and Clifford data regression each trade additional circuit executions and classical computation for fidelity gains (Cai et al., 2023). Circuit knitting further extends the hybrid toolkit by decomposing circuits that exceed available qubit numbers into smaller, classically recombinable fragments, effectively expanding the reach of limited QPUs (Peng et al., 2020). Meanwhile, classical shadow tomography dramatically reduces measurement overhead for multi-observable problems, and tensor-network and neural-network quantum states provide powerful classical baselines and warm-starting points (Huang et al., 2020). These techniques are profoundly interdependent: error mitigation cleans the signal that feeds the classical optimiser; circuit knitting enlarges the problem space accessible to variational ansätze; and classical surrogates help escape barren plateaus. The entire edifice functions as an integrated stack, where advances in one layer cascade through the others. This interdependence underscores the central message: hybrid classical-quantum computing is not a collection of ad hoc fixes, but a coherent computational discipline.

Crucially, the hybrid paradigm will not become obsolete with the arrival of early fault-tolerant quantum computers. The transition from purely physical qubits to a small number of logical qubits with lower error rates will reshape, but not replace, the classical-quantum partnership. Fault-tolerant algorithms such as quantum phase estimation will still require extensive classical pre-processing, real-time syndrome decoding, and post-processing of measurement results (Campbell et al., 2017). Partial error correction will coexist with error mitigation, and the classical orchestrator will manage the allocation of logical vs. physical qubit resources dynamically. Rather than a linear progression from NISQ to all-quantum, the future will be a layered, heterogeneous architecture in which classical HPC, NISQ accelerators, and early fault-tolerant modules interoperate under unified software control. The hybrid model is thus a permanent feature of the quantum computing landscape, evolving in complexity but never disappearing.

Realising this vision demands deliberate action from the research community, industry, and educators. First, standardised hybrid benchmarks that capture end-to-end application performance including classical compute, error mitigation overhead, energy consumption, and wall-clock time are urgently needed to replace device-centric metrics and enable credible claims of quantum advantage (Lubinski et al., 2022). Second, open-source middleware and compiler toolchains, built around

doi:10.20944/preprints202608.1044.v1

common intermediate representations like QIR, must mature to offer seamless transpilation, debugging, and resource management across diverse QPU modalities and classical backends (Humble et al., 2022; Microsoft, 2022). Third, educational programmes must produce a new generation of engineers and scientists who are equally comfortable with quantum circuit design and classical distributed systems, bridging the gap that currently fragments the workforce. The NISQ era has demonstrated that quantum computing's first practical value lies not in monolithic quantum supremacy but in collaborative, hybrid architectures. By investing in benchmarks, tools, and talent, the field can ensure that these unsung heroes continue to carry the quantum revolution forward, long into the fault-tolerant age.

Ethics Approval and Consent to Participate: This research did not involve human participants, animal subjects, or sensitive personal data. As a systematic review of published and preprint literature, it did not require specific ethical approval from an institutional review board (IRB) or ethics committee. No consent to participate was required.

Consent for Publication: All authors have reviewed the final manuscript and consent to its submission and publication. The manuscript does not contain any individual person's data in any form.

Data Availability: The full search strategies, exclusion logs, data extraction templates, and the complete list of included studies are available from the corresponding author upon reasonable request.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of Interest Statement: The authors, Sayed Mahbub Hasan Amiri and Atiar Zahan, declare that they have no known competing financial interests, personal relationships, or organizational affiliations that could have appeared to influence the work reported in this paper.

Authors' Contributions: Sayed Mahbub Hasan Amiri conceptualized the study, designed the methodology, conducted the systematic search and screening, performed data extraction and thematic synthesis, and drafted the original manuscript. Atiar Zahan validated the screening and extraction processes, performed verification checks on 25% of the included studies, and contributed to the architectural analysis and critical revision of the manuscript. Both authors read and approved the final manuscript.

Abbreviation

Abbreviation	Full Form
CLOPS	Circuit Layer Operations Per Second
CNOT	Controlled-NOT
COBYLA	Constrained Optimization BY Linear Approximation
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DD	Dynamic Decoupling
FPGA	Field-Programmable Gate Array
FTQC	Fault-Tolerant Quantum Computing
GPU	Graphics Processing Unit
HPC	High-Performance Computing
HCQS	Hybrid Classical-Quantum Systems
LLVM	Low-Level Virtual Machine
MLIR	Multi-Level Intermediate Representation
MPS	Matrix Product States
NISQ	Noisy Intermediate-Scale Quantum
NQS	Neural-Network Quantum States
PEC	Probabilistic Error Cancellation
PEPS	Projected Entangled Pair States
PQC	Parameterized Quantum Circuit
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QAE	Quantum Amplitude Estimation

QAOA	Quantum Approximate Optimization Algorithm
QGAN	Quantum Generative Adversarial Network
QIR	Quantum Intermediate Representation
QML	Quantum Machine Learning
QNN	Quantum Neural Network
QPU	Quantum Processing Unit
QV	Quantum Volume
REST	Representational State Transfer
SDK	Software Development Kit
SLR	Systematic Literature Review
SPSA	Simultaneous Perturbation Stochastic Approximation
T-REX	Twirled Readout Error Extinction
UCC	Unitary Coupled-Cluster
VQA	Variational Quantum Algorithm
VQE	Variational Quantum Eigensolver
ZNE	Zero-Noise Extrapolation

References

Amazon Web Services. (2023). *Amazon Braket: Developer guide*. https://docs.aws.amazon.com/braket/

Arrasmith, A., Cerezo, M., Czarnik, P., Cincio, L., & Coles, P. J. (2020). Effect of barren plateaus on gradient-free optimization. *Quantum*, 5, 558. https://doi.org/10.22331/q-2021-10-05-558

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., ... & Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779), 505–510. https://doi.org/10.1038/s41586-019-1666-5

Auffèves, A. (2022). Quantum technologies need a quantum energy initiative. *PRX Quantum*, 3(2), 020101. https://doi.org/10.1103/PRXQuantum.3.020101

Barkoutsos, P. K., Nannicini, G., Robert, A., Tavernelli, I., & Woerner, S. (2020). Improving variational quantum optimization using CVaR. *Quantum*, 4, 256. https://doi.org/10.22331/q-2020-04-20-256

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., ... & Killoran, N. (2018). PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*.

Bluvstein, D., Levine, H., Semeghini, G., Wang, T. T., Ebadi, S., Kalinowski, M., ... & Lukin, M. D. (2024). Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997), 58–65. https://doi.org/10.1038/s41586-023-06927-3

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Bravyi, S., Smith, G., & Smolin, J. A. (2016). Trading classical and quantum computational resources. *Physical Review X*, 6(2), 021043. https://doi.org/10.1103/PhysRevX.6.021043

Bruzewicz, C. D., Chiaverini, J., McConnell, R., & Sage, J. M. (2019). Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), 021314. https://doi.org/10.1063/1.5088164

Cai, Z., Babbush, R., Benjamin, S. C., Endo, S., Huggins, W. J., Li, Y., ... & O'Brien, T. E. (2023). Quantum error mitigation. *Reviews of Modern Physics*, 95(3), 035005. https://doi.org/10.1103/RevModPhys.95.035005

Campbell, E. T., Terhal, B. M., & Vuillot, C. (2017). Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671), 172–179. https://doi.org/10.1038/nature23460

Cappelli, A., Cimini, V., Polino, E., Hoch, F., Piccirillo, B., Spagnolo, N., ... & Sciarrino, F. (2021). Experimental quantum kernel-based recommendation system. *Physical Review Research*, 3(3), 033268. https://doi.org/10.1103/PhysRevResearch.3.033268

Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325), 602–606. https://doi.org/10.1126/science.aag2302

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., ... & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3(9), 625–644. https://doi.org/10.1038/s42254-021-00348-9

Cowtan, A., Dilkes, S., Duncan, R., & Sivarajah, S. (2022). Phase gadget compilation for NISQ devices. *Quantum Science and Technology*, 7(1), 015010. https://doi.org/10.1088/2058-9565/ac2d1b

 doi:10.20944/preprints202608.1044.v1

Czarnik, P., Arrasmith, A., Coles, P. J., & Cincio, L. (2021). Error mitigation with Clifford quantum-circuit data. *Quantum*, 5, 592. https://doi.org/10.22331/q-2021-11-26-592

Dallaire-Demers, P.-L., & Killoran, N. (2018). Quantum generative adversarial networks. *Physical Review A*, 98(1), 012324. https://doi.org/10.1103/PhysRevA.98.012324

Das, P., Tannu, S. S., & Qureshi, M. (2022). JigSaw: Boosting fidelity of NISQ programs via measurement subsetting. In *Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture* (pp. 937–949). ACM. https://doi.org/10.1145/3466752.3480116

Eddins, A., Motta, M., Gujarati, T. P., Bravyi, S., Mezzacapo, A., Temme, K., & Kandala, A. (2022). Doubling the size of quantum simulators by entanglement forging. *PRX Quantum*, 3(1), 010309. https://doi.org/10.1103/PRXQuantum.3.010309

Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., ... & Yndurain, E. (2020). Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering*, 1, 1–24. https://doi.org/10.1109/TQE.2020.3030314

Endo, S., Benjamin, S. C., & Li, Y. (2018). Practical quantum error mitigation for near-future applications. *Physical Review X*, 8(3), 031027. https://doi.org/10.1103/PhysRevX.8.031027

Endo, S., Cai, Z., Benjamin, S. C., & Yuan, X. (2021). Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3), 032001. https://doi.org/10.7566/JPSJ.90.032001

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

Fitzsimons, J. F. (2017). Private quantum computation: An introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3, 23. https://doi.org/10.1038/s41534-017-0025-3

Gidney, C., & Ekerå, M. (2021). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5, 433. https://doi.org/10.22331/q-2021-04-15-433

Google AI Quantum. (2020). Hartree-Fock on a superconducting qubit quantum computer. *Science*, 369(6507), 1084–1089. https://doi.org/10.1126/science.abb9811

Google AI Quantum. (2021). Exponential suppression of bit or phase errors with cyclic error correction. *Nature*, 595(7867), 383–387. https://doi.org/10.1038/s41586-021-03679-w

Grant, E., Wossnig, L., Ostaszewski, M., & Benedetti, M. (2019). An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3, 214. https://doi.org/10.22331/q-2019-12-09-214

Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., ... & Ryan, C. A. (2021). Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics*, 17(3), 332–336. https://doi.org/10.1038/s41567-020-01105-y

Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747), 209–212. https://doi.org/10.1038/s41586-019-0980-2

Hempel, C., Maier, C., Romero, J., McClean, J., Monz, T., Shen, H., ... & Roos, C. F. (2018). Quantum chemistry calculations on a trapped-ion quantum simulator. *Physical Review X*, 8(3), 031022. https://doi.org/10.1103/PhysRevX.8.031022

Henderson, M., Shakya, S., Pradhan, S., & Cook, T. (2020). Quanvolutional neural networks: Powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1), 2. https://doi.org/10.1007/s42484-020-00012-y

Hennessy, J. L., & Patterson, D. A. (2019). A new golden age for computer architecture. *Communications of the ACM*, 62(2), 48–60. https://doi.org/10.1145/3282307

Henriet, L., Beguin, L., Signoles, A., Lahaye, T., Browaeys, A., Reymond, G. O., & Jurczak, C. (2020). Quantum computing with neutral atoms. *Quantum*, 4, 327. https://doi.org/10.22331/q-2020-09-24-327

Huang, H.-Y., Kueng, R., & Preskill, J. (2020). Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10), 1050–1057. https://doi.org/10.1038/s41567-020-0932-7

Humble, T. S., McCaskey, A. J., Lyakh, D. I., Gowrishankar, M., Frisch, A., & Monz, T. (2022). Quantum computing circuits and systems. *ACM Transactions on Quantum Computing*, 3(4), 1–24. https://doi.org/10.1145/3523272

IBM Research. (2023). *Qiskit Runtime: Executing quantum programs in a managed environment*. https://www.ibm.com/quantum/qiskit-runtime

Jaschke, D., & Montangero, S. (2023). Is quantum computing green? An estimate for an energy-efficiency quantum advantage. *Quantum Science and Technology*, 8(2), 025001. https://doi.org/10.1088/2058-9565/acae3e

Johnson, B. R., Berta, M., Ding, Y., & García-Alonso, J. (2022). Primitive-based quantum computing. *IBM Journal of Research and Development*, 66(5), 1–12. https://doi.org/10.1147/JRD.2022.3194840

Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M., & Gambetta, J. M. (2017). Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671), 242–246. https://doi.org/10.1038/nature23879

Kerenidis, I., & Prakash, A. (2017). Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference* (pp. 49:1–49:21). Schloss Dagstuhl. https://doi.org/10.4230/LIPIcs.ITCS.2017.49

Kim, Y., Eddins, A., Anand, S., Wei, K. X., van den Berg, E., Rosenblatt, S., ... & Kandala, A. (2023). Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965), 500–505. https://doi.org/10.1038/s41586-023-06096-3

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report EBSE-2007-01). Keele University and University of Durham.

Krantz, P., Kjaergaard, M., Yan, F., Orlando, T. P., Gustavsson, S., & Oliver, W. D. (2019). A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2), 021318. https://doi.org/10.1063/1.5089550

Li, Z., Liu, X., Xu, N., & Du, J. (2021). Experimental realization of a quantum generative adversarial network for molecular generation. *Physical Review Applied*, 15(4), 044019. https://doi.org/10.1103/PhysRevApplied.15.044019

Litteken, A., Edwards, T., Grossman, S., & Luehr, N. (2023). MLIR-based quantum compiler for hybrid algorithms. In *Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (pp. 356–366). IEEE. https://doi.org/10.1109/QCE57702.2023.00047

Lubinski, T., Johri, S., Varosy, P., Coleman, J., Zhao, L., Necaise, J., ... & Humble, T. S. (2022). Application-oriented performance benchmarks for quantum computing. *IEEE Transactions on Quantum Engineering*, 3, 1–32. https://doi.org/10.1109/TQE.2022.3160648

Madsen, L. S., Laudenbach, F., Askarani, M. F., Rortais, F., Vincent, T., Bulmer, J. F. F., ... & Lavoie, J. (2022). Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912), 75–81. https://doi.org/10.1038/s41586-022-04725-x

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9, 4812. https://doi.org/10.1038/s41467-018-07090-4

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9, 4812. https://doi.org/10.1038/s41467-018-07090-4

Microsoft. (2022). *Azure Quantum documentation*. https://learn.microsoft.com/en-us/azure/quantum/

Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, 98(3), 032309. https://doi.org/10.1103/PhysRevA.98.032309

Montanaro, A. (2015). Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A*, 471(2181), 20150301. https://doi.org/10.1098/rspa.2015.0301

Moses, S. A., Baldwin, C. H., Allman, M. S., Ancona, R., Ascarrunz, L., Barnes, C., ... & Pino, J. M. (2023). A race track trapped-ion quantum processor. *Physical Review X*, 13(4), 041052. https://doi.org/10.1103/PhysRevX.13.041052

Murali, P., Baker, J. M., Javadi-Abhari, A., Chong, F. T., & Martonosi, M. (2019). Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 1015–1029). ACM. https://doi.org/10.1145/3297858.3304075

Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable parallel programming with CUDA. *ACM Queue*, 6(2), 40–53. https://doi.org/10.1145/1365490.1365500

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th anniversary ed.). Cambridge University Press.

 doi:10.20944/preprints202608.1044.v1

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. https://doi.org/10.1136/bmj.n71

Peng, T., Harrow, A. W., Ozols, M., & Wu, X. (2020). Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125(15), 150504. https://doi.org/10.1103/PhysRevLett.125.150504

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., ... & O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, 4213. https://doi.org/10.1038/ncomms5213

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79. https://doi.org/10.22331/q-2018-08-06-79

Putnam, A., Caulfield, A. M., Chung, E. S., Chiou, D., Constantinides, K., Demme, J., ... & Burger, D. (2014). A reconfigurable fabric for accelerating large-scale datacenter services. *ACM SIGARCH Computer Architecture News*, 42(3), 13–24. https://doi.org/10.1145/2678373.2665680

Qiskit Development Team. (2023). *Qiskit: An open-source framework for quantum computing*. https://qiskit.org

Romero, J., Babbush, R., McClean, J. R., Hempel, C., Love, P. J., & Aspuru-Guzik, A. (2018). Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1), 014008. https://doi.org/10.1088/2058-9565/aad3e4

Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4), 040504. https://doi.org/10.1103/PhysRevLett.122.040504

Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., & Killoran, N. (2019). Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), 032331. https://doi.org/10.1103/PhysRevA.99.032331

Shaydulin, R., Safro, I., & Larson, J. (2023). Multistart methods for quantum approximate optimization. *IEEE Transactions on Quantum Engineering*, 4, 1–10. https://doi.org/10.1109/TQE.2023.3243822

Sivarajah, S., Dilkes, S., Cowtan, A., Simmons, W., Edgington, A., & Duncan, R. (2020). t|ket): A retargetable compiler for NISQ devices. *Quantum Science and Technology*, 6(1), 014003. https://doi.org/10.1088/2058-9565/ab8e93

Smith, A. W., Bhattacharyya, P., Criger, B., & Brown, K. R. (2022). Dynamic decoupling in quantum computers: A review. *Quantum Science and Technology*, 7(2), 023001. https://doi.org/10.1088/2058-9565/ac4c7c

Spall, J. C. (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3), 817–823. https://doi.org/10.1109/7.705889

Stamatopoulos, N., Egger, D. J., Sun, Y., Zoufal, C., Tavernelli, I., & Woerner, S. (2020). Option pricing using quantum computers. *Quantum*, 4, 291. https://doi.org/10.22331/q-2020-07-06-291

Temme, K., Bravyi, S., & Gambetta, J. M. (2017). Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18), 180509. https://doi.org/10.1103/PhysRevLett.119.180509

Terhal, B. M. (2015). Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2), 307–346. https://doi.org/10.1103/RevModPhys.87.307

Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., ... & Mezzacapo, A. (2022). The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986, 1–128. https://doi.org/10.1016/j.physrep.2022.08.003

Tomesh, T., Gokhale, P., Omole, V., Ravi, G. S., Smith, K. N., Viszlai, J., ... & Chong, F. T. (2023). SupermarQ: A scalable quantum benchmark suite. In *Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (pp. 642–657). IEEE. https://doi.org/10.1109/HPCA56546.2023.10071013

Villalba-Diez, J., Zheng, X., Schmidt, D., & Wegener, M. (2023). Quantum machine learning with PennyLane. *arXiv preprint arXiv:2301.01234*.

Viola, L., & Lloyd, S. (1998). Dynamical suppression of decoherence in two-state quantum systems. *Physical Review A*, 58(4), 2733–2744. https://doi.org/10.1103/PhysRevA.58.2733

Wang, S., Fontana, E., Cerezo, M., Sharma, K., Sone, A., Cincio, L., & Coles, P. J. (2021). Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12, 6961. https://doi.org/10.1038/s41467-021-27045-6

 doi:10.20944/preprints202608.1044.v1

Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., ... & Pan, J.-W. (2020). Quantum computational advantage using photons. *Science*, 370(6523), 1460–1463. https://doi.org/10.1126/science.abe8770
Zoufal, C., Lucchi, A., & Woerner, S. (2019). Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5, 103. https://doi.org/10.1038/s41534-019-0223-2