Quantum Machine Learning: A Review of Hybrid Classical–Quantum Approaches

Bhavesh B. Prajapati* & *Riddhi B. Prajapati

IT Department, L. D. College of Engineering, Commissionerate of Technical Education, Government of Gujarat, India.
E-mails: b.b.prajapati@gmail.com, to.riddhi@gmail.com

Editors: ***Srikanth Radhakrishna* & *Danko D. Georgiev***

Article history: Submitted on October 6, 2025; Accepted on January 28, 2026; Published on January 30, 2026.

An interesting approach to using noisy intermediate-scale quantum (NISQ) devices is hybrid classical–quantum machine learning (QML). In these methods, classical processors handle optimisation and large-scale computation, while quantum hardware is devoted to tasks like feature mapping, nonlinear transformations, or kernel evaluation. Practical near-term demonstrations are made possible by this division of labour, which also mitigates existing hardware restrictions. In specialised fields like molecular modelling, materials discovery, and small-sample learning issues, recent advancements in variational quantum circuits, hybrid neural networks, and quantum kernel techniques have produced promising outcomes. However, scalability and wider applicability are still hampered by enduring issues including noise, barren plateaus, and the expense of repeated measurements. Long-term developments will require fault tolerance, logical qubits, and established software infrastructures, whereas near-term success depends on noise-aware algorithm design, repeatable experimental benchmarks, and enhanced error-mitigation strategies. When taken as a whole, these advancements show a viable path to achieving quantum advantage in machine learning.

Quanta 2026; 15: 1–12.

1 Introduction

From being largely a theoretical idea, quantum computing is now a quickly evolving experimental technology that has the potential to resolve issues that are still computationally unfeasible for classical systems. The evolution of scalable, fault-tolerant quantum algorithms is, however, hampered by several issues with the current generation of noisy intermediate-scale quantum (NISQ) devices, such as short coherence times, imperfect gate fidelities, and limited qubit availability, as shown by John Preskill [1] and further discussed by Marco Cerezo et al. [2]. Hybrid classical–quantum machine learning (QML) has become a useful approach to tackle these issues. Under this paradigm, conventional resources handle optimisation and large-scale data processing, whereas quantum processors are responsible for tasks like feature encoding, nonlinear mappings, or kernel evaluations. With today's hardware, this division of labour enables researchers to investigate significant quantum advantages, as demonstrated by S. Thanasilp et al. [3] and Y. Gujju et al. [4].

The terrain of hybrid QML is defined by a number of methodological approaches. While quantum kernel techniques take advantage of high-dimensional Hilbert space representations to improve small-data classification problems, variational quantum circuits (VQCs) and quantum–classical neural networks provide parameter optimisation using classical training loops as shown by Marcello Benedetti et al. [5]. Hybrid models are utilized to estimate molecular properties, improve diagnostic outcomes under limited data scenarios, and refine portfolio strategies in application areas such as computational chemistry, financial modelling, and healthcare analytics. In these domains, such techniques have already shown promise, as discussed by R. Prajapati and B. Prajapati [6].

Scalability is nevertheless hampered by practical constraints, despite these positive developments. The high cost of measurement, noise sensitivity, and barren plateaus in optimisation landscapes continue to be major challenges [2, 7]. Both algorithmic enhancements and collaboration with developments in quantum information science are necessary to overcome these problems. The ability of noise-resilient protocols such as quantum key distribution (QKD) to provide dependable performance in flawed hardware environments is demonstrated by perspectives from related domains, including quantum cryptography and quantum entropy research [8, 9]. It is anticipated that QML research will advance more quickly if similar concepts of error prevention, effective benchmarking, and hardware-aware design are applied.

In the near future, hybrid approaches are generally regarded as the most practical way to achieve quantum advantage in applied machine learning. Hybrid QML creates a workable balance between theoretical goals and engineering constraints by using NISQ devices just for certain subroutines and leaving large-scale optimisation to traditional processors.

2 Background

2.1 Quantum Computing Basics

By utilising quantum mechanical principles, quantum computing offers an alternative to traditional digital computation and presents a new method of information processing. Quantum bits, or qubits, are different from ordinary bits in that they can exist through a superposition, indicating that they can concurrently represent a mixture of both states, unlike traditional bits that are limited to binary values of 0 or 1. Because of this unique property, quantum systems can store and work with far more datasets than classical systems [10].

Entanglement, in which qubits become interconnected to the point where the state of one instantly affects the state of other, even though they are separated by great distances, is an equally significant concept. Numerous quantum computer activities and encryption methods rely on these quantum correlations, which surpass the classical assumption of locality [11–13].

Quantum circuits, which are structured collections of quantum gates, are used to carry out quantum processes. Qubit states can be changed coherently and probabilistically using gates like Hadamard, Pauli-X, and CNOT. Similar to logic gates in classical systems, these gates serve as the fundamental units of quantum algorithms, although they are governed by distinct mathematical and physical principles [14].

The NISQ era is the term utilised to explain contemporary quantum systems. This phrase describes devices with a limited quantity of qubits (usually tens to hundreds) that are prone to operational noise and defects because full-scale error correction is not present. For certain specialised applications, NISQ devices can still show advantages over classical systems in spite of these limitations, particularly when combined with classical processors in hybrid computational models [1].

The development of quantum-assisted machine learning solutions is made possible by the mixture of the phenomena of superposition, entanglement, quantum gate operations and the current technological status of NISQ devices.

2.2 Machine Learning Basics

Creating algorithms that allow systems to see patterns and make data-driven judgements without the need for explicitly written instructions is known as machine learning (ML), a key field within artificial intelligence. The three primary classifications that machine learning techniques fall under are supervised learning, unsupervised learning, and reinforcement learning [15].

In supervised learning, models are constructed using datasets that have both input characteristics and corresponding output labels. Identifying a generalizable link between inputs and precise outputs is the objective. In this area, support vector machines, decision trees, and neural networks are common techniques [16].

However, unsupervised learning methods use unlabelled datasets. The objective of these approaches is to identify underlying structures or clusters in the data. Algorithms like k-means clustering and principal component analysis (PCA) are commonly used in these types of tasks [17].

Reinforcement learning takes a different approach by letting an autonomous agent learn by interacting with its environment. By getting feedback in the form of incentives or penalties, the agent progressively enhances its operations to maximise long-term performance. This learning paradigm works particularly well when sequential control and decision-making are required [18].

The computational efficiency and scalability of traditional machine learning algorithms are becoming increasingly problematic, despite their notable success in a range of application domains. As data volume and complexity increase especially with the advent of deep learning models, so does the demand for substantial computational resources. Due to these constraints, researchers are exploring other approaches, such as quantum-enhanced machine learning, which may be able to alleviate some of the constraints in conventional techniques [5].

3 QML Basics

In order to create computational models that, under some conditions, might outperform conventional methods, a new multidisciplinary discipline known as Quantum Machine Learning (QML) blends machine learning and quantum computing. Purely quantum algorithms and mixed classical–quantum procedures are the two primary approaches in QML. Hybrid techniques are the most widely used and researched due to the constraints of current technology, particularly during the NISQ period.

3.1 What Is QML?

Quantum machine learning (QML) includes algorithmic frameworks in which quantum computers help with learning tasks that are typically performed by classical systems. Particularly in high-dimensional or probabilistic spaces, these algorithms may utilize quantum phenomena like superposition and entanglement to perform computations or transformations that are challenging for conventional systems [1]. Hardware from the NISQ era is particularly well-suited for hybrid models, which include quantum circuits in traditional machine learning processes. In these systems, classical components handle tasks like data management and parameter tuning, while quantum processors carry out specialised subroutines [19].

3.2 Types of QML: Fully Quantum vs. Hybrid Models

Fully quantum QML is defined as end-to-end learning pipelines fully implemented on quantum hardware. These models handle data encoding, model training, and inference using quantum resources. However, this type of QML requires fault-tolerant quantum devices, which are still in their development stage and not yet commonly available.

Conversely, hybrid QML blends the advantages of both classical and quantum computing. Here, parameterised quantum circuits (PQCs), quantum neural networks (QNNs), and quantum kernel techniques are used in classical optimisation frameworks [20]. Recent empirical evaluations have investigated hybrid architectures, including quantum convolutional neural networks (QCNNs), quantum-adapted ResNet models, and Quanvolution layers, with promising results for tasks like image recognition [21]. A comparative summary of these two QML paradigms is shown in Table 1.

3.3 Advantages and Challenges of Hybrid QML

3.3.1 Advantages

By reducing quantum circuit depth and offloading computationally intensive portions to classical processors, hybrid models complement existing NISQ hardware limitations [19]. These architectures have demonstrated beneficial parameter efficiency when compared to their traditional equivalents, using fewer adjustable parameters to occasionally achieve competitive accuracy [22]. Furthermore, especially in situations with low data or heavy noise, quantum kernel approaches can transfer data into higher-dimensional Hilbert spaces, enabling more powerful separation limits than classical support vector machines (SVMs) [23].

3.3.2 Challenges

Despite their potential, hybrid QML approaches provide certain difficulties. As circuit depth increases, gradient-based training methods become inefficient due to a common problem known as the barren plateau phenomenon, where the optimisation landscape flattens out [19]. Moreover, noise, decoherence, and restricted qubit fidelity in NISQ devices limit the overall reliability and model complexity [19]. Other practical challenges include the overhead of quantum data encoding, integration with traditional infrastructure, and increased computing complexity.

4 Architectures for Hybrid Classical–Quantum Learning

4.1 Variational Quantum Circuits (VQCs)

Variational quantum circuits (VQCs), a family of parameterised quantum circuits that are particularly well-suited to the constraints of the NISQ era, are the foundation of many hybrid quantum–classical algorithms. In a typical

Table 1: Comparison between Fully Quantum QML and Hybrid QML

Feature	Fully Quantum QML	Hybrid QML
Hardware Requirements	Fault-tolerant quantum	NISQ devices + classical computer
Feasibility (2025)	Experimental / future	Actively used
Training Type	Entirely quantum	Quantum + classical optimization
Application Status	Limited	Image classification, NLP, etc.

VQC procedure, a quantum circuit with parameters that are adjustable is configured, and its output is tracked to evaluate a cost function, and the parameters are adjusted using a traditional optimisation approach. Repeating this method iteratively allows quantum devices to perform certain computational subroutines while enabling classical computers to adjust parameters [24, 25].

The typical hybrid optimization cycle utilized in variational quantum algorithms is shown in Fig. 1. A parameterized quantum circuit made up of rotation and entangling gates first encodes classical input data into quantum states. A classical optimizer evaluates the measurement results and iteratively modifies the circuit parameters to minimize a task-specific cost function. This closed-loop interaction limits circuit depth while facilitating efficient training on NISQ-era hardware.

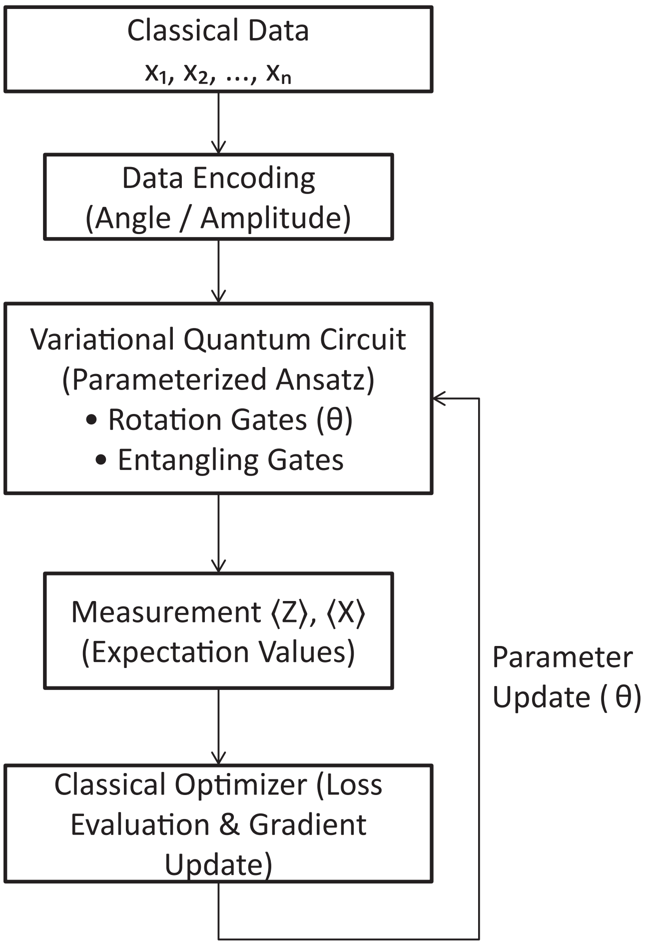

Figure 1: *VQC training workflow.*

Layers of single- and two-qubit gates typically make up a VQC architecture, with rotation angles and other gate settings being handled as learnable parameters. The circuit design, or ansatz, could be hardware-efficient, such as the one made to fit the connectivity of a specific quantum device, or problem-inspired, such as the unitary coupled-cluster ansatz frequently used in quantum chemistry. The ansatz selection has a significant impact on the expressive power and trainability of the circuit; more expressive circuits can capture complex quantum states but may be more challenging to tune [25, 26].

Examples of VQCs that are widely recognised include the Variational Quantum Eigensolver (VQE) and the Quantum Approximate Optimisation Algorithm (QAOA). With VQE, the ground-state energy of quantum systems is approximated by reducing energy readings over parameterised states [24]. In contrast, QAOA addresses combinatorial optimisation problems by mixing unitaries and alternating between problem-specific processes. Its settings are changed to get nearly ideal results [27].

Despite their potential, VQCs have many challenges. A major issue is the barren plateau phenomenon, which occurs when gradients vanish and learning is impeded when the optimisation terrain for deeper circuits becomes nearly flat [28, 29]. The errors caused by noise, decoherence, and low qubit fidelity in current NISQ devices can also impair accuracy. These algorithms likewise require many repeated measurements, or shots, for estimating cost functions with sufficient accuracy [25, 26]. Careful ansatz selection, noise-mitigation techniques, classical–quantum co-design methodologies, and sophisticated optimisers like the quantum natural gradient can all help to increase performance [30].

VQCs are essentially a versatile and flexible framework for computation that is strengthened by quantum mechanics. Progress in reducing noise, breaking through barren plateaus, and creating architectures that strike a compromise between expressiveness and trainability will be crucial to their practical impact [25, 30].

4.2 Quantum-Classical Neural Networks

Hybrid quantum neural networks (HQNNs), also known as quantum–classical neural networks (Q-CNNs), combine quantum processing layers with traditional neural network components. In these kinds of architectures, one or more quantum modules are integrated into a classical network to carry out specific tasks, usually data inference or embedding, while the classical portion handles pretreatment, post processing, and parameter optimization.

4.2.1 Using quantum layers to embed data

In hybrid networks, quantum layers are frequently used as a feature mapping stage, encoding conventional input vectors into quantum states. Widely used methods include amplitude encoding, rotation (angle) encoding, and domain-specific embedding schemes, the latter of which frequently uses entangling gates to identify correlations in the data [31, 32]. These embeddings can produce feature representations that could be difficult to acquire using classical approaches by projecting inputs into the exponentially vast Hilbert space of quantum states. Quantum kernel techniques, which have been experimentally verified on superconducting quantum hardware, are based on this idea [31].

4.2.2 Using quantum layers for inference

In an alternative configuration, quantum circuits are used to infer the model. In this instance, a parameterised quantum circuit processes the embedded state, and the measurement results are incorporated into later conventional layers, like softmax functions or fully connected layers, for the final decision. This arrangement enables the quantum block to function as a trainable nonlinear transformation and is comparable to a quantum filter or neuron [22, 33, 34].

4.2.3 Differentiability and hybrid training

Traditional gradient-based optimisation is typically used to train hybrid designs from start to finish. Specialised software toolkits make this integration seamless. For example, PennyLane offers differentiable quantum programming including automated differentiation via quantum circuits, enabling simultaneous optimisation of both quantum and conventional parameters [35, 36]. TensorFlow Quantum provides similar capability, allowing quantum circuits to be incorporated as native layers and trained alongside classical elements in the TensorFlow/Keras ecosystem [37]. Both frameworks interact with both simulators and real-world quantum devices, and they allow gradient estimation techniques including the parameter-shift rule and finite-difference methods.

4.2.4 Practical benefits and recent findings

On small-scale benchmarks, it has been shown that hybrid quantum–classical networks can sometimes meet or surpass classical baselines while requiring far fewer trainable parameters. This parameter efficiency could be helpful when interpretability or compactness are important considerations [22]. Thus far, applications have been explored in domain-specific applications like quantum chemistry and communication systems, as well as entity resolution and picture classification, with particularly promising results in scenarios with little training data.

4.2.5 Challenges and design factors

Hybrid QNNs have several drawbacks in spite of their promise. The substantial resources required to encode conventional inputs into quantum states may sometimes offset the expected performance gains. Moreover, current NISQ-era technology's low qubit counts, decoherence issues, and finite sampling rates restrict the depth and complexity of the model. Convergence in hybrid designs may also be slowed or stopped by issues known from variational quantum circuits, such as barren plateaus and noise-induced training difficulties [32, 36, 38]. Due to the computational cost of simulating vast quantum layers during development, scalability requires low-dimensional encodings and hardware-efficient designs.

In essence, quantum–classical neural networks provide a workable means of fusing quantum processing with modern deep learning frameworks. Platforms like PennyLane and TensorFlow Quantum facilitate the implementation of such models, and initial research in certain domains yields encouraging findings. For widespread practical use, however, advancements in co-design approaches, noise reduction, and data encoding techniques that optimise the complementarity among quantum layers and their classical equivalents will be required.

4.3 Quantum Kernel Methods

One important category of hybrid quantum–classical algorithms is represented by quantum kernel methods, which convert classical inputs into quantum states and use a quantum device to compute their pairwise inner products, or kernels. By incorporating these kernels into conventional kernel-based models, like support vector machines (SVMs) or kernel ridge regression, classical algorithms can then use quantum feature spaces that are hard to reproduce on classical hardware [31, 32].

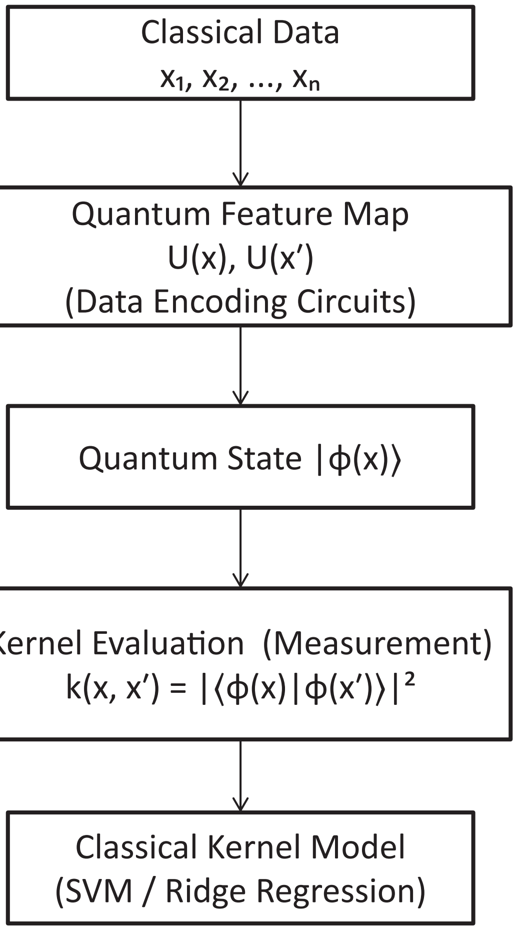

Figure 2: *Quantum kernel evaluation workflow.*

By encoding classical inputs into quantum states, estimating their overlap using quantum measurements, and evaluating the resulting kernel classically, quantum kernel methods compute similarity as shown in Fig. 2.

Typically, a quantum feature map is applied using either a parameterised or constant unitary transformation to encode the input vector x into a n-qubit quantum state. The fidelity, or other estimators using local measurement techniques, is then calculated as the correlation between two inputs [31, 32]. Once the kernel (Gram) matrix is obtained, sometimes by repeated quantum measurements, a classical kernel learner, such as an SVM, is trained to detect a decision boundary within the implicitly defined high-dimensional space.

One of the main benefits of quantum kernels is their expressiveness. Rich representations that achieve class separability—which is challenging to achieve in classical feature spaces because the Hilbert space dimension increases exponentially with the number of qubits—can be obtained from a well-designed quantum feature map. In certain theoretical situations, such mappings have been shown to yield observable distinctions between quantum and conventional model capabilities [39]. Another benefit is that kernel-based learning avoids the non-convex landscapes usually linked with variational quantum circuits by maintaining the convex optimisation structure of conventional techniques.

Despite these benefits, practical study has revealed serious limitations for immediate use. The high sample cost for accurately estimating kernel values is a key obstacle, especially if kernel entries are highly concentrated, which significantly reduces their discriminative potential by taking on nearly similar values for various inputs [3]. This type of concentration can be caused by global measurements, strong entanglement, quantum hardware noise, or the usage of excessively general or expressive embeddings. Analytical and empirical evidence suggests that these effects can impede generalisation and render the quantum kernel inefficient if the embedding design is not carefully considered [3, 40].

Many approaches have been developed to address these problems. Projected (local) kernels and covariant kernels preserve important discriminatory capabilities while mitigating kernel concentration and shot noise sensitivity by utilising data symmetries or computing similarities across smaller subsystems [3, 41]. Another approach is to design problem-inspired embeddings tailored to the structure of the dataset to maintain kernel informativeness and prevent exponential concentration [3]. Hybrid techniques, in which parameterised quantum embeddings are optimised together with conventional kernel alignment or regularisation, have also been studied to improve training stability and generality.

In conclusion, quantum kernel techniques offer a mathematically solid way to use quantum state spaces for supervised learning by fusing quantum embeddings with well-proven classical algorithms. Their practical efficacy is largely dependent on embedding decisions, measurement techniques, and hardware noise tolerance, despite their significant theoretical potential, particularly when using exponentially large Hilbert spaces. The present study aims to identify the design tenets and domains where quantum kernels can provide a real advantage over classical approaches [3, 40, 41].

5 Hybrid QML Applications

5.1 Quantum-Enhanced Classification

Hybrid quantum–classical models, which incorporate a quantum component such as a quantum kernel or parameterised quantum circuit to augment a conventional classifier, have been thoroughly researched for supervised classification. Compared to strictly classical models, research shows that these hybrid systems can produce richer data representations and, in some benchmark circumstances, yield better accuracy or lower parameter requirements.

Quantum layers have been successfully employed as feature mappers or kernel evaluators in recent experiments on small-to-medium-sized datasets, with classical layers handling the final classification [31, 32].

5.2 Quantum Reinforcement Learning

Hybrid quantum approaches have also helped reinforcement learning, especially in speeding up particular learning cycle components. Examples include estimating value functions, evaluating policies, and encoding probability distributions using quantum processes. In some situations, prototype hybrid Quantum Reinforcement Learning (QRL) agents that have been tested in dynamic contexts have shown promising improvements in sampling efficiency and adaptation over their classical counterparts. Performance gains, however, depend on the specific issue and can be impacted by quantum device noise. One important area for further research is the incorporation of quantum subroutines into traditional Q-learning [42].

5.3 Applications in Finance

Since the financial industry frequently works with huge, high-dimensional datasets and optimization challenges, it provides a vibrant testing ground for hybrid QML. These techniques have been used recently in time-series prediction, risk analysis, anomaly identification, and portfolio optimization. On limited datasets, hybrid models that make use of generative techniques, quantum kernels, or variational quantum circuits have demonstrated promising accuracy and robustness. However, issues with data encoding, noise resilience, and regulatory compliance must be resolved for real-world deployment [43, 44].

5.4 Chemistry and Drug Discovery

Ground-state energy estimation for tiny molecules has been made possible by hybrid quantum–classical algorithms, most notably the Variational Quantum Eigensolver (VQE), which has been at the forefront of near-term quantum chemistry applications. Recent developments apply quantum neural networks and quantum-enhanced generative models to drug discovery tasks such as conformer generation and molecular property prediction. Although these techniques have shown promise in proof-of-concept research, hardware constraints and the requirement for error mitigation still make it difficult to scale them to surpass classical computational chemistry [45].

5.5 Outlook and Implementation Considerations

When quantum subroutines work on compact, information-rich data encodings while classical components handle large-scale optimization and data handling, hybrid QML performs best across these domains. Algorithm design must be in line with hardware capabilities, noise must be managed, and rigorous benchmarking against robust classical baselines is essential for real-world deployment success. In order to translate hybrid QML from lab tests to real-world applications, the co-design approach in which algorithms, hardware, and data encoding are cooperatively optimized, remains essential [2, 46].

6 Current Challenges

The development of quantum machine learning (QML) is faced by several obstacles, including algorithmic, infrastructure, and technological ones. These constraints make it difficult to achieve a practical quantum advantage and limit the near-term scalability of QML solutions. Training bottlenecks in hybrid workflows, limited qubit resources and imperfect gates, noise and decoherence, and the immaturity of the software–hardware ecosystem are the four main problem areas that are identified (Table 2).

Table 2: *Challenging Areas*

Noise and decoherence
Limited qubit count and gate fidelity
Hybrid optimization bottlenecks
Software and hardware limitations

6.1 Noise and decoherence

Noise sources such as qubit dephasing, relaxation, crosstalk, and readout errors are inherent to devices in the NISQ era. These impacts weaken quantum states, distort measurement data, and ultimately affect algorithm performance. In addition to decreasing output fidelity, noise also affects the optimization landscape, leading to phenomena like variational objectives and kernels suppressing useful signals or noise-induced barren plateaus [2, 28]. Research indicates that hardware design has a major effect on noise characteristics, and many mitigation strategies have been put forth, such as error-aware transpilation, randomized compilation, and zero-noise extrapolation. Even though these techniques can increase accuracy in some situations, they frequently call for additional processing power and still do not achieve completely fault-tolerant performance [47, 48].

6.2 Limited qubit count and gate fidelity

Tens to several hundred qubits are typically available in modern quantum processors, and two-qubit gates have non-negligible error rates. Because cumulative gate faults cause performance to deteriorate exponentially with depth, the usable computing capacity is dependent on both the feasible circuit depth and the number of qubits [49]. Techniques like depth minimization, optimal qubit routing, and hardware-efficient ansatz have been spurred by this constraint. However, before large-scale, error-corrected computations are feasible, many target applications require either much improved gate reliability or significantly bigger qubit arrays [49, 50].

6.3 Hybrid optimization bottlenecks

Beyond the optimization difficulties inherent in quantum technology, hybrid quantum–classical training presents additional difficulties. Variational quantum algorithms (VQAs) may face extreme sensitivity to hyperparameter settings, high sampling (shot) needs for accurate gradient estimation, and barren plateaus [29, 51]. These plateaus could be caused by global cost function definitions, random circuit initializations, or even hardware noise. Localized cost functions, domain-informed ansatz creation, incremental layer training, and parameter initialization heuristics are some of the solutions proposed in recent research. These techniques may decrease expressivity and necessitate problem-specific adjustment, even though they can increase trainability [51]. Moreover, when executed on distant quantum computers, shot noise dramatically increases the number of quantum circuit evaluations every optimization step, leading to a rise in runtime and resource consumption [29].

6.4 Software and hardware limitations (integration and tooling)

The QML ecosystem as a whole is currently developing in its early phases. The lack of commonly recognized benchmarks, inadequate transpilers, hardware variability with mismatched native gate sets, and limited support for automated workflow orchestration among conventional high-performance computing (HPC) and quantum backends are some of the current obstacles [52, 53]. Because developers frequently have to manually adapt code to several SDKs, including Qiskit, Cirq, PennyLane, and TensorFlow Quantum, these problems impede repeatability and delay down experimentation. In order to facilitate long-term adoption in academic and industrial settings, recent studies highlight the necessity of strong middleware, improved software engineering standards, and standardized evaluation procedures [52, 54].

7 Future Outlook

Simultaneous advancements in quantum hardware, error-correcting methods, algorithm design, and supporting software ecosystems will influence the development of quantum machine learning (QML). While NISQ-era developments and hybrid algorithm tactics are expected to drive advancements in the near future, the long-term progress hinges on the development of large-scale, fault-tolerant quantum computers.

7.1 Prospects for Fault-Tolerant QML

Many of the present restrictions on QML, including restricted circuit depth, unstable multi-round training, and susceptibility to noise during data encoding, may be addressed via fault-tolerant quantum computing (FTQC). Before full-scale FTQC is available, research on transitional approaches, also known as Early Fault-Tolerant Quantum Computing (EFTQC), suggests systems with a limited amount of logical qubits and enhanced decoding techniques that can enable practical QML activities [19, 55]. The necessity of effective decoders, scalable error-correction protocols, and QML algorithms tailored for logical-qubit architectures are highlighted in strategic roadmaps [56]. Increasingly computationally intensive jobs, such as deep quantum neural networks, large-scale quantum kernel assessments, and sophisticated quantum generative models, may eventually be made possible by FTQC. Nevertheless, this will require solving major architectural challenges with hardware stability, decoder efficiency, and resource overhead [55, 56].

7.2 Near-Term Goals with NISQ Devices

Realistic goals at the current NISQ stage focus on creating algorithms that respect hardware limitations, generating repeatable performance standards, and proving concrete advantages over traditional approaches for specific challenges:

(1) producing reliable, hardware-based proof that hybrid QML models outperform robust classical baselines;

(2) creating noise-tolerant circuit ansatz and measurement schemes that maximize information extraction; and

(3) improving error-mitigation techniques, calibration procedures, and sampling effectiveness to lower experimental expenses are among the top priorities [4, 19].

Early claims should be interpreted with caution, according to the literature, which suggests that thorough performance comparisons and problem settings that make it evident when NISQ devices could offer real benefits [4]. Instead of general speedups, NISQ-era advancements are anticipated to produce specific, domain-focused advances in areas like materials research, chemistry, and small-sample categorization.

7.3 Software and Development Ecosystem Trends

The bridge between theoretical methods and actual quantum devices is provided by software frameworks. With features like high-level quantum layer abstractions, differentiable circuit construction for gradient-based optimization, and simple deployment to both simulators and real hardware, contemporary toolkits like Qiskit, PennyLane, Cirq, and TensorFlow Quantum are moving toward fully integrated hybrid-classical workflows [57]. Two main development trends are identified by recent surveys:

(1) a greater focus on hardware-specific compilation and execution, which aims to minimize circuit depth and align with native gate sets; and

(2) better orchestration for hybrid workloads, which includes cross-platform compatibility, cloud-native scheduling, and scalable integration with traditional machine learning pipelines [57, 58].

These software tooling developments are essential for speeding up the adoption of QML and enabling sophisticated, real-world applications, especially in the areas of standardizing benchmarks, guaranteeing reproducibility, and abstracting away hardware-specific variations.

8 Conclusion

The conceptual underpinnings and practical applications of hybrid classical–quantum machine learning (QML) have been examined in this study, with an emphasis on variational quantum circuits, quantum-classical neural network models, and quantum kernel-based techniques. The existing literature reveals a number of recurring themes. First, by allocating quantum processors to specific tasks—like feature encoding, nonlinear transformations, or cost function evaluation—and using classical systems for data preprocessing and optimization, hybrid approaches provide a workable way to take advantage of the capabilities of current noisy intermediate-scale quantum (NISQ) devices. Second, despite the field's progress from early theoretical concepts to experimental validation and benchmark studies, their efficacy for general-purpose applications is still hampered by enduring issues like noise sensitivity, optimization challenges like barren plateaus, and high measurement overheads [1–3].

In the immediate term, hybrid QML is still the most useful method for acquiring quantum advantage in applied machine learning contexts. Without requiring completely fault-tolerant hardware, targeted performance improvements in areas such as computational chemistry, low-data classification, and specialized optimization tasks have been demonstrated through the use of problem-specific ansätze, careful control over circuit depth, and the integration of quantum kernels or compact parameterized quantum blocks with classical models [2, 4]. However, recent studies indicate that expected benefits can occasionally fail to materialize in practice because of issues like hardware noise and kernel concentration, underscoring the significance of rigorous feature map selection, effective measurement procedures, and reliable benchmarking against robust classical baselines [3, 39].

Future progress will need concerted efforts on multiple fronts. The development of hardware-aware algorithms, transparent and repeatable benchmarking processes, and enhanced error mitigation and sampling optimisation techniques should be the main priorities in order to maximise NISQ-era performance in the near future. The addition of logical qubits, improved decoding efficiency, and scalable error correction will enable deeper and more expressive QML designs in the long run. In order to improve repeatability, reduce implementation complexity, and make it easier to transfer algorithms to different hardware platforms, software ecosystems such as Qiskit, PennyLane, Cirq, and associated middleware will need to be developed further [2].

The promise of hybrid QML as a transformative technology ultimately depends on reliable, verifiable evidence of its advantages in actual problem settings. The ability of hybrid QML to yield reliable, domain-relevant results will determine whether it becomes an essential part of machine learning procedures or remains a primarily exploratory research area.

References

[1] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum* 2018; **2**:79. `doi:10.22331/q-2018-08-06-79`.

[2] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, P. J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science* 2022; **2**(9):567–576. `doi:10.1038/s43588-022-00311-3`.

[3] S. Thanasilp, S. Wang, M. Cerezo, Z. Holmes. Exponential concentration in quantum kernel methods. *Nature Communications* 2024; **15**(1):5200. doi:10.1038/s41467-024-49287-w.

[4] Y. Gujju, A. Matsuo, R. Raymond. Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications. *Physical Review Applied* 2024; **21**(6):067001. doi:10.1103/PhysRevApplied.21.067001.

[5] M. Benedetti, E. Lloyd, S. Sack, M. Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology* 2019; **4**(4):043001. doi:10.1088/2058-9565/ab4eb5.

[6] R. B. Prajapati, B. B. Prajapati. Harnessing quantum uncertainty: Exploring the security landscape of quantum true random number generators. in: N. K. Chaubey, N. Chaubey (Eds.), Harnessing Quantum Cryptography for Next-Generation Security Solutions. IGI Global Scientific Publishing, Hershey, Pennsylvania, 2025. pp. 125–152. doi:10.4018/979-8-3693-9220-1.ch005.

[7] B. B. Prajapati, N. K. Chaubey. Realizing the quantum relative entropy of two noisy states using the Hudson–Parthasarathy equations. *International Journal of Advanced Computer Science and Applications* 2023; **14**(2):648–655. doi:10.14569/ijacsa.2023.0140275.

[8] B. B. Prajapati, N. K. Chaubey. Anatomy of quantum key distribution protocol: A comprehensive analysis. *Journal of Tianjin University Science and Technology* 2021; **54**(9):5k6pf. doi:10.17605/osf.io/5k6pf.

[9] N. K. Chaubey, B. B. Prajapati. Quantum Cryptography and the Future of Cyber Security. IGI Global, Hershey, Pennsylvania, 2020. doi:10.4018/978-1-7998-2253-0.

[10] M. A. Nielsen, I. L. Chuang. Quantum Computation and Quantum Information. 10th Edition. Cambridge University Press, Cambridge, 2010. doi:10.1017/cbo9780511976667.

[11] A. Einstein, B. Y. Podolsky, N. Rosen. Can quantum-mechanical description of physical reality be considered complete?. *Physical Review* 1935; **47**(10):777–780. doi:10.1103/PhysRev.47.777.

[12] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics* 1964; **1**(3):195–200. doi:10.1103/PhysicsPhysiqueFizika.1.195.

[13] J. S. Bell. On the problem of hidden variables in quantum mechanics. *Reviews of Modern Physics* 1966; **38**(3):447–452. doi:10.1103/RevModPhys.38.447.

[14] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A: Mathematical and Physical Sciences* 1985; **400**(1818):97–117. doi:10.1098/rspa.1985.0070.

[15] T. M. Mitchell. Machine Learning. McGraw-Hill, New York, 1997.

[16] C. M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York, 2006.

[17] I. T. Jolliffe. Principal Component Analysis. 2nd Edition. Springer, New York, 2002.

[18] R. S. Sutton, A. G. Barto. Reinforcement Learning: An Introduction. 2nd Edition. MIT Press, Cambridge, Massachusetts, 2018.

[19] S. Thudumu, J. Fisher, H. Du. Supervised quantum machine learning: A future outlook from qubits to enterprise applications 2025; arXiv:2505.24765.

[20] K. Zaman, T. Ahmed, M. A. Hanif, A. Marchisio, M. Shafique. A comparative analysis of hybrid-quantum classical neural networks 2024; arXiv:2402.10540.

[21] Y. Wang, J. Liu. A comprehensive review of quantum machine learning: from NISQ to fault tolerance. *Reports on Progress in Physics* 2024; **87**(11):116402. arXiv:2401.11351. doi:10.1088/1361-6633/ad7f69.

[22] L. Bischof, S. Teodoropol, R. M. Füchslin, K. Stockinger. Hybrid quantum neural networks show strongly reduced need for free parameters in entity matching. *Scientific Reports* 2025; **15**(1):4318. doi:10.1038/s41598-025-88177-z.

[23] D. T. Chang. Parameterized quantum circuits with quantum kernels for machine learning: A hybrid quantum-classical approach 2022; arXiv:2209.14449.

[24] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 2014; **5**(1):4213. doi:10.1038/ncomms5213.

[25] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, P. J. Coles. Variational quantum algorithms. *Nature Reviews Physics* 2021; **3**(9):625–644. `doi:10.1038/s42254-021-00348-9`.

[26] S. Sim, P. D. Johnson, A. Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum–classical algorithms. *Advanced Quantum Technologies* 2019; **2**(12):1900070. `doi:10.1002/qute.201900070`.

[27] E. Farhi, J. Goldstone, S. Gutmann. A quantum approximate optimization algorithm 2014; `arXiv:1411.4028`.

[28] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications* 2018; **9**(1):4812. `doi:10.1038/s41467-018-07090-4`.

[29] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, P. J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications* 2021; **12**(1):6961. `doi:10.1038/s41467-021-27045-6`.

[30] J. Stokes, J. Izaac, N. Killoran, G. Carleo. Quantum natural gradient. *Quantum* 2020; **4**:269. `doi:10.22331/q-2020-05-25-269`.

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

[32] M. Schuld, N. Killoran. Quantum machine learning in feature Hilbert spaces. *Physical Review Letters* 2019; **122**(4):040504. `doi:10.1103/PhysRevLett.122.040504`.

[33] D. D. Georgiev, S. K. Kolev, E. Cohen, J. F. Glazebrook. Computational capacity of pyramidal neurons in the cerebral cortex. *Brain Research* 2020; **1748**:147069. `doi:10.1016/j.brainres.2020.147069`.

[34] D. D. Georgiev. Quantum information in neural systems. *Symmetry* 2021; **13**(5):773. `doi:10.3390/sym13050773`.

[35] V. Dunjko, H. J. Briegel. Machine learning and artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics* 2018; **81**(7):074001. `doi:10.1088/1361-6633/aab406`.

[36] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. Di Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez Barrera, R. Moyard, Z. Niu, L. J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, N. Killoran. PennyLane: Automatic differentiation of hybrid quantum-classical computations 2018; `arXiv:1811.04968`.

[37] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa, E. Peters, O. Lockwood, A. Skolik, S. Jerbi, V. Dunjko, M. Leib, M. Streif, D. V. Dollen, H. Chen, S. Cao, R. Wiersema, H.-Y. Huang, J. R. McClean, R. Babbush, S. Boixo, D. Bacon, A. K. Ho, H. Neven, M. Mohseni. TensorFlow Quantum: A software framework for quantum machine learning 2020; `arXiv:2003.02989`.

[38] J. M. Arrazola, S. Jahangiri, A. Delgado, J. Ceroni, J. Izaac, A. Száva, U. Azad, R. A. Lang, Z. Niu, O. Di Matteo, R. Moyard, J. Soni, M. Schuld, R. A. Vargas-Hernández, T. Tamayo-Mendoza, C. Y.-Y. Lin, A. Aspuru-Guzik, N. Killoran. Differentiable quantum computational chemistry with PennyLane 2021; `arXiv:2111.09967`.

[39] H.-Y. Huang, R. Kueng, J. Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters* 2021; **126**(19):190505. `doi:10.1103/PhysRevLett.126.190505`.

[40] J. Schnabel, M. Roth. Quantum kernel methods under scrutiny: a benchmarking study. *Quantum Machine Intelligence* 2025; **7**(1):58. `doi:10.1007/s42484-025-00273-5`.

[41] M. Schuld, I. Sinayskiy, F. Petruccione. The quest for a quantum neural network. *Quantum Information Processing* 2014; **13**(11):2567–2586. `arXiv:1408.7005`. `doi:10.1007/s11128-014-0809-8`.

[42] A. Sannia, A. Giordano, N. L. Gullo, C. Mastroianni, F. Plastina. A hybrid classical–quantum approach to speed-up Q-learning. *Scientific Reports* 2023; **13**(1):3913. doi:10.1038/s41598-023-30990-5.

[43] P. Mironowicz, A. Shenoy-Hejamadi, A. Mandarino, A. E. Yilmaz, T. Ankenbrand. Applications of quantum machine learning for quantitative finance 2024; arXiv:2405.10119.

[44] R. Orús, S. Mugel, E. Lizaso. Quantum computing for finance: Overview and prospects. *Reviews in Physics* 2019; **4**:100028. doi:10.1016/j.revip.2019.100028.

[45] J. R. McClean, J. Romero, R. Babbush, A. Aspuru-Guzik. The theory of variational hybrid quantum–classical algorithms. *New Journal of Physics* 2016; **18**(2):023023. doi:10.1088/1367-2630/18/2/023023.

[46] Y. Liu, S. Arunachalam, K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics* 2021; **17**(9):1013–1017. doi:10.1038/s41567-021-01287-z.

[47] B. Khanal, P. Rivas, A. Sanjel, K. Sooksatra, E. Quevedo, A. Rodriguez. Generalization error bound for quantum machine learning in NISQ era–a survey. *Quantum Machine Intelligence* 2024; **6**(2):90. doi:10.1007/s42484-024-00204-w.

[48] P. García-Molina, A. Martin, M. Garcia de Andoin, M. Sanz. Mitigating noise in digital and digital–analog quantum computation. *Communications Physics* 2024; **7**(1):321. doi:10.1038/s42005-024-01812-5.

[49] J. Kalloor, M. Weiden, E. Younis, J. Kubiatowicz, B. De Jong, C. Iancu. Quantum hardware roofline: Evaluating the impact of gate expressivity on quantum processor design 2024; arXiv:2403.00132.

[50] S. S. Gill, O. Cetinkaya, S. Marrone, D. Claudino, D. Haunschild, L. Schlote, H. Wu, C. Ottaviani, X. Liu, S. P. Machupalli, K. Kaur, P. Arora, J. Liu, A. Farouk, H. H. Song, S. Uhlig, K. Ramamohanarao. Quantum computing: vision and challenges. in: R. Buyya, S. S. Gill (Eds.), Quantum Computing. Morgan Kaufmann, 2025. Ch. 2. pp. 19–42. doi:10.1016/B978-0-443-29096-1.00008-8.

[51] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, M. Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics* 2025; **7**(4):174–189. doi:10.1038/s42254-025-00813-9.

[52] J. M. Murillo, J. Garcia-Alonso, E. Moguel, J. Barzen, F. Leymann, S. Ali, T. Yue, P. Arcaini, R. Pérez-Castillo, I. García-Rodríguez de Guzmán, M. Piattini, A. Ruiz-Cortés, A. Brogi, J. Zhao, A. Miranskyy, M. Wimmer. Quantum software engineering: Roadmap and challenges ahead. *ACM Transactions on Software Engineering and Methodology* 2025; **34**(5):154. doi:10.1145/3712002.

[53] P. Mantha, F. J. Kiwit, N. Saurabh, S. Jha, A. Luckow. Pilot-Quantum: A quantum-HPC middleware for resource, workload and task management 2024; arXiv:2412.18519.

[54] E. Desdentado, C. Calero, M. A. Moraga, F. García. Quantum computing software solutions, technologies, evaluation and limitations: a systematic mapping study. *Computing* 2025; **107**(5):110. doi:10.1007/s00607-025-01459-2.

[55] A. Katabarwa, K. Gratsea, A. Caesura, P. D. Johnson. Early fault-tolerant quantum computing. *PRX Quantum* 2024; **5**(2):020101. doi:10.1103/PRXQuantum.5.020101.

[56] J. Bausch, A. W. Senior, F. J. H. Heras, T. Edlich, A. Davies, M. Newman, C. Jones, K. Satzinger, M. Y. Niu, S. Blackwell, G. Holland, D. Kafri, J. Atalaya, C. Gidney, D. Hassabis, S. Boixo, H. Neven, P. Kohli. Learning high-accuracy error decoding for quantum processors. *Nature* 2024; **635**(8040):834–840. doi:10.1038/s41586-024-08148-8.

[57] T. Farajollahpour. Quantum algorithm software for condensed matter physics 2025; arXiv:2506.09308.

[58] V. Stirbu, O. Kinanen, M. Haghparast, T. Mikkonen. Qubernetes: Towards a unified cloud-native execution platform for hybrid classic-quantum computing. *Information and Software Technology* 2024; **175**:107529. doi:10.1016/j.infsof.2024.107529.