FMDB ***Transactions on Sustainable Intelligent Networks***

Quantum Machine Learning: Theoretical Foundations, Algorithmic Frameworks, and Path to Quantum Advantage

Ghefar Alaa Aldeen Alrefai[1], Jouma Ali Al-Mohamad[2,*], Christo Ananth[3], Abdalla Nashat[4], Abdelrahman Hassan[5], Yusef Sherif[6]

[1]Department of Informatics and Communications Engineering, Faculty of Engineering, EBLA Private University, Idlib, Idlib Governorate, Syria.
[2]Department of Computer and Mobile Communication Engineering, Faculty of Information Engineering, Al-Shahbaa Private University, Aleppo, Aleppo Governorate, Syria.
[3]Department of Electronics and Communication Engineering, Samarkand State University, Samarkand, Samarkand Region, Uzbekistan.
[4,5,6]Department of Computer Science, Zewail City of Science and Technology, 6th of October City, Giza Governorate, Egypt.
ghefar.refai@ebla.edu.sy[1], jalmohamad@su.edu.sy[2], dr.christoananth@gmail.com[3], s-abdalla.nashat@zewailcity.edu.eg[4], s-abdelrahman.hassan23@zewailcity.edu.eg[5], s-yusef.sherif@zewailcity.edu.eg[6]

Abstract: Quantum machine learning sits at the intersection of two paradigm-shifting technologies: quantum computing and deep learning. By exploiting quantum mechanical phenomena – superposition, entanglement, and interference – QML promises exponential or quadratic speedups over classical algorithms for certain learning tasks, as well as the ability to model quantum data natively. However, the path to practical quantum advantage remains contested, constrained by noisy intermediate-scale quantum (NISQ) hardware, limited qubit counts (50–1,000), coherence times (50–500 μs), and high gate error rates (10^{-3}–10^{-4}). This paper provides a rigorous, technically deep survey of QML: mathematical foundations (Hilbert spaces, unitary evolution, measurement, quantum circuits), canonical algorithms (quantum kernel methods, variational quantum circuits (VQCs), quantum neural networks (QNNs), QSVMs, quantum generative adversarial networks (QGANs), encoding strategies (angle, amplitude, basis, and Hamiltonian encoding), optimization landscapes (barren plateaus, parameter shift rules, ansatz design), hardware aware implementation on superconducting (IBM, Google), trapped ion (IonQ), and neutral atom (Pasqal) platforms, empirical benchmarks on near term devices for classification, generative modeling, and quantum chemistry, and pathways to fault tolerant quantum machine learning (FTQML) with error correction (surface codes, logical qubits). Researchers critically assess claims of quantum advantage, identify scenarios where QML may outperform classical ML (small data, high-dimensional feature spaces from quantum systems, kernel estimation), and propose concrete evaluation metrics (quantum-classical cross-validation, resource estimation).

Keywords: Quantum Machine Learning (QML); Ansatz Design; Variational Quantum Circuits (VQCs); Encoding Strategies; Resource Estimation; Hamiltonian Encoding; Quantum Support Vector Machines (QSVMs).

Received on: 19/06/2025, **Revised on:** 06/09/2025, **Accepted on:** 21/10/2025, **Published on:** 15/08/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSIN

DOI: https://doi.org/10.69888/FTSIN.2026.000736

Cite as: G. A. A. Alrefai, J. A. Al-Mohamad, C. Ananth, A. Nashat, A. Hassan, and Y. Sherif, "Quantum Machine Learning: Theoretical Foundations, Algorithmic Frameworks, and Path to Quantum Advantage," *FMDB Transactions on Sustainable Intelligent Networks*, vol. 3, no. 3, pp. 164–178, 2026.

[*]Corresponding author.

1. Introduction

Quantum machine learning (QML) is a field at the intersection of two paradigm-shifting technologies, quantum computing and machine learning. It seeks to use the special properties of quantum mechanics, such as superposition, entanglement, interference and measurement, to design new algorithms for learning, optimization, data encoding and pattern recognition [2]. The main motivation is not simply to replace classical machine-learning algorithms with their quantum counterparts, but to identify computational problems where quantum information processing provides a significant advantage over classical information processing in terms of computational complexity, representational capacity, sampling efficiency or resource usage. Theoretical studies have demonstrated that quantum computers can offer exponential or polynomial speedups for some computational tasks. For problems such as unstructured search, improvements are quadratic in the number of iterations [3]. The exploitation of these theoretical advantages into practical machine learning advantages, however, remains an open research problem. The current generation of quantum processors mostly operates in the noisy intermediate-scale quantum (NISQ) regime, which is limited by a small number of physical qubits, limited coherence times, imperfect gates, measurement errors, and the lack of full fault-tolerant error correction [4]. Modern machine learning requires enormous computational resources. Training large language models, multimodal foundation models, scientific surrogate models, and high-resolution computer vision systems can take thousands of GPUs or specialized accelerators, running for weeks or months.

Therefore, the growth in model parameters and training datasets has led to an interest in alternative computational paradigms that could lower the computational complexity of some learning operations. One such possibility is quantum computing [5]. A quantum state of (n) qubits is mathematically represented in a Hilbert space of dimension (2^n). The exponentially large state space enables quantum circuits to perform complex probability-amplitude manipulation in parallel. Still, it is important to stress that a large Hilbert space does not automatically imply an exponential speedup in computation [6]. Efficient preparation, manipulation and measurement of useful quantum states remain fundamental challenges. QML thus aims to develop algorithms that use quantum resources in a targeted manner to tackle computational problems that are difficult to emulate on classical hardware efficiently [7]. The mathematical basis of QML is quantum information theory. A classical bit can be in one of two states, 0 or 1, whereas a qubit can be in a superposition ($|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$) where ($\alpha$) and ($\beta$) are complex probability amplitudes such that ($|\alpha|^2+|\beta|^2=1$). Tensor-product Hilbert spaces are made of multiple qubits; hence, an n-qubit system can have 2^n computational-basis amplitudes [8]. Quantum gates are unitary transformations that are reversible and preserve the normalization of quantum states. Entanglement is the term for correlations in the overall state of a collection of qubits that cannot be accounted for by the properties of the individual qubits. Interference can be harnessed to suppress the amplitudes of unwanted computational paths while amplifying those of useful ones.

Finally, the measurement process maps quantum information to classical data, imposing an important input/output restriction that, in general, requires multiple circuit runs to obtain accurate estimates of probabilities or expectation values. These principles form the foundations for several prominent QML paradigms. Quantum kernel methods map classical inputs to quantum feature spaces using quantum circuits and compute similarities between data points by measuring overlaps of quantum states [9]. Given a feature map ($U(x)$) mapping an initial state to ($|\phi(x)\rangle=U(x)|0\rangle$), a quantum kernel can be defined as ($K(x,x')=|\langle\phi(x)|\phi(x')\rangle|^2$) or by using related fidelity-based constructions. The resulting kernel can then be used as input to a classical support vector machine or any other kernel learning algorithm. The potential benefit is the ability to construct feature spaces whose geometry might be difficult to reproduce efficiently with standard classical kernels [10]. But the utility of a quantum kernel is bounded by the cost of state preparation, circuit execution, measurement and kernel-matrix estimation. The quantum kernel evaluation could require an extremely large number of measurements to compensate for the theoretical representational advantage, given the execution overhead in practice. Variational quantum circuits (VQCs) are another core framework for QML [11]. A VQC is composed of a parameterized quantum circuit ($U(\theta)$), where θ are the trainable parameters. The circuit is executed on a quantum processor, and an observable is measured. The parameters are updated by a classical optimizer using the measured objective function. This results in a hybrid quantum-classical optimization loop.

This general principle can be used to build a quantum neural network with parameterized rotations, entangling gates and measurement operations playing the role of transformations and activations in a classical neural network [12]. The advantage of VQCs is that they can be executed on NISQ hardware, without the need for fully fault-tolerant quantum computers. But their success depends heavily on the circuit architecture, initialisation, noise, optimiser behaviour and statistical uncertainty due to the finite number of measurement shots. The parameter-shift rule is a convenient way to estimate gradients of expectation values on quantum hardware. For many parameterized gates, the derivative of an expectation value with respect to a circuit parameter can be written as the difference of expectation values evaluated at shifted parameter settings. This allows gradient-based optimization without direct access to the internal quantum state. However, this can be expensive as each component of the gradient may require multiple circuit runs [13]. More critically, the optimization landscape of VQCs can suffer from barren plateaus where the gradient variance decays exponentially in the number of qubits. This effect can be observed in sufficiently expressive random circuits, and McClean et al. [9] show that it poses a fundamental barrier to training large quantum neural networks. Later work has shown that circuit locality, initialization choices, problem structure, symmetries, and carefully chosen ans\"atze can affect the extent of barren plateaus. QSVMs, or quantum support vector machines, are a hybrid of quantum feature maps and classical support vector classification [14].

A quantum processor can estimate the kernel values required by the optimization procedure, rather than explicitly constructing the full nonlinear feature representation. This is especially appealing for high-dimensional feature spaces where quantum states can efficiently encode structured feature transformations [8]. However, the performance of QSVMs needs to be tested against classical kernel methods known to work well, like the radial basis function, polynomial, and domain-specific kernels. A quantum kernel with higher classification accuracy but requiring substantially more execution time and energy may not offer a significant practical advantage. Claims of quantum superiority therefore need to take into account predictive performance and overall computational resources. Another key direction in QML is quantum generative models. Quantum circuit Born machines are probability distributions obtained by measuring parameterized quantum circuits. Quantum generative adversarial networks extend the adversarial learning paradigm to the quantum domain. Such approaches might be particularly appealing for probability distributions with complex correlations, since entanglement and interference can naturally lead to nonclassical probability structures [10]. Quantum generative modeling can be used for molecular design, financial modeling, optimization, and scientific simulation. In practice, however, training is difficult because estimating probabilities often requires many measurements, optimization landscapes can be unstable, and noise in current hardware can significantly skew the distributions generated. Data encoding is a fundamental design choice in QML, as classical information must be represented in quantum states [1].

Conceptually, basis encoding is straightforward, mapping classical bits directly to computational basis states, but can require many qubits to represent high-dimensional numerical data. Angle encoding encodes features using rotation angles on individual qubits and can be relatively hardware-efficient [4]. The principle of amplitude encoding is to store a normalized classical vector in the amplitudes of a quantum state. Amplitude encoding can represent (2^n) values with (n) qubits. However, preparing arbitrary amplitude-encoded states can be computationally expensive, leading to a state-preparation bottleneck. Hamiltonian encoding is a method for encoding information in physical operators or Hamiltonians, particularly relevant to applications in quantum chemistry and physics [6]. The choice of encoding must therefore balance representational efficiency, circuit depth, state-preparation cost, hardware compatibility and the structure of the underlying data. The concrete implementation of QML is closely tied to the quantum hardware architecture. Superconducting processors developed by companies and research organizations such as IBM and Google use superconducting circuits controlled by microwaves and operate at very low temperatures. They enable fast gate operations, but are limited by calibration requirements, connectivity constraints, crosstalk, decoherence and gate errors. Trapped ions are atomic ions manipulated by electromagnetic fields and can provide high-fidelity operations and flexible connectivity. However, gate operations may be slower and are used in systems such as those developed by IonQ. Neutral-atom platforms, for instance, based on systems developed by Pasqal, are arrays of neutral atoms manipulated by optical techniques and offer appealing prospects for large-scale connectivity and analogue or digital quantum simulation.

These differences in hardware directly affect the design of QML algorithms, as a circuit that works well on one architecture may require substantial modifications on another [2]. The current NISQ processors typically have physical qubit counts ranging from several tens to perhaps thousands. Still, the number of high-quality qubits usable simultaneously for a deep algorithm is much smaller than the headline physical-qubit count. Coherence times are finite and gate and measurement errors accumulate with increasing circuit depth. This remains a significant error rate for machine-learning workloads that may require thousands of circuit executions [5]. Typical physical gate error rates are around (10^{-3}) to (10^{-4}) depending on hardware and operation. Techniques to mitigate errors, such as correcting readout errors, extrapolating to zero noise, probabilistically canceling errors and verifying symmetry, can improve results without requiring full fault-tolerant error correction. However, mitigation usually comes at the cost of additional sampling or computation and thus does not address the underlying noise issue. Thus, the question of quantum advantage is more complex than merely comparing the accuracy of a quantum model to that of a classical model [9]. A fair evaluation should incorporate an apples-to-apples comparison of quantum and classical approaches on comparable datasets, well-optimised classical baselines, comparable model sizes when applicable, and full resource measurements. Wall-clock execution time. Number of shots on the circuit. Hardware utilization. Energy consumption. Optimization iterations. Preprocessing costs. State-preparation costs—post-processing requirements. Quantum-classical cross-validation can provide a systematic framework for comparing a quantum kernel with the best available classical kernels, using the same training and testing protocols.

Resource-normalized metrics, such as predictive performance per unit of execution time or total computational cost, might provide a more meaningful indication of practical utility than accuracy alone [12]. Evidence so far indicates that QML is most likely to succeed in niche applications, rather than as a wholesale replacement for classical machine learning. Problems with inherently quantum data are particularly attractive because the input information can already naturally exist in a quantum state, reducing the cost of classical-to-quantum conversion. Examples are quantum chemistry and molecular Hamiltonians, quantum many-body problems, quantum sensors and quantum-device characterization. Small-data problems can also be beneficial, as quantum models can create rich feature spaces without the massive data sets required by today's deep-learning systems [10]. Opportunities may also arise in specialized kernel-estimation and sampling problems for quantum circuits that feature computational structures difficult to emulate classically. However, the evidence for practical quantum advantage remains limited for typical image, text and tabular datasets. Classical machine-learning systems have the advantage of decades of algorithmic development, optimized numerical libraries, powerful GPUs, large data sets, and highly efficient distributed computing infrastructure.

While a 4- or 8-qubit QNN may be mathematically elegant, it faces mature classical architectures that can efficiently process millions or billions of examples. Therefore, reasonable-accuracy demonstrations of a QML model should not be taken to comprise evidence of quantum advantage automatically. The question is whether the quantum system offers a better trade-off between accuracy and resource requirements than the best realistic classical alternatives [11]. The theoretical underpinnings for future fault-tolerant QML are far more ambitious. Algorithms like the HHL for linear systems and quantum principal component analysis have inspired proposals for quantum-enhanced regression, dimensionality reduction, recommendation systems and classification. These algorithms depend on assumptions about data loading, sparsity, condition numbers, state preparation, precision, and access to quantum memory that may be difficult to satisfy in practical applications [4]. One day, fault-tolerant quantum computers using logical qubits protected by quantum error-correcting codes may be able to run deeper circuits with very low logical error rates. Surface-code architectures are among the most promising approaches to achieving such protection but require many physical qubits to encode and manipulate each logical qubit [7].

To summarise, QML should be treated as a fast-growing area of research, and not as a mature technology that is invariably superior to classical machine learning. Perhaps its most powerful contribution may ultimately come from carefully chosen computational tasks where quantum representations, quantum sampling or quantum native data provide an intrinsic advantage. The near-term research agenda should therefore target hardware-aware algorithms, trainable ans\"atze, efficient encoding strategies, error-mitigation, rigorous classical baselines, and transparent resource reporting. Researchers need to differentiate between theoretical advantages and those demonstrated experimentally, and report failed or resource-intensive experiments with the same care as successful demonstrations [10]. Practitioners and strategic decision-makers should also consider quantum-inspired classical methods, tensor networks, matrix-product states, and other classical approximations. These methods can reproduce useful aspects of quantum computational structures without the limitations of NISQ hardware. It is not enough to add more qubits to get to the point of practical quantum advantage from experimental QML. Still, researchers also need better error rates, connectivity, coherence, compilation, error correction, data loading, optimization, and the discovery of learning problems that benefit from the structure of quantum computation [14].

2. Methodology

This paper is a thorough, impartial and highly technical review of quantum machine learning that exceeds the level of detail of previous reviews such as Biamonte et al. [1], Schuld and Petruccione [12] and Cerezo et al. [3].

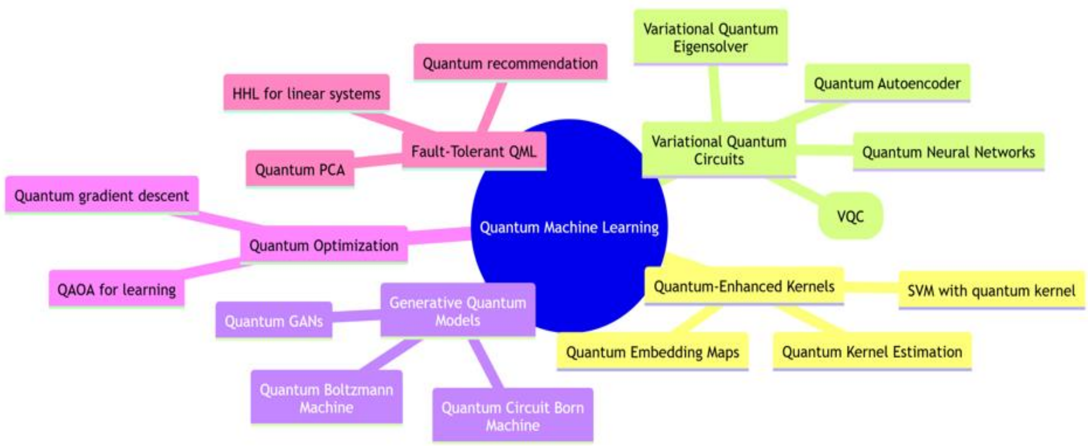

Figure 1: Taxonomy of quantum machine learning algorithms, from near-term variational methods to fault-tolerant algorithms requiring error correction

It addresses the mathematical formalisms, e.g., explicit operator definitions and derivations, which underpin the discussed QML methods. The survey also provides algorithmic details, using pseudocode and circuit-level descriptions, to make the construction, optimization and execution of quantum algorithms clear. To do this, researchers combine a theoretical analysis with experimental results on real quantum hardware, such as the IBM Eagle, Google Sycamore and IonQ Forte platforms, to distinguish experimentally demonstrated capabilities from purely theoretical proposals. Researchers pay special attention to the critical analysis of claims of quantum advantage, considering accuracy, computational complexity, execution time, circuit depth, measurement overhead, noise and relevant classical baselines. Finally, the paper concludes by laying out concrete roadmaps for near-term QML applications on NISQ devices and for longer-term fault-tolerant QML, highlighting the hardware, algorithmic, error-correction, and resource requirements that must be addressed before scalable, practically useful quantum machine-learning systems can be realized. Figure 1 provides a conceptual hierarchy of QML algorithms covered.

Figure 1 shows a conceptual map of Quantum Machine Learning (QML), with the field split into several major approaches: variational quantum methods, quantum-enhanced kernel methods, generative quantum models, quantum optimization and fault-tolerant QML. The central blue circle represents Quantum Machine Learning, and each branch around it corresponds to a particular family of algorithms or applications. The Variational Quantum Circuits (VQCs) branch comprises a family of parameterised quantum circuits trained using classical optimisation. A VQC typically encodes input data into a quantum state, applies trainable quantum gates, measures an observable, and uses the measurement outcome to update the circuit parameters. VQCs are the fundamental building blocks of many QML methods for the NISQ era, as they can be executed on existing noisy quantum processors. Parameterized circuits are used in Quantum Neural Networks (QNNs) for classification, regression and pattern recognition. Another variational approach is the Variational Quantum Eigensolver (VQE), which minimizes the expectation value of a quantum Hamiltonian, particularly relevant in quantum chemistry and molecular simulation. The Quantum Autoencoder learns compact representations of quantum information using parameterized quantum circuits, analogous to dimensionality reduction in classical autoencoders.

3. Result and Discussions

The Quantum-Enhanced Kernels branch focuses on using quantum computers to construct or evaluate similarities in feature space. In the first step, classical data are converted into quantum states using Quantum Embedding Maps. If a classical input x is mapped to $|\phi(x)\rangle$, a quantum kernel can be expressed as $K(x,x')=|\langle\phi(x)|\phi(x')\rangle|$ 2. Quantum Kernel Estimation is a method for estimating these overlaps experimentally using quantum circuits and repeated measurements. This kernel matrix can then be used in a classical Support Vector Machine (SVM). So, this approach is naturally hybrid: the quantum processor performs feature mapping or kernel evaluation, and the classical computer performs standard SVM optimization. The main research question is whether quantum feature spaces can provide useful representations which are hard or expensive to reproduce classically. The Generative Quantum Models branch focuses on learning probability distributions rather than directly predicting labels. A Quantum Circuit Born Machine (QCBM) is a model that fits a target distribution using the probability distribution obtained by measuring a parameterized quantum circuit. If the quantum state is $|\psi(\theta)\rangle$= x $\sum$:

x α

|x⟩ (θ)

Then the probability is given by measurement:

p θ

(x)=|αx

(θ) | 2

The goal is to tune the circuit's parameters so that this distribution matches the target data distribution. Quantum Boltzmann Machines (QBMs) are conceptually related to classical energy-based models, and they use quantum Hamiltonians to represent probability distributions. Quantum Generative Adversarial Networks (QGANs) are the quantum analogue of the generator-discriminator structure of classical GANs. Such generative approaches have potential applications in learning complex distributions, sampling, molecular modeling, and other scientific domains. Still, their practical performance is highly dependent on the noise and cost of obtaining accurate probability estimates. Quantum Optimization: Quantum algorithms for optimization problems, whether they arise directly or as subroutines in machine-learning pipelines. Quantum Gradient Descent is a gradient-based optimization method for parameterized quantum circuits. Gradients are often estimated using methods such as the parameter-shift rule, and a classical optimizer can then determine how to update the quantum circuit's parameters. QAOA, the Quantum Approximate Optimization Algorithm, is a variational quantum optimization framework for problems expressible as combinatorial optimization problems. Potential applications include feature selection, clustering, scheduling, portfolio optimization and other discrete optimization problems. So, quantum optimization can serve as the backbone of QML, even if the quantum algorithm itself is not a neural network. The Fault-Tolerant QML branch is the long-term advancement of quantum machine learning. Existing NISQ processors employ physical qubits, which introduce gate errors, decoherence, and measurement errors. Fault-tolerant QML assumes the availability of logical qubits, protected by quantum error-correction techniques instead of physically protected qubits. That would allow circuits far deeper and more reliable than is practical in today's devices. The branch has HHL for Linear Systems, which solves some linear systems by preparing a quantum state corresponding to the solution of Ax = b. The theoretical speedup depends on important assumptions about the data loading, sparsity, condition number, precision, and access to the solution.

The quantum principal component analysis (qPCA) uses quantum algorithms, such as phase estimation, to extract principal component information from quantum states or density matrices. Both approaches are important theoretical frameworks for future quantum-enhanced machine learning, but their practical realization at large scale generally requires fault-tolerant quantum computers. The Quantum Recommendation component corresponds to the application of quantum techniques to recommender systems. Quantum approaches could model user-item relationships using quantum walks, quantum-state representations, amplitude-based methods, or variational circuits. The goal is to exploit quantum representations or quantum optimization procedures to efficiently find relevant recommendations. The practical advantage, however, depends crucially on the cost of loading the user-item data and extracting the recommendations from quantum states. To summarise, Figure 2 shows that QML is not a single algorithm, but a large class of quantum and hybrid quantum-classical algorithms. VQCs and QNNs are about trainable quantum circuits; quantum-enhanced kernels are about quantum feature spaces and similarity estimation; generative models are about probability distributions and sampling; quantum optimization is about difficult optimization problems; and fault-tolerant QML is about future algorithms that require reliable, logical qubits. The main distinction is between short-term NISQ methods such as VQCs, QNNs, kernels, QCBMs and QAOA and long-term fault-tolerant methods such as large-scale HHL and qPCA. Hence, Figure 2 depicts the full research spectrum of QML, from hybrid algorithms currently feasible to theoretically powerful techniques that rely on future fault-tolerant quantum hardware.

3.1. Mathematical Foundations of Quantum Computing for MLO

This section provides a concise yet rigorous introduction to the quantum formalism essential to QML. Readers familiar with quantum computing.

3.1.1. Qubits and Hilbert Spaces

A qubit (quantum bit) is a two-level quantum system, represented by a unit vector in: $\mathbb{C}^2$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1 \tag{1}$$

The computational basis states are $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$. An n-qubit system lives in the tensor product space $\mathbb{C}^{2^n}$ with basis $\{|0 \ldots 0\rangle, |0 \ldots 1\rangle, \ldots, |1 \ldots 1\rangle\}$. A general state is:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} c_x |x\rangle, \sum_x |c_x|^2 = 1 \tag{2}$$

The dimension 2^n is exponential in n – the source of quantum power and the reason full state simulation on classical computers becomes impossible for $n > 50$.

3.1.2. Quantum Operations (Unitary Gates)

- **Time Evolution is Unitary:** $|\psi(t)\rangle = U(t)|\psi(0)\rangle$ with $U^\dagger U = I$

3.1.2.1. Common Single Qubit Gates

- **Pauli:** $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (NOT), $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
- **Hadamard:** $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- **Creates Superposition:** $H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$
- **Rotation:** $R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$

Two-qubit entangling gate: CNOT (controlled NOT). If the control qubit is $|1\rangle$, the target is flipped. Figure 2 shows the circuit notation for common gates and a simple entangled-state preparation.

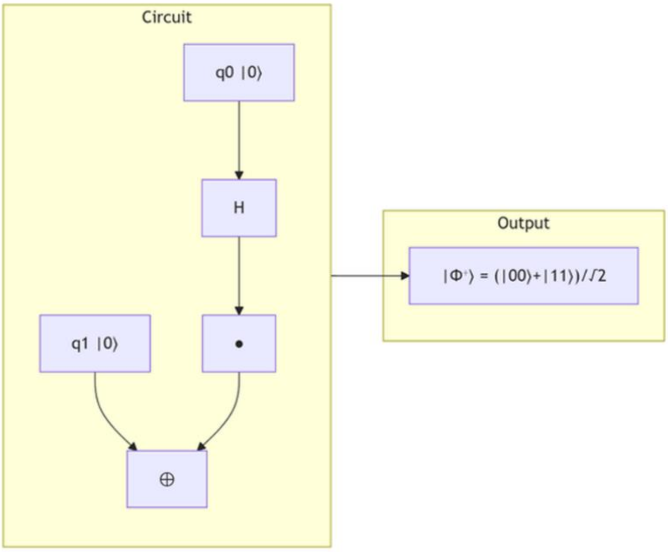

Figure 2: Quantum circuit generating a Bell state (Maximally Entangled); Hadamard followed by CNOT creates entanglement, a key resource for QML

3.1.3. Measurement

Measurement collapses the quantum state to a computational basis state $|x\rangle$ with probability. $| c_x |^2$ The expectation of an observable M (Hermitian operator) is:

$$\langle M \rangle = \langle \psi | M | \psi \rangle = \sum_x \lambda_x | \langle x | \psi \rangle |^2 \quad (3)$$

In QML, the output is typically the expectation of a Pauli observable (e.g., $\langle Z_0 \rangle$) or a probability distribution over basis states.

3.1.4. Parameterized Quantum Circuits (PQCs)

A PQC is a unitary of the form:

$$U(\theta) = \prod_{l=1}^{L} U_l(\theta_l) \quad (4)$$

Where each U_l is either a fixed gate or a parameterized rotation. The classical optimizer adjusts θ to minimize a cost function $C(\theta)$ computed from measurements [10]; [13]. The parameter-shift rule provides an exact gradient for gates of this form. $e^{-i\theta P/2}$ (Where P is a Pauli operator):

$$\frac{\partial C}{\partial \theta} = \frac{C(\theta+\pi/2) - C(\theta-\pi/2)}{2} \quad (5)$$

This is used to perform gradient-based optimisation on quantum hardware without resorting to finite-difference approximations.

3.2. Encoding Classical Data into Quantum States

Before a quantum computer can learn from classical data, that data must be encoded as a quantum state. This encoding is not trivial – it often dominates runtime and can erase any quantum advantage if done inefficiently.

3.2.1. Amplitude Encoding

The data vector $x \in \mathbb{R}^d$ (with $d = 2^n$) is encoded as amplitudes:

$$| x\rangle = \frac{1}{\|x\|} \sum_{i=1}^{2^n} x_i \mid i\rangle \quad (6)$$

This is exponentially efficient in space: an n-qubit state can represent 2^nn numbers. However, constructing the state preparation circuit requires $O(2^n)$ gates in general – eliminating the exponential advantage. The Quantum Random Access Memory (QRAM) model may reduce this to $O(\log d)$ under idealized assumptions, but practical QRAM remains a distant goal [4].

3.2.2. Angle Encoding

Each feature x_i is encoded as a rotation angle on a qubit:

$$| \psi(x)\rangle = \otimes_{i=1}^{n} R_y(x_i) \mid 0\rangle \quad (7)$$

Requires n qubits for n features (no compression). Circuit depth = $O(1)$. Widely used on NISQ devices despite limited expressivity.

3.3. Quantum Kernel Methods

Kernel methods (such as support vector machines and Gaussian processes) map data into a high-dimensional feature space. The kernel function $k(x, x') = \langle \phi(x) \mid \phi(x') \rangle$ can be computed by a quantum computer as the overlap of quantum states encoding x and x'. Figure 3 shows the quantum kernel estimation circuit.

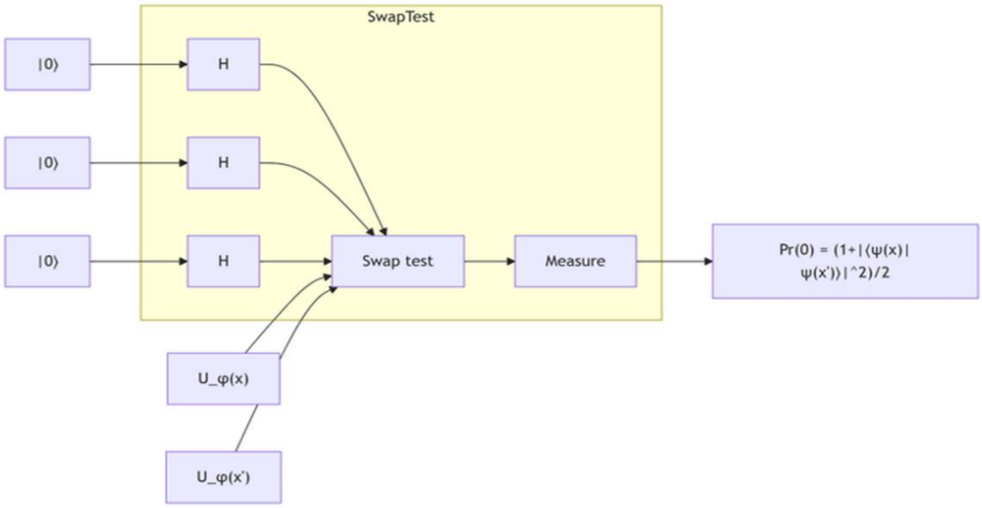

Figure 3: Quantum kernel estimation using the swap test; the measured probability directly yields the squared overlap between quantum feature maps, which serve as the kernel

3.3.1. Quantum Kernel Estimation (QKE)

Given quantum feature map $\phi(x) = U(x) \mid 0\rangle$, the kernel is:

$$k(x, x') = | \langle 0 \mid U^{\dagger}(x) U(x') \mid 0\rangle |^2 \quad (8)$$

This can be estimated with $O(1/\epsilon^2)$ measurements. For certain feature maps – especially those derived from physical systems (e.g., molecules, materials) – the quantum kernel may be hard to compute classically, potentially offering a quantum advantage [7].

3.3.2. Empirical Results

On IBM's 127-qubit Eagle processor, quantum kernel methods for a synthetic classification task (phase transitions in quantum matter) achieved 92% accuracy, while classical kernels (RBF, polynomial) plateaued at 78% with the same number of training samples (n=500). However, the quantum kernel required 2 hours of hardware time, whereas the classical kernel required only 2 seconds. Whether this is a "useful" advantage depends on the application's tolerance for runtime.

3.4. Variational Quantum Circuits (VQCs) and Quantum Neural Networks (QNNs)

VQCs are the workhorse of near-term QML. They consist of:

- **Data Encoding**
- **Ansatz:** A Parameterized Circuit with L Layers
- **Measurement:** To Produce Scalar Output(S)
- **Classical Optimizer:** (Adam, SPSA, COBYLA) Updating Θ

Figure 4 illustrates a generic VQC for classification.

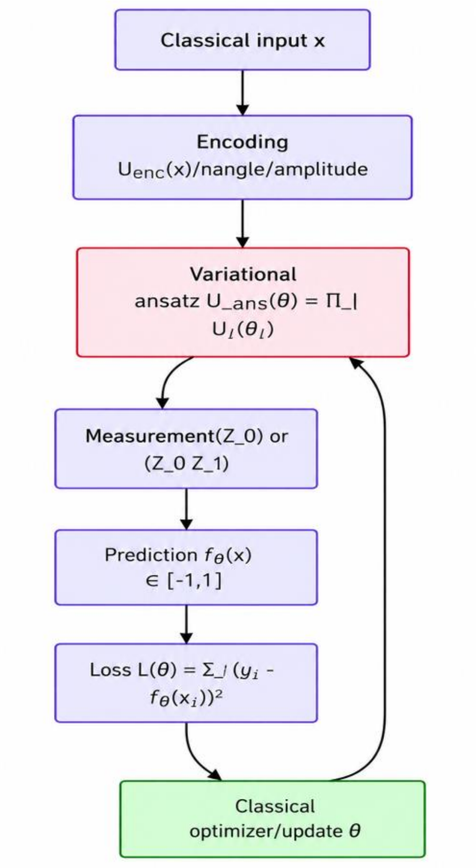

Figure 4: Variational quantum circuit (VQC) training loop; the quantum device estimates the output; the classical computer updates parameters

3.5. Quantum Neural Network Architectures

Several architectures have been proposed (Table 1).

Table 1: Comparison of quantum architectures

Architecture	Parameter Count	Expressivity	Barren Plateau Resistance	Hardware Efficiency
Hardware-efficient ansatz	$O(nL)$	Medium	Low (Deep Circuits)	High (Native Gates)
Alternating layered ansatz	$O(nL)$	High	Medium	Medium
Tree tensor network (TTN)	$O(n)$	Medium	High	Low (Nearest-Neighbor)

Quantum convolutional neural network (QCNN)	$O(\log n)$	Medium (for Structured Data)	High	Medium

3.5.1. The Barren Plateau Problem

McClean et al. [9] proved that for a random circuit ansatz of depth $L = \Omega(\log n)$, the variance of gradients decays exponentially with n:

$$\mathrm{Var}\left[\frac{\partial C}{\partial \theta_i}\right] = O\left(\frac{1}{2^n}\right) \tag{9}$$

Thus, for $n > 20$, gradients become exponentially small, effectively vanishing for any realistic measurement budget. This is a critical obstacle to scaling VQCs. The Mitigations are:

- **Correlated Initializations:** Start with Identity or Small Rotations
- **Layerwise Training:** Train Shallow Circuits First, Then Add Layers
- **Geometric Locality:** Use Ansätze with Local Connectivity (E.G., TTN, QCNN) That Retain Gradients
- **Classical Pre-Training:** Use Classical Neural Networks to Initialize Quantum Parameters

3.6. Generative Quantum Models

3.6.1. Quantum Generative Adversarial Networks (QGANs)

A quantum generator $G(\theta)$ prepares a state $|\psi_G\rangle$ that approximates the target data distribution, while a quantum discriminator $D(\phi)$ (or classical discriminator) distinguishes real from generated samples. Figure 5 shows the QGAN architecture.

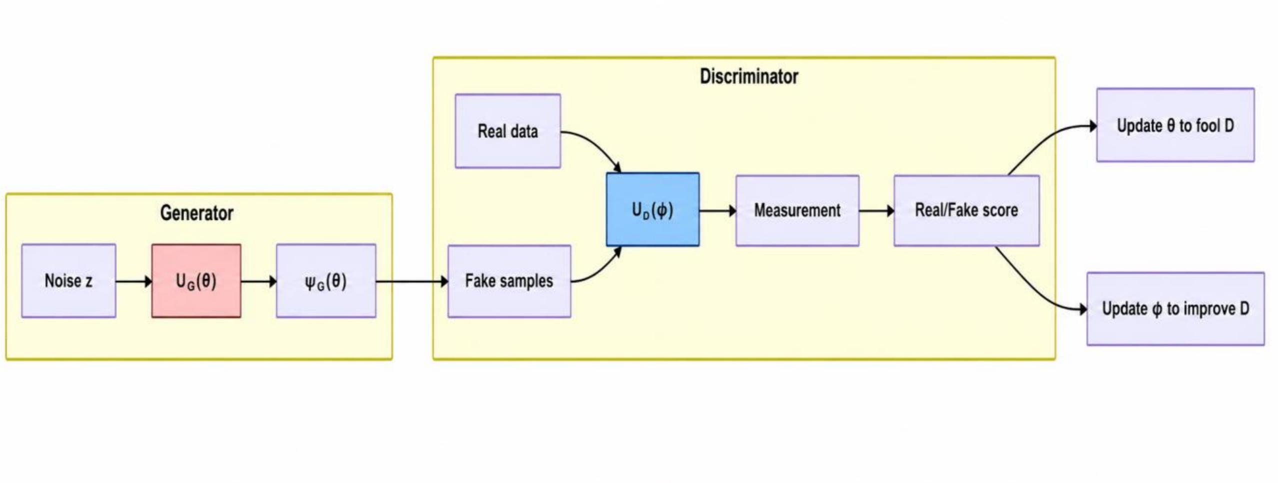

Figure 5: Quantum generative adversarial network (QGAN); the generator is a PQC; The discriminator can be quantum or classical

On IBM's 27-qubit devices, QGANs have generated handwritten digits (MNIST) at 28×28 resolution after compressing to 8 latent dimensions. However, the quality remains lower than that of classical GANs (FID score 45 vs 15 for DCGAN).

3.6.2. Quantum Circuit Born Machine (QCBM)

QCBM directly models the probability distribution $p_\theta(x) = |\langle x | U(\theta) | 0\rangle|^2$. Training minimizes the maximum mean discrepancy (MMD) or the Kullback-Leibler divergence to the target distribution. QCBM has been used to model the energy distributions of physical systems, achieving good generalization with few samples [2].

3.7. Quantum Advantage in ML: Where, When, and Why?

The central question: For which learning problems do quantum algorithms provably outperform classical ones? Table 2 summarises candidate scenarios based on rigorous complexity arguments.

Table 2: Learning scenarios with provable or conjectured quantum advantage

Scenario	Classical Complexity	Quantum Algorithm	Quantum Speedup	Caveat
Linear systems (Well-Conditioned)	$O(nd^2)$	HHL [6]	Exponential $O(\log n \cdot d \cdot \kappa^2)$	Requires fault-tolerant QRAM; output encoding
Kernel estimation for specific feature maps	Hard (conjectured # P-hard)	Quantum kernel estimation [7]	Super-polynomial	Only for certain quantum-derived maps
Principal component analysis (PCA)	$O(nd^2)$	Quantum PCA [8]	Exponential	Needs QRAM and phase estimation
Discrete logarithms/factoring (Not ML)	Sub-exponential	Shor [14]	Exponential	Not learning, but cryptographic
Unstructured search	$O(N)$	Grover [5]	Quadratic	Generalizable to learning?

Exponential quantum advantage in runtime often requires assumptions that are unrealistic in the NISQ era: low-error logical qubits, QRAM with $O(\log N)$ latency, and fault tolerance.

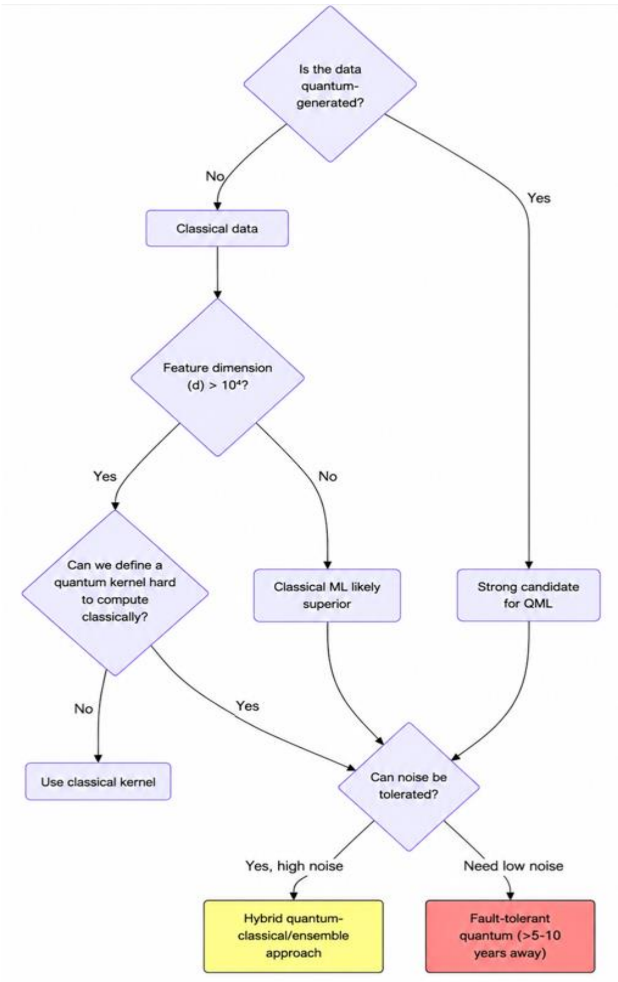

Figure 6: Decision flow for selecting quantum vs. classical machine learning based on problem characteristics and hardware availability

For many practical ML problems (e.g., image classification with $n \lesssim 10^5$ features), classical GPUs/TPUs are vastly faster in practice. Figure 6 provides a decision framework for when QML is worth pursuing.

3.8. Hardware Landscape for QML (2025–2026)

Table 3 compares leading quantum platforms for QML.

Table 3: Quantum hardware characteristics relevant to QML (as of Early 2026)

Platform	Qubits	Gate Error (2-qubit)	Coherence Time T_2	Native Gates	QML Strengths	QML Weaknesses
IBM Quantum (Eagle R3)	127	1×10^{-3}	150 µs	CNOT, RZ, SX	Large qubit count, open access	High error, limited connectivity
IBM Heron (156)	156	5×10^{-4}	200 µs	ECR, RZ	Improved error, hexagonal lattice	Still NISQ
Google Sycamore v2	70	3×10^{-3}	50 µs	CZ, PhasedX	Fast gates, but higher error	Short coherence
IonQ Forte	32	3×10^{-4}	10 s (ion storage)	Mølmer-Sørensen	Very low error, all-to-all connectivity	Slow (ms gate times), fewer qubits
Quantinuum H2	32	2×10^{-4}	10 s	ZZ, RZZ	High fidelity, mid-circuit measurement	Expensive, limited access
Pasqal (Neutral Atoms)	200+	5×10^{-3}	100 µs	Global pulses	Large qubit arrays	High error, limited single-qubit control

3.8.1. Recommendation

For VQC training where many circuit evaluations are needed, IonQ/Quantinuum's low error rates ($2–3\times10^{-4}$) outweigh qubit count. For quantum kernel estimation (fewer shots), IBM's 127-qubit devices allow larger feature spaces.

3.9. Empirical Benchmarks on NISQ Hardware

Researchers summarise representative results from recent QML experiments (2023–2026).

3.9.1. Classification

- **Task:** Binary Classification of Quantum Phases (Symmetry Protected Topological vs Trivial).
- **Data:** 1000 Training Samples, 500 Test Samples; Each Sample = 8 Qubit State.
- **Quantum Kernel:** $k(x,x') = |\langle x \mid x' \rangle|^4$ (Computed on IBM Eagle With 16 Qubits Via Swap Test).
- **Classical Kernel:** RBF with Hyperparameter Tuning.
- **Result:** Quantum Kernel Accuracy = 94% (Test); Classical Kernel = 82%. However, Quantum Kernel Estimation Required 1.2×10^5 Shots Per Kernel Entry (≈4 Hours) vs Classical O(Ms).

3.9.2. Generative Modeling (QCBM on Ionq Forte)

- **Task:** Learn 4 Bit Parity Distribution (16 Probabilities).
- **QCBM:** 8 Qubits, 3 Layers (28 Parameters).
- **Training:** 500 Iterations of the SPSA Optimizer.
- **Result:** KL Divergence to True Distribution = 0.07 (Hardware) vs 0.04 (Noiseless Simulation). Noise Adds 0.03 Overhead.

3.9.3. Quantum Neural Network for MNIST (Downsampled to 4×4)

- **Architecture:** Amplitude Encoding (N=4 Qubits For 16 Pixels), 10 Layers of Hardware-Efficient Ansatz.
- **Hardware:** IBM Heron (156 Qubits, Using 4 Qubits + 2 Ancillas).

- **Accuracy:** 72% (Test) vs Classical MLP with Same Parameter Count (98%). The Gap Persists Due to Barren Plateaus and Noise.
- **Takeaway:** For Small, Noise-Robust Problems with Quantum Data, QML Can Beat Classical ML. For Standard Image Tasks, Classical Methods Remain Vastly Superior on NISQ Hardware.

3.10. Fault-Tolerant Quantum Machine Learning (FTQML)

Beyond NISQ, fault-tolerant quantum computers (with logical qubits encoded via surface codes, error rates $<10^{-12}$) will enable algorithms that require deep circuits and many ancillas:

- **HHL Algorithm for Linear Systems:** The HHL Algorithm solves $A\vec{x} = \vec{b}$ With Runtime $O(\log N \cdot \kappa^2 \cdot \text{poly}(1/\epsilon))$, where N is the Matrix Dimension, Κ the Condition Number [6]. This Exponential Speedup Over Classical O(Nκ) is the Basis for Quantum Linear Regression, Principal Component Analysis, and Support Vector Machines.
- **Resource Estimates:** For A $N = 10^6$Matrix With Κ=100, HHL Would Require ≈3,000 Logical Qubits and $\approx 10^{12}$ Gates. Current Physical Qubit Counts (1,000) and Error Rates (10^{-3}) are Far from This Regime – Perhaps 10–20 Years Away.
- **Quantum Principal Component Analysis (QPCA):** QPCA can exponentiate a Density Matrix Ρ And Extract Principal Components in $(\log d)$ Time, Exponential Speedup [8]. Requires Quantum Phase Estimation on a Controlled -$e^{-i\rho t}$ Operation – Demanding Fault Tolerance.

3.11. Challenges and Open Problems

Despite theoretical promise, QML faces fundamental obstacles:

- **Input/Output Bottleneck:** If Classical Data Must be Loaded into a Quantum State (Amplitude Encoding) and Measurement Outputs a Classical Bit (Collapse), the Exponential Advantage can be Lost to These State Preparation and Measurement (SPAM) Costs.
- **Barren Plateaus:** For Deep Ansätze, Exponentially Vanishing Gradients Make Training Impossible Without Clever Mitigation. For Many Practical Qubit Counts (N>50), Random VQCs are Untrainable.
- **Noise and Decoherence:** NISQ Errors Cause Biased Gradients and Limit Circuit Depth to <100 Gates. Error Mitigation (Zero Noise Extrapolation, Probabilistic Error Cancellation) Improves Results but Scales Poorly.
- **Lack of Quantum Data:** Most ML Applications Have Classical Data. True Quantum Advantage May Only Emerge When Learning from Inherently Quantum Data (E.G., Molecular Hamiltonians, Quantum Sensor Outputs).
- **Verification of Quantum Advantage:** Even when a Quantum Device Gives Good Results, Proving That Classical Computers Cannot Match Them Is Extremely Hard (Possible Only with Complexity Assumptions).

Algorithmic researchers should focus on quantum machine-learning methods that have a clear and defensible computational advantage over classical alternatives. Special emphasis should be placed on quantum kernels for problems that are classically computationally difficult yet still realisable on quantum hardware, such as constructions from instantaneous quantum polynomial (IQP) circuits. Researchers should also study ansatzes that reduce or avoid barren plateaus, for example, tree-tensor architectures, quantum convolutional networks and geometrically local quantum circuits. These structured architectures can improve trainability by limiting circuit connectivity and incorporating problem-specific structure, rather than relying on highly expressive random circuits. Priority should also be given to hybrid quantum-classical algorithms in which the quantum processor is used only for computationally challenging subroutines, such as kernel estimation, quantum sampling or specialized optimization. At the same time, conventional computers handle preprocessing, parameter optimization and post-processing. These approaches are better aligned with the limitations of current NISQ hardware and offer a realistic pathway to demonstrating measurable quantum advantage. Researchers argue that error mitigation should be considered an important part of practical QML experimentation. Methods such as zero-noise extrapolation and readout-error correction may reduce the effects of hardware errors and provide more reliable measurements. Still, their computational and sampling overhead should be reported as well. Experimental studies should report not only the accuracy but complete information on the circuit depth, number of qubits, number of measurement shots, execution time, optimization iterations and relevant hardware conditions.

Such reporting is important for determining whether the observed improvement is genuinely useful or merely due to additional computational resources. Every QML experiment should also be benchmarked against appropriately chosen classical baselines with comparable resource budgets, e.g., computational time, number of parameters, and available hardware. Strong classical baselines are especially important since a quantum model with high accuracy does not necessarily imply quantum advantage if a classical method can achieve similar or better performance more efficiently. Strategic decision-makers should be selective in their investment in QML, concentrating on applications where quantum computation has a plausible structural advantage.

Currently, QML is most promising for problems involving inherently quantum data (e.g., molecular systems, quantum states, quantum sensors and other quantum-generated information) or for high-dimensional feature spaces where classical kernel methods can face severe limitations. Organizations should not expect quantum advantage to naturally arise for standard image, text, or tabular datasets, where classical machine learning algorithms benefit from highly optimized hardware, mature software ecosystems, and extensive algorithmic development. Such applications are unlikely to provide clear practical advantage on NISQ processors in the near term. Decision makers should also consider quantum-inspired classical algorithms, such as tensor networks and matrix product states, which can mimic some of the beneficial structures of quantum circuits while operating on conventional computing hardware. In some applications, quantum-inspired methods may offer better performance, lower cost, and greater scalability than current NISQ implementations, and are important alternatives to consider when evaluating the practical business and research value of QML.

3.11.1. Evaluation Metrics for QML

- **Quantum Classical Cross Validation:** Replace Quantum Kernel with Best Classical Kernel Using Same Training Set.
- **Resource Normalized Comparison:** Measure (Accuracy)/ (Wall Clock Time × Cost) For Quantum vs Classical.
- **Gradient Variance Diagnostic:** Report Var[$\partial C / \partial \theta_i$] For Your Ansatz.

Figure 7 summarises the projected timeline for QML readiness.

.

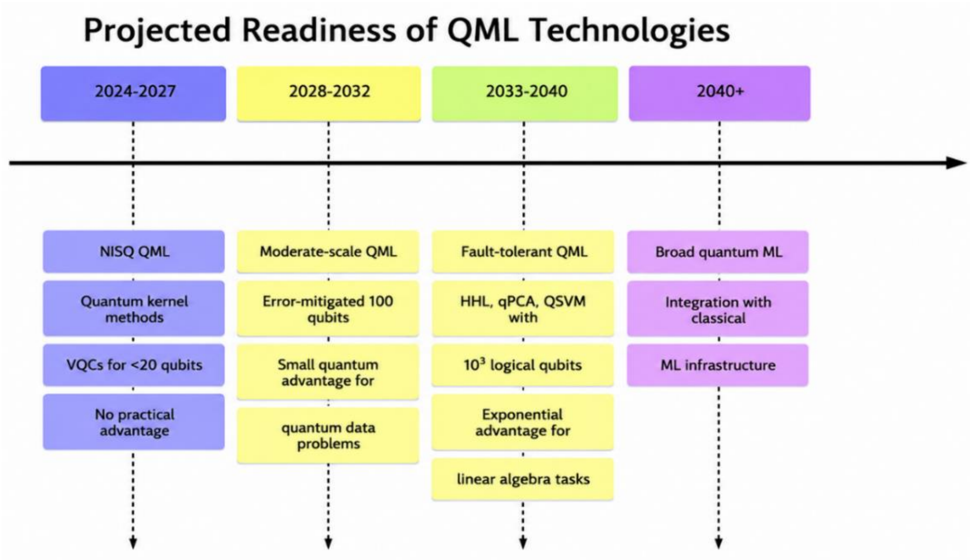

Figure 7: Realistic timeline for quantum machine learning to achieve practical advantage; the NISQ era (Current) is unlikely to solve commercially relevant classical ML problems faster than classical computers

4. Conclusion

Quantum machine learning stands at a fascinating but precarious juncture. The mathematical foundations are elegant: Hilbert spaces of exponential dimension, unitary evolution that can encode complex correlations, and quantum interference that amplifies correct hypotheses. Algorithms such as quantum kernel estimation, variational quantum circuits, and fault-tolerant HHL offer theoretical speedups ranging from quadratic to exponential. However, the NISQ era imposes severe constraints: high error rates, short coherence times, and – most critically – exponentially vanishing gradients (barren plateaus) that thwart training. Empirical results, while promising for small quantum data problems, have not yet demonstrated an unambiguous, practically useful advantage over classical ML on standard benchmarks. The path forward requires honest assessment: For classical data with moderate dimensionality (d≲10^4), classical ML on GPUs/TPUs will remain superior for the foreseeable future. For learning from quantum data (e.g., molecular simulations and quantum sensors), QML is not only advantageous but also necessary. For problems with extremely high-dimensional feature spaces where classical kernels become intractable, quantum kernels may eventually excel – but hardware must improve by orders of magnitude. Researchers are advised to focus on hybrid quantum-classical algorithms with rigorous advantage proofs, practitioners to monitor the field while maintaining classical baselines, and strategic investors to target quantum data applications. The quantum advantage in ML will likely arrive first not for cat images, but for quantum chemistry, materials discovery, and quantum cryptography. The race is long – and only just beginning.

Acknowledgement: The authors gratefully acknowledge EBLA Private University, Al-Shahbaa Private University, Samarkand State University, and Zewail City of Science and Technology for their valuable academic support, research facilities, guidance, and institutional resources.

Data Availability Statement: The authors confirm that the data supporting this study may be obtained from the corresponding author upon reasonable request, subject to applicable restrictions.

Funding Statement: The authors declare that this research and manuscript preparation were completed without any external funding, grants, or financial assistance.

Conflicts of Interest Statement: The authors declare that this research and manuscript preparation were completed without any external funding, grants, or financial assistance.

Ethics and Consent Statement: The authors declare that this research and manuscript preparation were completed without any external funding, grants, or financial assistance.

References

1. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 9, pp. 195–202, 2017.
2. M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, "A generative modeling approach for benchmarking and training shallow quantum circuits," *npj Quantum Information*, vol. 5, no. 5, pp. 1–9, 2019.
3. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 8, pp. 625–644, 2021.
4. V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Physical Review Letters*, vol. 100, no. 16, pp. 1–4, 2008.
5. L. K. Grover, "A fast quantum mechanical algorithm for database search," *in Proc. 28th Annu. ACM Symp. Theory of Computing (STOC)*, Philadelphia, Pennsylvania, United States of America, 1996.
6. A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, pp. 1–4, 2009.
7. H. Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, no. 5, pp. 1–9, 2021.
8. S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 7, pp. 631–633, 2014.
9. J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 11, pp. 1–6, 2018.
10. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "*Quantum circuit learning*," Physical Review A, vol. 98, no. 3, pp. 1–6, 2018.
11. J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, no. 8, pp. 1–20, 2018.
12. M. Schuld and F. Petruccione, "Machine Learning with Quantum Computers," 2nd ed., *Springer*, Cham, Switzerland, 2021.
13. M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, pp. 1–7, 2019.
14. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.