Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.082712

ARTICLE

Hybrid Classical-Quantum Transfer Learning with Noisy Quantum Circuits

Daniel Martín-Pérez[1], Francesc Rodríguez-Díaz[1], David Gutiérrez-Avilés[2], Alicia Troncoso[1] and Francisco Martínez-Álvarez[1,*]

[1]Data Science and Big Data Lab, Pablo de Olavide University, Seville, Spain
[2]Department of Computer Science, University of Seville, Seville, Spain

*Corresponding Author: Francisco Martínez-Álvarez. Email: fmaralv@upo.es

Received: 21 March 2026; Accepted: 22 June 2026; Published: 27 July 2026

ABSTRACT: Hybrid classical-quantum architectures have emerged as a practical response to the data and compute demands of modern deep learning, since a pretrained classical backbone can carry the feature extraction while a compact quantum head provides the trainable component. Quantum transfer learning is the most active instance of this idea. However, existing quantum transfer learning pipelines have been evaluated in isolation, typically on a single software framework and without a structured treatment of noise or statistical significance, which makes it difficult to assess how this paradigm contributes over fair classical baselines. A methodological benchmark for quantum transfer learning is proposed in this paper. The benchmark couples a common set of frozen pretrained convolutional backbones to several classical and quantum classification heads, implemented in both PennyLane and Qiskit, so that the contribution of nonlinearity, software framework, and quantum component can be analyzed separately. The benchmark has been applied to heterogeneous image datasets covering medical, biological, industrial, and general-vision domains, and most configurations have been executed in three environments: ideal simulation, noisy emulation calibrated on IBM Heron r2 and real IBM quantum hardware. Complementary analyses isolate the contribution of individual noise channels, examine scalability with qubit count and circuit depth and assess the presence of barren plateaus. Statistical reliability is ensured through multiple random seeds and pairwise Wilcoxon signed-rank tests, providing the first controlled cross-framework assessment of quantum transfer learning under calibrated noise and on real hardware.

KEYWORDS: Quantum computing; transfer learning; hybrid neural networks; variational circuits

1 Introduction

Artificial intelligence (AI) and machine learning (ML) have moved into most application areas during the last decade, with a clear impact in healthcare, industry, finance, and environmental management. Much of this progress has been driven by deep learning [1], and in particular by convolutional neural networks (CNNs), which now set the standard in image recognition, natural language processing (NLP), and other complex decision-making tasks. Their effectiveness, however, comes at a price. Large annotated datasets are required for training, computational demands keep growing, and the energy footprint of both training and deployment has become a growing concern [2]. These limits have motivated the search for alternative learning paradigms that can deliver competitive performance with smaller data and lower computational budgets.

Transfer learning is one of the strategies that have helped to mitigate these limits. The basic idea is well known: knowledge captured by models trained on large-scale datasets is reused on smaller target tasks, which keeps both data needs and computational cost under control [3]. In the purely classical case, denoted

Classical-Classical (CC) transfer learning, a pretrained neural network is fine-tuned on a smaller target dataset, and this scheme has become standard practice in many application areas. Even so, fine-tuning architectures with millions of parameters remains expensive in compute and energy, especially when the same backbone is reused across many downstream tasks [4].

In parallel, quantum machine learning (QML) [5] has become an active research direction with the potential to reshape parts of the machine learning workflow. Quantum-mechanical phenomena such as superposition and entanglement open the door to computational advantages on specific problem classes. Current hardware, however, is still in the noisy intermediate-scale quantum (NISQ) [6] regime, with limited qubit counts, short coherence times, and shallow trainable circuits. These constraints sharply restrict what fully quantum algorithms can do on realistic machine learning problems today [7].

A practical answer to these limits is to combine the two paradigms. Hybrid classical-quantum (CQ) approaches keep the heavy lifting on the classical side and use the quantum component only where it is most effective. Quantum Transfer Learning (QTL), and more specifically CQ transfer learning, is one of the most active instances of this idea. In a QTL pipeline, the classical part is a pretrained neural network used as a fixed feature extractor, usually called a backbone, and the quantum part is a compact variational circuit that plays the role of a trainable classification or regression head. The classical backbone reduces the input to a low-dimensional representation that already concentrates the relevant information, so the quantum head can stay small. This is what makes the pipeline compatible with current NISQ hardware in the first place.

Despite the growing interest in QTL, the existing studies do not yet form a systematic and controlled benchmark of hybrid classical-quantum transfer learning (CQ TL) pipelines. Each contribution typically evaluates a small number of architectures or datasets, relies on a single quantum software framework, and rarely treats noise or statistical significance as primary evaluation criteria. As a result, the contribution of each component of the pipeline (the classical backbone, the choice of software framework, the noise regime, and the quantum head itself) cannot be cleanly separated, and no general conclusions can be drawn about how QTL actually behaves under realistic experimental conditions.

This gap is precisely what the present study aims to fill, through a methodological benchmark that covers heterogeneous image classification tasks across medical, biological, industrial, and general-vision domains. Within the benchmark, classical and quantum classification heads are evaluated under identical conditions, both in PennyLane [8] and Qiskit [9], on top of pretrained convolutional backbones whose weights are kept frozen. Each configuration is executed in three environments (ideal simulation, noisy emulation calibrated on IBM Heron r2, and real IBM quantum hardware) and is assessed under statistical tests that support the resulting comparisons.

The contributions of this study can be summarized as follows:

1. This paper proposes a methodological framework for transfer learning between classical and quantum systems, which addresses the main gaps in the literature on QTL.
2. A controlled comparison protocol is established in which a common set of frozen basic structures is connected to various classical and quantum modules (so that nonlinearity, the software framework, and the quantum components can be analyzed separately) and evaluated using a statistical evaluation test with multiple random seeds and a set of metrics.
3. Configurations executed in three environments (ideal, noisy and real quantum hardware), and the role of individual noise channels is isolated through a dedicated ablation study.

Furthermore, the benchmark is organized around three research questions (RQ):

RQ_1 Does the choice of quantum software framework (PennyLane or Qiskit) affect the predictive performance of QTL pipelines when both are evaluated under matched experimental conditions?

RQ_2 Under which conditions do variational quantum heads match or surpass classical baselines of matched parameter count?

RQ_3 How does each individual noise channel (amplitude damping, phase damping, and depolarizing) contribute to the degradation observed under realistic noise conditions?

The rest of the paper is organized as follows. Section 2 surveys related work on transfer learning and hybrid quantum classifiers. Section 3 introduces the foundational concepts underlying the study. Section 4 presents the datasets, the experimental setup, the results, and their discussion. Section 5 closes the paper and outlines directions for future research.

2 Related Work

Existing proposals on QTL for image classification are reviewed below organized around three axes: the design of the quantum circuit that operates as the trainable head (Section 2.1), the application domains to which the dressed-circuit template has been ported (Section 2.2), and the strategies adopted to handle noise together with the trainability limits it imposes (Section 2.3). Finally, Section 2.4 provides a critical synthesis of the evaluation methodology, summarizes the coverage of the field in a comparative table, and states the gaps that the present study addresses.

2.1 Quantum Circuit Design

The dressed quantum circuit template was introduced in [10] and has shaped almost every subsequent proposal. A variational quantum circuit (VQC) is placed between two thin classical layers, one that compresses the backbone features into a small qubit register and another that maps the quantum measurements back to the class space, while the pretrained backbone is kept frozen so that only the quantum circuit and its two classical wrappers are trained. The proof of concept paired a ResNet18 backbone with the dressed head on the Hymenoptera and CIFAR datasets through PennyLane.

Most descendants keep this head and modify it only marginally. The template was extended in [11] to waste classification, tuberculosis radiographs, and concrete cracks, and adapted in [12] to high-dimensional remote-sensing images with a VGG16 backbone and strongly entangling layers measured through the local effective dimension. Three contributions depart more substantially: the post-variational scheme of [13] replaces circuit optimization with classical combinations of fixed-ansatz observables, adversarial training is built into a frozen-backbone hybrid in [14], and a hierarchical Quantum Convolutional Neural Network (QCNN) is used instead of a dressed circuit in [15], the only proposal of this group that combines a multi-ansatz classical-to-classical baseline with formal statistical testing.

The expressive capacity behind these circuits has been studied along two complementary lines. The effective-dimension framework of [16] measures model capacity from the geometry of the parameter space and reports that well-designed quantum networks can reach a higher effective dimension and train faster than parameter-matched classical ones. The quantum-kernel perspective interprets a parameterized circuit as a feature map into an exponentially large Hilbert space in [17], while [18] shows how the associated kernel can be estimated on hardware and used for supervised classification. Both views build on the quantum circuit learning paradigm of [19], which formalizes the training of parameterized circuits by gradient descent on near-term devices.

2.2 Application Domains

Medical imaging has been by far the most active application area. Tight circuits are applied to malignant-vs.-benign mammography on the Breast Cancer Digital Repository in [20] and to COVID-19 detection

from computed tomography scans in [21], the latter on three real IBM processors and three simulator backends. Diabetic retinopathy is addressed twice: with a ResNet-50 and a four-qubit classifier in [22], the only retinopathy study to report a Wilcoxon signed-rank test against three classical baselines, and with an Inception-V3 backbone in [23], which runs the same model across the Asia Pacific Tele-Ophthalmology Society 2019 fundus images using PennyLane, Qiskit, and Cirq simulators.

Beyond medical imaging, the template has been ported to a wide range of modalities. Industrial inspection is covered by seven-class waste sorting on the released TrashBox dataset in [24] and by concrete crack detection in [25]. A Class-Attention Vision Transformer is plugged into a QTL stage for deepfake detection on the DeepFake Detection Challenge corpus in [26], audio is addressed for spoken command recognition in [27] and for synthetic-speech detection on the Automatic Speaker Verification Spoofing 2021 corpus with a WavLM-Large model in [28], human pose recognition over 60 GHz Wi-Fi channels is tackled in [29], and BERT embeddings are combined with a quantum head for Italian linguistic acceptability on the Italian Corpus of Linguistic Acceptability in [30]. Two further references lie outside CQ TL for image classification but illustrate broader uses of transfer learning in quantum contexts: between quantum-chemistry computations and experimental measurements [31], and between classical neural-network quantum state representations [32]. Despite this variety of domains and modalities, the experimental protocol shared by these proposals is remarkably uniform, an aspect analyzed in Section 2.4.

2.3 Noise Handling Strategies

The theoretical picture of noisy variational training is well established. Local Pauli noise has been shown in [33] to induce barren plateaus whose gradients vanish exponentially in the qubit count as the ansatz depth grows linearly, a result that complements the ideal simulator analysis of [34] and the cost-function-dependent variant of [35].

The empirical side is much thinner. Real-hardware runs in [21] and [25] both report accuracy degradation, but neither isolates the contribution of amplitude damping, phase damping, and depolarizing noise, and [14] considers adversarial perturbations rather than physical noise. No CQ TL pipeline has performed a structured channel-by-channel ablation on hardware-calibrated profiles, which leaves open the question of which physical channel dominates the degradation of a transfer-learned quantum head.

2.4 Research Gaps and Positioning

In the proposals reviewed above, the experimental protocol is markedly uniform, and this uniformity is, in the view of the present study, the main weakness of the field. Almost all heads are built on PennyLane or Qiskit, and the four-to-eight qubit register together with the noise-agnostic, single-framework evaluation has barely changed across the field. Accurate reporting is essential. Indicator dashboards rarely include a complete set of metrics, such as precision, recall, F1 score, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC). In addition, the classical baseline is usually a single linear classifier on the same backbone, while parameter-matched multilayer perceptrons of comparable capacity are rarely included. Formal statistical testing is equally rare: it appears in only one of the reviewed studies, whereas the standard recommendations of [36], Wilcoxon for two classifiers and Friedman with post-hoc tests for several, are widely accepted, but rarely put into practice. Controlled comparisons between software stacks are scarce as well: the multi-simulator runs of the COVID and retinopathy studies report no formal cross-framework testing, the twelve-model benchmark of [37] (six binary tasks comprising 160 datasets, all in PennyLane) finds that classical baselines often outperform quantum ones, and the hybrid CNN study of [38] stays within a single framework. As a result, no controlled PennyLane-vs.-Qiskit comparison has been reported for CQ TL under real hardware. Sensitivity analysis over qubit count and circuit depth inside CQ TL pipelines has not been brought together in any single contribution either.

Table 1 summarizes the works reviewed and compares them with our proposal and concludes that no prior contribution jointly addresses the experimental dimensions covered by the present study. The most complete proposals reach at most four or five of the fifteen dimensions, leaving channel-by-channel noise decomposition, parameter-matched classical baselines, and a fully controlled PennyLane-vs.-Qiskit comparison under hardware-calibrated noise as open gaps. The design choices of the present study (head taxonomy, noise modeling, evaluation protocol, and measurement instrumentation) are positioned around these gaps, so that each one maps onto one or more of the dimensions in the last row of the table.

Table 1: Coverage of fifteen experimental dimensions by the seventeen reviewed QTL studies and by the present study (last row). Symbols: ✓ explicit treatment, (✓) motivational discussion without supporting experiments, ~ partial, × absent, N/R not reported. Column codes: *Backbone* (classical backbone; RN = ResNet, VGG = VGG-family, DN = DenseNet, Linear = Linear baseline, Mix = several); *Framework* (Quantum framework; PL = PennyLane, QK = Qiskit, Cirq = Cirq, SF = Strawberry Fields, TFQ = TensorFlow Quantum); *Qubits* (number of qubits); *CD* (circuit depth); *Real HW* (execution on real quantum hardware); *Model* (noise model; HW = real hardware only, Gen = generic, Cal = hardware-calibrated); *CH-CH* (channel-by-channel ablation); *Seeds* (multiple seeds reported); *SST* (statistical significance tests); *Metric* (metric panel; 1 = accuracy only, 2 = accuracy plus one or two metrics, 3 = three or more metrics, Full = complete panel including precision, recall, F1, ROC-AUC); *MF* (controlled multi-framework comparison); *PM* (parameter-matched classical baselines); *Abl* (qubit and/or depth ablation); *BP* (explicit barren-plateau analysis).

Study	Setup					Noise		Evaluation			Methodology			
	Backbone	**Framework**	**Qubits**	**CD**	**Real HW**	**Model**	**CH-CH**	**Seeds**	**SST**	**Metric**	**MF**	**PM**	**Abl**	**BP**
[10]	RN18	PL/SF	4	6	✓	HW	×	N/R	×	1	~	×	×	×
[11]	Mix	PL	4	2-6	×	—	×	N/R	×	1	×	×	×	×
[12]	VGG16	PL	3	3-6-9	×	—	×	N/R	×	2	×	×	✓	×
[13]	Mix	PL/QK	4-8	5-6-7	×	—	×	✓	×	2	~	×	✓	(✓)
[14]	RN18	PL	4	6	×	—	×	N/R	×	2	×	×	~	×
[15]	CNN	PL	8	3	×	—	×	✓	✓	1	×	×	✓	(✓)
[20]	Mix	PL	4	1-6	×	—	×	N/R	×	Full	×	×	✓	×
[21]	RN18	PL/QK/Cirq	4	N/R	✓	Gen	×	N/R	×	Full	~	×	×	×
[22]	RN50	Cirq	4	N/R	×	—	×	N/R	✓	3	×	×	×	×
[23]	IncV3	PL/QK/Cirq	4	6	×	—	×	N/R	×	3	~	×	×	×
[24]	Mix	PL	4	N/R	×	—	×	N/R	×	2	×	×	×	×
[25]	Mix	PL/QK	4	1-6	✓	Cal	×	✓	×	2	~	×	✓	×
[26]	CaiT	PL/QK/Cirq	4	10	×	—	×	×	×	Full	~	×	×	×
[27]	1D-CNN	PL	N/R	N/R	×	—	×	×	×	1	×	×	×	×
[28]	WavLM	PL	N/R	N/R	×	—	×	N/R	×	2	×	×	×	×
[29]	Linear	PL	8	1	×	—	×	×	×	1	×	×	×	×
[30]	BERT	PL	10	6-8	×	—	×	N/R	×	2	×	×	×	✓
Ours	Mix	PL/QK	4-6	1-5	✓	Cal	✓	✓	✓	Full	✓	✓	✓	✓

3 Methodology

This paper compares five pre-trained classical base models and different transfer learning heads: three classical baseline models and two quantum hybrid variants implemented using PennyLane and Qiskit. The goal is not to demonstrate quantum advantage, but to evaluate whether compact quantum classifiers can offer effective and efficient alternatives to classical heads under realistic NISQ constraints.

Classical models use frozen, pretrained convolutional backbones with trainable classifier heads, including a linear baseline, a parameter-matched shallow multilayer perceptron (MLP), which we will refer to as MLP-SM, and a higher-capacity MLP, which we call MLP-LG. The quantum-hybrid models use four-qubit, three-layer variational circuits fed by the same backbone features and evaluated under both ideal and noisy

conditions. In both PennyLane and Qiskit, ring CNOT entanglement is used, SPSA optimization is applied in all scenarios, and the convolutional backbones remain frozen.

All experiments are repeated with five random seeds, reporting the mean and standard deviation to ensure a fair comparison between the different architectures. The following subsections describe the classic transfer learning reference models (Section 3.1) and the PennyLane and Qiskit quantum hybrid configurations in ideal and noisy execution environments (Sections 3.2–3.5).

3.1 Classical-Classical Transfer Learning

Our classical baseline employs transfer learning [39,40] using CNNs pretrained on ImageNet-1k [41,42]. We evaluate pretrained architectures that vary in design, parameter efficiency, and feature extraction capabilities.

In the training, all backbone weights remain frozen, whereas only the final classification head is optimized. To isolate quantum benefits, we implement three classical baseline heads. First, a linear classifier maps extracted representations directly to class logits as defined in Eq. (1).

$$f_{\text{linear}}(\mathbf{x}) = \mathbf{W}^{\top}\phi(\mathbf{x}) + \mathbf{b} \tag{1}$$

Second, MLP-SM model maps the representations through a low-dimensional hidden layer of dimension $h = 4$ using the Hardtanh activation function, a bounded clipping function, as defined in Eq. (2).

$$f_{\text{MLP-A}}(\mathbf{x}) = \mathbf{W}_{\text{post}}\text{Hardtanh}(\mathbf{W}_{\text{pre}}\phi(\mathbf{x})) \tag{2}$$

where $\mathbf{W}_{\text{pre}} \in \mathbb{R}^{4\times d}$ and $\mathbf{W}_{\text{post}} \in \mathbb{R}^{C\times 4}$ limit the trainable weights to approximately 12 parameters (excluding frozen features), matching the VQC parameter count. Third, the MLP-LG model is evaluated as a high-capacity baseline with layers $d \to 128 \to 64 \to C$ and ReLU activations. The complete training procedure is outlined in Algorithm 1.

Algorithm 1: Classical transfer learning training

Input: Dataset $\mathcal{D}$, Pretrained Model $\mathcal{M}_{pre}$, Classes C
Output: Trained Classical Model $\mathcal{M}$

Initialize $\mathcal{M} \leftarrow \mathcal{M}_{pre}$ (e.g., ResNet18);
foreach *parameter* $p \in \mathcal{M}_{backbone}$ **do**
    $p.requires\_grad \leftarrow$ False; //Freeze backbone
end
$d_{in} \leftarrow$ GetInputFeatures($\mathcal{M}_{classifier}$);
$\mathcal{M}_{classifier} \leftarrow$ Linear(d_{in}, C); //Replace head
while *epoch* $\leq$ *max_epochs* **do**
    foreach *batch* $(x, y) \in \mathcal{D}_{train}$ **do**
        $f \leftarrow \mathcal{M}_{backbone}(x)$; //Extract features
        $logits \leftarrow \mathcal{M}_{classifier}(f)$;
        $\mathcal{L} \leftarrow$ CrossEntropy($logits, y$);
        Update $\mathcal{M}_{classifier}$ via Gradient Descent;
    end
end

3.2 Classical-Quantum PennyLane (Ideal)

The PennyLane standard architecture integrates a parameterized quantum circuit as the classification layer while retaining a frozen classical backbone for feature extraction [11]. The hybrid pipeline transforms input images through the following sequence depicted in Eq. (3).

$$\mathbf{x} \xrightarrow{\phi_{\text{CNN}}} \mathbf{f} \xrightarrow{\mathbf{W}_{\text{pre}}} \mathbf{z} \xrightarrow{\tanh(\cdot)\times\frac{\pi}{2}} \boldsymbol{\theta} \xrightarrow{\text{QC}} \langle \hat{O}_i \rangle \xrightarrow{\mathbf{W}_{\text{post}}} \mathbf{y} \tag{3}$$

Classical features $\mathbf{f} \in \mathbb{R}^d$ from the frozen backbone undergo linear projection via $\mathbf{W}_{\text{pre}} \in \mathbb{R}^{n\times d}$ (d is the depth of the circuit) to match the quantum layer dimensionality of $n = 4$ qubits. The hyperbolic tangent activation scaled by $\pi/2$ ensures angle-encoded inputs $\boldsymbol{\theta} \in [-\pi/2, \pi/2]^n$ remain within valid rotation ranges. The parameterized quantum circuit $\text{QC}(\boldsymbol{\theta}, \boldsymbol{\phi})$ processes these encoded features, producing expectation values $\langle \hat{O}_i \rangle$ of Pauli-Z observables on all qubits. Finally, a post-quantum linear layer $\mathbf{W}_{\text{post}} \in \mathbb{R}^{C\times n}$ maps these quantum measurements to class logits $\mathbf{y}$.

We employ a VQC with $n = 4$ qubits and depth $D = 3$ [18,19]. This configuration is empirically justified by a sensitivity analysis (qubits $n \in \{4, 8, 16\}$ and depth $D \in \{1, 3, 5\}$), which results will be disccus in Section 4.5.

Classical features are first encoded into quantum states via single-qubit R_Y rotations [43], mapping input data $\boldsymbol{\theta}$ into quantum-state amplitudes via Eq. (4).

$$\hat{U}_{\text{enc}}(\boldsymbol{\theta}) = \prod_{i=0}^{n-1} R_Y^{(i)}(\theta_i) \tag{4}$$

The subsequent variational layers combine entanglement and parameterized transformations. Each of the $D = 3$ layers implements a ring CNOT pattern, creating ring-topology connectivity via Eq. (5).

$$\hat{U}_{\text{ent}} = \text{CNOT}_{0,1} \cdot \text{CNOT}_{1,2} \cdot \text{CNOT}_{2,3} \cdot \text{CNOT}_{3,0} \tag{5}$$

This ring configuration ensures maximal qubit connectivity [44], where the last qubit wraps around to entangle with the first, forming a complete ring. Following entanglement, trainable R_Y gates apply parameterized rotations [45] using Eq. (6).

$$\hat{U}_{\text{rot}}^{(l)}(\boldsymbol{\phi}_l) = \prod_{i=0}^{n-1} R_Y^{(i)}(\phi_{l,i}) \tag{6}$$

where $\boldsymbol{\phi}_l = (\phi_{l,0}, \phi_{l,1}, \phi_{l,2}, \phi_{l,3})$ represents trainable parameters for layer l. The complete circuit combines these operations sequentially, as shown in Eq. (7).

$$\hat{U}_{\text{circuit}}(\boldsymbol{\theta}, \{\boldsymbol{\phi}_l\}) = \prod_{l=0}^{D-1} \left[\hat{U}_{\text{rot}}^{(l)}(\boldsymbol{\phi}_l) \cdot \hat{U}_{\text{ent}} \right] \cdot \hat{U}_{\text{enc}}(\boldsymbol{\theta}) \tag{7}$$

Measurement extracts outputs as expectation values of the Pauli-Z operator on all qubits using Eq. (8).

$$\langle \hat{O}_i \rangle = \langle \psi | \hat{Z}_i | \psi \rangle, \quad i = 0, 1, 2, 3 \tag{8}$$

producing a 4-dimensional feature vector for the post-quantum classifier using Pauli Z. The total number of trainable quantum parameters amounts to $n \times D = 4 \times 3 = 12$.

The resulting ideal PennyLane circuit is illustrated in Fig. 1, which shows the initial encoding of the R_Y feature, the three variational layers with CNOT operations, and the final Pauli-Z expectation measurements used as quantum features. The corresponding inference and training pipeline is summarized in Algorithm 2.

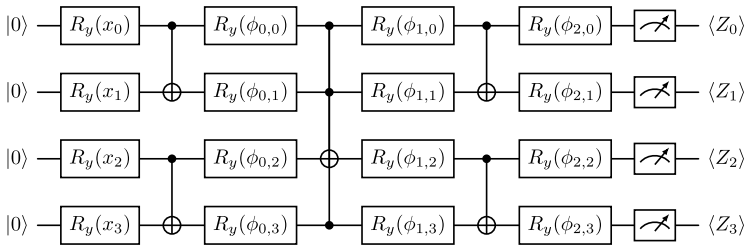

Figure 1: PennyLane ideal quantum circuit.

Algorithm 2: Classical-quantum PennyLane (ideal)

Input: Features f, Qubits n, Layers D
Output: Expectation Values $\langle Z\rangle$
Initialize Device 'default.qubit';
Define Circuit $U(\theta, \phi)$:;
  AngleEmbedding(θ); //Encode data
  for $l \leftarrow 1$ **to** D **do**
    BasicEntanglerLayers(ϕ_l); //Ring CNOTs + Rot
  end
  return $[\langle Z_0\rangle, \ldots, \langle Z_{n-1}\rangle]$;
Forward Pass:;
  $x_{proj} \leftarrow W_{pre} \cdot f$; //Reduce dimension
  $\theta \leftarrow \tanh(x_{proj}) \cdot \frac{\pi}{2}$; //Scale to $[-\pi/2, \pi/2]$
  $q_{out} \leftarrow$ Execute $U(\theta, \phi)$;
  $logits \leftarrow W_{post} \cdot q_{out}$;

3.3 Classical-Quantum PennyLane (Noisy)

To simulate realistic quantum hardware, we implement noise models based on calibration data from IBM Quantum devices (Section 4.4.4). The PennyLane noisy architecture employs a density matrix simulator to model decoherence and gate errors.

We calculate the damping probabilities using exponential decay formulas derived from coherence times. For single-qubit gates ($t_{\text{gate}}^{(1q)} = 32$ ns), the amplitude and phase decay probabilities are given by Eqs. (9) and (10).

$$\gamma_{1q} = 1 - \exp\left(-\frac{t_{\text{gate}}^{(1q)}}{T_1}\right) \approx 1.28 \times 10^{-4} \tag{9}$$

$$\lambda_{1q} = 1 - \exp\left(-\frac{t_{\text{gate}}^{(1q)}}{T_2}\right) - \gamma_{1q} \approx 8.53 \times 10^{-5} \tag{10}$$

In the case of two-qubit CZ gates ($t_{\text{gate}}^{(2q)} = 68$ ns), the longer execution time results in higher damping probabilities. The corresponding amplitude and phase damping probabilities are defined in Eqs. (11) and (12), respectively.

$$\gamma_{2q} = 1 - \exp\left(-\frac{t_{\text{gate}}^{(2q)}}{T_1}\right) \approx 2.72 \times 10^{-4} \tag{11}$$

$$\lambda_{2q} = 1 - \exp\left(-\frac{t_{\text{gate}}^{(2q)}}{T_2}\right) - \gamma_{2q} \approx 1.81 \times 10^{-4} \tag{12}$$

These probabilities feed into Kraus operator representations for amplitude damping $\mathcal{E}_{\text{AD}}(\gamma)$ and phase damping $\mathcal{E}_{\text{PD}}(\lambda)$. The noisy circuit maintains the identical topology to PennyLane standard (described above), but inserts these noise channels after every quantum operation to physically model the relaxation and dephasing inherent to superconducting qubits. Algorithm 3 formalizes the noisy PennyLane execution.

Algorithm 3: Classical-quantum PennyLane (noisy)

Input: Features f, Noise Params $\Gamma_{1q}, \Lambda_{1q}, \Gamma_{2q}, \Lambda_{2q}$
Initialize Device 'default.mixed';
Define Noisy Circuit $U_{noisy}(\theta, \phi)$:;
    AngleEmbedding(θ);
    Apply $\mathcal{E}_{AD}(\Gamma_{1q}) \circ \mathcal{E}_{PD}(\Lambda_{1q})$ on all qubits;
    for $l \leftarrow 1$ **to** D **do**
        foreach *CNOT* (q_i, q_j) *in Entangler* **do**
            Apply CNOT (q_i, q_j);
            Apply $\mathcal{E}_{AD}(\Gamma_{2q}) \circ \mathcal{E}_{PD}(\Lambda_{2q})$ on q_i, q_j;
        end
        foreach *Rotation* R_Y *on* q_i **do**
            Apply $R_Y(\phi_{l,i})$;
            Apply $\mathcal{E}_{AD}(\Gamma_{1q}) \circ \mathcal{E}_{PD}(\Lambda_{1q})$ on q_i;
        end
    end
    return $[\langle Z_0 \rangle, \ldots, \langle Z_{n-1} \rangle]$;

3.4 Classical-Quantum Qiskit (Ideal)

The Qiskit architecture is implemented using the Qiskit Machine Learning framework [46]. To ensure a completely fair comparison with PennyLane, the execution and optimization settings were homogenized. In the ideal setting, we employ Qiskit's aer simulator with SPSA.

The circuit architecture, shown in Fig. 2, employs a ring entangling pattern combined with Hadamard initialization. Hadamard gates first create superposition states in all qubits using Eq. (13).

$$|\psi_0\rangle = \bigotimes_{i=0}^{n-1} H^{(i)}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle \tag{13}$$

where each Hadamard gate $H^{(i)}$ prepares qubit i in uniform superposition, creating an equal probability distribution across all 2^n computational basis states. Classical data embedding follows through R_Y rotations using Eq. (14).

$$\hat{U}_{\text{enc}}(\boldsymbol{\theta}) = \prod_{i=0}^{n-1} R_Y^{(i)}(\theta_i) \tag{14}$$

where each rotation $R_Y^{(i)}(\theta_i) = \exp(-i\theta_i Y/2)$ encodes a classical feature θ_i through rotation around the Y axis.

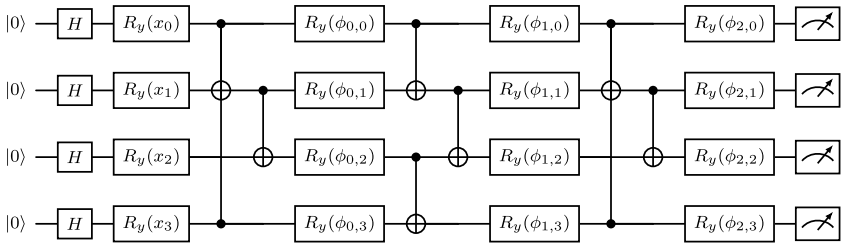

Figure 2: Qiskit quantum circuit employing brick-wall entangling pattern (4 qubits, depth = 3).

Variational layers employ parameterized circuit blocks. For each layer $l \in 0, 1, 2$, the transformation combines CNOT operations with trainable R_Y rotations, as defined in Eq. (15).

$$\hat{U}_{\text{var}}^{(l)} = \hat{U}_{\text{rot}}(\boldsymbol{\phi}_l) \cdot \hat{U}_{\text{ent}}^{\text{odd}} \cdot \hat{U}_{\text{ent}}^{\text{even}} \tag{15}$$

The even CNOT layer entangles adjacent qubit pairs $(0, 1)$ and $(2, 3)$ using Eq. (16).

$$\hat{U}_{\text{ent}}^{\text{even}} = \text{CNOT}_{0,1} \cdot \text{CNOT}_{2,3} \tag{16}$$

while the odd layer targets the intermediate pair $(1, 2)$ using Eq. (17)

$$\hat{U}_{\text{ent}}^{\text{odd}} = \text{CNOT}_{1,2} \tag{17}$$

This alternating pattern ensures efficient entanglement distribution [44], thereby establishing connectivity between all qubit pairs within the layered structure. Trainable rotations following entanglement apply parameterized R_Y gates using Eq. (18).

$$\hat{U}_{\text{rot}}(\boldsymbol{\phi}_l) = \prod_{i=0}^{n-1} R_Y^{(i)}(\phi_{l,i}) \tag{18}$$

The complete circuit combines initialization, encoding, and variational layers as specified in Eq. (19).

$$\hat{U}_{\text{Qiskit}}(\boldsymbol{\theta}, \{\boldsymbol{\phi}_l\}) = \prod_{l=0}^{D-1} \hat{U}_{\text{var}}^{(l)}(\boldsymbol{\phi}_l) \cdot \hat{U}_{\text{enc}}(\boldsymbol{\theta}) \cdot \prod_{i=0}^{n-1} H^{(i)} \tag{19}$$

with total trainable parameters reaching $n \times D = 4 \times 3 = 12$.

All n qubits are assessed in the computational basis, producing binary strings $\mathbf{b} \in \{0, 1\}^n$. The interpretation function maps binary strings to class labels by computing the Hamming weight using Eq. (20).

$$f_{\text{interpret}}(\mathbf{b}) = \text{HammingWeight}(\mathbf{b}) \mod C \tag{20}$$

where $\text{HammingWeight}(\mathbf{b}) = \sum_{i=0}^{n-1} b_i$ counts the number of ones in the binary string and C denotes the number of classes. The outputs probability distributions over classes by aggregating binary strings that map to identical labels using Eq. (21). The full Qiskit ideal pipeline is described in Algorithm 4.

$$P(c) = \sum_{\mathbf{b}: f(\mathbf{b}) = c} |\langle \mathbf{b} | \psi(\boldsymbol{\theta}, \boldsymbol{\phi}) \rangle|^2 \tag{21}$$

Algorithm 4: Classical-quantum Qiskit (ideal)

Input: Features f, Qubits $n = 4$, Depth $D = 3$
Initialize 'EstimatorQNN' with Gradient support;
Define Circuit QC:;
  Apply Hadamard to all q;
  Apply $R_Y(\theta)$; //Feature Map
  for $l \leftarrow 1$ **to** D **do**
    Apply Brick-Wall CNOTs;
    Apply $R_Y(\phi_l)$; //Variational
  end
Define Observables $O = \{ZIII, IZII, \dots\}$;
Forward Pass:;
  $\theta \leftarrow \tanh(W_{in} \cdot f) \cdot \frac{\pi}{2}$;
  $exp\_vals \leftarrow$ Estimator.run(QC, θ, O);
return exp_vals;

3.5 Classical-Quantum Qiskit (Noisy)

The Qiskit Noisy architecture employs the Qiskit Aer simulator [47], incorporating depolarizing noise channels and finite-shot measurements.

To ensure a fair comparison between frameworks, both PennyLane and Qiskit implementations use identical underlying noise parameters. While PennyLane implements amplitude and phase damping channels, Qiskit employs thermal relaxation combined with depolarizing noise. Both approaches are calibrated to achieve equivalent circuit fidelity for depth-3, 4-qubit circuits, as measured by Eq. (22).

$$F_{\text{target}} \approx 0.94\text{–}0.96 \tag{22}$$

This high-fidelity range reflects the significantly improved performance of Heron r2 processors.

Qiskit implements thermal relaxation errors using the same T_1 and T_2 values, combined with depolarizing channels calibrated to match Heron r2 gate error rates. These values are descripted in Section 4.4.

These values are identical to the gate error rates used in PennyLane, ensuring equivalent noise in all frameworks.

Using unified noise parameters based on the Heron r2 specifications, we ensure that both frameworks are evaluated under comparable equivalent-circuit fidelity conditions. For a depth-3, 4-qubit circuit (approximately 28 total gates), we estimate using Eq. (23):

$$F_{\text{PL}} \approx F_{\text{QK}} \approx 0.94\text{–}0.96 \tag{23}$$

This higher fidelity range reflects the improved coherence and gate quality of Heron processors compared to earlier generations. The equivalence ensures that observed performance differences reflect framework-specific factors rather than noise intensity discrepancies.

Both Qiskit and PennyLane noisy configurations employ finite-shot sampling and SPSA optimization to resolve the execution asymmetry. This introduces statistical sampling noise given by Eq. (24).

$$\sigma_{\text{shot}}[\hat{P}(b)] \approx \frac{1}{\sqrt{N_{\text{shots}}}} \approx 3.1\% \tag{24}$$

This sampling variance adds to the hardware-calibrated thermal relaxation and depolarizing gate errors, creating equivalent stochasticity in both software platforms. This homogenization isolates the influence of the software implementation from the optimization method, enabling a fair comparison of their computational and memory resources under realistic NISQ constraints. Algorithm 5 summarizes the noisy Qiskit inference procedure.

Algorithm 5: Classical-quantum Qiskit (noisy)

Input: Features f, Backend B, Shots $N = 1024$
Output: Class Probabilities
Build 'NoiseModel' from backend B calibration;
Initialize 'SamplerQNN' with N shots and NoiseModel;
Define Interpret Function $I(b) = \text{Hamming}(b) \pmod{C}$;
Forward Pass:;
    $\theta \leftarrow \text{ScaleFeatures}(f)$;
    Sample binary strings $b \sim |\psi(\theta, \phi)|^2$ (N times);
    Compute raw counts $\{b : \text{count}\}$;
    Map counts to classes using $I(b)$;
    $P(c) \leftarrow \frac{\sum_{b:I(b)=c} \text{count}(b)}{N}$;
    return $P(c)$;

As a summary of the methodological setup, Fig. 3 presents a overview of the five evaluated architectures, highlighting the shared feature extraction stage and the classical and quantum transfer learning heads considered in this work.

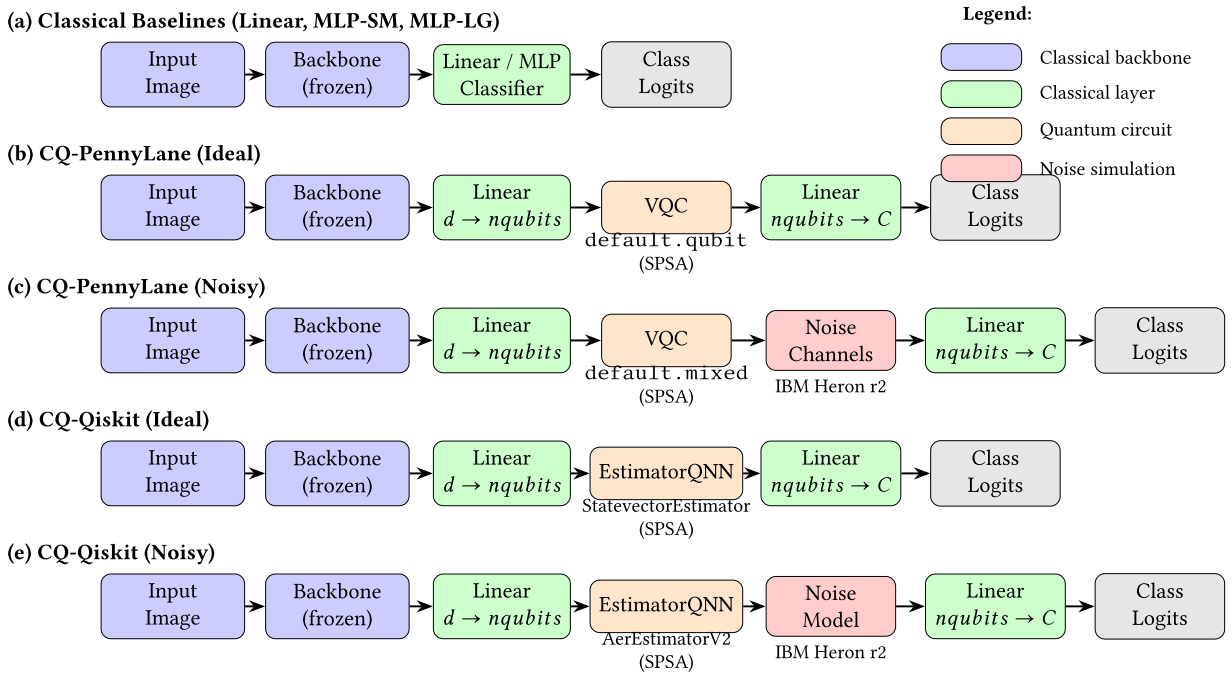

Figure 3: Overview of the evaluated architectures: three classical baselines and four hybrid classical-quantum variants.

4 Results

This section reports the empirical evaluation of the proposed benchmark. The study design and the evaluated configurations are first described in Section 4.1. The datasets and evaluation metrics are then introduced in Sections 4.2 and 4.3, respectively. Section 4.4 details the experimental setup, including the computing environment, training protocol, noise calibration procedure, and real quantum hardware configuration. Finally, Section 4.5 presents the experimental results and discussion, covering predictive performance, computational cost, statistical significance analysis, sensitivity studies, noise decomposition, real-hardware experiments, and the main limitations encountered.

4.1 Study Design

This experimental study is organized along three dimensions of variation. Every combination of these dimensions defines a configuration that is evaluated in the remainder of this section.

1. Dataset. Four binary image classification datasets are considered, namely Hymenoptera, Brain Tumor MRI, Cats vs. Dogs, and Solar Dust. The four datasets span heterogeneous domains, sample sizes, and difficulty levels. A description is provided in Section 4.2.
2. Backbone. Four pretrained convolutional architectures are employed as frozen feature extractors. The selection covers a representative spectrum of parameter counts and output dimensionalities:
   (a) ResNet-18 [48]: 11.7M parameters, residual connections, 512-dimensional features.
   (b) MobileNet-V2 [49]: 3.5M parameters, depthwise separable convolutions, 1280-dimensional features.
   (c) EfficientNet-B0 [50]: 5.3M parameters, compound scaling, 1280-dimensional features.
   (d) RegNet-X-400MF [51]: 5.2M parameters, Neural Architecture Search (NAS)-designed architecture, 400-dimensional features.
3. Head. Seven classification heads are evaluated. The first three correspond to classical baselines, and the remaining four correspond to hybrid quantum heads that are organized along two further factors, namely the quantum framework (PennyLane or Qiskit) and the execution environment (noise-free simulation or noise-aware emulation). In particular, the classical heads used are:
   (a) A single fully connected layer that maps the backbone embedding directly to the class logits (Linear). This head establishes the simplest classical baseline and constitutes the reference configuration against which every other head is compared.
   (b) A two-layer multilayer perceptron with a hyperbolic-tangent nonlinearity, designed to have a parameter count matched to the VQC (mlp-sm). By holding the parameter budget constant, this baseline isolates the contribution of a single nonlinearity from the contribution of expanded model capacity.
   (c) A higher-capacity multilayer perceptron with two hidden layers of 128 and 64 units and ReLU activations, totaling several thousand trainable parameters (mlp-lg). This baseline establishes the upper performance ceiling reachable with conventional architectures on the same backbone embeddings.

Analogously, the quantum heads used are:

1. The hybrid PennyLane head executed on the ideal statevector simulator, with exact expectation values and no shot-based sampling (PL-Ideal). This configuration provides the upper bound of the PennyLane stack under idealized conditions.
2. The same PennyLane head executed on the mixed-state simulator, with amplitude-damping, phase-damping, and depolarizing channels calibrated against the IBM Heron r2 backend (PL-Noisy).

This configuration quantifies the accuracy degradation produced by realistic decoherence within the PennyLane stack.

3. The hybrid Qiskit head implemented through the StatevectorEstimator primitive, with exact expectation values and no shot-based sampling (QK-Ideal). This configuration mirrors the PennyLane ideal regime within the Qiskit ecosystem and enables framework-controlled comparisons.
4. The same Qiskit head executed through the AerEstimatorV2 primitive, with a Heron r2 calibrated noise model (depolarizing channels and readout error) and 1024 shots per circuit evaluation (QK-Noisy). This configuration quantifies the joint impact of shot noise and decoherence within the Qiskit stack.

Each configuration in the resulting $4 \times 4 \times 7$ grid is replicated with five random seeds, namely $\{0, 42, 123, 456, 789\}$, and all metrics are reported as mean plus or minus standard deviation. For instance, the configuration (Hymenoptera, RegNet-X-400MF, PL-Noisy) refers to the evaluation of the Hymenoptera dataset using the RegNet-X-400MF backbone as the frozen feature extractor and the PennyLane noisy quantum classification head.

4.2 Datasets

We evaluate four binary image datasets of heterogeneous difficulty: Hymenoptera (ants vs. bees), Brain Tumor MRI (tumor vs. no tumor), Cats vs. Dogs, and Solar Dust (defect vs. normal). All images are resized to 224×224 and preprocessed with ImageNet statistics. In all datasets, the training partition is split 80% for training and 20% for validation, while the held-out test partition is reserved exclusively for final evaluation and reporting.

Table 2 provides a comparative overview of the four datasets, highlighting the diversity in scale, domain, and complexity.

Table 2: Comprehensive dataset characteristics summary.

Dataset	Train	Test	Total	Difficulty	Domain
Hymenoptera [52]	245	153	398	Medium	Entomology
Brain Tumor [53]	2000	400	2400	High	Medical Imaging
Cats vs. Dogs [54]	3000	600	3600	Medium-Low	General Vision
Solar Dust [55]	240	60	300	High	Industrial

4.3 Metrics

The primary classification metric is the test accuracy, which is defined in Eq. (25).

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i), \tag{25}$$

where N represents the number of test samples, y_i is the true label, $\hat{y}_i$ is the predicted label, and $\mathbb{I}(\cdot)$ represents the indicator function.

To evaluate the classification behavior under class imbalances, the macro-averaged precision and recall are also calculated. The macro-averaged precision is defined in Eq. (26).

$$\text{Precision}_{\text{macro}} = \frac{1}{C} \sum_{c=1}^{C} \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \tag{26}$$

where C represents the number of classes, TP_c is the number of true positives for class c, and FP_c is the number of false positives for class c. The macro-averaged recall is depicted in Eq. (27).

$$\mathrm{Recall}_{\mathrm{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \tag{27}$$

where FN_c is the number of false negatives for class c.

The macro-averaged F1-score is calculated as shown in Eq. (28).

$$\mathrm{F1}_{\mathrm{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{2 \times \mathrm{Precision}_c \times \mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}, \tag{28}$$

where $\mathrm{Precision}_c$ and Recall_c are the precision and recall values for class c.

The discrimination ability of each classifier is measured using the Area Under the Receiver Operating Characteristic curve. The area under this curve is defined in Eq. (29).

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR}), \tag{29}$$

where the True Positive Rate (TPR) and False Positive Rate (FPR) are calculated via Eq. (30).

$$\mathrm{TPR}(\theta) = \frac{\mathrm{TP}(\theta)}{\mathrm{TP}(\theta) + \mathrm{FN}(\theta)}, \quad \mathrm{FPR}(\theta) = \frac{\mathrm{FP}(\theta)}{\mathrm{FP}(\theta) + \mathrm{TN}(\theta)}, \tag{30}$$

where $\mathrm{TN}(\theta)$ is the number of true negatives for a given threshold θ.

Additionally, the training execution time in seconds, denoted as t_s, is tracked as the primary metric of computational efficiency. The classification accuracy and other statistical metrics are reported in the range $[0, 1]$ or as percentages. For improved readability, percentage points are discussed when comparing different classification heads.

4.4 Experimental Setup

This subsection details the conditions under which all configurations are evaluated. The computing platform used for the simulations is reported first, followed by the training hyperparameters shared by all classical and quantum heads. The calibration of the noise model employed in the noisy emulations is then described, and the IBM quantum processor used for the real-hardware experiments is specified at the end.

4.4.1 Computing Platform

All experiments were executed on a standardized computing platform, summarized in Table 3.

Quantum circuit simulation is restricted to a single thread because, for 4-qubit state vectors (16 complex amplitudes), the cost of OS-level thread synchronization exceeds the cost of the underlying linear-algebra operations. The convolutional backbones, in contrast, operate on tensors that are large enough to benefit from PyTorch's default multi-threaded CPU parallelization. The same threading policy is applied in both the ideal and noisy regimes.

Table 3: Hardware and software environment used in all experiments.

Component	Specification
CPU	Intel Xeon Platinum 8260 (2.40 GHz, 4 cores)
RAM	16 GB
Operating system	Linux (CentOS/RHEL kernel 4.18)
Python	3.10.20
PyTorch	2.4.1
TorchVision	0.19.1
PennyLane	0.42.0
Qiskit	1.2.4
Qiskit Machine Learning	0.8.1
Qiskit Aer	0.15.1

4.4.2 Training Configuration

Regarding hyperparameters, the same configuration is applied for all seven heads to isolate the impact of the quantum components.

Table 4 shows that the optimizer is Adam ($\eta = 10^{-3}$, batch size 16) with a StepLR scheduler (decaying by $\gamma = 0.9$ every 3 epochs). A weight decay of 0.01 is applied only to classical heads to provide architectural symmetry with bounded quantum operations. The models are trained for 10 epochs.

Table 4: Unified training hyperparameters for all architectures.

Parameter	Value
Optimizer	Adam
Learning rate (η)	1×10^{-3}
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1×10^{-8}
Weight decay	0.01 (classical heads)/0 (quantum heads)
LR scheduler	StepLR
Step size	3 epochs
Gamma (γ)	0.9
Batch size	16
Total epochs	10
Gradient clipping	None
Quantum-specific settings:	
Number of qubits (n)	4
Circuit depth (D)	3
PennyLane device (clean)	`lightning.qubit/default.qubit`
PennyLane device (noisy)	`default.mixed`
PennyLane gradient	SPSA (ideal)/SPSA (noisy)
Qiskit shots	1024 (noisy)/None (ideal)

(Continued)

Table 4 (continued)

Parameter	Value
Qiskit gradient	SPSA (ideal)/SPSA (noisy)
Circuit type	VQC
Variational parameters per layer	4 (n)
Total trainable quantum parameters	12 ($n \times D$)

Quantum heads use 4 qubits and circuit depth 3. Noisy emulations employ 1024 shots and realistic IBM Heron r2-calibrated noise models: a composite depolarizing noise model (p_{1q} = 0.02%, p_{2q} = 0.5%, readout error 1.2%) using the `AerEstimatorV2` primitive in Qiskit, and amplitude/phase damping channels in PennyLane.

Both ideal and noisy environments use SPSA for gradient estimation under matched shot counts to isolate framework influence from optimization variance.

The StepLR scheduler reduces the learning rate by a factor $\gamma = 0.9$ every 3 epochs, providing gentle annealing without aggressive decay. This conservative schedule proved effective in all datasets without requiring dataset-specific tuning.

Both PennyLane and Qiskit variants employ shallow VQCs with four qubits and depth three. The number of variational parameters per layer is 4, producing a total of 12 trainable quantum parameters.

Using SPSA in both the ideal and noisy regimes keeps the gradient estimator constant in noise conditions, so any performance gap between the two regimes can be attributed to noise rather than to the optimizer.

4.4.3 Noise Model Calibration

Noise parameters for emulation are based on publicly available specifications for IBM Quantum Heron r2 processors (2024–2025). These processors feature improved coherence times and gate fidelities compared to earlier Falcon-generation devices. Table 5 presents the unified calibration parameters used in both PennyLane and Qiskit implementations.

Table 5: IBM Heron r2 processor specifications used for noise calibration and real hardware experiments.

Parameter	Value	Usage
Coherence properties:		
T_1 (energy relaxation)	250 µs	PL damping
T_2 (dephasing time)	150 µs	PL dephasing
Gate timing:		
Single-qubit gate time	32 ns	PL noise calculations
Two-qubit gate time (CZ)	68 ns	PL noise calculations
Error rates:		
Single-qubit gate error (p_{1q})	0.02%	Qiskit depolarizing
Two-qubit gate error (p_{2q})	0.5%	Qiskit depolarizing
Readout error (SPAM)	1.2%	Both

For PennyLane noisy emulations, we compute amplitude and phase damping probabilities from coherence times using exponential decay formulas (Eqs. (9)–(12)). For Qiskit noisy simulations, we apply depolarizing channels with error rates matching the gate error specifications. This unified calibration ensures that both frameworks experience the same noise intensity, enabling a fair cross-platform comparison.

In both frameworks, dynamical decoupling has been enabled during the idle windows of the variational circuit to suppress dephasing on the qubits that wait between active gates, so that the noisy emulations reflect the same mitigation scheme that is later deployed on real hardware.

4.4.4 IBM Quantum Backend

To validate our results, we conducted experiments on real IBM Quantum hardware accessed through IBM Quantum Cloud services. For real Quantum Processing Unit (QPU) experiments, experiments were executed on ibm_torino, a 133-qubit IBM Heron r2 processor. Table 6 summarizes the execution configuration.

Table 6: Real quantum hardware execution configuration.

Parameter	Value
Backend	`ibm_torino`
Processor type	IBM Heron r2
Total qubits	133
Qubits used	4
Circuit depth (transpiled)	~49
Shots per circuit	1024
Transpilation level	1
Error mitigation	Dynamical decoupling

4.5 Discussion

This section discusses the experimental results obtained with the proposed benchmark. The analysis begins with the predictive performance and training-time evaluation of the classical and quantum configurations (Section 4.5.1). Statistical significance tests are then performed to determine whether the observed differences are consistent across datasets and backbone architectures (Section 4.5.2). Subsequently, sensitivity and noise-decomposition studies are conducted to investigate the influence of circuit design choices and individual noise channels (Sections 4.5.3 and 4.5.4, respectively). Finally, experiments on real IBM quantum hardware are presented (Section 4.5.5) and the main limitations associated with current devices are discussed (Section 4.5.6).

4.5.1 Performance and Training Time Study

From the performance point of view, we only present test accuracy and F1-Score in Table 7 for the baseline heads (Linear, MLP-SM and MLP-LG) and, in Table 8, for the quantum heads (PL-Ideal, PL-Noisy, QK-Ideal and QK-Noisy). Additional metrics, together with training and validation loss/accuracy curves, confusion matrix, ROC curve and Precision-Recall curve, can be found in the Supplementary Material (Tables S1–S15 and Figs. S1–S5).

Table 7: Main experimental results (classical heads): test accuracy (%) and F1-score (%) for all datasets, backbones, and heads. Values reported as mean ± std over 5 random seeds.

Dataset	Backbone	Linear		MLP-SM		MLP-LG	
		Acc	F1	Acc	F1	Acc	F1
Hymenoptera	ResNet-18	97.9 ± 0.7	97.9 ± 0.7	87.2 ± 22.3	84.1 ± 29.1	97.4 ± 1.3	97.4 ± 1.3
	MobileNet-V2	97.4 ± 0.0	97.4 ± 0.0	97.2 ± 0.6	97.2 ± 0.6	96.9 ± 2.1	96.9 ± 2.2
	EfficientNet-B0	97.2 ± 0.6	97.2 ± 0.6	97.4 ± 0.9	97.4 ± 0.9	97.2 ± 0.6	97.2 ± 0.6
	RegNet-400MF	96.9 ± 1.1	96.9 ± 1.2	97.2 ± 1.7	97.2 ± 1.7	97.9 ± 1.1	97.9 ± 1.1
Brain Tumor	ResNet-18	97.4 ± 0.5	97.4 ± 0.5	97.2 ± 1.1	97.1 ± 1.1	96.0 ± 2.2	95.9 ± 2.2
	MobileNet-V2	95.9 ± 0.3	95.9 ± 0.3	96.5 ± 0.4	96.5 ± 0.4	96.3 ± 0.4	96.3 ± 0.4
	EfficientNet-B0	97.3 ± 0.4	97.3 ± 0.4	97.8 ± 0.3	97.8 ± 0.3	98.0 ± 0.2	97.9 ± 0.2
	RegNet-400MF	96.0 ± 0.2	96.0 ± 0.2	96.2 ± 0.3	96.2 ± 0.3	96.3 ± 0.7	96.3 ± 0.7
Cats vs. Dogs	ResNet-18	98.3 ± 0.7	98.3 ± 0.7	97.9 ± 0.5	97.9 ± 0.5	98.1 ± 0.5	98.1 ± 0.5
	MobileNet-V2	99.1 ± 0.1	99.1 ± 0.1	99.3 ± 0.1	99.3 ± 0.1	99.0 ± 0.1	99.0 ± 0.1
	EfficientNet-B0	99.2 ± 0.4	99.2 ± 0.4	99.3 ± 0.5	99.3 ± 0.5	99.0 ± 0.4	99.0 ± 0.4
	RegNet-400MF	98.6 ± 0.2	98.6 ± 0.2	98.7 ± 0.4	98.7 ± 0.4	98.6 ± 0.1	98.6 ± 0.1
Solar Dust	ResNet-18	87.5 ± 4.2	87.3 ± 4.5	88.5 ± 1.9	88.4 ± 2.0	91.3 ± 1.4	91.3 ± 1.4
	MobileNet-V2	91.5 ± 1.1	91.5 ± 1.1	91.0 ± 0.9	91.0 ± 0.9	88.8 ± 2.5	88.8 ± 2.5
	EfficientNet-B0	92.5 ± 0.8	92.5 ± 0.8	92.8 ± 1.9	92.8 ± 1.9	92.7 ± 2.8	92.6 ± 2.8
	RegNet-400MF	82.0 ± 2.5	81.7 ± 2.6	86.0 ± 2.2	85.9 ± 2.3	89.2 ± 3.1	89.1 ± 3.2

Table 8: Main experimental results (quantum heads): test accuracy (%) and F1-score (%) for datasets, backbones, and heads. Values reported as mean ± std over 5 random seeds.

Dataset	Backbone	PL-Ideal		PL-Noisy		QK-Ideal		QK-Noisy	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
Hymenoptera	ResNet-18	96.4 ± 1.4	96.4 ± 1.4	90.3 ± 4.8	90.1 ± 4.9	84.4 ± 14.9	83.2 ± 17.1	82.6 ± 21.3	79.2 ± 27.8
	MobileNet-V2	98.2 ± 0.7	98.2 ± 0.7	95.4 ± 3.9	95.3 ± 4.1	97.2 ± 1.4	97.2 ± 1.4	96.9 ± 1.9	96.9 ± 2.0
	EfficientNet-B0	96.9 ± 0.7	96.9 ± 0.7	96.2 ± 1.6	96.1 ± 1.6	96.2 ± 1.8	96.1 ± 1.8	95.6 ± 2.7	95.6 ± 2.7
	RegNet-400MF	97.7 ± 2.3	97.7 ± 2.3	82.1 ± 18.4	78.2 ± 25.9	79.5 ± 22.0	74.6 ± 29.1	87.4 ± 19.8	83.8 ± 27.8
Brain Tumor	ResNet-18	97.2 ± 0.7	97.2 ± 0.7	96.9 ± 0.8	96.9 ± 0.8	86.8 ± 20.8	83.9 ± 27.2	96.1 ± 1.0	96.1 ± 1.0
	MobileNet-V2	96.7 ± 0.3	96.7 ± 0.3	95.8 ± 0.4	95.8 ± 0.4	96.2 ± 0.5	96.2 ± 0.5	96.2 ± 0.4	96.1 ± 0.4
	EfficientNet-B0	97.9 ± 0.3	97.9 ± 0.3	97.0 ± 0.4	97.0 ± 0.4	97.9 ± 0.4	97.9 ± 0.4	97.9 ± 0.1	97.8 ± 0.1
	RegNet-400MF	96.4 ± 0.5	96.4 ± 0.5	95.6 ± 0.9	95.6 ± 0.9	95.7 ± 0.4	95.7 ± 0.4	95.5 ± 0.3	95.5 ± 0.3
Cats vs. Dogs	ResNet-18	98.1 ± 0.5	98.1 ± 0.5	97.5 ± 0.9	97.5 ± 0.9	97.4 ± 1.0	97.4 ± 1.0	97.7 ± 0.6	97.7 ± 0.6
	MobileNet-V2	99.2 ± 0.3	99.2 ± 0.3	99.1 ± 0.5	99.1 ± 0.5	99.3 ± 0.2	99.3 ± 0.2	99.2 ± 0.3	99.2 ± 0.3
	EfficientNet-B0	98.8 ± 0.5	98.8 ± 0.5	99.0 ± 0.5	99.0 ± 0.5	99.0 ± 0.4	99.0 ± 0.4	98.8 ± 0.6	98.8 ± 0.6
	RegNet-400MF	99.0 ± 0.2	99.0 ± 0.2	98.1 ± 1.0	98.1 ± 1.0	98.8 ± 0.1	98.8 ± 0.1	98.4 ± 0.5	98.4 ± 0.5
Solar Dust	ResNet-18	73.0 ± 21.0	66.3 ± 30.1	63.3 ± 18.3	53.3 ± 27.4	70.8 ± 19.2	64.1 ± 28.3	63.0 ± 20.4	53.5 ± 28.9
	MobileNet-V2	90.5 ± 1.3	90.5 ± 1.3	89.3 ± 2.7	89.3 ± 2.7	81.5 ± 12.3	80.0 ± 14.4	76.3 ± 21.2	71.0 ± 28.7
	EfficientNet-B0	92.7 ± 1.4	92.7 ± 1.4	91.3 ± 2.1	91.3 ± 2.1	88.3 ± 4.5	88.3 ± 4.6	88.8 ± 3.9	88.8 ± 3.9
	RegNet-400MF	83.7 ± 5.3	83.5 ± 5.5	65.5 ± 8.9	63.8 ± 9.8	68.2 ± 13.2	65.7 ± 16.2	65.7 ± 7.3	64.1 ± 9.4

The ideal quantum classification heads achieve test accuracies that match the performance of the classical baselines. For example, on the Hymenoptera dataset with the MobileNet-V2 backbone, PL-Ideal

achieves 98.2 ± 0.7%, which matches the classical LINEAR baseline at 97.4 ± 0.0% and the MLP-SM baseline at 97.2 ± 0.6%. On the Cats vs. Dogs dataset, all quantum and classical configurations achieve accuracies above 98.0%. These results confirm that VQCs can function as classification heads when paired with pre-trained classical backbones. However, QK-IDEAL shows lower mean accuracies and larger standard deviations in specific configurations. For example, on the Hymenoptera dataset with the ResNet-18 backbone, QK-IDEAL achieves a test accuracy of 84.4 ± 14.9%, representing a decrease compared to PL-IDEAL (96.4 ± 1.4%). This difference is due to the SPSA optimizer, which is susceptible to convergence failures depending on the initial parameters.

The simulation of physical noise causes a decrease in classification accuracy across the configurations. On the Hymenoptera dataset with the ResNet-18 backbone, PL-NOISY and QK-NOISY achieve accuracies of 90.3 ± 4.8% and 82.6 ± 21.3%. These values are lower than the corresponding ideal simulation performances. On the Solar Dust dataset with the RegNet-400MF backbone, the accuracy of PL-NOISY drops to 65.5 ± 8.9%, showing a severe susceptibility to physical noise. In contrast, the Cats vs. Dogs dataset is robust against quantum noise. The test accuracies for PL-NOISY and QK-NOISY remain above 97.5% across all backbones. This behavior indicates that datasets with highly separable feature distributions are less affected by hardware noise during training.

The generalization capability of variational quantum classifiers varies with the characteristics of the dataset and the capacity of the model. On highly separable datasets such as Brain Tumor, the features extracted by the classical backbone are well dispersed, and both classical and quantum classifiers saturate at comparable accuracy levels near 98%. On small-sample datasets such as Hymenoptera, high-capacity classical models can overfit, while simple linear models may struggle to resolve overlapping decision boundaries. The VQC operates with only 12 trainable parameters, which limits its capacity and acts as a form of implicit regularization. As a result, the quantum head reaches accuracies that are comparable to the classical baselines on these small-sample datasets, without producing a systematic advantage.

Regarding training time, these are reported in seconds (Table 9) for the baseline heads (LINEAR, MLP-SM and MLP-LG) and, in Table 10, for the quantum heads (PL-IDEAL, PL-NOISY, QK-IDEAL and QK-NOISY).

Table 9: Classical training time (s) across datasets, backbones, and heads. Values reported as mean ± std over 5 random seeds.

Dataset	Backbone	LINEAR	MLP-SM	MLP-LG
Hymenoptera	ResNet-18	95.9 ± 0.3	97.4 ± 1.9	101.1 ± 1.4
	MobileNet-V2	69.2 ± 5.5	70.7 ± 3.9	80.9 ± 10.3
	EfficientNet-B0	108.3 ± 0.7	109.6 ± 0.5	77.2 ± 2.6
	RegNet-400MF	49.8 ± 1.5	56.7 ± 3.3	56.7 ± 0.4
Brain Tumor	ResNet-18	466.6 ± 17.5	472.0 ± 1.8	466.8 ± 1.6
	MobileNet-V2	417.7 ± 36.8	330.7 ± 15.8	308.5 ± 18.3
	EfficientNet-B0	433.9 ± 37.0	474.1 ± 6.6	396.2 ± 57.9
	RegNet-400MF	220.2 ± 7.0	235.5 ± 17.1	250.1 ± 5.7
Cats vs. Dogs	ResNet-18	699.9 ± 14.2	708.8 ± 27.8	723.1 ± 22.2
	MobileNet-V2	607.0 ± 8.2	473.8 ± 18.4	513.4 ± 40.7
	EfficientNet-B0	616.2 ± 59.7	616.8 ± 61.0	603.0 ± 38.1
	RegNet-400MF	352.3 ± 17.4	355.3 ± 17.4	381.8 ± 7.9

(Continued)

Table 9 (continued)

Dataset	Backbone	Linear	MLP-SM	MLP-LG
Solar Dust	ResNet-18	298.2 ± 8.8	286.1 ± 16.3	286.5 ± 19.1
	MobileNet-V2	266.3 ± 6.6	251.7 ± 19.3	257.0 ± 15.4
	EfficientNet-B0	275.6 ± 22.2	269.0 ± 7.1	276.2 ± 9.1
	RegNet-400MF	220.8 ± 20.3	221.7 ± 7.1	213.1 ± 4.9

Table 10: Quantum training time (s) across datasets, backbones, and heads. Values reported as mean ± std over 5 random seeds.

Dataset	Backbone	PL-Ideal	PL-Noisy	QK-Ideal	QK-Noisy
Hymenoptera	ResNet-18	179.0 ± 0.6	1022.3 ± 58.3	214.9 ± 1.5	188.3 ± 14.9
	MobileNet-V2	161.9 ± 5.8	959.1 ± 11.4	204.3 ± 11.7	137.6 ± 2.3
	EfficientNet-B0	167.2 ± 3.4	969.5 ± 5.7	228.5 ± 2.5	156.4 ± 1.9
	RegNet-400MF	145.9 ± 16.0	926.2 ± 13.2	176.4 ± 4.7	122.4 ± 1.1
Brain Tumor	ResNet-18	869.5 ± 51.7	5023.8 ± 271.7	1098.8 ± 3.7	857.8 ± 83.5
	MobileNet-V2	768.5 ± 25.7	4760.3 ± 64.3	970.9 ± 34.7	686.8 ± 13.8
	EfficientNet-B0	794.8 ± 36.0	4811.9 ± 60.2	1025.2 ± 29.6	736.4 ± 3.9
	RegNet-400MF	645.9 ± 27.7	4607.7 ± 65.5	880.4 ± 14.7	638.2 ± 81.0
Cats vs. Dogs	ResNet-18	1279.5 ± 22.5	7630.4 ± 294.1	1580.9 ± 91.5	1201.4 ± 15.4
	MobileNet-V2	1112.9 ± 23.2	7091.9 ± 32.0	1406.0 ± 75.4	1101.0 ± 146.0
	EfficientNet-B0	1195.7 ± 34.9	7217.7 ± 60.4	1602.7 ± 52.4	1221.4 ± 157.7
	RegNet-400MF	967.5 ± 31.6	7582.9 ± 91.4	1322.2 ± 51.2	1002.7 ± 116.5
Solar Dust	ResNet-18	413.9 ± 25.8	1664.2 ± 72.0	452.1 ± 29.9	397.6 ± 28.9
	MobileNet-V2	385.1 ± 36.0	1579.6 ± 55.7	443.1 ± 9.7	340.8 ± 6.5
	EfficientNet-B0	375.0 ± 6.2	1733.3 ± 37.2	453.8 ± 23.3	356.2 ± 2.5
	RegNet-400MF	360.3 ± 16.2	1541.4 ± 46.8	394.4 ± 19.6	322.6 ± 6.6

The training times show a large difference in resource requirements between classical and quantum classifiers. Classical heads complete training in less than 730 s. In contrast, quantum simulations require significantly more resources due to the overhead of simulating quantum states.

For the ideal configurations, PennyLane is faster than Qiskit. On the Brain Tumor dataset with the ResNet-18 backbone, PL-Ideal completes training in 869.5 ± 51.7 s, while qk-ideal requires 1098.8± 3.7 s. Both frameworks use the SPSA gradient estimator, so this difference is attributable to the simulation backend rather than to the optimizer. PennyLane's performs exact statevector evolution with minimal Python overhead per circuit evaluation, whereas Qiskit's primitive introduces an additional layer of abstraction (transpilation, observable handling, and primitive-level dispatch) that adds latency on circuits as small as four qubits.

For the noisy configurations, the relative performance is reversed. The noisy Qiskit simulator (QK-Noisy) is faster than the ideal Qiskit simulator (QK-Ideal) in several configurations. On the Hymenoptera dataset with the ResNet-18 backbone, qk-noisy requires 188.3 ± 14.9 s, while qk-ideal requires 214.9± 1.5 s. Because both configurations share the same SPSA optimizer and therefore the same

number of circuit evaluations per training step, this difference is also attributable to the simulation backend. QK-Ideal relies on the Python-based `StatevectorEstimator`, which computes exact expectation values from the full statevector at each call. QK-Noisy relies on the `AerEstimatorV2` primitive, which is backed by the C++ Aer simulator and is highly optimized for shot-based and noisy density-matrix simulations. For circuits of only four qubits, the C++ runtime dominates the per-evaluation cost, so QK-Noisy runs faster than QK-Ideal despite the additional cost of sampling 1024 shots and applying the noise channels.

4.5.2 Statistical Analysis

To evaluate whether statistically significant performance differences exist among the evaluated configurations, we applied the Friedman test, a non-parametric procedure for comparing multiple classifiers across multiple datasets simultaneously. This test was selected following the recommendations of Demšar [36], who established it as the standard methodology for multi-classifier benchmarking in machine learning research.

For each of the 16 dataset–backbone combinations (4 datasets × 4 backbone architectures), repeated executions under different random seeds were first averaged to obtain a single representative performance estimate per condition, avoiding pseudoreplication. The Friedman test then ranks the seven evaluated classification heads (Linear, MLP-SM, MLP-LG, PL-Ideal, PL-Noisy, QK-Ideal, and QK-Noisy) within each condition and assesses whether the observed rank distributions deviate significantly from the null hypothesis that all configurations achieve equivalent performance.

The Friedman test revealed statistically significant differences among the evaluated configurations ($\chi^2(6) = 40.43$, $p = 3.75 \times 10^{-7}$). Consequently, post-hoc pairwise comparisons were performed using the Nemenyi procedure. The critical difference (CD) at significance level $\alpha = 0.05$ was computed as

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{31}$$

where $k = 7$ is the number of configurations, $N = 16$ is the number of experimental conditions, and q_α is the critical value from the Studentized range distribution. For the present experimental design, the resulting critical difference was $CD = 2.25$.

Table 11 reports the average ranks obtained by each configuration. Rank 1 corresponds to the best-performing method. The last column indicates the statistical groups obtained from the Nemenyi post-hoc analysis. Configurations sharing at least one group label are not significantly different from each other, whereas configurations that do not share any group label exhibit statistically significant differences at $\alpha = 0.05$.

The results reveal three distinct performance tiers. The highest-ranked group (Group A) comprises MLP-SM, PL-Ideal, and MLP-LG. The differences among these configurations are smaller than the critical difference threshold and therefore cannot be considered statistically significant. This result indicates that the ideal PennyLane implementation achieves a level of predictive performance comparable to the strongest classical baselines. Importantly, although PL-Ideal obtains one of the best average ranks, the statistical analysis does not support the claim of a consistent quantum advantage over the best-performing classical alternatives.

Linear occupies an intermediate position and belongs simultaneously to Groups A and B. This overlap indicates that its performance cannot be statistically distinguished from either the best-performing methods or from QK-Ideal. Such behaviour is consistent with the overall results reported in Tables 7 and 8, where the linear baseline already achieves highly competitive performance across most datasets and backbone architectures.

Table 11: Average ranks from the Friedman test for 16 dataset-backbone conditions ($N = 16$, $k = 7$). Rank 1 indicates best performance. Configurations sharing at least one group label are not significantly different according to the Nemenyi post-hoc test ($\alpha = 0.05$, $CD = 2.25$).

Configuration	Type	Average Rank	Group
MLP-SM	Classical	2.59	A
PL-Ideal	Quantum (ideal)	2.72	A
MLP-LG	Classical	3.16	A
Linear	Classical	3.22	A-B
QK-Ideal	Quantum (ideal)	4.94	B
PL-Noisy	Quantum (noisy)	5.69	C
QK-Noisy	Quantum (noisy)	5.69	C

QK-Ideal forms Group B together with Linear. Although its average rank remains relatively competitive, it is consistently lower than those obtained by PL-Ideal and the strongest classical configurations. This result suggests that the effectiveness of ideal quantum representations is partially dependent on the implementation framework and the optimisation dynamics associated with each software stack.

The lowest-ranked configurations are PL-Noisy and QK-Noisy, which form Group C. Neither noisy configuration shares a statistical group with the highest-ranked methods, indicating a significant degradation in predictive performance when realistic quantum noise is introduced. This degradation is consistently observed for datasets and backbone architectures and supports the conclusion that noise remains one of the principal limitations of current QML systems.

4.5.3 Sensitivity Analysis

To assess how the choice of qubit count and circuit depth affects the ideal-regime accuracy, a sensitivity study is conducted on the Hymenoptera dataset with the ResNet-18 backbone, sweeping $n \in \{4, 8, 16\}$ qubits and $D \in \{1, 3, 5\}$ layers. Due to space limitations, this section only reports the results for the Hymenoptera case with ResNet-18. The corresponding accuracies are summarized in Table 12. Results for the remaining datasets and backbones are reported in the Supplementary Material (Section Tables S16–S31).

Table 12: Senstitivy analysis: test accuracy (%) vs. number of qubits and circuit depth (hymenoptera, ResNet-18, seed 42).

Framework	Qubits	Depth = 1	Depth = 3	Depth = 5
PL-Ideal	4	96.2	94.5	98.7
	8	96.2	96.2	97.4
	16	98.7	97.4	96.2
QK-Ideal	4	47.4	47.4	47.4
	8	92.3	89.8	87.2
	16	91.0	50.0	52.6

For PL-ideal, the accuracy remains within a narrow band between 94.5% and 98.7% across the entire grid. This indicates that the PennyLane stack is comparatively insensitive to the specific (n, D) choice within

the explored range, which is consistent with the well-connected ring Controlled NOT (CNOT) entanglement and the Pauli-Z expectation readout employed in this framework.

For QK-Ideal, by contrast, the behavior is markedly less uniform. Several configurations remain within a comparable accuracy band (for example, $4q/D = 3 \rightarrow 89.8\%$, $8q/D = 1 \rightarrow 92.3\%$, and $16q/D = 1 \rightarrow 91.0\%$), but other points collapse to near-chance performance for a binary task (around 47%–52%), namely $4q/D = 1$, $4q/D = 5$, $8q/D = 3$, $16q/D = 3$, and $16q/D = 5$.

These results show that increasing the number of qubits or the circuit depth does not lead to consistent accuracy improvements within the explored range. In PennyLane, the accuracy remains relatively stable across the grid, while in Qiskit several configurations collapse to near-chance performance, especially at some deeper or wider settings. Therefore, the four-qubit, depth-three configuration used in the main experiments was retained as a light, hardware-compatible operating point rather than a fully optimized architecture. Also note that similar conclusions can be drawn when analyzing the results reported in the Supplementary Material.

4.5.4 Noise Decomposition Study

To identify which physical noise channel contributes most to the degradation observed in the pl-noisy and qk-noisy models, an ablation is performed in which the three channels (amplitude damping, phase damping, and depolarizing) are activated one at a time. The single-channel and baselines for all four datasets are reported in Table 13, where each value is averaged on the four backbones to summarize the channel behavior independently of the specific feature extractor.

Table 13: Impact of discrete quantum noise channels on accuracy in for both PL-Noisy and QK-Noisy. Each single-channel row reports the accuracy (%) when only one of the three noise channels is active. The baseline row reports the accuracy of the noisy models with the three channels active simultaneously. All values are averaged for the four backbones (ResNet-18, MobileNet-V2, EfficientNet-B0, and RegNet-400MF).

Dataset	Active Noise Channel	Accuracy (PL-Noisy)	Accuracy (QK-Noisy)
Hymenoptera	Amplitude damping	95.8	96.8
	Phase damping	95.2	95.2
	Depolarizing	95.2	84.3
	Baseline (all three channels)	91.0	84.2
Brain Tumor	Amplitude damping	96.4	96.4
	Phase damping	96.9	95.9
	Depolarizing	95.2	96.9
	Baseline (all three channels)	96.3	95.8
Cats vs. Dogs	Amplitude damping	97.0	98.8
	Phase damping	97.0	98.6
	Depolarizing	98.2	98.6
	Baseline (all three channels)	98.4	95.4
Solar Dust	Amplitude damping	79.4	82.9
	Phase damping	77.9	74.6
	Depolarizing	81.2	84.2
	Baseline (all three channels)	77.4	80.2

The magnitude of the per-channel effect is strongly dataset-dependent, as Table 13 shows. On Brain Tumor and Cats vs. Dogs, the baseline noisy configurations already achieve accuracies close to those obtained under ideal simulation, and activating each channel independently produces only minor variations. This indicates that the visual representations extracted by the pretrained backbones remain sufficiently separable to tolerate moderate levels of quantum noise without a substantial loss of predictive performance.

In contrast, the impact of individual channels is more pronounced on Hymenoptera and Solar Dust, which are also the datasets where the noisy configurations exhibit the largest degradation relative to their ideal counterparts. For Hymenoptera, all three isolated channels achieve accuracies above 95% in PennyLane, whereas the combined-noise baseline drops to 91.0%. Similarly, in Solar Dust, the isolated-channel accuracies range from 77.9% to 81.2%, while the complete noisy model reaches only 77.4%. These results suggest that the degradation cannot be attributed to a single dominant source of noise, but rather to the cumulative interaction of multiple error mechanisms acting simultaneously throughout the variational circuit.

The comparison between PennyLane and Qiskit reveals a broadly consistent qualitative behavior. In both frameworks, the datasets that are intrinsically easier to separate, such as Brain Tumor and Cats vs. Dogs, remain largely insensitive to the activation of individual channels. However, some framework-specific differences emerge. For example, depolarizing noise produces a considerably larger degradation on Hymenoptera in Qiskit than in PennyLane (84.3% vs. 95.2%), suggesting that the shot-based sampling and measurement strategy employed by the Qiskit implementation may amplify the effect of stochastic gate errors in certain optimization regimes. Conversely, for Solar Dust, the Qiskit implementation appears slightly more resilient to amplitude damping and depolarizing noise, achieving accuracies above 82% in both cases.

When averaged on all datasets, no single channel consistently emerges as the dominant source of degradation in either framework. Phase damping appears particularly harmful on Solar Dust, while depolarizing noise produces the strongest effect on Hymenoptera within the Qiskit implementation. On the other hand, Brain Tumor exhibits only marginal sensitivity to any individual channel. This heterogeneous behavior indicates that the influence of quantum noise depends not only on the physical error mechanism itself but also on the structure of the feature space generated by the classical backbone and on the difficulty of the target classification task.

These results suggest that the largest performance losses arise from the simultaneous action of multiple noise processes rather than from any isolated channel. Consequently, effective error-mitigation strategies for hybrid quantum transfer learning should target the combined effect of decoherence, stochastic gate errors, and measurement uncertainty instead of focusing exclusively on a single physical source of noise.

4.5.5 Real Quantum Hardware Results

The hybrid model is executed on real IBM quantum hardware using the Hymenoptera dataset on the `ibm_torino` processor, with both frameworks trained for 10 epochs. Dynamical decoupling has been enabled on `ibm_torino` for both frameworks, matching the mitigation scheme already applied in the noisy emulations. Table 14 reports the test accuracy obtained by both frameworks in the three execution regimes considered in this study: ideal simulation, noisy simulation, and real hardware, with the simulation values taken from Table 8 and the real-hardware values taken from the corresponding runs on `ibm_torino`.

For the Qiskit implementation, the best-performing backbone on real hardware (MobileNet-V2, 95.32 ± 0.79%) closely matches the corresponding ideal simulation (97.2 ± 1.4%, a drop of 1.9 points), and a comparable behavior is observed for EfficientNet-B0. On ResNet-18 and RegNet-X-400MF, however, the real-hardware mean exceeds the ideal-simulation mean. This inversion is not a hardware advantage: as reported in the sensitivity analysis, several SPSA initializations of QK-IDEAL on these two backbones converge

to near-chance regimes, which depress the simulator mean and inflate its standard deviation (up to ±22.0 points on RegNet-X-400MF). The real-hardware runs, in contrast, exhibit small standard deviations (below 1.7 points on every backbone), and the across-backbone variation observed on real hardware (74.62–95.32%) is comparable in span to the variation observed in simulation but with markedly lower run-to-run dispersion.

Table 14: Comparison of test accuracy (%) across execution regimes (hymenoptera, ibm_torino, 10 epochs). Values reported as mean ± std over 5 random seeds.

Backbone	PL-Ideal	PL-Noisy	Real HW	QK-Ideal	QK-Noisy	Real HW
MobileNet-V2	98.2 ± 0.7	95.4 ± 3.9	**88.45 ± 1.84**	97.2 ± 1.4	96.9 ± 1.9	**95.32 ± 0.79**
ResNet-18	96.4 ± 1.4	90.3 ± 4.8	86.32 ± 2.08	84.4 ± 14.9	82.6 ± 21.3	92.15 ± 1.08
RegNet-X-400MF	97.7 ± 2.3	82.1 ± 18.4	81.24 ± 2.65	79.5 ± 22.0	87.4 ± 19.8	87.43 ± 1.68
EfficientNet-B0	96.9 ± 0.7	96.2 ± 1.6	68.42 ± 3.05	96.2 ± 1.8	95.6 ± 2.7	74.62 ± 2.35

For the PennyLane implementation, the ideal-simulation results are uniformly high across backbones, with MobileNet-V2 reaching 98.2 ± 0.7% and ResNet-18 reaching 96.4 ± 1.4%. The transition to real hardware leads to substantially larger accuracy drops than those observed with Qiskit: MobileNet-V2 falls from 95.4 ± 3.9% in noisy simulation to 88.45 ± 1.84% on real hardware, and EfficientNet-B0 falls from 96.2 ± 1.6% to 68.42 ± 3.05%. This indicates that, with the present training budget, the PennyLane pipeline is more sensitive to hardware-induced degradation than the Qiskit pipeline.

The cross-framework comparison on real hardware shows that the PennyLane runs underperform the Qiskit runs on every backbone, with backbone-to-backbone gaps in a narrow range of 5.8 to 6.9 percentage points. This consistent gap indicates that the Qiskit implementation is more resilient to hardware-induced degradation than the PennyLane implementation in this experimental setup.

Table 15 reports the training times measured on `ibm_torino` for both frameworks. The Qiskit configuration completes training in an average of 575 s per run (range 508–684 s), while the PennyLane configuration takes an average of 686 s per run (range 676–701 s). These figures indicate that real-hardware training, while substantially slower than simulation, remains practically feasible for moderate-sized datasets.

Table 15: Training times (s) on real quantum hardware (hymenoptera, `ibm_torino`, 10 epochs). Results reported as mean ± std over 5 random seeds.

Backbone	PennyLane	Qiskit
MobileNet-V2	688 ± 20	562 ± 13
ResNet-18	701 ± 18	684 ± 17
RegNet-X-400MF	676 ± 16	508 ± 12
EfficientNet-B0	679 ± 22	548 ± 15

4.5.6 Quantum Computing Limitations

The deployment of variational classifier circuits on Heron r2 processors such as `ibm_torino` is constrained by the connectivity of the physical device. The Heron r2 layout follows a heavy-hex coupling map, which does not natively support the entanglement patterns used by either of the two frameworks evaluated in this study: PennyLane employs a ring CNOT pattern that closes the loop across the four qubits, and Qiskit employs an alternating brick-wall pattern that pairs qubits in two-by-two blocks. In both cases, the transpiler inserts additional two-qubit `SWAP` operations to route the quantum state across the available couplers, which

inflates both the gate count and the executed circuit depth. As reported in Table 6, the four-qubit, depth-three virtual circuit is transpiled to a physical depth of approximately 49, an order-of-magnitude expansion that propagates the per-gate error budget across a much longer execution path and reduces the estimated end-to-end fidelity of the variational head. The four-qubit register adopted in the main experiments mitigates this expansion by keeping the input dimensionality of the variational circuit small, so that the depth and gate count after transpilation remain within a range that current NISQ devices can execute with usable fidelity.

A second well-known limitation of variational quantum classifiers is the appearance of barren plateaus during training, that is, regions of the parameter space where the variance of the cost-function gradient vanishes exponentially as the number of qubits grows [34,35]. The two factors relevant to the present setup are the use of entangling layers and the action of physical noise channels in the noisy regime, while the observables employed in both frameworks are local Pauli-Z projectors and therefore do not introduce the additional flattening associated with global observables. The four-qubit, depth-three configuration adopted in the main experiments sits well below the qubit count at which barren plateaus typically manifest empirically, and the sensitivity analysis reported in Section 4.5 provides indirect evidence consistent with this picture: QK-IDEAL converges reliably at four qubits but collapses to near-chance accuracy at several eight- and sixteen-qubit configurations, a pattern compatible with the flattening of the optimization landscape that is associated with vanishing-gradient regimes. While a direct measurement of gradient variances across qubit counts is not reported here, this indirect evidence supports the choice of a lightweight four-qubit variational head as a regime in which trainability is preserved.

5 Conclusions

A methodological benchmark for hybrid CQ TL has been proposed in this paper. The benchmark couples four frozen pretrained convolutional backbones to seven classification heads, namely three classical baselines and four hybrid quantum heads implemented in PennyLane and Qiskit under ideal and IBM Heron r2-calibrated noisy emulation, and the resulting grid has been applied to four heterogeneous binary image datasets and executed on the `ibm_torino` processor. Three findings can be highlighted. First, ideal quantum heads reach accuracies that are statistically comparable to classical baselines of matched parameter count. Second, the noisy emulation produces a significant degradation, with phase damping showing the largest aggregate impact among the channels analyzed. Third, on real quantum hardware, the Qiskit pipeline is more resilient than the PennyLane pipeline, while PennyLane remains faster in ideal simulation and Qiskit becomes faster in noisy simulation due to its C++ Aer backend.

The conclusions reported here apply to binary image classification with four-qubit, depth-three variational heads compatible with current NISQ hardware. The real-hardware campaign has been centered on a single dataset and a single Heron r2 processor, and the per-channel noise decomposition has been performed on the PennyLane mixed-state device, where individual Kraus channels can be inserted directly.

Several lines of research are planned for the future. Beyond the dynamic decoupling applied in noisy simulations and on real hardware for both frameworks, additional quantum error mitigation strategies, such as noise-free extrapolation and probabilistic error cancellation, will be incorporated into noisy workflows, along with error-sensitive training schemes. Future hardware implementations at IBM will take into account hardware-sensitive ansatz design and qubit layout selection to reduce routing overhead, SWAP insertion, and the depth of the transpilated circuit. Since degradation in real hardware is not solely explained by SPSA stochasticity, future work will also explore adaptive shot allocation, noise-resistant optimizer settings, and alternative gradient estimation strategies, such as parameter shift estimators, when latency and shot budgets permit. Finally, the framework's scalability will be evaluated on wider registers and deeper circuits, and the

benchmark will be extended to multi-class and non-visual tasks, as well as additional quantum backbones and backends.

Acknowledgement: The Junta de Andalucía's Centro de Informática Científica de Andalucía (CICA) is acknowledged for providing the computing resources on the Hércules supercomputer, which were instrumental in performing the simulations/calculations presented in this work. This work has been carried out under the initiative of the Agencia Digital de Andalucía (ADA), making use of infrastructure provided by Centro de Inteligencia Artificial de Andalucía (ANIA).

Funding Statement: The authors would like to thank the Spanish Ministry of Science and Innovation for the support under the projects PID2023-146037OB-C21 and PID2023-146037OB-C22 funded by MICIU/AEI/10.13039/501100011033. We also acknowledge Pablo de Olavide University for funding the project PPI2505.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Daniel Martín-Pérez and Francisco Martínez-Álvarez; methodology, Daniel Martín-Pérez and Francisco Martínez-Álvarez; software, Daniel Martín-Pérez; validation, David Gutiérrez-Avilés and Alicia Troncoso; formal analysis, David Gutiérrez-Avilés and Francesc Rodríguez-Díaz; investigation, Daniel Martín-Pérez and Francesc Rodríguez-Díaz; resources, Alicia Troncoso and Francisco Martínez-Álvarez; writing—original draft preparation, Daniel Martín-Pérez and Francesc Rodríguez-Díaz; writing—review and editing, David Gutiérrez-Avilés and Francesc Rodríguez-Díaz; supervision, David Gutiérrez-Avilés and Francisco Martínez-Álvarez; funding acquisition, Alicia Troncoso and Francisco Martínez-Álvarez. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The complete source code developed for this study, including training scripts, preprocessing pipelines, model architectures, and yaml configuration files, is available in the public repository: https://github.com/Data-Science-Big-Data-Research-Lab/QTL. The datasets analyzed in this study are public and can be accessed using the download scripts and instructions provided in the repository.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmes.2026.082712/s1.

References

1. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA, USA: MIT Press; 2016.
2. Schwartz R, Dodge J, Smith JA, Etzioni O. Green AI. Commun ACM. 2020;63(12):54–63.
3. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345–59. doi:10.1109/tkde.2009.191.
4. Dwivedi P, Kajal M. Efficient deep learning training: an energy-conscious adaptive framework. Intl J Data Sci Anal. 2025;20:4741–55.
5. Rodríguez-Díaz F, Gutiérrez-Avilés D, Troncoso A, Martínez-Álvarez F. A survey of quantum machine learning: foundations, algorithms, frameworks, data and applications. ACM Comput Surv. 2025;58(4):1–35.
6. Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018;2:79.
7. Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, et al. Noisy intermediate-scale quantum (NISQ) algorithms. Rev Mod Phys. 2022;94(1):015004. doi:10.1103/revmodphys.94.015004.
8. Bergholm V, Izaac J, Schuld M, Gogolin C, Ahmed S, Ajith V, et al. PennyLane: automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968. 2018.
9. IBM Quantum. Qiskit: an open-source framework for quantum computing. IBM; 2017 [cited 2026 May 20]. Available from: https://qiskit.org/.

10. Mari A, Bromley TR, Izaac J, Schuld M, Killoran N. Transfer learning in hybrid classical-quantum neural networks. Quantum. 2020;4:340.
11. Mogalapalli H, Abburi M, Nithya B, Bandreddi SKV. Classical-quantum transfer learning for image classification. SN Comput Sci. 2022;3(1):20.
12. Otgonbaatar S, Schwarz G, Datcu M, Kranzlmüller D. Quantum transfer learning for real-world, small, and high-dimensional remotely sensed datasets. IEEE J Select Topics Appl Earth Observ Remote Sens. 2023;16:9223–30. doi:10.1109/jstars.2023.3316306.
13. Yogaraj K, Quanz B, Vikas T, Mondal A, Mondal S. Post-variational classical quantum transfer learning for binary classification. Sci Rep. 2025;15(1):23682. doi:10.1038/s41598-025-08887-2.
14. Khatun A, Usman M. Quantum transfer learning with adversarial robustness for classification of high-resolution image datasets. Adv Quantum Technol. 2025;8(1):2400268. doi:10.1002/qute.202400268.
15. Kim J, Huh J, Park DK. Classical-to-quantum convolutional neural network transfer learning. Neurocomputing. 2023;555(4):126643. doi:10.1016/j.neucom.2023.126643.
16. Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A, Woerner S. The power of quantum neural networks. Nat Comput Sci. 2021;1(6):403–9. doi:10.1038/s43588-021-00084-1.
17. Schuld M, Killoran N. Quantum machine learning in feature hilbert spaces. Phys Rev Lett. 2019;122(4):040504. doi:10.1103/physrevlett.122.040504.
18. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. Nature. 2019;567(7747):209–12. doi:10.1038/s41586-019-0980-2.
19. Mitarai K, Negoro M, Kitagawa M, Fujii K. Quantum circuit learning. Phys Rev A. 2018;98(3):032309.
20. Azevedo V, Silva C, Dutra I. Quantum transfer learning for breast cancer detection. Quantum Mach Intell. 2022;4(1):5. doi:10.1007/s42484-022-00062-4.
21. Acar E, Yilmaz I. COVID-19 detection on IBM quantum computer with classical-quantum transfer learning. Turkish J Electr Eng Comput Sci. 2021;29(1):46–61.
22. Bali M, Mishra VP, Yenkikar A, Chikmurge D. QuantumNet: an enhanced diabetic retinopathy detection model using classical deep learning-quantum transfer learning. MethodsX. 2025;14:103185.
23. Mir A, Yasin U, Khan SN, Athar A, Jabeen R, Aslam S. Diabetic retinopathy detection using classical-quantum transfer learning approach and probability model. Compu, Mat Contin. 2022;71(2):3733–54. doi:10.32604/cmc.2022.022524.
24. Kumsetty NV, Nekkare AB, Kamath SS, Kumar A. TrashBox: trash detection and classification using quantum transfer learning. In: Proceedings of the 31st Conference of Open Innovations Association. Piscataway, NJ, USA: IEEE; 2022. p. 125–30.
25. Geng A, Moghiseh A, Redenbach C, Schladitz K. Hybrid quantum transfer learning for crack image classification on NISQ hardware. AIP Conf Proc. 2025;3182:130001. doi:10.1063/5.0246500.
26. Kati BE, Küçüksille EU, Sariman G. Enhancing deepfake detection through quantum transfer learning and class-attention vision transformer architecture. Appl Sci. 2025;15(2):525.
27. Qi J, Tejedor J. Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway, NJ, USA: IEEE; 2022. p. 8627–31.
28. Wang T, Wang H, Chen Z, Yamagishi J. Quantum transfer learning using the large-scale unsupervised pre-trained model WavLM-large for synthetic speech detection. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway, NJ, USA: IEEE; 2023. p. 1–8.
29. Koike-Akino T, Wang P, Wang Y. Quantum transfer learning for Wi-Fi sensing. In: Proceedings of the IEEE International Conference on Communications Workshops. Piscataway, NJ, USA: IEEE; 2022. p. 1–8.
30. Buonaiuto G, Guarasci R, Minutolo A, De Pietro G, Esposito M. Quantum transfer learning for acceptability judgements. Quantum Mach Intell. 2024;6(1):13.
31. Vermeire FH, Green WH. Transfer learning for solvation free energies: from quantum chemistry to experiments. Chem Eng J. 2021;418:129307.

32. Zen R, My L, Tan R, Hébert F, Gattobigio M, Miniatura C, et al. Transfer learning for scalability of neural-network quantum states. Phys Rev E. 2020;101(5):053301.
33. Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L, et al. Noise-induced barren plateaus in variational quantum algorithms. Nat Commun. 2021;12(1):6961. doi:10.1038/s41467-021-27045-6.
34. McClean JR, Boixo S, Smelyanskiy VN, Babbush R, Neven H. Barren plateaus in quantum neural network training landscapes. Nat Commun. 2018;9(1):4812.
35. Cerezo M, Sone A, Volkoff T, Cincio L, Coles PJ. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun. 2021;12(1):1791. doi:10.1038/s41467-021-21728-w.
36. Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Lear Res. 2006;7:1–30.
37. Bowles J, Ahmed S, Schuld M. Better than classical? The subtle art of benchmarking quantum machine learning models. arXiv:2403.07059. 2024.
38. Shahzad MA. Hybrid Quantum-classical model for image classification. arXiv:2509.13353. 2025.
39. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: NIPS'14: Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge, MA, USA: MIT Press; 2014. p. 3320–8.
40. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. Proc IEEE. 2021;109(1):43–76.
41. Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2009. p. 248–55.
42. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. Intl J Comput Vis. 2015;115(3):211–52. doi:10.1007/s11263-015-0816-y.
43. Schuld M, Sinayskiy I, Petruccione F. Simulating a Perceptron on a quantum computer. Phys Lett A. 2015;379(7):660–3.
44. Sim S, Johnson PD, Aspuru-Guzik A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv Quantum Technol. 2019;2(12):1900070. doi:10.1002/qute.201900070.
45. Farhi E, Neven H. Classification with quantum neural networks. arXiv:1802.06002. 2018.
46. Qiskit Machine Learning contributors. Qiskit machine learning. GitHub Repository; 2023 [cited 2026 May 23]. Available from: https://github.com/qiskit-community/qiskit-machine-learning.
47. Qiskit Aer contributors. Qiskit Aer: a high-performance simulator for quantum circuits. GitHub Repository; 2024 [cited 2026 May 23]. Available from: https://github.com/Qiskit/qiskit-aer.
48. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2016. p. 770–8.
49. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2018. p. 4510–20.
50. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning. Cambridge, MA, USA: PMLR; 2019. p. 6105–14.
51. Radosavovic I, Kosaraju RP, Girshick R, He K, Dollár P. Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2020. p. 10428–36.
52. TheDataSith. Hymenoptera dataset. Kaggle; 2021 [cited 2026 May 23]. Available from: https://www.kaggle.com/datasets/thedatasith/hymenoptera.
53. Nickparvar M. Brain tumor MRI dataset. Kaggle; 2021 [cited 2026 May 23]. Available from: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.
54. Cukierski W. Dogs vs. cats dataset. Kaggle competition; 2013 [cited 2026 May 23]. Available from: https://www.kaggle.com/c/dogs-vs-cats.
55. Garladinne-Hemanth S. Solar panel dust detection dataset. Kaggle; 2023 [cited 2026 May 23]. Available from: https://www.kaggle.com/datasets/hemanthsai7/solar-panel-dust-detection