Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Pixel-Translation-Equivariant Quantum Convolutional Neural Networks via Fourier Multiplexers

Dmitry Chirkov

dmitrii.chirkov@metalab.ifmo.ru

ITMO University

Igor Lobanov

ITMO University

Research Article

Keywords:

Posted Date: August 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10574918/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Pixel-Translation-Equivariant Quantum Convolutional Neural Networks via Fourier Multiplexers

Dmitry Chirkov[1*] and Igor Lobanov[1]

[1]School of Physics and Engineering, ITMO University, St. Petersburg, Russia .

*Corresponding author(s). E-mail(s): dmitrii.chirkov@metalab.ifmo.ru;

Abstract

Symmetry is a central source of inductive bias in learning, but in quantum models the symmetry available to a circuit is shaped by the data encoding. We show that this changes what it means for a quantum circuit to be convolutional. For address/amplitude image encodings such as FRQI, pixel translations on the finite periodic lattice are represented by pixel cyclic shifts (PCS), which act as modular addition on the spatial index register, whereas many MERA-inspired QCNN templates are equivariant under cyclic permutations of physical qubits. We formalize this mismatch between PCS and qubit cyclic shifts (QCS), and construct QCNN layers that commute exactly with the PCS symmetry induced by the encoding. We characterize the resulting layer family as the unitary commutant of the encoded translation operator, yielding a Fourier-space constructive form for PCS-equivariant quantum convolution. Building on this characterization, we construct a multiscale PCS-QCNN with measurement-conditioned pooling and analyze its trainability with gradient diagnostics that separate parameter-count effects from suppression of the optimization-relevant gradient norm. Exact PCS equivariance is enforced for the unitary quantum convolutional blocks on their active registers; pooling yields a multiscale/coarse equivariance structure, and in the reported hybrid classifier an unconstrained linear-softmax head maps readout probabilities to labels, so label invariance is learned rather than imposed. To make the empirical comparison diagnostic of convolutional inductive bias, we use a translated-MNIST benchmark with matched classical CNN/MLP controls that exhibit a separation between translation-aware and dense architectures. On this benchmark, the PCS-QCNN achieves higher accuracy than a matched non-PCS random-basis quantum control. Full-MNIST

experiments provide a non-translated reference, and finite-shot diagnostics identify a possible train–deploy mismatch at fixed shot budget, making sampling cost a deployment-relevant part of model selection.

1 Introduction

Symmetry is one of the most reliable ways to build inductive bias. In classical vision, convolutional neural networks (CNNs) hard-code translation equivariance by restricting layer maps to those that commute with pixel shifts (up to boundary conventions), and this architectural constraint is a central source of their sample efficiency and generalization. Classical literature and textbook treatments support this intuition: translation-aware architectural constraints and weight sharing can improve generalization and sample efficiency relative to fully connected alternatives on vision tasks [1, 2].

Quantum machine learning raises the same symmetry question in a setting where the data encoding is part of the model. Variational quantum circuits can be executed on noisy intermediate-scale quantum (NISQ) devices [3], and quantum convolutional neural networks (QCNNs) have been proposed as quantum analogues of CNNs, often inspired by multiscale tensor-network structures such as MERA [4, 5]. However, in the quantum setting the meaning of translation is not purely a property of the circuit: it is jointly determined by how classical data are encoded into a quantum state and which symmetry the circuit preserves.

Two strands of prior work set the context for this distinction. First, there is a growing empirical literature on quantum-convolution-style models for classical images, typically using hybrid models with classical preprocessing and relatively small quantum circuits. MNIST-like benchmarks in this literature use heterogeneous task definitions, resolutions, preprocessing choices, and readout protocols [6–12]; a broader overview of representative settings and reported accuracies is provided in Supplementary Table S1. Second, there is a symmetry-focused literature on equivariant quantum architectures, including QCNN variants designed to be equivariant under cyclic shifts, broader permutation groups, or more general task symmetries with resource-aware implementations [5, 13–17]. Taken together, these strands expose a conceptual ambiguity: for FRQI and related quantum image representations [18, 19], the spatial symmetry of the encoded state need not coincide with a symmetry of qubit labels. This ambiguity matters because a circuit can be symmetric in qubit space while failing to respect the translation symmetry of the encoded data. The evaluation in [20] was performed on standard benchmark datasets. However, such datasets are not necessarily suitable for evaluating the advantages of convolutional architectures, as they do not require exploiting translational symmetry. A similar issue is well known in classical machine learning [1, 2]: convolutional neural networks demonstrate their advantage over fully connected models primarily on tasks where the data distribution exhibits the corresponding spatial symmetries. Therefore, to properly assess convolutional quantum architectures, one should consider benchmarks specifically designed to test their

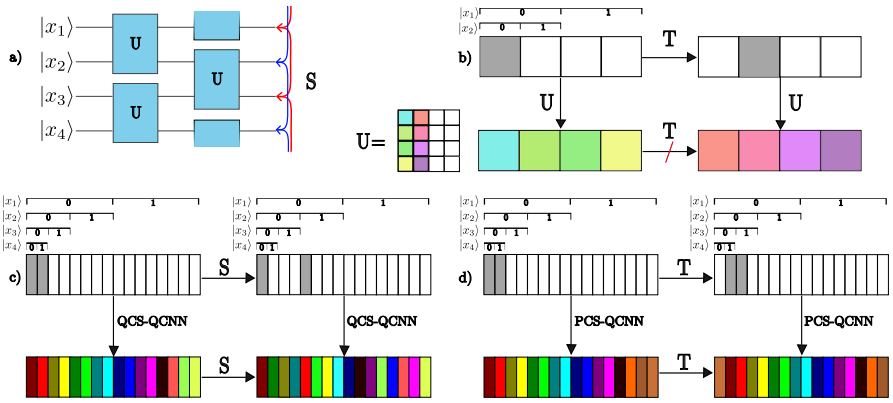


Fig. 1 Schematic comparison of qubit- vs pixel-translation symmetry for address/amplitude encodings. (a) A typical QCNN convolution pattern based on repeating the same local unitary on different qubit pairs (adapted from Ref. [21]). (b) Under address encoding, translating the input image corresponds to modular addition on the index register, so reusing the same local unitary across qubit positions does not, in general, enforce pixel-translation equivariance. (c) The same pattern can still be equivariant under cyclic permutations of qubits (QCS). (d) Required symmetry: equivariance under cyclic shifts of pixel indices (PCS), i.e., commutation with the translation action induced by the encoding.

ability to exploit such symmetries. In this work, we address this issue by using the translated MNIST dataset, where translational equivariance is essential for achieving good generalization.

The analysis focuses on amplitude/address-type image encodings, specifically FRQI-like states (Sec. 5.2). On the finite periodic lattice used for the exact symmetry statement, pixel translations are represented by pixel cyclic shifts (PCS), which act as modular addition on the binary-valued index register. By contrast, many existing QCNN designs enforce commutation with cyclic permutations of physical qubits (or, more generally, subgroups of the qubit permutation group) [13]. We refer to this register-level symmetry as the qubit cyclic shift (QCS). PCS and QCS coincide only for pixel-to-qubit encodings; under address encoding they generally act differently, so a QCS-equivariant circuit need not implement pixel-translation equivariance. Permutation-equivariant designs remain useful when the task symmetry can be represented as a qubit permutation under the chosen embedding, but for address encodings translation acts natively on the index register and should therefore be enforced directly at that level. Figure 1 illustrates this mismatch. The design principle is that quantum convolution should preserve the translation symmetry present after encoding, not merely a symmetry of qubit labels. Throughout the paper, "convolution" is used in this symmetry/commutant sense rather than in the finite-support-kernel or local-receptive-field sense.

We address this by separating QCS from PCS at the level of the encoded Hilbert space and formalizing when the two symmetries do, and do not, coincide (Sec. 2). Once PCS is identified as the relevant data symmetry, the allowed unitary layers are fixed by its commutant. Since the QFT diagonalizes translation operators, any PCS-equivariant layer can be represented by mapping the spatial index registers to the

Fourier basis, applying a block-diagonal transformation that acts independently on Fourier modes (a multiplexer), and returning to the computational basis with the IQFT. The Fourier diagonalization itself is standard; the key point is that, for address encodings, it must be applied to the pixel-shift operator on the index register rather than to a permutation of qubit labels. This yields a direct recipe for designing quantum convolution layers whose symmetry is enforced by construction. We then build a deep PCS-QCNN quantum core with measurement-induced pooling, where intermediate measurement outcomes select mode-dependent blocks in subsequent layers. The benchmark PCS layers use fully general translation-equivariant Fourier multiplexers; locality or finite spatial support would be an additional structural constraint rather than part of the definition studied here.

A separate practical concern in deep variational quantum models is trainability. For many circuit families, gradient magnitudes can decay rapidly with system size and/or depth (barren plateaus). This phenomenon is well documented for highly expressive random circuits [22], while QCNN-style locality can mitigate it in certain regimes [23]. For our multiplexer-based PCS-QCNN, the trainability analysis separates a mathematically elementary accounting effect from the practical optimization question (Sec. 3). Because a fully general Fourier multiplexer contains exponentially many mode-wise parameters in the depth-scaling family, the expected mean per-coordinate second moment can be forced down by parameter count alone. This coordinate-average effect is distinct from exponential suppression of the full gradient norm or of every active coordinate. The main results therefore report both the empirical-loss quantum-gradient norm and the per-sample RMS quantum-gradient norm.

MNIST is used as the benchmark for encoding-aligned translation equivariance (Sec. 4). A methodological requirement is that any claimed gain from quantum convolution should be visible in a matched classical comparison. Standard centered MNIST alone is a weak standalone diagnostic of convolutional inductive bias, because dense classical models can perform well under favorable preprocessing. We therefore use matched classical CNN/MLP controls to identify a translated-MNIST regime in which translation-aware and dense architectures separate, and then use the same regime to probe the effect of enforcing PCS symmetry in the quantum model against a matched non-PCS random-basis control. Full-MNIST experiments provide a non-translated reference for the architecture and preprocessing choices, while the translated-MNIST configuration in Fig. 2 provides the main benchmark for the symmetry-aligned control.

The numerical scope is as follows. The reported PCS-QCNN is an idealized symmetry benchmark: FRQI-like inputs are initialized directly as dense tensors, fully general Fourier multiplexers are applied as explicit block-diagonal statevector maps, and hardware-oriented structured/compressed PCS layers would define a different model class. Exact PCS equivariance is claimed for the unitary quantum blocks on their active registers; pooling and the unconstrained linear-softmax head do not impose exact one-pixel label invariance. The quantum comparison uses a matched non-PCS random-basis ablation rather than a QCS-equivariant baseline, and the finite-shot diagnostics use a single direct-16×16 full-MNIST reference run.

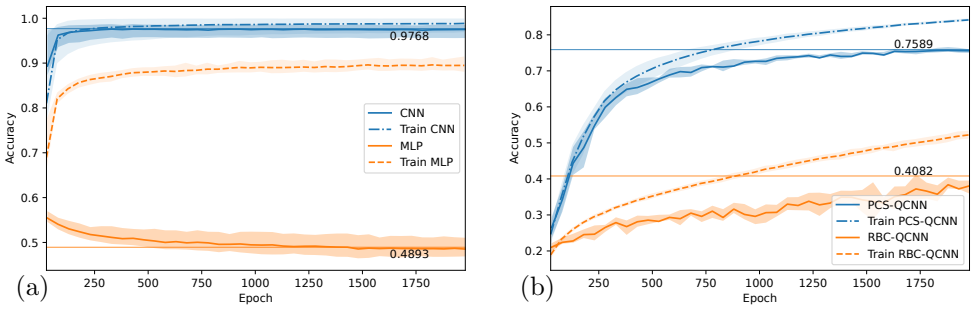


Fig. 2 Fixed benchmark controls used to expose translation-sensitive inductive bias. Training uses a balanced subset of 1000 images per class, and the standard MNIST test split membership is kept unchanged. Digits are resized from 28×28 to 16×16, placed on a 32×32 canvas, and randomly translated with maximum offset 8 pixels per axis. (a) Classical baselines on this translated benchmark: a convolutional CNN reference and a pure-dense MLP control. (b) Quantum models: PCS-QCNN and a matched non-PCS random-basis control (RBC) QCNN on the same translated benchmark. Solid lines show mean test accuracy, train curves show mean train accuracy, and shaded bands show the 25th–75th percentile range over 3 seeds. The corresponding full-MNIST classical control without translations is shown separately in Supplementary Fig. S3.

Finally, because the PCS-QCNN is evaluated through output probabilities, we test how conclusions drawn from exact statevector inference change under finite-shot sampling. Even on noiseless hardware, finite-shot sampling introduces stochasticity that can distort loss landscapes and change the effective performance of a trained model. Accordingly, we report both infinite-shot (exact) simulator accuracy and finite-shot inference. In the representative finite-shot diagnostic, continued infinite-shot training is associated with a sharper readout-space loss landscape and lower accuracy at some fixed shot budgets (Sec. 6.5).

2 Symmetry-aligned quantum convolution

A symmetry-based definition of convolution gives a common language for the classical and quantum settings. Throughout the paper we work with discrete images on a periodic lattice. We use a standard periodic model in which translation is represented by a cyclic shift, so that symmetry claims can be stated without boundary artifacts [24, 25].

2.1 Classical convolution and translation symmetry

The classical notion of convolution can be stated purely in symmetry terms: a linear layer is convolutional (in the circular/periodic sense) exactly when it commutes with discrete translations [24, 25]. This convention guarantees translation equivariance but does not by itself impose a finite-support spatial kernel. On a 1D periodic grid of length N, let T_k denote the cyclic shift, $(T_k x)_j = x_{j-k}$. A matrix A is circulant if and only if $T_k A = A T_k$ for all k, and in that case $y = Ax$ is a circular convolution with shared weights. The same perspective extends to multi-channel and multi-dimensional signals: translation-equivariant linear maps are precisely block-circulant operators.

A key structural fact is Fourier diagonalization: translations are diagonal in the discrete Fourier basis, so any translation-equivariant linear map becomes block-diagonal in Fourier space (each Fourier mode transforms independently on the channel space) [24, 25]. This is the classical template we mirror below: in the quantum construction, the QFT plays the role of the Fourier basis change and the multiplexer plays the role of per-mode channel mixing. For a more detailed classical primer (including multi-channel and 2D formulas), see Supplementary Section S3.

2.2 Quantum registers, image encodings, and two shift symmetries

We fix notation for the two symmetry operations already introduced in Fig. 1: the qubit cyclic shift S (QCS) and the pixel cyclic shift T (PCS). The purpose is to keep the classical and quantum cases parallel: in both settings convolution is defined by commutation with the translation action. The only difference is that in the quantum setting the translation action depends on the encoding.

We begin by specifying the register layout. We use an index register (or registers) to represent spatial coordinates and a feature register to store channels. In the 2D case we use registers $\mathrm{Reg}\, x$ and $\mathrm{Reg}\, y$ with n_x and n_y qubits, so that $N_x = 2^{n_x}$ and $N_y = 2^{n_y}$ pixels are represented along each axis. We use a color register $\mathrm{Reg}\, c$ (grayscale corresponds to one qubit, RGB to three qubits) and an auxiliary feature register $\mathrm{Reg}\, f$. In the canonical PCS-QCNN analyzed in this paper, the size of $\mathrm{Reg}\, f$ is fixed within a given architecture instance; allowing it to grow across layers is a possible extension discussed in Sec. 2.5.

Next we distinguish the relevant encoding families. The central distinction is whether pixels are mapped to qubits (threshold/pixel-to-qubit encodings) or to basis states of an index register (address or amplitude-based encodings). The analysis focuses on amplitude-based image encodings, where spatial structure is represented by the binary index. A convenient representative is an FRQI-like state in which the index registers encode coordinates while a color qubit carries grayscale information:

$$|\psi(x)\rangle = \frac{1}{\sqrt{N_x N_y}} \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} |u\rangle_x |v\rangle_y |\phi_{u,v}\rangle , \tag{1}$$

where $|\phi_{u,v}\rangle$ is a single-qubit state whose amplitudes encode pixel brightness. In the benchmark studied here, each preprocessed pixel is represented by a grayscale value $x_{u,v} \in [0,1]$ (see Supplementary Section S2.1.1 for the exact preprocessing convention). The encoder maps this value affinely to an angle $p_{u,v} = a + (b-a)x_{u,v}$; the PCS-QCNN experiments use $(a,b) = (0,\pi)$, so

$$|\phi_{u,v}\rangle = \sin(p_{u,v})\,|0\rangle + \cos(p_{u,v})\,|1\rangle . \tag{2}$$

Equation (1) is the conceptual normalized form; in the numerical protocol of Sec. 5.2 we use the same local feature states but omit the global $1/\sqrt{N_x N_y}$ prefactor during state initialization and restore the overall normalization when computing the final

readout probabilities. For the symmetry discussion, what matters is that translation of pixels acts as a permutation of the computational basis of the index register.

With this setup, the pixel cyclic shift (PCS) is defined as follows. In one spatial dimension (a signal of length $N = 2^n$), the pixel cyclic shift operator T acts on the index basis as modular addition:

$$T\,|j\rangle = |j+1 \pmod N\rangle\,. \tag{3}$$

In the classical case the same action on coordinates is $(Tx)_j = x_{j-1}$, consistent with the definition of T_k in Sec. 2.1. In two dimensions we use commuting generators T_x and T_y acting on the two index registers:

$$T_x\,|u\rangle_x\,|v\rangle_y = |u+1 \pmod{N_x}\rangle_x\,|v\rangle_y\,, \tag{4}$$

$$T_y\,|u\rangle_x\,|v\rangle_y = |u\rangle_x\,|v+1 \pmod{N_y}\rangle_y\,, \tag{5}$$

and similarly in three dimensions one introduces T_x, T_y, T_z acting on three index registers. These translations act trivially on the feature and color registers.

By contrast, the qubit cyclic shift (QCS) acts on tensor factors: S is defined as a cyclic permutation of tensor factors in a register of L physical qubits:

$$S\big(|q_0\rangle\,|q_1\rangle\cdots|q_{L-1}\rangle\big) = |q_{L-1}\rangle\,|q_0\rangle\cdots|q_{L-2}\rangle\,. \tag{6}$$

Many MERA-inspired QCNN templates can be made equivariant under this symmetry by imposing cyclic weight sharing together with periodic or symmetrized block placements and pooling/readout rules compatible with the same cyclic action. We will refer to commutation with S as qubit translational symmetry (QCS), and commutation with T (or with T_x, T_y) as pixel translational symmetry (PCS).

This gives a precise distinction between layer symmetry and data symmetry. The operator T represents the data symmetry, namely translation in pixel space, while the operator S represents a register symmetry, namely relabeling of qubits. A quantum circuit is convolutional for image data only if its symmetry matches the translation action induced by the chosen encoding. This distinction is the source of the mismatch addressed in this paper.

Lemma 1 (Mismatch between QCS and PCS)**.** *Let an image be encoded either (i) by a pixel-to-qubit map that assigns one pixel value to one qubit, or (ii) by an address/amplitude encoding in which pixel locations are represented by an index register as in* (1)*. In case (i), the pixel cyclic shift T is implemented by a qubit permutation and can coincide with S (up to a convention for shift direction). In case (ii), the pixel cyclic shift T acts by modular addition on the binary index and therefore permutes computational basis states of the index register; in general this action is not equal to any fixed cyclic permutation of the physical qubits, and commutation with S does not imply commutation with T.*

Lemma 1 formalizes the point illustrated in Fig. 1(b-d): for address encodings, enforcing QCS is not the same as enforcing the translation symmetry of pixels. A

complete proof is given in Supplementary Section S4. Consequently, to reproduce the classical convolutional inductive bias under amplitude-based encoding, we must construct layers that commute with T (or with T_x, T_y) rather than with S.

2.3 MERA-QCNN as a QCS-equivariant architecture

Many QCNN architectures are inspired by multiscale tensor-network structures such as MERA [5]. Operationally, they apply a repeated local gate pattern along a line (or lattice) of qubits and interleave it with pooling implemented by measurements and classical control. Such templates can be made QCS-equivariant by imposing cyclic weight sharing together with periodic boundary conditions, a placement pattern containing the full cyclic orbit of each local block, and pooling/readout rules compatible with the same qubit permutation. Without these additional choices, a standard brickwork layout generally commutes only with the subgroup preserved by its tiling and pooling pattern, rather than with the full one-site cyclic shift S.

For pixel-to-qubit encodings, QCS equivariance can indeed serve as a translation bias. For address/amplitude encodings, however, Lemma 1 shows that QCS does not generally coincide with pixel translation (PCS). Thus, a QCS-equivariant layout is not guaranteed to be pixel-translation-equivariant for encoded images. This motivates our approach: enforce commutation with the encoding-induced shift T directly.

2.4 Characterization of PCS-equivariant quantum convolution layers

We construct a quantum analogue of a convolutional linear layer for address encoding by requiring commutation with pixel translations. We start in one dimension to keep notation parallel to Fig. 1, and then extend to two and three dimensions.

Consider the Hilbert space

$$\mathcal{H} = \mathcal{H}_{\text{idx}} \otimes \mathcal{H}_{\text{feat}}, \qquad \mathcal{H}_{\text{idx}} \cong \mathbb{C}^N,$$

where $\mathcal{H}_{\text{idx}}$ is spanned by $\{|j\rangle\}_{j\in\mathbb{Z}_N}$ and $\mathcal{H}_{\text{feat}}$ collects the color/feature qubits. A unitary layer U is PCS-equivariant in 1D if

$$U\,(T \otimes I) = (T \otimes I)\,U, \tag{7}$$

where T is defined in (3). In 2D we require commutation with both generators:

$$U\,(T_x \otimes I) = (T_x \otimes I)\,U, \qquad U\,(T_y \otimes I) = (T_y \otimes I)\,U, \tag{8}$$

with T_x, T_y defined in (4). In 3D one adds the analogous condition for T_z.

The commutation relations (7)-(8) are the direct quantum counterpart of the classical condition $T_k A = A T_k$ from Sec. 2.1. They fix the allowed form of U.

Theorem 1 (Fourier-multiplexer form of PCS-equivariant unitaries)**.** *Let T be the cyclic shift on $\mathcal{H}_{\text{idx}}$ defined by (3). A unitary U on $\mathcal{H}_{\text{idx}} \otimes \mathcal{H}_{\text{feat}}$ satisfies (7) if and*

only if it can be written as

$$U = (F_N^\dagger \otimes I)\, \mathcal{B}\, (F_N \otimes I), \tag{9}$$

where F_N is the N-point quantum Fourier transform on the index register and $\mathcal{B}$ is block-diagonal in the Fourier basis,

$$\mathcal{B} = \bigoplus_{k=0}^{N-1} U_k, \tag{10}$$

with arbitrary unitaries U_k acting on $\mathcal{H}_{\text{feat}}$. Equivalently, $\mathcal{B}$ is a multiplexer: it applies a feature transformation U_k conditioned on the Fourier mode k. In two dimensions, U commutes with both T_x and T_y if and only if

$$U = ((F_{N_x} \otimes F_{N_y})^\dagger \otimes I)\, \mathcal{B}\, ((F_{N_x} \otimes F_{N_y}) \otimes I), \tag{11}$$

where $\mathcal{B}$ is block-diagonal over the pair of modes (k_x, k_y):

$$\mathcal{B} = \bigoplus_{k_x=0}^{N_x-1} \bigoplus_{k_y=0}^{N_y-1} U_{k_x,k_y}. \tag{12}$$

The extension to three dimensions is obtained by replacing $F_{N_x} \otimes F_{N_y}$ with $F_{N_x} \otimes F_{N_y} \otimes F_{N_z}$ and indexing blocks by (k_x, k_y, k_z).

The proof idea is as follows. The shift operator T is diagonal in the Fourier basis: $F_N T F_N^\dagger = \text{diag}(\omega^k)$. Hence (7) is equivalent to commutation with a non-degenerate diagonal operator on the index register, which forces U to preserve each Fourier eigenspace. This yields the block-diagonal structure (10) in the Fourier basis, and (9) follows by conjugation. In two (and three) dimensions, the commuting family $\{T_x, T_y(, T_z)\}$ is simultaneously diagonalized by the tensor-product Fourier transform, yielding (11)-(12). The full proof, including the projector argument for the commutant, is given in Supplementary Section S4.

Theorem 1 is the quantum analogue of the classical Fourier characterization of convolution: translation-equivariant linear layers become block-diagonal in the Fourier representation, with each mode independently mixing channels. The Fourier-based construction is therefore not an ad hoc ansatz: once the encoded pixel-shift operator is chosen as the symmetry, it is the corresponding unitary commutant.

Operationally, a PCS convolutional layer maps the spatial index registers to the Fourier basis with the QFT, applies a multiplexer $\mathcal{B}$ that performs a mode-dependent transformation on feature qubits, and then maps the index registers back with the IQFT. An explicit two-layer example is shown in Fig. 3. A schematic decomposition of $\mathcal{B}$ into controlled gates is shown in Supplementary Fig. S7.

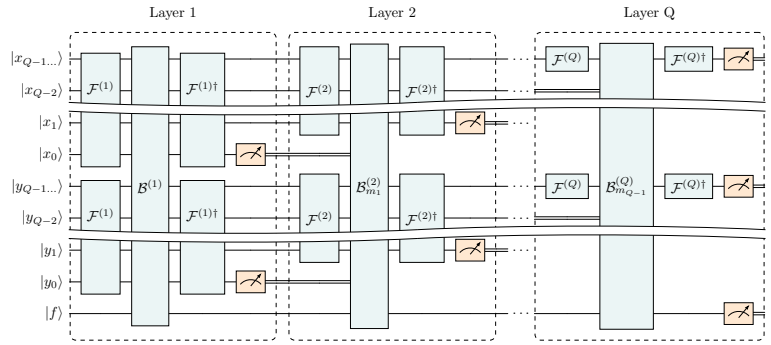


Fig. 3 Schematic of a multilayer PCS-QCNN. Each non-final layer maps the active index registers to the Fourier basis with $\mathcal{F}^{(\ell)}$, applies the mode multiplexer $\mathcal{B}^{(\ell)}$, and maps back with $\mathcal{F}^{(\ell)\dagger}$ before pooling measurements are performed on the least significant qubits of the active computational index registers. The resulting classical outcome m_ℓ is recorded and used as a classical control for the next multiplexer block. After repeated pooling steps (indicated by $\cdots$), the final layer acts on a smaller active index register and is followed by full measurement of the remaining quantum output. Here $x_{Q-1\ldots}$ and $y_{Q-1\ldots}$ denote all remaining index qubits with labels $Q-1, Q, \ldots$.

2.5 From linear PCS layers to a deep QCNN: measurement-induced pooling and deferred conditioning

A composition of unitary PCS-equivariant layers remains a unitary transformation. Therefore, if we only stack layers of the form (11), the overall map remains linear in the state amplitudes. To build a deep architecture analogous to classical CNNs, we introduce a measurement-induced pooling stage between convolutional blocks. In classical networks this role is played by pointwise activations and pooling. Here pooling is implemented by measurement and classical control. At the density-matrix level the process remains linear and CPTP; non-unitarity and branch conditioning enter through measurement, and normalized postselected branch states are nonlinear functions of the pre-measurement state.

We adopt a pooling mechanism inspired by MERA-QCNN but adapted to the Fourier-indexed PCS setting. After a convolutional block we measure the least significant active computational-index qubit per spatial axis (in 2D: one in Reg x and one in Reg y) in the computational (Z) basis, obtaining a classical bit string $m_\ell \in \{0,1\}^d$. In the reduced Fourier junction of Sec. 2.6, this spatial-parity measurement is represented in Fourier space by mixing the alias pair $k = q$ and $k = q + N/2$ for each surviving coarse mode q. Conditioned on m_ℓ, we apply the next trainable block. This produces two effects at once: it introduces a measurement-conditioned branch structure with non-unitary branch updates, and it reduces spatial resolution by removing the measured fine-scale index qubits from subsequent processing. This is the quantum analogue of classical pooling: information carried by fine spatial scales is aggregated into coarser degrees of freedom that remain available to later layers. The scope of exact equivariance is therefore layerwise: each unitary block is exactly PCS-equivariant on the active index register at that depth. Pooling yields a multiscale/coarse equivariance structure rather than exact one-pixel equivariance of the entire readout/classifier pipeline. After pooling, the next block is constrained to commute with the induced

translation on the reduced register. In particular, after one pooling step a unit shift on the coarse lattice corresponds to a shift by 2 sites on the previous lattice (and by 2^r sites after r pooling steps, relative to the original resolution).

In classical CNNs, decreasing spatial resolution is often accompanied by an increase in the number of channels. In the formal model used in this paper (Sec. 3.1), we keep the feature-register size fixed within each architecture instance: n_f (equivalently $D_f = 2^{n_f}$) does not vary with depth. Allowing the feature register to grow across layers is a possible extension, but it is not part of the canonical definition used in the gradient-accounting discussion below.

With these ingredients, one level of PCS-QCNN in 2D can be summarized as

$$\mathcal{L}_\ell = \mathcal{C}_{m_\ell}^{(\ell+1)} \circ \mathcal{M}_{\text{pool}}^{(\ell)} \circ (\mathcal{F}_x^{(\ell)\dagger} \otimes \mathcal{F}_y^{(\ell)\dagger}) \circ \mathcal{B}^{(\ell)} \circ (\mathcal{F}_x^{(\ell)} \otimes \mathcal{F}_y^{(\ell)}). \tag{13}$$

Here, $\mathcal{F}_x^{(\ell)}$ and $\mathcal{F}_y^{(\ell)}$ denote QFTs on the active index registers at depth ℓ (so their register size changes with pooling depth). Equivalently, in the notation of Sec. 3.1, $\mathcal{F}^{(\ell)} = \mathcal{F}_x^{(\ell)} \otimes \mathcal{F}_y^{(\ell)}$. Also, $\mathcal{M}_{\text{pool}}^{(\ell)}$ measures one pooling qubit per spatial axis and outputs $m_\ell \in \{0,1\}^d$, while $\mathcal{C}_{m_\ell}^{(\ell+1)}$ denotes conditional selection of the next trainable block. The full network is obtained by repeating this pattern several times, followed by a readout stage.

2.6 Fourier cancellation at the interface of PCS-QCNN layers

A practical advantage of the Fourier-based PCS construction is that intermediate QFT blocks do not have to be re-applied from scratch at every depth. Consecutive PCS-equivariant layers contain the inverse Fourier transform at the end of one layer next to the forward Fourier transform at the start of the next layer. Pooling measures (and then discards) the parity bit in the computational-index split $r = 2q + s$, while the next layer's QFT acts only on the surviving coarse index q. As a result, the interface formed by the IQFT, the pooling measurement, and the next-layer QFT collapses to a fixed, parameter-free junction from fine Fourier modes to coarse Fourier modes.

Operationally, this junction can be implemented using only local gates in the Fourier alias ordering $k = q + \sigma N/2$: a Hadamard on each alias selector σ, its measurement, and a conditional diagonal phase-gradient on the remaining index register. This yields a cleaner multilayer description in which the only trainable operations are the Fourier-mode multiplexer blocks $\mathcal{B}^{(\ell)}$, while the inter-layer wiring is fixed. This simplification is also useful in the trainability analysis, since it isolates the parameter dependence to the multiplexer blocks. A full derivation and an explicit circuit diagram for the reduced junction are given in Supplementary Section S6.1 (see also Supplementary Fig. S8).

2.7 Readout, hybrid classifier, and the role of shots

After the quantum convolutional part, classical CNNs typically use dense layers for classification. We do not implement a quantum fully connected head [10, 11, 26]; the architecture is restricted to encoding-aligned quantum convolution for amplitude-based encodings. Instead, we use an architecturally simple classical classifier operating

on the probability vector extracted from the quantum state by measurement. In the notation of Sec. 3.1, for each input x the quantum core produces

$$p_\Theta(\cdot \mid x) \in \Delta^{D_{\text{out}}}, \tag{14}$$

which is then mapped to M-class probabilities by

$$q(x) = \text{softmax}\big(W\, p_\Theta(\cdot \mid x) + b\big) \in \Delta^M. \tag{15}$$

The training loss is the cross-entropy in Eq. (27). This final linear head is not symmetry-tied across readout coordinates. Thus, the exact PCS-equivariance statement in this section concerns the unitary quantum convolutional blocks on their active registers and the induced coarse-register action after pooling.

On physical hardware, $p_\Theta(\cdot \mid x)$ is estimated from repeated measurements (shots). Therefore the shot budget N_{shot} becomes an explicit hyperparameter of the model: it controls the variance of the readout and affects both training and inference. In the numerical experiments, idealized performance uses the exact distribution, and the finite-shot regime is analyzed separately in Sec. 6.

2.8 Idealized benchmark model and resource scaling

Resource scaling is central for this construction because the reported benchmark model is not a gate-compiled hardware circuit. For an $N \times N$ grayscale image under address encoding, the index registers require $2\lceil \log_2 N \rceil$ qubits, and the feature register contributes n_f qubits. Circuit depth is dominated by the forward and inverse QFTs on the index registers and by the synthesis of the Fourier-mode multiplexer $\mathcal{B}$. For a standard (textbook) decomposition, an m-qubit QFT uses $O(m^2)$ elementary gates (in particular, $m(m-1)/2$ controlled-phase gates) and has depth $O(m^2)$ without additional parallelization [27, 28]. In our architecture QFT acts separately on the x and y index registers, so this contribution remains modest across the resolutions considered here (up to 32×32, i.e., $n_x = n_y \leq 5$). The main bottleneck is the multiplexer: compiling a fully general mode-dependent block-diagonal operator with n_c control qubits and n_f target (feature) qubits requires a number of elementary gates that grows exponentially in n_c in the worst case [29, 30]. The benchmarks in this paper apply the multiplexer as an ideal block-diagonal operator at the statevector level. Practical NISQ realizations require additional structure (e.g., parameter sharing or a low-rank / low-depth ansatz for the mode blocks), which defines a different model class and changes expressivity/accuracy trade-offs. Additional scaling discussion is provided in Supplementary Section S7.1.

3 Gradient accounting and trainability diagnostics

Deep variational quantum models can suffer from barren plateaus, where gradients become too small for practical training [22]. For the fully general Fourier-multiplexer PCS-QCNN, there is an additional elementary issue: the number of trainable scalar parameters can grow exponentially with the number of active index qubits. This

makes coordinate-wise gradient statements delicate. Even if the total gradient norm is perfectly visible, the gradient energy per parameter can become exponentially small simply because the same amount of signal is distributed over exponentially many independent Fourier-mode blocks.

The resulting average per-coordinate quantity is an accounting baseline, whereas practical trainability is governed by the gradient of the loss actually optimized. Accordingly, Sec. 6.4 reports the corresponding gradient norms directly.

For a dataset or minibatch $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^{N}$, write

$$g_i := \nabla_\Theta \mathcal{L}(\Theta, W, b \mid x_i, c_i). \tag{16}$$

The optimizer for the averaged loss sees the empirical-loss gradient

$$G_\mathcal{D} := \nabla_\Theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\Theta, W, b \mid x_i, c_i) = \frac{1}{N} \sum_{i=1}^{N} g_i, \tag{17}$$

and the training-scale quantity is $\|G_\mathcal{D}\|_2$, with no division by the number of quantum coordinates. We also report the per-sample RMS diagnostic

$$R_\mathcal{D} := \left(\frac{1}{N} \sum_{i=1}^{N} \|g_i\|_2^2 \right)^{1/2}, \tag{18}$$

which measures the typical single-sample gradient before cancellations between samples. These two quantities answer different questions: $R_\mathcal{D}$ can remain large even when the averaged training gradient $\|G_\mathcal{D}\|_2$ is small because the sample gradients point in different directions.

3.1 Depth-scaling accounting

We use the same hybrid model as in Sec. 2. For the accounting statement, the depth-scaling family keeps

$$n_l,\ d,\ D_f,\ M \text{ fixed}, \qquad n_{\text{idx}} = n_l + Q - 1, \tag{19}$$

where Q is the number of PCS-QCNN layers, d is the number of spatial axes, n_l is the number of index qubits per axis left after pooling, $D_f = 2^{n_f}$ is the feature-register dimension, and M is the number of classes. The final readout dimension is therefore fixed:

$$D_{\text{idx}} := 2^{dn_l}, \qquad D_{\text{out}} := D_{\text{idx}} D_f. \tag{20}$$

At layer ℓ, the active index width per axis is

$$n_\ell = n_{\text{idx}} - \ell + 1 = n_l + Q - \ell. \tag{21}$$

With $\mathcal{F}^{(\ell)}$ denoting the d-fold QFT on the active index register, a conditional PCS-equivariant layer has the form

$$U^{(\ell)}(m_{\ell-1}) := \big(\mathcal{F}^{(\ell)\dagger} \otimes \mathbb{I}\big)\, \mathcal{B}^{(\ell)}(m_{\ell-1})\, \big(\mathcal{F}^{(\ell)} \otimes \mathbb{I}\big), \tag{22}$$

where

$$\mathcal{B}^{(\ell)}(m_{\ell-1}) = \sum_{k\in[2^{dn_\ell}]} |k\rangle\langle k| \otimes V_k^{(\ell)}(m_{\ell-1}) \tag{23}$$

is a Fourier-mode multiplexer. As in the benchmark architecture, later multiplexers are conditioned only on the most recently pooled d-bit outcome, so the branch multiplicity is

$$b_1 = 1, \qquad b_\ell = 2^d \quad (\ell \geq 2). \tag{24}$$

The complete measurement-and-feedforward quantum process defines a probability distribution

$$p_\Theta(z \mid x), \qquad z \in [D_{\text{out}}], \tag{25}$$

where Θ denotes all quantum parameters. The linear-softmax head maps $p_\Theta(\cdot \mid x)$ to class probabilities by

$$q(x) = \text{softmax}\big(W p_\Theta(\cdot \mid x) + b\big) \in \Delta^M. \tag{26}$$

For label $c \in \{1, \ldots, M\}$ the loss is

$$\mathcal{L}(\Theta, W, b \mid x, c) := -\log q_c. \tag{27}$$

For the full Pauli-exponential block ansatz in Eq. (37), each mode-wise feature block has $p_{\text{blk}} = 4^{n_f}$ real scalar parameters. The quantum-core parameter count is therefore

$$P_Q = p_{\text{blk}} \sum_{\ell=1}^{Q} b_\ell\, 2^{d(n_l+Q-\ell)}. \tag{28}$$

In particular,

$$P_Q \geq p_{\text{blk}}\, 2^{d(n_l+Q-1)}, \tag{29}$$

so P_Q grows exponentially in Q at fixed n_l, d, n_f. Let $G_Q = \nabla_{\Theta_Q} \mathcal{L}(\Theta_Q, W, b \mid x, c) \in \mathbb{R}^{P_Q}$ be the gradient with respect to the quantum-core parameters for any fixed sample, minibatch-average loss, or empirical-risk average. In the initialization protocol used for the benchmark, all quantum Pauli coefficients in Eq. (37) are sampled independently from $\text{Unif}(0, 2\pi)$. The classical head is initialized independently by the fan-in uniform rule

$$\begin{aligned} W_{az} &\sim \text{Unif}\Big(-D_{\text{out}}^{-1/2}, D_{\text{out}}^{-1/2}\Big), \quad a \in [M], \\ b_a &\sim \text{Unif}\Big(-D_{\text{out}}^{-1/2}, D_{\text{out}}^{-1/2}\Big), \quad z \in [D_{\text{out}}], \end{aligned} \tag{30}$$

independently over all entries, where $D_{\text{out}} = 2^{dn_l} D_f$ is fixed in the depth-scaling regime. The elementary sensitivity estimate in Supplementary Section S8 gives

$$\mathbb{E}\|G_Q\|_2^2 \leq \frac{8M}{3} p_{\text{blk}} Q, \tag{31}$$

where the expectation is over this independent initialization. Dividing by (29) yields

$$\mathbb{E}\left[\frac{1}{P_Q}\|G_Q\|_2^2\right] \leq \frac{8MQ}{3\,2^{d(n_l+Q-1)}}. \tag{32}$$

For the ten-class MNIST heads used in the experiments, the numerator constant is $80Q/3$. Thus the average per-coordinate second moment is upper-bounded by a quantity scaling as $Q2^{-dQ}$ in this formal depth-scaling family. This establishes the parameter-count accounting baseline. It follows from the exponential growth of P_Q, while aggregate gradient norms are measured separately in Sec. 6.4 through $\|G_{\mathcal{D}}\|_2$ and $R_{\mathcal{D}}$.

4 Benchmark choice: MNIST for convolutional inductive bias

MNIST provides a widely used and computationally tractable benchmark [31]. We use it to test translation-equivariant inductive bias under limited qubit and depth budgets. Because quantum-machine-learning benchmarks require careful matched controls [32, 33], the MNIST protocol below is used to isolate the convolution-sensitive part of the comparison.

The practical benefit of convolution is regime-dependent. On non-translated full-data MNIST, dense models can narrow the gap to CNNs, which makes the convolutional effect harder to diagnose. Supplementary Fig. S3 shows this directly for the classical controls used here: on the full standard MNIST split resized to 32×32 without translations, the CNN and MLP remain much closer than on translated-MNIST (99.15% vs 96.74% final mean test accuracy). To make the translation-sensitive inductive bias explicit, we use a translated-MNIST regime: each digit is resized to 16×16, embedded in a 32×32 canvas, and translated by a random integer offset of at most 8 pixels per axis. We train on a balanced subset of 1000 examples per class and check explicitly that a convolutional classical CNN reaches higher accuracy than a classical MLP on this translated task. The same translated benchmark is then used for the PCS-QCNN versus matched non-PCS RBC-QCNN comparison in Fig. 2(b).

Quantum image models are also constrained by input dimension. Even MNIST's native 28×28 resolution is too costly for straightforward state preparation and fully general multiplexer implementations on near-term hardware, so classical preprocessing remains practically important. In the reported experiments we work with power-of-two canvases and a small set of controlled preprocessing choices: the translated benchmark uses digits resized to 16×16 and placed on a 32×32 canvas, while the full-MNIST sweep in Secs. 5 and 6 uses four settings: direct preprocessing to 8×8, direct preprocessing to 16×16, centered embedding of native 28×28 images in a 32×32 canvas, and direct preprocessing to 32×32. Alternative compression methods such as PCA can reduce dimension more aggressively but may suppress spatial correlations and are therefore less suitable for isolating convolutional inductive bias.

Finally, a large body of QCNN-related work has used MNIST under heterogeneous settings (binary vs multi-class tasks, varying downscaling, and different train/test protocols). We use a fully specified protocol and include a comparable-budget classical reference effect (CNN vs MLP) to isolate architectural conclusions about quantum convolution. For a literature map of QCNN-like approaches on MNIST and representative reported settings/accuracies, see Supplementary Section S1.

5 Benchmark

The benchmark protocol fixes the data preprocessing, hybrid model (encoding, quantum core, classical head), training and evaluation procedures, and simulation modes (statevector vs finite-shot sampling). The MNIST-specific motivation is given separately in Sec. 4; the present section gives the full experimental specification.

5.1 Data, splits, and preprocessing

We use two MNIST regimes. For Figs. 2 and 5(a), we use a translated-MNIST benchmark built from a balanced subset of 1000 training examples per class; the standard MNIST test split membership is kept unchanged. Within this translated regime, both train and test images are resized from 28×28 to 16×16 by bilinear interpolation, placed on a zero-filled 32×32 canvas, and translated by an independently sampled integer offset with maximum magnitude 8 pixels along each axis. The prepared translated train and test sets are fixed within each independent run and reused for all training epochs and test evaluations. The PCS symmetry is formulated on a periodic lattice to obtain an exact finite group action, while this benchmark uses only the corresponding non-wrapping translation regime: the resized digit patch remains inside the 32×32 canvas, so no occupied pixel is transported across the boundary. Thus, the translated-MNIST split probes realistic bounded translation stability on the same discrete lattice without requiring the dataset to contain complete cyclic orbits. These translated-MNIST runs use train/test batch sizes 256/1600.

For Fig. 5(b) and all finite-shot and diagnostic follow-up plots, we use the full-MNIST split (60,000 training images and 10,000 test images) with no random translations. The evaluated full-MNIST preprocessing settings are direct preprocessing to 8×8, direct preprocessing to 16×16, centered embedding of the native 28×28 images in a 32×32 canvas, and direct preprocessing to 32×32. These full-MNIST runs use train/test batch sizes 512/16000. In all cases the preprocessing protocol outputs grayscale tensors normalized to $[0, 1]$.

5.2 Encoding and state preparation

Each input image is encoded by a FRQI-like brightness map [18]. Let $x_{u,v} \in [0, 1]$ denote the preprocessed grayscale value at pixel (u, v). The encoder maps it affinely to an angle

$$p_{u,v} = a + (b - a)x_{u,v}, \tag{33}$$

and writes the local feature state from Eq. (2) into the least-significant feature qubit. For $n_f > 1$, all remaining auxiliary feature qubits are initialized in $|0\rangle^{\otimes(n_f-1)}$; in

our feature-register ordering the local state is $|0\rangle^{\otimes(n_f-1)} \otimes |\phi_{u,v}\rangle$, with $|\phi_{u,v}\rangle$ on the least-significant feature qubit. For the reported PCS-QCNN experiments we use the brightness interval $(a, b) = (0, \pi)$. We choose this a priori as a fixed one-period convention for the local real FRQI color map, not by optimizing it on the reported test set. Locally, $|\phi(p+\pi)\rangle = -|\phi(p)\rangle$, so intervals longer than π revisit the same one-qubit rays. In the coherent FRQI image superposition, however, such signs can become address-dependent relative phases rather than a global phase of the full image state. A separate low-data brightness-range sweep varies the upper endpoint b while fixing $a = 0$ and is reported only as a post-hoc sensitivity check of this convention (Supplementary Section S2.2 and Fig. S1). Thus an image on an $N_x \times N_y$ canvas is represented using $\log_2 N_x + \log_2 N_y$ index qubits together with n_f feature-register qubits (the grayscale/color qubit plus optional auxiliary feature qubits).

The input state is initialized directly as a dense tensor rather than by compiling an explicit state-preparation circuit. Relative to the conceptual normalized form in Eq. (1), the numerical realization used here omits the global $1/\sqrt{N_x N_y}$ factor during initialization. Encoded sample norms are therefore $\sqrt{N_x N_y}$ rather than 1. The measurement layer compensates for this by dividing the final marginals by the known overall spatial normalization factor, so the classifier still receives normalized readout probabilities. Approximate and structured FRQI loading circuits remain relevant for future hardware-oriented implementations [26, 34, 35].

5.3 Models under comparison and hyperparameters

We evaluate the proposed translation-symmetry-aligned quantum convolutional architecture using the complete hybrid model shown in Fig. 4. The model consists of a FRQI-like encoder, a PCS-QCNN quantum core, a marginal measurement stage, an optional finite-shot sampling layer, and a linear-softmax head. The head applies a single linear map of size $(D_{\text{out}}, 10)$, where D_{out} is the dimension of the measured probability vector passed to the classifier. The resulting logits are converted to probabilities by a softmax and trained with the cross-entropy loss.

We begin with the quantum part (PCS-QCNN). A model of depth Q is obtained by composing Q PCS-equivariant layers in the sense of Sec. 2. At layer ℓ, the active index registers are mapped to the Fourier basis, processed by a trainable Fourier-mode multiplexer, and returned to the computational basis; after each non-final layer, one active index qubit per spatial axis is pooled via deferred measurement and moved into an explicit condition register, while the final layer is followed directly by readout. Later multiplexers are conditioned on the most recently pooled x- and y-bits. In the reported results we use the reduced Fourier junction, so each explicit interface consisting of an IQFT, pooling, and the next-layer QFT is replaced by the analytically equivalent fixed junction described in Sec. 2.6. At the density-matrix level the resulting process is linear and CPTP. In an individual-branch statevector description, the normalized postselected branch states are nonlinear functions of the pre-measurement state.

The reported benchmarks use two PCS-QCNN parameter families. Figure 2(b) fixes a translated-MNIST model with $Q = 3$ and $n_f = 2$ on a 32×32 canvas. Figure 5(a) sweeps $Q \in \{1, 2, 3, 4, 5\}$ and $n_f \in \{1, 2, 3\}$ on the same translated benchmark. Figure 5(b) and all finite-shot follow-up plots use a full-MNIST size sweep with

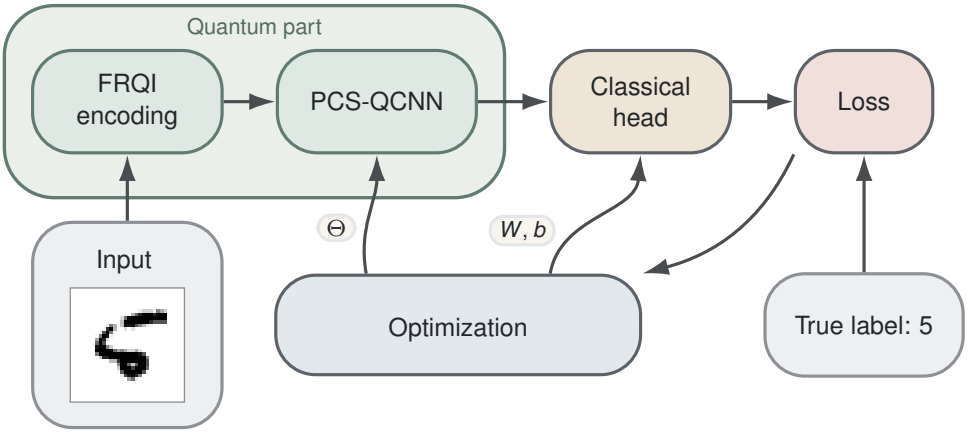


Fig. 4 Overview of the hybrid model used in our experiments. The Encoding block prepares an FRQI-like image state on the spatial index registers and the feature register. The PCS-QCNN block applies one or more PCS-equivariant quantum layers with measurement-induced pooling implemented through deferred conditioning. The final measured probability tensor (or its finite-shot histogram estimate at evaluation time) is fed into a linear-softmax head that produces 10-class logits.

$Q = 1$ and $n_f = 3$ across four preprocessing settings: direct preprocessing to 8×8, direct preprocessing to 16×16, centered embedding of native 28×28 images in a 32×32 canvas, and direct preprocessing to 32×32; the follow-up reference model for Figs. 7, 8, and Supplementary Figs. S4, S5, and S6 is the medium-resolution 16×16 setting.

Let $n_{\mathrm{idx}} = \log_2 N$ denote the number of index qubits per spatial axis for an $N \times N$ preprocessed image. The total number of logical qubits in the quantum part is

$$n_{\mathrm{tot}} = 2n_{\mathrm{idx}} + n_f. \tag{34}$$

For the full-readout setting used in all reported figures, the final marginal measurement sums over the explicit condition registers and leaves only the surviving active spatial axes together with the feature register. Hence

$$n_l = n_{\mathrm{idx}} - Q + 1, \qquad n_{\mathrm{meas}} = 2n_l + n_f, \tag{35}$$

and the classifier input dimension is

$$D_{\mathrm{out}} = 2^{n_{\mathrm{meas}}} = 2^{2n_l + n_f}. \tag{36}$$

Therefore the classical head size is determined by the number of active qubits left after pooling, not by the total logical register size.

The classifier head is architecturally simple but not always smaller than the quantum core in parameter count. It is a single biased linear-softmax map $\mathbb{R}^{D_{\mathrm{out}}} \to \mathbb{R}^{10}$, so the classifier contributes exactly $10D_{\mathrm{out}} + 10$ trainable parameters. We denote by P_Q the number of trainable real parameters in the quantum core. These counts are summarized for the reported sweeps in Supplementary Tables S2 and S3.

The trainable Fourier-mode blocks also determine the hardware-facing cost model. In the benchmark model, each mode-dependent feature-register block $U^{(\ell)}_{k_x,k_y}(m)$ is parameterized as a general $U(2^{n_f})$ unitary via the exponential map

$$U^{(\ell)}_{k_x,k_y}(m) \;=\; \exp\Big(i \sum_{\alpha\in\mathcal{P}_{n_f}} \theta^{(\ell)}_{k_x,k_y,\alpha}(m)\, P_\alpha\Big), \tag{37}$$

where $\mathcal{P}_{n_f}$ denotes the full n_f-qubit Pauli-string basis, including the all-identity string. The identity coefficient is retained: it is only a phase for a single feature block, but mode- and branch-dependent block phases become relative phases inside the full Fourier multiplexer and therefore are part of the general PCS-equivariant layer. This gives 4^{n_f} real parameters per Fourier mode (and, for $\ell \geq 2$, per pooling branch m). At initialization, every scalar coefficient $\theta^{(\ell)}_{k_x,k_y,\alpha}(m)$ in Eq. (37) is sampled independently from $\mathrm{Unif}(0, 2\pi)$. This specifies the coordinate initialization of the exponential ansatz; it is not a Haar draw from $U(2^{n_f})$. Since we only use $n_f \leq 3$ in the reported benchmarks, each individual mode block acts on at most three qubits and can be synthesized with $O(4^{n_f})$ elementary gates using standard compilation methods; the dominant challenge is multiplexing these blocks across many Fourier modes. For an $N_x \times N_y$ input, the number of Fourier modes is N_xN_y. At fixed n_f, a fully general PCS layer therefore scales linearly in the number of spatial modes, $p_{\mathrm{PCS}} = O(N_xN_y)$, whereas an unconstrained dense mode-mixing map scales as $p_{\mathrm{dense}} = O((N_xN_y)^2)$. The benchmark PCS layer uses the general translation-equivariant family: we do not impose finite spatial support or mode-wise weight sharing inside the Fourier blocks. The corresponding multiplexer $\mathcal{B}^{(\ell)}$ is applied as an explicit block-diagonal unitary in the Fourier basis, i.e., as $\bigoplus_{k_x,k_y} U^{(\ell)}_{k_x,k_y}(m)$ controlled by the active Fourier indices and the selected condition branch. For simulation, we apply this operator directly as an explicit block-diagonal map. Rather than explicitly branching on mid-circuit measurement outcomes, pooling and feedforward are implemented through the equivalent deferred-measurement form (keeping pooled qubits as condition-register controls and marginalizing them in the final readout), which is operationally identical to measuring those qubits in the computational basis and classically conditioning the next layer [27].

On real NISQ hardware, the dominant cost would be compiling these large multiplexers into one- and two-qubit gates. In the worst case (no structure shared across modes), standard decompositions of uniformly controlled unitaries require a number of elementary gates that grows exponentially in the number of control qubits (hence polynomially in the number of pixels) and can quickly dominate depth as the image resolution increases [29, 30]. The controls are matched within each comparison family: the classical CNN and MLP have nearly equal parameter counts, while the PCS-QCNN and RBC-QCNN have identical quantum-core and classifier-head parameter counts. Hardware-oriented structured/compressed multiplexer designs are discussed in Supplementary Section S7.1.

5.4 Training objective and optimization

The training objective is the standard multiclass cross-entropy loss. For each sample (x, c), the quantum core produces a readout vector $p_\Theta(\cdot \mid x)$, the classifier head outputs logits $z(x) = W\, p_\Theta(\cdot \mid x) + b$, and class probabilities are obtained as $q(x) = \mathrm{softmax}(z(x))$. The minibatch loss is therefore

$$\ell = -\frac{1}{B} \sum_{b=1}^{B} \log q_{c_b}(x_b), \tag{38}$$

which is the minibatch version of Eq. (27).

All reported models are optimized end-to-end with Adam. The classical head uses fan-in uniform initialization: with fan-in D_{out}, both weights and biases are sampled independently from $\mathrm{Unif}(-D_{\mathrm{out}}^{-1/2}, D_{\mathrm{out}}^{-1/2})$. The classical baselines use learning rate 10^{-2}, while the hybrid PCS-QCNN models use learning rate 3×10^{-2}. The translated-MNIST fixed runs (Fig. 2), the translated-MNIST architecture sweep (Fig. 5(a)), and the full-MNIST size sweep (Fig. 5(b)) are all trained for 2000 epochs with test evaluation every 10 epochs. The finite-shot follow-up analyses use two subsets of these epochs: Fig. 7 uses epochs $10, 100, 200, \ldots, 800$, while Fig. 8 uses epochs 100 and 800.

5.5 Simulation modes: statevector vs finite-shot readout

All experiments are executed on a classical simulator. We use two simulation modes for the quantum part:

In the infinite-shot (statevector / exact-probability) mode, the simulator computes the exact quantum state (statevector) and the exact output probabilities for the measured qubits. This removes sampling noise and corresponds to the formal $N_{\mathrm{shot}} \to \infty$ limit. To avoid ambiguity, all training is performed exclusively in this infinite-shot mode. This choice isolates architectural effects (symmetry, depth, feature register size, conditioning structure) from the stochasticity of finite-sampling training, and it reflects the fact that stable finite-shot training for larger hybrid models remains computationally expensive.

In the finite-shot mode, the measured probability vector is estimated from a finite number of measurement shots N_{shot} by sampling from the output distribution. Equivalently, for each test image with exact readout vector $p_\Theta(\cdot \mid x)$, the classifier receives the empirical frequency vector $\hat{p}$ obtained from N_{shot} multinomial draws from $p_\Theta(\cdot \mid x)$. This emulates the fundamental finite-sampling noise of quantum readout even on ideal (noise-free) hardware. We apply the finite-shot mode only at inference time. Concretely, for a fixed trained model, we evaluate the classifier on the test set by replacing the exact probability vector with its finite-shot estimate and report accuracy as a function of N_{shot} (e.g., $128, 256, 512, 1024, 2048$ shots).

Thus, throughout the paper, training always uses infinite-shot (exact) quantum readout, whereas inference is reported in both infinite-shot (exact) and finite-shot modes.

5.6 Baselines and matched controls

We include two classical references and one matched quantum control around the translated-MNIST benchmark from Sec. 5.1. First, Fig. 2(a) compares a convolutional classical CNN against a pure-dense MLP control on translated-MNIST with 16×16 digits placed on a 32×32 canvas. The CNN is the four-convolution architecture shown in Supplementary Fig. S2(a), and the MLP is the three-hidden-layer dense control shown in Supplementary Fig. S2(b). Their trainable-parameter counts are of the same order (47,034 for the CNN and 47,947 for the MLP), so the main architectural difference remains the presence or absence of convolutional weight sharing. Supplementary Fig. S3 then applies the same pair to the full standard MNIST split resized directly to 32×32 without translations, which acts as a control showing that non-translated MNIST alone is a less stringent test of convolutional inductive bias.

Second, Fig. 2(b) compares the PCS-QCNN against a matched non-PCS RBC-QCNN on the same translated benchmark. In this control, each layer replaces the Fourier-basis change and its inverse by a fixed, non-trainable, layer-specific random spatial unitary R_ℓ and its adjoint $R_\ell^\dagger$, applied identically on the active x- and y-registers. Each R_ℓ is generated by exponentiating an i.i.d. Gaussian Pauli-basis Hermitian generator of the appropriate active-index dimension. The control keeps the same depth Q, feature-qubit count n_f, multiplexer parameter count, pooling schedule, full readout, and classical head as the PCS-QCNN; unlike the reported PCS-QCNN model, it uses explicit pooling rather than the reduced Fourier-junction shortcut. This preserves the matched architecture and parameter count but generally removes the PCS-equivariance guarantee. For the fixed translated run used in Fig. 2(b), both quantum variants use $Q = 3$, $n_f = 2$, full readout, and 39,434 trainable parameters in total (36,864 quantum and 2,570 in the classifier). The results of these baseline comparisons are reported in Sec. 6 and summarized numerically in Table 1.

5.7 Numerical realization

All reported results are obtained by dense-tensor statevector simulation. The encoder, Fourier transforms, reduced Fourier junctions, multiplexers, marginal measurement, finite-shot sampling layer, and classifier are evaluated directly on the full state tensor rather than through a gate-by-gate hardware emulation. Training uses exact-probability statevector evolution; the finite-shot studies replace the exact readout by multinomial histograms sampled from the same trained quantum output distribution. The reproducibility package is available in [36].

6 Results

The empirical results for the hybrid PCS-QCNN follow the benchmark protocol fixed in Sec. 5. They address the effect of the PCS construction relative to a matched non-PCS RBC-QCNN, the classical CNN/MLP control across translated and non-translated MNIST regimes, infinite-shot behavior across architecture and input-size sweeps, and finite-shot readout as a resource constraint. As fixed above, the quantum control is a matched non-PCS ablation rather than a QCS baseline, and the finite-shot

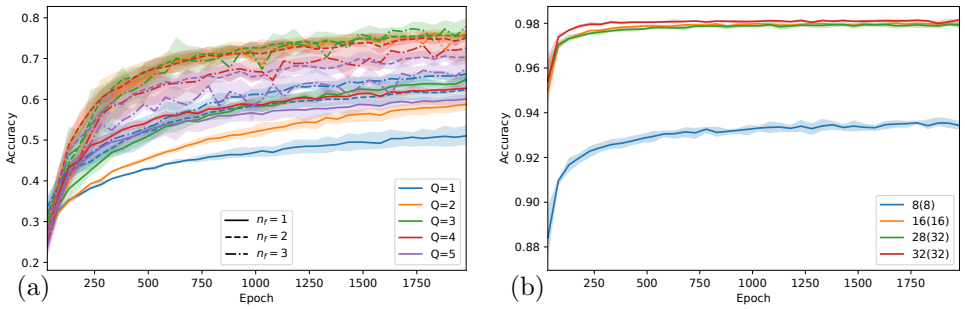


Fig. 5 Infinite-shot test-accuracy dynamics for the PCS-QCNN sweeps. (a) Translated-MNIST architecture sweep over quantum layers $Q \in \{1, \ldots, 5\}$ and feature-qubit counts $n_f \in \{1, 2, 3\}$ with full readout on the translated benchmark with 16×16 digits placed on a 32×32 canvas; colors encode Q and line styles encode n_f. (b) Full-MNIST size sweep for the canonical $Q = 1$, $n_f = 3$ PCS-QCNN across four preprocessing settings: direct 8×8, direct 16×16, centered 28×28 in a 32×32 canvas, and direct 32×32. In both panels, solid curves show mean test accuracy and shaded bands show the 25th–75th percentile range over 3 seeds.

diagnostics use a single direct-16×16 full-MNIST reference run. Unless stated otherwise, all models are trained in the infinite-shot (exact-probability) simulator mode; at inference time we report both infinite-shot performance and finite-shot reevaluations obtained by sampling the quantum readout distribution with a fixed shot budget.

6.1 Infinite-shot inference performance and learning dynamics

Figure 5(a) reports the translated-MNIST architecture sweep under infinite-shot inference. In this sweep, several multi-layer models exceed the one-layer configurations with the same feature width, but performance is not monotone in depth. The largest displayed mean occurs at the intermediate setting $Q = 3$, $n_f = 3$ (79.15%), but because the sweep reports only three-seed means and no uncertainty estimates, we treat this as a descriptive sweep outcome rather than a resolved architecture ranking. Increasing from $n_f = 1$ to wider feature registers often improves performance, although the trend is not monotone at the deepest settings. Thus, on the translated benchmark, both depth and feature width materially affect performance.

Figure 5(b) isolates input resolution on full-MNIST using the canonical $Q = 1$, $n_f = 3$ PCS-QCNN. Moving beyond 8×8 gives a clear gain: the 8×8 model has the lowest final mean (93.21%), while the 16×16, 28×28-on-32×32, and direct-32×32 models all reach high final accuracies. Among the displayed preprocessing choices, the direct-32×32 run attains the highest final mean (98.13%). For a map of QCNN-like results under heterogeneous MNIST settings, see Supplementary Section S1. A detailed error breakdown for the final direct-16×16 full-MNIST reference model is provided in Supplementary Section S2.7, including a confusion matrix and representative misclassified examples. The remaining errors are consistent with visual ambiguity between similar digits after aggressive downscaling to a 16×16 quantum input.

Model / baseline	Params	Accuracy (%)
Classical CNN	47,034	97.68
Classical MLP	47,947	48.93
PCS-QCNN	39,434	75.89
RBC-QCNN	39,434	40.82

Table 1 Final mean test accuracies for the translated benchmark entries in Fig. 2 (1000 training examples per class, with digits resized to 16×16, placed on a 32×32 canvas, and translated by at most 8 pixels along each axis for the classical and quantum runs). The parameter column reports the total number of trainable parameters. Both quantum variants use $Q = 3$, $n_f = 2$, and full readout. The reported value in each row is the mean over 3 seeds.

6.2 Hyperparameter sweep and parameter accounting

The translated-MNIST architecture sweep spans $Q \in \{1, 2, 3, 4, 5\}$ and $n_f \in \{1, 2, 3\}$. In this sweep, depth and feature width are both relevant: wider feature registers often help, and the top observed means occur at intermediate depths, but the dependence is not monotone and the sweep should not be read as a precise architecture ranking. Exact parameter counts for the reported architecture and size sweeps are summarized in Supplementary Tables S2 and S3; the corresponding accuracy trends are shown directly in Fig. 5.

6.3 Matched controls on translated-MNIST

In the translated-MNIST regime from Sec. 5.1, we include two comparisons: (i) classical CNN versus classical MLP on translated-MNIST with 16×16 digits on a 32×32 canvas, and (ii) the PCS-QCNN versus the matched non-PCS RBC-QCNN under the translated hybrid protocol (Sec. 5.6). The results are summarized in Table 1.

Figure 2(a) shows a wide separation between the classical CNN and MLP on the translated benchmark: with 16×16 digits placed on a 32×32 canvas, the CNN reaches a final mean test accuracy of 97.68%, whereas the dense MLP reaches 48.93%. Supplementary Fig. S3 shows that the same classical architectures are much closer on full 32×32 MNIST without translations (99.15% vs 96.74%), illustrating that non-translated MNIST alone is a less stringent benchmark for testing convolutional inductive bias. Within the quantum family, Fig. 2(b) shows a 35.08 percentage-point gap between the PCS-QCNN and the matched non-PCS RBC-QCNN: 75.89% versus 40.82% final mean test accuracy. Thus, although the fixed $Q = 3$, $n_f = 2$ PCS-QCNN does not match the classical CNN on this translated benchmark, the comparison provides evidence for a PCS-specific contribution relative to a matched symmetry-broken quantum control.

6.4 Optimization measurements

The parameter-count estimate in Sec. 3 controls the mean per-coordinate second moment. The practical trainability question is whether the empirical-loss gradient seen by the optimizer is numerically visible. We measure two initialization-time gradient diagnostics in a depth-scaling family on $2^Q \times 2^Q$ inputs, with fixed post-pooling size $n_l = 1$ and fixed feature register $n_f = 3$. Figure 6 reports the empirical-loss

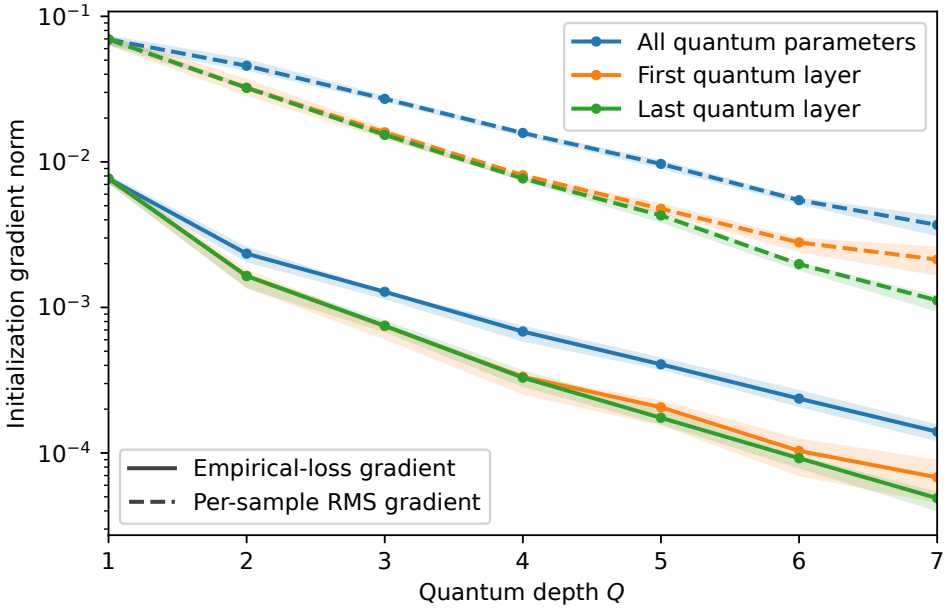


Fig. 6 Initialization-time quantum-gradient diagnostics in the depth-scaling family. Solid curves show the empirical-loss gradient norm $\|G_{\mathcal{D}}\|_2$ and dashed curves show the per-sample RMS gradient norm $R_{\mathcal{D}}$, defined in Eqs. (17) and (18). Colors indicate whether the selected coordinates are all quantum parameters, the first quantum layer, or the last quantum layer.

gradient norm $\|G_{\mathcal{D}}\|_2$ and the per-sample RMS gradient norm $R_{\mathcal{D}}$, each for all quantum parameters, for the first quantum layer, and for the last quantum layer. The solid curves isolate the actual averaged training signal, while the dashed curves show the typical single-sample signal before inter-sample cancellation. Thus the diagnostic separates two possible sources of small gradients: attenuation of the single-sample gradients themselves and cancellation when the empirical loss is averaged over the dataset. For each depth $Q \in \{1, \ldots, 7\}$, the diagnostics are computed at initialization on the same deterministic class-balanced subset $\mathcal{D}$ of $N = 256$ test samples. For this diagnostic only, the selected raw test images are resized to $2^{Q-1} \times 2^{Q-1}$, placed on a $2^Q \times 2^Q$ canvas, and translated by fixed integer offsets within the available border; the resulting depth-specific inputs are held fixed across the 12 random initializations. For a fixed random initialization, $\|G_{\mathcal{D}}\|_2$ is obtained by differentiating the cross-entropy loss averaged over this subset and then taking the Euclidean norm of the selected quantum-parameter coordinates. By contrast, $R_{\mathcal{D}}$ is obtained by differentiating each single-sample loss separately, taking the Euclidean norm on the same selected coordinate set, squaring, averaging these squared norms over the 256 samples, and finally taking the square root. The three selected coordinate sets are the full quantum core, the first quantum layer, and the last quantum layer. Neither diagnostic is divided by the number of coordinates. The plotted line is the mean over 12 independent random initializations, and the shaded band is the corresponding 25th–75th percentile range.

We also monitor the Shannon entropy of the full readout distribution under finite-shot reevaluation in Supplementary Fig. S4. The entropy tracks how broad the sampled output histograms remain as training proceeds, separating finite-shot readout effects from the gradient-scale diagnostics above.

6.5 Finite-shot inference and a representative train–deploy mismatch

The practical cost of inference on quantum hardware scales with the number of measurement shots used to estimate the readout distribution. Therefore, shot budget becomes an explicit hyperparameter (Sec. 5.5). We evaluate finite-shot inference by sampling from the exact output distribution produced by the statevector simulator and then running the classical head on the estimated probability vector.

Figure 7 reevaluates the representative direct-16×16 full-MNIST reference model ($Q = 1$, $n_f = 3$) at selected training epochs and several shot budgets. Each finite-shot point is one stochastic full-test-set pass in which each test image receives one independent N_{shot}-sample multinomial histogram, while the infinite-shot curve is deterministic. In this run, reducing the shot budget degrades accuracy. At fixed shot budget, finite-shot accuracy does not track infinite-shot accuracy monotonically: the exact-probability curve improves and then stays high, whereas the finite-shot curves peak earlier and can decline under continued infinite-shot training. These diagnostics identify a possible train–deploy mismatch even in this noiseless setting: exact-probability optimization can find a solution that is more fragile under a fixed inference-time shot budget. We probe this behavior below through shot-noise propagation in the hybrid readout.

6.6 Mechanism analysis: loss distributions under finite-shot sampling

To probe this finite-shot sensitivity, we examined the distribution of loss values obtained when replacing the exact readout vector by its finite-shot estimate. Figure 8 shows the resulting batch-mean test cross-entropy distributions for the representative direct-16×16 $Q = 1$, $n_f = 3$ model at two training stages (epochs 100 and 800) and for several shot budgets. These histograms use the same trained model. For each epoch and shot budget, the 10,000-sample test set is split into 40 nonoverlapping batches of 250 images; each finite-shot histogram entry is one batch-mean cross-entropy under one of 100 independent resampling passes, giving $40 \times 100 = 4000$ entries per shot budget. The infinite-shot reference contains one exact pass over the same 40 batches.

Two qualitative effects are visible. First, there exists a practical threshold in the shot budget at which the loss distribution changes character: at sufficiently large N_{shot} the distribution remains concentrated near the infinite-shot value, while at smaller budgets it develops pronounced tails and, in some cases, multi-modality. Second, in this run the shape of the distribution changes with training duration: the longer-trained model can exhibit rare but extremely large loss outliers under limited shots (rightmost bins in Fig. 8), even when the infinite-shot accuracy is higher. These outliers are consistent with the longer-trained model being sharper with respect to perturbations of

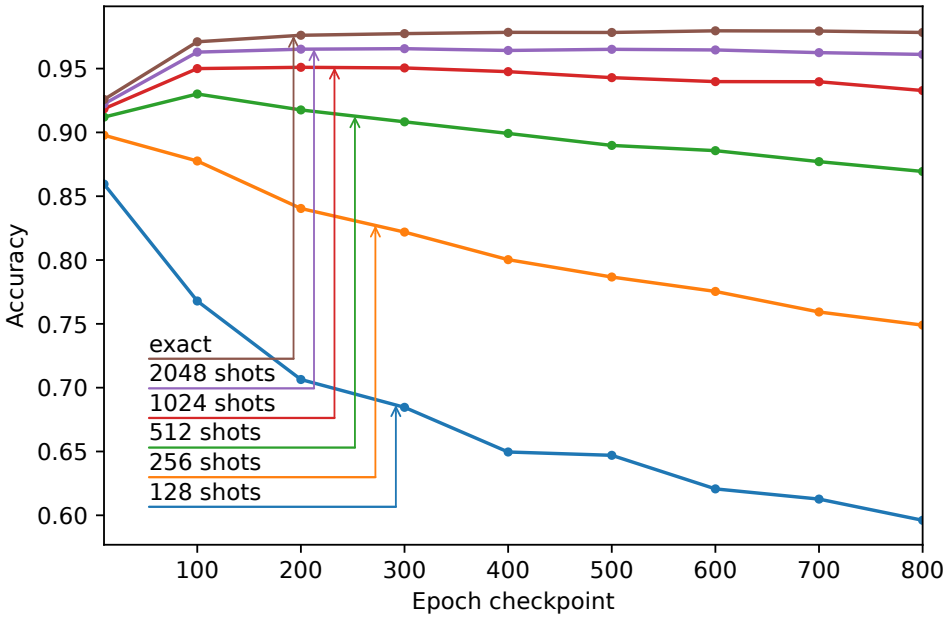


Fig. 7 Finite-shot reevaluation of one representative direct-16×16 full-MNIST reference PCS-QCNN ($Q = 1$, $n_f = 3$). Each curve corresponds to an inference-time shot budget (128, 256, 512, 1024, 2048, or infinite-shot exact readout). Each finite-shot point is one stochastic full-test-set pass with one independently sampled multinomial readout histogram per test image; the exact curve is deterministic. Training always uses exact probabilities; only inference is resampled.

the readout distribution, so that a small probability-estimation error can occasionally push the classical head into a high-loss regime.

6.7 Geometric view of finite-shot sensitivity

A local loss-landscape probe in readout-probability space tests the same interpretation for the same trained model. The resulting landscape is consistent with the same pattern: the longer-trained model has a sharper effective loss surface along finite-shot perturbation directions. The construction and the corresponding figure are reported in Supplementary Section S2.6.

7 Conclusions

Quantum convolution is fundamentally an encoding-aware symmetry constraint. The relevant symmetry is not fixed by the QCNN layout alone, but by how translations of the original data act after encoding. The circuit should be equivariant under the translation action induced on the quantum state by the chosen data encoding. For FRQI-like address/amplitude encodings, pixel translations act as modular addition on the index registers (PCS), which generally differs from cyclic permutations of physical

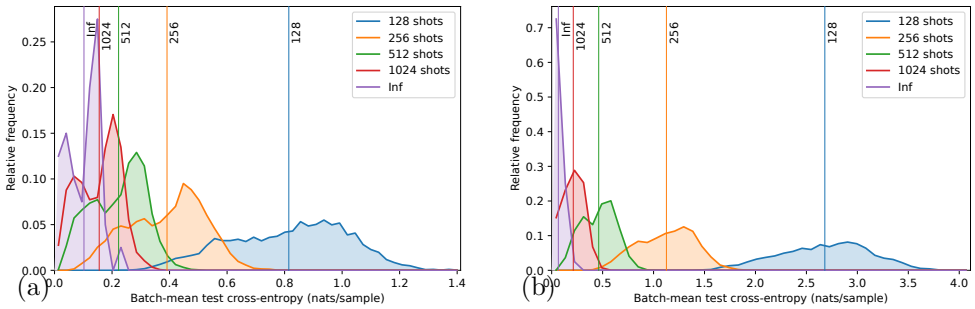


Fig. 8 Batch-mean test cross-entropy distributions for one representative direct-16 × 16 $Q = 1$, $n_f = 3$ model under finite-shot readout. (a) After 100 training epochs. (b) After 800 training epochs. Colored histograms correspond to 128, 256, 512, 1024, and infinite-shot readout. Each finite-shot entry is the mean cross-entropy on one 250-sample test batch under one of 100 independent multinomial resampling passes, giving 4000 entries per finite-shot histogram; the exact reference has 40 entries. Vertical lines mark sample-weighted mean losses.

qubits (QCS) enforced by many MERA-inspired QCNN layouts. Although our implementation is based on the FRQI encoding, the underlying idea is more general. Any data encoding that can be expressed through linear transformations admits, in principle, a corresponding linear operator implementing the same translationally equivariant convolutional mapping. However, while such symmetry-preserving constructions can be defined in principle, their practical realization may be nontrivial, and some encodings may lead to operators that are difficult to implement efficiently. Determining how to implement these operators with low quantum resource requirements for different encoding schemes remains an open problem and an interesting direction for future work.

This perspective yields a constructive characterization of PCS-equivariant quantum layers. At the layer level, the resulting unitary family is the commutant of the encoded translation operator, giving a Fourier-space constructive form for PCS-equivariant quantum convolution. We then built a multiscale PCS-QCNN quantum core with measurement-conditioned pooling and analyzed its trainability. The trainability analysis separates the elementary parameter-count mechanism that suppresses the expected mean per-coordinate second moment from the empirical gradient norms directly relevant to optimization. This distinction matters because a per-sample RMS diagnostic does not include cancellations between examples, whereas training follows the gradient of the averaged empirical loss.

Empirically, the benchmarks show two signatures of encoding-aligned convolution under the controls used here. On translated-MNIST, the classical CNN/MLP control separates by 48.75 percentage points, showing that the task is sensitive to translation-aware inductive bias, and within the quantum family the PCS-QCNN exceeds the matched non-PCS RBC-QCNN by 35.08 percentage points. On the full-MNIST size sweep, all larger spatial resolutions outperform the smallest 8×8 configuration, with direct 32×32 preprocessing giving the highest final mean in the displayed sweep. The finite-shot diagnostics further show that shot budget can change the effective ranking of checkpoints, making sampling cost a deployment-relevant hyperparameter.

Taken together, these results support preserving the encoded PCS structure relative to a parameter-matched symmetry-broken quantum control in an idealized statevector benchmark. They do not settle PCS versus QCS empirically, hardware-native performance, or systematic finite-shot degradation across independently trained models. Those questions require matched QCS-equivariant baselines, compiled state-preparation and multiplexer circuits, structured/compressed PCS parameterizations, and shot/noise-aware training protocols.

Another open question is how far the architecture should be pushed toward an end-to-end quantum model. In the present hybrid design, the classical head provides inexpensive nonlinear decision layers, while a fully quantum replacement would need measurement-conditioned branching or other effectively non-unitary mechanisms and could significantly increase shot requirements. Whether such a fully quantum readout is advantageous under realistic shot budgets remains unresolved.

A second open challenge is hardware-native training. This study uses infinite-shot simulation for optimization and finite-shot sampling at inference; extending optimization itself to shot-limited, noisy hardware requires shot-efficient gradient estimation and update rules that stay stable in large multiplexed circuits. Related design questions include principled feature-growth policies across pooling levels and constraints on mode-dependent blocks that preserve symmetry while controlling depth and sampling cost.

Data and Code Availability

The code that supports the findings of this study is openly available in [36].

Acknowledgments

The research was supported by ITMO University Research Projects in AI Initiative (project 640111).

References

[1] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989) https://doi.org/10.1162/neco.1989.1.4.541

[2] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, ??? (2016). https://www.deeplearningbook.org

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

[4] Vidal, G.: Entanglement renormalization. Physical Review Letters **99**(22), 220405 (2007) https://doi.org/10.1103/PhysRevLett.99.220405

[5] Cong, I., Choi, S., Lukin, M.D.: Quantum convolutional neural networks. Nature Physics **15**(12), 1273–1278 (2019) https://doi.org/10.1038/s41567-019-0648-8

[6] Oh, S., Choi, J., Kim, J.K., Kim, J.: Quantum convolutional neural network for resource-efficient image classification: A quantum random access memory (QRAM) approach. In: Proceedings of the 2021 International Conference on Information Networking (ICOIN), pp. 50–52. IEEE, ??? (2021). https://doi.org/10.1109/ICOIN50884.2021.9333906

[7] Jing, Y., Li, X., Yang, Y., Wu, C., Fu, W., Hu, W., Li, Y., Xu, H.: RGB image classification with quantum convolutional ansatz. Quantum Information Processing **21**(3), 101 (2022) https://doi.org/10.1007/s11128-022-03442-8

[8] Huang, S.-Y., An, W.-J., Zhang, D.-S., Zhou, N.-R.: Image classification and adversarial robustness analysis based on hybrid quantum-classical convolutional neural network. Optics Communications **533**, 129287 (2023) https://doi.org/10.1016/j.optcom.2023.129287

[9] Wei, S.-J., Chen, Y.-H., Zhou, Z.-R., Long, G.-L.: A quantum convolutional neural network on NISQ devices. APL Photonics **7**(5), 056103 (2022) https://doi.org/10.1063/5.0083444

[10] Li, Y., Zhou, R.-G., Xu, R., Luo, J., Hu, W.: A quantum deep convolutional neural network for image recognition. Quantum Science and Technology **5**(4), 044003 (2020) https://doi.org/10.1088/2058-9565/ab9f93

[11] Li, W., Chu, P.-C., Liu, G.-Z., Tian, Y.-B., Qiu, T.-H., Wang, S.-M.: An image classification algorithm based on hybrid quantum classical convolutional neural network. Quantum Engineering **2022**, 5701479 (2022) https://doi.org/10.1155/2022/5701479

[12] Easom-McCaldin, P., Bouridane, A., Belatreche, A., Jiang, R.: On depth, robustness and performance using the data re-uploading single-qubit classifier. IEEE Access **9**, 65127–65139 (2021) https://doi.org/10.1109/ACCESS.2021.3075492

[13] Das, S., Caruso, F.: Permutation-equivariant quantum convolutional neural networks. Quantum Science and Technology **10**(1), 015030 (2024) https://doi.org/10.1088/2058-9565/ad8e80

[14] Larocca, M., Sauvage, F., Sbahi, F.M., Verdon, G., Coles, P.J., Cerezo, M.: Group-invariant quantum machine learning. PRX Quantum **3**(3), 030341 (2022) https://doi.org/10.1103/PRXQuantum.3.030341

[15] Meyer, J.J., Mularski, M., Gil-Fuster, E., Mele, A.A., Arzani, F., Wilms, A., Eisert, J.: Exploiting symmetry in variational quantum machine learning. PRX Quantum **4**(1), 010328 (2023) https://doi.org/10.1103/PRXQuantum.4.010328

[16] Schatzki, L., Larocca, M., Nguyen, Q.T., Sauvage, F., Cerezo, M.: Theoretical guarantees for permutation-equivariant quantum neural networks. npj Quantum Information **10**, 12 (2024) https://doi.org/10.1038/s41534-024-00804-1

[17] Chinzei, K., Tran, Q.H., Maruyama, K., Oshima, H., Sato, S.: Splitting and parallelizing of quantum convolutional neural networks for learning translationally symmetric data. Physical Review Research **6**(2), 023042 (2024) https://doi.org/10.1103/PhysRevResearch.6.023042

[18] Le, P.Q., Dong, F., Hirota, K.: A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quantum Information Processing **10**(1), 63–84 (2011) https://doi.org/10.1007/s11128-010-0177-y

[19] Zhang, Y., Lu, K., Gao, Y., Wang, M.: NEQR: A novel enhanced quantum representation of digital images. Quantum Information Processing **12**(8), 2833–2860 (2013) https://doi.org/10.1007/s11128-013-0567-z

[20] Bermejo, P., Braccia, P., Rudolph, M.S., Holmes, Z., Cincio, L., Cerezo, M.: Quantum convolutional neural networks are effectively classically simulable. PRX Quantum **7**, 020304 (2026) https://doi.org/10.1103/8qt9-72ts

[21] Gong, L.-H., Pei, J.-J., Zhang, T.-F., Zhou, N.-R.: Quantum convolutional neural network based on variational quantum circuits. Optics Communications **550**, 129993 (2024) https://doi.org/10.1016/j.optcom.2023.129993

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

[23] Pesah, A., Cerezo, M., Wang, S., Volkoff, T., Sornborger, A.T., Coles, P.J.: Absence of barren plateaus in quantum convolutional neural networks. Physical Review X **11**(4), 041011 (2021) https://doi.org/10.1103/PhysRevX.11.041011

[24] Gray, R.M.: Toeplitz and circulant matrices: A review. Foundations and Trends in Communications and Information Theory **2**(3), 155–239 (2006) https://doi.org/10.1561/0100000006

[25] Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, ??? (2013)

[26] Lü, Y., Gao, Q., Lü, J., Ogorzałek, M., Zheng, J.: A quantum convolutional neural network for image classification. In: Proceedings of the 40th Chinese Control Conference (CCC), pp. 6329–6334. IEEE, ??? (2021). https://doi.org/10.23919/CCC52363.2021.9550027

[27] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information:

10th Anniversary Edition. Cambridge University Press, Cambridge (2010)

[28] Cleve, R., Watrous, J.: Fast parallel circuits for the quantum fourier transform. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 526–536. IEEE, ??? (2000). https://doi.org/10.1109/SFCS.2000.892140

[29] Bergholm, V., Vartiainen, J.J., Möttönen, M., Salomaa, M.M.: Quantum circuits with uniformly controlled one-qubit gates. Physical Review A **71**(5), 052330 (2005) https://doi.org/10.1103/PhysRevA.71.052330

[30] Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum-logic circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**(6), 1000–1010 (2006) https://doi.org/10.1109/TCAD.2005.855930

[31] LeCun, Y., Cortes, C., Burges, C.J.C.: MNIST Handwritten Digit Database. http://yann.lecun.com/exdb/mnist/. Accessed 2026-04-23 (2010)

[32] Schuld, M., Killoran, N.: Is quantum advantage the right goal for quantum machine learning? PRX Quantum **3**(3), 030101 (2022) https://doi.org/10.1103/PRXQuantum.3.030101

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

[34] Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., Sadeghi, H., Amin, M.H.: Quantum variational autoencoder. Quantum Science and Technology **4**(1), 014001 (2018) https://doi.org/10.1088/2058-9565/aada1f

[35] Oseledets, I.V.: Tensor-train decomposition. SIAM Journal on Scientific Computing **33**(5), 2295–2317 (2011) https://doi.org/10.1137/090752286

[36] Chirkov, D., Lobanov, I.: Pixel-Translation-Equivariant Quantum Convolutional Neural Networks via Fourier Multiplexers. Zenodo. DOI: 10.5281/zenodo.19814130 (2026). https://doi.org/10.5281/zenodo.19814130 . https://doi.org/10.5281/zenodo.19814130

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplementary.pdf