DISCRETELY PARAMETRIZED QUANTUM CIRCUITS AS QUANTUM CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Parametrized quantum circuits (PQCs) are typically parametrized continuously, whereas low-resolution quantum control motivates restricting circuit parameters to a discrete set. This distinction can become consequential at low resolution, where projecting a continuously optimized circuit onto executable parameter values can substantially perturb the implemented circuit and degrade task performance. We therefore study *discretely parametrized quantum circuits* (DPQCs), whose trainable circuit parameters are defined and optimized directly over a discrete set, in contrast to continuously parametrized quantum circuits (CPQCs). To better preserve input structure under finite resolution, we introduce resolution-aware angular re-uploading, which progressively uploads input information across multiple resolution levels during classical data encoding into the circuit. We further develop spectrum-informed shift-compatible optimization (SISCO), which exploits the spectral structure of Pauli-generated PQCs to optimize discrete circuit parameters. Across multiple benchmarks, our DPQC framework generally outperforms alternative approaches and is particularly effective at low-resolution regime. With a nominal 5-bit resolution for circuit parametrization (i.e. $2^5 = 32$ executable values), our resolution-aware DPQCs trained with SISCO achieve 93.56%, 77.27%, and 86.56% mean accuracy on Imagenette, CIFAR-10, and STL-10, respectively, closely matching the corresponding CPQC references of 92.44%, 77.43%, and 86.62%. These results show that jointly designing data encoding and optimization for DPQCs can retain competitive performance without relying on continuous circuit parametrization, providing an algorithmic framework for variational quantum learning under low resolutions.

1 INTRODUCTION

PQCs form the computational backbone of many variational quantum algorithms. They encode classical data or quantum information into quantum states, transform these states through trainable unitary operations, and produce task-dependent outputs from quantum measurements. This paradigm has been explored across quantum chemistry Nakayama et al. (2025); Haidar et al. (2023); Tao et al. (2022), combinatorial optimization Ye et al. (2023); Zhou et al. (2023), drug discovery Batra et al. (2021); Li et al. (2021), and quantum machine learning Wang et al. (2026); Zhao et al. (2024). Training typically follows a hybrid quantum–classical loop, in which a quantum processor evaluates circuit outputs or gradient-related quantities and a classical optimizer updates the circuit parameters Farhi & Neven (2018); Peruzzo et al. (2014); Biamonte et al. (2017); Schuld et al. (2015); Cerezo et al. (2022); Huang et al. (2021). Most existing formulations assume continuously valued circuit parameters.

Circuit parameters, however, are ultimately realized through finite-resolution classical control electronics. Practical control stacks are constrained by factors including digital-to-analog converter (DAC) resolution, control bandwidth, waveform-memory capacity, and cryogenic power consumption Pauka et al. (2021); Xue et al. (2021); Prathapan et al. (2022). Rather than modeling a particular control stack, we adopt a circuit-level finite-resolution abstraction in which each trainable gate parameter is restricted to a finite set $\mathcal{A}_K$ containing K executable values. At coarse resolution, this restriction can become particularly consequential: parameters optimized over a continuous domain must subsequently be projected onto the executable set $\mathcal{A}_K$, potentially perturbing the implemented circuit and degrading task performance; see Appendix B for a detailed analysis.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

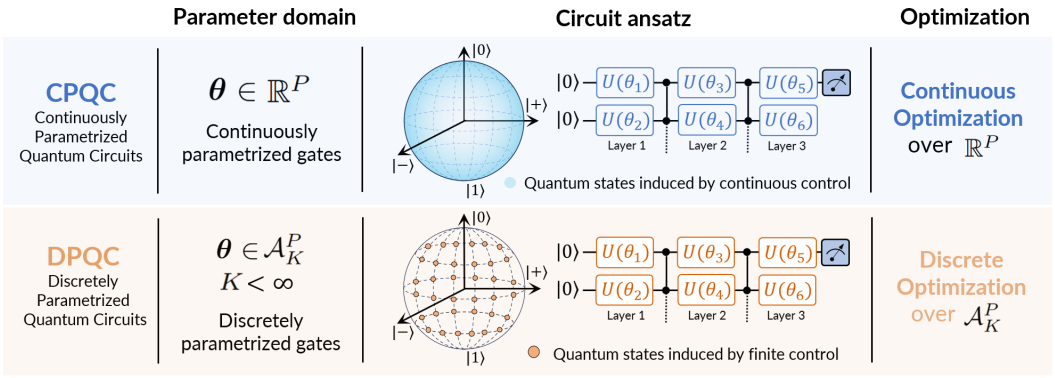

Figure 1: Conceptual comparison of CPQCs and DPQCs under the same circuit ansatz. CPQCs operate over continuous parameter spaces, whereas DPQCs restrict circuit parameters to the finite executable set $\mathcal{A}_K^P$, reducing control-resolution requirements.

We therefore formulate DPQCs, in which the trainable parameter vector is defined and optimized directly over the executable space $\mathcal{A}_K^P$ for a circuit with P parameters. Within this formulation, we address two challenges: (i) representing inputs effectively under finite-resolution encoding, and (ii) effective optimization over discrete circuit parameters. *For the former, we introduce resolution-aware angular re-uploading, which decomposes each classical input into resolution-ordered angular contributions and progressively uploads them across circuit depth.* Interleaved mixing unitaries couple these contributions, allowing the circuit to progressively integrate increasingly fine-grained input information and enable cross-resolution interaction. *For the latter, we develop SISCO, which exploits spectral structures of Puli-generated PQCs to optimize circuit parameters directly over $\mathcal{A}_K^P$.* In summary, our main contributions include:

- We formulate DPQCs as a framework whose circuit parametrization and optimization are constrained to a finite parameter set $\mathcal{A}_K$ motivated by finite-resolution quantum control;
- We introduce resolution-aware angular re-uploading, which distributes multi-resolution input components across circuit depth and couples them through inter-scale mixing unitaries, enabling a broader family of input-dependent spectral interactions;
- We develop SISCO, a discrete circuit optimizer that combines coordinate-wise spectral reconstruction with spectrum-informed multi-line search while maintaining all circuit evaluations and parameter updates within $\mathcal{A}_K^P$.

We evaluate the framework through controlled circuit-optimization benchmarks and multi-class image-classification tasks. The former provides statistically robust assessments of discrete optimization behavior across diverse circuit landscapes, while the latter evaluates the framework in representative downstream learning settings. *To our knowledge, this work provides the first study of PQC under finite resolution execution, incorporating the constraints jointly into input encoding, circuit parametrization, and discrete circuit optimization with evaluations across multi-class image-classification benchmarks.* We note that current QML experiments for image classifications operate at smaller scales than modern classical deep learning, with such experiments commonly focusing on relatively compact benchmarks such as MNIST Wu et al. (2023); Cong et al. (2019); Silver et al. (2023) and CIFAR-10 Bayal et al. (2026); Schleich et al. (2024). However, they are representative of the scale currently accessible to present-day QML, where qubit counts, simulation cost, and data-encoding and measurement overhead remain substantial constraints. *Accordingly, our objective is not to compete with large-scale classical vision models in absolute predictive accuracy, but to establish a principled framework for representation and optimization when PQCs operate under low resolutions.*

2 Related Work

Finite-Resolution Quantum Control. Quantum processors rely on classical control stacks to synthesize quantum gates Pauka et al. (2021); Krantz et al. (2019); Frank et al. (2022). These

stacks involve digital controllers, waveform memories, DACs, pulse-shaping modules, and cryogenic interfaces, and are subject to finite precision, bandwidth, and power constraints Pauka et al. (2021); Xue et al. (2021); Prathapan et al. (2022); Maurya & Tannu (2022). These considerations motivate algorithmic models in which gate parameters are restricted to a finite set of executable values. Probabilistic angle interpolation (PAI) Koczor et al. (2024) is also motivated by the finite-resolution setting, but addresses a different objective: it approximates expectation values of a continuously parametrized circuit by randomly executing nearby discrete gate settings. In contrast, we study DPQCs whose parameters are themselves defined and optimized over the discrete parameter space.

Quantum Data Encoding and Re-uploading. Data encoding plays a central role in the expressiveness of PQCs Schuld & Killoran (2019). Repeated data encoding enriches the input dependence of PQCs and can expand their accessible Fourier spectra Pérez-Salinas et al. (2020); Schuld et al. (2021). Within this broader paradigm, we design resolution-aware angular re-uploading as a structured generalization of finite-resolution angular encoding. Specifically, our encoder decomposes the input into resolution-ordered angular components and progressively uploads them from coarse to fine, with trainable mixing unitaries coupling successive levels. This design preserves conventional angular encoding as a special case while enabling additional cross-resolution spectral interactions.

Optimization of Discretely Parametrized Models. Discrete and low-precision optimization is well studied in classical neural networks, where quantized weights or activations are commonly trained using straight-through estimators (STEs) with continuously valued latent variables Hubara et al. (2018); Guo (2018); Yin et al. (2019); Huh et al. (2023); Qu et al. (2025). These approaches typically retain continuous optimization variables while enforcing discretization during forward execution. PQC optimization, in contrast, is predominantly formulated over continuous circuit parameters. Recent work Nádori et al. (2025) shows that profile-informed line-search strategies can effectively exploit nonlocal landscape information to navigate challenging PQC optimization landscapes. This motivates line search as a natural foundation for our discrete optimization setting. Building on this principle, our SISCO formulates the search directly over the executable space $\mathcal{A}_K^P$ and extends the single-line construction to multiple spectrum-informed directions. All circuit evaluations and parameter updates therefore remain within $\mathcal{A}_K^P$.

3 Background: Parametrized Quantum Circuits

An n-qubit pure quantum state is represented by a unit vector $|\psi\rangle \in \mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}$. A PQC prepares a quantum state by applying a parametrized unitary $U(\boldsymbol{\theta})$ to the computational basis state $|0^n\rangle$: $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0^n\rangle$, $\boldsymbol{\theta} \in \mathbb{R}^P$. Here, $\boldsymbol{\theta}$ collects the P trainable circuit parameters. Common variational architectures alternate parametrized Pauli rotations with fixed entangling operations. A representative L-layer circuit can be written as $U(\boldsymbol{\theta}) = \prod_{\ell=1}^{L} \left(E_\ell \prod_{j \in \mathcal{I}_\ell} R_{G_j}(\theta_j) \right)$, $R_{G_j}(\theta_j) = \exp\left(-\frac{i}{2}\theta_j G_j\right)$, where E_ℓ denotes a fixed entangling unitary, $\mathcal{I}_\ell$ indexes the parametrized gates in layer ℓ, and G_j is the Hermitian generator of the j-th rotation, typically a local Pauli operator or a tensor product thereof. Given a Hermitian observable O, the corresponding circuit output is $f(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|O|\psi(\boldsymbol{\theta})\rangle = \langle 0^n|U^\dagger(\boldsymbol{\theta})OU(\boldsymbol{\theta})|0^n\rangle$. One or more such expectation values can serve as quantum features, predictions, logits, energies, or objective estimates, depending on the downstream task. PQC training typically follows a hybrid quantum–classical loop: a quantum processor evaluates circuit outputs or gradient-related quantities, while a classical optimizer updates $\boldsymbol{\theta}$. Additional background on quantum computing and variational quantum circuits is provided in Appendix A.

4 Method

We consider PQCs whose trainable circuit parameters are discrete, which we refer to as DPQCs. Due to the periodic dependence of circuit outputs on gate parameters, we define this finite parameter set over a single period of 2π. For a resolution $K \in \mathbb{N}$, we define the spacing $\Delta_K = 2\pi/K$ and the finite parameter set $\mathcal{A}_K = \{\ell\Delta_K \mid \ell = 0, 1, \ldots, K-1\}$. Here, ℓ indexes elements of $\mathcal{A}_K$. For a circuit with P trainable parameters, we optimize $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P) \in \mathcal{A}_K^P$, so that trainable circuit configurations remain executable at the prescribed resolution. Our framework addresses two complementary aspects of learning under this constraint. We start by introducing resolution-aware angular re-uploading, which progressively encodes input information across multiple resolution levels

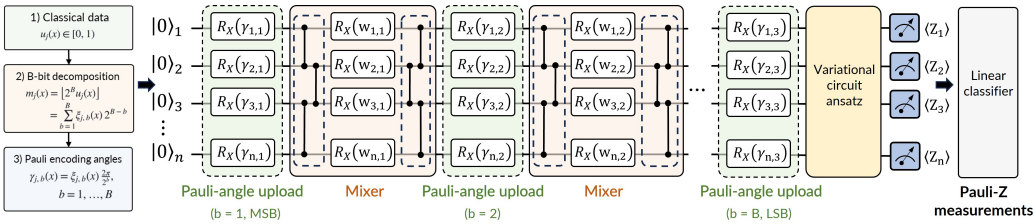

Figure 2: Overview of the proposed DPQC classifier. Inputs are decomposed into B-bit levels and encoded through resolution-aware Pauli rotations uploaded from coarse to fine. It is followed by a variational circuit ansatz, local Pauli-Z measurements, and a linear classifier producing output logits.

for enhanced circuit encoding expressivity; see Sec. 4.1. Then, we introduce *SISCO*, which exploits the spectral structure of PQCs to directly optimize the circuit parameters over $\mathcal{A}_K^P$; see Sec. 4.2.

4.1 Resolution-aware Angular Re-uploading

Classical inputs must be encoded into quantum states before they can be processed by a quantum circuit. A standard choice is angular encoding Mitarai et al. (2018); Schuld & Killoran (2019), which maps normalized inputs $u_j(\mathbf{x}) \in [0,1)$ to Pauli rotation angles: $u_j(\mathbf{x}) \mapsto R_\alpha(2\pi u_j(\mathbf{x}))$, $\alpha \in \{X,Y,Z\}$. Here, j indexes the input dimension and α specifies the rotation axis. For a DPQC, however, the Pauli angles used for input encoding are restricted to $\mathcal{A}_K$, which prevents encoding arbitrary continuous input values. A direct approach is to quantize the encoding angles to the resolution supported by $\mathcal{A}_K$ and perform finite-resolution angular encoding. Instead, we introduce resolution-aware angular re-uploading, which encodes input across progressively finer resolutions to expose its multi-resolution structure. *This construction retains conventional finite-resolution angular encoding as a special case, while yielding cross-resolution spectral interactions beyond the reach of the conventional approach; see Appendix C.* An exemplary spectral input structure comparison between our encoding and the conventional approach is provided in Fig. 3.

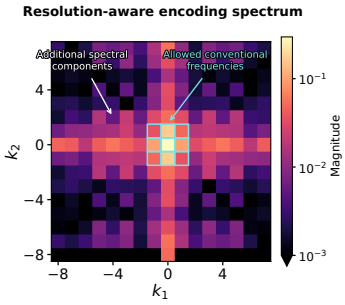

Figure 3: Illustration of enhanced encoding expressivity through the Fourier spectrum. The axes k_1, k_2 denote discrete Fourier frequency indices for two input dimensions.

Specifically, we represent $u_j(\mathbf{x}) \in [0,1)$ at B-bit resolution as $\widetilde{u}_j(\mathbf{x}) = \frac{m_j(\mathbf{x})}{2^B} = \sum_{b=1}^{B} \xi_{j,b}(\mathbf{x})\, 2^{-b}$, where $m_j(\mathbf{x}) = \lfloor 2^B u_j(\mathbf{x}) \rfloor$. $B \in \mathbb{N}$ denotes the input bit depth and $\xi_{j,b}(\mathbf{x}) \in \{0,1\}$ is the binary coefficient at resolution level b. This decomposition represents the input through a collection of multiscale contributions $\{\xi_{j,b}(\mathbf{x})2^{-b}\}_{b=1}^{B}$. We map these contributions to corresponding Pauli rotation angles for input encoding as $\boldsymbol{\gamma}_j(\mathbf{x}) = \{\gamma_{j,b}(\mathbf{x})\}_{b=1}^{B}$, $\gamma_{j,b}(\mathbf{x}) = \xi_{j,b}(\mathbf{x})\frac{2\pi}{2^b}$. Resulting angles $\gamma_{j,b}(\mathbf{x})$ naturally form a coarse-to-fine hierarchy: smaller b corresponds to larger angular contributions, whereas larger b introduces progressively finer corrections. To ensure introduced multi-scale Pauli angles $\gamma_{j,b}(\mathbf{x})$ remain within $\mathcal{A}_K$, we consider a nominal resolution of b_{ctrl} bits, corresponding to $K = 2^{b_{\text{ctrl}}}$. Choosing the input bit depth $B \le b_{\text{ctrl}} = \log_2 K$ then ensures that all resolution-level angle lies in $\mathcal{A}_K$.

We next upload these multiscale Pauli gates that encode input information from coarse to fine. At resolution level $b \in \{1,\dots,B\}$, the corresponding Pauli angles are applied across the n qubits through $U_{\text{up}}(\boldsymbol{\gamma}_b(\mathbf{x})) = \bigotimes_{j=1}^{n} R_X(\gamma_{j,b}(\mathbf{x}))$, where $\boldsymbol{\gamma}_b(\mathbf{x}) = (\gamma_{1,b}(\mathbf{x}),\dots,\gamma_{n,b}(\mathbf{x})) \in \mathcal{A}_K^n$ and $R_X(\gamma) = \exp(-\mathrm{i}\gamma X/2)$. Between consecutive resolutions indexed by b and $b+1$, we introduce a trainable mixing unitary $U_{\text{mix}}(\boldsymbol{w}_b) = E_{\text{CZ}}^{\dagger}[\bigotimes_{j=1}^{n} R_X(w_{b,j})]E_{\text{CZ}}$, where E_{CZ} is a fixed CZ-based entangling unitary and $\boldsymbol{w}_b \in \mathcal{A}_K^n$ contains the trainable mixer parameters. These trainable mixers need to be interleaved between successive resolution-level uploads such that they transform the intermediate state before the next finer-resolution component is introduced, allowing information from different resolution levels to interact throughout the encoding process. Combining the data

uploads with these mixing operations gives the complete data encoding module: $U_{\mathrm{enc}}(\mathbf{x};\mathbf{W}) = U_{\mathrm{up}}(\boldsymbol{\gamma}_B(\mathbf{x}))\prod_{b=B-1}^{1}\left[U_{\mathrm{mix}}(\boldsymbol{w}_b)U_{\mathrm{up}}(\boldsymbol{\gamma}_b(\mathbf{x}))\right]$, $\mathbf{W} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{B-1})$. A key structural property of our construction is that conventional angular encoding is recovered exactly when the inter-scale mixers are disabled. When all mixer parameters $w_{b,j}$ are set to zero, every mixing unitary reduces to the identity, $U_{\mathrm{mix}}(\boldsymbol{w}_b) = E_{\mathrm{CZ}}^{\dagger}\left[\bigotimes_{j=1}^{n} R_X(0)\right]E_{\mathrm{CZ}} = I$. The encoder therefore becomes $U_{\mathrm{enc}}(\mathbf{x};\mathbf{0}) = \bigotimes_{j=1}^{n} R_X\left(\sum_{b=1}^{B}\gamma_{j,b}(\mathbf{x})\right) = \bigotimes_{j=1}^{n} R_X(2\pi\widetilde{u}_j(\mathbf{x}))$, which is precisely conventional angular encoding under finite resolutions. *Thus, resolution-aware angular re-uploading preserves the conventional encoder as an exact member of its representational family, while nontrivial mixers enable additional cross-resolution spectral interactions from the same input.* We next introduce SISCO for directly optimizing the trainable DPQC parameters over $\mathcal{A}_K$.

4.2 Spectrum-Informed Shift-Compatible Optimization

We optimize trainable circuit parameters over $\mathcal{A}_K^P$. Since each θ_i parameterizes a Pauli-generated rotation, the circuit output is a first-order trigonometric polynomial in θ_i Ostaszewski et al. (2021); Wierichs et al. (2022). Its coordinate-wise output profile can be reconstructed at any $\alpha \in \mathcal{A}_K$ as

$$\boldsymbol{h}_i(\alpha) = \boldsymbol{h}_0 + \frac{\boldsymbol{h}_i^+ - \boldsymbol{h}_i^-}{2}\sin\delta_i(\alpha) + \left(\frac{\boldsymbol{h}_i^+ + \boldsymbol{h}_i^-}{2} - \boldsymbol{h}_0\right)(1-\cos\delta_i(\alpha)), \qquad \delta_i(\alpha) = \alpha - \theta_i. \tag{1}$$

Here, $\boldsymbol{h}_0$ is the current circuit output while $\boldsymbol{h}_i^{\pm}$ are obtained by shifting only θ_i by $\pm\pi/2$. We establish that these shifted evaluations remain in $\mathcal{A}_K$ whenever $4 \mid K$. Note that gradient extraction requires the same condition; see Appendix E. Applying the downstream objective to reconstructed outputs gives the coordinate-wise loss profile $\ell_i(\alpha)$ and loss reduction $g_i(\alpha) = \mathcal{L}(\boldsymbol{\theta}) - \ell_i(\alpha)$. Following the line-search principle for PQC optimization Nádori et al. (2025), we first construct a profile-informed search direction $\boldsymbol{c}^{(1)}$: $c_i^{(1)} \in \arg\max_{c\in\mathbb{Z}_K} g_i(\theta_i + \Delta_K c)$, $\boldsymbol{c}^{(1)} = (c_1^{(1)},\ldots,c_P^{(1)})$ collects coordinate-wise loss-optimal moves into profile-informed search directions.

While $\boldsymbol{c}^{(1)}$ captures the information from separable coordinate-wise loss profiles, its construction overlooks cross-parameter inseparable interaction structures. Our SISCO addresses this limitation via extending the single-line construction to an interaction-aware multi-line search over $\mathcal{A}_K^P$ with the hyperparameter M defining the number of search directions. At each circuit update, in addition to $\boldsymbol{c}^{(1)}$, we choose remaining $M-1$ directions to expose complementary cross-parameter interaction structure characterized by the circuit's Fourier responses.

To characterize these interactions, we exploit the Fourier structure of the PQC along a candidate search direction $\boldsymbol{c} \in \mathbb{Z}_K^P$. For any pair of parameters θ_i and θ_j, a line step $t \in \mathbb{Z}_K$ changes them by $\Delta_K t c_i$ and $\Delta_K t c_j$, respectively. *As each parameter controls a Pauli-generated rotation in the circuit, the interaction induced by jointly updating θ_i and θ_j along the search line can be characterized through the pairwise Fourier modes* $(\sigma,\tau) \in \{-1,1\}^2$:

$$[A_{ij}(c_i,c_j)]_{t,(\sigma,\tau)} = \left(e^{\mathrm{i}\sigma\Delta_K t c_i} - 1\right)\left(e^{\mathrm{i}\tau\Delta_K t c_j} - 1\right). \tag{2}$$

A detailed derivation is provided in Appendix D. The interaction pattern exposed over the complete search line is captured using the normalized Gram descriptor $\boldsymbol{v}(c_i,c_j) = \frac{\mathrm{vec}\left(A_{ij}(c_i,c_j)^H A_{ij}(c_i,c_j)\right)}{\|A_{ij}(c_i,c_j)^H A_{ij}(c_i,c_j)\|_F}$, with $\boldsymbol{v}(c_i,c_j) = \mathbf{0}$ whenever $c_i = 0$ or $c_j = 0$. Thus, $\{\boldsymbol{v}(c_i,c_j)\}_{i<j}$ characterizes the pairwise Fourier interaction patterns exposed by $\boldsymbol{c}$. We use these descriptors to construct subsequent search directions according to two criteria: each direction should be **(i)** promising under reconstructed coordinate-wise loss profiles with large predicted loss reductions $g_k(\cdot)$ and **(ii)** complementary to interaction patterns covered by previous directions. Specifically, given $\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(m-1)}$, we select

$$\boldsymbol{c}^{(m)} \in \arg\max_{\boldsymbol{c}\in\mathbb{Z}_K^P} \sum_{i<j}\left[\max_{t\in\mathbb{Z}_K}\sum_{k\in\{i,j\}} g_k\big(\theta_k + t\Delta_K c_k\big)\right]\left\|\left(\boldsymbol{I} - \boldsymbol{\Pi}_{ij}^{(m-1)}\right)\boldsymbol{v}(c_i,c_j)\right\|_2^2. \tag{3}$$

Here, $\boldsymbol{\Pi}_{ij}^{(m-1)}$ denotes the orthogonal projector onto the span of the nonzero descriptors $\{\boldsymbol{v}(c_i^{(r)},c_j^{(r)})\}_{r=1}^{m-1}$ induced by the previously selected lines. The first factor in Eq. 3 measures the profile-predicted utility of jointly updating parameters i and j, whereas the projection residual

$\left\| \left(\boldsymbol{I} - \boldsymbol{\Pi}_{ij}^{(m-1)} \right) \boldsymbol{v}(c_i, c_j) \right\|_2^2$ measures the interaction structure contributed by the candidate direction beyond that already represented by the selected lines. A larger residual therefore indicates greater Fourier complementarity. We solve Eq. 3 using standard discrete coordinate ascent Jäger & Schöbel (2020); Shen et al. (2015), sequentially optimizing each component of $\boldsymbol{c}$. After selecting all M lines, we perform the line search and update the parametrized circuit to the configuration with the lowest observed loss. Each SISCO update requires $2P$ shifted evaluations for coordinate-wise reconstruction, together with at most $M(K-1)$ additional evaluations along the selected search lines. Thus, $N_{\mathrm{SISCO}} \leq 2P + 1 + M(K-1)$, with further details provided in Appendix J. SISCO concentrates the evaluation budget on a small set of profile-informed and Fourier-complementary search lines rather than exploring the K^P configurations of $\mathcal{A}_K^P$ exhaustively.

5 Experiments

In this section, we evaluate the proposed DPQC framework from two complementary perspectives. First, we study controlled discrete circuit optimization on a large collection of randomly generated circuit optimization instances, which enables a robust statistical analysis of discrete optimization behavior independently of any particular downstream task. Second, we evaluate the complete framework on multi-class image classification using Imagenette Deng et al. (2009), CIFAR-10 Krizhevsky et al. (2009), and STL-10 Coates et al. (2011); it serves as a representative downstream QML application in the literature for evaluating how finite resolution affects representation and learning performance. Together, these experiments explore whether properly-designed and -trained DPQCs remain competitive relative to their continuously parametrized counterparts.

Experimental Setup. For the controlled optimization benchmark, following common practice in studies of variational-circuit training McClean et al. (2018); Cerezo et al. (2021); Larocca et al. (2022), we construct and evaluate independently generated circuit-optimization instances to characterize optimizer behavior across a broad collection of discrete landscapes. For image classification, as image data dimensionality typically exceeds what can be encoded into the qubits, we use a pretrained and frozen ResNet-18 to extract compact representations that are subsequently processed by a dressed quantum classifier with linear input and output mappings, following established practice Mari et al. (2020); Zhao et al. (2024). Under finite-resolution encoding, the outputs of the classical linear input mapping are projected to $\tilde{u}_j(\mathbf{x})$ before DPQC encoding with the projector treated as identity for gradient calculation in the backward pass. For the circuit ansatzes, We consider three representative choices: the hardware-efficient ansatz (HEA) Zhao et al. (2024), the strongly entangling ansatz (SEA) Schuld et al. (2020), and the Hadamard-staggered ansatz (HSA) Mari et al. (2020). Circuit parameters are initialized from a Gaussian distribution following Zhang et al. (2022) and optimized to minimize cross-entropy loss. Our experiments are performed with Pennylane Bergholm et al. (2018); physical-device experiments additionally use IBM quantum hardware.

Comparison Protocols. We evaluate the optimization and encoding components of our framework separately. *For discrete circuit optimization, we compare our SISCO with two representative projection-based straight-through-estimator (STE) baselines with: deterministic nearest-value projection and stochastic neighboring projection; see Sec. 2.* While originally developed for low-resolution training of quantized neural networks, we find that these methods naturally extend to our setting by maintaining continuously valued latent circuit parameters, executing their discrete projections in the forward pass, and using surrogate gradients for optimization. They therefore provide natural baselines for assessing whether our direct optimization using SISCO over $\mathcal{A}_K^P$ offers an advantage over continuous optimization with projected execution. *For input encoding, we compare our resolution-aware angular re-uploading with conventional angular encoding under the same finite resolution setting.*

Default Configuration. Unless otherwise specified, all experiments use $n = 4$ qubits and $L = 6$ variational layers following the HEA ansatz pattern. For DPQCs, we set K = 16, corresponding to $\Delta_{16} = 2\pi/16$ and restrict all trainable circuit parameters to $\mathcal{A}_{16}$. Inputs are encoded using $B = 4$ resolution levels, satisfying $2^B \mid K$. When K is varied, B is chosen to remain compatible with the prescribed executable parameter set. During training, we choose $M = 3$ search directions for SISCO per update. The linear mappings are optimized using gradient descent with a learning rate of 0.01. *Unless stated otherwise, results are reported over five independent seeded runs.*

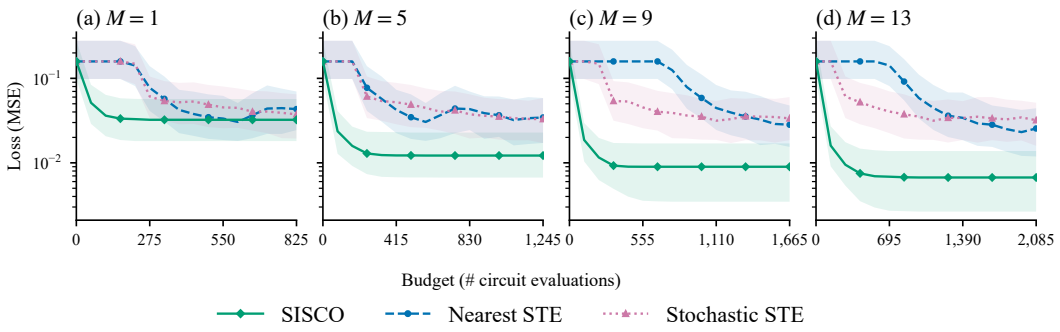

Figure 4: Budget-matched comparison of SISCO and STE baselines over 500 random circuit-optimization instances. Curves report MSE, with shaded regions denoting interquartile ranges.

Benchmarking on Random Circuit-Optimization Instances. We evaluate different optimizers through a controlled benchmark of 500 independently generated discrete circuit-optimization instances. This large collection enables a systematic assessment of optimizer behavior across discretely parametrized circuit landscape. For each instance, we independently sample a target configuration $\boldsymbol{\theta}^\star \in \mathcal{A}_K^P$ and an initial configuration $\boldsymbol{\theta}^{(0)} \in \mathcal{A}_K^P$, and minimize the mean squared error (MSE) between their circuit outputs. Lower MSE therefore indicates more effective discrete circuit optimization. We compare SISCO with the two STE baselines under matched circuit-evaluation budgets. For the STE methods, gradients for quantum circuits are evaluated using the parameter-shift rule, requiring two shifted circuit evaluations per trainable parameter, or $2P$ evaluations per gradient update. As detailed in Sec. 4.2, SISCO uses the same $2P$ shifted evaluations to reconstruct the coordinate-wise profiles and incurs at most an additional $M(K-1)$ evaluations for its M discrete search lines. We therefore align the methods by cumulative circuit evaluations for fair comparisons. For SISCO, we consider $M \in \{1, 5, 9, 13\}$ to study the effect of additional search directions. Note that SISCO falls back to the discrete adaption of the line-search principle Nádori et al. (2025) when $M = 1$, which uses the profile-informed direction without additional Fourier-complementary search lines. Fig. 4 reports the average MSE and interquartile range across the 500 instances. Under matched circuit-evaluation budgets, our SISCO exhibits faster loss reduction than the gradient-based STE baselines. *With $M = 1$, the final solution qualities remain broadly comparable to the STE methods. As M increases, SISCO consistently attains lower MSE not reachable by STE methods.* We further examine in Appendix F whether our proposed Fourier-complementarity criterion more effectively identifies parameter-interaction structure as designed when selecting additional search directions.

Classification Behavior under Finite Resolution Settings. We next consider the two-spiral classification task Lang (1989), a widely used QML benchmark with a highly nonlinear decision boundary Mari et al. (2020); Altmann et al. (2023); Li et al. (2022); Dudas et al. (2026). Its two-dimensional input domain enables direct visualization of how finite resolution affects both optimization dynamics and the learned decision boundary. We first construct an idealized reference by training a CPQC and project its converged parameters onto $\mathcal{A}_K^P$. *This post-training projection serves only as a diagnostic: it isolates the effect of finite-resolution execution while still assuming access to continuously valued circuit parameters during training, an assumption that is incompatible with the finite-resolution setting considered here.*

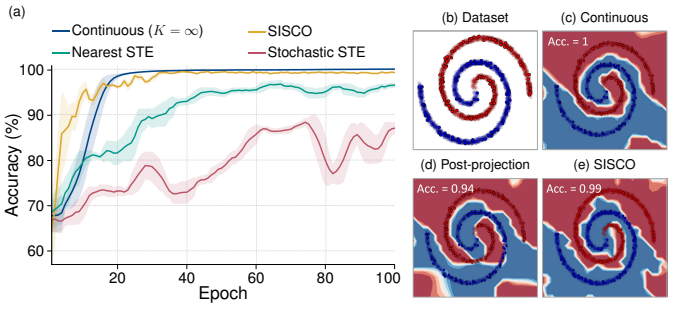

Figure 5: Training dynamics for the first 100 epochs and decision geometry on two-spiral classification. (a) Training accuracy of SISCO and the STE baselines, with the corresponding CPQC as the reference. (b) Two-spiral dataset. (c–e) Decision boundaries obtained by continuous training, its post-training projection, and direct discrete training with SISCO, respectively.

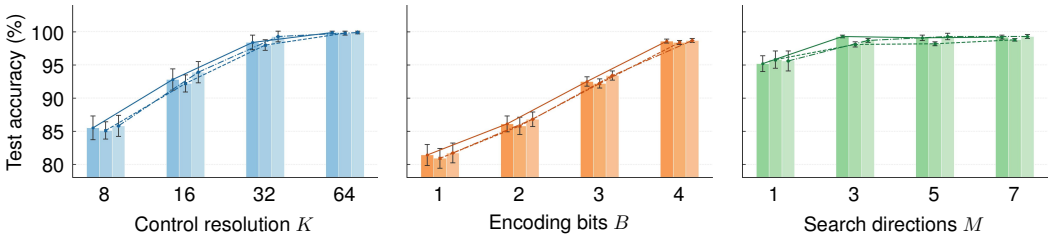

Figure 6: Hyperparameter sensitivity of DPQC training. We show behavior of test accuracy w.r.t. finite-resolution K (left), input encoding depth B (middle), and number of SISCO search directions M (right), respectively.

Fig. 5 compares the resulting optimization behavior and decision boundaries. Fig. 5-(a) compares the optimization trajectories of SISCO and the two STE baselines against the continuously parametrized circuit reference trained with standard gradient descent. *SISCO converges substantially faster than the projection-based STE methods and closely approaches the CPQC performance.* Fig. 5-(b–e) provide a complementary view with the learned decision geometry: while the continuously trained CPQC produces an accurate separator, post-training projection visibly distorts the learned boundary under finite-resolution execution. In contrast, direct DPQC training with SISCO learns a decision geometry that closely resembles the CPQC reference. *Overall, these results demonstrate the effectiveness of direct discrete optimization with SISCO, which achieves high classification accuracy and learns a decision boundary closely aligned with the continuous-control reference under finite-resolution constraints.*

We next evaluate our resolution-aware angular re-uploading against conventional angular encoding under the same finite resolution and otherwise identical training settings; results are reported in Tab. 1. Resolution-aware angular re-uploading consistently reduces task loss across SISCO and both STE baselines, suggesting that the improvement arises from the encoding architecture itself rather than a specific circuit optimizer. *This observation is consistent with our theoretical result in Appendix C: conventional angular encoding is recovered exactly as the identity-mixer special case of our resolution-aware angular encoding, whereas noncommuting mixers can introduce additional cross-resolution spectral interactions from the same input.*

Table 1: Effect of resolution-aware input encoding.

Encoding	Nearest STE	Stochastic STE	SISCO
Conventional	0.32 ± 0.16	0.21 ± 0.16	0.16 ± 0.02
Resolution-aware	$\mathbf{0.17 \pm 0.1}$	$\mathbf{0.17 \pm 0.05}$	$\mathbf{0.14 \pm 0.1}$

Hyperparameter Sensitivity. We examine the sensitivity of DPQC performance to the resolution K, encoding depth B, and number of search lines M, varying one factor at a time while keeping the others fixed; results are summarized in Fig. 6. We first vary $K \in \{8, 16, 32, 64\}$ and fix $B = 3$ as the largest encoding depth (see Sec. 4.1 for details) supported across all resolutions in this sweep. Classification accuracy improves as the execution resolution gets finer (i.e. larger K) and starts to saturate at $K = 32$. We next fix $K = 16$ and vary $B \in \{1, 2, 3, 4\}$. Performance improves consistently as the encoder incorporates progressively finer resolution levels together with their mixing operations, supporting the benefit of the proposed coarse-to-fine encoding structure. Finally, we vary M, the number of search lines evaluated at each circuit update. Increasing M improves solution quality, while most of the gain is obtained with only a few additional directions. This indicates that SISCO captures much of the interaction geometry relevant to effective search with a compact multi-line construction, achieving strong optimization quality at moderate circuit-evaluation cost in the tested setting.

Multi-class Image Classifications. We evaluate the complete DPQC framework on three multi-class image classification benchmarks: Imagenette, CIFAR-10, and STL-10, using experimental setups described above in Sec. 5. Consistent with preceding experiments, the corresponding CPQC serves as the continuously parametrized reference, while all DPQC methods operate under the prescribed finite resolution. Tab. 2 characterizes the converged classification accuracy across resolutions, encoding schemes, and optimization methods, with all training methods using the same number of circuit-parameter updates. We observe three consistent trends. First, our resolution-aware angular re-uploading predominantly improves classification accuracy over conventional angular encoding across

Table 2: Image-classification accuracy (%) with the strongly entangling ansatz.

Dataset	CPQC	Encoding	Training protocol	$K = 8$	$K = 16$	$K = 32$
Imagenette	92.44 ± 2.28	Conventional	Nearest STE	83.70 ± 6.55	90.65 ± 3.10	91.36 ± 2.41
			Stochastic STE	79.77 ± 10.20	90.73 ± 4.87	91.77 ± 2.12
			SISCO	$\mathbf{89.25 \pm 6.07}$	$\mathbf{91.76 \pm 1.23}$	$\mathbf{92.64 \pm 1.62}$
		Resolution-aware	Nearest STE	90.25 ± 3.96	93.32 ± 1.83	93.44 ± 1.03
			Stochastic STE	87.55 ± 5.80	92.52 ± 3.52	91.77 ± 1.52
			SISCO	$\mathbf{92.39 \pm 2.24}$	$\mathbf{93.38 \pm 0.25}$	$\mathbf{93.56 \pm 1.06}$
CIFAR-10	77.43 ± 1.49	Conventional	Nearest STE	69.56 ± 4.73	73.57 ± 3.56	74.74 ± 0.51
			Stochastic STE	67.30 ± 5.95	73.19 ± 4.31	75.53 ± 0.92
			SISCO	$\mathbf{72.46 \pm 2.21}$	$\mathbf{74.78 \pm 2.41}$	$\mathbf{76.37 \pm 1.58}$
		Resolution-aware	Nearest STE	71.18 ± 5.70	75.90 ± 2.79	76.59 ± 0.79
			Stochastic STE	$\mathbf{74.47 \pm 1.55}$	$\mathbf{76.10 \pm 2.12}$	74.36 ± 2.36
			SISCO	74.12 ± 1.93	75.82 ± 0.34	$\mathbf{77.27 \pm 1.14}$
STL-10	86.62 ± 2.06	Conventional	Nearest STE	76.46 ± 8.79	81.92 ± 5.54	84.07 ± 3.39
			Stochastic STE	69.20 ± 9.13	78.27 ± 7.03	83.31 ± 3.72
			SISCO	$\mathbf{80.62 \pm 5.58}$	$\mathbf{84.21 \pm 2.58}$	$\mathbf{85.33 \pm 2.44}$
		Resolution-aware	Nearest STE	83.00 ± 5.25	83.04 ± 2.64	86.08 ± 1.93
			Stochastic STE	78.06 ± 8.48	80.96 ± 5.12	85.84 ± 1.66
			SISCO	$\mathbf{83.96 \pm 3.49}$	$\mathbf{86.21 \pm 1.20}$	$\mathbf{86.56 \pm 1.73}$

Bold: best training protocol within each encoding using the full seed set;
Shading: better encoding under the same training protocol using the full matched seed set.

datasets, resolutions, and optimization methods. The gain persists for our SISCO and alternative STE baselines, suggesting that the proposed encoding improves the PQC capacity to model finite-resolution inputs rather than benefiting a particular optimizer. This observation is consistent with theoretical result in Appendix C and observations of our previous experiments, which establishes that our resolution-aware angular re-uploading introduces richer input-dependent cross-resolution spectral interactions. Moreover, SISCO remains competitive across resolutions and is particularly effective in the coarse-resolution regime, achieving the best performance mostly at $K = 8$. With same number of circuit-parameter updates, this shows that our SISCO reaches better converged solutions when the parameter space is restricted. Finally, SISCO generally exhibits lower across-seed variability than the STE baselines, suggesting more stable optimization in addition to improved average performance. Compared with the continuously parametrized CPQC reference, across all three datasets, we observe that the gap to CPQC generally narrows as the resolution increases. *With resolution-aware encoding and SISCO, the DPQC at* $K = 32$ *reaches* 93.56%*,* 77.27%*, and* 86.56% *accuracy on Imagenette, CIFAR-10, and STL-10, respectively, compared with CPQC accuracies of* 92.44%*,* 77.43%*, and* 86.62%*; in all three cases, the difference in mean accuracy lies within one reported standard deviation of the corresponding CPQC reference, with the DPQC slightly exceeding the CPQC mean on Imagenette.* These results indicate that, at moderate resolutions, appropriately designed encoding and discrete optimization can retain predictive performance closely comparable to continuously parametrized circuits without requiring continuously valued circuit parameters.

6 Discussion and Conclusion

This work studies variational circuit learning under finite-resolution execution. Within this framework, we introduce two components: resolution-aware angular re-uploading for structured input encoding across progressively finer resolution levels, and SISCO for circuit optimization over the executable parameter space. On randomized circuit-optimization instances, SISCO achieves faster loss reduction under matched circuit-evaluation budgets, while the controlled ablation confirms the benefit of Fourier-complementary search directions. Across encoding comparisons, resolution-aware angular re-uploading improves performance across different optimization methods, indicating an enhanced ability of the circuit to model finite resolution inputs. On image classifications, with merely a nominal 5-bit resolution, our complete DPQC framework can closely match the corresponding CPQC references across selected datasets. Together, these results show that properly designed input encoding and optimization can enable DPQCs under finite-resolution execution to retain predictive performance close to continuously parametrized circuits. Our work therefore provides a principled algorithmic framework for variational circuit learning under limited parametrization resolutions. We include a discussion on limitations of our method and potential future work in Appendix H.

AI USE STATEMENT

In this work, we used generative AI tools to: [1] provide feedback on the presentation of the methodology and experiments; [2] support data analysis. We have not used generative AI tools for: [1] generate synthetic data sets, [2] help develop theoretical models or conceptual frameworks, [3] formulate mathematical claims, [4] provide critical ingredients for proving mathematical claims, [5] assist in the writing of proofs, [6] propose or refine hypotheses, [7] design or provide feedback on research methodology or experiments, [8] implement methods, [9] assist with translation, [10] clean and reformat dataset, [11] interpret results. We have reviewed all AI-assisted work. We checked and verified LLM-generated code and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the methodological, theoretical, and experimental details required to reproduce the results throughout the main paper and supplementary material. The complete definitions of the DPQC framework, resolution-aware angular re-uploading, and SISCO are given in the Method section, with the corresponding theoretical analyses, derivations, and proofs provided in the appendix. Experimental settings, including datasets, model and circuit configurations, initialization, optimization hyperparameters, training schedules, are specified in the experimental setup The appendix further provides the complete training procedure, circuit-evaluation accounting, hardware execution details. All datasets used in this work are publicly available and cited in the main text. Source code and trained checkpoints will be released upon acceptance.

REFERENCES

Philipp Altmann, Leo Sünkel, Jonas Stein, Tobias Müller, Christoph Roch, and Claudia Linnhoff-Popien. Sequent: Towards traceable quantum machine learning using sequential quantum enhanced training. *arXiv preprint arXiv:2301.02601*, 2023.

Kushal Batra, Kimberley M Zorn, Daniel H Foil, Eni Minerali, Victor O Gawriljuk, Thomas R Lane, and Sean Ekins. Quantum machine learning algorithms for drug discovery applications. *Journal of chemical information and modeling*, 61(6):2641–2647, 2021.

Shashank Bayal, Rushikesh Govind Dawane, Komal Komal, Santosh Kumar Vipparthi, and Subrahmanyam Murala. Queenet: Quantum-enhanced expressive network for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7883–7892, 2026.

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, Bharath AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, 12 (1):1791, 2021.

Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature computational science*, 2(9):567–576, 2022.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Julien Dudas, Baptiste Carles, Elie Gouzien, Julie Grollier, and Danijela Marković. Training the parametric interactions in an analog bosonic quantum neural network with fock basis measurement. *Scientific Reports*, 2026.

Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

David J Frank, Sudipto Chakraborty, Kevin Tien, Pat Rosno, Thomas Fox, Mark Yeck, Joseph A Glick, Raphael Robertazzi, Ray Richetta, John F Bulzacchelli, et al. A cryo-cmos low-power semi-autonomous qubit state controller in 14nm finfet technology. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pp. 360–362. IEEE, 2022.

Yunhui Guo. A survey on methods and theories of quantized neural networks. *arXiv preprint arXiv:1808.04752*, 2018.

Mohammad Haidar, Marko J Rančić, Thomas Ayral, Yvon Maday, and Jean-Philip Piquemal. Open source variational quantum eigensolver extension of the quantum learning machine for quantum chemistry. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 13(5):e1664, 2023.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *journal of machine learning research*, 18(187):1–30, 2018.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*, pp. 14096–14113. PMLR, 2023.

Sven Jäger and Anita Schöbel. The blockwise coordinate descent method for integer programs: S. jäger, a. schöbel. *Mathematical Methods of Operations Research*, 91(2):357–381, 2020.

Bálint Koczor, John J. L. Morton, and Simon C. Benjamin. Probabilistic interpolation of quantum rotation angles. *Physical Review Letters*, 132(13):130602, 2024. doi: 10.1103/PhysRevLett.132.130602.

Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied physics reviews*, 6(2), 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Kevin J Lang. Learning to tell two spirals apart. In *Proceedings of the 1988 connectionist models summer school*, pp. 52–59. Morgan Kaufmann, 1989.

Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J Coles, and Marco Cerezo. Diagnosing barren plateaus with tools from quantum optimal control. *Quantum*, 6: 824, 2022.

Junde Li, Mahabubul Alam, M Sha Congzhou, Jian Wang, Nikolay V Dokholyan, and Swaroop Ghosh. Drug discovery approaches using quantum machine learning. In *2021 58th ACM/IEEE design automation conference (DAC)*, pp. 1356–1359. IEEE, 2021.

Zejian Li, Valentin Heyraud, Kaelan Donatella, Zakari Denis, and Cristiano Ciuti. Machine learning via relativity-inspired quantum dynamics. *Physical Review A*, 106(3):032413, 2022.

Andrea Mari, Thomas R Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4:340, 2020.

Satvik Maurya and Swamit Tannu. Compaqt: Compressed waveform memory architecture for scalable qubit control. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1059–1077. IEEE, 2022.

Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812, 2018.

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

Jakab Nádori, Gregory Morse, Barna Fülöp Villám, Zita Majnay-Takács, Zoltán Zimborás, and Péter Rakyta. Batched line search strategy for navigating through barren plateaus in quantum circuit training. *Quantum*, 9:1841, 2025.

Akimoto Nakayama, Kosuke Mitarai, Leonardo Placidi, Takanori Sugimoto, and Keisuke Fujii. Vqe-generated quantum circuit dataset for machine learning. *Physical Review Research*, 7(3): 033048, 2025.

Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, 2021.

SJ Pauka, K Das, R Kalra, A Moini, Y Yang, M Trainer, A Bousquet, C Cantaloube, N Dick, GC Gardner, et al. A cryogenic cmos chip for generating control signals for multiple qubits. *Nature Electronics*, 4(1):64–70, 2021.

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

Mridula Prathapan, Peter Mueller, Christian Menolfi, Matthias Brändli, Marcel Kossel, Pier Andrea Francese, David Heim, Maria Vittoria Oropallo, Andrea Ruffino, Cezar Zota, et al. A cryogenic sram based arbitrary waveform generator in 14 nm for spin qubit control. In *ESSCIRC 2022-IEEE 48th European Solid State Circuits Conference (ESSCIRC)*, pp. 57–60. IEEE, 2022.

Xiaoyi Qu, David Aponte, Colby Banbury, Daniel P Robinson, Tianyu Ding, Kazuhito Koishida, Ilya Zharkov, and Tianyi Chen. Automatic joint structured pruning and quantization for efficient neural network training and compression. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15234–15244. IEEE, 2025.

Philipp Schleich, Marta Skreta, Lasse B Kristensen, Rodrigo A Vargas-Hernández, and Alán Aspuru-Guzik. Quantum deep equilibrium models. *Advances in Neural Information Processing Systems*, 37:31940–31967, 2024.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 37–45, 2015.

Daniel Silver, Tirthak Patel, William Cutler, Aditya Ranjan, Harshitta Gandhi, and Devesh Tiwari. Mosaiq: Quantum generative adversarial networks for image generation on nisq computers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7030–7039, 2023.

Yanxian Tao, Xiongzhi Zeng, Yi Fan, Jie Liu, Zhenyu Li, and Jinlong Yang. Exploring accurate potential energy surfaces via integrating variational quantum eigensolver with machine learning. *The Journal of Physical Chemistry Letters*, 13(28):6420–6426, 2022.

Shuteng Wang, Christian Theobalt, and Vladislav Golyanik. Quantum visual fields with neural amplitude encoding. *Advances in Neural Information Processing Systems*, 38:158535–158559, 2026.

David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, 2022.

Wenjie Wu, Ge Yan, Xudong Lu, Kaisen Pan, and Junchi Yan. Quantumdarts: differentiable quantum architecture search for variational quantum algorithms. In *International conference on machine learning*, pp. 37745–37764. PMLR, 2023.

Xiao Xue, Bishnu Patra, Jeroen PG van Dijk, Nodar Samkharadze, Sushil Subramanian, Andrea Corna, Brian Paquelet Wuetz, Charles Jeon, Farhana Sheikh, Esdras Juarez-Hernandez, et al. Cmos-based cryogenic control of silicon quantum circuits. *Nature*, 593(7858):205–210, 2021.

Xinyu Ye, Ge Yan, and Junchi Yan. Towards quantum machine learning for constrained combinatorial optimization: a quantum qap solver. In *International Conference on Machine Learning*, pp. 39903–39912. PMLR, 2023.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Kaining Zhang, Liu Liu, Min-Hsiu Hsieh, and Dacheng Tao. Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits. *Advances in Neural Information Processing Systems*, 35:18612–18627, 2022.

Jiaming Zhao, Wenbo Qiao, Peng Zhang, and Hui Gao. Quantum implicit neural representations. *arXiv preprint arXiv:2406.03873*, 2024.

Zeqiao Zhou, Yuxuan Du, Xinmei Tian, and Dacheng Tao. Qaoa-in-qaoa: solving large-scale maxcut problems on small quantum machines. *Physical Review Applied*, 19(2):024027, 2023.

Supplementary Material

A Gate-Based Quantum Computing and Parametrized Quantum Circuits

We provide the basic concepts of gate-based quantum computing and parametrized quantum circuits used throughout this work.

Quantum States. The elementary unit of quantum information is a quantum bit, or qubit. A pure single-qubit state is represented by a normalized vector in the two-dimensional Hilbert space $\mathbb{C}^2$,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad \alpha, \beta \in \mathbb{C}, \qquad |\alpha|^2 + |\beta|^2 = 1, \tag{4}$$

where $|0\rangle = [1, 0]^\top$ and $|1\rangle = [0, 1]^\top$ form the computational basis. For an n-qubit system, the Hilbert space is

$$\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}, \tag{5}$$

with dimension 2^n. A general pure state can therefore be written as

$$|\psi\rangle = \sum_{\mathbf{b}\in\{0,1\}^n} c_{\mathbf{b}}\,|\mathbf{b}\rangle\,, \qquad \sum_{\mathbf{b}\in\{0,1\}^n} |c_{\mathbf{b}}|^2 = 1, \tag{6}$$

where $|\mathbf{b}\rangle = |b_1\rangle \otimes \cdots \otimes |b_n\rangle$ denotes a computational-basis state. Quantum states can more generally be represented by density operators. A pure state $|\psi\rangle$ corresponds to

$$\rho = |\psi\rangle\langle\psi|\,, \tag{7}$$

whereas a statistical mixture $\{p_j, |\psi_j\rangle\}$ is represented as

$$\rho = \sum_j p_j\,|\psi_j\rangle\langle\psi_j|\,, \qquad p_j \geq 0, \qquad \sum_j p_j = 1. \tag{8}$$

A valid density operator is positive semidefinite and satisfies $\mathrm{Tr}(\rho) = 1$.

Gate-based Quantum Computation. In the gate-based model, quantum states are transformed by unitary operators. An n-qubit quantum gate is represented by $U \in \mathbb{C}^{2^n \times 2^n}$ satisfying

$$U^\dagger U = U U^\dagger = I_{2^n}. \tag{9}$$

Applying U to a pure state gives $|\psi'\rangle = U\,|\psi\rangle$, while a density operator evolves as $\rho' = U\rho U^\dagger$. A quantum circuit is formed by composing a sequence of unitary gates. For gates $U_1, \ldots, U_T$ applied sequentially, the resulting circuit unitary is

$$U = U_T U_{T-1} \cdots U_1. \tag{10}$$

Common single-qubit gates include the Pauli operators

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{11}$$

together with the identity operator I. Parametrized rotations generated by Pauli operators take the form

$$R_G(\theta) = \exp\!\left(-\frac{i}{2}\theta G\right), \qquad G \in \{X, Y, Z\}. \tag{12}$$

In particular,

$$R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{13}$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \qquad R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \tag{14}$$

The angle θ determines the applied rotation and commonly serves as a trainable circuit parameter in variational quantum algorithms. Multi-qubit gates introduce interactions and entanglement among qubits. Two standard examples are the controlled-NOT and controlled-Z gates,

$$\mathrm{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X, \tag{15}$$

and

$$\mathrm{CZ} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z. \tag{16}$$

Parametrized single-qubit rotations together with fixed multi-qubit entangling gates form the basic building blocks of the circuit architectures considered in this work.

Quantum Measurement. Quantum measurement provides the interface through which classical information is extracted from a quantum state. An observable is represented by a Hermitian operator $O = O^\dagger$. For a pure state $|\psi\rangle$, its expectation value is

$$\langle O \rangle_\psi = \langle \psi | O | \psi \rangle , \tag{17}$$

or, equivalently, for a density operator,

$$\langle O \rangle_\rho = \mathrm{Tr}(O\rho). \tag{18}$$

On physical quantum hardware, expectation values are estimated from repeated circuit executions and measurements. Many observables of practical interest can be decomposed into Pauli strings,

$$O = \sum_j c_j P_j, \qquad P_j \in \{I, X, Y, Z\}^{\otimes n}, \tag{19}$$

so that their expectation values can be estimated through measurements in the corresponding Pauli bases. In quantum machine learning, collections of local Pauli expectation values are commonly used as circuit outputs and subsequently mapped to task-dependent predictions.

Parametrized Quantum Circuits. Parametrized quantum circuits introduce trainable parameters into the gate sequence and form the computational core of variational quantum algorithms. A circuit with P trainable parameters implements a unitary

$$U(\boldsymbol{\theta}), \qquad \boldsymbol{\theta} = (\theta_1, \ldots, \theta_P) \in \mathbb{R}^P, \tag{20}$$

which prepares the state

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\psi_0\rangle . \tag{21}$$

A common layered architecture alternates parametrized rotations with fixed entangling operations,

$$U(\boldsymbol{\theta}) = \prod_{\ell=1}^{L} \left[E_\ell \prod_{j \in \mathcal{I}_\ell} R_{G_j}(\theta_j) \right], \tag{22}$$

where E_ℓ denotes the entangling unitary in layer ℓ, $\mathcal{I}_\ell$ indexes its parametrized rotations, and G_j is the corresponding Hermitian generator. When processing classical data, an input $\mathbf{x}$ is incorporated through an input-dependent unitary, giving

$$|\psi(\mathbf{x}; \boldsymbol{\theta})\rangle = U(\mathbf{x}; \boldsymbol{\theta}) |0\rangle^{\otimes n} . \tag{23}$$

For a Hermitian observable O, the corresponding circuit output is

$$f_O(\mathbf{x}; \boldsymbol{\theta}) = \langle \psi(\mathbf{x}; \boldsymbol{\theta}) | O | \psi(\mathbf{x}; \boldsymbol{\theta}) \rangle . \tag{24}$$

A collection of such expectation values can be used as task-dependent circuit outputs. Training typically follows a hybrid quantum–classical procedure. The quantum processor evaluates circuit-dependent quantities for the current parameters, while a classical optimizer updates $\boldsymbol{\theta}$ to minimize an objective

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \tag{25}$$

For quantum machine learning, $\mathcal{L}$ is generally defined from the circuit predictions and the corresponding training labels.

Parameter-shift Structure. An important property of Pauli-generated rotations is that derivatives of measured expectation values can be obtained from shifted circuit evaluations. For a parameter θ_j entering through

$$R_{G_j}(\theta_j) = \exp\left(-\frac{i}{2}\theta_j G_j\right), \tag{26}$$

with a Pauli generator G_j, the parameter-shift rule gives

$$\frac{\partial f_O}{\partial \theta_j} = \frac{1}{2}\left[f_O\left(\theta_j + \frac{\pi}{2}\right) - f_O\left(\theta_j - \frac{\pi}{2}\right)\right], \tag{27}$$

where all remaining circuit parameters are held fixed. The same spectral structure implies that, with all other parameters fixed, the circuit output is a first-order trigonometric polynomial in θ_j. This coordinate-wise dependence is central to the shift-compatible spectral reconstruction used by SISCO in Sec. 4.2.

B Finite-Resolution Projection and Circuit Perturbation

We provide a characterization of how the resolution K bounds the perturbation induced when continuously valued circuit parameters are projected onto the executable parameter set $\mathcal{A}_K = \{\ell\Delta_K\}_{\ell=0}^{K-1}$, where $\Delta_K = 2\pi/K$. Let $Q_K(\alpha)$ denote the nearest executable value to $\alpha \in \mathbb{R}$ under the periodic angular distance, and apply Q_K elementwise to a parameter vector. Since adjacent values in $\mathcal{A}_K$ are separated by Δ_K, every angle satisfies

$$d_{2\pi}(\alpha, Q_K(\alpha)) \le \frac{\Delta_K}{2} = \frac{\pi}{K}, \tag{28}$$

where $d_{2\pi}(\alpha, \beta) = \min_{r\in\mathbb{Z}} |\alpha - \beta + 2\pi r|$ denotes the periodic angular distance. Thus, K directly controls the maximum angular perturbation introduced by projection onto the executable parameter set.

This angular bound also induces a corresponding perturbation bound at the circuit level. Consider a parametrized circuit $U(\theta)$ containing P trainable Pauli-generated rotations $R_{G_i}(\theta_i) = \exp(-i\theta_i G_i/2)$, with $G_i^2 = I$, interleaved with arbitrary fixed unitaries. For any parameter θ_i,

$$\begin{aligned} \|R_{G_i}(\theta_i) - R_{G_i}(Q_K(\theta_i))\|_2 &= 2\left|\sin\left(\frac{d_{2\pi}(\theta_i, Q_K(\theta_i))}{4}\right)\right| \\ &\le \frac{1}{2} d_{2\pi}(\theta_i, Q_K(\theta_i)) \le \frac{\pi}{2K}. \end{aligned} \tag{29}$$

Using a telescoping decomposition of the two circuit products and the unitarity of all remaining gates therefore gives

$$\|U(\theta) - U(Q_K(\theta))\|_2 \le \sum_{i=1}^{P} \|R_{G_i}(\theta_i) - R_{G_i}(Q_K(\theta_i))\|_2 \le \frac{P\pi}{2K}. \tag{30}$$

Consequently, for any Hermitian observable O with $\|O\|_2 \le 1$, the corresponding circuit output $f(\theta) = \langle 0^n | U^\dagger(\theta) O U(\theta) | 0^n \rangle$ satisfies

$$|f(\theta) - f(Q_K(\theta))| \le \min\left\{2, \frac{P\pi}{K}\right\}. \tag{31}$$

Hence, under the finite-resolution abstraction considered in this work, coarser executable parameter sets permit larger worst-case perturbations when a continuously parametrized circuit is projected onto $\mathcal{A}_K^P$, whereas the bound decreases as $\mathcal{O}(P/K)$ with increasing resolution. This result provides an algorithmic interpretation of K. Importantly, K characterizes the executable angular resolution of our model; its quantitative correspondence to device-specific controller properties, such as DAC resolution, waveform-memory capacity, bandwidth, or control power, depends on the underlying hardware architecture and is not assumed in our analysis.

C Expressivity Analysis of Resolution-Aware Angular Re-uploading

We compare resolution-aware angular re-uploading with conventional angular encoding under finite resolution. For a resolution K, we consider an encoding resolution B compatible with the executable parameter set, i.e., $2^B \mid K$. The resulting circuit input takes values on the discrete domain $\mathbf{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_{2^B}^n$, where $m_j \in \mathbb{Z}_{2^B}$ is the integer representation of the j-th input dimension at resolution B. Because Pauli-generated angular encoding induces periodic dependence on the encoded inputs, the resulting circuit outputs can be naturally characterized through their Fourier spectra. On this discrete domain, the functions $\exp\left(\frac{2\pi i}{2^B}\mathbf{k}^\top \mathbf{m}\right)$ form an orthogonal Fourier basis on $\mathbb{Z}_{2^B}^n$, where $\mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{Z}_{2^B}^n$ denotes the corresponding discrete Fourier frequency index. Any measured circuit output $f : \mathbb{Z}_{2^B}^n \to \mathbb{R}$ can therefore be written as

$$f(\mathbf{m}) = \sum_{\mathbf{k} \in \mathbb{Z}_{2^B}^n} \widehat{f}(\mathbf{k}) \exp\left(\frac{2\pi i}{2^B}\mathbf{k}^\top \mathbf{m}\right), \tag{32}$$

with Fourier coefficients

$$\widehat{f}(\mathbf{k}) = 2^{-Bn} \sum_{\mathbf{m} \in \mathbb{Z}_{2^B}^n} f(\mathbf{m}) \exp\left(-\frac{2\pi i}{2^B}\mathbf{k}^\top \mathbf{m}\right). \tag{33}$$

Conventional Angular Encoding. For conventional angular encoding of resolution B, the j-th input dimension is uploaded once through $R_X\left(\frac{2\pi m_j}{2^B}\right)$. Using the eigendecomposition of the Pauli-X generator,

$$R_X\left(\frac{2\pi m_j}{2^B}\right) = \sum_{s \in \{-1,1\}} e^{-i\pi s m_j / 2^B} P_s, \qquad P_s = \frac{I + sX}{2}, \tag{34}$$

the phases appearing in a measured expectation value take the form

$$e^{i\pi(s-s')m_j/2^B} = e^{2\pi i k_j m_j / 2^B}, \qquad k_j = \frac{s - s'}{2} \in \{-1, 0, 1\}. \tag{35}$$

Thus, when each input dimension is uploaded through a single Pauli rotation, the resulting Fourier support satisfies

$$\operatorname{supp}(\widehat{f}_{\text{ang}}) \subseteq \Omega_{\text{ang}} := \{-1, 0, 1\}^n \pmod{2^B}. \tag{36}$$

Any circuit layers or ansatz pattern applied after this upload can change the coefficients associated with these frequencies, but cannot introduce additional frequencies in $\mathbf{m}$. Conventional angular encoding is therefore restricted to the input spectrum generated by a single aggregate upload.

Resolution-aware Angular Re-uploading. Resolution-aware angular re-uploading instead exposes the input to the circuit through separate resolution levels. Recall that $m_j = \sum_{b=1}^{B} \xi_{j,b} 2^{B-b}$, $\xi_{j,b} \in \{0, 1\}$, and that resolution level b controls the rotation $R_X(\gamma_{j,b}) = R_X\left(\xi_{j,b}\frac{2\pi}{2^b}\right)$. Because $\xi_{j,b}$ is binary,

$$R_X\left(\xi_{j,b}\frac{2\pi}{2^b}\right) = I + \xi_{j,b}\left[R_X\left(\frac{2\pi}{2^b}\right) - I\right], \tag{37}$$

so the input dependence introduced by this upload is determined directly by the Fourier spectrum of the bit function $\xi_{j,b}(m_j)$. As a function on $\mathbb{Z}_{2^B}$, the b-th bit has period 2^{B-b+1}. Its discrete Fourier transform is therefore supported on

$$\Omega_b = \{0\} \cup \left\{(2r+1)2^{b-1} : 0 \le r < 2^{B-b}\right\} \pmod{2^B}. \tag{38}$$

Hence, different resolution levels expose different frequency scales of the same B-bit input. Earlier levels vary more slowly over the discrete input domain, whereas later levels introduce progressively finer-scale variations.

The complete encoder combines these level-wise angular uploads across circuit depth. If all intermediate mixing unitaries are identities, rotations about the same axis combine additively, and the encoder reduces exactly to conventional angular encoding. *The representational difference arises when the*

intermediate mixing unitaries do not commute with the surrounding input-dependent rotations. In this case, the state produced by one resolution level is transformed before subsequent levels are applied, allowing the measured response to contain products of input-dependent terms from different resolution levels and qubits. Because multiplication in the input domain corresponds to convolution in the discrete Fourier domain, an interaction involving resolution levels $b_1, \dots, b_q$ can contribute frequencies contained in the modular sumset

$$\Omega_{b_1} \oplus \cdots \oplus \Omega_{b_q} = \left\{ (k_1 + \cdots + k_q) \bmod 2^B : k_r \in \Omega_{b_r} \right\}. \tag{39}$$

Cross-qubit interactions analogously generate multivariate combinations of these coordinate-wise frequencies. Thus, resolution-aware angular re-uploading operates on exactly the same input as conventional angular encoding, but exposes that information through a richer circuit structure. In particular, conventional angular encoding is recovered exactly as the identity-mixer special case, while noncommuting mixing unitaries can introduce additional cross-level and cross-qubit spectral interactions from the same discretized input. The proposed encoder therefore contains conventional angular encoding as a special case within its representational family while allowing a broader set of spectral interactions to emerge from the same input.

D Derivation of Fourier-Complementary Search Directions

We derive the pairwise Fourier descriptor used to select the additional search directions in Eq. 3. As each trainable parameter controls a Pauli rotation, the multivariate Fourier support of every measured circuit feature is contained in $\{-1, 0, 1\}^P$ and the corresponding circuit feature vector admits the representation

$$\boldsymbol{h}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\omega} \in \Omega} \widehat{\boldsymbol{h}}_{\boldsymbol{\omega}} e^{\mathrm{i}\boldsymbol{\omega}^\top \boldsymbol{\theta}}, \qquad \Omega \subseteq \{-1, 0, 1\}^P. \tag{40}$$

We use this known spectral support to characterize the pairwise parameter interactions exposed by a candidate search direction. For parameters $i < j$, consider a pairwise Fourier component

$$f_{ij}^{\sigma,\tau}(\boldsymbol{\theta}) = \widehat{h}_{\sigma \boldsymbol{e}_i + \tau \boldsymbol{e}_j} e^{\mathrm{i}(\sigma\theta_i + \tau\theta_j)}, \qquad \sigma, \tau \in \{-1, 1\}, \tag{41}$$

where $\boldsymbol{e}_i$ denotes the i-th standard basis vector in $\mathbb{R}^P$. Along a candidate direction $\boldsymbol{c} \in \mathbb{Z}_K^P$, a line step $t \in \mathbb{Z}_K$ changes the two parameters by

$$\delta_i = \Delta_K t c_i, \qquad \delta_j = \Delta_K t c_j. \tag{42}$$

To isolate the response induced specifically by varying the two parameters jointly, we remove their individual contributions through the inclusion–exclusion residual

$$\begin{aligned} r_{ij}^{\sigma,\tau}(t; \boldsymbol{c}) &= f_{ij}^{\sigma,\tau}(\boldsymbol{\theta} + \delta_i \boldsymbol{e}_i + \delta_j \boldsymbol{e}_j) - f_{ij}^{\sigma,\tau}(\boldsymbol{\theta} + \delta_i \boldsymbol{e}_i) \\ &\quad - f_{ij}^{\sigma,\tau}(\boldsymbol{\theta} + \delta_j \boldsymbol{e}_j) + f_{ij}^{\sigma,\tau}(\boldsymbol{\theta}) \\ &= f_{ij}^{\sigma,\tau}(\boldsymbol{\theta}) \left(e^{\mathrm{i}\sigma\Delta_K t c_i} - 1 \right) \left(e^{\mathrm{i}\tau\Delta_K t c_j} - 1 \right). \end{aligned} \tag{43}$$

The first factor contains the unknown circuit-dependent Fourier coefficient, whereas the remaining factors depend only on the candidate direction and the discrete parameter set. We therefore collect these coefficient-independent responses into

$$[A_{ij}(c_i, c_j)]_{t,(\sigma,\tau)} = \left(e^{\mathrm{i}\sigma\Delta_K t c_i} - 1 \right) \left(e^{\mathrm{i}\tau\Delta_K t c_j} - 1 \right), \tag{44}$$

where $t \in \mathbb{Z}_K$ and $(\sigma, \tau) \in \{-1, 1\}^2$. Thus, $A_{ij}(c_i, c_j) \in \mathbb{C}^{K \times 4}$, and each column records how one admissible pairwise Fourier mode responds along the search line defined by (c_i, c_j).

We summarize the response pattern over the complete line through the Gram matrix $A_{ij}^H A_{ij}$ and define the normalized descriptor

$$\boldsymbol{v}(c_i, c_j) = \frac{\mathrm{vec}\big(A_{ij}(c_i, c_j)^H A_{ij}(c_i, c_j)\big)}{\|A_{ij}(c_i, c_j)^H A_{ij}(c_i, c_j)\|_F}, \tag{45}$$

with $\boldsymbol{v}(c_i, c_j) = \mathbf{0}$ whenever $c_i = 0$ or $c_j = 0$. Here, $(\cdot)^H$ denotes conjugate transpose, $\mathrm{vec}(\cdot)$ vectorizes a matrix, and $\|\cdot\|_F$ denotes the Frobenius norm. The Gram matrix captures the correlations

among the admissible pairwise Fourier responses across all line steps, while normalization removes their overall scale. Importantly, the resulting descriptor depends only on how the candidate direction probes the known Fourier support and does not require estimating the unknown Fourier coefficients. Given the previously selected directions $\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(m-1)}$, $\boldsymbol{\Pi}_{ij}^{(m-1)}$ is the orthogonal projector onto the subspace spanned by their nonzero pairwise descriptors,

$$\operatorname{span}\left\{\boldsymbol{v}\left(c_i^{(r)}, c_j^{(r)}\right) : 1 \le r < m,\ \boldsymbol{v}\left(c_i^{(r)}, c_j^{(r)}\right) \neq \mathbf{0}\right\}. \tag{46}$$

The projection residual

$$\left\| \left(\boldsymbol{I} - \boldsymbol{\Pi}_{ij}^{(m-1)} \right) \boldsymbol{v}(c_i, c_j) \right\|_2^2 \tag{47}$$

measures the component of the candidate pairwise response that is not represented by the previously selected search directions. A larger residual therefore indicates that the candidate direction probes pairwise Fourier-response structure that is less covered by the current multi-line search. This motivates its use as the Fourier-complementarity term in Eq. 3.

E Shift Compatibility of the Executable Parameter Set

SISCO reconstructs each coordinate-wise circuit-output profile using circuit evaluations in which the corresponding trainable parameter is shifted by $\pm\pi/2$. For these shifted parameter configurations to remain executable, the discrete parameter set

$$\mathcal{A}_K = \{\ell \Delta_K : \ell \in \mathbb{Z}_K\}, \qquad \Delta_K = \frac{2\pi}{K}, \tag{48}$$

must be closed under required $\pm\pi/2$ parameter shifts. The following result characterizes this condition exactly.

Lemma 1 (Closure under parameter shifts)**.** *For every $\theta \in \mathcal{A}_K$, both $\theta \pm \pi/2$ belong to $\mathcal{A}_K$ if and only if K is a multiple of 4.*

Proof. Any $\theta \in \mathcal{A}_K$ can be written as $\theta = \ell\Delta_K$ for some $\ell \in \mathbb{Z}_K$. Since $\frac{\pi}{2} = \frac{K}{4}\Delta_K$, shifting θ by $\pm\pi/2$ corresponds to shifting its discrete parameter index by $\pm K/4$. If K is a multiple of 4, then $K/4$ is an integer, and therefore

$$\theta \pm \frac{\pi}{2} = \left(\ell \pm \frac{K}{4}\right)\Delta_K \in \mathcal{A}_K. \tag{49}$$

Conversely, suppose that $\mathcal{A}_K$ is closed under $\pm\pi/2$ shifts. Applying this condition at $\theta = 0$ requires $\pi/2 \in \mathcal{A}_K$. Hence, $\pi/2 = q\Delta_K$ for some integer q, which implies $q = K/4$. Therefore, $K/4$ must be an integer, and K must be a multiple of 4. *Thus, K being a multiple of 4 is both necessary and sufficient for the $\pm\pi/2$ parameter-shift evaluations used by SISCO and by the standard parameter-shift rule for gradient estimation to remain within the same executable parameter set.* □

F Effect of Fourier Complementarity in SISCO

We conduct a controlled ablation to examine whether Fourier complementarity improves the selection of additional search directions beyond what can be achieved by a purely profile-informed multi-line search. Recall that SISCO selects its first direction from the coordinate-wise loss profiles and uses the Fourier-complementarity criterion to diversify the remaining $M - 1$ directions. To isolate the role of this criterion from the benefit of evaluating multiple search lines itself, we construct a *Q-only* variant that retains the same multi-line search procedure but selects all additional directions using profile-predicted utility alone, without the Fourier-complementarity term.

We evaluate both methods on the same controlled circuit-optimization benchmark used in the main experiments. The two variants share identical target circuits, initializations, coordinate-profile reconstructions, numbers of search directions, line-search budgets, and update procedures; they differ only in how the additional directions are selected. SISCO uses the full objective in Eq. 3,

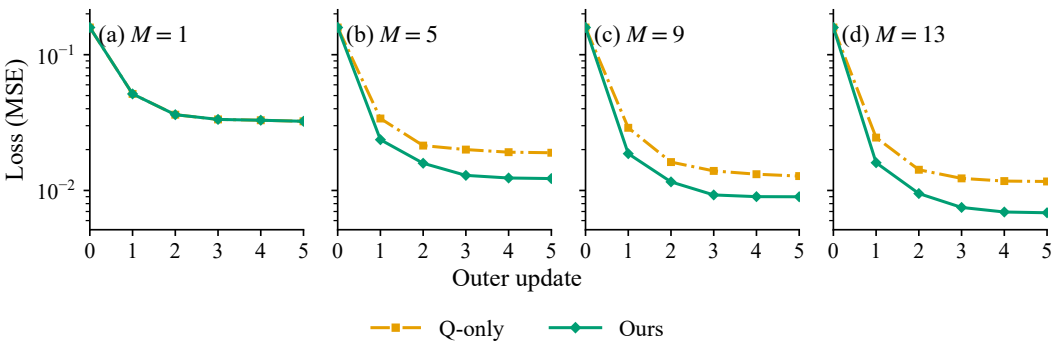

Figure 7: Effect of Fourier complementarity on discrete circuit optimization. The curves report averaged MSE over $T = 500$ target problems. Q-only removes Fourier complementarity from our SISCO on direction selection.

combining profile-predicted utility with Fourier complementarity, whereas *Q-only* retains only the profile-based utility. Consequently, the comparison directly isolates whether explicitly favoring Fourier-complementary directions provides additional value beyond profile-informed multi-line search.

For $M = 1$, both methods contain only the same profile-informed first direction and therefore coincide. For $M > 1$, they evaluate the same number of additional search lines under the same circuit-evaluation budget, while differing only in the information used to select those lines. As shown in Fig. 7, SISCO consistently reaches lower loss than *Q-only* for $M > 1$, with the advantage becoming more pronounced as additional directions are introduced. *These results provide strict evidence that explicitly accounting for complementary pairwise Fourier-response structure enables more effective selection of additional search directions and, consequently, improved discrete circuit optimization.*

G Hardware execution on physical quantum processing units (QPUs)

We further evaluate the DPQC model on a physical QPU to assess whether its predictive behavior is retained beyond ideal circuit simulation. This experiment is not intended to emulate or measure a restricted native resolution of the IBM device, since user-level pulse control is not available on the current IBM Quantum platform. Rather, it evaluates whether a DPQC learned within the prescribed discrete parameter space remains executable and retains its predictive behavior after hardware compilation and physical QPU execution. As illustrated in Fig. 8, each logical circuit is first compiled into an executable circuit through Instruction Set Architecture (ISA) transpilation, including hardware-aware qubit layout, connectivity-constrained routing, and translation into the device-native gate set. The compiled circuits are then executed on IBM Kingston, and measurement outcomes from 512 shots are converted into local Pauli-Z expectation estimates used by the classifier.

Tab. 3 compares QPU execution with the corresponding ideal-simulator results. For Imagenette, CIFAR-10, and STL-10, we choose to evaluate a class-balanced subset of 600 test images per dataset; this number is chosen in accordance with the maximum QPU access budget we have. While this limits the scale of the hardware evaluation, it provides a controlled and class-balanced comparison across multiple image benchmarks and enables direct comparison with the corresponding ideal-simulator predictions. The hardware-executed DPQCs retain strong predictive performance overall. On CIFAR-10 and STL-10, QPU accuracy reaches 73.83% and 84.50%, respectively, remaining within 0.7 percentage points of the corresponding ideal-simulator results. The two-spiral task similarly retains 96.50% accuracy on hardware, compared with 99.50% under ideal simulation. Imagenette exhibits a larger simulator-to-hardware gap, decreasing from 93.67% to 86.00%, while still maintaining substantial classification performance under physical QPU execution. Overall, these results show

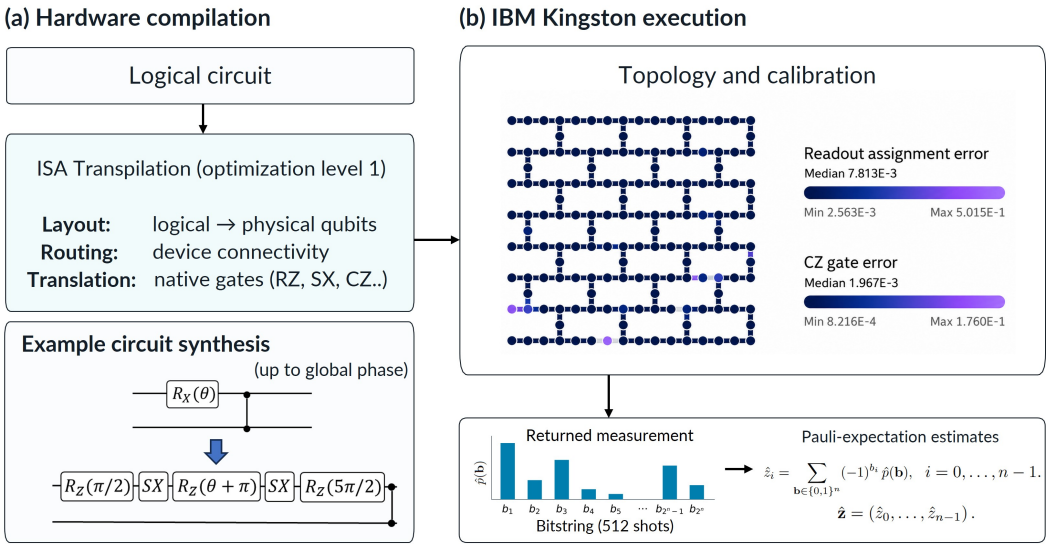

Figure 8: Compilation and execution on IBM Kingston. (a) Logical circuits are transpiled to the hardware instruction set. (b) The compiled circuits are executed on IBM Kingston with the device topology and calibration information given; measurement bitstrings from 512 shots are converted into local Pauli-Z expectation estimates.

that the learned DPQC models retain much of their predictive performance after compilation and execution on real quantum hardware, despite the constrained QPU evaluation budget.

Table 3: Classification accuracy under ideal simulation and IBM Kingston QPU execution.

Dataset	Instances	Ideal simulator	QPU (IBM Kingston)
Two-spiral	-	99.50	96.50
Imagenette	600	93.67	86.00
CIFAR-10	600	74.50	73.83
STL-10	600	85.17	84.50

H Limitations and Future Work

Optimizing a DPQC directly over $\mathcal{A}_K^P$ is inherently combinatorial, since jointly selecting updates for multiple circuit parameters requires searching over a discrete configuration space whose size grows exponentially with the number of parameters. The resulting joint discrete optimization problem is combinatorial and can be NP-hard under general formulations. In this work, building on existing profile-informed line-search methods for PQCs, our SISCO leverages the circuit's Fourier structure to model pairwise parameter interactions and select a limited but informative set of complementary search directions, providing a controllable trade-off between interaction-aware exploration and quantum-evaluation cost. Developing efficient methods for such coupled discrete optimization therefore remains an important direction for future work. In particular, reformulating DPQC updates as structured combinatorial optimization problems, such as Ising-type formulations, may provide a promising avenue for leveraging specialized hardware, including Ising machines, to explore interacting parameter configurations more efficiently. Beyond optimization, extending the discrete DPQC formulation to other quantum machine-learning applications is also promising. For example, quantum implicit neural representations provide a natural setting for investigating finite-resolution circuit parameterization in continuous signal representation tasks.

Algorithm 1 SISCO Training of DPQCs

Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_r, y_r)\}_{r=1}^{N}$; number of epochs N_{epoch}; resolution K; executable parameter set $\mathcal{A}_K$; number of search directions M; classical parameters $\boldsymbol{\phi}$; discrete parameter indices $\boldsymbol{z} \in \mathbb{Z}_K^P$.
for epoch $= 1$ **to** N_{epoch} **do**
  Set $\boldsymbol{\theta} = \Delta_K \boldsymbol{z} \in \mathcal{A}_K^P$ and execute the reference circuit to obtain the current circuit output $\boldsymbol{h}_0$ and loss $\mathcal{L}(\boldsymbol{\theta})$.
  for $i = 1$ **to** P **do**
    Form the shift-compatible indices $z_i^{\pm} = z_i \pm K/4$, and execute the corresponding circuits to obtain $\boldsymbol{h}_i^{+}$ and $\boldsymbol{h}_i^{-}$.
    Reconstruct the coordinate-wise circuit-output profile $\boldsymbol{h}_i(\alpha)$ and loss profile $\ell_i(\alpha)$ for all $\alpha \in \mathcal{A}_K$ using Eq. 1.
    Compute the corresponding loss reductions $g_i(\alpha) = \mathcal{L}(\boldsymbol{\theta}) - \ell_i(\alpha),\ \forall\, \alpha \in \mathcal{A}_K$.
  end for
  Construct the profile-informed first direction by $c_i^{(1)} \in \arg\max_{c \in \mathbb{Z}_K} g_i(\theta_i + \Delta_K c)\,,\ i = 1, \ldots, P$, and set $\boldsymbol{c}^{(1)} = \left(c_1^{(1)}, \ldots, c_P^{(1)}\right)$.
  Sequentially select $\boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(M)}$ using Eq. 3.
  Evaluate the unique executable configurations $\boldsymbol{\theta}^{(m,t)} = \Delta_K \left(\boldsymbol{z} + t\boldsymbol{c}^{(m)}\right),\ t \in \mathbb{Z}_K,\ m = 1, \ldots, M$, and update $\boldsymbol{z}$ to the index vector corresponding to the configuration with the lowest observed loss.
  Update the classical parameters $\boldsymbol{\phi}$ by gradient-based optimization with the circuit parameters fixed.
end for
Output: Optimized classical parameters $\boldsymbol{\phi}^{\star}$ and executable circuit parameters $\boldsymbol{\theta}^{\star} = \Delta_K \boldsymbol{z}^{\star} \in \mathcal{A}_K^P$.

I Complete Training Procedure

The algorithm 1 summarizes the complete training procedure.

J Circuit Evaluation Cost during Training

We provide a detailed accounting of the circuit-evaluation cost of one SISCO update. For a circuit with P trainable parameters, coordinate-wise spectral reconstruction requires two shift-compatible evaluations $\boldsymbol{h}_i^{+}$ and $\boldsymbol{h}_i^{-}$ for each parameter θ_i. This results in

$$N_{\text{profile}} = 2P \tag{50}$$

circuit evaluations. Importantly, these evaluations are sufficient to reconstruct the complete coordinate-wise output and loss profiles over all $\alpha \in \mathcal{A}_K$ through Eq. 1; no additional circuit executions are required to evaluate the remaining admissible values along each coordinate.

After constructing the M search directions, SISCO evaluates the corresponding discrete search lines. Each line contains at most K executable configurations, one of which is the current configuration that has already been evaluated. Therefore, each selected direction requires at most $K - 1$ additional circuit evaluations, giving

$$N_{\text{line}} \leq M(K-1). \tag{51}$$

Combining the profile-reconstruction and line-search costs yields

$$N_{\text{SISCO}} \leq 2P + 1 + M(K-1). \tag{52}$$

For comparison, a standard parameter-shift gradient evaluation requires $2P$ shifted circuit evaluations, or $2P + 1$ evaluations when including the reference circuit used to evaluate the current objective. *SISCO therefore uses the same set of reference and shifted evaluations to reconstruct the coordinate-wise profiles, and incurs additional quantum evaluations only for the selected search lines.* Rather than exhaustively exploring the K^P configurations in $\mathcal{A}_K^P$, the method concentrates this additional evaluation budget on a limited number M of profile-informed and Fourier-complementary directions.

K IMAGE CLASSIFICATION UNDER DIFFERENT ANSATZ PATTERNS

To assess whether the observed behavior of DPQCs depends on a particular variational circuit architecture, we repeat the multi-class image-classification experiments using the strongly entangling and Hadamard-staggered ansatzes while keeping the remaining experimental setup unchanged. The results are reported in Tables 4 and 5. Consistent with the main-text results in Tab. 2, the same qualitative trends persist across both tested alternative ansatz patterns. Resolution-aware angular re-uploading generally improves over conventional angular encoding across datasets and resolutions. Direct discrete optimization remains particularly effective at coarse resolutions. Although the relative performance of individual optimizers varies across architectures and datasets, these overall trends remain stable. This consistency across substantially different variational ansatzes indicates that the observed benefits of the proposed DPQC framework are not specific to the circuit architecture used in the main experiments.

Table 4: Top-1 image-classification accuracy (%) with the hardware efficient ansatz.

Dataset	CPQC	Encoding	Training protocol	$K = 8$	$K = 16$	$K = 32$
Imagenette	93.82 ± 0.97	Conventional	Nearest STE	81.84 ± 12.64	$\mathbf{91.54 \pm 1.60}$	92.59 ± 1.72
			Stochastic STE	45.63 ± 26.23	89.15 ± 2.74	91.19 ± 1.15
			SISCO	$\mathbf{88.52 \pm 1.32}$	91.25 ± 1.12	$\mathbf{92.99 \pm 0.77}$
		Resolution-aware	Nearest STE	84.60 ± 12.40	92.62 ± 0.71	94.19 ± 0.43
			Stochastic STE	55.00 ± 27.69	93.08 ± 1.30	$\mathbf{94.42 \pm 0.56}$
			SISCO	$\mathbf{90.92 \pm 0.46}$	$\mathbf{93.61 \pm 0.58}$	93.64 ± 0.48
CIFAR-10	78.56 ± 0.89	Conventional	Nearest STE	67.77 ± 1.57	$\mathbf{72.52 \pm 3.56}$	71.91 ± 0.62
			Stochastic STE	53.62 ± 14.41	69.81 ± 3.88	75.35 ± 1.71
			SISCO	$\mathbf{72.01 \pm 1.05}$	71.16 ± 0.81	$\mathbf{75.87 \pm 0.98}$
		Resolution-aware	Nearest STE	70.48 ± 1.02	72.85 ± 0.86	73.53 ± 0.36
			Stochastic STE	60.08 ± 11.51	71.89 ± 4.39	75.17 ± 1.37
			SISCO	$\mathbf{74.94 \pm 1.99}$	$\mathbf{73.98 \pm 0.72}$	$\mathbf{77.14 \pm 0.53}$
STL-10	85.45 ± 1.96	Conventional	Nearest STE	70.23 ± 16.15	$\mathbf{82.09 \pm 2.65}$	83.21 ± 3.81
			Stochastic STE	31.09 ± 9.97	73.27 ± 6.87	82.19 ± 6.34
			SISCO	$\mathbf{78.14 \pm 3.45}$	81.50 ± 3.38	$\mathbf{84.11 \pm 1.97}$
		Resolution-aware	Nearest STE	77.90 ± 14.22	84.98 ± 0.79	85.27 ± 1.40
			Stochastic STE	39.35 ± 17.41	79.09 ± 4.94	84.86 ± 4.42
			SISCO	$\mathbf{81.03 \pm 2.74}$	$\mathbf{86.31 \pm 1.41}$	$\mathbf{86.81 \pm 1.65}$

Bold: best training protocol within each encoding using the full seed set;
Shading: better encoding under the same training protocol using the full matched seed set.

Table 5: Top-1 image-classification accuracy (%) with the Hadamard-staggered ansatz.

Dataset	CPQC	Encoding	Training protocol	$K = 8$	$K = 16$	$K = 32$
Imagenette	92.80 ± 1.26	Conventional	Nearest STE	86.96 ± 4.73	89.88 ± 0.95	89.89 ± 0.79
			Stochastic STE	79.46 ± 5.88	88.92 ± 6.48	90.99 ± 1.15
			SISCO	$\mathbf{87.58 \pm 1.88}$	$\mathbf{90.02 \pm 1.96}$	$\mathbf{91.61 \pm 0.49}$
		Resolution-aware	Nearest STE	89.53 ± 4.42	88.18 ± 4.34	92.50 ± 1.39
			Stochastic STE	81.34 ± 23.31	$\mathbf{92.60 \pm 0.38}$	92.99 ± 0.61
			SISCO	$\mathbf{90.71 \pm 1.04}$	92.35 ± 1.99	$\mathbf{93.14 \pm 0.97}$
CIFAR-10	77.81 ± 0.71	Conventional	Nearest STE	68.47 ± 3.23	71.72 ± 3.32	$\mathbf{73.09 \pm 1.75}$
			Stochastic STE	$\mathbf{69.95 \pm 2.73}$	69.14 ± 3.39	72.27 ± 4.62
			SISCO	68.27 ± 1.54	$\mathbf{72.93 \pm 3.01}$	72.31 ± 0.44
		Resolution-aware	Nearest STE	71.45 ± 2.21	73.24 ± 2.77	75.40 ± 0.55
			Stochastic STE	70.51 ± 2.86	72.50 ± 2.17	73.56 ± 0.69
			SISCO	$\mathbf{71.72 \pm 1.12}$	$\mathbf{73.76 \pm 1.30}$	$\mathbf{75.41 \pm 1.00}$
STL-10	82.97 ± 2.99	Conventional	Nearest STE	71.54 ± 5.78	79.24 ± 4.51	80.17 ± 3.75
			Stochastic STE	70.83 ± 7.29	76.43 ± 7.74	80.60 ± 6.31
			SISCO	$\mathbf{75.90 \pm 2.00}$	$\mathbf{83.11 \pm 2.27}$	$\mathbf{81.73 \pm 5.24}$
		Resolution-aware	Nearest STE	76.62 ± 6.30	82.84 ± 2.54	84.45 ± 1.72
			Stochastic STE	80.19 ± 6.52	79.24 ± 8.91	82.88 ± 5.02
			SISCO	$\mathbf{80.87 \pm 3.54}$	$\mathbf{85.23 \pm 1.66}$	$\mathbf{85.10 \pm 2.34}$

Bold: best training protocol within each encoding using the full seed set;
Shading: better encoding under the same training protocol using the full matched seed set.

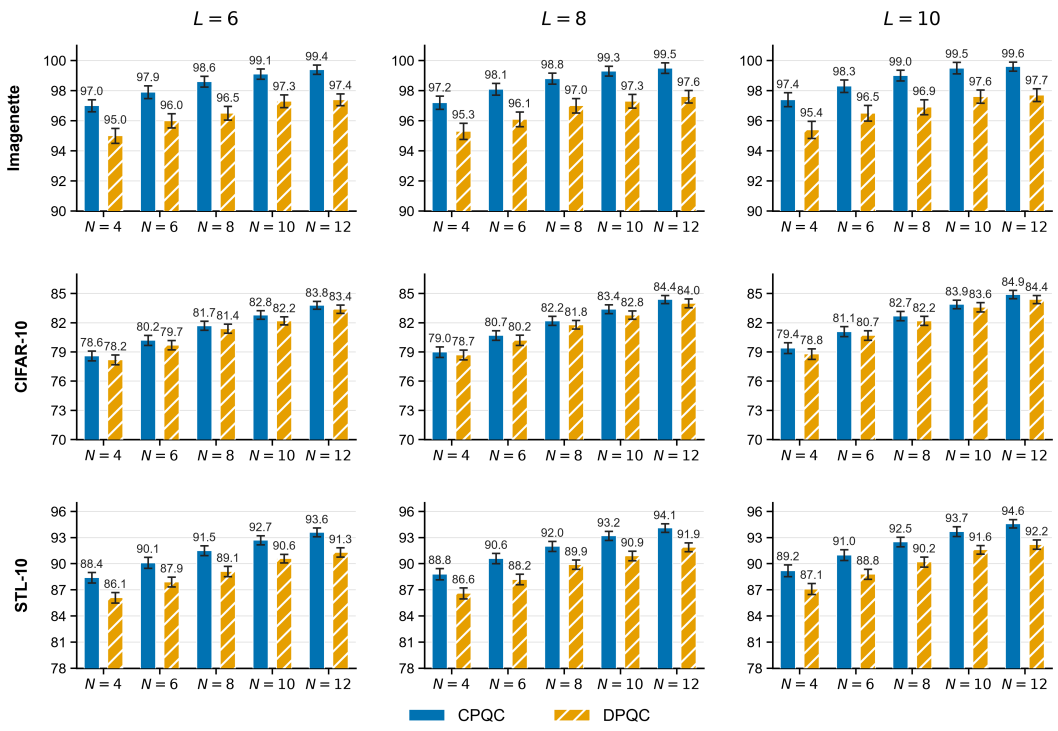

Figure 9: **Scaling behavior of CPQCs and DPQCs with qubit count and circuit depth.** Top-1 image-classification accuracy is reported for Imagenette (top row), CIFAR-10 (middle row), and STL-10 (bottom row) at circuit depths $L \in \{6, 8, 10\}$ and qubit counts $n \in \{4, 6, 8, 10, 12\}$.

L Scaling Behavior across Circuit Scales

To examine whether finite-resolution parameterization remains effective as circuit scale increases, we evaluate DPQCs across a range of circuit widths and depths, using the corresponding CPQCs as continuously parametrized references. Fig. 9 reports Top-1 accuracy on Imagenette, CIFAR-10, and STL-10 for $n \in \{4, 6, 8, 10, 12\}$ qubits and $L \in \{6, 8, 10\}$ variational layers in the ansatz. Across all three datasets, the DPQC performance closely matches the corresponding CPQC throughout the evaluated circuit scales, with no systematic increase in the performance gap as either circuit width or depth grows. This result suggests that the predictive performance retained under finite-resolution remains stable as the circuit architecture scales, indicating that the effect of discrete parameterization does not become more pronounced for the larger circuits considered here. This complements the analysis in the main text.