

Establishing Baselines for Photonic Quantum Machine Learning: Insights from an Open, Collaborative Initiative

Cassandre Notton^{*1}, Vassilis Apostolou², Agathe Senellart³, Anthony Walsh², Daphne Wang², Yichen Xie¹², Songqinghao Yang⁷, Ilyass Mejdoub^{5,6}, Oussama Zouhry⁴, Kuan-Cheng Chen^{9,10}, Chen-Yu Liu¹¹, Ankit Sharma¹³, Edara Yaswanth Balaji¹⁴, Soham Prithviraj Pawar¹⁵, Ludovic Le Frioux⁸, Valentin Macheret⁸, Antoine Radet⁸, Valentin Deumier²⁴, Ashesh Kumar Gupta²⁰, Gabriele Intocchia²², Dimitri Jordan Kenne¹⁹, Chiara Marullo¹⁸, Giovanni Massafra¹⁸, Nicolas Reinaldet²¹, Vincenzo Schiano Di Cola²³, Danylo Kolesnyk^{16, 17}, Yelyzaveta Vodovozova¹⁶, Rawad Mezher², Pierre-Emmanuel Emeriau², Alexia Salavrakos², and Jean Senellart²

¹Quandela Quantique Inc., Montréal, Canada

²Quandela SAS, Massy, France

³Universite Paris Cité, INRIA, Inserm, HeKA, Paris, France

⁴Ecole Polytechnique, Palaiseau, France

⁵Télécom Paris, Palaiseau, France

⁶ENS Paris Saclay, Gif-sur-Yvette, France

⁷Cavendish Lab., Department of Physics, University of Cambridge, Cambridge, UK

⁸Scaleway, Paris, France

⁹QuEST, Imperial College, London, United Kingdom

¹⁰I-X Centre for AI in Science, Imperial College London, London, United Kingdom

¹¹National Taiwan University, Taipei, Taiwan

¹²La Salle College, Hong Kong

¹³University of Delhi, New Delhi, India

¹⁴Indian Institute of Technology Hyderabad, Telangana, India

¹⁵International Institute of Information Technology, Bangalore, India

¹⁶Technical University of Munich, Garching, Germany

¹⁷Ludwig Maximilian University, Munich, Germany

¹⁸ICAR-CNR, Naples, Italy

¹⁹Department of Mathematics and Physics, University of Campania, Caserta, Italy

²⁰Dipartimento Interuniversitario di Fisica, Università degli Studi di Bari Aldo Moro, Bari, Italy

²¹Invent Vision, Brazil

²²Department of Mathematics and Applications, University of Naples Federico II, Naples, Italy

²³Quantum2Pi Srl, Naples, Italy

²⁴Ecole des Mines de Paris - PSL, Paris, France

October 31, 2025

Abstract

The Perceval Challenge is an open, reproducible benchmark designed to assess the potential of photonic quantum computing for machine learning. Focusing on a reduced and hardware-feasible version of the MNIST digit classification task or near-term photonic processors, it offers a concrete framework to evaluate how photonic quantum circuits learn and generalize from limited data. Conducted over more than three months, the challenge attracted 64 teams worldwide in its first phase. After an initial selection, 11 finalist teams were granted access to GPU resources for large-scale simulation and photonic hardware execution through cloud service. The results establish the first unified baseline of photonic machine-learning performance, revealing complementary strengths between variational, hardware-native, and hybrid approaches. This challenge also underscores the importance of open, reproducible experimentation and interdisciplinary collaboration, highlighting how shared benchmarks can accelerate progress in quantum-enhanced learning. All implementations are publicly available in a single shared repository¹, supporting transparent benchmarking and cumulative research. Beyond this specific task, the Perceval Challenge illustrates how systematic, collaborative experimentation can map the current landscape of photonic quantum machine learning and pave the way toward hybrid, quantum-augmented AI workflows.

^{*}cassandre.notton@quandela.com

¹<https://github.com/Quandela/HybridAIQuantum-Challenge>

1 Introduction

Early research on Quantum Machine Learning (QML) centered on algorithms built from well-known quantum subroutines such as quantum phase estimation, aiming to demonstrate provable speedups over classical methods [1, 2]. These approaches, however, assumed access to fault-tolerant quantum computers, which remain under development. With the advent of the NISQ era [3], and inspired by the widespread success of neural networks in classical machine learning, the QML community began shifting its interest towards more practical, hardware-compatible strategies, such as variational quantum algorithms [4] and quantum kernel methods [5, 6].

As the field matured, a countercurrent emerged [7, 8]. Rigorous benchmarking studies revealed that many widely cited QML models, even after extensive hyperparameter tuning, did not outperform classical baselines on standard tasks [9, 10]. Subsequent efforts have extended this line of work to other domains, such as time-series prediction [11]. These findings recalibrated expectations and underscored the need for systematic evaluation methodologies, especially as models grow too large to be simulated classically. In precisely these regimes, where claims of quantum advantage are most compelling, rigorous benchmarking becomes the most challenging.

Alongside this methodological shift, advances in experimental platforms—including demonstrations of algorithms on superconducting devices [6, 12], neutral atoms [13, 14], trapped ions [15, 16], and photonic platforms [17, 18]—have sparked interest in tailoring algorithms to hardware-specific constraints. On the one hand, noise, limited scale, and compilation challenges make hardware-aware design essential, with error mitigation strategies [19, 20] and hybrid quantum–classical workflows emerging as standard tools. On the other hand, working directly with experimental devices also invites algorithm design around native quantum primitives—operations and encodings that arise naturally in a given platform—rather than imposing abstractions better suited to idealized fault-tolerant models. This dual motivation has given rise to a growing body of hardware-adapted QML approaches.

Photonics provides a particularly compelling testbed. Quandela recently introduced a linear-optical quantum processor [21] and its companion simulation framework, *Perceval* [22], enabling both cloud-based access and algorithm development. Beyond qubit encodings, linear optics can serve as a standalone computational paradigm, motivating the design of “photon-native” algorithms [23–30]. This creates an opportunity to explore QML models that are not only hardware-compatible, but hardware-driven.

In classical machine learning, progress has often been accelerated by open competitions and shared benchmarks, such as the Netflix Prize [31], as well as challenges hosted on the platforms like Kaggle [32], which provide datasets and leaderboards for broad community participation. Motivated by this tradition, we organized the **Perceval Quest** [33]: a six-month-long hackathon dedicated to exploring QML within the framework of linear optics. We believe such events encourage innovation, broaden participation beyond the quantum community, and promote interdisciplinary collaboration with the wider AI ecosystem. Similar initiatives, like the 2024 Airbus–BMW Quantum Computing Challenge [34] demonstrate how benchmarking competitions can accelerate innovation across quantum technologies.

To ground our challenge in a familiar benchmark, we chose the iconic MNIST dataset [35]. MNIST has long served as a proving ground for computer vision models and provides a well-understood reference point for comparing quantum and classical approaches. Participants in the Perceval Quest were asked: *what kinds of models can be built using linear optics and the Perceval framework, and how do they compare to classical solutions—not only in terms of accuracy, but also in number of parameters and convergence speed?*

The main contributions of this work are as follows:

- Provide a **systematic review** of the diverse approaches explored throughout the *Perceval Quest*, organizing them into coherent methodological categories and identifying common design patterns: (i) photonic kernels, neural networks, and convolutional models, where the interferometer functions as an end-to-end feature extractor; (ii) enhanced CNNs and hybrid feature extractors, where it operates as a quantum annotator; and (iii) transfer learning and self-supervised learning (SSL) paradigms, where it supports model fine-tuning. We also propose a novel method that exploits the computational properties of photonic quantum processors through permanent-based computation.
- Highlight the **migration of methods** from non-photonic QML paradigms to photonic implementations, noting that several of these ideas were developed further and submitted as independent articles [36, 37];
- Place a strong emphasis on reproducibility, consolidating all implementations into a unified open-source repository [33] to ensure transparency and facilitate further research;

- Establish benchmarking practices by comparing photonic models against one another and, whenever possible, against classical baselines—not only in terms of accuracy, but also model size, parameter efficiency, and convergence speed.

Although no clear heuristic quantum advantage was observed at this stage, the challenge provides a systematic foundation through reproducible code, benchmarking protocols, and diverse algorithmic strategies, paving the way for more decisive tests as photonic hardware matures.

2 Problem definition and classical solutions

As outlined above, the participants were asked to design algorithms that integrate machine learning with linear-optical quantum computing to classify the MNIST dataset, making use of the Perceval framework to simulate the quantum components and interface with available hardware. The MNIST dataset [35] consists of black-and-white images of handwritten digits, normalized to fit within a 28×28 pixel box. The dataset contains 70,000 images in total: 60,000 for training and 10,000 for testing.

MNIST has played a crucial role in the development and validation of computer vision models, from traditional machine learning techniques to modern deep neural networks. State-of-the-art convolutional neural networks have achieved near-perfect performance, with test errors as low as 0.6%, while simpler architectures, such as multilayer perceptrons with ReLU activations, report errors around 1.1% [38].

This historical role makes MNIST a natural candidate for benchmarking QML approaches. However, several important caveats must be considered. One of the main challenges is the relatively high dimensionality of the dataset—each image has 784 features—which exceeds the effective dimensionality accessible to most current quantum hardware and simulators (whether measured in qubits, modes, or feasible circuit depth). As noted in [9], most QML models using MNIST as a benchmark therefore apply classical preprocessing techniques, such as PCA, to reduce the dimensionality of the dataset. Another common simplification is to restrict the dataset to only two out of the ten digits [39], thereby reducing the task to binary classification, which is substantially less complex than the full multi-class problem. Such simplifications make it difficult to draw meaningful conclusions about the specific role of the quantum component or the intrinsic value of QML models. For this reason, in the present challenge we opted for a reduced version of the dataset that retains all ten classes and the full image resolution, while limiting the number of samples to make the task more challenging for classical models. Participants could still apply classical preprocessing techniques but were required to provide detailed comparisons with classical machine learning models trained and evaluated on the same dataset.

Moreover, participants were encouraged to conduct ablation studies to systematically assess the impact of the quantum components within their approaches. To ensure fair evaluation, a classical benchmark model was provided as a reference point. This benchmark, based on a convolutional neural network (CNN), achieved a test error of approximately 3%—a reasonable baseline given the reduced size of the challenge dataset. Together, these requirements emphasized reproducibility and rigorous benchmarking, enabling meaningful comparisons between photonic and classical models.

To push the boundaries of current technological capabilities, participants were provided with access to high-performance computing resources. These included powerful graphics processing units (GPUs) capable of handling large-scale simulations via Scaleway’s Quantum-as-a-Service platform [40], as well as access to Quandela’s Quantum Processing Units (QPUs) through its cloud platform [41]. This infrastructure ensured that participants could explore models beyond the limits of standard hardware, thereby enabling stress test of both algorithms and simulation frameworks.

3 Related work

MNIST as a benchmark. The MNIST dataset has long been one of the most widely used testbeds in machine learning research due to its simplicity, accessibility, and well-understood properties [35, 42]. It provides a balanced classification task that is neither trivial nor excessively complex, making it an attractive reference point for methodological comparisons across decades of work. A further advantage is that MNIST lends itself to *bidirectional complexity shaping*. Reducing the dataset can make classification easier (e.g., using well-separated binary pairs), but it can also create harder problems when the chosen digits are visually confusable (such as 3/5 or 4/9). Similarly, dimensionality reduction with PCA may improve efficiency at moderate levels, yet aggressive compression can remove discriminative structure and degrade accuracy, while expanded encodings push the problem into higher-dimensional

regimes. Community variants extend this idea by explicitly increasing task difficulty through rotations, affine distortions, or clutter (Rotated/Cluttered MNIST, affMNIST, MNIST-C), and through drop-in replacements that are empirically more challenging (Fashion-MNIST, EMNIST, Kuzushiji-MNIST). These knobs make MNIST a flexible benchmark that can be tuned both below and beyond its original complexity, allowing researchers to systematically probe learning models under controlled variations [43–50]. This adaptability has allowed MNIST to play a central role in systematically probing learning architectures in a controlled, progressive manner, culminating in landmark results such as the multi-column deep neural networks of Cireřan et al. [51], which were among the first to achieve near-human performance.

Landscape of QML+MNIST. To contextualize our study, we assembled a keyword-filtered snapshot of arXiv papers that mention both quantum machine learning and MNIST in the title or abstract ($n = 244$, 2015–2025). Figure 1 shows the temporal trend, broken down by modality: gate-based approaches dominate, with smaller but sustained activity in annealing, photonic, and quantum-inspired models. Table 1 quantifies this distribution and adds further indicators such as binary vs. multiclass usage, code availability, and hardware execution.

System dimensionality. Table 1 also reports the range and variability of input dimensionalities used in QML+MNIST studies. While raw MNIST has 784 features, most works operate on substantially reduced embeddings, with mean dimensionalities of only 30–100 across modalities. At the same time, some studies expand inputs into the thousand-dimensional regime (up to 3530 for gate-based and 1550 for photonic). The large variances (on the order of 10^4 – 10^5 for gate-based and photonic) highlight the heterogeneity of preprocessing choices. This confirms that dimensionality reduction and encoding strategies are major uncontrolled variables in the literature, complicating fair comparison. Our challenge therefore standardizes the full 784-dimensional task while tracking parameter efficiency.

Photonic contributions. Photonic approaches remain underrepresented in our snapshot (14 papers, $\sim 6\%$), though they place stronger emphasis on multiclass classification (12/14) and exhibit above-average code availability (43%). Most of these works are simulator-based, with only three reporting hardware-only results. Notably, a recent parallel effort by Sakurai *et al.* (2025) introduces a boson-sampling-powered quantum optical reservoir computing model and applies it to MNIST, signaling growing interest in more sophisticated photonic-native methodologies [52]. This trajectory underscores both the opportunity and rising need for standardized photonic benchmarks.

Positioning of the present work. Existing studies mostly establish feasibility under reduced datasets and without consistent baselines or ablations. By contrast, our challenge is explicitly photonic-native, uses the full ten-class MNIST task (rather than downsampled or binary subsets), and evaluates not only accuracy but also parameter efficiency, FLOPs, and convergence speed. With all implementations released in a unified repository, we aim to provide a reproducible benchmark suite that directly addresses the gaps evident in Table 1 and Figure 1.

The surveyed literature demonstrates both the breadth of modalities and the heterogeneity of experimental setups (dimensionality, task type, hardware vs. simulator). However, it also reveals a striking lack of consistency: most works rely on strong dataset simplifications, custom encodings, or narrow binary tasks, making cross-comparison difficult. This aligns with the broader concerns articulated by Schuld and Killoran [8]: the field often frames itself in terms of “quantum advantage” over classical ML, yet current tools, datasets, and hardware only support highly restricted experiments. They advocate shifting the emphasis away from outperforming classical ML toward model building, theoretical frameworks, and software infrastructure that prepare QML for realistic scales. Our present work follows this spirit. Rather than seeking immediate advantage, we aim to establish reproducible, photonic-native benchmarks on a full multiclass MNIST task—a step toward standardized evaluation practices that can ground future debates on expressivity, efficiency, and scalability.

4 Preliminaries: linear optical quantum computing

In quantum linear optics, information is encoded in the Fock states of photons distributed among spatial or temporal modes. For a system of n photons in m modes, the input state can be written as $|\vec{n}_{\text{in}}\rangle = |n_1^{\text{in}}, n_2^{\text{in}}, \dots, n_m^{\text{in}}\rangle$ where n_i^{in} denotes the number of photons in mode i and $\sum_i n_i^{\text{in}} = n$. The input state is propagated through an interferometer and then measured by photon number detectors (or threshold detectors). This yields a vector $\vec{n}_{\text{out}} = (n_1^{\text{out}}, n_2^{\text{out}}, \dots, n_m^{\text{out}})$, which describes the arrangement of n photons in m modes, and where $\sum_i n_i^{\text{out}} = n$ in the absence of loss.

Transformations between input and output Fock states are governed by the evolution of the creation

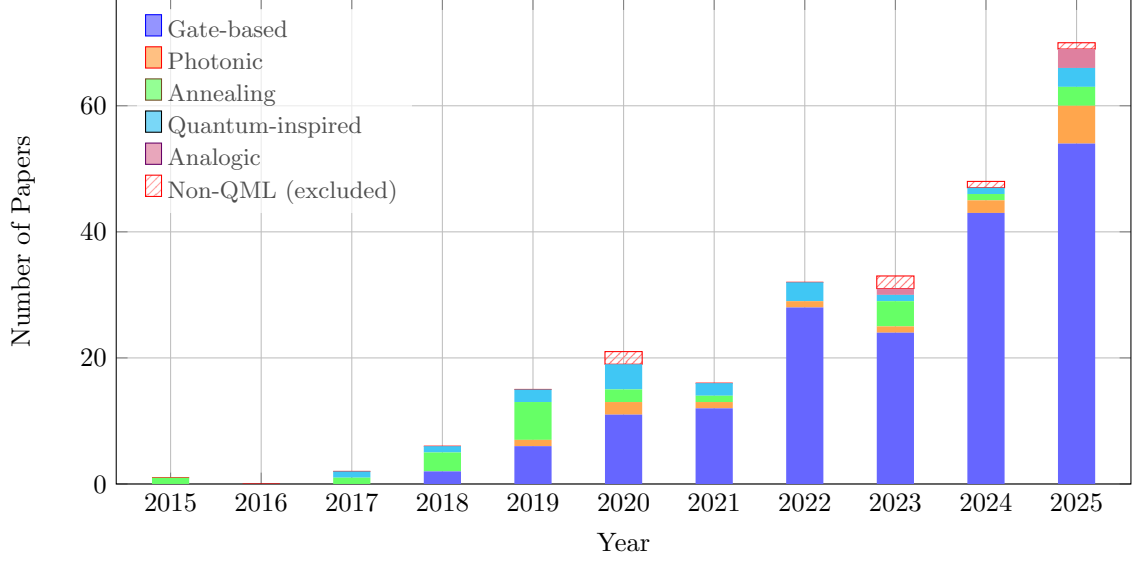


Figure 1: Number of QML+MNIST papers per year, broken down by modality. There is a clear increasing trend in the number of publications that include the terms MNIST and quantum machine learning on their title or abstract over the years. This rise reflects the adoption of MNIST as a benchmark for testing novel quantum approaches.

operators under the unitary describing the linear optical network. The fundamental gates in such a network are:

- Phase shifters which are $U(1)$ transformations acting on a single mode: $\hat{P}_\phi = (e^{i\phi})$, where $\phi \in [0, 2\pi]$,
- Beam splitters which are $U(2)$ transformations acting on pairs of modes, described by²:

$$\hat{U}_{BS}(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & i \sin(\frac{\theta}{2}) \\ i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix},$$

where θ is related to reflectivity of the coupler.

Any arbitrary unitary transformation $U \in U(m)$ over the optical modes can be decomposed into a sequence of such beam splitters and phase shifters, following the triangular [53] or rectangular [54] decompositions, also known as Reck and Clements decompositions, respectively. The resulting interferometer is a universal linear-optical processor, in the sense that it can perform any linear-optical operation.

The probability of observing an output state $|\vec{n}_{out}\rangle$ given input state $|\vec{n}_{in}\rangle$ and unitary U is given by:

$$p(\vec{n}_{out}) = \frac{|\text{Perm}(U_{\vec{n}_{in} \rightarrow \vec{n}_{out}})|^2}{n_1^{in}! \dots n_m^{in}! n_1^{out}! \dots n_m^{out}!},$$

where $U_{\vec{n}_{in} \rightarrow \vec{n}_{out}}$ is a submatrix of U obtained by taking n_i^{in} times the i th row of U , then n_j^{out} times the j th row of that matrix. $\text{Perm}(\cdot)$ denotes the matrix permanent, which is defined for a matrix M as:

$$\text{Perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^n M_{i, \sigma(i)}.$$

The problem of sampling from such an output distribution corresponds to the definition of boson sampling which was proposed by Aaronson and Arkhipov [55].

Quandela’s current hardware is based on quantum linear optics. The Ascella QPU described in [21] consists of a deterministic single-photon source based on a quantum dot, followed by a demultiplexer, thus preparing input Fock states. They are sent to a chip corresponding to a 12-mode interferometer containing thermo-optic phase shifters and directional couplers, which enable reconfigurable unitary

²Note that other beam splitter conventions exist that can involve additional parameters.

	Gate	Photonic	Anneal	Q-inspired	Analog	Non-QML
Number of papers	180	14	22	18	4	6
% of total	73.8%	5.7%	9.0%	7.4%	1.6%	2.5%
Binary tasks (#)	85	2	5	1	2	0
Multiclass (#)	95	12	17	17	2	6
Code available (#)	48	6	3	3	0	2
% with code	27%	43%	14%	17%	0%	33%
Hardware only (#)	8	3	9	4	1	-
Simulator+HW (#)	78	5	8	1	1	-
Simulator only (#)	94	6	5	13	2	-
Min.–Max. dimension	1–3530	1–1550	1–784	1–784	1–255	1–196
Mean [Var OOM]	72 [10 ⁴]	104 [10⁴]	34 [10 ⁴]	31 [10 ³]	-	-

Table 1: Summary of 244 arXiv papers (2015–2025) mentioning “QML+MNIST” in title/abstract. The reported dimensionality refers to the effective size of the quantum system used for encoding: for gate-based models this typically corresponds to number of qubits (and, where specified, expanded feature maps), for photonic models to the number of modes and/or photons, and for annealing or quantum-inspired models to effective variable counts. Values are those explicitly reported in the papers. Means and variances are computed per modality.

transformations. Output states are measured with SNSPDs. On top of that, an active stabilization and machine-learned transpilation is implemented to compensate for fabrication imperfections and phase shifts [56]. This architecture provides a fully programmable and controllable linear optical processor on which quantum circuits defined in *Perceval* can be executed natively.

At the software level, *Perceval* serves as the primary interface between algorithm design, numerical simulation, and hardware execution. It allows users to define photonic circuits through a high-level Python API, which automatically translates them into optical networks composed of phase shifters and beam splitters. It supports circuit composition and visualization, as well as multiple simulation backends, i.e. algorithms optimized for different tasks given a specific input state: *CliffordClifford2017* efficiently samples individual single output states; *Naive* based on Ryser algorithm [57], computes the probability or probability amplitude of obtaining a given output state; and *SLOS* and *Stepper* describe the exact complete output state either by evaluating the entire circuit or by evolving the quantum state incrementally through circuit gates. *Perceval* also support different types of photon detectors, and include noise modeling for photon loss, imperfect components, photon distinguishability, and single-photon purity, as well as hardware connectivity. For the purpose of this challenge, it was extended with ad-hoc gradient backpropagation capabilities, enabling integration into machine-learning workflows.

When designing an algorithm based on a programmable interferometer, the parameters will correspond to the phase shifts applied by the phase shifters and the mixing angles of the beam splitters. In a fully simulated model, both sets of parameters can be freely optimized in order to, for instance, minimize a task-dependent loss function. However, in practical photonic hardware, the beam splitters are usually implemented as fixed 50:50 couplers, while only the phase shifters are tunable. Consequently, the optimization effectively acts on the set of controllable phases which are sufficient to modulate the overall interferometer unitary and hence the output photon-count statistics.

The Quandela framework unifies a programmable photonic interferometer and the *Perceval* software library into a consistent environment for quantum machine-learning experiments. Its photonic-native representation allows researchers to benchmark realistic quantum algorithms on both simulated and physical hardware. In the context of this challenge, this stack enabled the implementation and training of variational photonic circuits to classify MNIST digits.

5 Model proposals and results

In this section, we present thirteen methods developed by the participants, which can be grouped within three main approaches, as is shown in Figure 2. In the first approach, models are trained end-to-end, utilizing the interferometer for feature extraction. In the second approach, models employ the quantum

interferometer for annotation purposes. In the third approach, the proposals use the interferometer for fine-tuning, either through transfer learning or for refining and correcting the model.

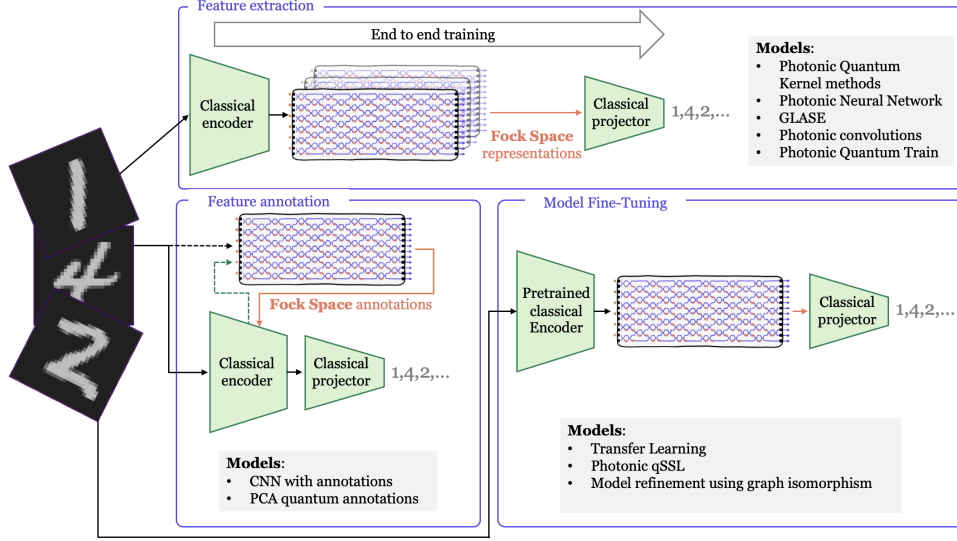


Figure 2: Three hybrid circuits trends observed in the challenge. When the photonic interferometer is used as a **feature extractor**, the model is trained end-to-end. When the photonic interferometer is used as a **feature annotator**, image or representations are passed through the encoder and their representations through the interferometer are fed as annotations to the encoder. In the case of **model fine-tuning**, a pretrained encoder is used and the photonic interferometer is used in the projection head, either for transfer learning, model refinement or self-supervised learning.

Most of the results stem from numerical simulations, both ideal (noise-free) and noisy. Additionally, several experiments were executed on hardware, more specifically on Quandela’s QPUs accessible through a cloud platform [41], and many simulations also leveraged GPUs through Scaleway’s Quantum-as-a-Service platform [40], as highlighted in Section 2. All quantum models are evaluated against classical baseline models, with comparisons made in terms of training and testing accuracy, number of trainable parameters, and computational cost measured in floating-point operations per second (FLOPS).

For each proposal, the details on interferometer architecture, data encoding, and hyperparameter values can be found in the Supplementary Materials.

5.1 Photonic interferometer for feature extraction

For the first category of models, the photonic interferometer is used to extract meaningful features from the data: the photonic interferometer acts as a feature extractor. It is sometimes combined with a classical encoder to enhance performance.

5.1.1 A quantum kernel method

Proposal. This first model is a photonic quantum-kernel whose classical counterpart is a Support Vector Machine (SVM) [58].

In this first approach, the photonic circuit is a m -mode photonic interferometer whose design follows [59]. Beginning from a Fock state $|n_1, n_2, \dots, n_m\rangle$, alternating layers of beam splitters and phase shifters are applied (Fig. 3). The resulting multi-mode photonic state $|\psi_{\vec{\varphi}}\rangle$ captures the structure of the data in a space of dimension $\binom{m+n-1}{n}$.

We estimate each kernel $\kappa(\vec{x}_i, \vec{x}_j) = |\langle \psi_{\vec{\varphi}_i}, \psi_{\vec{\varphi}_j} \rangle|^2$ by repeated photon-number measurements on each of the two circuits, yielding an $N \times N$ kernel matrix. Optionally, we apply a nonlinear post-processing step—either a sigmoid transform $\tanh(\alpha \kappa + \beta)$ or a polynomial map $(\gamma \kappa + c)^d$ —to adjust the kernel geometry before supplying it to a classical one-versus-all multiclass SVM solver.

Results. We benchmark on 600 training and 60 validation MNIST samples, balanced across digits. Each image is center-cropped and then downsampled to 14×14 using Principal Component Analysis (PCA).

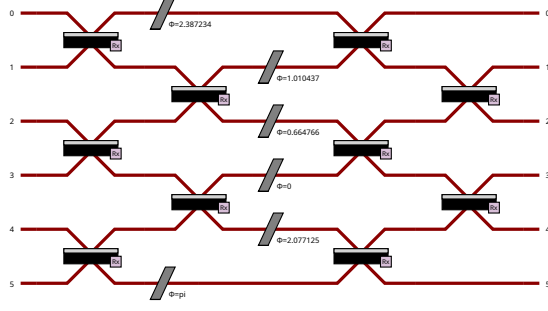


Figure 3: Example photonic circuit diagram with $m = 6$ modes. Photons are injected, pass through repeated beam-splitter (BS) layers, and accumulate phase shifts set by the PCA features. The measurement yields an m -mode photon-number distribution that encodes the feature vector.

This is equivalent to retaining $m = 20$ principal components. We construct a circuit with 20 modes and 5 photons.

Our validation results, summarized in Table 2, show that the linear classical kernel achieved the highest accuracy of 90.00%, while both the sigmoid and polynomial classical kernels reached 88.33%. The photonic quantum-kernel SVM, simulated without noise, using a sigmoid-transformed fidelity, attained 85.00% accuracy.

Model	Kernel	Val. acc. (%)
Classical SVM	Linear	90.0
Classical SVM	Sigmoid	88.3
Classical SVM	Polynomial	88.3
Photonic Q-SVM	Linear	82.0
Photonic Q-SVM	Polynomial	83.0
Photonic Q-SVM	Sigmoid	85.0

Table 2: Validation accuracy on reduced MNIST (600/train, 60/val).

As illustrated in Figure 4, increasing the number of injected photons n leads to a clear improvement in classification accuracy, indicating that larger photon counts can further narrow the gap with classical approaches.

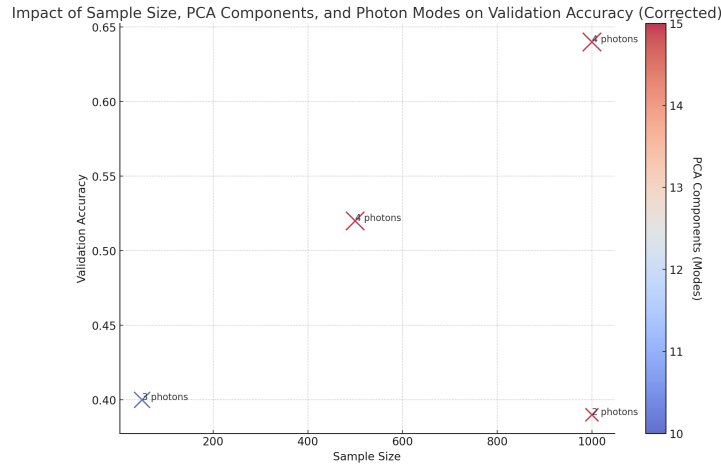


Figure 4: Accuracy vs. number of photonic modes m for the sigmoid-transformed kernel.

5.1.2 Leveraging the unitary dilation matrix for feature extraction

Proposal. This method leverages the *unitary dilation theorem* [25] that states that if $A \in M_n(\mathbb{C})$ of bounded norm, and $\|A\| \leq 1$ then

$$U_A := \begin{pmatrix} A & \sqrt{\mathbb{I}_{n \times n} - AA^\dagger} \\ \sqrt{\mathbb{I}_{n \times n} - A^\dagger A} & -A^\dagger \end{pmatrix}$$

This $2n \times 2n$ unitary matrix, twice as big as the original data ($n \times n$), shows a way of encoding a (scaled-down version of) A into a linear optical circuit. Following [25], the post-selection consists in observing n photons in the first n output modes. Therefore, to build the Unitary Dilation Encoding Neural Network (UDENN), a row of beam splitters is applied on the first n output modes. Following this row, the trainable model consists of L layers of a trainable unitary matrix followed by a brickwork construction of generic 2-modes circuit, to reproduce the effect of a classical convolutional kernel, extracting features from neighboring input values. One observation is that the associated circuit tends to shift the photons from one half of the circuit to another: the brickwork are therefore placed accordingly, as shown in Figure 5. A final post-selection condition is applied so that every photons end up in the same half of the circuit.

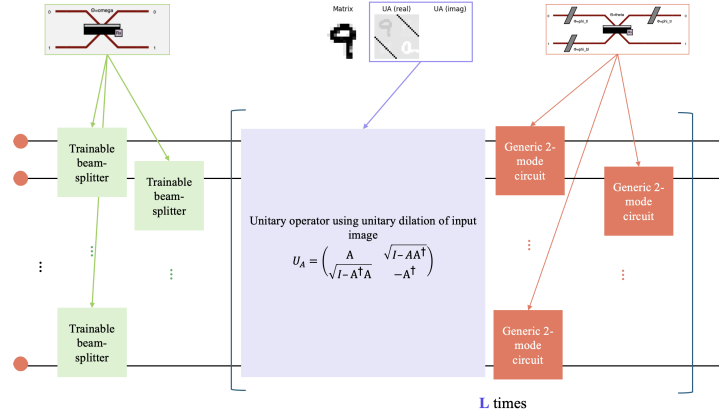


Figure 5: The trainable circuit is made of a first row of beam splitters, then L blocks of Unitary matrices followed by circuits made with generic 2-modes circuits

The output of this optical circuit is therefore made of the number of photons detected in each selected mode. A classical linear layer interprets this output and maps it to the 10 labels of the MNIST dataset. During training, the classical components were optimized using the Adam optimizer, while the optical components employed Simultaneous Perturbation Stochastic Approximation (SPSA). The two subsystems were trained in an alternating fashion. Here, a model with $L = 6$ blocks is chosen and the classical part is made of one hidden layer, for a total of 565 ($= 325 + 240$) trainable parameters.

Results. For fair comparison, the Unitary Dilation Encoding Neural Network (UDENN) quantum model with unitary dilation encoding and the classical baseline were matched for trainable parameters and input dimension (14×14). Validation set performance was assessed using accuracy and confusion matrix metrics. More information on the training dynamics is given in Appendix B. During the challenge, the implementation was done with slow optimization using SPSA and therefore, the training of the hybrid model is slower than the training of the classical model (1 sec/epoch versus 1.4 hours/epoch). This is mainly due to the postselection involved in the hybrid model. However, we can envision improvements to this training time by using probability boosting techniques in [25]. Another potential improvement lies in optimization of the back-end so the model can run faster. Table 3 presents the best validation accuracy across the training for the classical and hybrid models. Performance evaluation revealed that the Hybrid UDENN achieved 46.73% accuracy on the validation set. While this represents a performance gap compared to the classical CNN, the result substantially exceeds random baseline performance, validating the efficacy of the unitary dilation encoding approach for feature extraction in this quantum-classical hybrid framework.

5.1.3 A photonic quantum neural network

Proposal. This model consists of a **photonic neural network**. In this approach, the inputs of the interferometer are the 28×28 images whose pixels values are scaled using learnable weights. These

Model	Validation accuracy (%)
Classical CNN	52.33
Hybrid UDNN	46.73

Table 3: Best validation accuracy for a 5-epoch training for the UDNN

scaled pixels are encoded in phase shifters, following a circuit set-up as proposed in [23], shown in Figure 6. It is composed of 2 trainable generic interferometers from [54] and an encoding layer. We follow an encoding strategy such as $\forall x \in \mathbb{R}^{784}, S(x) = \lambda x$, where $\lambda \in \mathbb{R}^{784}$ is learned through gradient descent. Further details on this model design and ablation studies are presented in Appendix C.

The photonic neural network outputs are classified into the 10 MNIST digit classes through a trainable linear classification layer. The complete model is trained end-to-end in simulation using the Adam optimizer [60] with cross-entropy loss as the objective function.

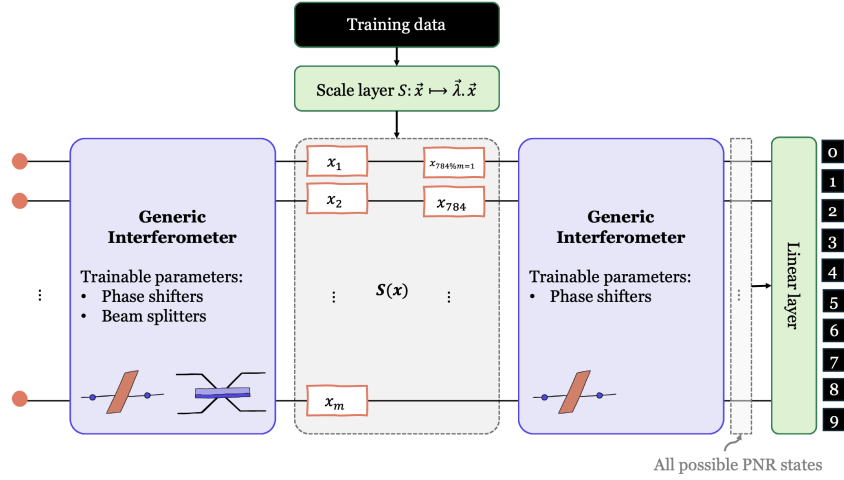


Figure 6: Photonic Quantum Neural Network, composed of two trainable generic interferometers [54], in purple, of an encoding layer (in gray) and two classical trainable layers (in green)

Results. Based on the ablation study explained in Appendix C, we find that an interferometer of 10 modes provide satisfying results. Increasing the number of modes would not provide valuable increase in performance compared to increase computational complexity. To provide a fair comparison with a classical model, in terms of number of trainable parameters, we chose a 2-layer MLP with ReLU activation that has 21475 parameters. We reproduce our experiments 5 times and our results are shown in Table 4.

Model	Test Accuracy	Number of parameters
Hybrid Photonic qNN	81.31 ± 2.04	21084
Classical MLP	94.14 ± 0.4	21475
SVM	95.38	7850

Table 4: Test Accuracy and Number of parameters for the different models trained

While the hybrid photonic qNN does not demonstrate superior performance compared to classical approach for this classification task, it achieves a respectable 81.31% accuracy on MNIST, indicating the viability of quantum-photonic architectures for machine learning applications. This result suggests that hybrid photonic systems, though not yet competitive with state-of-the-art classical methods, represent a promising foundation for future quantum machine learning implementations.

5.1.4 GLASE: Gradient-free Light-based Adaptive Surrogate Ensemble

Proposal. In this approach, we leverage a surrogate model to enable end-to-end backpropagation through a photonic quantum neural network (QNN). This model is a photonic adaptation of previous work that was designed for qubit-based circuits [37]. In classical machine learning, the backpropagation

algorithm is key to efficient training of deep neural networks. Typically for QNNs, training schemes rely on gradient estimation through the finite-difference method or the parameter-shift rule [61,62] – however, they both require a much higher cost in circuit evaluations and can be unstable in some settings. Our method circumvents this challenge by introducing a neural-network-based surrogate model that learns from data generated by the photonic backend. This allows integration into standard deep learning workflows while maintaining compatibility with quantum optical circuits simulated via boson sampling.

Our pipeline begins with a lightweight convolutional neural network (CNN) that extracts feature embeddings $\mathbf{z} \in \mathbb{R}^{256}$ from each 28×28 MNIST image $\mathbf{x}$. These features are mapped to quantum optical circuit parameters via a classical encoder $\phi = \Pi(\mathbf{z})$, where $\phi \in \mathbb{R}^M$ represents the programmable phase shifts in an M -mode photonic interferometer. The photonic circuit used is based on a boson sampling setup, where N indistinguishable photons propagate through a fixed linear optical network governed by a unitary transformation $U(\phi)$. To enable differentiable learning, we introduce a surrogate neural network $g_\alpha(\phi)$ trained to approximate the expected photon count per mode $\langle \hat{\mathbf{n}} \rangle : g_\alpha(\phi) \approx \langle \hat{\mathbf{n}} \rangle$. These values then go through a softmax layer to perform the classification task. More details on the mathematical background of this approximation as well as on the encoding strategy are given in Appendix D.

The surrogate model is periodically updated by minimizing the squared error loss

$$\mathcal{L}_{\text{sur}} = \|g_\alpha(\phi) - \langle \hat{\mathbf{n}} \rangle\|^2,$$

using simulated outputs from the quantum photonic backend. During training, this surrogate replaces the quantum layer in the backpropagation pass, allowing gradients to flow from the output loss $\ell_{\text{CE}}(\hat{y}, y)$ to the CNN encoder parameters θ . The full training loss becomes

$$\mathcal{L}_{\text{total}} = \ell_{\text{CE}}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{\text{sur}},$$

where λ controls the regularization strength for the surrogate fit (empirically set to 0.5 in our experiments).

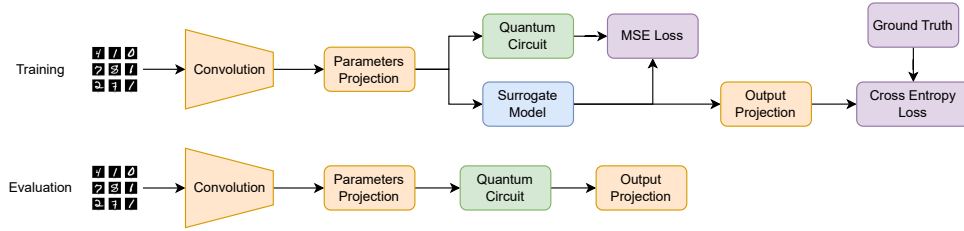


Figure 7: The GLASE architecture. A CNN encodes MNIST images into latent features which are mapped to photonic circuit parameters. A surrogate model approximates the photonic circuit’s output statistics to enable backpropagation during training. During the evaluation phase, the surrogate is not required and quantum hardware can be employed.

On top of enabling more efficient training, we believe that the surrogate network may also avoid optimization issues like barren plateaus [63], or those stemming from discrete hardware constraints. Indeed, in our numerical experiments, we found that the surrogate-assisted optimization remained stable throughout training. Overall, the surrogate model serves as a differentiable proxy that is periodically re-trained to match the behaviour of the true quantum photonic simulator, allowing for end-to-end optimization of the pipeline. Once trained, the surrogate can even be deployed as a fast approximate emulator for photonic inference when real hardware is unavailable.

Results. We conduct experiments on both simulated photonic backends and real photonic hardware to evaluate the performance of GLASE on MNIST classification. Training is performed for 50 epochs using a subset of 6,000 training and 1,000 validation images. The hybrid model integrates a photonic QNN backend with $P = 3$ photons over $M = 20$ optical modes. We report both training metrics and classification accuracy under different settings.

We first benchmark GLASE using the CliffordClifford2017 boson sampling simulator provided by the Perceval platform. Two GLASE variants with different network capacities are compared to classical baselines, including a mini ResNet and a vanilla multilayer perceptron (MLP). We find that both GLASE models outperform their classical counterparts in validation accuracy while using fewer parameters.

Real QPU Validation. To validate GLASE under hardware constraints, we deploy a compressed version using 16 modes and 3-photons on Quandela’s photonic QPU (Ascella), which supports real-time

Model	Params	Train Acc	Val Acc	Val Loss
Vanilla MLP	670k	100.00%	97.00%	0.2192
Mini ResNet	716k	100.00%	98.17%	0.1405
QNN (380k)	380k	100.00%	99.00%	0.1044
QNN (517k)	517k	100.00%	99.33%	0.0961

Table 5: Performance of GLASE and classical baselines on simulated MNIST subset.

boson sampling experiments. Inference is performed with 1,000 shots per image on 150 test samples and results can be found in Table 6.

Backend	Val Acc (%)	Val Loss
CliffordClifford2017 (sim)	91.00	0.3780
sim:sampling:h100	94.17	0.3125
qpu:ascella	76.79	0.8070

Table 6: Validation accuracy and loss on real and simulated photonic backends.

We attribute the accuracy drop on real hardware to noise such as photon loss and distinguishability, shot uncertainty, and hardware imperfections such as mode mismatch. Despite this, the result significantly outperforms random guessing (10%) and demonstrates the surrogate model’s potential for generalization across simulator and physical implementations.

Overall, GLASE’s performance on simulators is competitive with state-of-the-art classical models while using fewer parameters. On real QPUs, performance is limited by current hardware constraints, but shows encouraging trends, suggesting that the framework will scale well as quantum photonic processors mature. The modular surrogate-assisted training can be adapted to new interferometer configurations, making GLASE highly flexible for future hardware generations. In Appendix D, we share further insights about our results.

5.1.5 A photonic native quantum convolutional neural network

Proposal. The introduction of Convolutional Neural Networks (CNNs) in classical machine learning has revolutionised the field of computer vision. At the heart of the success of CNNs is an important inductive bias, namely *translation invariance*. Now, translation invariance is highly dependent on the strategy used to encode the images as quantum state. Here, we will use amplitude encoding on qudits. More specifically, for an $N \times N$ greyscale (i.e. 2-dimensional) image $\mathbf{x} = (x_{i,j})_{i,j=0}^{N-1}$, we will use $2N$ modes and 2 photons and encode it as:

$$|\psi_{in}\rangle = \sum_{i,j=0}^{N-1} \frac{x_{i,j}}{\|\mathbf{x}\|_2} |e_i\rangle |e_j\rangle \quad (1)$$

where the state $|e_i\rangle$ (resp. $|e_j\rangle$) is a state over N mode and a single photon such that the photon is located at the position i (resp. j). This choice of encoding will allow us to define translation invariant operations with respect to the input state.

In our approach, we produced a photonic analogue of the LeNet architecture [42] which consists:

- Photonic *convolutional layers* that implement several (dependent) local convolutions using repeated local interferometers. This operation is translation invariant (with respect to specific translations, see Appendix E.2)
- *Pooling layers* which reduces the dimension of the image using adaptive photon injections [64]. As for the photonic convolutions, this operation is also translation invariant
- A photonic *dense layer* which is simply a global universal interferometers on all the modes. This operation is the analogue of a classical dense layer which dismisses the 2D structure of the image and processes all of the obtained features together.

The output of the photonic circuit is a probability distribution over the possible detection patterns; for example, using photon-number resolving detectors, this therefore gives us $\binom{m_r + 1}{2}$ different probabilities where m_r is the number of remaining modes after pooling (recall that we only have 2 photons throughout the circuit). Since we required to discriminate between 10 different classes for the MNIST classification task, we treat the obtained probability distribution as a vector, and feed it to a classical (trained) linear transformation to obtain 10 different scores, followed by a softmax which will normalise those scores. The output of the full process will be the probability distribution over the ten different classes. The overall architecture is depicted in Figure 8.

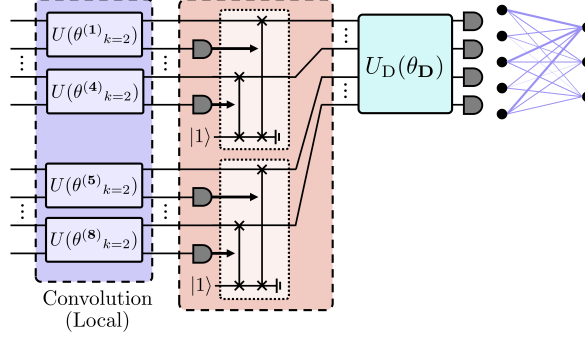


Figure 8: Architecture of the photonic QCNN

This approach was largely inspired from the qubit-based Hamming-weight preserving QCNN [65] and was developed at the same time as the very similar framework presented in [66].

Results. To train the overall model, we converted Perceval circuits (with feed-forward for the pooling layers) into a PyTorch module, therefore allowing us to train the model using backpropagation and the Adam algorithm [60]. The models were trained for 40 epochs (with a batch size of 100) with the cross entropy loss. We then compared the performance of our model with a classical CNN containing the same number of layers, the same kernel sizes, and the same final classical linear layer.

For the simulation to be tractable, we reduced the size of the images from 28×28 to 4×4 and 12×12 images, and restricted ourselves to only have one convolutional layer (and a single pooling layer). The evolution of the loss function and accuracies are shown in Figure 9. We observed that, for 4×4 images, the Quantum CNN (QCNN) clearly outperforms the classical equivalent, reaching a 58% test accuracy as opposed to 40% for the classical CNN, while having less trainable parameters (126 trainable parameters for the QCNN and 165 parameters for the classical CNN). For larger images, namely the 12×12 images, the classical CNN reaches a higher final accuracy (93% train accuracy, 90% test accuracy) compared to the QCNN (89% train accuracy, 88% test accuracy). However, the performance of the QCNN still remains close to the classical equivalent, while having significantly less parameters to train (926 parameters to train for the QCNN as opposed to 3681 parameters for the classical CNN).

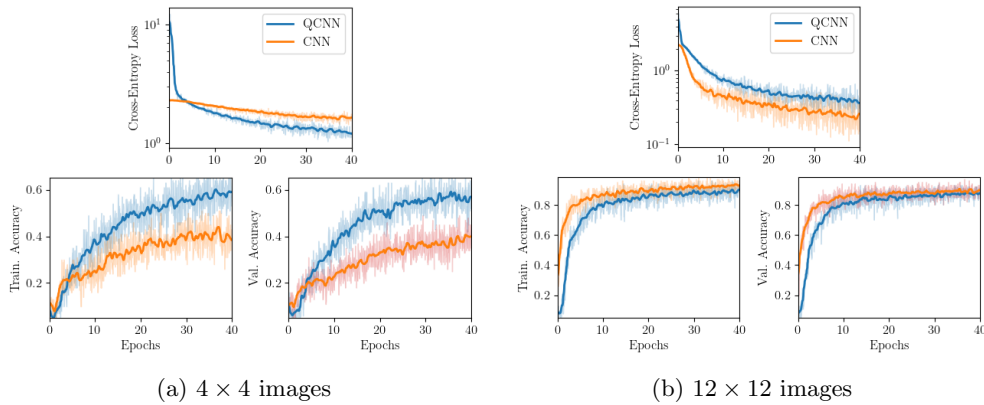


Figure 9: Comparison of the comparison of the QCNN and classical CNN.

5.1.6 A convolutional layer using a photonic quantum kernel

Proposal. We propose a hybrid quantum-classical image classification model that leverages a *Photonic Quantum Kernel* (PQK) alongside a classical CNN. The PQK acts like a convolutional kernel by processing each 2×2 image patch through a photonic interferometer. Pixel intensities are normalized and encoded as phase shifts, modulating the path of single photons through beam splitter layers. The output detection pattern becomes the quantum feature vector for that patch. The full image is scanned with stride 2, producing a feature map composed of 5 or 20 channels per patch. Boson sampling is implemented using *Perceval*. This method leverages quantum interference in high-dimensional Hilbert spaces to encode complex local correlations that classical kernels may miss.

Here, 2 types of PQK are proposed following different kernel strategies: a reservoir one and a trainable one. For a $k \times k$ kernel, $m = \lceil \text{kernel_size}^2/2 \rceil$ modes are used. The first m pixels are encoded in a phase shifter on each mode. This first layer of phase shifters is followed by a row of beam splitters and followed by the remaining $m - 1$ pixels to be encoded.

Two types of models are then built. The **hybrid model** leverages one (Model **A** on Figure 10) or two PQK convolutions (Model **B**) with ReLU activation, followed by a MLP. The first convolution as a 3×3 kernel, a stride of 1 and an output size of 16, while the second one has a 5×5 kernel, a stride of 1 and an output size of 32. The **parallel model** consists of two parallel branches. The first branch is a classical pathway where a conventional CNN with ReLU activation is applied to the grayscale input. The second branch is a quantum pathway (either trainable or non-trainable) that applies the two PQK-based convolutions with ReLU activation. The outputs from both branches are then concatenated and fed to a dense layer that maps them to the 10 classes. Further details about these branches are provided in Appendix F.

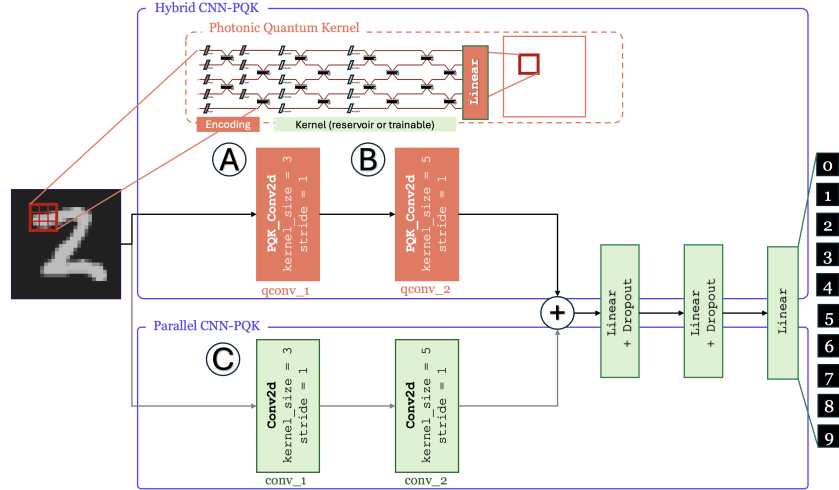


Figure 10: CNN with PCK architectures. In the single layer hybrid architecture (A), only one PQK convolution is used and then the output is forwarded to a MLP for final classification. For the two-layer hybrid architecture (B), the image is forwarded through 2 PQK convolutions with increasing kernel size (3×3 then 5×5), then through a MLP. For the Parallel CNN-PCK (C), the image is forwarded through 2 quantum convolutions on the one hand, and through 2 classical convolutions with similar kernel size (3×3 then 5×5) on the other end. Then, both outputs are concatenated and forwarded through a MLP for final classification

Results. We present our findings comparing the baseline CNN model and our hybrid quantum-classical architectures enhanced with Photonic Quantum Kernels (PQK). Our study highlights how the dimensionality and encoding strategy of PQKs influence model accuracy, learning speed, and generalization. Table 7 presents training and validation accuracy, while the figures in Appendix F support our qualitative analysis of the experimental outcomes.

For the hybrid models, it seems that 2 layers

The experimental results presented in Table 7 highlight the effectiveness of integrating quantum components into classical architectures for image classification. While the full classical CNN achieved the highest validation accuracy (97.67%), hybrid models with trained quantum encodings showed competitive performance, particularly the 1-layer hybrid PQK (92.58%) and the parallel PQK configurations (up

Model	Epochs	Quantum Encoding	Val Accuracy (%)	Train Accuracy (%)
Full Baseline CNN (no quantum)	5	None	97.67	98.05
Hybrid PQK 1 layer	5	Reservoir	11.08	12.33
Hybrid PQK 1 layer	5	Trained	92.58	92
Hybrid PQK 2 layers	5	Reservoir	10.58	12.33
Hybrid PQK 2 layers	5	Trained	11.8	12.33
Parallel PQK	5	Reservoir	97.12	95.33
Parallel PQK	5	Trained	96.77	95.67

Table 7: Validation and Test accuracies on MNIST (in %) after 5 epochs of training.

to 97.12%). In contrast, models using untrained reservoir encodings, especially in sequential (hybrid) architectures, performed poorly (11%), suggesting that quantum circuits require optimization to extract meaningful features. Notably, parallel quantum-classical integration appears more robust, preserving performance even with untrained quantum layers.

Conclusion: This analysis confirms that photonic quantum kernels can benefit classical models when used in a hybrid framework. Effectiveness depends heavily on the encoding scheme and feature dimensionality. Well-designed quantum circuits — particularly Type 1 encoding with sufficient entanglement — yield useful features that complement classical learning. These insights open the path to extending PQK-based models to more complex datasets and deeper architectures.

Remark: the encoding strategy described above presents a significant flaw: the phase shifters in the first row (initial phase) have no influence on the output distribution of the interferometer. Moreover, no light propagates through some of those elements due to the periodic input state. Consequently, for a 3×3 kernel, only 4 of the 9 pixels actually contribute to the transformation. While this is clearly a limitation of the current design, it also uncovers an interesting phenomenon: despite this masked behaviour, the system remains capable of classifying MNIST effectively. This suggests that MNIST can be processed with a partially masked kernel. Moreover, with a stride of 1, the masked kernel eventually covers the entire image, allowing all pixels to be considered over the course of the convolutional operation.

5.1.7 A convolutional layer using a photonic feature map

Proposal. Here, we introduce `qconv2d`, a parametric quantum convolution. Two-dimensional convolution is a common layer in neural networks for visual applications. Inspired by [67], the dot product of traditional convolution is replaced by a quantum circuit. The sliding window principle is preserved. Here, the quantum convolution is made of a photonic circuit with two parts: a fixed and untrained feature map to encode the data, followed by a trainable ansatz. Different architectures are considered for these 2 components and are depicted in Table 8. These models differ in terms of number of input photons, fixed and trainable components.

The output of such circuit is made of the probability of each possible output Fock state. The quantum convolution layer outputs m matrices, where m is the number of modes.

Figure 11 presents the overall framework while more details on the feature maps and ansatz are given in the Appendix G.

Results. The photonic circuit was simulated using a differentiable quantum layer. From results shown in Table 12a, quantum implementations lead to lower validation accuracy, but converges faster than classical models as shown on Figure 12b. Moreover, it is also highlighting that there is plenty room for improvement since classical machine learning does 10% better. Furthermore, the 10% performance gap between the hybrid model and classical CNN indicates room for improvement in the quantum-classical architecture. This differential suggests that with refined quantum circuit design, the hybrid approach may achieve competitive performance with conventional methods.

5.1.8 Photonic Quantum-Train

Proposal. At the core of the photonic Quantum-Train (QT) framework [36], a parametrized quantum circuit (PQC) is used as a *parameter generator* for a classical neural network (NN). The key observation is that a modest number of PQC controls can induce a joint distribution over exponentially (or

	Input State	Feature Map - Circuit	Ansatz - Circuit
1) Tristan	$ 1, 0, 1, 0, 0, 1, 0, 0, 1\rangle$	• 27 components (BS.H, BS.v and PS), in triangle-like shape • Set 9×3 angles using pixel value, with 3 different transformations: $-2\pi x$, $2\pi x$, $\sin(2\pi x)$	• MZI mesh with a depth of 6 • 96 components, 48 learnable parameters
2) Dagonet	$ 0, 0, 1, 0, 0, 0, 1, 0, 0\rangle$	• 15 BS in a cross arrangement • 9 phase shifters (one per mode) with angles set using pixel values: $(0.5 - x)\frac{\pi}{2}$	• Set of 1 BS.H + 2PS, and 1 BS.v + 2PS • 96 components, 96 learnable parameters

Table 8: Quantum circuit configurations for feature mapping and ansatz. The first configuration is made of a mix feature map [Achilles] and a MZI ansatz [Penarddun] while the second is made of a dispatch feature map [Odysseus] with a custom ansatz [Gofanon]

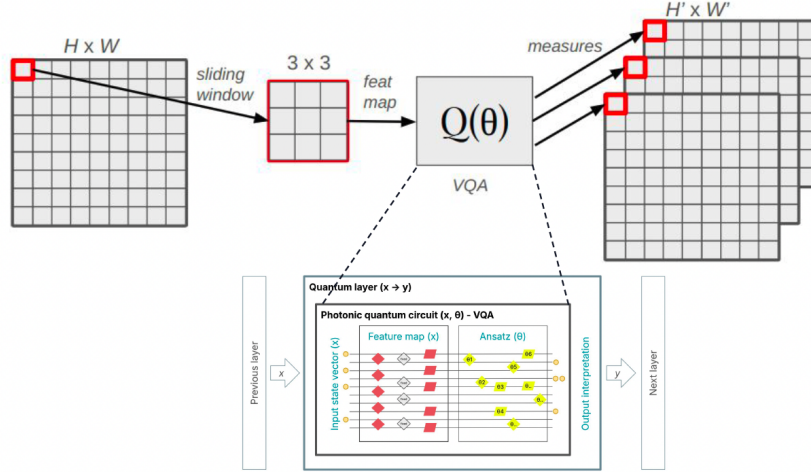


Figure 11: The two-dimension quantum convolution with 3×3 kernels and a stride of 2, with no dilatation and no padding. The output consists of 9 matrices of 13×13 coordinates

combinatorially) many computational-basis outcomes, with the number of degrees of freedom dictated by the Hilbert-space sector being addressed. We exploit this by mapping measurement probabilities to real-valued NN weights through a learned, low-rank tensor-network map.

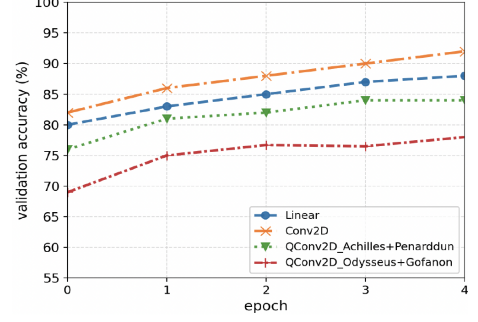
We instantiate two photonic quantum neural networks (QNNs), $QNN_1(\vec{\theta}^{(1)})$ and $QNN_2(\vec{\theta}^{(2)})$, with M_1 and M_2 optical modes, respectively. Each device is operated in a fixed-excitation (Hamming-weight) subspace with N_1 and N_2 excitations. In Appendix H.1, we show that, by steering the QNN controls one can populate at least m effective degrees of freedom for the target NN. A learnable *mapping model* G_v based on a matrix-product state (MPS) [68] allows to map the outputs of the QNN to the weights of the NN. This hybrid models can be trained using gradient descent as demonstrated in Appendix H.2. Figure 13 depicts the photonic quantum train scheme.

We implement the photonic QNN with a programmable multi-mode interferometer realized as a rectangular mesh of nearest-neighbour two-mode Mach-Zehnder Interferometers (MZIs), each composed of two balanced beam splitters and internal/external phase shifters, following the decomposition of Clements *et al.* [54]. More details of this decomposition are given in Appendix H.3.

Results. Our implementation (adapted from the *Perceval* library [22]) programmatically assigns and updates the interferometer parameters. Users may supply explicit lists $\{\theta_\ell\}$ and $\{\phi_\ell\}$ or initialize them randomly. On hardware, thermo-optic or electro-optic modulators provide continuous, real-time tuning of internal and external phases, supporting closed-loop optimization to minimize a task-specific cost. This decomposition yields a hardware-friendly layout with minimal optical depth, low mode-dependent loss accumulation, and robust reconfigurability—features that are advantageous for boson sampling,

Model	Validation accuracy	Time per epoch
linear	88	00'01"
conv2d ($s = 3$) + linear	90	00'01"
Tristan	84	07'20"
Dagonet	78	07'05"

(a) Validation accuracy after 5 epochs for the classical and hybrid models



(b) Validation accuracy over 5 training epochs

Figure 12: Comparison of classical and hybrid quantum models

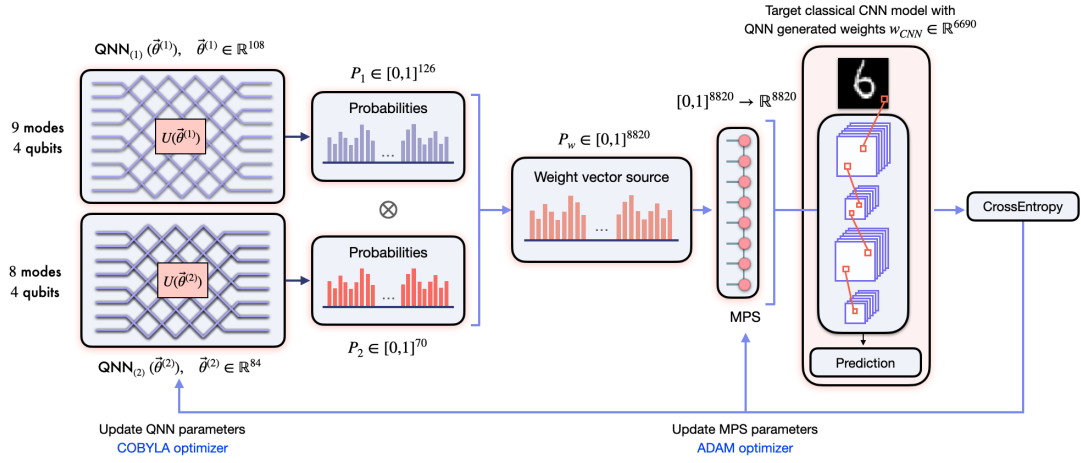


Figure 13: Overview of the photonic quantum-train scheme: 2 QNNs are trained with COBYLA optimizers so their outputs are mapped to the weights of a classical NN using a classical matrix-product state, trained using Adam optimizer.

quantum-enhanced machine learning, and other protocols relying on low-noise, large-scale multi-mode interference.

A thorough study of the parameter efficiency in photonic QT (presented in Appendix H.4) allows us to conclude that models with higher bond dimensions achieve consistently lower training loss and higher training accuracy, indicating enhanced expressiveness and optimization.

Figure 14 compares the photonic QT framework with classical model compression techniques, including weight sharing and pruning. The left panel shows testing accuracy versus model size, while the right panel plots the generalization error. QT offers superior accuracy for a given parameter count and outperforms classical baselines in the low-parameter regime.

Table 9 summarizes the number of trainable parameters needed to achieve comparable testing accuracy across different methods. The photonic QT model with bond dimension $D = 10$ achieves 95.5% accuracy using just 3292 parameters, less than half the size of the full classical CNN. At $D = 4$, QT requires only 688 parameters to achieve over 93% accuracy—surpassing both pruning and weight sharing at similar sizes.

In summary, the photonic QT framework exhibits strong parameter efficiency by achieving competitive performance with substantially fewer parameters. While higher bond dimensions improve accuracy, they also increase generalization error, indicating a trade-off that must be balanced. Classical compression techniques offer alternative strategies, but their performance saturates below that of the quantum-enhanced model. These results underscore the potential of photonic quantum systems for efficient neural network training and invite further exploration into hybrid training and regularization techniques to mitigate overfitting in high-capacity QNNs.

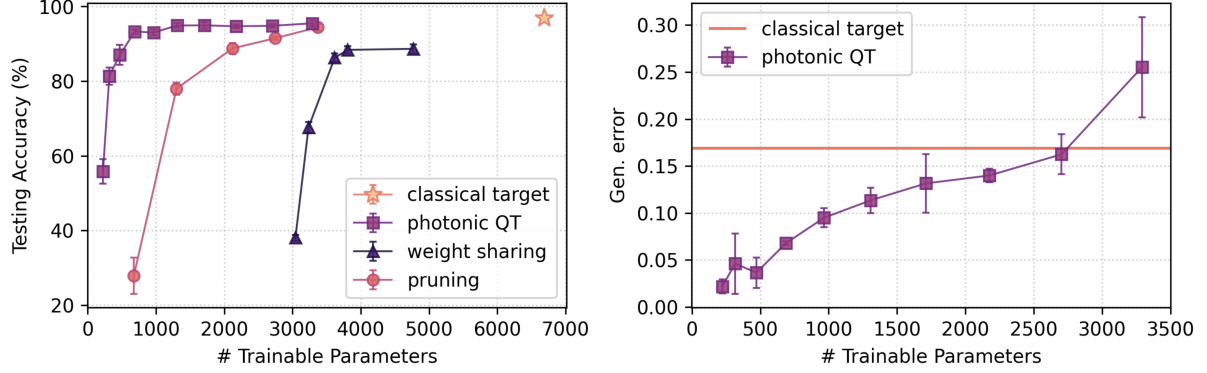


Figure 14: (Left) Testing accuracy vs. number of trainable parameters. (Right) Generalization error. Photonic QT outperforms weight sharing and pruning in accuracy, but shows an increasing generalization error as parameter count rises [36]

Table 9: Number of trainable parameters required to achieve comparable test accuracy [36].

Method	# Parameters	Test Acc. (%)
Original CNN	6690	96.89 ± 0.31
Weight Sharing	4770	88.67 ± 1.21
Pruning	3370	94.44 ± 0.92
Photonic QT ($D = 10$)	3292	95.50 ± 0.84
Photonic QT ($D = 4$)	688	93.29 ± 0.62

5.2 Photonic interferometer for quantum annotations

The following models present innovative frameworks that utilize photonic systems to perform feature annotation and enhancement. Here, the use of trainable classical models is necessary – the classical models are combined with photonic interferometers in the hope of improving the overall performance.

5.2.1 Enrich classical CNN representations

Proposal. In this method, we combine a photonic quantum system which is employed as a feature extractor, with a classical machine learning model. The core of our approach is a boson-sampler-based quantum embedding [64, 69], which transforms each image into a unique Fock-state probability distribution. The quantum embedding is used either as input to a classical neural network or concatenated with the original pixel data to enhance the feature set.

We encode the input data directly into the properties of the photonic circuit. Each MNIST image (28×28) is flattened and reduced to $d = 126$ dimensions using PCA, retaining approximately 93.7% of the variance. This dimension matches the number of programmable phase parameters in a 12-mode interferometer.

The reduced feature vector is directly mapped to the phase shifters of the 12-mode photonic interferometer. We explored multiple mesh configurations (triangular, rectangular, and convolution-inspired), but our final model uses two sequential rectangular meshes. Single photons are injected into predefined modes, and the interferometer transforms the state via beam splitter and phase shifter layers. The output photon count distribution (dimension $\binom{12}{2} = 66$ for $n = 2$ photons) forms the quantum embedding. Figure 15 is the layout used, two sequential rectangular meshes with programmable PS layers between BS stages.

We believe that a boson sampler can act as a nonlinear feature map: small differences in the PCA-reduced input can yield strongly decorrelated permanents and thus nearly orthogonal output distributions. This can improve class separability in the downstream classifier. We elaborate on this question in Appendix I.

To optimize the model’s performance, we utilize Tree-structured Parzen Estimators [70] for hyperparameter optimization. This method systematically searches for the best values for key parameters, such

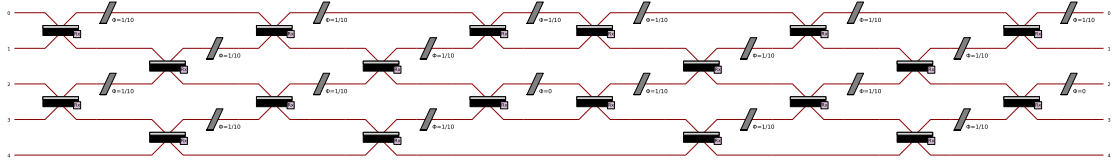


Figure 15: Interferometer consisting of two sequential rectangular meshes of beam splitters (BS) and phase shifters (PS), as implemented in the Perceval framework. Rectangular meshes [54] offer the same universality as triangular meshes but with reduced optical depth, which can lower noise accumulation and ease calibration.

as the learning rate, number of photons, and number of modes, ensuring an efficient training process. We also developed a “fast approach” to address the long simulation times encountered with the full dataset. This strategy uses a smaller subset of the MNIST data and fewer shots, allowing us to quickly prototype and test the model’s feasibility and learning capabilities. This rapid experimentation enables us to gather preliminary results and insights in a fraction of the time, paving the way for more extensive, full-scale tests on real QPUs.

Results. We performed our simulations using Perceval’s SLOS backend (noiseless) and, for benchmarking, we used the GPU-enabled `sim:sampling:214` backend. Remote execution using `sim:sampling:214` was slower than local SLOS but provided a more realistic evaluation setting. We note that backend choice affected runtime but not accuracy. For the classical baseline, we employed an MLP trained on the 126 PCA features, and both models are matched for comparable parameter counts.

Our approach reaches a validation accuracy of 96.50% with a final cross-entropy loss of 0.1239, surpassing the classical PCA baseline (93.8%). Macro-averaged F1-score is 0.9636, with per-class F1 ranging from 0.942 to 0.984. The gains of our hybrid model were largest for digits 3, 5, and 9.

5.2.2 Hybrid feature extractor

Proposal. The following model is a hybrid feature extractor consisting of 2 main components depicted in Figure 16:

1. A trainable **Quantum Block** encodes the classical data into the Fock space. We consider input states that are Fock states of the form $|1, 0, 1, 0, \dots\rangle$. We define a quantum circuit alternating encoding and processing layers. The goal of the encoding layers is to inject the input data as phases in the circuit, while the preprocessing layers define the trainable quantum parameters. Specifically, the encoding layers are formed by at most one phase shifter per mode, with a total number of phase shifters being equal to the input data dimension d . This dimension verifies by construction $d \leq m$. The processing layers consist of fixed beam splitters and trainable phase shifters in a triangular configuration. A grouping output strategy is used to partition the m modes into 10 disjoint groups.
2. **Classical Block.** In addition to the quantum block, we define a feedforward classical layer with a ReLU activation. This layer outputs a classical embedding of the d -dimensional input vector.
3. **Postprocessing layer.** This last layer consists in a learnable fully connected layer which maps the concatenated embeddings from the Quantum and Classical blocks to the 10 classes

Results. To investigate the impact of each component in our architecture, we compare three models:

- Classical-only model: the quantum blocks are removed.
- Almost-fully-quantum (AFQ) model: In the encoder, we consider only the quantum encoder, without the classical encoding.
- The proposed hybrid model: the full approach described above with both classical and quantum parts.

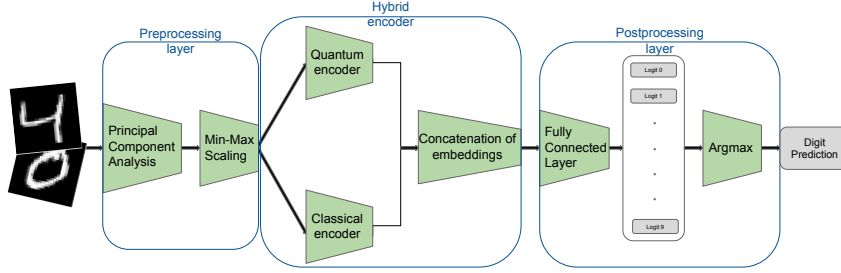


Figure 16: Architecture of the hybrid feature extractor. We can distinguish three main parts: a preprocessing layer where PCA is applied to the data, a hybrid encoder made of photonic interferometer and a classical layer, and a post-processing layer to map to the 10 classes of MNIST.

The classical and hybrid models have a fairly similar number of parameters (2290 and 2122) and FLOPs (852.48 KFLOPs and 750.72 KFLOPs), while the AFQ model only had 112 parameters due to computational constraints.

In Figure 17, we present the mean test accuracies computed over 25 independent runs. Overall, the AFQ model demonstrates inferior performance compared to both the classical and hybrid counterparts. Additionally, the hybrid model exhibits a marginal improvement over the classical model, suggesting that the inclusion of the QuantumLayer may contribute to extracting features beneficial for the classification task.

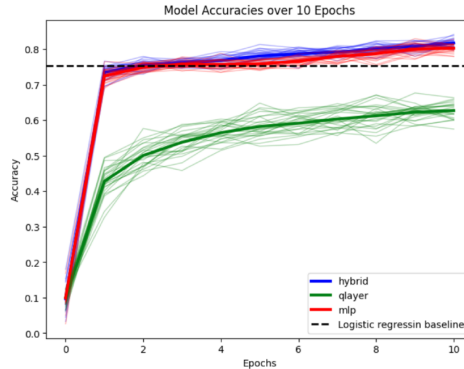


Figure 17: Average test accuracies over 25 runs.

5.3 Photonic interferometer for model fine tuning

In this category of models, photonic quantum circuits are using for fine-tuning, and combined with trainable classical encoders.

5.3.1 Transfer learning

Proposal. We propose a hybrid quantum–classical transfer-learning (TL) architecture based on a photonic interferometer. This technique is inspired by [71] where the authors highlight the possibility of transferring some pre-acquired knowledge at the classical-quantum interface.

The transfer learning framework consists of two stages: a classical feature-extractor pretrained on a large-scale image dataset, followed by a photonic neural network that refines and classifies MNIST features in the optical domain. The classical feature extractor is the backbone of a pretrained convolutional neural network (e.g., ResNet-18 [72]) trained on CIFAR-10. The learned representation $z \in \mathbb{R}^{256}$ is then encoded using two consecutive methods: first, a classical linear encoding maps the 256-dimensional representation vector to the target input dimension; second, quantum feature embedding maps the classical features to phase shifters applied to specific modes. These methods are detailed in Appendix K. The photonic circuit itself is composed of a series of Mach–Zehnder Interferometer (MZI) blocks arranged in a cascaded Fourier-mesh topology (Figure 18), parameterized by programmable phase-shifters $\{\theta_i\}$.

Through proper sequential arrangement, these interferometers can realize the complete set of $SU(2)$ operations necessary for universal optical quantum processing [54]. The output layer measures the number of photons at each of the m modes channels after processing through the optical network. These numbers are then mapped classically to the 10 classes.

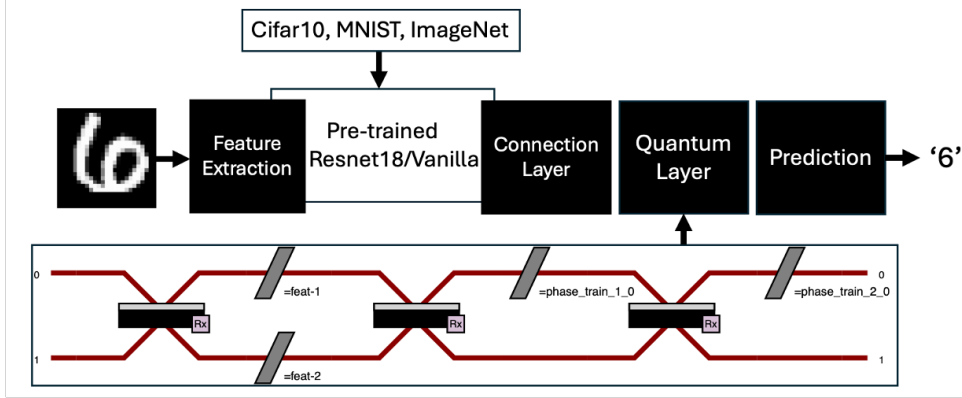


Figure 18: Schematic of the hybrid photonic transfer-learning architecture. The pretrained CNN extracts a 256-dimensional feature vector from each MNIST image. These features are encoded into optical modes, which then propagate through a programmable photonic interferometer consisting of layered MZI blocks. Photon-counting detectors at each of the 10 output waveguides produce class scores.

Results. To investigate the viability of quantum-enhanced classifiers in a transfer learning (TL) setting, we performed a series of experiments using both classical and quantum post-processing. Specifically, we tested several TL pipelines involving ResNet18 (pretrained on ImageNet or CIFAR-10) and a simple CNN trained on CIFAR-10. The goal was to evaluate the statistical effectiveness of quantum encodings with respect to classical models under a variety of transfer conditions.

Each TL strategy used either full MNIST or selected MNIST classes as the target dataset. For the quantum models, the final classification step was replaced by a photonic encoding and simulated boson sampling circuit. In contrast, classical baselines retained fully classical linear classification heads.

We conducted a series of transfer learning experiments to evaluate the effectiveness of classical versus quantum classifiers, using ResNet18 and a vanilla CNN across several source-target configurations. All results are presented in Table 10. The first experiment reproduced the setup from Schuld et al [71], where a ResNet18 model pretrained on ImageNet was used as a feature extractor for MNIST classification. As in the original paper, the classical model achieved over 90% accuracy without retraining the backbone. The quantum classifier, built using a boson sampling layer, managed to reach accuracy of $\sim 67\%$. This result is better than random guessing in a 10-class setting but still it’s not close to the accuracy achieved by the classical model. Extending this to general 10-class MNIST classification with the same ImageNet-trained ResNet18, we observed similarly high classical performance (over 92%). The performance of the quantum model is also similar to the previous experiment ($\sim 67\%$). We then evaluated binary TL tasks by selecting visually distinct MNIST digits (e.g., “1” vs. “8”), where classical models achieved near-perfect classification accuracy ($>99\%$). The quantum models managed to performed equally well in this simplified task. The achieved accuracy was $\sim 98\%$. A similar outcome emerged when classifying visually similar digits like “3” and “5”; although the task was more difficult, the classical model still surpassed 97% accuracy, and the quantum classifier managed to reach $\sim 96\%$.

To isolate the benefit of feature alignment, we trained ResNet18 directly on full MNIST and transferred it to a 2-class MNIST subset (“3” vs “5”), to better map the method in [71]; unsurprisingly, the classical model reached $>99\%$ accuracy, whereas the quantum model reached 98%. We also tested domain transfer by using ResNet18 trained on CIFAR-10 and applying it to MNIST. Despite the domain mismatch, classical performance remained relatively high ($\sim 86\%$), reflecting the general utility of early convolutional layers. The quantum classifier yielded 46%. Both models were evaluated on the 10-class case. Finally, we implemented the transfer from a shallow CNN (either one or two convolutional layers followed by a fully connected layer) trained on CIFAR-10. Even with this simpler architecture, classical accuracy exceeded 95% (in the 10-class case), while quantum performance was around 55% in the 10-class case and $>99\%$ in the binary case. Across all settings, the classical models benefited substantially from transfer learning, while the quantum models managed to match the performance of classical methods in some tasks. However, in more complex tasks where the dataset contains 10 classes, even though their

performance was better than random guessing, it was much worse compared to classical models.

Mode selection: In all experiments involving the boson sampling layer, we explored a range of photon and mode numbers to assess their impact on the model performance. Specifically, we varied the number of modes from 10 (the expected minimum needed to potentially encode digit identity in a 10-class task) up to 24 (our largest tested system size for a suitable runtime). For the number of photons, we experimented with values between 2 and 4 to make it compatible with the first linear layer. These choices were guided by the need to balance computational tractability with the expressive capacity of the quantum system. However, despite this range of configurations, none of the tested combinations led to a noticeable improvement in classification accuracy.

Experiment	Classical Accuracy	Quantum Accuracy
ResNet18 (ImageNet) → MNIST (Reproduction)	~93%	~67%
ResNet18 (ImageNet) → MNIST (10-class)	>92%	~67%
ResNet18 (ImageNet) → MNIST (1 vs. 8)	>99%	~98%
ResNet18 (ImageNet) → MNIST (3 vs. 5)	>97%	~96%
ResNet18 (Full MNIST) → MNIST (2-class)	>99%	~98%
ResNet18 (CIFAR-10) → MNIST	~86%	~46%
Vanilla CNN (CIFAR-10) → MNIST	>95%	~55%

Table 10: Comparison of classical and quantum transfer learning accuracy across different source-target setups. Classical models consistently outperform quantum counterparts, often by a significant margin.

5.3.2 Self-supervised learning

Proposal. This model employs Self-Supervised Learning (SSL) to extract meaningful feature representations through pretext tasks, thereby eliminating the need for labeled data during backbone training. SSL represents a significant paradigm in machine learning that enables the exploitation of vast unlabeled datasets for representation learning. Prominent frameworks in this domain include SimCLR [73] and Barlow Twins [74]. More precisely, the objective is to leverage photonic quantum computing to learn from unlabelled data. A previous work [75] leverages a gate-based framework as a representation network in a SSL framework. Here, we propose to use a photonic interferometer as a projector network from the representation space to the loss space. The self-supervised framework is as follows: an encoder is taking the 28×28 MNIST images as inputs and map them to a representation space of dimension $\mathbb{R}^r$, then, these representations are encoded in phase shifters following [23] implementation. The interferometer outputs are subsequently compared using a similarity metric *sim*, which serves as the self-supervised loss function for the system. The underlying principle is to enforce invariance in the learned representations: augmented views derived from the same input image should yield similar representations when processed through the interferometer, thereby encouraging the model to learn transformation-invariant features

A fundamental component of the SSL paradigm is data augmentation [73]: given an input image $\vec{x}_i$, two distinct augmentation functions are applied to generate transformed versions. Common augmentation techniques include cropping, rotation, blurring, and color distortion, as performed in [75]. However, these transformations must preserve the visual distinguishability of the underlying object, which constrains the applicable augmentation strategies for certain datasets. Specifically, for MNIST, the grayscale nature and orientation-dependent semantic content preclude the use of rotation and color distortion. MNIST is not a great candidate for SSL learning but our goal here is to provide a proof of concept for photonic quantum SSL. Therefore, we perform crops at the top left and bottom right of the MNIST image. Gaussian blurring with a low probability was investigated but it was not benefiting the training. Figure 19 depicts the overall SSL framework.

Results. For fair comparison with a classical baseline, the quantum layer can be replaced by a Linear layer that maps the representations to a loss space of similar dimension as a Quantum Layer would (i.e. $\binom{m+n-1}{n}$ with photon number resolving detectors or $\binom{m}{n}$ with single photon receptors, where m and n stands for the number of modes and photons). To evaluate the learned representations, we perform a linear evaluation: the trained backbone is frozen and a fully connected layer is trained on top of it to map the representations to the correct number of classes. To assess the utility of the SSL training, we also perform a linear evaluation on a frozen random backbone. Table 11 shows the results after training different backbones for 100 epochs and performing linear evaluation during 50 epochs. All experiments are reproduced 5 times. It is important to highlight that these results are not comparable with a fully

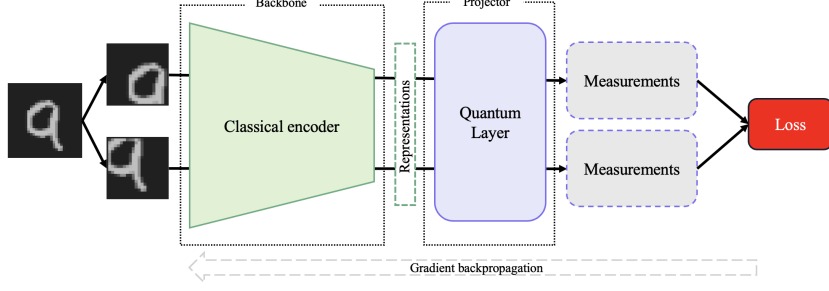


Figure 19: Self-Supervised Learning framework made of a classical backbone, a quantum projector and a classical loss.

trained model: for instance, a MLP without activation (LINEAR: $400 \rightarrow 8 \rightarrow 10$) can provide a validation accuracy of 91.13%. To evaluate the efficacy of our approach, we conduct comparative analysis across three backbone configurations: those trained using the proposed qSSL pipeline, those trained via classical SSL methods, and untrained networks with random weight initialization serving as a baseline.

Model	Loss	Hidden dim	Quantum	Trained	No bunching	Val. ACC.
MLP (LINEAR: 400 → 8)	INFONCE	8	Yes	Yes	False	35.03 ± 3.47
					True	42.8 ± 5.83
				No	False	47.43 ± 3.15
					True	45.70 ± 4.03
			No	Yes	False	37.9 ± 3.48
					True	39.97 ± 1.66
				No	False	40.77 ± 4.58
					True	39.9 ± 2.81
			Random			
MLP (LINEAR: 400 → 32 → 8)	INFONCE	32-8	Yes	Yes	False	28.2 ± 1.24
					True	32.17 ± 2.69
				No	False	29.27 ± 1.03
					True	30.03 ± 3.78
			No	Yes	False	34.57 ± 5.56
					True	24.4 ± 4.43
				No	False	32.57 ± 7.24
					True	27.8 ± 2.28
			Random			

Table 11: Validation Accuracy for different models after linear evaluation

5.3.3 Future Direction: Leveraging graph isomorphism to classify digits

Proposal. In this section we outline a novel approach for handwritten digit classification that leverages the computational properties of photonic quantum processors. Although a detailed evaluation of its performance is left for future work, we present here the main steps of the proposed method.

Most traditional approaches rely on convolutional neural networks [76, 77] or support vector machines [78, 79] operating on pixel intensities. Our method transforms MNIST images into graphs through superpixel segmentation [80–83], then uses the matrix permanent of the adjacency matrix of each graph (as well as various of its subgraphs) as a quantum feature. These quantities are particularly relevant in the context of photonic quantum computing, as the matrix permanent governs the probability amplitudes of multiphoton interference events and therefore constitutes the core algorithmic primitive of linear-optical quantum computing [25, 84].

In this approach, the underlying intuition is to represent MNIST images as graphs and to hypothesize that graphs corresponding to the same digit share identical permanent values. Interestingly, this representation is inherently robust to standard image transformations such as rotations or horizontal flips, since these operations preserve the graph structure and therefore its permanent. As a result, the method

will fail to distinguish between digits such as 6 and 9, which are related by such transformations. This invariance, however, also indicates that the representation captures a fundamentally different type of information than traditional image-analysis methods based on local pixel intensities. As an illustrative example of its discriminative power, consider two images represented by graphs G_1 and G_2 of equal size. If these graphs satisfy Theorem 2 in [25]—that is, if all subgraphs of G_1 and G_2 have the same permanent under some fixed bijection of vertices—then G_1 and G_2 are isomorphic, and consequently represent the same underlying graph structure.

Each digit is thus represented by a set of numbers being the permanents of the adjacency matrix of its graph and some randomly chosen subgraphs. These features capture both geometric and topological features of the handwritten character. This approach demonstrates how quantum photonic systems can extract meaningful complementary information that classical models may overlook. Through their intrinsic ability to estimate matrix permanents via boson-sampling protocols, photonic devices can enrich classical neural networks with quantum-derived features relevant to computer vision. Rather than competing with standard encodings, these quantum-estimated features provide orthogonal information that can be leveraged by classical networks.

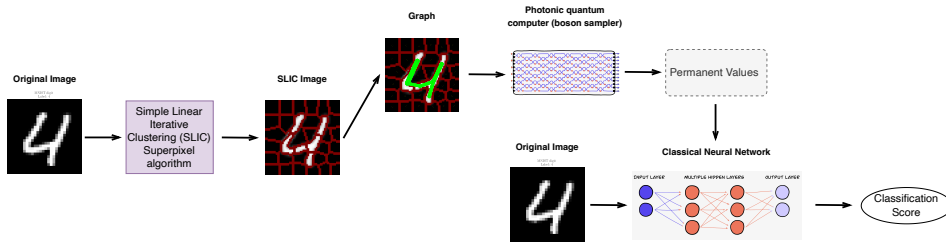


Figure 20: Workflow of the proposed hybrid quantum-classical model for classifying MNIST digit images by transforming them into graphs and using photonic quantum computers to compute their permanent. First, each image is divided into superpixels using SLIC, and the centroid of each superpixel can be treated as a node in the corresponding graph. To ensure a fixed-size graph representation, we select K nodes and edges are constructed between nodes associated with high-intensity regions (illustrated in green). The resulting graph structures serve as inputs to a photonic quantum processor, which is used to evaluate permanent values. Finally, the quantum-generated features are combined with a classical neural network for the final classification step.

We will now describe our protocol for constructing a graph from a MINST image. The first step is the *superpixel segmentation* [81]. Each 576 pixel MNIST image is transformed into an *coarse-grained* image of $M < 576$ pixels (called *superpixels*) using the Simple Linear Iterative Clustering (SLIC) algorithm [80,85]. Then, we choose the K superpixels of highest intensities, discarding all other superpixels. The centroids of these K superpixels correspond to the K nodes of our graph, we label these nodes $1, \dots, K$ according to some arbitrary ordering. Our rule for constructing the K edges of our graphs is

1. For all nodes $i = 1$, sweep over all nodes $j \neq i$ and choose j_m such that the geometric distance $|i - j_m|$ is minimized.
2. Connect the pair (i, j_m) by an edge.
3. Repeat steps 1 and 2 for all values $i \in \{2, \dots, K\}$ with an additional constraint : discard j_m and restart the search if (i, j_m) are already connected by an edge in a previous step.

In this way, we obtain for each image an associated graph G with K vertices and edges. More complex rules for constructing G are possible—for instance, one could ensure that an edge connecting two high-density regions does not cross a large low-density area, thereby preserving the spatial coherence of digit strokes. However, we believe the above simple rule already enables non trivial performances.

For each image G , we feed into the classical neural network a set $\mathcal{S}$ of values corresponding to the permanents of the various subgraphs of G . In the worst case, as described in Theorem 2 in [25], $|\mathcal{S}|$ is the number of all possible subgraphs of G of any size. It would be interesting to investigate whether we obtain meaningful performance enhancements when using a smaller number of subgraphs, selected at random. Indeed, the flexibility of our approach lies in the fact that M , K , and $|\mathcal{S}|$ can all be adjusted to optimize performance.

In order to compute the various permanents of the subgraphs using a linear optical circuit, we use the approach of [25] (see also section 5.1.2) where the adjacency matrix of G is encoded onto the linear optical circuit, by unitary dilation, and relevant output events are post-selected on to estimate $\text{Per}^2(A_{G_s})$ (and consequently $\text{Per}(A_{G_s})$), where A_{G_s} is a $(0, 1)$ matrix representing the adjacency matrix of a subgraph G_s of G .

In Figure 21, we provide examples of original MNIST images being transformed into graphs using the technique described previously. Also, Figure 20 contains a diagram that describes the workflow of the model.

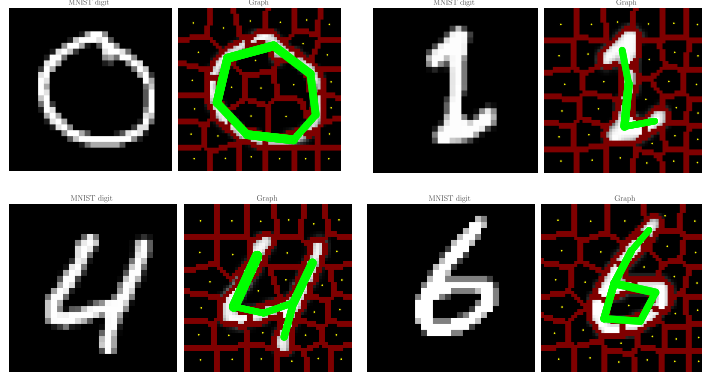


Figure 21: Examples of MNIST digit images transformed into fixed-size graph representations. Black segments denote superpixels corresponding to background regions, while white segments highlight digit strokes. Green nodes and edges represent high-intensity areas capturing the main digit structure. The number of nodes and edges is fixed to ensure uniform graph size, enabling their use as inputs to the photonic quantum processor.

6 Discussion

The results obtained from the Perceval Challenge did not reveal any clear evidence of a heuristic quantum advantage on the studied task. This outcome is both expected and informative: the classification problem considered here is already fully solved by classical machine-learning techniques and does not exhibit any structural features for which quantum computation would be expected to provide an edge. Rather than seeking to outperform classical methods, this work provides the first unified set of baseline performances for a wide range of photonic machine-learning (ML) strategies. These results offer a foundation upon which future studies can build, in line with the argument formulated in [8]. The challenge demonstrates that systematic benchmarking and reproducible experimentation are more valuable at this stage than isolated claims of superiority. In this sense, the Perceval Challenge is less a competition than a collective exploration of what photonic learning systems can currently achieve.

6.1 Lessons from the collective experiment

Gathering thirteen independently developed methods within a common evaluation framework provides a rare snapshot of the current design space of photonic machine learning. The approaches covered a broad spectrum—from fully end-to-end quantum models (kernels, neural networks and CNN), to architectures where the interferometer served as a feature extractor (enhanced CNNs, enhanced MLP), to methods exploring fine-tuning or transfer-learning strategies (Transfer Learning and quantum Self Supervised Learning (SSL)). While these experiments covered diverse uses of photonic components, none of the approaches explicitly followed a hybrid strategy or aimed at augmenting a state-of-the-art classical model with a quantum module. Exploring such combinations—where quantum photonic circuits could enrich or specialize parts of a classical learning pipeline—could therefore represent an important avenue for future research. This perspective aligns with the idea that near-term quantum ML may benefit less from full replacement of classical architectures than from targeted integration within them.

The Challenge also highlights that hardware-native approaches, such as those relying on the application of the intrinsic permanent computation at the core of boson sampling, should be viewed as complementary to more generic variational strategies. While variational models offer flexibility and

trainability, hardware-native circuits embody the physical expressivity of the underlying photonic platform. Comparing both families under a unified benchmark provides insight into how future architectures might blend these paradigms—leveraging hardware efficiency while retaining algorithmic adaptability. The coexistence of these approaches within the same challenge illustrates the field’s diversity and the need for frameworks that allow fair comparison between fundamentally different learning paradigms.

Crucially, the Perceval Challenge represents the first initiative of its kind and scale in photonic quantum computing, bringing together a wide range of teams and methodologies within a common experimental setting. Participants from diverse backgrounds—spanning quantum information, quantum optics, computer science, and artificial intelligence—worked over several months to develop, test, and refine their approaches. This diversity of expertise fostered cross-pollination of ideas and demonstrated that significant progress in quantum machine learning can emerge from open, interdisciplinary collaboration.

Finally, this collective effort underscored the need for scalable frameworks and robust tooling capable of supporting even larger and more complex benchmarking activities. Conducting the Challenge required the development of dedicated infrastructure to manage submissions, training, and evaluation in a reproducible way. Building on this experience, expanding such infrastructure to handle larger datasets, deeper circuits, and hardware-in-the-loop testing will be essential to sustain community-scale progress. Establishing a standardized, open framework for photonic ML experimentation could make collaborative challenges routine rather than exceptional—accelerating discovery and solidifying best practices across the field.

6.2 Reproducibility and methodological convergence

An essential contribution of this work lies in its commitment to reproducibility. The full codebase of all the thirteen implementations is publicly available in a single repository (<https://github.com/Quandela/HybridAIQuantum-Challenge>), allowing others to rerun, modify, or extend the experiments. This level of transparency remains uncommon in quantum ML research, where bespoke setups and restricted access to hardware often hinder replication. In addition to the open availability of the code, reproducibility in this Challenge was also reinforced by the fact that most of the proposed approaches were hardware-compliant, designed with the constraints of photonic quantum processors in mind. Several groups even confirmed their results on actual quantum photonic hardware (QPU), demonstrating that the reported performances are not limited to simulation environments. This combination of software transparency and experimental validation represents a strong step toward reproducible and verifiable research in photonic machine learning.

The Perceval Challenge thus establishes a practical standard for openness—similar in spirit to the role of ImageNet or GLUE in the AI community, where progress depends on shared baselines and community-wide evaluation protocols.

Equally significant is the interdisciplinary diffusion of ideas observed throughout the Challenge. Several participants came from classical AI backgrounds, bringing with them rigorous practices in hyperparameter tuning, ablation studies, and validation methodology. Their contributions demonstrate how methodological rigour from the AI world can directly benefit quantum research, ensuring that claims are statistically grounded and experimentally reproducible. This exchange exemplifies a broader cultural shift: quantum ML is evolving from proof-of-principle demonstrations toward data-driven, engineering-oriented experimentation.

6.3 Limitations and path forward

A consistent limitation reported by nearly all participants was the computational cost of photonic simulation. The time required to simulate quantum interferometers constrained the exploration of larger architectures and datasets. Consequently, no “large-system” experiments were attempted. Yet, even within these constraints, several photonic approaches achieved performances close to classical baselines—an encouraging sign that small-scale quantum circuits can already encode nontrivial structure. This suggests that with dedicated hardware acceleration and improved simulation tools, progress could follow the same trajectory as classical AI, which took more than a decade to master MNIST but advanced rapidly once reproducible pipelines became standard.

Moreover, while no quantum advantage has been demonstrated, some of the results point to directions worth further exploration. Certain methods exhibited promising behaviors, such as indications that performance might improve with more photons that comparable accuracy could be achieved with sig-

nificantly fewer parameters, or that training could converge faster in specific setups. These preliminary signs of potential merit systematic investigation through larger-scale experiments and hardware-based studies, which we leave for future work.

Looking ahead, the Challenge results emphasize the importance of systematic discovery over serendipitous breakthroughs. Faster simulators, standardized datasets, and accessible hardware backends could collectively accelerate iteration cycles, enabling a more data-driven exploration of photonic architectures. Extending future challenges to larger or more diverse datasets, or including dedicated hardware tracks, would help bridge the simulation–experiment gap and provide more realistic performance estimates. Notably, two of the approaches presented here have already led to independent scientific publications³ [36,37] and two additional works are currently under submission. This outcome further attests to the scientific value and lasting impact of this collective effort.

6.4 Conclusion remarks

In summary, the Perceval Challenge did not uncover a heuristic quantum advantage—but it has achieved something arguably more fundamental: it has mapped the baseline landscape of photonic machine learning and established the infrastructure for cumulative progress. The field now benefits from open, reproducible implementations spanning a range of hybrid and hardware-native paradigms. Together, these results suggest that quantum photonics can meaningfully contribute to learning tasks, provided it is integrated into hybrid pipelines and studied with methodological rigor. Echoing Schuld’s perspective, the key question shifts from “*Can quantum models outperform classical ones?*” to “*How might quantum systems enrich the process of learning itself?*”. The challenge outcomes point toward a phase of convergence—between physics-based and data-driven paradigms—where photonic computing, guided by reproducible methodology and community collaboration, could accelerate the next generation of hybrid intelligence.

7 Acknowledgments

This work has been co-funded by the UFOQO Project financed by the French State as part of France 2030.

8 Author contributions

The Quandela team wishes to thank all participants of the challenge for their active involvement and contributions to the project. Team contributions are detailed below:

- Y. Xie formed the Quantum Tree team and developed the surrogate approach described in Section 5.1.4. He was the winner of the challenge;
- P. Yang formed the Quantum Nomad team and developed the photonic transfer learning approach described in 5.3.1. He ranked second in our challenge;
- O. Zouhry and I. Mejdoub formed the Solal team and developed the feature engineering approach described in Sections 5.2.2 and 5.2.2. They obtained the third place in our challenge;
- A. Sharma, E.Y. Balaji and S.P. Pawar formed the Qubiteers team and developed the convolutional kernel described in Section 5.1.6. They received a special prize for their findings;
- K.C. Chen and Chen-Yu Liu formed the QTX team and developed the distributed approach described in Section 5.1.8;
- V. Deumier formed the Lancelot team and developed the unitary dilation approach described in Section 5.1.2;
- C. Marullo, G. Massafra, D.J. Kenne, A.K. Gupta, N. Reinaldet, G. Intoccia and V. Schiano Di Cola formed the Quantum Naples team and developed the feature annotation approach described in Section 5.2.1;
- D. Kolesnyk and Y. Vodovozova formed the Qool team and developed the photonic kernel in Section 5.1.1;

Additionally,

³one accepted to a major conference and another released as an open preprint on arXiv

- C. Notton drafted the manuscript and assembled all code in the repository;
- V. Apostolou was a member of the Quandela CodeQalibur team and helped develop the approaches in Sections 5.1.3 and 5.3.2. He helped revised this manuscript;
- A. Senellart was a member of the Quandela CodeQalibur team and helped developed the approaches in Sections 5.1.3 and 5.3.2;
- D. Wang and A. Walsh were members of the Quandela QLOQroaches team and developed the photonic QCNN presented in Section 5.1.5;
- R. Mezher was a member of the Quandela CodeQalibur team and revised this manuscript;
- P.E. Emeriau was a mentor in the challenge and revised this manuscript;
- A. Salavrakos contributed to the writing and organization of this manuscript;
- J. Senellart conceived and supervised the challenge, and guided the overall logic and narrative of the manuscript, notably in the introduction and discussion sections.

References

- [1] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.
- [2] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, July 2014.
- [3] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, August 2018.
- [4] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, August 2021.
- [5] Maria Schuld. Supervised quantum machine learning models are kernel methods, 2021.
- [6] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019.
- [7] Nathan Wiebe. Key questions for the quantum machine learner to ask themselves. *New Journal of Physics*, 22(9):091001, sep 2020.
- [8] Maria Schuld and Nathan Killoran. Is quantum advantage the right goal for quantum machine learning? *PRX Quantum*, 3:030101, Jul 2022.
- [9] Joseph Bowles, Shah Nawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059*, 2024.
- [10] Armando Angrisani, Alexander Schmidhuber, Manuel S Rudolph, M Cerezo, Zoë Holmes, and Hsin-Yuan Huang. Classically estimating observables of noiseless quantum circuits. *arXiv preprint arXiv:2409.01706*, 2024.
- [11] Tobias Fellner, David Kreplin, Samuel Tovey, and Christian Holm. Quantum vs. classical: A comprehensive benchmark study for predicting time series with variational quantum machine learning, 2025.
- [12] Brian Coyle, Maxwell Henderson, Justin Chan Jin Le, Niraj Kumar, Marco Paini, and Elham Kashefi. Quantum versus classical generative modelling in finance. *Quantum Science and Technology*, 6(2):024013, apr 2021.
- [13] Boris Albrecht, Constantin Dalyac, Lucas Leclerc, Luis Ortiz-Gutiérrez, Slimane Thabet, Mauro D’Arcangelo, Julia R. K. Cline, Vincent E. Elfving, Lucas Lassablière, Henrique Silvério, Bruno Ximenez, Louis-Paul Henry, Adrien Signoles, and Loïc Henriët. Quantum feature maps for graph machine learning on a neutral atom quantum processor. *Physical Review A*, 107(4), April 2023.

- [14] Milan Kornjača, Hong-Ye Hu, Chen Zhao, Jonathan Wurtz, Phillip Weinberg, Majd Hamdan, Andrii Zhdanov, Sergio H. Cantu, Hengyun Zhou, Rodrigo Araiza Bravo, Kevin Bagnall, James I. Basham, Joseph Campo, Adam Choukri, Robert DeAngelo, Paige Frederick, David Haines, Julian Hammett, Ning Hsu, Ming-Guang Hu, Florian Huber, Paul Niklas Jepsen, Ningyuan Jia, Thomas Karolyshyn, Minho Kwon, John Long, Jonathan Lopatin, Alexander Lukin, Tommaso Macrì, Ognjen Marković, Luis A. Martínez-Martínez, Xianmei Meng, Evgeny Ostroumov, David Paquette, John Robinson, Pedro Sales Rodriguez, Anshuman Singh, Nandan Sinha, Henry Thoreen, Noel Wan, Daniel Waxman-Lenz, Tak Wong, Kai-Hsin Wu, Pedro L. S. Lopes, Yuval Boger, Nathan Gemelke, Takuya Kitagawa, Alexander Keesling, Xun Gao, Alexei Bylinskii, Susanne F. Yelin, Fangli Liu, and Sheng-Tao Wang. Large-scale quantum reservoir learning with an analog quantum computer, 2024.
- [15] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe. Training of quantum circuits on a hybrid quantum computer. *Science Advances*, 5(10):eaaw9918, 2019.
- [16] Teppei Suzuki, Takashi Hasebe, and Tsubasa Miyazaki. Quantum support vector machines for classification and regression on a trapped-ion quantum computer. *Quantum Machine Intelligence*, 6(1):31, 2024.
- [17] V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund, S. Wölk, H. J. Briegel, and P. Walther. Experimental quantum speed-up in reinforcement learning agents. *Nature*, 591(7849):229–233, March 2021.
- [18] 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), July 2014.
- [19] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. *Physical Review X*, 8(3), July 2018.
- [20] James Mills and Rawad Mezher. Mitigating photon loss in linear optical quantum circuits: classical postprocessing methods outperforming postselection. *arXiv preprint arXiv:2405.02278*, 2024.
- [21] Nicolas Maring, Andreas Fyrrillas, Mathias Pont, Edouard Ivanov, Petr Stepanov, Nico Margaria, William Hease, Anton Pishchagin, Thi Huong Au, Sébastien Boissier, Eric Bertasi, Aurélien Baert, Mario Valdivia, Marie Billard, Ozan Acar, Alexandre Brioussel, Rawad Mezher, Stephen C. Wein, Alexia Salavrakos, Patrick Sinnott, Dario A. Fioretto, Pierre-Emmanuel Emeriau, Nadia Belabas, Shane Mansfield, Pascale Senellart, Jean Senellart, and Niccolò Somaschi. A versatile single-photon-based quantum computing platform. *Nature Photonics*, 2024.
- [22] Nicolas Heurtel, Andreas Fyrrillas, Grégoire de Gliniasty, Raphaël Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, Luka Music, Nadia Belabas, Benoît Valiron, Pascale Senellart, Shane Mansfield, and Jean Senellart. Perceval: A software platform for discrete variable photonic quantum computing. *Quantum*, 7:931, February 2023.
- [23] Beng Yee Gan, Daniel Leykam, and Dimitris G Angelakis. Fock state-enhanced expressivity of quantum machine learning models. *EPJ Quantum Technology*, 9(1):16, 2022.
- [24] Alexia Salavrakos, Nicolas Maring, Pierre-Emmanuel Emeriau, and Shane Mansfield. Photon-native quantum algorithms. *Materials for Quantum Technology*, 5(2):023001, apr 2025.
- [25] Rawad Mezher, Ana Filipa Carvalho, and Shane Mansfield. Solving graph problems with single photons and linear optics. *Physical Review A*, 108(3):032405, 2023.
- [26] Tigran Sedrakyan and Alexia Salavrakos. Photonic quantum generative adversarial networks for classical data. *Optica Quantum*, 2(6):458–467, Dec 2024.
- [27] Alexia Salavrakos, Tigran Sedrakyan, James Mills, Shane Mansfield, and Rawad Mezher. Error-mitigated photonic quantum circuit born machine. *Physical Review A*, 111(3), March 2025.

- [28] Zhenghao Yin, Iris Agresti, Giovanni de Felice, Douglas Brown, Alexis Toumi, Ciro Pentangelo, Simone Piacentini, Andrea Crespi, Francesco Ceccarelli, Roberto Osellame, Bob Coecke, and Philip Walther. Experimental quantum-enhanced kernels on a photonic processor. *arXiv preprint arXiv:2407.20364*, 2024.
- [29] Francesco Hoch, Eugenio Caruccio, Giovanni Rodari, Tommaso Francalanci, Alessia Suprano, Taira Giordani, Gonzalo Carvacho, Nicolò Spagnolo, Seid Koudia, Massimiliano Proietti, Carlo Liorni, Filippo Cerocchi, Riccardo Albiero, Niki Di Giano, Marco Gardina, Francesco Ceccarelli, Giacomo Corrielli, Ulysse Chabaud, Roberto Osellame, Massimiliano Dispenza, and Fabio Sciarrino. Quantum machine learning with adaptive boson sampling via post-selection. *Nature Communications*, 16(1), January 2025.
- [30] Liam Lysaght, Timothée Goubault, Patrick Sinnott, Shane Mansfield, and Pierre-Emmanuel Emeriau. Quantum circuit compression using qubit logic on qudits, 2024.
- [31] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6, New York, August 2007. ACM.
- [32] Kaggle. Kaggle: Your machine learning and data science community. <https://www.kaggle.com/>, 2025.
- [33] Quandela and Scaleway. The first perceval quest (hybrid ai-quantum challenge). <https://github.com/Quandela/HybridAIQuantum-Challenge>.
- [34] Airbus and BMW. Quantum computing challenge 2024. <https://www.airbus.com/en/innovation/digital-transformation/quantum-technologies/airbus-and-bmw-quantum-computing-challenge>, 2024.
- [35] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [36] Kuan-Cheng Chen, Chen-Yu Liu, Yu Shang, Felix Burt, and Kin K Leung. Distributed quantum neural networks on distributed photonic quantum computing. *arXiv preprint arXiv:2505.08474*, 2025.
- [37] Yichen Xie. Quantum surrogate-driven image classifier: A gradient-free approach to avoid barren plateaus. *arXiv preprint arXiv:2505.05249*, 2025.
- [38] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2(3):5, 2022.
- [39] Tak Hur, Leeseok Kim, and Daniel K Park. Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, 4(1):3, 2022.
- [40] Scaleway. Scaleway’s Quantum-as-a-Service platform. <https://labs.scaleway.com/en/qaas/>.
- [41] Quandela. Quandela’s cloud platform. <https://cloud.quandela.com>.
- [42] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [43] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [44] Ilya Sutskever and Geoffrey E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1017–1024, 2011.
- [45] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. jaderberg2015spatial. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.
- [46] Tijmen Tieleman. affNIST: a modified version of MNIST. <http://www.cs.toronto.edu/~tijmen/affNIST/>, 2013.

- [47] Norman Mu and Justin Gilmer. MNIST-C: A testbed for robustness. In *arXiv preprint arXiv:1906.02337*, 2019.
- [48] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [49] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. Emnist: Extending mnist to handwritten letters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [50] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.
- [51] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.
- [52] Akira Sakurai, Kyo Inoue, Yoshihisa Yamamoto, and Shuntaro Takeda. Quantum optical reservoir computing powered by boson sampling. *Optica Quantum*, 3(3):238–249, 2025.
- [53] Michael Reck, Anton Zeilinger, Herbert J Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical review letters*, 73(1):58, 1994.
- [54] William R Clements, Peter C Humphreys, Benjamin J Metcalf, W Steven Kolthammer, and Ian A Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.
- [55] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011.
- [56] Andreas Fyrillas, Olivier Faure, Nicolas Maring, Jean Senellart, and Nadia Belabas. Scalable machine learning-assisted clear-box characterization for optimally controlled photonic circuits. *Optica*, 11(3):427–436, 2024.
- [57] Herbert John Ryser. *Combinatorial mathematics*, volume 14. American Mathematical Soc., 1963.
- [58] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
- [59] Chao Ding, Shi Wang, Yaonan Wang, and Weibo Gao. Quantum machine learning for multiclass classification beyond kernel methods. *Physical Review A*, 111(6):062410, 2025.
- [60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [61] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [62] Axel Pappalardo, Pierre-Emmanuel Emeriau, Giovanni de Felice, Brian Ventura, Hugo Jaunin, Richie Yeung, Bob Coecke, and Shane Mansfield. Photonic parameter-shift rule: Enabling gradient computation for photonic quantum computers. *Physical Review A*, 111(3), March 2025.
- [63] Jarrod R McClean, Annabelle Bohrdt, George S Barron, and et al. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.
- [64] Léo Monbroussou, Eliott Z. Mamon, Hugo Thomas, Verena Yacoub, Ulysse Chabaud, and Elham Kashefi. Towards quantum advantage with photonic state injection, October 2024.
- [65] Léo Monbroussou, Jonas Landman, Letao Wang, Alex B Grilo, and Elham Kashefi. Subspace preserving quantum convolutional neural network architectures. *Quantum Science and Technology*, 10(2):025050, March 2025.
- [66] Léo Monbroussou, Beatrice Polacchi, Verena Yacoub, Eugenio Caruccio, Giovanni Rodari, Francesco Hoch, Gonzalo Carvacho, Nicolò Spagnolo, Taira Giordani, Mattia Bossi, Abhiram Rajan, Niki Di Giano, Riccardo Albiero, Francesco Ceccarelli, Roberto Osellame, Elham Kashefi, and Fabio Sciarino. Photonic quantum convolutional neural networks with adaptive state injection, 2025.

- [67] Shangshang Shi, Zhimin Wang, Ruimin Shang, Yanan Li, Jiaxin Li, Guoqiang Zhong, and Yongjian Gu. Hybrid quantum-classical convolutional neural network for phytoplankton classification. *Frontiers in Marine Science*, 10:1158548, 2023.
- [68] Chen-Yu Liu, Chu-Hsuan Abraham Lin, and Kuan-Cheng Chen. Quantum-train with tensor network mapping model and distributed circuit ansatz. *arXiv preprint arXiv:2409.06992*, 2024.
- [69] Beng Yee Gan, Daniel Leykam, and Dimitris G. Angelakis. Fock state-enhanced expressivity of quantum machine learning models. *EPJ Quantum Technology*, 9(1):1–23, December 2022.
- [70] Nam Nguyen and Kwang-Cheng Chen. Quantum embedding search for quantum machine learning. *IEEE Access*, 10:41444–41456, 2022.
- [71] Andrea Mari, Thomas R Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4:340, 2020.
- [72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [73] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.
- [74] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.
- [75] Ben Jaderberg, Lewis W Anderson, Weidi Xie, Samuel Albanie, Martin Kiffner, and Dieter Jaksch. Quantum self-supervised learning. *Quantum Science and Technology*, 7(3):035005, 2022.
- [76] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.
- [77] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 24(2):394–407, 2019.
- [78] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.
- [79] Yichuan Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.
- [80] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. Slic superpixels. Technical report, Technical report EPFL, 2010.
- [81] Murong Wang, Xiabi Liu, Yixuan Gao, Xiao Ma, and Nouman Q Soomro. Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28–39, 2017.
- [82] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13964–13973, 2020.
- [83] Zhengqin Li and Jiansheng Chen. Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1356–1363, 2015.
- [84] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011.
- [85] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

- [86] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 2002.
- [87] Grégoire De Glinasty, Paul Bagourd, Sébastien Draux, and Boris Bourdoncle. Simple rules for two-photon state preparation with linear optics. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 706–711. IEEE, 2024.
- [88] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.
- [89] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

A A Quantum Kernel method

Implementation details

We now describe our approach to classifying the MNIST dataset using classical and quantum kernels. All images of the dataset are first reduced from their original 784-dimensional pixel representation to the top $m = 20$ principal components via principal component analysis (PCA); each component is then normalized to lie in $[0, 1]$ and rescaled by a factor of π to form the feature vector $\vec{\varphi} \in [0, \pi]^m$. For the classical baseline, we train standard SVMs on $N_{\text{train}} = 600$ examples and validate on $N_{\text{val}} = 60$, exploring linear, polynomial, and sigmoid kernels. The linear kernel $\kappa(\vec{x}_i, \vec{x}_j) = \langle \vec{x}_i, \vec{x}_j \rangle$ achieves the highest validation accuracy of 90.00%, while the polynomial kernel $\kappa(\vec{x}_i, \vec{x}_j) = (\gamma \langle \vec{x}_i, \vec{x}_j \rangle + c)^d$ and the sigmoid kernel $\kappa(\vec{x}_i, \vec{x}_j) = \tanh(\gamma \langle \vec{x}_i, \vec{x}_j \rangle + c)$ each reach 88.33% under optimal hyperparameter settings determined by a five-fold cross-validation grid search.

B Leveraging the unitary dilation matrix for feature extraction

B.1 Training the UDENN

In this part, we describe how the UDENN is trained in an alternating fashion. During the challenge TensorFlow was used for the classical part and the model was not trainable in an end-to-end manner. Therefore, the model is divided as two subsystems with an optical and classical components as depicted on Figure 22.

Drawing from automatic control theory, our hybrid approach leverages the concept of time-scale separation found in singularly perturbed systems. Just as fast and slow subsystems with well-separated eigenvalues can be controlled independently without destabilizing the composite system, the quantum feature extraction and classical processing components operate on sufficiently different computational scales to permit independent optimization strategies.

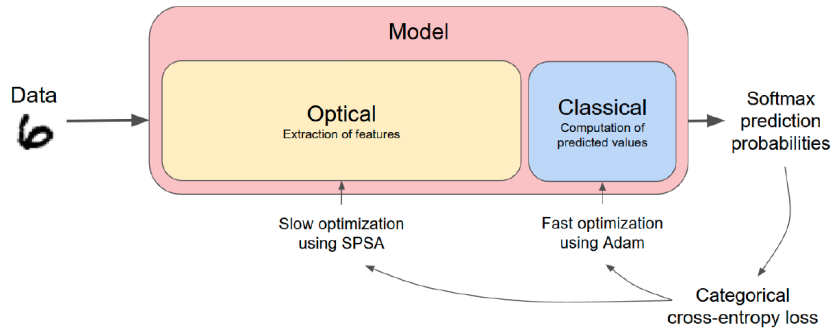


Figure 22: The hybrid model is trained in an alternating fashion

For the UDENN, the training is performed as follows: for each batch of training data, a full optimization step of the classical parameters is performed while a slight update of the optical parameters is performed in order to ensure the convergence of the whole model.

For the optimization of the optical parameters, the stochastic sub-gradient method Simultaneous perturbation stochastic approximation (SPSA) [86] is used. In the results presented in Table 3, the model was trained for 5 epochs.

B.2 Discussion about the results

It is important to note that the optical component of the hybrid model likely operates below its full potential due to limited parameter optimization. The SPSA algorithm, while suitable for derivative-free optimization in quantum systems, requires numerous iterations to achieve convergence due to its stochastic nature. In our implementation, the optical parameters underwent only a limited number of updates, potentially constraining the model’s representational capacity. Future work should explore more efficient optimization strategies, such as implementing classical gradient-based methods where feasible, to fully realize the quantum component’s learning potential.

C A Photonic Quantum Neural Network

In this approach, the goal is to implement a quantum NN. We want to create feature maps that map nonlinear data to a higher dimensional feature space in which a linear decision boundary can be found. Figure 6 presents the overall architecture of the model, inspired by [8, 23].

Firstly, we want to investigate the encoding technique. Therefore, we propose an ablation study to define the best encoding function S that maps the data from the MNIST dataset to the quantum circuit. Secondly, we investigate on the necessity of repeating this encoding L times. For these first studies, we will fix the number of modes m of the circuit.

C.1 The encoding strategy

In this section $\vec{x} = (x_1, x_2, \dots, x_d)$ is the data we want to encode in our circuit. Firstly, we want to know what kind of data we want to encode in our circuit: should we encode the raw normalized data, partial data or PCA components?

In these experiments, $L = 1$ and $m = 10$. For different types of input (all of the images or PCA with $n_{components}$, we vary the encoding strategy:

1. A phase/angular embedding: $\forall \vec{x} \in \mathbf{R}^d, S(\vec{x}) = 2\pi\vec{x}$
2. A linear embedding: $\forall \vec{x} \in \mathbf{R}^d, S(\vec{x}) = \vec{x}$
3. A learnable scaling embedding: $\forall \vec{x} \in \mathbf{R}^d, S(\vec{x}) = \vec{\lambda} \cdot \vec{x}$ where $\vec{\lambda} \in \mathbf{R}^d$ is the vector of learnable scales.

C.1.1 Phase embedding

This embedding provides a periodic representation of the data and effectively scales the normalized input to cover a full circle in radians. Figure 23 presents the best validation accuracy of the quantum NN with different encoding strategy, compared to the classical baseline. The number of trainable parameters in these models are also given.

Figure 23 presents in red the validation accuracy of the qNN under different encoding strategies. Additionally, from the training curves, we observe that, for $n_{components} > 10$ or the whole image, the model struggles to be trained and, even though the losses decrease slightly, the model plateaus.

C.1.2 Linear embedding

This embedding provides a "cropped" angular representation of the data as it only projects to $[0, 1]$. Figure 23 displays validation accuracy and number of parameters. Figure 23 presents in blue the validation accuracy of the qNN under different encoding strategies. Moreover, from the training curves, we observe that, for $n_{components} > 10$ or the whole image, the model struggles to be trained and, even though the losses decrease slightly, the model plateaus but later in the training than with phase encoding.

C.1.3 Learnable scaling embedding

Here, the model can determine the optimal scaling factor for the given task through the training process. We can write the unitary matrix of the encoding layer such as

$$U_e = \begin{pmatrix} \prod_{k=1, k \pmod m = 1}^{k=784} e^{i\lambda_k x_k} & 0 & \dots & 0 \\ 0 & \prod_{k=1, k \pmod m = 2}^{k=784} e^{i\lambda_k x_k} & \dots & 0 \\ 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \prod_{k=1, k \pmod m = 0}^{k=784} e^{i\lambda_k x_k} \end{pmatrix}$$

Figure 23 presents in purple the validation accuracy of the qNN under different encoding strategies. We observe that these learnable embeddings provides a better way to represent the data on the interferometer.

C.1.4 Frequency of apparition of the data L

Here, we tune the frequency of apparition of the data. Table 12 presents the best validation accuracy for $L \in \{1, 2, 3, 5\}$. From these results, it seems that the data encoding strategy does not benefit from multiple apparition of the data.

Frequency apparition L	Best Val. ACC. (quant.)	#param (quant.)
1	61.95	21294
2	59.81	22238
3	51.15	23182
5	40.98	25070

Table 12: Best validation accuracy results for the quantum NNs with different frequency of apparition of the data

C.1.5 Conclusion

From Figure 23, we conclude that the learnable embedding provides the best accuracy throughout the different encoding strategies. Moreover, there is no benefit in repeating the data : we keep $L = 1$.

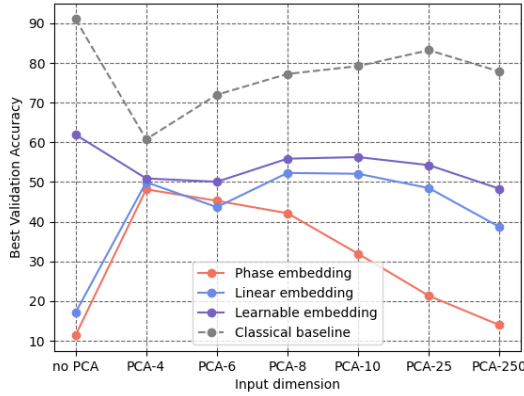


Figure 23: Validation accuracy with respect to the different embedding strategies and input type

C.2 Training the quantum Neural Network

C.2.1 Training pipeline and hyperparameters

Influence of the learning rate: in the following experiment, we study the influence of the learning rate in the dynamic of the training. Previous experiments were done with a learning rate $lr=0.01$. The optimizer used is Adam vanilla: `optimizer = torch.optim.Adam(model.parameters(), lr = lr)`. From Table 13, it seems that the Quantum Layer benefits from a larger learning rate. Figure 24 confirms

lr	Best Val. Accuracy
0.01	61.95
0.1	66.93
0.05	71.45
0.005	52.49
0.001	26.64

Table 13: Validation accuracy based on the learning rate

this observation: with a small learning rate ($lr = 0.001$, $lr = 0.005$), the convergence is very slow and even plateaus for $lr = 0.001$. The best convergence occurs for $lr = 0.05$.

Influence of the weight decay: for Adam optimizer, the weight decay is a regularization technique that aims at preventing overfitting by penalizing large weights. Here, we first use `weight_decay = 0` and then decrease it but the best results are observed with `weight_decay = 0`. One interpretation could be that there is no large gradients or weights to penalize here.

Influence of β_1, β_2 : these are the coefficients used for computing running averages of gradient and its square. The default value is (0.9, 0.999). We obtain better results on this specific validation set using (0.8, 0.9999) and :

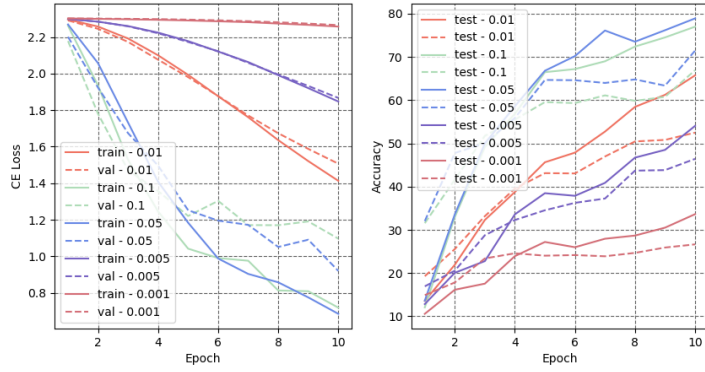


Figure 24: Training and validations losses and accuracies with respect to the learning rate

- a lower β_1 means a reduction of the momentum's influence, which makes the optimizer more responsive to recent gradients by allowing quicker changes.
- a higher β_2 means that we create more stable adaptive learning rates by taking longer history of squared gradients into account and that can prevent aggressive learning rate fluctuations

C.2.2 Influence of the modes and number of photons

Influence of the number of modes: Figure 25 presents the validation accuracy with respect to the number of modes. Overall, increasing the number of modes seems to allow better generalization and therefore better accuracy.

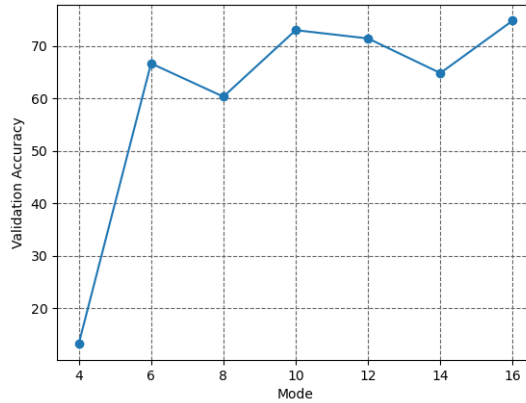


Figure 25: Validation accuracy with respect to number of modes

Influence of the number of photons: Figure 26 shows the validation accuracy with respect to the number of photons for an interferometer with 10 modes. Overall, it seems that more photons, placed "one out of two" from the first mode allows better generalization

C.2.3 Conclusion on the MNIST Dataset

Using the circuit from Figure 6 with $L = 1$, 10 modes, `input_state = [1,0,1,0,1,0,1,0,1,0]`. Table 14 presents the validation accuracy for different sizes of training sets compared to a linear classifier made of 2 linear layers, with similar number of parameters, and a SMV (using `svm.SVC(kernel="linear")` with `scikit-learn`). For better visualization, Figure 27 presents the same results. We observe that the quantum NN does not perform as well as the classical classifiers and needs more training samples to achieve good enough results. Additionally, we can observe the t-SNE (t-distributed Stochastic Neighbor

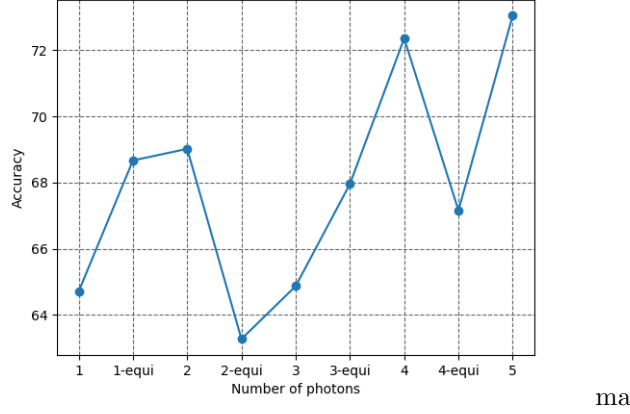


Figure 26: Validation accuracy with respect to number of photons and their entry in the interferometer, N-equi corresponds to N photons equi-placed at the entry, whereas N corresponds to an input state such as $[1,0,1,0,\dots,N,0]$. Here, we use 10 modes.

Embedding) of these classifiers. t-SNE is a dimensionality reduction technique used for visualizing high-dimensional data in 2D or 3D space. Unlike PCA, which preserves global structure, t-SNE emphasizes preserving the local relationships between points, making it particularly effective for visualizing complex datasets where clusters exist. tSNE for the classifiers trained on 5000 samples is shown in Figure 28. We observe that the representations provided by the classical classifier are of higher quality and more discernable than the ones provided by the quantum NN.

Model	Validation ACC	# parameters	Training Samples
quantum NN	24 ± 1.62	21294	50
Linear Layer	58.98 ± 1.07	25450	50
SVM (linear kernel)	58	7850	50
quantum NN	38.39 ± 0.43	21294	100
Linear Layer	74.62 ± 0.23	25450	100
SVM (linear kernel)	71.33	7850	100
quantum NN	52.38 ± 2.53	21294	250
Linear Layer	82.31 ± 0.19	25450	250
SVM (linear kernel)	79.83	7850	250
quantum NN	68.23 ± 1.8	21294	500
Linear Layer	87.59 ± 0.49	25450	500
SVM (linear kernel)	86.5	7850	500
quantum NN	73.39 ± 2.39	21294	1000
Linear Layer	90.94 ± 0.37	25450	1000
SVM (linear kernel)	88.17	7850	1000
quantum NN	77.15 ± 2.28	21294	2500
Linear Layer	90.86 ± 0.25	25450	2500
SVM (linear kernel)	91.33	7850	2500
quantum NN	83.02 ± 1.61	21294	5000
Linear Layer	90.6 ± 0.54	25450	5000
SVM (linear kernel)	91.33	7850	5000

Table 14: Validation accuracy for different sizes of training sets for the quantum NN, a linear classifier and a SVM with linear kernel

Influence of the learned scale embedding: considering that we use the learnable embeddings, we could wonder if the learned scale parameters have meanings for the data. Figure 29 displays the learned parameters (scaled between $[0, 2\pi]$) overlayed on top of validation samples from the MNIST dataset. We do not observe any specific patterns highlighted, but it seems that the model draws more attention to the region at the center of the image.

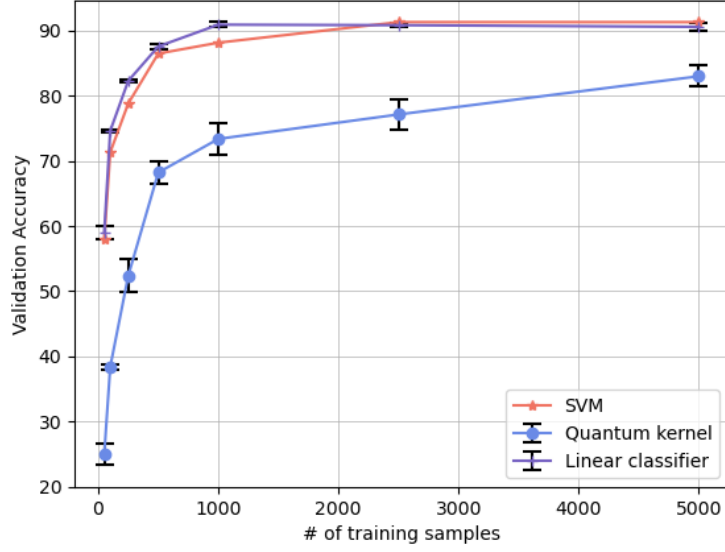


Figure 27: Validation accuracy for different sizes of training sets with different models

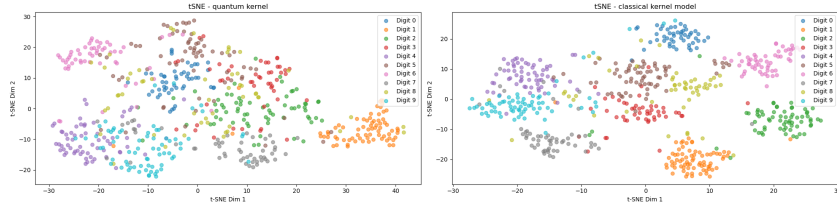


Figure 28: tSNE for the quantum NN and the linear kernel on 5000 training samples. The representations provided by the classical classifier are of higher quality and more discernable (distinct clusters) than the ones provided by the quantum NN

D GLASE: Gradient-free Light-based Adaptive Surrogate Ensemble

D.1 Mathematical background

Recall from Section 4 that, given a photon number state basis $\mathbf{n} = (n_1, \dots, n_M)$, the output probability distribution over multimode measurement outcomes is defined as

$$p_{\mathbf{n}}(\phi) = \frac{|\text{Perm}(U_{\mathbf{n}})|^2}{n_1! \cdots n_M!},$$

where $U_{\mathbf{n}}$ is the submatrix of U corresponding to the detected modes and $\text{Perm}(\cdot)$ denotes the matrix permanent. In practice, due to the exponential cost of computing the full distribution, we compute the expected photon count per mode:

$$\langle \hat{n}_i \rangle = \sum_{\mathbf{n}} n_i \cdot p_{\mathbf{n}}(\phi),$$

which serves as the output signal from the photonic layer. To enable differentiable learning, we introduce a surrogate neural network $g_{\alpha}(\phi)$ trained to approximate this mapping:

$$g_{\alpha}(\phi) \approx \langle \hat{\mathbf{n}} \rangle.$$

D.2 Data encoding process

We adopt a phase-based bosonic embedding approach. Each latent feature vector $\mathbf{z}$ is projected into the interferometer's parameter space using a fixed learnable projection $\phi = \Pi(\mathbf{z})$, where the phase shifts

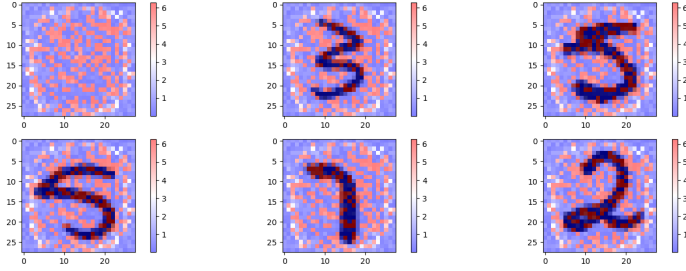


Figure 29: The learned scale parameters in the embeddings applied to different images in the validation set of MNIST

define the transformation matrix $U(\phi)$. This is implemented through Perceval’s GenericInterferometer object. The resulting expectation values of photon numbers in each mode—collected from multi-shot simulations or real QPU executions—serve as the input to a downstream softmax layer for classification.

D.3 Discussion

The experimental behavior of our GLASE architecture reinforces the importance of architectural alignment, surrogate modeling fidelity, and photonic system expressivity in hybrid quantum-classical pipelines. Our results show that surrogate-assisted optimization can effectively stabilize training and outperform purely classical models, but several subtleties emerge when analyzing surrogate interactions and QPU deployment.

One key insight lies in the update frequency of the surrogate model. While frequent updates ensure tighter alignment between the neural surrogate and the true photonic expectation values, they also introduce overhead and potential overfitting to intermediate simulation noise. We observed that updating every 5 epochs provides the best trade-off, too infrequent updates cause performance to degrade, and overly frequent updates yield diminishing returns.

The complexity of the surrogate model is another dimension of the trainability-expressivity trade-off. A deeper surrogate approximates the photonic behavior more accurately, particularly for larger circuits with more photons and modes. However, we found that a 3-layer MLP was sufficient to approximate most photonic behaviors while maintaining computational efficiency. Adding more depth did not yield further performance gains, suggesting it is not necessary for the surrogate to match the full expressivity of the quantum system, only provide a smooth local approximation for backpropagation.

We also investigated the scaling behavior with respect to photons and modes. Intuitively, we believe that increasing the number of modes allows the photonic network to capture higher-dimensional structure in the data, while adding more photons should provide richer interference patterns. Together, this can increase classification performance—especially for datasets like MNIST where digit class boundaries benefit from nonlinear transformations. However, practical constraints in QPU mode count limited real-hardware deployment to 16 modes, which led to a noticeable drop in performance due to noisy sampling and collision effects.

A significant takeaway from our experiments is the importance of preserving alignment between classical feature extractors and the structure of the photonic encoder. The GLASE approach relies on a linear map $\phi = \Pi(\mathbf{z})$ from classical features to phase parameters. While this works well in simulation, misalignment or limited resolution in hardware (e.g., phase discretization or mode crosstalk) can severely degrade performance.

Moreover, our method introduces a secondary optimization loop that must be tuned with care. If the surrogate fails to accurately approximate photon statistics, especially in regions of parameter space far from training samples, it can misguide gradient updates. Fortunately, in our experiments, the surrogate loss consistently correlated with downstream classification loss, providing a reliable signal for updating the front-end encoder.

Hardware deployment revealed another critical bottleneck: postselection. While our simulated circuits assume access to full probability distributions or expectation values, real QPU shots must be interpreted through postselected collision-free events, introducing sampling variance and limiting effective throughput. This challenge, combined with photon loss and phase instability, highlights the need for robust, noise-aware surrogate modeling and potentially hybrid training strategies that alternate between simulated and real data.

Finally, we note that GLASE avoids issues related to gradient-based quantum learning. Because it sidesteps gradient flow through the quantum layer entirely, training is governed solely by the behavior of the surrogate and the classical front-end. This makes it particularly well-suited for noisy intermediate-scale quantum (NISQ) devices, where gradient instability and sampling noise are major obstacles.

In summary, our findings confirm that surrogate-based learning offers a viable and scalable strategy for training photonic quantum neural networks. The surrogate acts not only as a practical tool for gradient approximation but also as a bridge that harmonizes classical learning dynamics with the structure of photonic computation. Future work may explore jointly learned encoding schemes, adaptive surrogate architectures, and the integration of noise models to further improve robustness on real hardware. Additionally, extending this approach to time-resolved photonic systems or continuous-variable encodings may expand its applicability to broader quantum machine learning domains.

E A photonic native quantum convolutional neural network

We provide more details about the photonic QCNN architecture.

E.1 Data encoding

In order to encoding classical 2D-structures of dimension $N_1 \times N_2$, such as greyscale images, we make use of a strategy which uses 2 blocks of respectively N_1 modes and N_2 modes, each containing a single photon. This can be seen as encoding a path-encoded qudit, where:

$$\begin{aligned} |e_0\rangle &= |1, 0, 0, \dots, 0\rangle \\ |e_1\rangle &= |0, 1, 0, \dots, 0\rangle \\ &\vdots \\ |e_{N_1-1}\rangle &= |0, 0, 0, \dots, 1\rangle \end{aligned}$$

The values of the pixel are then encoded using the amplitude of the corresponding state. In addition, since we require the input quantum state to be normalised, we rescale the images such that for an image $\mathbf{x} = (x_{i,j})_{i,j}$:

$$|\psi_{in}\rangle = \sum_{i,j} \frac{x_{i,j}}{\|\mathbf{x}\|_2} |e_i\rangle |e_j\rangle \quad (2)$$

where:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i,j} x_{i,j}^2} \quad (3)$$

Since this is a state containing 2 photons, there exists a probabilistic procedure to prepare the input state using ancilla photons and mode, and heralding [87].

This choice of encoding keeps the local features local and is therefore very useful for image processing tasks. In addition, although we here only focus on 2D structures, this encoding can also easily be extended to arbitrary tensors by simply adding more registers (i.e. qudits) to the input state.

E.2 Convolutional layer

Given the encoding described in the previous section, it is possible to define translational invariant operations on the input data. We first define the kernel size K , which corresponds to the size of the receptive field (for simplicity, we will assume that K is the same in both dimensions). Then, we define two operations U_1 and U_2 on K modes each. The operation U_1 (resp. U_2) are then applied in parallel on $\lfloor \frac{N_1}{K} \rfloor$ (resp. $\lfloor \frac{N_2}{K} \rfloor$) distinct blocks of K modes. We will take these unitaries U_1 and U_2 to be universal interferometers.

These operations are, by design, translation invariant with respect to horizontal and vertical shifts by K pixels (but not arbitrary translations). In fact, a $K \times K$ quantum filter will produce $K \times K$ convolutions acting locally on each patch. This can be seen as follows. Each patch:

$$|\psi\rangle = \sum_{i,j=0}^{K-1} \alpha_{i,j} |e_{n_1 K+i}\rangle |e_{n_2 K+i}\rangle \quad (4)$$

is sent to the new state:

$$U_1 \otimes U_2 |\psi\rangle = \sum_{k,l=0}^{K-1} \sum_{i,j=0}^{K-1} W_{i,j}^{(k,l)} \alpha_{i,j} |e_{n_1 K+k}\rangle |e_{n_2 K+l}\rangle \quad (5)$$

where for each $k, l = 0, \dots, K-1$ the filter $W^{(k,l)}$ is defined as:

$$W_{i,j}^{(k,l)} = (U_1)_{i,k} (U_2)_{j,l} \quad (6)$$

E.3 Pooling layer

The aim of the pooling layer is to reduce the dimension of the image. Therefore, our approach is to use measurements in order to discard some of the mode. In particular, we will decide to measure every other mode, such that the dimension of the image after pooling is halved.

However, if one of the photon is measured, we will leave the encoding space, as one of the register will no longer contain a photon. In order to stay within the encoding space, we will then *adaptively inject a photon* to the corresponding mode whenever a photon is measured. Since there are 2 photons in total in the circuit, there is a maximum of 2 photons that need to be injected during a pooling layer. Then, an adaptive measurement will redirect a photon stores on an ancilla mode to the correct mode whenever a photon is measured.

E.4 Dense layer

The dense layer consists simply of a generic interferometer over all the available modes. It is the only operations (after state preparation) where the photons are allowed to interfere.

F A convolutional layer using a photonic quantum kernel

This section aims at describing the different encoding strategies used in the photonic PQK described in Section 5.1.6, the training parameters conduct to the results presented in Section 5.1.6 and an ablation study on the different hyperparameters.

F.1 Encoding strategy

The PQK acts like a convolutional kernel by processing each $k \times k$ image patch through a photonic interferometer. For a $k \times k$ kernel, $m = \lceil \text{kernel_size}^2 / 2 \rceil$ modes are used. The first m pixels are encoded in a phase shifter on each mode. This first layer of phase shifters is followed by a row of beam splitters and followed by the remaining $m-1$ pixels to be encoded.

This encoding scheme is extended by introducing trainable interference parameters. After the pixel values are encoded, interference between modes is governed by these adjustable parameters. An example of this trainable kernel circuit is provided in Figure 31.

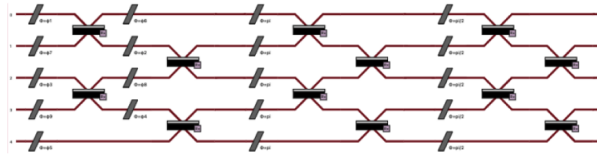


Figure 30: Photonic quantum kernel circuit: Type 2 (delayed) encoding.

F.2 Hybrid architecture components

We provide implementation details about the different components of the hybrid framework:

- **Classical Branch:** Applies standard CNN operations on the raw 28×28 grayscale image, extracting spatial features using convolution, pooling, and ReLU. We have two layers of classical CNN. The first layer has a kernel size of 3×3 with a padding of 0 and outputs 16 filters. The input for this layer is $(1, 28, 28)$ and the output is of the shape $(16, 26, 26)$. The second layer has the kernel

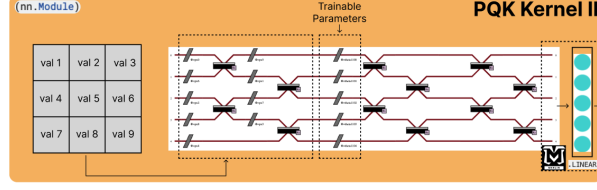


Figure 31: Photonic quantum kernel circuit: Type 2 (delayed) encoding.

of size 5×5 , again with padding 0 and outputs 32 filters. For this layer, the input is (16, 26, 26) and output is (32, 22, 22).

- **Quantum Branch (Non-Trainable):** Applies PQK-based convolutions with stride 2 over 2×2 patches. Each patch yields a 5 or 20 channel feature vector, aggregated into a $14 \times 14 \times N$ quantum feature map. These are passed through one or two small convolutional layers with ReLU to enhance representation. The PQK convolution was accelerated with the help of multi-threading and multiple sessions managed with the help of Scaleway. The input images are pre-processed with the help of these non-trainable kernel convolutions and then given as input to the classical post-NN to get the final class label.
- **Quantum Branch (Trainable Kernel):** We also implement a parallel-branched model where the quantum branch performs the convolution operations of the trainable Type 2 PQK. In this case, the original input images cannot be pre-processed and stored and used later when required. Since the kernel circuit in this case contains trainable parameters, the PQK convolutions need to be applied every time the model is called. To implement this, a custom convolution class is implemented and the kernel circuit can be trained.
- **Fusion:** Classical and quantum outputs are concatenated channel-wise. The combined tensor is flattened and passed through a dense network (128 hidden units, 10 output classes). This fusion allows the network to exploit both standard pixel features and high-order quantum-derived patterns. In a slightly different setup, the outputs from the final classical and PQK layers are concatenated to form tensors of shape (64, 22, 22). This is passed through a classical CNN layer of kernel size 3×3 with a padding of 1. The output from this layer is (32, 22, 22). This tensor is then flattened and then passed through a classical feedforward neural network (FNN) ($15488 \rightarrow 512 \rightarrow 64 \rightarrow 10$). This setup can be seen in Figure 10.

F.3 Training set-up

All models are trained on the subset of MNIST dataset used in this challenge, consisting of 6,000 training and 1,000 test images. Training is performed with mini-batches of size 32, using the Adam optimizer with a learning rate of 10^{-3} and cross-entropy loss. The baseline CNN converges within 20 epochs. Hybrid models train up to 50 epochs with early stopping. For 5-channel PQK embeddings, convergence may require up to 100 epochs due to reduced input dimensionality.

F.4 Training curves

Here, we provide training losses and accuracies for the different types of PQK. First, those of the PQK with two convolutional layers of Type 2 with kernel sizes of 3 and 5, are presented in Figure 32. Then, the curves for the PQK with only one convolutional layer and with a kernel size of 3 are displayed in Figure 33. Then, the curves of the Hybrid model are given in Figure 34. For additional references, the curves of the classical CNN are displayed in Figure 35 and 36.

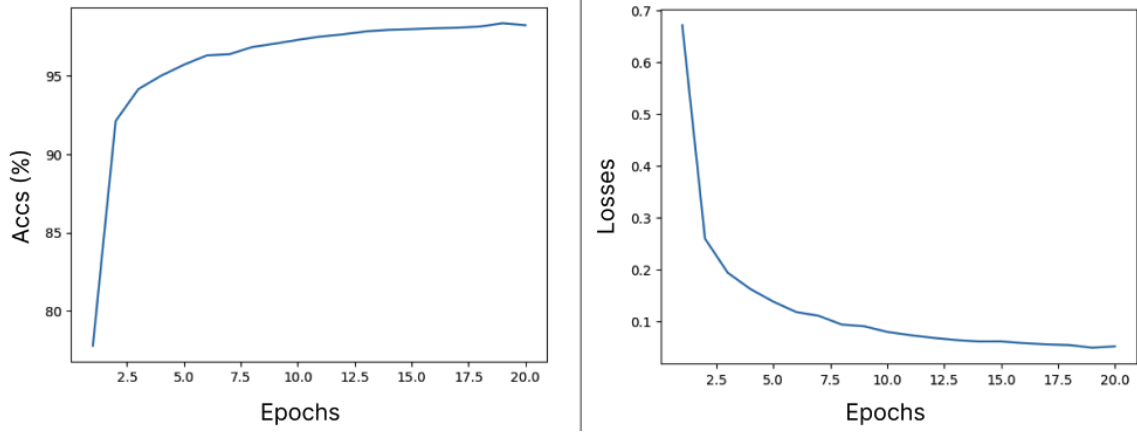


Figure 32: Training accuracies (left) and losses (right) for 20 epochs with the model having two convolution layers of Type 2 PQK, with kernel sizes 3 and 5 respectively.

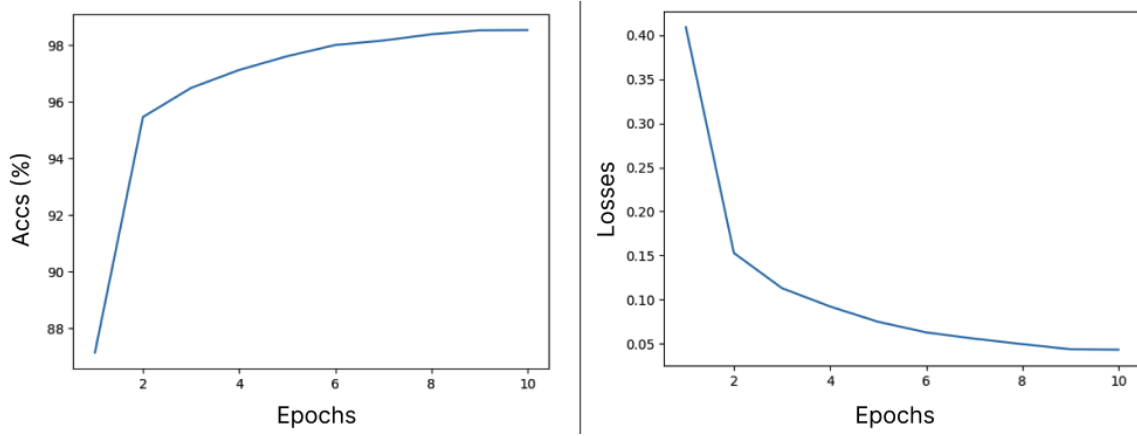


Figure 33: Training accuracies (left) and losses (right) for 10 epochs with the model having one convolution layer of Type 2 PQK, with kernel sizes 3.

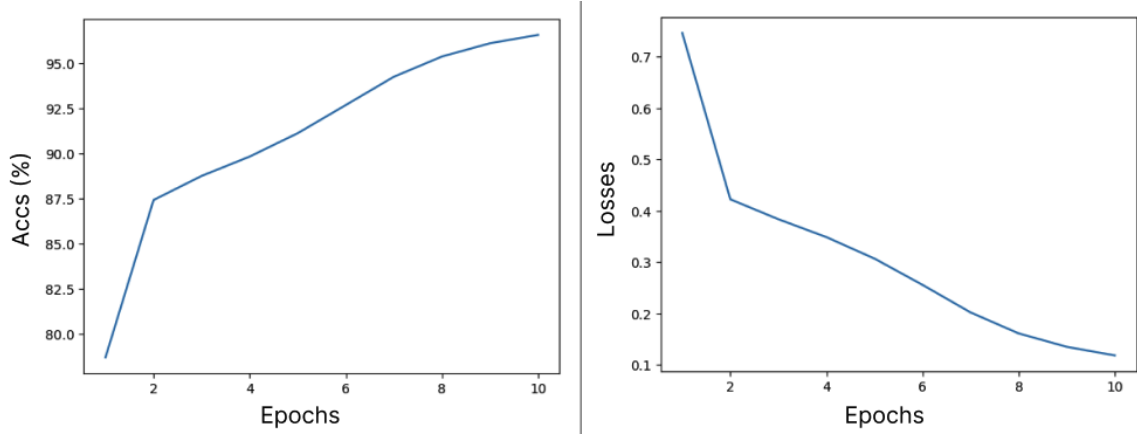


Figure 34: Training accuracies (left) and losses (right) for 10 epochs with classical-quantum parallel channel model; both channels with two convolution layers (Type 2 PQK for quantum) of kernel sizes 3 and 5.

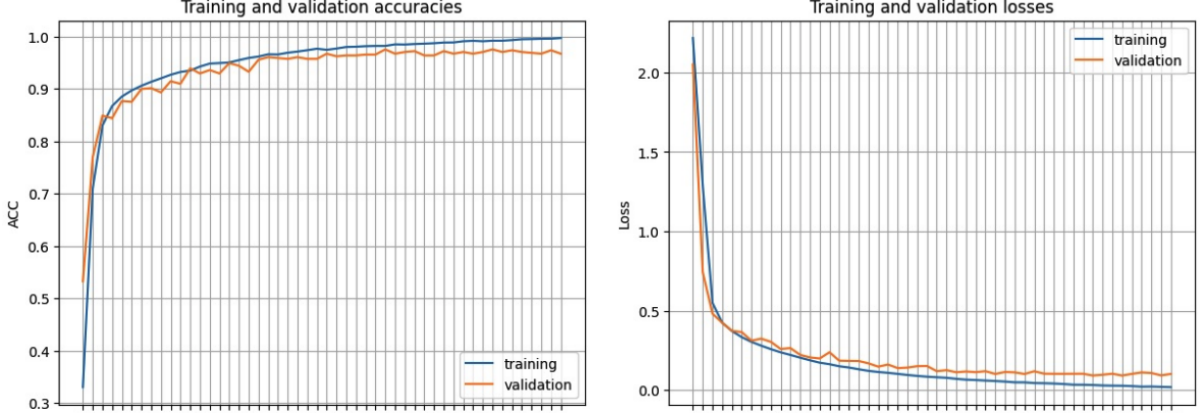


Figure 35: Training and validation metrics for the Classical models with original dataset as input.

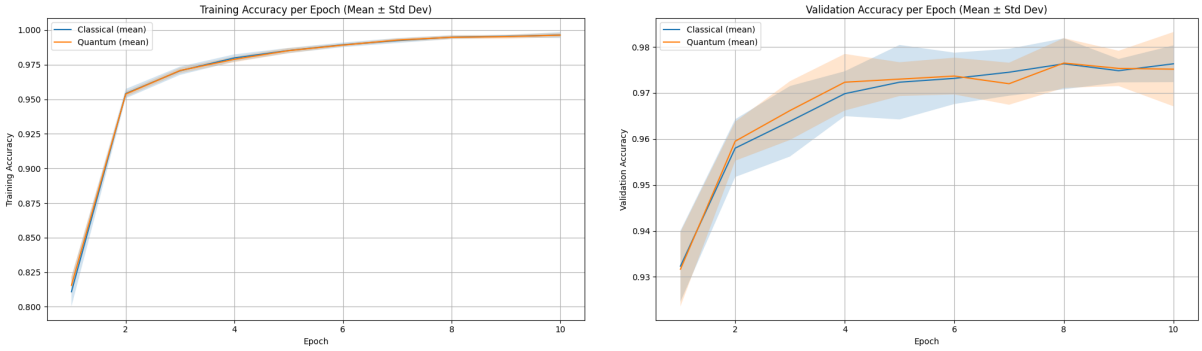


Figure 36: Classical training accuracy over epochs

F.5 Formalization of the Unitary used

This encoding kernel implements delayed encoding of the pixels in the patch. The number of modes m for a kernel of size k is given by:

$$m = \lceil k^2/2 \rceil \quad (7)$$

The encoding of pixels in the Type 2 kernel circuit is not a single layer of PS gate assignments. In this circuit, the encoding is done by alternate layers of PS gate and BS assignments. The encoding layer can be seen as

$$U_{\text{enc}}^{(\text{T2})} = M_{L2} \cdot U_{\text{enc}}^{(2)} \cdot M_{L1} \cdot U_{\text{enc}}^{(1)} \quad (8)$$

where,

$$U_{\text{enc}}^{(1)} = \text{diag}(e^{ip_0}, e^{ip_1}, \dots, e^{ip_{m-1}})_{m \times m}; \quad U_{\text{enc}}^{(2)} = \text{diag}(e^{ip_m}, e^{ip_{m+1}}, \dots, e^{ip_{k^2-1}}, 1, \dots, 1)_{m \times m}; \quad (9)$$

Here, p_k is the k^{th} pixel value from the kernel patch. $U_{\text{enc}}^{(1)}$ and $U_{\text{enc}}^{(2)}$ are the matrices for the first and second layers of pixel encoding.

Finally, for a model with L such layers, the overall PQK unitary transformation becomes:

$$U_{\text{PQK}} = \left(\prod_{\ell=1}^L \left(U_{BS}^{(\ell)} \cdot U_{\text{trainable}}^{(\ell)} \right) \right) \cdot U_{\text{enc}}^{(\text{T2})} \quad (10)$$

We began by varying the quantum feature dimensionality. Comparing 5-channel and 20-channel PQK embeddings under identical training settings revealed that the richer 20-channel representation substantially accelerates learning and improves accuracy. Specifically, the 20-channel models reached 90% validation accuracy within about 15 epochs, while their 5-channel counterparts often required over 80 epochs to achieve a similar level. This confirms that a higher number of quantum-derived features

provides greater expressivity, enabling the classifier to capture finer-grained correlations in the image patches.

Next, we examined the encoding strategies. Entropy measurements of the circuit outputs showed that the encoding yields near-maximal entropy (0.99), indicating noise-like outputs.

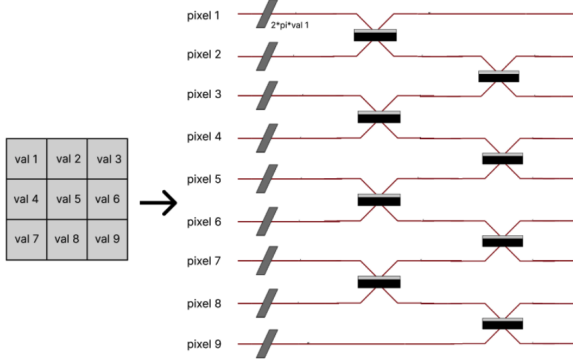


Figure 37: Type 1 PPK circuit: simultaneous encoding (one mode per pixel).

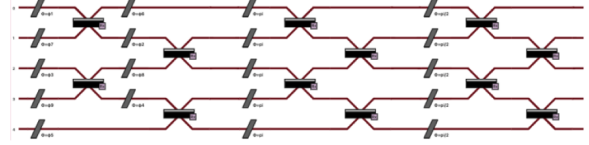


Figure 38: Type 2 PPK circuit: delayed encoding (half the modes at a time).

To assess the necessity of the classical branch, we trained a PPK-only variant by removing the CNN backbone. Despite attaining 98.4% accuracy on the training set, this model collapsed to 67.5% on validation, demonstrating severe overfitting. The drop underscores that quantum features alone lack sufficient structure for generalization and that their integration with classical pixel-based features is essential for robust classification.

We also tested the convolution stride used for PPK scanning. Our default stride-2 configuration processes 196 patches per image, while stride-1 scanning generates 729 overlapping patches. Although stride-1 yields slightly smoother quantum feature maps, it does not offer any meaningful boost in validation accuracy but incurs roughly 3.7 times greater computational cost. As a result, stride-2 remains the optimal choice for balancing performance with efficiency.

Finally, we probed the depth of the PPK interferometer by sweeping the number of beam-splitter layers. Shallow circuits (1–2 layers) underutilize quantum interference and register lower validation accuracy (96.5%) with low output entropy (0.21). In contrast, overly deep circuits (7–8 layers) produce almost random embeddings (entropy 0.99) and also degrade accuracy (97.0%). A middle ground of 3–5 layers achieves both structured entanglement (entropy 0.49) and peak performance (99.0%), confirming that a moderate circuit depth best balances complexity and information preservation.

G A convolutional layer using a photonic feature map

G.1 More details on the feature maps and ansatz

The two-dimension quantum convolution is made of a photonic circuit with two parts:

- a **feature map** to encode the input using a fixed input Fock state and containing beam splitters (*BS*) and phase shifters (*PS*). The parameters of the phase shifters were fixed or used to encode the input. Two architectures were considered: *Achilles*, which is made of a fixed set of *BS* to dispatch 2 photons over all modes, followed by one *PS* on each mode whose angles encode the input pixels x_i using $(x_i - 0.5) \times \frac{\pi}{2}$, and *Odysseus*, made of 27 components ($(BS.H, BS.Ry, \text{ and } PS)$) parametrized by inputs x_i using three variants: $\{-2\pi x, 2\pi x, \sin(2\pi x)\}$
- the **ansatz** consisting of *BS* and *PS* whose parameters are learned during the training. Two architectures were considered here as well: the *Penarddun* architecture consists of a rectangle arrangement of Mach-Zender interferometers with a depth of 6, totalling 48 learnable parameters, and the *Gofanon* architecture which is made up of repeating $BS.H + 2PS$ and $BS.Ry + 2PS$ for a total of 96 variable parameters.

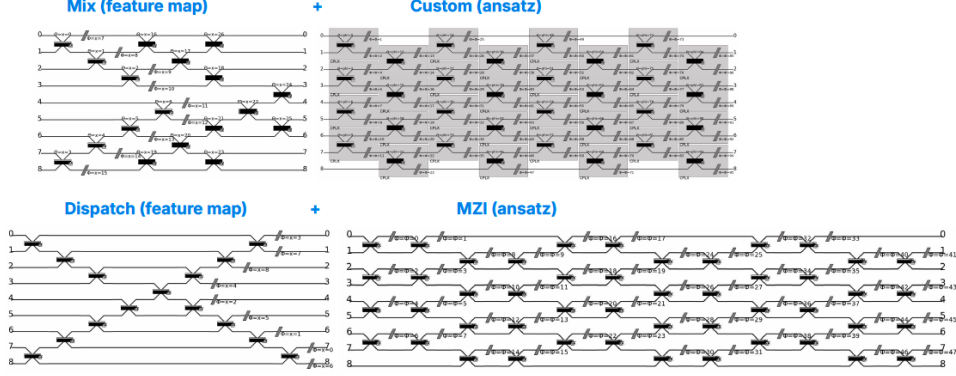


Figure 39: Views of the photonic circuit for feature maps + ansatz

G.2 Study of the output mapping

Two output mapping methodologies were employed to interface the quantum convolution layer with classical processing stages. In both mapping strategies, the quantum layer produces m matrices related to the photon number distributions over Fock state. The first mapping strategy implements maximum likelihood estimation by selecting the most probable Fock state (seen as an m -dimensional vector) for each image patch, generating discrete photon count matrices with integer entries bounded by the number of photons in circuit. The second strategy computes expectation values by performing probability-weighted summation over all Fock states, yielding continuous-valued matrices representing average photon occupancy per mode. From Figure 40, we observe that the first method (argmax) converges faster than the second method in terms of epochs. Moreover, the first method converges faster in terms of training time as well (30 to 50% faster).

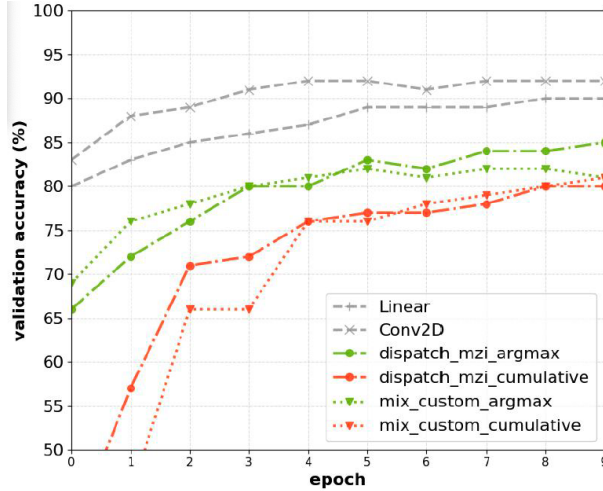


Figure 40: Comparison of training dynamics for the different output mapping strategies

The proposed `qconv2d` achieves efficient training with minimal circuit depth while maintaining spatial localization capabilities. A key finding is that effective MNIST classification can be accomplished using compact feature maps and a single quantum ansatz, whereas classical 2D convolution typically requires multiple kernels. This indicates potential representational efficiency gains in the quantum convolution paradigm.

H Photonic Quantum-Train

H.1 Combinatorial capacity of the architecture and mapping strategy

Architecture and combinatorial capacity. Consider a target NN with parameter vector $w_{\text{CNN}} = (w_1, \dots, w_m) \in \mathbb{R}^m$. We instantiate two photonic quantum neural networks (QNNs), $\text{QNN}_1(\vec{\theta}^{(1)})$ and

Hyperparameter	Description	Value
Input size	$(\phi_i\rangle, \langle\phi_i \psi(\vec{\theta}^{(i)})\rangle ^2)$ features [68]	$\lceil \log_2 m \rceil + 1$
Bond dimension	MPS internal dimension	1–10

Table 15: Configuration of the mapping model $G_{\mathbf{v}}$.

$\text{QNN}_2(\vec{\theta}^{(2)})$, with M_1 and M_2 optical modes, respectively. Each device is operated in a fixed-excitation (Hamming-weight) subspace with N_1 and N_2 excitations.⁴ The corresponding numbers of distinct measurement events are $C(M_i, N_i) = \binom{M_i}{N_i}$. Let $P_1 \in \Delta_{C(M_1, N_1)}$ and $P_2 \in \Delta_{C(M_2, N_2)}$ denote the outcome-probability vectors (elements of the probability simplices). We form the joint vector by the Kronecker product

$$P_w \equiv P_1 \otimes P_2 \in \Delta_{C(M_1, N_1) C(M_2, N_2)}, \quad (11)$$

and choose the sector sizes to satisfy

$$C(M_1, N_1) C(M_2, N_2) \geq m. \quad (12)$$

Thus, by steering the QNN controls one can populate at least m effective degrees of freedom for the target NN. In the interferometer meshes used here, the number of continuous controls scales quadratically with the mode count (e.g., $O(M_i^2)$ tunables per device), so that a comparatively small number of quantum parameters governs a combinatorially large set of probabilities.

Mapping probabilities to real-valued weights. Because P_w is supported on $[0, 1]$ and normalized, while $w_{\text{CNN}} \in \mathbb{R}^m$, we introduce a learnable *mapping model* $G_{\mathbf{v}}$ based on a matrix-product state (MPS) [68]:

$$G_{\mathbf{v}} : [0, 1]^{C(M_1, N_1) C(M_2, N_2)} \longrightarrow \mathbb{R}^{C(M_1, N_1) C(M_2, N_2)}. \quad (13)$$

Let Π_m denote the projection onto the first m components. The target parameters are then defined by

$$w_{\text{CNN}} = \Pi_m(G_{\mathbf{v}}(P_w)), \quad (14)$$

i.e., any surplus components beyond m are discarded once the target vector is filled. The task loss $\mathcal{L} = \mathcal{L}(w_{\text{CNN}})$ is evaluated by the classical model, while being implicitly a function of the quantum and mapping parameters $(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \mathbf{v})$. Table 15 shows the configuration of the mapping model.

H.2 Gradient propagation

Gradients via the chain rule. Let $x \in \{\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \mathbf{v}\}$ collectively denote the quantum and mapping parameters. Differentiating $\mathcal{L}$ through the generation pipeline yields

$$\nabla_x \mathcal{L} = \left(\frac{\partial w_{\text{CNN}}}{\partial x} \right)^T \nabla_{w_{\text{CNN}}} \mathcal{L}, \quad (15)$$

where $\partial w_{\text{CNN}} / \partial x$ is the Jacobian capturing the sensitivity of the classical weights to the underlying quantum controls and to the mapping parameters. For hardware execution, the entries involving quantum controls are estimated with parameter-shift rules (and variants) for gates with suitable generators [61, 88].

Parameter updates. With learning rate $\eta > 0$, a first-order update reads

$$\vec{\theta}_{t+1}^{(i)} = \vec{\theta}_t^{(i)} - \eta \nabla_{\vec{\theta}^{(i)}} \mathcal{L}, \quad \mathbf{v}_{t+1} = \mathbf{v}_t - \eta \nabla_{\mathbf{v}} \mathcal{L}. \quad (16)$$

In our implementation, $\mathbf{v}$ is optimized with ADAM, while $\vec{\theta}^{(i)}$ are tuned with COBYLA (derivative-free) when gradients are noisy or costly to evaluate. Figure 13 provides a schematic of the photonic QT pipeline; detailed hyperparameters are given below.

⁴Equivalently, one may view M_i two-level modes measured in the Hamming-weight- N_i sector; this yields the binomial dimension $\binom{M_i}{N_i}$. In photonic number-state language, this corresponds to a hard-core (no-bunching) model.

H.3 More details on the photonic implementation

We implement the photonic QNN with a programmable multi-mode interferometer realized as a rectangular mesh of nearest-neighbour two-mode Mach-Zehnder Interferometers (MZIs), each composed of two balanced beam splitters and internal/external phase shifters. This architecture follows the decomposition of Clements *et al.* [54], which provides an efficient, fully parameterized factorization of any $m \times m$ unitary U into $m(m-1)/2$ two-mode blocks with interleaved single-mode phases, arranged in $O(m)$ layers (linear optical depth).

Formally, write

$$U = \left[\prod_{\ell=L}^1 \left(D_{\text{out}}^{(\ell)} \prod_{(i,j) \in \mathcal{P}_\ell} B_{(i,j)}(\theta_\ell, \phi_\ell) \right) \right] D_{\text{in}}. \quad (17)$$

where $B_{(i,j)}(\theta, \phi)$ acts nontrivially only on modes i and j , D_{in} and $D_{\text{out}}^{(\ell)}$ are diagonal phase shifts, and $\{\mathcal{P}_\ell\}_{\ell=1}^L$ is a sequence of disjoint nearest-neighbour pairs implementing the rectangular mesh. Each two-mode block is an $\text{SU}(2)$ transformation with real angle θ (effective reflectivity) and phase ϕ :

$$B(\theta, \phi) = \begin{pmatrix} \cos \theta & -e^{-i\phi} \sin \theta \\ e^{i\phi} \sin \theta & \cos \theta \end{pmatrix}, \quad (18)$$

and adjacent single-mode phase shifters provide full $\text{U}(2)$ freedom on each pair. The construction uses $m(m-1)/2$ two-mode blocks, guaranteeing universality for any target U at fixed mode count, with an optical depth that scales linearly in m .

Experimental workflow. We initialize m input modes in QNN-specified photonic states (e.g., single-photon Fock states for fixed-excitation sectors). The state then propagates through alternating layers of $B(\theta_\ell, \phi_\ell)$ blocks and diagonal phase shifters in a checkerboard pattern. If $\hat{a}_i^\dagger$ and $\hat{a}_j^\dagger$ create photons in modes i and j , the action of a single two-mode unit in layer ℓ is

$$\begin{pmatrix} \hat{a}_i^\dagger \\ \hat{a}_j^\dagger \end{pmatrix} \rightarrow \begin{pmatrix} \cos \theta_\ell & e^{-i\phi_\ell} \sin \theta_\ell \\ -e^{i\phi_\ell} \sin \theta_\ell & \cos \theta_\ell \end{pmatrix} \begin{pmatrix} \hat{a}_i^\dagger \\ \hat{a}_j^\dagger \end{pmatrix}, \quad (19)$$

followed by mode-local phase shifts. Repeating across all layers realizes the global U *in situ*, enabling arbitrary multi-mode transformations required for QNN training.

H.4 Parameter efficiency in Photonic QT

To evaluate the parameter efficiency of the photonic QT framework, we implement a classification task based on the Quandela challenge using a subset of the MNIST dataset. The baseline target model is a classical convolutional neural network (CNN) comprising $m = 6690$ trainable parameters. Following the quantum parameter generation scheme described previously, we employ two photonic QNNs with configurations $(M_1 = 9, N_1 = 4)$ and $(M_2 = 8, N_2 = 4)$. These yield $C(9, 4) = 126$ and $C(8, 4) = 70$ distinct measurement outcomes, respectively. Thus, the joint space produces $126 \times 70 = 8820$ candidate parameters, from which the first 6690 values are selected to initialize the classical CNN weights.

The total number of trainable quantum parameters is $108 + 84 = 192$, corresponding to the internal degrees of freedom in the two interferometers. Additionally, the matrix product state (MPS) mapping model contributes further trainable parameters, governed by its bond dimension D . We vary the bond dimension from $D = 1$ to $D = 10$ to examine its effect on model performance.

Figure 41 illustrates the training loss and accuracy over 200 epochs for various bond dimensions. The left panel shows that models with higher bond dimensions achieve consistently lower training loss, indicating enhanced expressiveness and optimization. The right panel confirms this trend, as training accuracy improves with increasing bond dimension and saturates at high performance.

The last row of Table 17 summarizes the performance of the reference classical CNN model. It achieves near-perfect training accuracy (99.98%) and high testing accuracy (96.89%) using all 6690 parameters. This establishes a benchmark against which we compare photonic QT and classical compression baselines.

Table 16 reports the performance of the photonic QT framework across bond dimensions $D = 1$ to 10. As D increases, the total number of parameters grows from 223 to 3292. Testing accuracy improves substantially, from 55.8% to 95.5%, approaching the classical baseline. However, the generalization error also increases, reflecting a trade-off between model capacity and overfitting. For example, the lowest bond dimension ($D = 1$) yields the smallest generalization error (0.0219), while $D = 10$ gives the highest (0.2552).

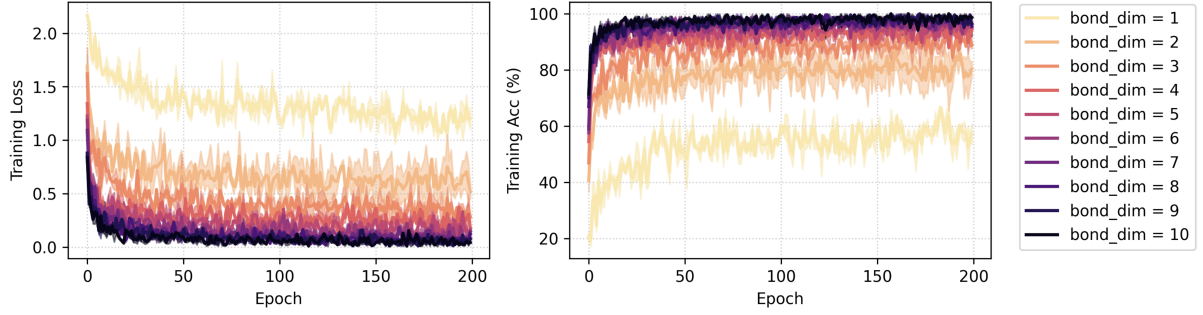


Figure 41: Training loss (left) and accuracy (right) over 200 epochs for various MPS bond dimensions. Higher bond dimensions achieve better optimization and accuracy [36]

# of Parameters	Train Acc. (%)	Test Acc. (%)	Gen. Error
6690	99.983 ± 0.02	96.890 ± 0.31	0.1690 ± 0.005

Table 16: Performance of the original classical CNN.

I Enrich classical CNN representations

To understand what the benefit may be of using a boson-sampler-based embedding, we explored how well it separates data classes in high-dimensional feature space. In our analysis, we observed that photon count distributions resulting from images of different classes tend to be highly distinct — often nearly orthogonal — in the embedding space. Even without any classical training, a simple nearest-centroid classifier based on these distributions could perform well above a random baseline.

This behavior is supported by theory: let U be an $m \times m$ unitary interferometer, and suppose we inject n photons into specified input modes. Recall from Section 4 that the output distribution over Fock states is given by:

$$P_U(\vec{n}) = \frac{|\text{Perm}(U_{\vec{n}})|^2}{n_1! \cdots n_m!}$$

where $U_{\vec{n}}$ is a submatrix of U corresponding to the input/output configuration $\vec{n}$. Variations in image features correspond to variations in the encoded phases in the circuit, which thus define different matrices U . Through the permanent function, this can yield very different output photon-count distributions.

A key question is whether boson-sampling embeddings naturally cluster data by class. Because output probabilities are governed by matrix permanents of interferometer submatrices, even small phase changes (from input features) lead to sharp variations in the photon-count distribution. Intuitively, this should map different classes to nearly orthogonal regions of Fock space.

For each sample x with class label c , let P_x^c denote the corresponding output distribution. We define

Bond Dim.	# Params	Train Acc. (%)	Test Acc. (%)	Gen. Error
1	223	58.26 ± 2.34	55.78 ± 3.27	0.0219 ± 0.007
2	316	83.34 ± 2.77	81.38 ± 2.28	0.0462 ± 0.032
3	471	88.69 ± 1.67	87.06 ± 2.66	0.0364 ± 0.016
4	688	93.92 ± 0.45	93.29 ± 0.62	0.0679 ± 0.002
5	967	95.45 ± 0.39	93.04 ± 0.77	0.0950 ± 0.010
6	1308	96.95 ± 0.02	94.92 ± 0.60	0.1135 ± 0.013
7	1711	97.77 ± 0.22	94.96 ± 0.82	0.1315 ± 0.031
8	2176	97.87 ± 0.78	94.71 ± 0.47	0.1399 ± 0.007
9	2703	98.37 ± 0.12	94.84 ± 0.48	0.1624 ± 0.021
10	3292	98.99 ± 0.34	95.50 ± 0.84	0.2552 ± 0.053

Table 17: Performance of photonic QT with varying MPS bond dimensions [36]

the class-average prototype distribution as

$$\langle P_c \rangle := \frac{1}{|X_c|} \sum_{x \in X_c} P_x^c, \quad (20)$$

where X_c is the set of validation samples belonging to class c . To quantify separability, we first introduce the general Kullback–Leibler (KL) divergence [89] for two discrete distributions P, Q over the same outcome space:

$$KL(P||Q) := \sum_i P(i) \ln \frac{P(i)}{Q(i)}. \quad (21)$$

To avoid singularities due to zero probabilities, we compute a smoothed KL divergence by adding a small constant $\epsilon = 10^{-12}$ to the two probabilities. Using this measure, we define for each sample:

$$KL_{\text{true}} = KL(P_x^c || \langle P_c \rangle), \quad (22)$$

$$KL_{\text{best-wrong}} = \min_{c' \neq c} KL(P_x^c || \langle P_{c'} \rangle), \quad (23)$$

$$KL_{\text{diff}} = KL_{\text{best-wrong}} - KL_{\text{true}}. \quad (24)$$

If $KL_{\text{true}} \ll KL_{\text{best-wrong}}$, then x is closer (in the KL sense) to its own class centroid than to any other. This unsupervised clustering effect suggests that the boson-sampling embedding intrinsically preserves class structure, offering interpretability advantages compared to classical embeddings.

Across $N = 600$ validation samples we computed the following:

- 89.3% satisfied $KL_{\text{true}} < KL_{\text{best-wrong}}$,
- mean $KL_{\text{true}} = 0.63$,
- mean $KL_{\text{best-wrong}} = 4.69$,
- mean margin $KL_{\text{diff}} = 4.06 \pm 3.14$ nats (95% CI [3.80, 4.31]),
- effect size: Cohen’s $d = 1.29$, Wilcoxon $p \approx 3.6 \times 10^{-70}$.

We expand on this further in Figures 42 and 43, together with Table 18. We note that the unsupervised KL nearest centroid accuracy of $\sim 89\%$ is competitive with certain classical kernels of random features applied without supervised training. Nevertheless, separability is not uniform across all classes as shown in Table 18. This suggests that combining the boson-sampling embedding with lightweight supervised fine-tuning could further enhance performance.

Table 18: Per-class KL summary.

class	n_samples	acc_by_KL	mean_KL_true	mean_KL_best_wrong	mean_KL_diff
0	49	0.9388	0.3088	6.3825	6.0737
1	74	0.9730	0.1678	6.1394	5.9716
2	67	0.9254	0.5033	5.0455	4.5422
3	49	0.8980	0.6241	4.1241	3.5001
4	64	0.9375	0.4253	4.1344	3.7091
5	66	0.8485	0.9255	3.9167	2.9913
6	55	0.9636	0.2881	5.7302	5.4422
7	54	0.8148	1.0155	3.9621	2.9466
8	52	0.7692	1.2285	3.7839	2.5554
9	70	0.8429	0.9182	3.7045	2.7863
Global KL-based accuracy:			0.8933		

If $P_x(\mathbf{n})$ is the output distribution for image x , and let $\langle P_c \rangle$ be the average distribution for class c . Then, for a given test input x from class c , we typically found:

$$KL(P_x || \langle P_c \rangle) \ll KL(P_x || \langle P_{c'} \rangle), \quad \forall c' \neq c,$$

which indicates that the quantum embedding clusters inputs of the same class around distinct modes in the output space.

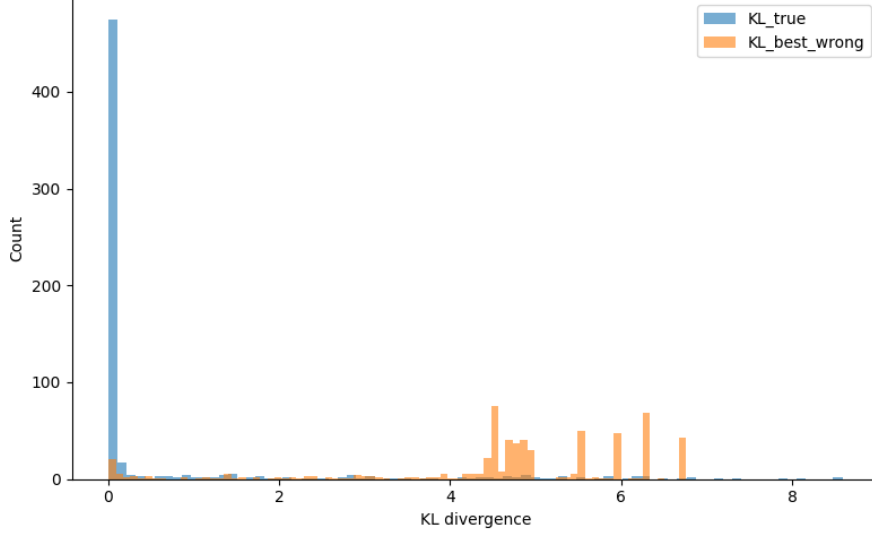


Figure 42: The histogram compares KL_{true} (blue) and $KL_{\text{best-wrong}}$ (orange). The blue distribution is concentrated near zero, clearly left-shifted relative to orange, confirming that samples are consistently closer to their true prototype.

Empirically, we also measured cosine similarities between feature vectors of different classes and found low overlap. This suggests that the boson sampler projects images from different classes into nearly orthogonal directions, a property often desired in kernel methods.

In summary, our empirical findings indicate that the fixed boson sampling embedding offers a powerful mechanism for *unsupervised* class separation, effectively simplifying the task for the downstream classical classifier.

J Hybrid Feature Extractor

J.1 Data Preprocessing

First of all, we process the input MNIST images using Principal Component Analysis (PCA), which allows us to project each image to a smaller dimension d , where $d \leq m$, with m being the number of optical modes. This projection is essential as feeding all of the image features (784 in our case) is not feasible in practice.

Following this PCA, we scale the resulting d -dimensional vector using a min-max normalization. This maps the vector features to a range $[0, 1]$ which meets the requirements of the quantum layer.

J.2 Training set-up

As for the **training set-up**, all experiments were carried out in simulation with a GPU. Each experiment was repeated 25 times to ensure reliable averages.

Each considered architecture was trained on 6000 MNIST images for 10 epochs, while the testing set contains 1000 MNIST images. During the training, parameter updates were conducted using Adam optimizer with categorical cross-entropy loss.

K Transfer Learning

K.1 Feature Encoding Technique

Our amplitude-encoding strategy involves two distinct approaches, designed to test the effectiveness of embedding classical image data into a bosonic quantum system. The main difference between the two is a classical pre-processing step that is applied to the features of the dataset before they are encoded in the quantum circuits.

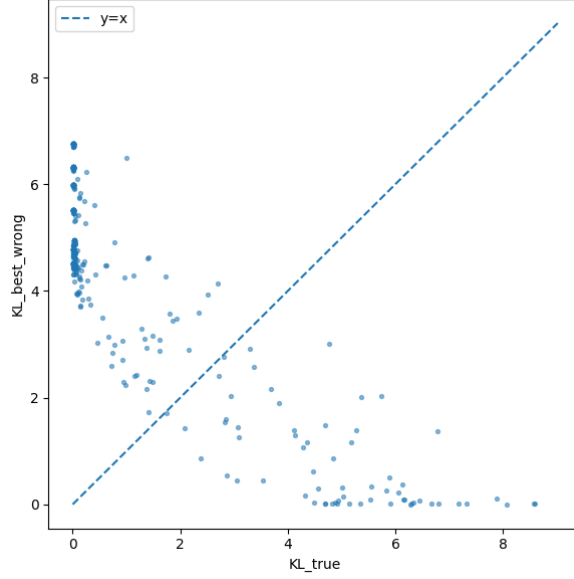


Figure 43: Comparison of the KL-based measures for the validation samples. Most points lie in a region of low KL_{true} and high $KL_{\text{best-wrong}}$.

1. **Classical Linear Encoding.** A linear classical layer is applied on the representation space to transform the representations before injecting them into the bosonic system. This layer is specifically trained to reshape and re-map the input data (e.g., MNIST) to better match the target input space. Unlike feature vectors extracted from pretrained models such as those trained on CIFAR-10, this custom layer adapts to the unique structure and feature distribution of MNIST. This approach attempts to provide a better inductive bias for subsequent classification, yet it still remains fully classical.
2. **Quantum Feature Embedding via Linear Optics.** Classical features are mapped into a quantum photonic state using phase shifters applied to specific optical modes. These phase-encoded modes then propagate through a linear optical network composed of MZIs. The motivation for this strategy is to exploit the natural statistical structure and expressivity of boson samplers. We hypothesized that encoding information in this way would allow the beam splitter network to naturally uncover useful correlations in the representation space, due to its high-dimensional interference pattern. However, due to the linearity and passive nature of the optical circuit, this encoding may not optimally preserve class-separability or inject the necessary nonlinear transformations for effective learning.

Retrospect: The results of our experiments provide a clear and consistent picture of the limitations faced when incorporating static boson sampling layers into a transfer learning pipeline aimed at MNIST digit classification. Across the most difficult transfer learning setups (where the dataset contains 10 classes)—whether using ResNet18 pretrained on ImageNet or CIFAR-10, or using shallow vanilla CNNs—the quantum models with boson sampling consistently underperformed, yielding classification accuracy worse than classical models but significantly better than random guessing. For simpler tasks of binary classification of both distant (e.g., 1 vs 8) and similar (e.g., 3 vs 5) digit classes the performance of the two models (classical or quantum) was comparable with the achieved validation accuracies being identical.

In stark contrast, classical transfer learning models not only achieved accuracies well above chance but often approached or reached 100% on MNIST, even without fine-tuning. This points to a clear expressivity mismatch between classical convolutional features and the fixed transformation implemented by the boson sampling layer.

A central insight from these observations is the trade-off between expressivity and trainability. Classical architectures like ResNet18 have been honed to extract hierarchically rich features from image data. In contrast, boson sampling circuits apply fixed, non-trainable linear optics transformations, which operate in a vastly different representation space based on quantum interference. Without the capacity for

alignment with the learned classical features, these quantum layers often fail to act meaningfully, instead injecting randomness that disrupts rather than enhances classification.

Moreover, reducing the classical architecture to a minimal vanilla CNN further exacerbates this misalignment. With only one or two convolutional layers and a linear head, the feature extraction is significantly limited, making it even less likely that the quantum layer will find anything useful to amplify or transform meaningfully.

Additionally, the effectiveness of the quantum encoding strategy cannot be overstated. If classical-to-quantum encoding fails to preserve the structure embedded in classical features, then the boson sampler effectively operates on noise. In our experiments, both encoding strategies—(1) a linear classical layer to reshape data into a format expected by the quantum model and (2) direct encoding of classical features into optical phases—did not lead to performance improvement. This suggests that the encoding stage plays a critical role and may act as a limiting factor for the performance of the model.

Noise is another concern, especially when considering potential implementation on near-term quantum hardware. Even in simulations, the lack of error correction or regularization mechanisms may lead to performance degradation. While classical models benefit from robust training mechanisms, over-parametrization, and redundancy, quantum models are fragile and prone to performance collapse from small perturbations.

Future work should consider more expressive and adaptive quantum architectures. Trainable quantum circuits could allow gradient-based optimization and better alignment with classical layers. Another promising direction involves improving classical-to-quantum encoding, perhaps by learning the encoding itself via an auxiliary network. Furthermore, analyzing gradient flow across the hybrid model could uncover bottlenecks and suggest architectural changes to improve synergy. Finally, reevaluating the role of the boson sampler—not as a classifier but as a pre-processing feature extractor—may uncover new roles for static quantum optics in machine learning pipelines.