Continuous–Variable Quantum Fourier Neural Operator for Solving Partial Differential Equations

Paolo Marcandelli *, Stefano Mariani, Martina Siena, Stefano Markidis *

Posted Date: 18 May 2026

doi: 10.20944/preprints202605.1199.v1

Keywords: Fourier neural operator; continuous-variable quantum computing; photonic machine learning; neural operator; partial differential equations

 doi:10.20944/preprints202605.1199.v1

Article

Continuous–Variable Quantum Fourier Neural Operator for Solving Partial Differential Equations

Paolo Marcandelli [1,*], Stefano Mariani [1], Martina Siena [1] and Stefano Markidis [2,*]

[1] Department of Civil and Environmental Engineering, Politecnico di Milano, Milan 20133, Italy
[2] Department of Computational Science and Technology, KTH Royal Institute of Technology, Stockholm, Sweden
* Correspondence: paolo.marcandelli@polimi.it (P.M.); markidis@kth.se (S.M.)

Abstract

Fourier Neural Operators have become a central tool for learning solution operators of partial differential equations, but their spectral layers remain entirely classical and rely on digital Fourier processing. In this work, we introduce the Continuous-Variable Quantum Fourier Neural Operator (CV-QFNO), a Gaussian photonic formulation of the FNO spectral layer. The proposed architecture maps the essential operations of Fourier-domain operator learning, Fourier transformation, mode selection, and channel mixing, onto native continuous-variable optical primitives. In this way, CV-QFNO provides a photonic quantum analogue of the truncated spectral mechanism underlying the classical FNO, while avoiding the compilation overhead and spectral mismatch that arise in qubit-based Quantum FNO constructions. We extend the framework to both one- and two-dimensional operator learning and validate it on standard PDE benchmarks, including Burgers' equation, heat equation, Navier–Stokes dynamics, and Darcy flow. The results show that the proposed model preserves the predictive accuracy, resolution generalisation, and spectral inductive bias of Fourier neural operators while using a structurally constrained photonic parameterisation. Since all experiments are performed as classical simulations, the contribution should be understood as an architectural and algorithmic blueprint for photonic neural operators, rather than as a demonstration of quantum computational advantage.

Keywords: fourier neural operator; continuous-variable quantum computing; photonic machine learning; neural operator; partial differential equations

1. Introduction

Partial differential equations (PDEs) are central to the mathematical modelling of physical and engineering systems, but their numerical solution can become computationally demanding in regimes requiring repeated simulations, real-time prediction, uncertainty quantification, or high-dimensional parametric studies. This has motivated the development of learning-based surrogate models, and in particular neural operators, which aim to approximate solution operators between function spaces rather than individual solution instances [1–3]. Among them, the Fourier Neural Operator (FNO) has become a particularly influential architecture for PDE learning, since it parameterises the nonlocal part of the operator in Fourier space, where many PDE dynamics admit a compact spectral representation [2].

The spectral structure of FNO makes it especially relevant from the perspective of quantum computing. Each FNO layer applies a Fourier transform, performs a learnable transformation on a finite set of retained modes, suppresses the remaining modes, and maps the result back to physical space through the inverse Fourier transform. Thus, the FNO spectral layer is not merely a neural network block, but a truncated spectral operator. This structure suggests a natural route toward quantum and photonic implementations, where Fourier transforms and mode-wise operations may be embedded directly into the computational substrate rather than treated as external classical routines.

Qubit-based Quantum Fourier Neural Operators have recently explored this direction by combining unary encodings, Hamming-weight-preserving circuits, and quantum Fourier transforms [4–6].

 doi:10.20944/preprints202605.1199.v1

In this setting, data are represented in fixed-Hamming-weight subspaces and trainable spectral mixing layers are realised through orthogonal transformations implemented by Reconfigurable Beam Splitter (RBS) gates. This formulation has three important advantages. First, the unary-QFT provides a structured quantum implementation of the Fourier transform: while its gate count scales as $O(N_s \log N_s)$, analogously to the classical FFT, its circuit depth can scale as $O(\log N_s)$ under suitable hardware assumptions allowing parallel operations on disjoint qubit pairs. Second, the architecture provides a direct quantum analogue of the Fourier layer by combining Fourier-domain processing with mode-wise trainable transformations. Third, it constrains the trainable spectral weights to orthogonal matrices, reducing the number of parameters with respect to the unconstrained complex tensors used in classical FNOs. Orthogonal parameterisations are also known to improve conditioning, gradient propagation, and training stability in classical and quantum learning models [7–9].

However, the qubit formulation also presents two structural limitations. First, RBS gates are not native operations on standard qubit hardware and must be compiled into elementary one- and two-qubit gates, introducing a substantial entangling-gate overhead. This makes full QFNO circuits difficult to realise on near-term qubit devices. Second, the parallel QFNO construction does not exactly reproduce the truncated spectral action of the classical FNO: while the selected Fourier modes are processed by trainable circuits, the non-selected modes remain present in the quantum state instead of being set to zero [5]. This introduces a systematic spectral mismatch with the classical FNO layer, whose high-frequency modes are explicitly filtered before applying the inverse Fourier transform.

Continuous-variable (CV) quantum computing provides a natural framework to address both issues. In photonic CV systems, information is carried by bosonic modes and processed through Gaussian operations such as beam splitters, phase shifters, squeezers, and loss channels [10–13]. These operations are directly suited to spectral processing: the Cooley–Tukey structure of the Fourier transform can be mapped onto optical networks of beam splitters and phase rotations, while trainable orthogonal mixing can be implemented by passive interferometers. Moreover, unwanted Fourier modes can be physically suppressed by routing them through loss channels with transmissivity $\eta = 0$, thereby recovering the explicit truncation mechanism of the classical FNO.

In this work, we introduce the Continuous-Variable Quantum Fourier Neural Operator (CV-QFNO), a Gaussian photonic formulation of the FNO spectral layer. The construction builds on the continuous-variable quantum Fourier layer introduced in [14], where a full matrix is embedded into the off-diagonal covariance block of a bipartite Gaussian state and Fourier transformations are implemented through optical Gaussian circuits. Here, we extend that framework from a fixed Fourier processing layer to a trainable neural-operator architecture. In the proposed CV-QFNO, Fourier transformation, mode selection, and channel mixing are all mapped onto native CV optical primitives: optical quantum Fourier transforms perform the spectral change of basis, zero-transmissivity loss channels remove non-retained modes, and pyramidal interferometers implement trainable orthogonal mixing across channels.

This formulation should not be interpreted as a claim of quantum computational advantage. All experiments in this work are performed as classical simulations of the corresponding Gaussian transformations. Rather, the contribution is architectural and algorithmic: we show that the continuous-variable photonic formalism provides a structurally faithful implementation of the FNO spectral mechanism, while avoiding the qubit compilation overhead and the spectral mismatch of previous QFNO constructions.

The main contributions of this work are as follows:

- We introduce the CV-QFNO, a Gaussian photonic formulation of the Fourier Neural Operator spectral layer in which Fourier transformation, mode selection, and channel mixing are expressed in terms of native continuous-variable optical operations.
- We show that the parallel qubit QFNO suffers from a systematic spectral mismatch with the classical FNO, since non-selected Fourier modes are left unchanged. The CV-QFNO removes this

 doi:10.20944/preprints202605.1199.v1

mismatch by using loss channels with transmissivity $\eta = 0$, thereby recovering the truncated spectral structure of the classical FNO.
- We develop both one- and two-dimensional CV-QFNO architectures. The one-dimensional construction assigns each retained Fourier mode to an independent optical branch, while the two-dimensional extension introduces cross-circuit interferometric coupling to realise channel mixing across spatial Fourier modes.
- We provide a resource comparison between qubit and continuous-variable formulations, highlighting that the qubit implementation requires a large number of compiled entangling gates, whereas the CV construction is naturally expressed through passive linear-optical elements.
- We benchmark FNO, QFNO, and CV-QFNO on representative PDE operator-learning tasks, including Burgers' equation, heat equation, Navier–Stokes dynamics, and Darcy flow. The results show that CV-QFNO preserves the accuracy and resolution-generalisation behaviour of FNO while using a constrained orthogonal spectral parameterisation.

2. Methods

2.1. Classical Fourier Neural Operator

We briefly recall the classical Fourier Neural Operator (FNO) [2], focusing only on the spectral structure that will serve as the reference target for the qubit and continuous-variable formulations developed below. In the neural-operator framework, the goal is to approximate a solution operator

$$\mathcal{G}^\dagger : \mathcal{A} \to \mathcal{U},$$

mapping input functions to output functions. Given an input $a : D_j \to \mathbb{R}^{d_a}$, the FNO first lifts it pointwise to a latent field $v_t : D_j \to \mathbb{R}^{d_v}$, where t indexes the Fourier layer within the stacked FNO architecture, i.e., v_t denotes the latent field before the t-th Fourier block and v_{t+1} the corresponding output. The latent representation is then updated by Fourier layers of the form

$$v_{t+1}(x) = \sigma\Big(W v_t(x) + \mathcal{F}^{-1}\big(R_\phi \cdot \mathcal{F}(v_t)\big)(x)\Big), \tag{1}$$

here W is a learnable pointwise linear map, σ is a nonlinear activation function, and $\mathcal{F}$, $\mathcal{F}^{-1}$ denote the Fourier and inverse Fourier transforms along the spatial variables. The symbol "$\cdot$" denotes the mode-wise action of the spectral weights on the Fourier coefficients; in discrete form this corresponds to a matrix–vector or tensor contraction over the channel dimension.

We now make Equation (1) explicit in one and two spatial dimensions. This is not intended as a full review of FNOs, but as a preparation for the quantum constructions in Sections 2.2 and 2.3: both architectures operate on matrix or tensor representations of the latent field, and must reproduce the Fourier transform, the mode-wise channel mixing, and the suppression of non-retained modes.

One-dimensional case. We first consider the common one-dimensional structure underlying the Burgers' and heat-equation benchmarks reported in Section 3. Their specific PDE definitions and numerical settings are given there; here we only need the latent representation on which the FNO spectral layer acts. For both 1D benchmarks, the lifting map P promotes the input $a\colon D_j \to \mathbb{R}^{d_a}$ to the latent field $v_t\colon D_j \to \mathbb{R}^{d_v}$, evaluated at each of the N_s grid points $\{x_j\}_{j=1}^{N_s}$. Collecting these evaluations column-wise defines the latent matrix

$$A = \begin{pmatrix} v_t(x_1) & v_t(x_2) & \cdots & v_t(x_{N_s}) \end{pmatrix} \in \mathbb{R}^{N_c \times N_s}, \qquad N_c = d_v, \tag{2}$$

where each column $v_t(x_j) \in \mathbb{R}^{N_c}$ is the channel vector at spatial location x_j. Applying the Fourier transform $\mathcal{F}$ along the spatial direction transforms each row of A, producing

$$\hat{A} = \begin{pmatrix} \hat{v}_t(1) & \hat{v}_t(2) & \cdots & \hat{v}_t(N_s) \end{pmatrix} \in \mathbb{C}^{N_c \times N_s}, \tag{3}$$

 doi:10.20944/preprints202605.1199.v1

where the hat denotes quantities represented in Fourier space. Thus, $\hat{A}$ is the Fourier-space representation of the latent matrix A, obtained by applying $\mathcal{F}$ along the spatial direction, and $\hat{v}_t(k) \in \mathbb{C}^{N_c}$ collects the k-th Fourier coefficient across all channels simultaneously. The learnable operator $R_\phi^{(k)} \in \mathbb{C}^{N_c \times N_c}$ then acts along the channel direction on each retained mode independently,

$$\hat{v}_t(k) \;\mapsto\; R_\phi^{(k)}\, \hat{v}_t(k), \qquad k = 1, \dots, K, \tag{4}$$

while all non-retained modes are set to zero. Since $R_\phi^{(k)}$ is an unconstrained matrix in $\mathbb{C}^{N_c \times N_c}$, writing $R_\phi^{(k)} = \mathrm{Re}(R_\phi^{(k)}) + i\,\mathrm{Im}(R_\phi^{(k)})$, the product $R_\phi^{(k)}\hat{v}_t(k)$ *cross-mixes* the real and imaginary parts of the mode vector:

$$\begin{aligned} \mathrm{Re}\big(R_\phi^{(k)}\hat{v}_t(k)\big) &= \mathrm{Re}(R_\phi^{(k)})\,\mathrm{Re}(\hat{v}_t(k)) - \mathrm{Im}(R_\phi^{(k)})\,\mathrm{Im}(\hat{v}_t(k)), \\ \mathrm{Im}\big(R_\phi^{(k)}\hat{v}_t(k)\big) &= \mathrm{Im}(R_\phi^{(k)})\,\mathrm{Re}(\hat{v}_t(k)) + \mathrm{Re}(R_\phi^{(k)})\,\mathrm{Im}(\hat{v}_t(k)). \end{aligned} \tag{5}$$

The two output components are therefore independent linear combinations of *both* input components, with no structural coupling between $\mathrm{Re}(R_\phi^{(k)})$ and $\mathrm{Im}(R_\phi^{(k)})$. This unconstrained complex parameterisation uses $2N_c^2$ real degrees of freedom per retained mode. The resulting filtered Fourier matrix is

$$\hat{A}' = \begin{pmatrix} R_\phi^{(1)}\hat{v}_t(1) & \cdots & R_\phi^{(K)}\hat{v}_t(K) & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{C}^{N_c \times N_s}, \tag{6}$$

where the last $N_s - K$ columns are zeroed out. Since $\mathcal{F}^{-1}$ acts along the spatial direction, i.e., row-wise across the N_s columns of $\hat{A}'$, it can be applied outside the matrix to recover the output latent matrix,

$$A' = \mathcal{F}^{-1}(\hat{A}') \in \mathbb{R}^{N_c \times N_s}, \qquad A' = \begin{pmatrix} v_{t+1}(x_1) & \cdots & v_{t+1}(x_{N_s}) \end{pmatrix}, \tag{7}$$

whose j-th column $v_{t+1}(x_j)$ is the updated channel vector at spatial location x_j, consistently with Equation (1) evaluated at $x \in D_j \subset \mathbb{R}$.

Two-dimensional case. For two-dimensional operator-learning problems, such as the Navier–Stokes and Darcy-flow benchmarks reported in Section 3, the latent field is evaluated on an $N_x \times N_y$ spatial grid. After lifting, we collect the channel values at each spatial location into a tensor $A \in \mathbb{R}^{N_c \times N_x \times N_y}$, with $N_c = d_v$, defined by

$$A_{:,i,j} = v_t(x_i, y_j) \in \mathbb{R}^{N_c}, \qquad i = 1, \dots, N_x, \quad j = 1, \dots, N_y. \tag{8}$$

Here the colon in $A_{:,i,j}$ denotes all entries along the channel dimension: for each spatial grid point (x_i, y_j), the slice $A_{:,i,j}$ is the N_c-dimensional vector collecting all latent channels at that location. Applying the 2D Fourier transform $\mathcal{F}_{2D}$ along both spatial directions transforms each channel independently, producing

$$\hat{A}_{:,k_1,k_2} = \hat{v}_t(k_1, k_2) \in \mathbb{C}^{N_c}, \qquad k_1 = 1, \dots, N_x, \quad k_2 = 1, \dots, N_y, \tag{9}$$

where $\hat{v}_t(k_1, k_2)$ collects the (k_1, k_2)-th 2D Fourier coefficients across all channels simultaneously. The learnable operator $R_\phi^{(k_1,k_2)} \in \mathbb{C}^{N_c \times N_c}$ then acts along the channel direction on each retained mode pair independently,

$$\hat{v}_t(k_1, k_2) \;\mapsto\; R_\phi^{(k_1,k_2)}\, \hat{v}_t(k_1, k_2), \qquad k_1 = 1, \dots, K_x, \quad k_2 = 1, \dots, K_y, \tag{10}$$

while all non-retained mode pairs are set to zero. As in the 1D case (cf. Equation (5)), the unconstrained complex matrix $R_\phi^{(k_1,k_2)} \in \mathbb{C}^{N_c \times N_c}$ cross-mixes the real and imaginary parts of each retained 2D Fourier

 doi:10.20944/preprints202605.1199.v1

mode vector, with $2N_c^2$ real degrees of freedom per retained mode pair. The resulting filtered Fourier tensor is

$$\hat{A}'_{:,k_1,k_2} = \begin{cases} R_\phi^{(k_1,k_2)}\, \hat{v}_t(k_1,k_2) & k_1 \le K_x,\ k_2 \le K_y, \\ 0 & \text{otherwise}, \end{cases} \tag{11}$$

with $\hat{A}' \in \mathbb{C}^{N_c \times N_x \times N_y}$. Since $\mathcal{F}_{2D}^{-1}$ acts along both spatial directions, it can be applied outside the tensor to recover the output latent field,

$$\begin{aligned} A' &= \mathcal{F}_{2D}^{-1}(\hat{A}') \in \mathbb{R}^{N_c \times N_x \times N_y}, \\ A'_{:,i,j} &= v_{t+1}(x_i, y_j), \end{aligned} \tag{12}$$

whose (i,j)-th slice $v_{t+1}(x_i, y_j)$ is the updated channel vector at spatial location (x_i, y_j), consistently with Equation (1) evaluated at $(x,y) \in D_j \subset \mathbb{R}^2$.

2.2. Quantum FNO: Qubit Formulation

In this section we review the qubit-based Quantum Fourier Neural Operator introduced in [5], which provides the direct precursor to our continuous-variable formulation. We present the core theoretical ingredients of the architecture, namely data encoding, the unary quantum Fourier transform, and the learnable spectral mixing layer, and then examine the two structural limitations that motivate the transition to a continuous-variable framework: the hardware compilation overhead inherent to the qubit setting, and the systematic spectral mismatch with the classical FNO introduced by the parallel circuit variant. The notation for the latent representation depends on the spatial dimensionality of the problem: in one dimension the latent field is collected into a matrix $A \in \mathbb{R}^{N_c \times N_s}$, whereas in two dimensions it is collected into a tensor $A \in \mathbb{R}^{N_c \times N_x \times N_y}$. The following treatment, as presented in [5], focuses on the one-dimensional setting, so that the latent representation is the matrix A defined in Equation (2). For an extension to higher-dimensional domains we refer the reader to [6].

Data encoding. The goal of the encoding step is to embed the latent matrix $A \in \mathbb{R}^{N_c \times N_s}$ defined in Equation (2) into a quantum state, so that subsequent quantum operations reproduce the spectral mixing of Equation (4) at the quantum level. This matrix is encoded as a quantum state in the *unary basis* by constructing the two-register state

$$|A\rangle = \frac{1}{\|A\|} \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} a_{ij}\, |e_j\rangle |e_i\rangle, \tag{13}$$

where $|e_k\rangle$ denotes the k-th unary basis vector (a computational basis state with a single 1 at position k). This encoding requires $N_c + N_s$ qubits and can be loaded with a butterfly circuit with depth $O(\log N_c + 2N_c \log N_s)$ and gate count $(N_c - 1) + (2N_c - 1)(N_s - 1)$. See [5,15] for more information about unary encoding and circuit depth.

Unary Quantum Fourier Transform. The classical FFT butterfly structure is reproduced quantumly via a circuit composed of single-qubit phase gates and parameterised two-qubit Reconfigurable Beam Splitter (RBS) gates [5]. Applied to the lower (N_s-qubit) register of the encoded state $|A\rangle$ in Equation (13), this *Unary-QFT* implements the row-wise discrete Fourier transform of A entirely within the unary subspace. The full two-register action is

$$\mathrm{QFT}\,|A\rangle = \frac{1}{\|A\|} \sum_{i=1}^{N_c} \mathrm{QFT}\left(\sum_{j=1}^{N_s} a_{ij}\, |e_j\rangle \right) |e_i\rangle = \frac{1}{\|A\|} \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} \hat{a}_{ij}\, |e_j\rangle |e_i\rangle =: |\hat{A}\rangle, \tag{14}$$

where $\hat{a}_{ij}$ are the Fourier coefficients of the i-th row of A, so that the quantum state $|\hat{A}\rangle$ encodes the transformed matrix defined in Equation (3). This is the quantum analogue of applying $\mathcal{F}$ row-wise to A along the spatial direction: the channel index i (upper register) is untouched, while the spatial

 doi:10.20944/preprints202605.1199.v1

index j (lower register) is transformed, in direct correspondence with the classical operation $A \mapsto \hat{A}$. The circuit requires $O(N_s \log N_s)$ gates and runs in depth $O(\log N_s)$ when disjoint qubit pairs can be addressed in parallel, as in trapped-ion or cold-atom hardware.

Parallel Quantum Fourier Layer and its limitations The architecture we describe is the Parallel QFL of [5]. After the Unary-QFT has mapped the encoded state $|A\rangle$ to $|\hat{A}\rangle$ (see Equation (14)), the intended operation is to reproduce the filtered spectral matrix of Equation (6). This is done, at the level of the target FNO layer, by applying an independent learnable mixing $R_\phi^{(k)}$ to each of the K retained Fourier modes and setting the remaining $N_s - K$ modes to zero. To this end, K independent quantum circuits are prepared, one for each retained mode $k \in \{1, \ldots, K\}$. In circuit k, a pyramidal interferometer W_Q^k (Figure 1) is applied to the upper (N_c-qubit) register, controlled on the k-th qubit of the lower register: the gate fires only when the lower register is in state $|e_k\rangle$, i.e., only for Fourier mode k.

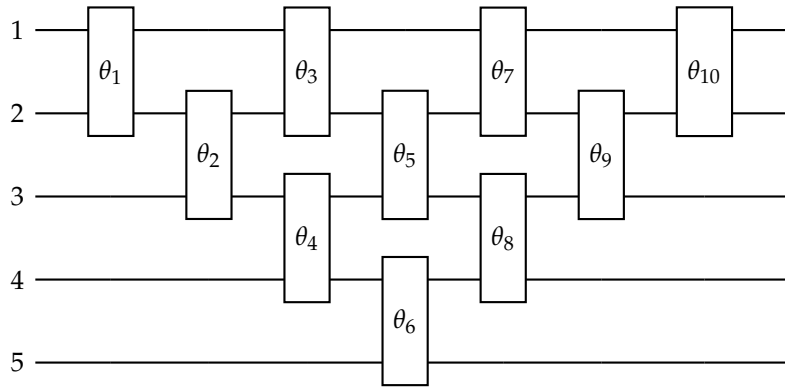


Figure 1. Pyramidal interferometer on $N_c = 5$ modes implementing an arbitrary 5×5 orthogonal matrix in the unary subspace. Each box represents an RBS(θ_k) gate with an independent trainable parameter θ_k. Gates in the same column act on disjoint wire pairs and run in parallel. The circuit uses $\frac{N_c(N_c-1)}{2} = 10$ gates arranged in $2N_c - 3 = 7$ sequential timesteps, with at most $\lfloor N_c/2 \rfloor = 2$ gates per timestep, achieving circuit depth $O(N_c)$.

Since each RBS(θ) gate restricted to the unary subspace acts as a Givens rotation $G_{ij}(\theta) \in O(N_c)$, and any orthogonal matrix admits a decomposition into at most $\frac{N_c(N_c-1)}{2}$ such rotations [16], the pyramidal circuit with one RBS per pair (i, j) provides a universal parameterisation of $O(N_c)$ [17]. The resulting weight matrices $\{W_Q^k\}_{k=1}^K$ are therefore fully learnable orthogonal operators. This is a deliberate architectural choice: by restricting the spectral mixing to $O(N_c) \subset U(N_c)$, the same real matrix W_Q^k is applied independently to the real and imaginary parts of each Fourier mode vector

$$\mathrm{Re}(W_Q^k \hat{v}_t(k)) = W_Q^k \,\mathrm{Re}(\hat{v}_t(k)), \qquad \mathrm{Im}(W_Q^k \hat{v}_t(k)) = W_Q^k \,\mathrm{Im}(\hat{v}_t(k)), \tag{15}$$

without the cross-coupling between real and imaginary components that characterises the classical FNO complex weight (cf. Equation (5)). The orthogonality constraint reduces the parameter count per mode from $2N_c^2$ to $\frac{N_c(N_c-1)}{2}$, while improving gradient conditioning and reducing overfitting [7–9].

While the restriction to $O(N_c)$ can in principle be relaxed to $U(N_c)$ to recover the cross-mixing structure of the classical FNO (cf. Equation (5)), we retain the real orthogonal parameterisation for the training-efficiency reasons discussed in Section 2.4. The effective action of circuit k is

$$|\hat{A}\rangle \longmapsto \frac{1}{\|A\|} \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} \tilde{a}_{ij}^{(k)} |e_j\rangle |e_i\rangle, \qquad \tilde{a}_{ij}^{(k)} = \begin{cases} \left(W_Q^k \hat{v}_t(k)\right)_i & j = k, \\ \left(\hat{v}_t(j)\right)_i & j \neq k. \end{cases} \tag{16}$$

After applying the Unary-IQFT to the lower register, the output of circuit k is

$$\mathrm{output}_k = \mathcal{F}^{-1}\left[\hat{v}_t(1), \ldots, W_Q^k \hat{v}_t(k), \ldots, \hat{v}_t(N_s)\right], \tag{17}$$

 doi:10.20944/preprints202605.1199.v1

which modifies only mode k but carries the full Fourier content of all other modes $j \neq k$. Summing the K circuit outputs therefore does not reproduce the classical FNO target $A^* = \mathcal{F}^{-1}[W_Q^1 \hat{v}_t(1), \ldots, W_Q^K \hat{v}_t(K), 0, \ldots, 0]$: each circuit contributes a spurious background from the non-selected modes, and the sum accumulates K such copies. To see this explicitly, decompose each output by linearity of $\mathcal{F}^{-1}$:

$$\begin{aligned} \text{output}_k &= \mathcal{F}^{-1}[\hat{v}_t(1), \ldots, \hat{v}_t(N_s)] + \mathcal{F}^{-1}[0, \ldots, (W_Q^k - I)\hat{v}_t(k), \ldots, 0] \\ &= A + \mathcal{F}^{-1}[\Delta \hat{A}^{(k)}], \end{aligned} \tag{18}$$

where $\Delta \hat{A}^{(k)} \in \mathbb{C}^{N_c \times N_s}$ has $(W_Q^k - I)\hat{v}_t(k)$ in column k and zero elsewhere. Summing over all K circuits and applying linearity of $\mathcal{F}^{-1}$ to collect the $\Delta \hat{A}^{(k)}$ terms:

$$\begin{aligned} \sum_{k=1}^{K} \text{output}_k &= K \cdot A + \mathcal{F}^{-1}\left[\sum_{k=1}^{K} \Delta \hat{A}^{(k)}\right] \\ &= K \cdot A + \mathcal{F}^{-1}[W_Q^1 \hat{v}_t(1), \ldots, W_Q^K \hat{v}_t(K), 0, \ldots, 0] - \mathcal{F}^{-1}[\hat{v}_t(1), \ldots, \hat{v}_t(K), 0, \ldots, 0] \\ &= A^* + (K-1) \cdot A + \mathcal{F}^{-1}[0, \ldots, 0, \hat{v}_t(K+1), \ldots, \hat{v}_t(N_s)], \end{aligned} \tag{19}$$

where $A^* = \mathcal{F}^{-1}[W_Q^1 \hat{v}_t(1), \ldots, W_Q^K \hat{v}_t(K), 0, \ldots, 0]$ is the classical FNO target defined in Equation (7), and we used $A = A_K + \mathcal{F}^{-1}[0, \ldots, 0, \hat{v}_t(K+1), \ldots, \hat{v}_t(N_s)]$ with $A_K = \mathcal{F}^{-1}[\hat{v}_t(1), \ldots, \hat{v}_t(K), 0, \ldots, 0]$. The deviation from A^* has two distinct sources: $(K-1) \cdot A$, arising from the $K-1$ spurious background copies accumulated across circuits, and the residual contribution of the non-selected modes $k > K$, which are never zeroed.

Note that if, in each circuit k, all modes $j \neq k$ were set to zero before the IQFT, the output of circuit k would reduce to $\mathcal{F}^{-1}[0, \ldots, W_Q^k \hat{v}_t(k), \ldots, 0]$, and the sum over K circuits would give exactly A^* by linearity of $\mathcal{F}^{-1}$. This is precisely the fix introduced in Section 2.3: routing the non-selected modes through loss channels with transmissivity $\eta = 0$ erases their contribution before summation, restoring exact equivalence with Equation (7) and Equation (12).

In terms of circuit resources, the K pyramidal interferometers run on independent registers and therefore contribute a total of $K \cdot \frac{N_c(N_c-1)}{2}$ RBS gates, equivalently $K \cdot N_c(N_c-1)$ CZ gates, at an overall depth of $2N_c - 3 = O(N_c)$, since all K circuits execute in parallel.

RBS gate and hardware overhead. Throughout the whole QFNO implementation, the elementary two-qubit operation is the RBS gate. This gate is not native to any current qubit platform and must be compiled into hardware primitives, incurring a substantial overhead that places the full QFNO beyond near-term reach and motivates the continuous-variable reformulation of Section 2.3. The RBS gate at angle θ acts on two qubits as

$$\text{RBS}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{20}$$

which, restricted to the unary subspace $\{|01\rangle, |10\rangle\}$, implements a 2×2 rotation by θ. On standard qubit hardware there is no native RBS gate, so it must be compiled into elementary operations. A minimal decomposition is

$$\text{RBS}(\theta) = (H \otimes H)\ \text{CZ}\ (R_Y(\theta) \otimes R_Y(-\theta))\ \text{CZ}\ (H \otimes H), \tag{21}$$

requiring 2 entangling (CZ) gates and 6 single-qubit gates per RBS. Here H denotes the single-qubit Hadamard gate, which maps computational basis states to equal superpositions, while CZ denotes the

 doi:10.20944/preprints202605.1199.v1

controlled-Z entangling gate, which applies a phase flip when both qubits are in state $|1\rangle$ [18]. Table 1 summarises the depth and CZ-gate count for each phase of a single Parallel QFNO layer.

Table 1. Circuit resources per phase of a single Parallel QFNO layer. Each RBS gate compiles to 2 CZ gates (Equation (21)). Encoding counts from [15]; QFT/IQFT butterfly from [5].

Phase	Depth	RBS gates	CZ gates
Encoding	$\log_2 N_c + 2N_c \log_2 N_s$	$(N_c-1)+(2N_c-1)(N_s-1)$	$2[(N_c-1)+(2N_c-1)(N_s-1)]$
Unary-QFT	$\log_2 N_s$	$\frac{N_s}{2}\log_2 N_s$	$N_s \log_2 N_s$
QFL (K parallel circuits)	$2N_c-3$	$K \cdot \frac{N_c(N_c-1)}{2}$	$K \cdot N_c(N_c-1)$
Unary-IQFT	$\log_2 N_s$	$\frac{N_s}{2}\log_2 N_s$	$N_s \log_2 N_s$
Total	$O(N_c \log N_s)$	$O(KN_c^2 + N_cN_s + N_s \log N_s)$	

For the representative choice $N_c = N_s = 32$, $K = 4$ the total number of CZ gates is 8 256 per layer at a sequential circuit depth of 396. Current superconducting and trapped-ion processors achieve two-qubit gate fidelities of 99–99.9% [19]. At 99% fidelity the overall circuit fidelity per layer is

$$0.99^{8256} \approx 10^{-36},$$

and even at the optimistic 99.9% level it falls to $0.999^{8256} \approx 2.6 \times 10^{-4}$, rendering the output indistinguishable from noise for any practically relevant (N_c, N_s). This positions the full QFNO squarely beyond near-term NISQ capabilities, motivating the shift to a photonic continuous-variable architecture in which the analogous operations are implemented by passive beam splitters, linear-optics elements that realise exact unitary rotations on the field quadratures without any qubit–qubit entangling interaction. The dominant noise source in photonic platforms is propagation loss [20], which is qualitatively different from and typically far smaller than the coherent error rates of two-qubit gates on current qubit hardware.

In summary, the qubit QFNO has two fundamental limitations: a spectral mismatch with the classical FNO (Equation (19)) and a prohibitive hardware overhead on qubit platforms, despite the RBS gate being natively a linear-optical element. These are the main motivations for the CV-QFNO introduced in Section 2.3.

2.3. Continuous-Variable Quantum Fourier Neural Operator

We now introduce the Continuous-Variable Quantum Fourier Neural Operator (CV-QFNO), which constitutes the main architectural contribution of this work. The proposed model reformulates the spectral layer of the Fourier Neural Operator within a continuous-variable photonic framework, using the qubit-based QFNO reviewed in Section 2.2 as its direct conceptual precursor.

The qubit-based QFNO encodes data in the unary subspace and implements the learnable spectral mixing through orthogonal rotations realised by RBS gates. Since an RBS gate is the discrete-variable analogue of a linear-optical beam splitter, the structure of the QFNO already suggests a natural connection with photonic hardware. Moving to a full CV formulation [11] makes this connection explicit: the Fourier transform, the trainable orthogonal mixing, and the suppression of unwanted modes can all be expressed directly in terms of Gaussian photonic operations.

This reformulation is not only a change of computational substrate, but also addresses a key limitation of the qubit construction. As shown in Section 2.2, the parallel QFNO architecture leaves the non-selected Fourier modes unchanged, rather than setting them to zero. This produces a systematic mismatch with the classical FNO spectral layer (cf. Equation (19)), whose action is explicitly truncated in Fourier space. In the CV setting, by contrast, unused modes can be erased directly by routing them through loss channels with transmissivity $\eta = 0$. Physically, this corresponds to sending the corresponding optical modes to a beam dump; mathematically, it removes their correlations with the active register. As a result, the CV-QFNO can reproduce the truncated spectral operator of the classical

 doi:10.20944/preprints202605.1199.v1

FNO exactly on the retained modes.

Bipartite Gaussian encoding and CV-QFT. We recall only the elements of the continuous-variable Fourier layer that are needed to construct the CV-QFNO. The full derivation of the bipartite Gaussian encoding, the TMS-SVD loading procedure, and the Cooley–Tukey optical QFT is given in [14].

The latent matrix $A \in \mathbb{R}^{N_c \times N_s}$ introduced in Equation (2) is encoded into a bipartite Gaussian state with two registers. The first register, denoted by r_1, contains N_c bosonic modes associated with the channel dimension, while the second register, denoted by r_2, contains N_s bosonic modes associated with the spatial dimension. We write x_{r_1}, p_{r_1} for the vectors of position and momentum quadratures of the modes in register r_1, and x_{r_2}, p_{r_2} for the corresponding quadrature vectors of register r_2.

Using the TMS-SVD encoding of [14], the singular values of A are loaded by independent two-mode squeezing gates, while the left and right singular vectors are implemented by passive interferometers acting on the two registers. The resulting bipartite Gaussian state encodes the latent matrix in the inter-register covariance blocks as

$$\sigma_{x_{r_1} x_{r_2}} = A, \qquad \sigma_{p_{r_1} p_{r_2}} = -A, \qquad \sigma_{x_{r_1} p_{r_2}} = \sigma_{p_{r_1} x_{r_2}} = 0. \tag{22}$$

Here $\sigma_{x_{r_1} x_{r_2}}$ denotes the block of the covariance matrix collecting all second-order correlations between the x-quadratures of registers r_1 and r_2. Analogously, $\sigma_{p_{r_1} p_{r_2}}$ collects the inter-register p-quadrature correlations, while $\sigma_{x_{r_1} p_{r_2}}$ and $\sigma_{p_{r_1} x_{r_2}}$ are the mixed inter-register blocks. Thus, the classical latent matrix A is represented exactly as the x-quadrature cross-correlation block of the bipartite covariance matrix.

On this representation, Fourier transformations are implemented by applying Cooley–Tukey optical QFT circuits to the corresponding registers. Each butterfly of the classical FFT is mapped to Gaussian photonic operations, namely phase rotations and 50:50 beam splitters, as shown in [14]. Applying the QFT to the spatial register r_2 implements the Fourier matrix F_{N_s} on the N_s spatial modes. Since the latent matrix is stored in the cross-covariance block $\sigma_{x_{r_1} x_{r_2}} = A$, this operation transforms all rows of A simultaneously:

$$\sigma'_{x_{r_1} x_{r_2}} = \sigma_{x_{r_1} x_{r_2}} F_{N_s}^{\top} = A F_{N_s}^{\top} = \hat{A}. \tag{23}$$

This is the one-dimensional Fourier representation used in the 1D CV-QFNO construction, where the Fourier transform acts along the spatial direction while the channel register is left unchanged.

Analogously, applying the QFT to the channel register r_1 implements F_{N_c} on the N_c channel modes, giving

$$\sigma'_{x_{r_1} x_{r_2}} = F_{N_c} \sigma_{x_{r_1} x_{r_2}} = F_{N_c} A. \tag{24}$$

When QFTs are applied simultaneously to both registers, the two-dimensional Fourier transform is obtained:

$$\hat{A} = F_{N_c} A F_{N_s}^{\top}. \tag{25}$$

Since the covariance matrix is real, the complex matrix $\hat{A}$ cannot be stored in a single covariance block. Instead, its real and imaginary parts are distributed across two inter-register blocks,

$$\begin{aligned} \sigma'_{x_{r_1} x_{r_2}} &= \mathrm{Re}\left(F_{N_c} A F_{N_s}^{\top}\right) = \mathrm{Re}(\hat{A}), \\ \sigma'_{x_{r_1} p_{r_2}} &= \mathrm{Im}\left(F_{N_c} A F_{N_s}^{\top}\right) = \mathrm{Im}(\hat{A}). \end{aligned} \tag{26}$$

Equivalently,

$$\hat{A} = \sigma'_{x_{r_1} x_{r_2}} + i\, \sigma'_{x_{r_1} p_{r_2}}. \tag{27}$$

Thus, the CV-QFT provides direct access to the complex Fourier coefficients through covariance blocks of the Gaussian state. For the two-register transform, the optical QFT stage has gate count

 doi:10.20944/preprints202605.1199.v1

$O(N_c \log N_c + N_s \log N_s)$ and depth $O(\log(N_c N_s))$, excluding the classical SVD preprocessing and encoding costs discussed in [14].

This covariance-block representation also determines how channel mixing is implemented in the CV-QFNO. A retained Fourier mode k is represented by its real and imaginary parts across the two covariance blocks above. Applying a passive interferometer $W^{(k)} \in O(N_c)$ to the channel register therefore acts on both parts in parallel,

$$\mathrm{Re}(\hat{a}^{(k)}) \mapsto W^{(k)} \mathrm{Re}(\hat{a}^{(k)}), \qquad \mathrm{Im}(\hat{a}^{(k)}) \mapsto W^{(k)} \mathrm{Im}(\hat{a}^{(k)}). \tag{28}$$

Consequently, as in the qubit QFNO of Section 2.2, the CV-QFNO uses real orthogonal spectral mixing matrices rather than unconstrained complex matrices. This reduces the number of trainable parameters from $2N_c^2$ to $\frac{N_c(N_c-1)}{2}$ per retained mode and provides a direct passive-interferometric implementation of the mixing layer. In principle, this restriction could be relaxed to unitary mixing in a more general photonic architecture; here we retain the orthogonal parameterisation for consistency with the qubit formulation and for the training-efficiency reasons discussed in Section 2.4.

The novel contribution of the CV-QFNO lies in faithfully reproducing the action of R_ϕ in Equation (1): after the Fourier transform, each retained mode k must be mixed along the *channel* dimension by an independent learnable weight matrix, while all non-selected modes are set to zero. The local W branch of Equation (1) remains classical and is unchanged; it is the global convolution branch that requires a dedicated optical implementation. We now describe how this is achieved first for a 1D input and then for the 2D case.

1D architecture: K parallel circuits. In a discrete-variable setting, the one- or two-dimensional Quantum Fourier Layer could in principle be implemented through controlled operations on the upper (channel) register conditioned on the lower (spatial) register. To avoid the resulting CNOT-type overhead one may instead adopt the Parallel QFNO strategy of Section 2.3, which directly inspires the present CV construction. However, the CV Gaussian formalism only admits passive linear-optics operations, namely beam splitters and phase shifters, which act jointly on all modes and therefore cannot implement this type of conditional logic. As a result, sequential mode-selective weight application within a single circuit is not available in this framework. To overcome this limitation, we follow the intuition of the Parallel QFNO and run K independent optical circuits in parallel, one for each retained Fourier mode k. Each circuit receives the same latent matrix A and proceeds as follows; a schematic representation is provided in Figure 2.

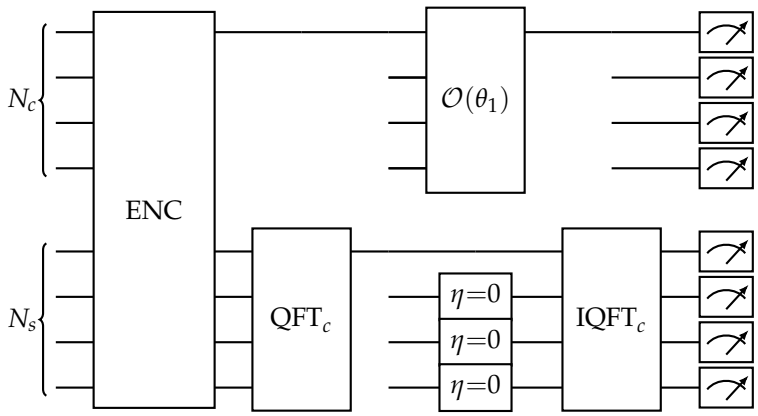


Figure 2. CV-FNO spectral layer — branch $k = 1$ (1D case). The latent matrix $A \in \mathbb{R}^{N_c \times N_s}$ is encoded via TMS-SVD into the off-diagonal covariance block $\sigma_{x_{r_1} x_{r_2}}$, with the n-mode upper register (r_1) and the m-mode lower register (r_2). A row-wise QFT$_c$ on the lower register places $\hat{A}$ in that block. The pyramidal interferometer $O(\theta_1)$ is applied to the upper register; loss channels $\eta = 0$ are applied to all lower-register modes $j \neq 1$, leaving mode $k = 1$ active. Each of the remaining $K - 1$ branches is structurally identical, with weight $O(\theta_k)$ and mode k left free. An IQFT$_c$ followed by homodyne measurement yields the output contribution of mode $k = 1$.

 doi:10.20944/preprints202605.1199.v1

As shown in Sec. 2.3, A is loaded into a $(N_c + N_s)$-mode Gaussian state via TMS-SVD (Equation (22)), setting $\sigma_{x_{r_1} x_{r_2}} = A$ and $\sigma_{x_{r_1} p_{r_2}} = 0$. Then, QFT_c is applied to the lower N_s-mode register and real and imaginary parts of $\hat{A}$ distribute across two covariance blocks:

$$\sigma'_{x_{r_1} x_{r_2}} = \mathrm{Re}(\hat{A}), \qquad \sigma'_{x_{r_1} p_{r_2}} = \mathrm{Im}(\hat{A}), \tag{29}$$

so that $\hat{A} = \sigma'_{x_{r_1} x_{r_2}} + i\,\sigma'_{x_{r_1} p_{r_2}}$, the CV analogue of Equation (3). An orthogonal interferometer $O(\theta_k) \in O(N_c)$ acting on the upper register applies the same real matrix to both blocks simultaneously (cf. Equation (28)), which is the exact photonic analogue of the FNO applying a weight matrix to both the real and imaginary parts of each complex Fourier mode vector. For notational compactness, all subsequent operations in this section are written directly on the full complex matrix $\hat{A}$, with the understanding that each step acts equivalently on both covariance blocks. At this point, the first K parallel circuits are assigned independent trainable pyramidal interferometers $O(\theta_k)$, with one interferometer for each retained Fourier mode k.

We now focus on a generic k-th circuit; as will be shown below, the full spectral output is then recovered by summing the contributions of all K circuits through the linearity of the inverse Fourier transform. In the k-th circuit, absent further constraints, applying $O(\theta_k)$ to the upper N_c-mode register would mix the channel dimension across all N_s Fourier columns simultaneously, yielding

$$\hat{A}'^{(k)} = \Big(O(\theta_k)\hat{v}_t(1) \quad \cdots \quad O(\theta_k)\hat{v}_t(N_s) \Big) = O(\theta_k)\hat{A}, \tag{30}$$

which does not reproduce the mode-selective operation of Equation (6).

To isolate the k-th mode, all lower-register modes $j \neq k$ are routed through loss channels with transmissivity $\eta = 0$, which replace those modes with vacuum and erase their correlations with the upper register. As a result, only the k-th column of $\hat{A}$ survives across the active covariance blocks, and the k-th parallel circuit produces the filtered Fourier contribution

$$\hat{A}'^{(k)} = \Big(0 \quad \cdots \quad 0 \quad O(\theta_k)\hat{v}_t(k) \quad 0 \quad \cdots \quad 0 \Big) \in \mathbb{C}^{N_c \times N_s}, \tag{31}$$

where the only nonzero column is the k-th one, thereby realising Equation (4) exactly. The inverse Fourier transform is then applied independently within each parallel circuit. In the k-th branch this produces

$$A'^{(k)} = \mathcal{F}^{-1}\Big(\hat{A}'^{(k)} \Big) \in \mathbb{R}^{N_c \times N_s}, \tag{32}$$

and homodyne measurement returns the corresponding contribution to the latent output in the physical domain. The K branches are combined only at this stage, by summing their measured outputs:

$$A' = \sum_{k=1}^{K} A'^{(k)} = \sum_{k=1}^{K} \mathcal{F}^{-1}\Big(\hat{A}'^{(k)} \Big). \tag{33}$$

Using the linearity of $\mathcal{F}^{-1}$, one obtains

$$A' = \mathcal{F}^{-1}\left(\sum_{k=1}^{K} \hat{A}'^{(k)} \right) = \mathcal{F}^{-1}\Big(\Big(O(\theta_1)\hat{v}_t(1) \quad \cdots \quad O(\theta_K)\hat{v}_t(K) \quad 0 \quad \cdots \quad 0 \Big) \Big), \tag{34}$$

thereby recovering exactly the filtered spectral structure of Equation (7).

Optical resource count. Since the CV-QFNO is built entirely from beam splitters and phase shifters, except for the two-mode squeezing gates used in the encoding step, it incurs no entangling-gate compilation overhead. Table 2 summarises the circuit depth and beam-splitter count associated with each phase of a single CV-QFNO layer.

 doi:10.20944/preprints202605.1199.v1

Table 2. Optical resources per phase of a single CV-QFNO layer (K parallel circuits). BS denotes a passive beam splitter; the encoding step additionally requires $r = \min(N_c, N_s)$ two-mode squeezers (depth 1). Interferometer counts use the Clements rectangular decomposition [21].

Phase	Depth	Beam splitters
TMS-SVD encoding (U, Σ, V)	$1 + \max(N_c,\ N_s)$	$\frac{N_c(N_c-1)}{2} + \frac{N_s(N_s-1)}{2}$
CV-QFT$_c$	$\log_2 N_s$	$\frac{N_s}{2}\log_2 N_s$
Loss channels (K circuits)	1	$K(N_s-1)$
CV-QFL (K parallel pyramidal)	$2N_c - 3$	$K \cdot \frac{N_c(N_c-1)}{2}$
CV-IQFT$_c$	$\log_2 N_s$	$\frac{N_s}{2}\log_2 N_s$
Total	$O(\max(N_c, N_s))$	$O(KN_c^2 + N_c^2 + N_s^2)$

For the representative choice $N_c = N_s = 32$ and $K = 4$, the above expressions yield a total of 3 260 passive beam splitters and 32 two-mode squeezers, at an overall sequential depth of 105. By comparison, the corresponding qubit QFNO layer requires 8 256 CZ gates at depth 396 (Table 1). Since beam splitters are low-loss passive optical elements [20], the fidelity argument that renders the qubit implementation impractical does not carry over to the CV setting, making the CV-QFNO a substantially more realistic photonic architecture. However, we note that the TMS-SVD encoding requires a classical singular value decomposition of the input matrix prior to each forward pass, introducing an additional preprocessing cost of order $O(\min(N_c, N_s)^2 \max(N_c, N_s))$, which is absent in both the qubit and classical FNO pipelines and should therefore be accounted for in any end-to-end latency comparison.

At the same time, it is important to acknowledge the practical limitations of the present construction. Although the CV-QFNO removes the severe entangling-gate overhead of the qubit formulation, a full hardware realisation of the architecture considered here remains far beyond current photonic quantum technology, especially because the model requires multiple parallel optical branches and precise control of Gaussian operations, loss channels, and readout. In addition, quantum machine learning training remains hybrid, with parameter optimisation performed classically, so that any potential advantage at the circuit level must be weighed against the substantial optimisation overhead of the full training loop. For these reasons, the CV-QFNO should presently be regarded primarily as a theoretical and architectural framework, useful for showing how a continuous-variable quantum Fourier neural operator could be constructed in a manner faithful to the spectral structure of the classical FNO. For classical PDE problems on classical hardware, the standard FNO remains by far the most practical and effective choice, and the main value of the CV formulation lies not in immediate applicability, but in clarifying what a coherent photonic quantum analogue of the FNO would look like.

2D architecture: K_xK_y parallel branches. In the two-dimensional setting, the latent representation is the tensor $A \in \mathbb{R}^{N_c \times N_x \times N_y}$ introduced in Equation (8), where N_c denotes the number of latent channels, while N_x and N_y denote the number of sampled grid points along the two spatial directions. Thus, $A_{l,:,:} \in \mathbb{R}^{N_x \times N_y}$ denotes the two-dimensional spatial field associated with the l-th latent channel. Unlike the 1D case, where the latent representation is a matrix with one channel axis and one spatial axis, the 2D tensor has a three-index structure (channel, x-spatial, and y-spatial). This structure does not admit, within the present Gaussian construction, a direct bipartite encoding that preserves the desired separation between channel mixing and the two spatial Fourier directions.

For this reason, we allocate N_c independent optical circuits, one for each channel l, and encode in the l-th circuit the spatial slice $A_{l,:,:} \in \mathbb{R}^{N_x \times N_y}$ via TMS-SVD into a bipartite Gaussian state. In this 2D construction, the two registers are denoted by r_x and r_y: r_x contains N_x bosonic modes associated with the grid points along the x-direction, while r_y contains N_y bosonic modes associated with the grid

 doi:10.20944/preprints202605.1199.v1

points along the y-direction. By Equation (22), the x-quadrature inter-register covariance block in the l-th circuit satisfies

$$\sigma^{(l)}_{x_{r_x} x_{r_y}} = A_{l,:,:}, \qquad l = 1, \ldots, N_c. \tag{35}$$

A two-dimensional CV-QFT is then applied independently within each circuit by acting simultaneously with QFT_x on register r_x and QFT_y on register r_y, as in Equation (26). Because the 2D DFT result $\hat{A}_{l,:,:} = F_{N_x} A_{l,:,:} F_{N_y}^{\top} \in \mathbb{C}^{N_x \times N_y}$ is complex, real and imaginary parts are distributed across two covariance blocks:

$$\sigma^{(l)}_{x_{r_x} x_{r_y}} \longmapsto \mathrm{Re}(\hat{A}_{l,:,:}), \qquad \sigma^{(l)}_{x_{r_x} p_{r_y}} \longmapsto \mathrm{Im}(\hat{A}_{l,:,:}), \tag{36}$$

so that $\hat{A}_{l,k_1,k_2} = (\sigma^{(l)}_{x_{r_x} x_{r_y}} + i\, \sigma^{(l)}_{x_{r_x} p_{r_y}})_{k_1,k_2}$ (cf. Equation (27)). The trainable pyramidal interferometer $O(\theta_{k_1,k_2}) \in O(N_c)$ acting across the N_c circuits applies the same real orthogonal matrix simultaneously to both covariance blocks,

$$\mathrm{Re}(\hat{v}_t(k_1,k_2)) \mapsto O(\theta_{k_1,k_2})\, \mathrm{Re}(\hat{v}_t(k_1,k_2)), \qquad \mathrm{Im}(\hat{v}_t(k_1,k_2)) \mapsto O(\theta_{k_1,k_2})\, \mathrm{Im}(\hat{v}_t(k_1,k_2)), \tag{37}$$

the two-dimensional analogue of Equation (28). For notational compactness, the remainder of this section writes all operations on the full complex quantities $\hat{A}_{l,:,:}$ and $\hat{v}_t(k_1,k_2)$, with the understanding that each step acts equivalently on both covariance blocks.

We now focus on a fixed retained mode pair (k_1,k_2) with $k_1 \in \{1,\ldots,K_x\}$ and $k_2 \in \{1,\ldots,K_y\}$; the full output is then recovered by summing the contributions of all retained mode pairs, as we have previously done in the one-dimensional problem. To isolate the pair (k_1,k_2), all r_x-modes $i \neq k_1$ and all r_y-modes $j \neq k_2$ are routed through loss channels with transmissivity $\eta = 0$, see Figure 3. This erases all correlations except those associated with the selected Fourier entry, so that in the l-th circuit only the scalar coefficient $\hat{A}_{l,k_1,k_2}$ survives. Collecting these coefficients across the N_c circuits defines the channel vector

$$\hat{v}_t(k_1,k_2) = \big(\hat{A}_{1,k_1,k_2}, \ldots, \hat{A}_{N_c,k_1,k_2}\big)^{\top} \in \mathbb{C}^{N_c}. \tag{38}$$

A trainable pyramidal interferometer $O(\theta_{k_1,k_2}) \in O(N_c)$ then acts across the N_c circuits on the first register (r_x), connecting the k_1-th mode wire of each circuit into a shared beam-splitter network. Because r_x carries the row index of each covariance block, this action corresponds to a left-multiplication and implements the channel-mixing operation of Equation (10):

$$\hat{v}_t(k_1,k_2) \longmapsto O(\theta_{k_1,k_2})\, \hat{v}_t(k_1,k_2). \tag{39}$$

Equivalently, the branch associated with the retained mode pair (k_1,k_2) produces the filtered Fourier contribution

$$\hat{A}'^{(k_1,k_2)}_{l,i,j} = \begin{cases} \big[O(\theta_{k_1,k_2})\, \hat{v}_t(k_1,k_2)\big]_l, & (i,j) = (k_1,k_2), \\ 0, & \text{otherwise}, \end{cases} \tag{40}$$

that is, a tensor whose only nonzero spatial-frequency entry is the selected mode pair (k_1,k_2).

The two-dimensional inverse Fourier transform is then applied independently within each branch, yielding the corresponding contribution in the physical domain. After homodyne measurement, the full latent output is recovered by summing the contributions of all retained mode pairs:

$$A' = \sum_{k_1=1}^{K_x} \sum_{k_2=1}^{K_y} \mathcal{F}_{2D}^{-1}\Big(\hat{A}'^{(k_1,k_2)}\Big). \tag{41}$$

By linearity of $\mathcal{F}_{2D}^{-1}$, this can be written as

$$A' = \mathcal{F}_{2D}^{-1}\left(\sum_{k_1=1}^{K_x} \sum_{k_2=1}^{K_y} \hat{A}'^{(k_1,k_2)}\right), \tag{42}$$

 doi:10.20944/preprints202605.1199.v1

which exactly reconstructs the filtered spectral tensor of Equation (12). A schematic representation of a single mode-pair branch is shown in Figure 3.

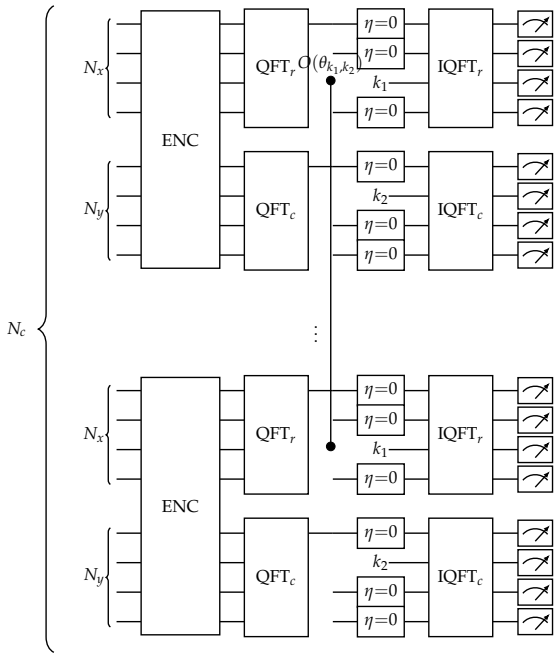


Figure 3. CV-FNO spectral layer — mode pair (k_1, k_2) (2D case). N_c optical circuits run in parallel, one per channel l. Each circuit encodes the 2D slice $A_{l,:,:}$ via TMS-SVD and applies a 2D CV-QFT on both registers. The pyramidal interferometer $O(\theta_{k_1,k_2})$ connects the k_1-th upper-register wire across all N_c circuits, mixing the channel dimension for all column modes. Loss channels $\eta = 0$ on upper-register modes $i \neq k_1$ and lower-register modes $j \neq k_2$ then isolate the target mode pair, realising Equation (10). Homodyne measurement yields the output contribution of mode pair (k_1, k_2).

Optical resource count (2D). The 2D layer comprises $N_c \times K_x K_y$ parallel circuit instances, one per channel per retained mode pair, plus $K_x K_y$ cross-circuit interferometers for channel mixing. Table 3 summarises the resource budget.

Table 3. Optical resources per phase of a single CV-QFNO layer in the 2D case ($K_x K_y$ retained mode pairs, N_c channels). Depths of phases acting on disjoint registers are computed in parallel; the sequential circuit depth is dominated by the encoding and the channel mixing interferometer. Interferometer counts use the Clements rectangular decomposition [21].

Phase	Depth	Beam splitters (total)
TMS-SVD encoding (N_c circuits per branch)	$1 + \max(N_x,\, N_y)$	$N_c K_x K_y \left[\frac{N_x(N_x-1)}{2} + \frac{N_y(N_y-1)}{2}\right]$
2D CV-QFT (both registers in parallel)	$\max(\log_2 N_x,\, \log_2 N_y)$	$N_c K_x K_y \left[\frac{N_x}{2}\log_2 N_x + \frac{N_y}{2}\log_2 N_y\right]$
Loss channels (r_x and r_y)	1	$N_c K_x K_y\,(N_x + N_y - 2)$
Channel mixing $O(\theta_{k_1,k_2})$ ($K_x K_y$ interferometers)	$2N_c - 3$	$K_x K_y \cdot \frac{N_c(N_c-1)}{2}$
2D CV-IQFT (both registers in parallel)	$\max(\log_2 N_x,\, \log_2 N_y)$	$N_c K_x K_y \left[\frac{N_x}{2}\log_2 N_x + \frac{N_y}{2}\log_2 N_y\right]$
Total	$O(\max(N_x, N_y, N_c))$	$O\left(N_c K_x K_y (N_x^2 + N_y^2) + K_x K_y N_c^2\right)$

Compared with the 1D case (Table 2), the sequential depth grows only modestly: the 2D QFT replaces a single-register transform with a parallel two-register transform (depth $\max(\log_2 N_x, \log_2 N_y)$ instead of $\log_2 N_s$), leaving the overall depth dominated by the encoding and channel-mixing stages. However, the beam-splitter count scales as $O(N_c K_x K_y (N_x^2 + N_y^2))$, reflecting the N_c parallel circuit copies required by the 2D architecture; this additional cost is a direct consequence of the three-index structure of the 2D latent tensor and is shared by any photonic implementation of the 2D operator.

 doi:10.20944/preprints202605.1199.v1

2.4. Simulation Setup and Orthogonal Parameterisation

The FNO, QFNO, and the psoposed CV-QFNO are algorithmically distinct architectures rooted in different computational substrates and physical principles: classical tensor operations, qubit circuits restricted to a structured Hilbert subspace, and Gaussian photonic modes, respectively. Despite these differences, all three models admit exact classical linear-algebraic representations. The quantum gates and optical transformations appearing in QFNO and CV-QFNO can therefore be replaced by their corresponding matrix actions, allowing the full architectures to be simulated and trained in PyTorch without introducing an additional approximation of the underlying circuit model. This is the standard validation strategy in quantum-inspired machine learning prior to the availability of hardware capable of running the complete architecture at the required scale [4,5,15], and it is the approach adopted in this work.

In the numerical experiments, the three models share the same overall neural-operator structure, consisting of a lifting map, four spectral blocks, and a final projection. They differ only in the spectral convolution layer, where the quantum-inspired structure enters. In the classical FNO, each retained Fourier mode is mixed across the channel dimension by an unconstrained complex matrix $R_\phi^{(k)} \in \mathbb{C}^{N_c \times N_c}$. In QFNO and CV-QFNO, this operation is replaced by an orthogonal matrix $W^{(k)} \in O(N_c)$, reflecting the unitary or passive-interferometric origin of the corresponding quantum transformation. Earlier quantum-inspired networks typically construct such orthogonal matrices as ordered products of $M = N_c(N_c-1)/2$ Givens rotations,

$$W = G_{i_1 j_1}(\theta_1)\, G_{i_2 j_2}(\theta_2) \cdots G_{i_M j_M}(\theta_M),$$

thereby mirroring the gate-by-gate structure of RBS circuits or optical beam-splitter networks [4,15,16]. Although physically natural, this representation is less convenient for classical training, since the matrix W must be reconstructed at every forward pass through a sequential product of $O(N_c^2)$ rotation factors.

In this work, both QFNO and CV-QFNO instead use a matrix-exponential parameterisation of the orthogonal spectral-mixing matrices. For each retained mode, we introduce a skew-symmetric generator $A \in \mathbb{R}^{N_c \times N_c}$, with $A^\top = -A$, whose independent upper-triangular entries are the trainable parameters:

$$A_{ij} = \theta_{ij}, \quad A_{ji} = -\theta_{ij} \quad \text{for } i < j, \qquad A_{ii} = 0, \qquad W^{(k)} = \exp(A). \tag{43}$$

This guarantees orthogonality by construction, since $\exp(A)^\top = \exp(A^\top) = \exp(-A) = \exp(A)^{-1}$. The parameterisation uses exactly $M = N_c(N_c-1)/2$ free parameters, the same number as the Givens decomposition, but avoids the explicit sequential product of rotations during training. In PyTorch, the exponential is evaluated by `torch.linalg.matrix_exp`, which uses scaling-and-squaring methods with Padé approximants [22], at computational cost $O(N_c^3)$. This cost belongs only to the classical simulation: on a physical CV photonic device, a trained orthogonal transformation would be compiled into beam-splitter and phase-shifter settings, for instance through the Clements decomposition [21], and executed by passive propagation through the interferometric network rather than by explicit matrix exponentiation.

The remaining distinction between QFNO and CV-QFNO concerns the treatment of Fourier modes beyond the retained cutoff. In the classical FNO, non-retained modes are set to zero before applying the inverse Fourier transform. CV-QFNO reproduces this operation exactly in the simulated model by initialising the output spectrum to zero and writing only the retained, orthogonally mixed modes; this is the classical linear-algebraic counterpart of routing the non-selected optical modes through loss channels with transmissivity $\eta = 0$. QFNO, by contrast, follows the behaviour of the parallel qubit construction: the full input spectrum is copied, and only the retained modes are overwritten by the mixed values. The modes beyond the cutoff therefore remain unchanged, because they are not explicitly discarded by the qubit circuit. This difference is the source of the systematic spectral contamination discussed in Section 2.2.

 doi:10.20944/preprints202605.1199.v1

The simulations reported here should therefore be interpreted as a classical validation of the proposed architectures, not as a demonstration of quantum computational advantage. Since every quantum or photonic operation is replaced by its matrix equivalent and executed on a GPU, the resulting pipeline is a classical algorithm: it does not realise quantum speedup, exponential state-space compression, or genuine quantum parallelism. Its purpose is architectural. The experiments show that the CV formulation provides a spectrally faithful and hardware-compatible quantum analogue of the FNO layer, with a native mechanism for exact mode suppression and a direct mapping of Fourier and mixing operations onto Gaussian photonic primitives. Any practical computational advantage would require deployment on native photonic hardware, where these transformations are carried out by the physical propagation of light rather than by explicit matrix multiplication, or the use of non-Gaussian resources leading to regimes not efficiently captured by classical Gaussian emulation. These perspectives are discussed further in the conclusion.

All simulations were run on a workstation equipped with an Intel Core i7-14700F processor, 32 GB DDR5 RAM, and an NVIDIA GeForce RTX 4070 GPU with 12 GB of GDDR6X VRAM. PyTorch training was accelerated on the GPU using CUDA.

3. Results

We evaluate the proposed CV-QFNO against two reference architectures: the classical Fourier Neural Operator (FNO) [2] and the qubit-based Quantum Fourier Neural Operator (QFNO) [5]. The comparison is performed on four standard PDE benchmarks: the one-dimensional Burgers' equation, the one-dimensional heat equation, the two-dimensional Navier–Stokes equation in vorticity form, and the two-dimensional Darcy flow problem. Within each benchmark, all models are trained using identical data splits, training schedules, and optimisation settings.

The main result of the paper is the formulation of a CV photonic analogue of the FNO spectral layer that faithfully reproduces its truncated Fourier structure. As discussed above, this formulation is particularly natural in CV quantum computing: the Fourier transform, mode filtering, and trainable channel mixing can be implemented directly in terms of Gaussian photonic operations, such as beam splitters, phase shifters, loss channels, and interferometric networks. In this sense, the CV-QFNO provides a hardware-compatible route to a photonic implementation of the FNO spectral mechanism, while avoiding the qubit-gate compilation overhead of discrete-variable QFNO constructions.

Within this perspective, the numerical benchmarks show that CV-QFNO retains competitive predictive accuracy with respect to both FNO and QFNO, while using substantially fewer trainable parameters than the classical model. This parameter reduction follows from the orthogonal parameterisation of the spectral-mixing matrices induced by the pyramidal interferometers. Since CV-QFNO and QFNO use the same orthogonal channel-mixing structure, their trainable-parameter counts are identical in all experiments.

3.1. One-Dimensional Experiments

We first consider two one-dimensional operator-learning benchmarks: the viscous Burgers' equation,

$$\partial_t u(x,t) + u(x,t)\,\partial_x u(x,t) = \nu\,\partial_{xx} u(x,t), \qquad x \in [0,1],\ t \in (0,T], \tag{44}$$

and the heat equation,

$$\partial_t u(x,t) = \alpha\,\partial_{xx} u(x,t), \qquad x \in [0,1],\ t \in (0,T]. \tag{45}$$

Both problems are posed on the one-dimensional spatial domain $D = [0,1]$, so $d = 1$, and govern the time evolution of a scalar field $u : D \times [0,T] \to \mathbb{R}$, where x is the spatial coordinate and t is time. In Equation (44), $u(x,t)$ denotes the scalar velocity field, $\nu > 0$ is the viscosity, while $\partial_t u$, $\partial_x u$, and $\partial_{xx} u$ denote differentiation with respect to time, the first spatial derivative, and the second spatial

 doi:10.20944/preprints202605.1199.v1

derivative, respectively. In Equation (45), $u(x,t)$ represents the temperature field and $\alpha > 0$ is the diffusion coefficient.

In both cases, the learning task is formulated as the approximation of a solution operator mapping an initial condition to the solution at a later time. Following the standard FNO setup, the model input is the initial condition concatenated with the spatial coordinate as a positional feature,

$$a(x) = \big(u(x,0),\, x\big) \in \mathbb{R}^{d_a}, \qquad d_a = 2, \tag{46}$$

and the target output is the scalar solution at the final time,

$$u(x,T) \in \mathbb{R}^{d_u}, \qquad d_u = 1. \tag{47}$$

Thus, each training sample corresponds to an input–output pair

$$a(\cdot) = \big(u(\cdot,0),\, \cdot\big) \longmapsto u(\cdot,T),$$

and the learned operator approximates

$$\mathcal{G}^\dagger : a(\cdot) \mapsto u(\cdot,T).$$

In the implementation, the input is evaluated on a uniform grid of N_s spatial points and lifted to a latent representation with N_c channels. Therefore, at each Fourier layer, the latent state has the matrix structure $A \in \mathbb{R}^{N_c \times N_s}$ introduced in Equation (2).
The Fourier transform is applied along the spatial direction, i.e., across the N_s grid points like in Equation 23, while the spectral mixing acts along the channel dimension on each retained Fourier mode. This is precisely the one-dimensional structure used by the classical FNO, the qubit QFNO, and the CV-QFNO.

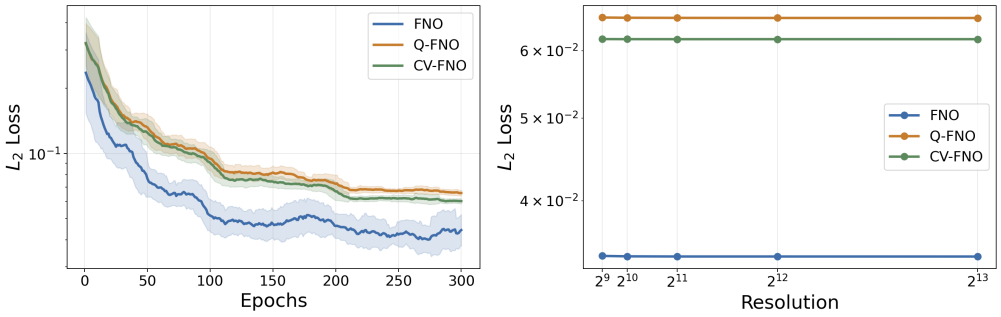
The two PDEs probe complementary regimes of operator learning. Burgers' equation combines nonlinear advection with viscous diffusion and may develop sharp solution features, making it a nonlinear benchmark for spectral operator learning. The heat equation is linear and purely diffusive; in Fourier space, its coefficients decay exponentially in time, making it a controlled benchmark for evaluating whether the learned spectral layer captures dissipative dynamics.

Both one-dimensional experiments use the same training protocol. We use $N_{\text{train}} = 200$ input–output pairs and $N_{\text{test}} = 20$ test pairs, discretised on a uniform grid of 512 points. All models are trained for 300 epochs with the Adam optimiser [23], using an initial learning rate of 10^{-3}, reduced by a factor of two every 100 epochs. For both benchmarks, we analyse the models through three complementary diagnostics: the validation loss over training, zero-shot resolution generalisation, and spectral accuracy. The first diagnostic compares the optimisation behaviour of the architectures; the second measures whether a model trained at one resolution can be evaluated on unseen grids; and the third assesses whether the learned operator reproduces the relevant Fourier content of the target dynamics.

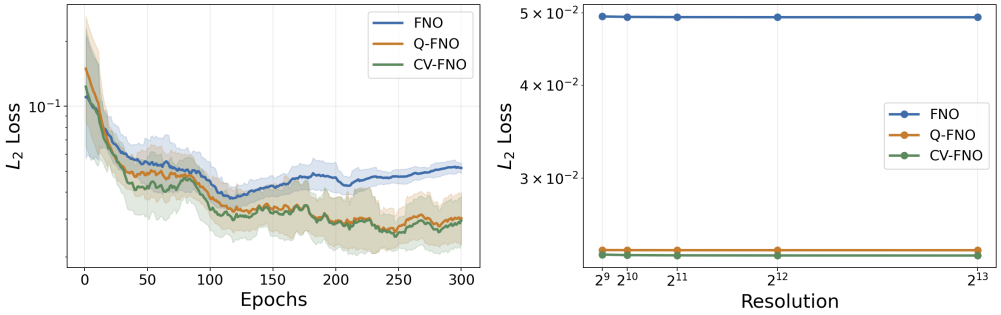
Figure 4 reports the loss evolution and resolution generalisation for both one-dimensional benchmarks. On Burgers' equation, the classical FNO contains 419,009 trainable parameters, whereas QFNO and CV-QFNO each use 122,561, corresponding to a 3.4× reduction. On the heat equation, FNO uses 28,145 parameters, while QFNO and CV-QFNO use 9,329, giving a 3.0× reduction. This parameter reduction, however, should not be interpreted as a direct wall-clock speedup in the present classical simulations. In QFNO and CV-QFNO, the quantum or photonic operations are emulated through classical linear algebra, and a substantial part of the cost comes from constructing the orthogonal spectral mixing matrices, as described in Sec. 2.4. Therefore, training-time comparisons would mix the effect of fewer trainable parameters with the overhead of simulating the quantum-inspired layer. Despite this smaller parameter count, the two quantum-inspired models retain accuracy comparable to the classical FNO, as confirmed by the converged test errors in Table 4(a)–(b). The significance of this

 doi:10.20944/preprints202605.1199.v1

result is primarily architectural rather than computational: it shows that the FNO spectral layer can be replaced by a more constrained orthogonal parameterisation without a substantial loss of predictive performance. Such a constraint may improve optimisation stability and become advantageous in future native quantum or photonic implementations.



(**a**) Burgers' equation.



(**b**) Heat equation.

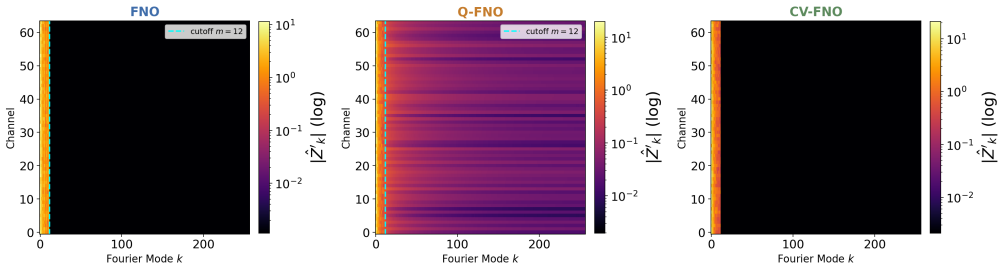
Figure 4. Training dynamics and resolution generalisation — 1D benchmarks. Left: loss evolution over the epochs for a single training run; the shaded band represents the local variability computed over a sliding window of 20 epochs. Right: relative ℓ_2 test loss as a function of spatial resolution. Models are trained at a fixed resolution and evaluated zero-shot at all resolutions shown.

The loss curves highlight different optimisation behaviours across the two benchmarks. For Burgers' equation, FNO converges to a relative ℓ_2 error of $(2.67 \pm 0.55) \times 10^{-2}$ (Table 4a), but its trajectory is visibly noisier and more oscillatory, whereas QFNO and CV-QFNO converge to $(6.63 \pm 0.20) \times 10^{-2}$ and $(6.05 \pm 0.15) \times 10^{-2}$, respectively, with considerably smoother descent. For the heat equation, the situation is reversed: FNO reaches $(5.22 \pm 0.36) \times 10^{-2}$ (Table 4b) but shows signs of overfitting during training, while QFNO and CV-QFNO converge to lower errors of $(3.10 \pm 0.85) \times 10^{-2}$ and $(2.92 \pm 0.82) \times 10^{-2}$, respectively, with more stable trajectories. These trends are consistent with the regularising effect of the orthogonal spectral-mixing layers: by constraining the learnable matrices to the orthogonal group, QFNO and CV-QFNO reduce the effective degrees of freedom of the spectral operator, improve the conditioning of the optimisation problem, and mitigate both noisy training dynamics and overfitting.

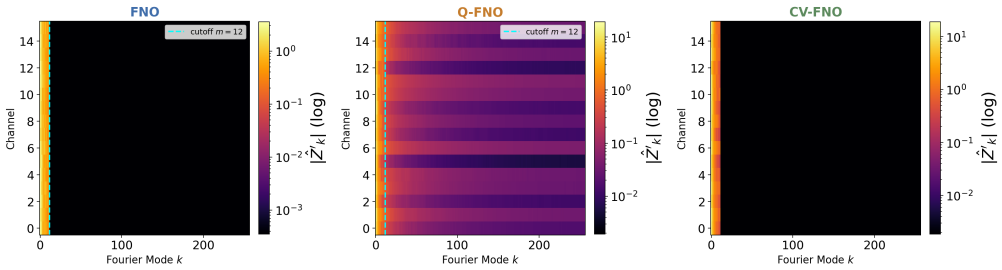
The resolution-generalisation results, shown in the right panels of Figure 4, confirm that all three architectures preserve the resolution-invariant character expected from neural operators. The models are trained at the base resolution of 512 grid points and evaluated zero-shot at the other resolutions shown, without retraining. Across both Burgers' and heat benchmarks, QFNO and CV-QFNO remain competitive with FNO over the full tested range, with no systematic degradation at unseen spatial resolutions.

Figure 5 provides a direct spectral diagnostic of the learned operators. In both benchmarks, CV-QFNO reproduces the reference spectrum with essentially the same fidelity as FNO and confines the spectral activation to the retained modes.

 doi:10.20944/preprints202605.1199.v1



(**a**) Burgers' equation.



(**b**) Heat equation.

Figure 5. Spectral analysis — 1D benchmarks. Power spectra of the predicted solutions and ground truth for FNO, QFNO, and CV-QFNO. The classical FNO and the proposed CV-QFNO exhibit consistent spectral behaviour: both reproduce the relevant low-frequency content and enforce a sharp truncation beyond the retained Fourier modes. In contrast, QFNO displays nonzero spectral contributions beyond the cutoff, visible as horizontal channel-dependent bands, due to the non-suppressed Fourier modes inherited from the parallel qubit construction.

As explained in Section 2.4, the experiments are performed as classical PyTorch simulations of the corresponding CV operations. Therefore, the mode-isolation step is implemented numerically by setting the non-selected Fourier modes to zero, which is the classical linear-algebra equivalent of routing those modes through CV loss channels with transmissivity $\eta = 0$. The resulting spectra show that this simulated loss-channel mechanism preserves the truncated spectral inductive bias of the classical FNO without introducing spurious spectral artefacts.

3.2. Two-Dimensional Experiments

We next consider two two-dimensional operator-learning benchmarks: the incompressible Navier–Stokes equation in vorticity form,

$$\partial_t \omega(x,y,t) + \big(u(x,y,t)\cdot\nabla\big)\omega(x,y,t) = \nu\,\Delta\omega(x,y,t), \qquad (x,y)\in[0,1]^2,\ t\in(0,T], \tag{48}$$

and the Darcy flow equation,

$$-\nabla\cdot\big(a(x,y)\,\nabla u(x,y)\big) = f(x,y), \qquad (x,y)\in[0,1]^2. \tag{49}$$

Both problems are posed on the two-dimensional spatial domain $D = [0,1]^2$, so $d = 2$, but they represent different classes of PDE operators. In Equation (48), ω denotes the scalar vorticity, $u = (u_x, u_y)$ is the divergence-free velocity field, $\Delta\omega$ is the spatial Laplacian of the vorticity, and $\nu > 0$ is the kinematic viscosity. The velocity is recovered from the vorticity through the Biot–Savart relation. In Equation (49), $a(x,y)$ is the spatially varying permeability, $u(x,y)$ is the pressure field, ∇u is the pressure gradient, and $f(x,y)$ is a prescribed source term.

For the Navier–Stokes benchmark, the learning task is formulated as a one-step time-evolution problem. The input consists of the spatial coordinates concatenated with T_{in} consecutive vorticity snapshots,

$$a(x,y) = \big(x,\ y,\ \omega(x,y,t_1),\ldots,\omega(x,y,t_{T_{\text{in}}})\big)\in\mathbb{R}^{d_a}, \qquad d_a = T_{\text{in}} + 2, \tag{50}$$

 doi:10.20944/preprints202605.1199.v1

and the target output is the vorticity at the next time step,

$$u(x,y) = \omega(x,y,t_{T_{\text{in}}+1}) \in \mathbb{R}^{d_u}, \qquad d_u = 1. \tag{51}$$

Thus, the learned operator approximates the local-in-time evolution map

$$\mathcal{G}^\dagger : \big(x,\, y,\, \omega(\cdot,t_1),\ldots,\omega(\cdot,t_{T_{\text{in}}})\big) \mapsto \omega(\cdot,t_{T_{\text{in}}+1}).$$

For Darcy flow, the learning task is instead static. The input is the permeability field concatenated with the spatial coordinates,

$$a(x,y) = \big(x,\, y,\, a(x,y)\big) \in \mathbb{R}^{d_a}, \qquad d_a = 3, \tag{52}$$

and the target output is the scalar pressure solution,

$$u(x,y) \in \mathbb{R}^{d_u}, \qquad d_u = 1. \tag{53}$$

The corresponding solution operator is

$$\mathcal{G}^\dagger : a(\cdot,\cdot) \mapsto u(\cdot,\cdot),$$

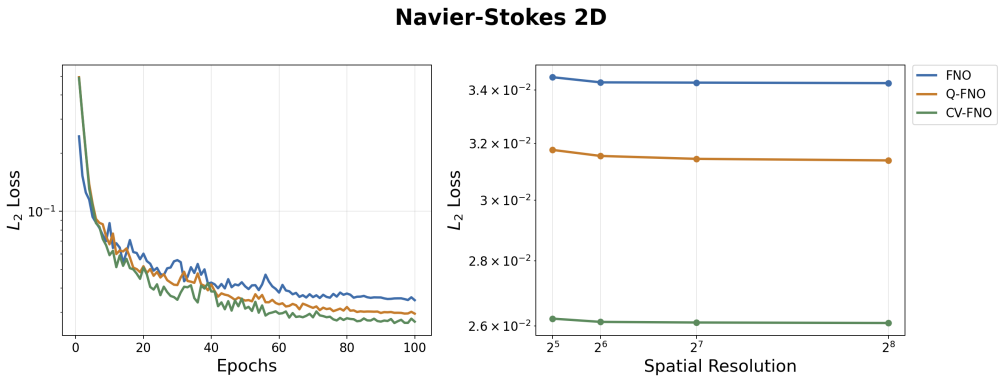
where $a(x,y)$ denotes the permeability coefficient in Equation (49). This elliptic operator is nonlocal: local changes in the permeability can affect the pressure solution across the whole domain.

In the implementation, both benchmarks are evaluated on a 64×64 uniform grid. After the lifting map, the latent field is represented by the tensor $A \in \mathbb{R}^{N_c \times N_x \times N_y}$ with $N_x = N_y = 64$, as introduced in Equation (8), where N_c is the latent channel dimension. The Fourier transform is applied along the two spatial directions, see Equation 25, while the spectral mixing acts along the channel dimension for each retained Fourier mode pair (k_1, k_2). This is the two-dimensional spectral structure used by the classical FNO and reproduced, with constrained orthogonal mixing, by the QFNO and CV-QFNO architectures.

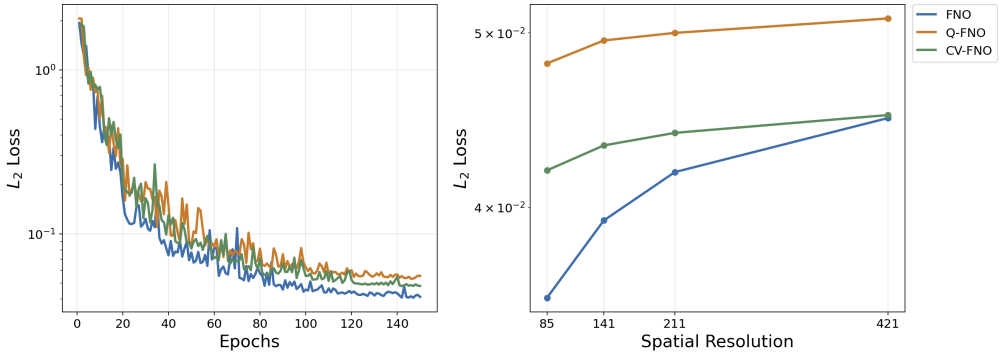
All experiments are designed to test the proposed architecture in a low-data regime. For the Navier–Stokes benchmark, we use $N_{\text{train}} = 25$ trajectories and $N_{\text{test}} = 5$, training the models for 100 epochs. For Darcy flow, we use $N_{\text{train}} = 400$ samples and $N_{\text{test}} = 100$, training for 150 epochs. The different dataset sizes reflect the different structure of the two learning problems. In the time-stepping formulation of Navier–Stokes, consecutive vorticity fields are strongly correlated, the dynamics is smooth over short time intervals, and each trajectory contributes multiple input–output pairs to the effective training set. Darcy flow, by contrast, is a static elliptic problem: each input is an independent realisation of a thresholded Gaussian random permeability field $a(x,y) \in \{3, 12\}$, paired with the corresponding pressure solution $u(x,y)$. Since there is no temporal structure to exploit, and since the elliptic solution operator $a \mapsto u$ is nonlocal through the associated Green's operator, a larger ensemble is required to represent the input distribution adequately.

Both benchmarks are trained with the Adam optimiser, using an initial learning rate of 2×10^{-3}, reduced by a factor of two every 20 epochs. Figure 6 shows the training loss and resolution generalisation curves for both benchmarks. All three architectures converge on both problems; converged test errors are reported in Table 4(c)–(d). On Navier–Stokes, FNO reaches a relative ℓ_2 of $(3.57 \pm 0.05) \times 10^{-2}$, while QFNO and CV-QFNO attain $(3.01 \pm 0.03) \times 10^{-2}$ and $(2.72 \pm 0.05) \times 10^{-2}$, respectively, with $4.1\times$ fewer parameters (926,517 for FNO versus 223,797 for the quantum-inspired models). On Darcy flow, FNO converges to $(4.27 \pm 0.16) \times 10^{-2}$, compared with $(5.51 \pm 0.13) \times 10^{-2}$ for QFNO and $(4.95 \pm 0.16) \times 10^{-2}$ for CV-QFNO, again with a $4.1\times$ parameter reduction (3,698,817 versus 910,977). The consistent parameter advantage across all four benchmarks confirms that the orthogonal spectral parameterisation provides a robust structural benefit independent of spatial dimension and PDE class.

 doi:10.20944/preprints202605.1199.v1



(**a**) Navier–Stokes equation (vorticity formulation).



(**b**) Darcy flow.

Figure 6. Training dynamics and resolution generalisation — 2D benchmarks. Left: training loss over epochs. Right: relative ℓ_2 test loss across spatial resolutions. Loss curves are shown without the shaded variability band of the 1D case, as the 2D optimisation trajectories are considerably smoother. QFNO and CV-QFNO achieve accuracy competitive with FNO while using 4.1× fewer parameters on both benchmarks.

3.3. Summary of Benchmark Results

We conclude the numerical evaluation by collecting in Table 4 the final test errors obtained by all models across the four benchmark problems. The table reports the relative ℓ_2 error at convergence, computed as mean and standard deviation over the last 20 training epochs. This provides a compact comparison of the classical FNO, the qubit-based QFNO, and the proposed CV-QFNO under the same benchmark-specific data splits and optimisation settings.

Overall, the results confirm that the CV-QFNO preserves the predictive accuracy of Fourier-based neural operators while using an orthogonally constrained spectral parameterisation. Across the 1D and 2D benchmarks, its performance is comparable to that of the classical FNO and consistently improves over the qubit-based QFNO in the tested settings. The improvement over QFNO is especially relevant because CV-QFNO reproduces the truncated spectral structure of the FNO exactly, whereas the parallel qubit construction leaves non-retained Fourier modes unchanged.

 doi:10.20944/preprints202605.1199.v1

Table 4. Convergence summary. Relative ℓ_2 test error at convergence, reported as mean $\pm$ standard deviation over the last 20 epochs of training (matching the variability band in Figures 4 and 6).

(a) Burgers' equation (1D)

Model	Relative ℓ_2 (mean $\pm$ std)
FNO	$(2.67 \pm 0.55) \times 10^{-2}$
QFNO	$(6.63 \pm 0.20) \times 10^{-2}$
CV-QFNO	$(6.05 \pm 0.15) \times 10^{-2}$

(b) Heat equation (1D)

Model	Relative ℓ_2 (mean $\pm$ std)
FNO	$(5.22 \pm 0.36) \times 10^{-2}$
QFNO	$(3.10 \pm 0.85) \times 10^{-2}$
CV-QFNO	$(2.92 \pm 0.82) \times 10^{-2}$

(c) Navier–Stokes (2D)

Model	Relative ℓ_2 (mean $\pm$ std)
FNO	$(3.57 \pm 0.05) \times 10^{-2}$
QFNO	$(3.01 \pm 0.03) \times 10^{-2}$
CV-QFNO	$(2.72 \pm 0.05) \times 10^{-2}$

(d) Darcy flow (2D)

Model	Relative ℓ_2 (mean $\pm$ std)
FNO	$(4.27 \pm 0.16) \times 10^{-2}$
QFNO	$(5.51 \pm 0.13) \times 10^{-2}$
CV-QFNO	$(4.95 \pm 0.16) \times 10^{-2}$

4. Discussion

The results presented in this work should be interpreted primarily as an architectural and algorithmic validation of the proposed CV-QFNO framework. Although the construction is motivated by CV quantum computing and by its natural photonic implementation, all numerical experiments are performed as classical simulations. Every optical or quantum operation is replaced by its matrix-algebraic equivalent and executed on a GPU. The resulting pipeline is therefore a classical algorithm, and no quantum or photonic hardware is involved in the reported results. Consequently, the experiments should not be interpreted as a demonstration of quantum computational advantage.

This distinction is particularly important in the present Gaussian setting. Gaussian CV circuits admit efficient classical descriptions in terms of covariance matrices, and the simulations performed in this work exploit exactly this linear-algebraic structure. The value of the proposed model therefore does not lie in outperforming classical neural operators on conventional hardware, but in showing that the spectral mechanism of the FNO admits a coherent CV photonic formulation. In particular, the CV-QFNO identifies a direct optical analogue of the three operations that define the FNO spectral layer: Fourier transformation, mode selection, and channel mixing. Within this formulation, the $\eta = 0$ loss-channel mechanism provides a principled photonic counterpart of the mode-zeroing step of the classical FNO, resolving the spectral mismatch of the qubit-based QFNO.

The numerical results support this interpretation. Across the one- and two-dimensional benchmarks, CV-QFNO achieves predictive accuracy comparable to the classical FNO and the qubit-based QFNO, while using a substantially smaller number of trainable parameters than the classical FNO. This reduction follows from the orthogonal constraint imposed on the spectral-mixing matrices. However, it should not be interpreted as a direct wall-clock speedup in the present classical simulations. In QFNO and CV-QFNO, the quantum or photonic operations are emulated through classical linear algebra, and a significant part of the computational cost is associated with constructing the orthogonal mixing matrices. Training-time comparisons would therefore conflate the effect of a smaller parameter count with the overhead of simulating the quantum-inspired layer.

The spectral diagnostics further clarify the structural difference between the architectures. The classical FNO and the proposed CV-QFNO both enforce a hard cutoff beyond the retained Fourier modes. By contrast, the parallel qubit-QFNO construction leaves non-selected modes unchanged, which produces residual spectral activations beyond the cutoff. This behaviour is visible in the power spectra and directly reflects the spectral mismatch analysed in Section 2.2. The CV-QFNO removes this mismatch by suppressing non-retained modes through zero-transmissivity loss channels before the inverse Fourier transform.

 doi:10.20944/preprints202605.1199.v1

A second limitation concerns hardware realisability. Although the CV-QFNO removes the entangling-gate compilation overhead of the qubit formulation, a full photonic implementation of the architecture remains beyond current integrated photonic capabilities. The required number of optical modes, parallel branches, Gaussian transformations, loss channels, and readout stages would pose substantial experimental challenges. Moreover, training remains a hybrid process in which the optimisation of the trainable parameters is performed classically. Any advantage at the circuit level would therefore need to be assessed together with the cost of data preparation, readout, and classical optimisation.

For the classical PDE benchmarks considered here, standard neural operators remain the most practical approach. They are accurate, computationally efficient, and directly deployable on conventional hardware. The parameter reduction observed in CV-QFNO is structurally interesting and reflects the orthogonal parameterisation of the spectral layer, but it does not by itself provide a sufficient practical reason to replace classical FNOs with photonic hardware for these tasks. The main significance of the present work is therefore conceptual: it establishes a faithful CV photonic blueprint for the FNO spectral layer and clarifies how a quantum-photonic operator-learning architecture can reproduce the truncated Fourier inductive bias of the classical model.

5. Conclusions

This work introduced the Continuous-Variable Quantum Fourier Neural Operator (CV-QFNO), a Gaussian photonic analogue of the spectral layer of the classical Fourier Neural Operator. Building on bipartite Gaussian encoding and the optical quantum Fourier transform, CV-QFNO represents latent fields in the covariance structure of Gaussian states and maps Fourier-domain processing, mode selection, and channel mixing onto native continuous-variable optical operations.

The main architectural contribution is a CV spectral mixing layer that recovers the truncated spectral structure of the classical FNO. In the one-dimensional setting, each retained Fourier mode is processed in an independent optical branch: non-selected modes are removed through loss channels with transmissivity $\eta = 0$, while a pyramidal interferometer applies a learnable orthogonal transformation across the channel dimension. The construction is then extended to two spatial dimensions through a cross-circuit interferometric coupling, providing a photonic generalisation of the same mode-wise principle.

This formulation addresses two limitations of qubit-based QFNOs. First, it avoids the compilation overhead associated with RBS gates, since the corresponding transformations are native linear-optical operations in the CV setting. Second, it removes the spectral mismatch of the parallel qubit-QFNO architecture, where non-selected Fourier modes remain unchanged, by suppressing them directly through $\eta = 0$ loss channels. As a result, CV-QFNO restores the hard spectral truncation that defines the classical FNO layer.

We also introduced a matrix-exponential parameterisation of the orthogonal spectral-mixing matrices, which preserves the number of trainable parameters of the Givens construction while avoiding an explicit sequential product of rotations during classical training. Numerical experiments on Burgers', heat, Navier–Stokes, and Darcy-flow benchmarks show that CV-QFNO achieves competitive predictive accuracy with FNO and QFNO, while using substantially fewer trainable parameters than the classical FNO. The spectral diagnostics confirm the key structural feature of the CV construction: CV-QFNO reproduces the hard cutoff of the classical FNO spectrum, whereas QFNO retains residual activations beyond the selected modes.

Overall, CV-QFNO provides a coherent Gaussian photonic formulation of Fourier-domain operator learning. Rather than demonstrating immediate computational advantage, the present work defines an architectural blueprint for photonic neural operators.

The most promising future directions concern settings in which the input data are themselves optical or quantum in nature, rather than classical fields already discretised on a digital grid. For natively optical inputs, such as coherent signals generated by interferometers, wavefront sensors, or

 doi:10.20944/preprints202605.1199.v1

free-space optical communication systems, a photonic implementation of CV-QFNO could in principle operate directly on the incoming field, reducing or bypassing the analogue-to-digital conversion and classical encoding overheads. For intrinsically quantum inputs, as in quantum sensing, quantum simulation, or quantum communication, the bipartite CV architecture could act directly on state-level correlations rather than on classical samples extracted from them.

A longer-term route toward computational regimes beyond efficient classical emulation would require moving beyond the Gaussian framework. Non-Gaussian resources, such as cubic-phase operations or photon-number-resolving measurements, could introduce nonlinear spectral interactions outside the efficiently simulable Gaussian sector. Exploring these directions, together with higher-dimensional generalisations of the architecture and applications to more complex time-dependent problems, represents a natural next step for this line of research.

Author Contributions: P.M. conceived the research idea, developed the methodology, implemented the models, carried out all experiments, and wrote the manuscript. S. Ma. and M. S. contributed to the revision of the manuscript, supported the research, and acquired the funding. S. Mk. contributed to the scientific discussion and provided guidance and feedback throughout the project. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by ICSC – "National Research Centre in High Performance Computing, Big Data and Quantum Computing", funded by the European Union – NextGenerationEU.

Data Availability Statement: The code and data supporting the findings of this study are available in the GitHub repository of the corresponding author. They are currently private and will be made publicly available upon publication of this article.

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

References

1. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators. *Nature Machine Intelligence* **2021**, *3*, 218–229. https://doi.org/10.1038/s42256-021-00302-5.
2. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; et al. Fourier Neural Operator for Parametric Partial Differential Equations. *International Conference on Learning Representations (ICLR)* **2021**.
3. Kovachki, N.; Li, Z.; Liu, B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs. *Journal of Machine Learning Research* **2023**, *24*, 1–97.
4. Kerenidis, I.; Prakash, A. Quantum machine learning with subspace states, 2022, [arXiv:quant-ph/2202.00054].
5. Jain, N.; Landman, J.; Mathur, N.; Kerenidis, I. Quantum Fourier networks for solving parametric PDEs. *Quantum Science and Technology* **2024**, *9*, 035026.
6. Marcandelli, P.; He, Y.; Mariani, S.; Siena, M.; Markidis, S. Partitioned Hybrid Quantum Fourier Neural Operators for Scientific Quantum Machine Learning. *arXiv preprint* **2025**.
7. Li, S.; Jia, K.; Wen, Y.; Liu, T.; Tao, D. Orthogonal Deep Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, *43*, 1352–1368. https://doi.org/10.1109/TPAMI.2019.2948352.
8. Wang, J.; Chen, Y.; Chakraborty, R.; Yu, S.X. Orthogonal Convolutional Neural Networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11502–11512. https://doi.org/10.1109/CVPR42600.2020.01152.
9. Monbroussou, L.; Mamon, E.Z.; Landman, J.; Grilo, A.B.; Kukla, R.; Kashefi, E. Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning. *Quantum* **2025**, *9*, 1745. https://doi.org/10.22331/q-2025-05-15-1745.
10. Lloyd, S.; Braunstein, S.L. Quantum Computation over Continuous Variables. *Physical Review Letters* **1999**, *82*, 1784–1787. https://doi.org/10.1103/physrevlett.82.1784.
11. Weedbrook, C.; Pirandola, S.; García-Patrón, R.; Cerf, N.J.; Ralph, T.C.; Shapiro, J.H.; Lloyd, S. Gaussian quantum information. *Reviews of Modern Physics* **2012**, *84*, 621–669. https://doi.org/10.1103/revmodphys.84.621.

 doi:10.20944/preprints202605.1199.v1

12. Lau, H.K.; Pooser, R.; Siopsis, G.; Weedbrook, C. Quantum Machine Learning over Infinite Dimensions. *Physical Review Letters* **2017**, *118*. https://doi.org/10.1103/physrevlett.118.080501.
13. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters* **2019**, *122*. https://doi.org/10.1103/physrevlett.122.040504.
14. Marcandelli, P.; Mariani, S.; Siena, M.; Markidis, S. A Continuous-Variable Quantum Fourier Layer: Applications to Filtering and PDE Solving, 2026, [arXiv:quant-ph/2603.17847].
15. Cherrat, E.A.; Kerenidis, I.; Mathur, N.; Landman, J.; Strahm, M.; Li, Y.Y. Quantum Vision Transformers. *Quantum* **2024**, *8*, 1265. https://doi.org/10.22331/q-2024-02-22-1265.
16. Kerenidis, I.; Landman, J.; Mathur, N. Classical and Quantum Algorithms for Orthogonal Neural Networks, 2021, [arXiv:quant-ph/2106.07198].
17. Reck, M.; Zeilinger, A.; Bernstein, H.J.; Bertani, P. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **1994**, *73*, 58–61. https://doi.org/10.1103/PhysRevLett.73.58.
18. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press.
19. Krantz, P.; Kjaergaard, M.; Yan, F.; Orlando, T.P.; Gustavsson, S.; Oliver, W.D. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews* **2019**, *6*. https://doi.org/10.1063/1.5089550.
20. Flamini, F.; Spagnolo, N.; Sciarrino, F. Photonic quantum information processing: A review. *Reports on Progress in Physics* **2018**, *82*, 016001. https://doi.org/10.1088/1361-6633/aad5b2.
21. Clements, W.R.; Humphreys, P.C.; Metcalf, B.J.; Kolthammer, W.S.; Walmsley, I.A. An Optimal Design for Universal Multiport Interferometers, 2017, [arXiv:physics.optics/1603.08788].
22. Higham, N.J. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications* **2005**, *26*, 1179–1193. https://doi.org/10.1137/04061101X.
23. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), 2015.