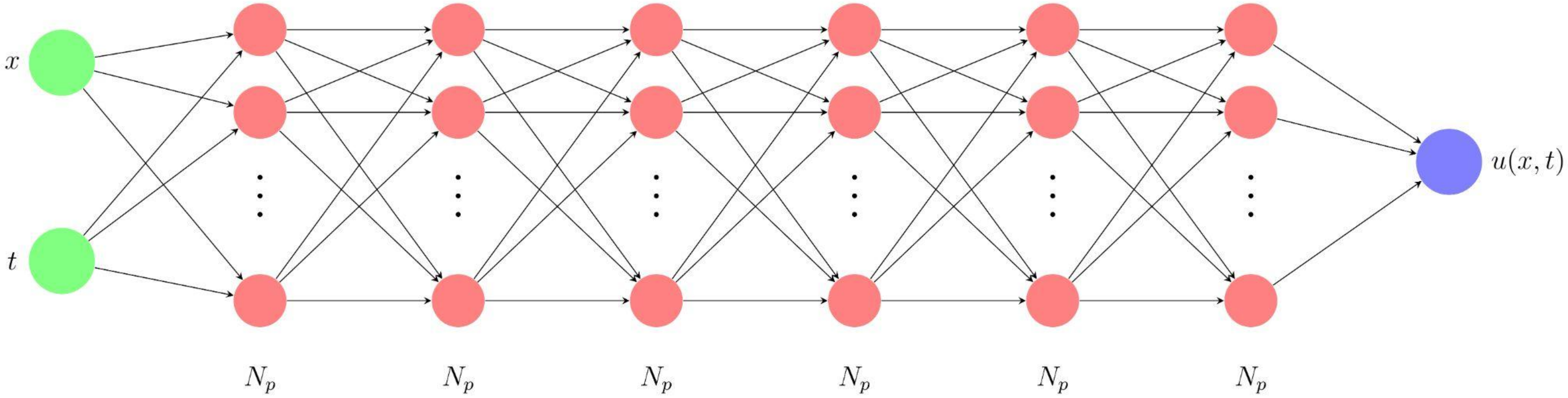
Péter Kovács
(HUN–REN Wigner Research Centre for Physics)

Collaborators: Zoltán Kolarovszki (HUN-REN Wigner RCP & ELTE),
Zoltán Zimborás (University of Helsinki & HUN-REN Wigner RCP)

Comparison of Classical Digital, Classical Photonic, and Quantum Photonic NNs for Solving PDEs: A Case Study



GPU Day 2025

HUN–REN Center, Budapest

May 22–23, 2025

Outline

Introduction

- PINNs aim to solve ODEs and PDEs
- Successfully applied to many eq. such as the heat equation, Poisson equation, Navier-Stokes eq.
- Also used to solve fractional equations, integral-differential equations and stochastic PDEs
- Can be used to tackle inverse problems, i.e. determine some parameters of the PDE

Introduction

- PINNs aim to solve ODEs and PDEs
- Successfully applied to many eq. such as the heat equation, Poisson equation, Navier–Stokes eq.
- Also used to solve fractional equations, integral–differential equations and stochastic PDEs
- Can be used to tackle inverse problems, i.e. determine some parameters of the PDE

Why should we use instead of classical numerical techniques, like FEM, FDM or FVM (Finite Element, Difference, Volume Method)?

Introduction

Why should we use instead of classical numerical techniques?

- Some PDEs are notoriously difficult to solve using standard methods, like convection dominated convection–diffusion equations
- NN are universal approximators
- PINNs do not require a mesh (operate on scattered points of Ω)
- PINNs can incorporate observational or experimental data directly into the training

Introduction

Aim of this talk

- To demonstrate how a PiNN can be defined in cases of
  - Classical digital
  - Classical photonic
  - Quantum photonic

  Networks, using the Burgers' equation as a specific example.
- Show their differences/advantages

General form of PINNs

$$\mathcal{F}(u(x);\lambda) = f(x), \quad x \in \Omega$$
$$\mathcal{B}(u(x)) = g(x), \quad x \in \delta\Omega$$

where

$$x = [x_1, \ldots, x_{d-1}; t], \Omega \subset \mathbb{R}^d$$

$\mathcal{F}$ is a non-linear differential operator, while $\mathcal{B}$ stands for arbitrary initial and/or boundary conditions. $f(x)$ and $g(x)$ representing the data, $u(x)$ is the solution, while λ denotes some physical parameters

to be more specific, in 1+1 D a large class of equations:

$$\partial_t u(x,t) + \mathcal{N}(u(x,t)) = 0$$
$$x \in [a,b], \; t \in [0,T]$$

eqs. includes conservation laws, diffusion processes, kinetic equations, etc.

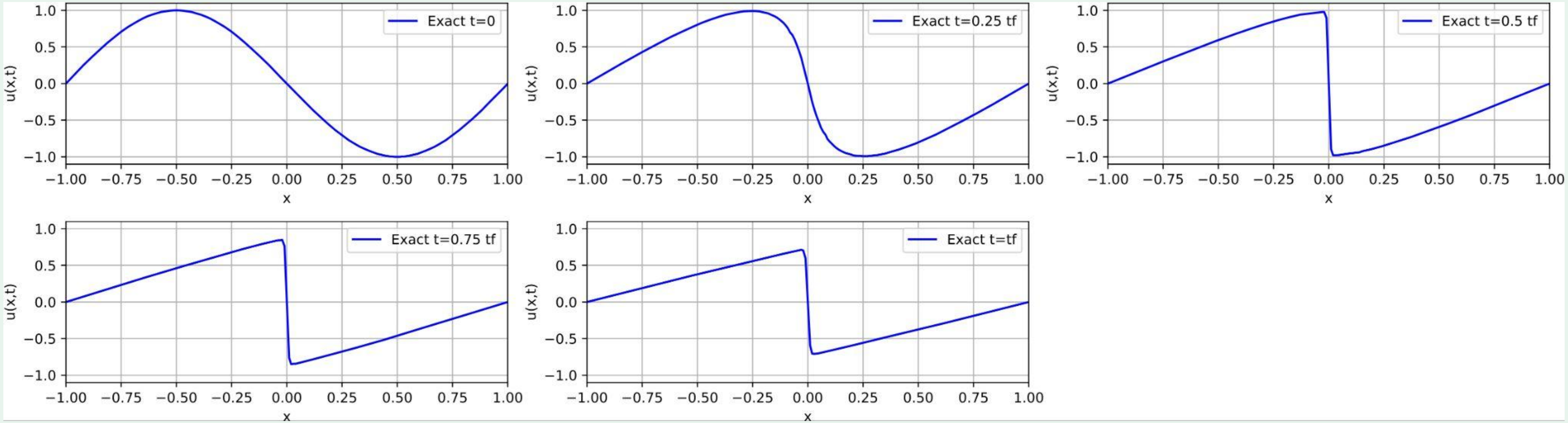
J Sci Comput 92, 88 (2022)

J. Comput. Phys., 378 (2019) 686

Burgers' equation and its solution

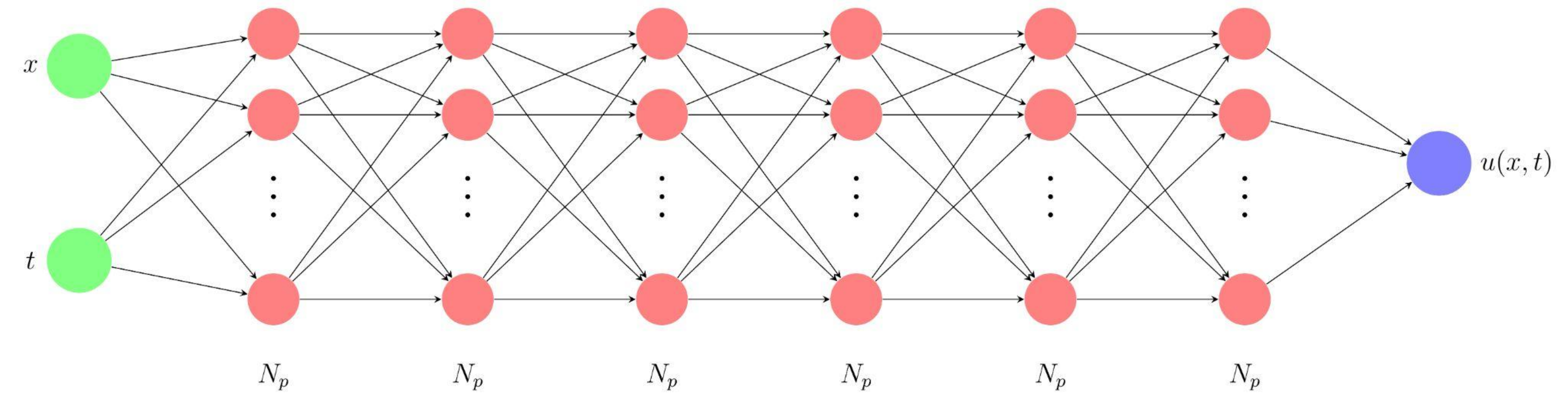
$$\frac{\partial u(x,t)}{\partial t} + u\frac{\partial u(x,t)}{\partial x} = \nu\frac{\partial^2 u(x,t)}{\partial x^2}, \text{with } \nu \geq 0,$$

nonlinear PDE that can be found in areas of fluid dynamics, nonlinear acoustics, gas dynamics, and traffic flow



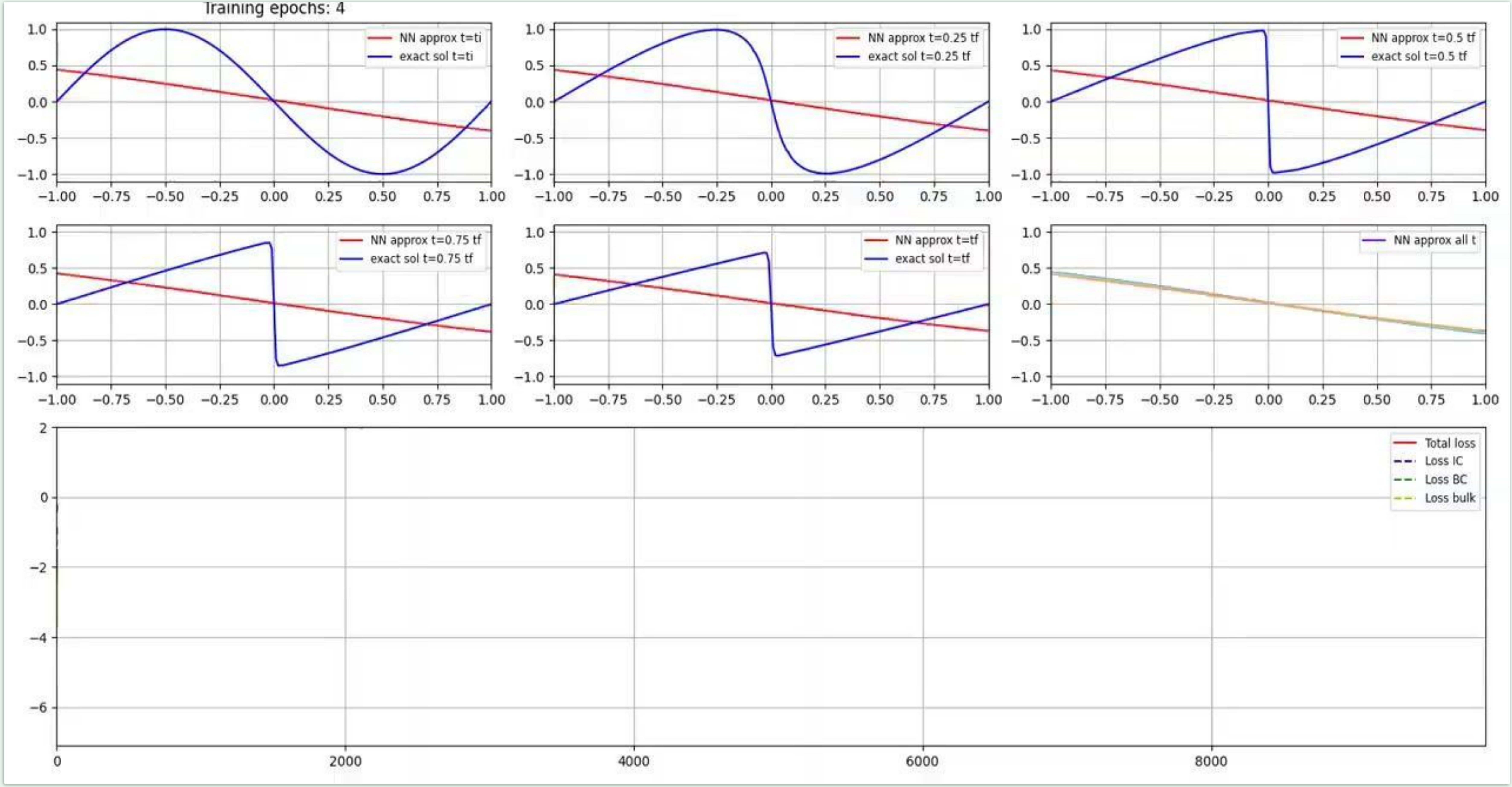
Solution of the Burgers' equation at five different time steps, $t = 0.0, 0.25, 0.5, 0.75, 1.0$, using the boundary conditions $u(-1,t) = 0, u(1,t) = 0$ and the initial condition $u(x, t=0) = -\sin(\pi x)$. The formula for the exact solution can be found in [C. Basdevant *et al*, Computer & fluids 14 (1986) 23–41]

Digital NN layout

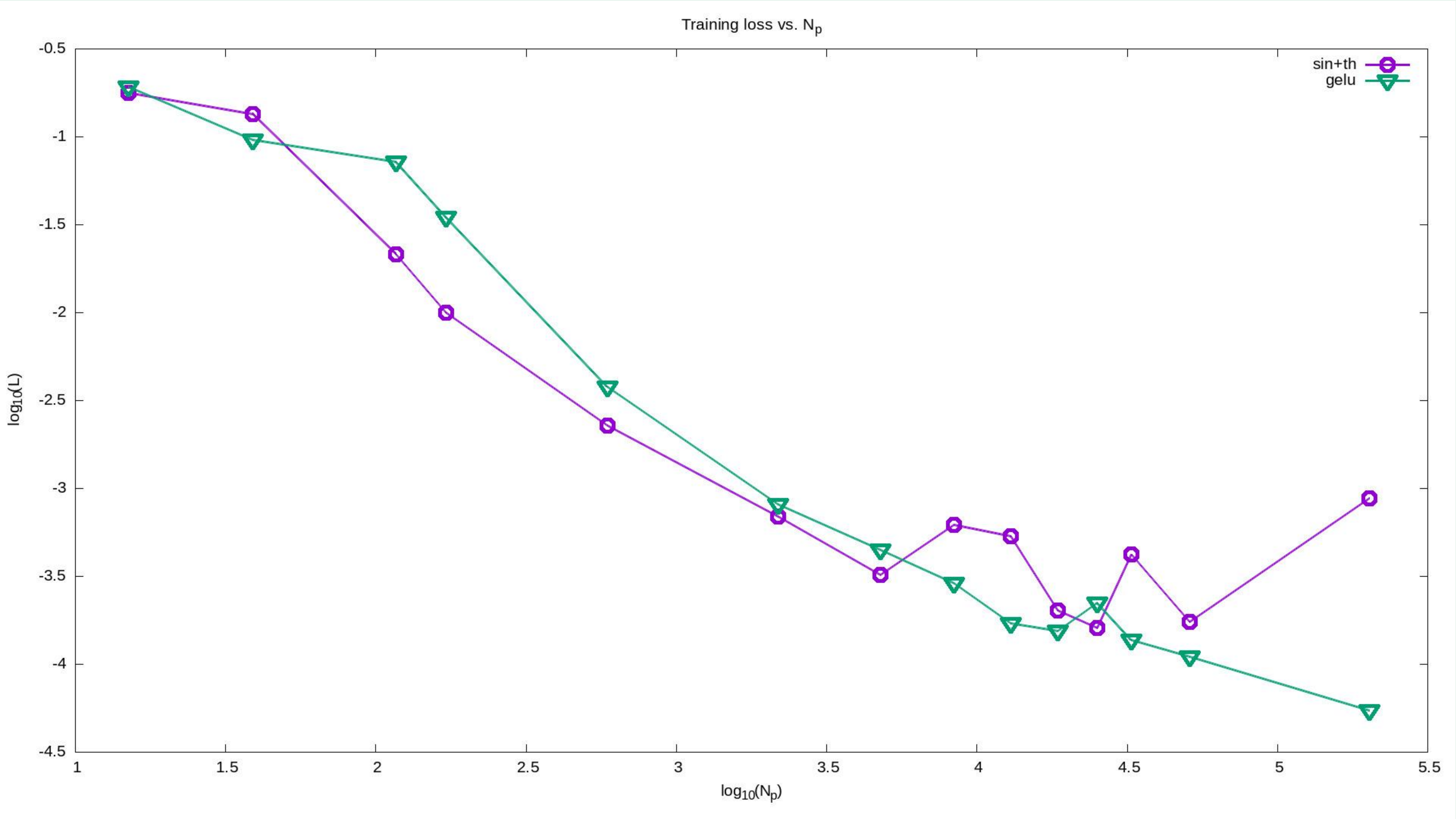


- The input points (x, t) for the BC and IC were chosen to be equidistant with size 200
- 5000 collocation points were sampled from the uniform distribution
- A sin activation function for the first three hidden layers, while $tanh$ for the last three layers
- The number of perceptrons in the hidden layer N_p was changed from 2 to 50 to check the expressivity
- The total number of parameters ranged from 39 to 12951
- We used TensorFlow with the Adam optimizer for training, learning rate was 10^{-3}
- Number of epochs = 5000–10000

Comparison of the exact and NN solution and the training curve for the classical digital NN



Change of Tot Loss min with N_p



- N_p is changed from 1 to 200
- $N_{\text{tot}} = 5N_p^2 + 9N_p + 1$
- N_{tot} from 15 to 201801
- # of epochs = 8000
- sin+th shows a plateau, while GELU still decreases

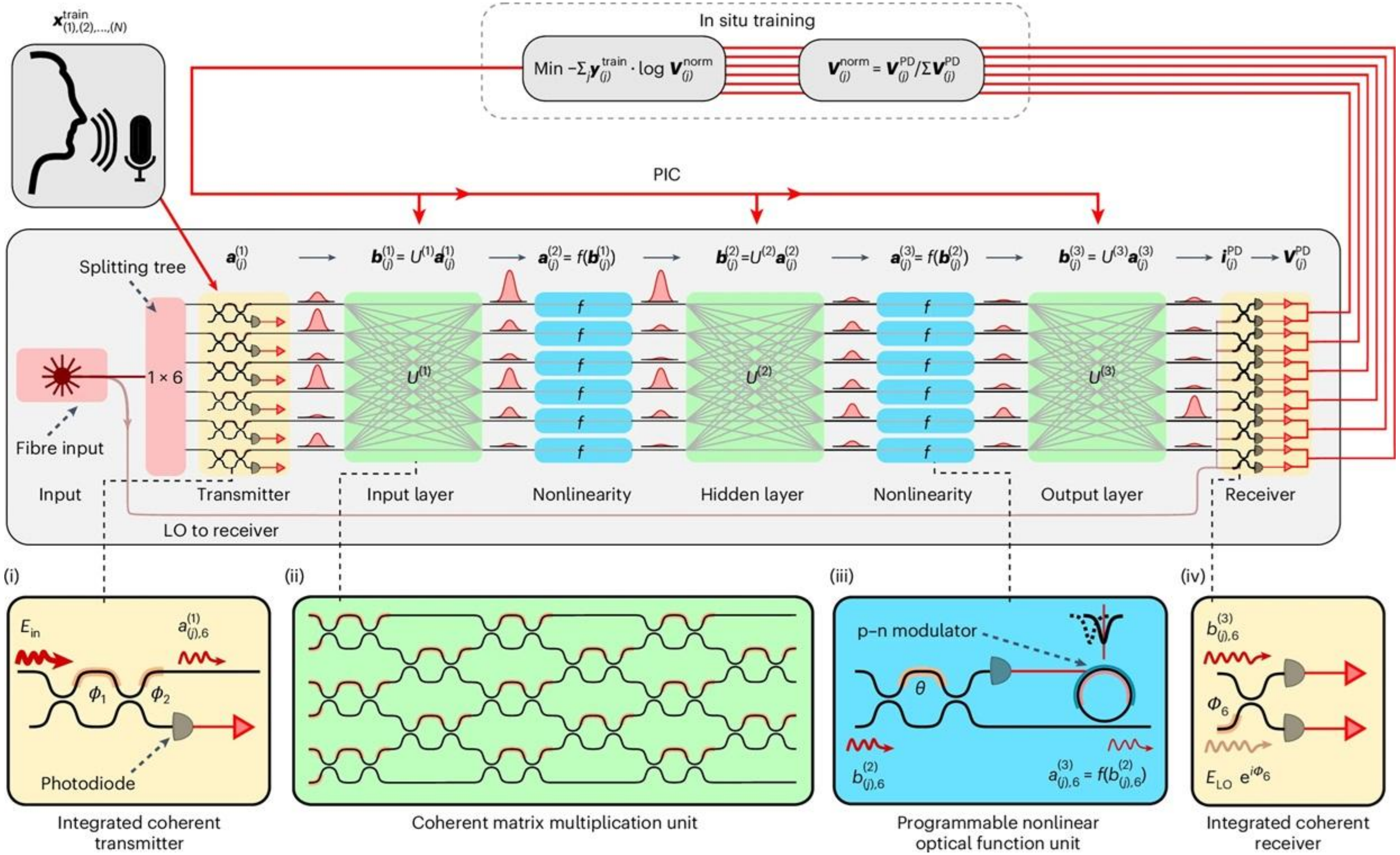
$$\mathrm{GELU}(x) = x \cdot \Phi(x) = x \cdot \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

In practice it is used as:

$$\mathrm{GELU}(x) \approx 0.5\,x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\,(x + 0.044715\,x^3)\right)\right)$$

Photonic NN, FICONN

FICONN = fully integrated coherent optical neural network



I. the amplitude and phase of $x_{(j)}$ is encoded into the optical field $a^{(1)}_{(j)}$

II. MZI (Mach–Zehnder interferometer) mesh that implements a unitary transformation

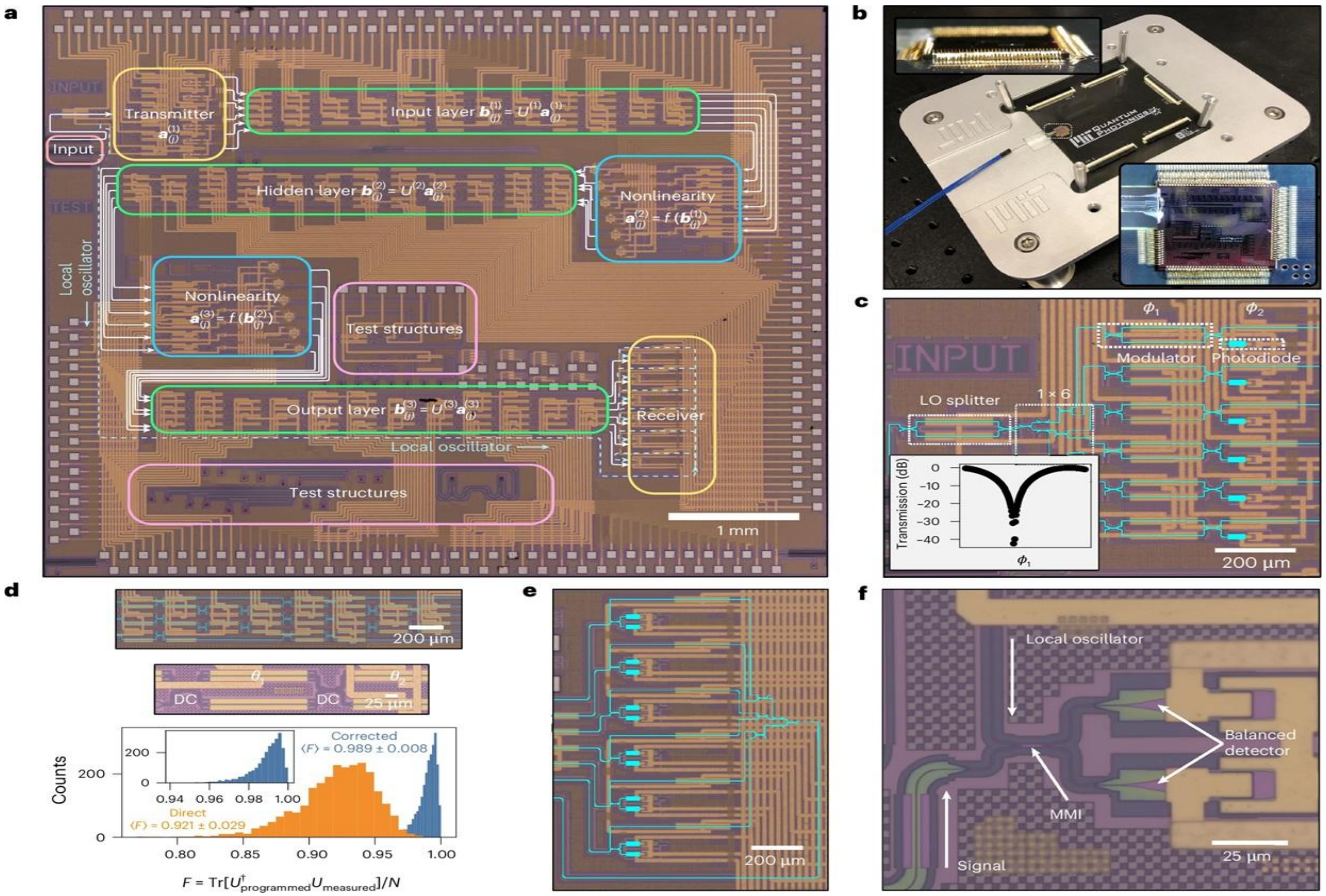
III. A NOFU (nonlinear optical function unit) realize activation function by tapping off part of the signal. Various functions can be realized

IV. An ICR (integrated coherent receiver) reads out the result of the DNN with a local oscillator field

[Nat. Photon. 18, 1335–1343 (2024)]

Fabricated PIC

PIC = photonic integrated circuit



a. Microscopic image of the fabricated PIC, colors refer to the same units as in prev. schematic figure

b. Photonic packaging for testing

c. Splitting the input signal to six input channel

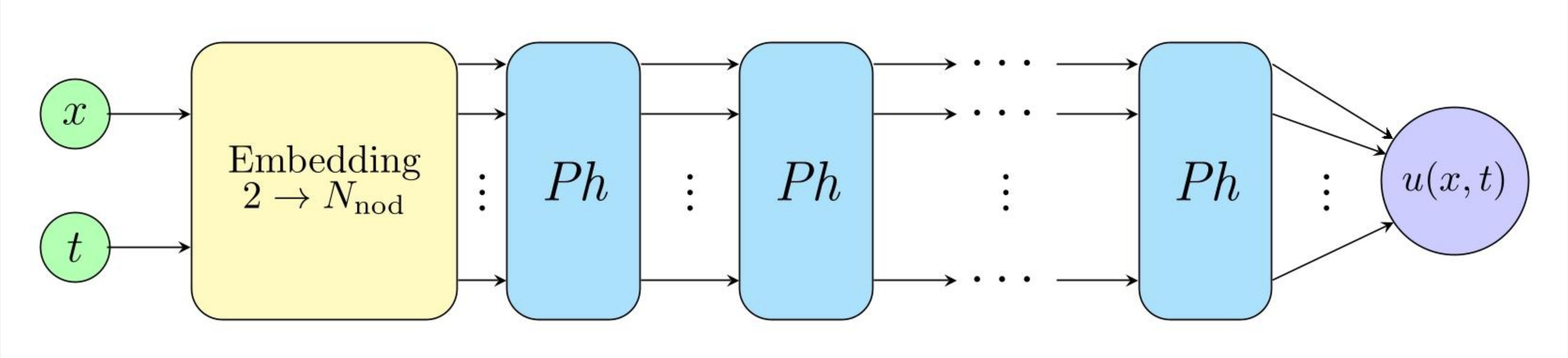
d. CMXU (coherent matrix multiplication unit), i.e. an MZI mesh

e. ICR (integrated coherent receiver), i.e. the readout

f. One channel of the readout

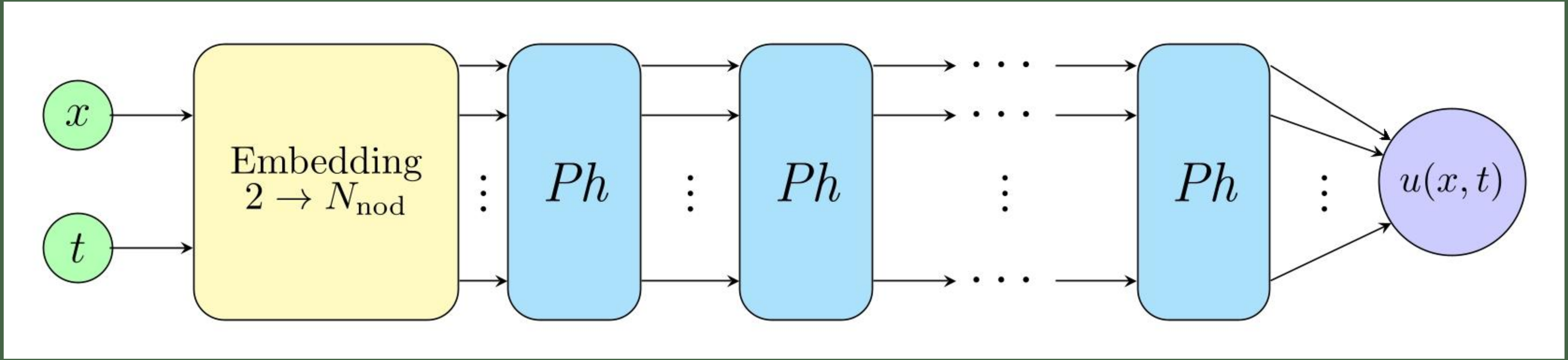
[Nat. Photon. 18, 1335–1343 (2024)]()

Photonic NN layout



- Number of inputs, N_{nod}, must be equal to the number of outputs, so an embedding layer is needed that transforms the input data into vectors of dimension N_{nod}
- Different possibilities for the embedding layer: spatial multiplexing, Fourier feature layer
- The hidden layers Ph realize a transformation of the form: $U_1 \times D \times U_2$ plus some nonlinearities acting independently on each node.
- The term $U_1 \times D \times U_2$ represents an SVD decomposition of an arbitrary real matrix. Realized with [Piquasso]
- The predicted function $u(x,t)$ is the real part of the sum of optical outputs.

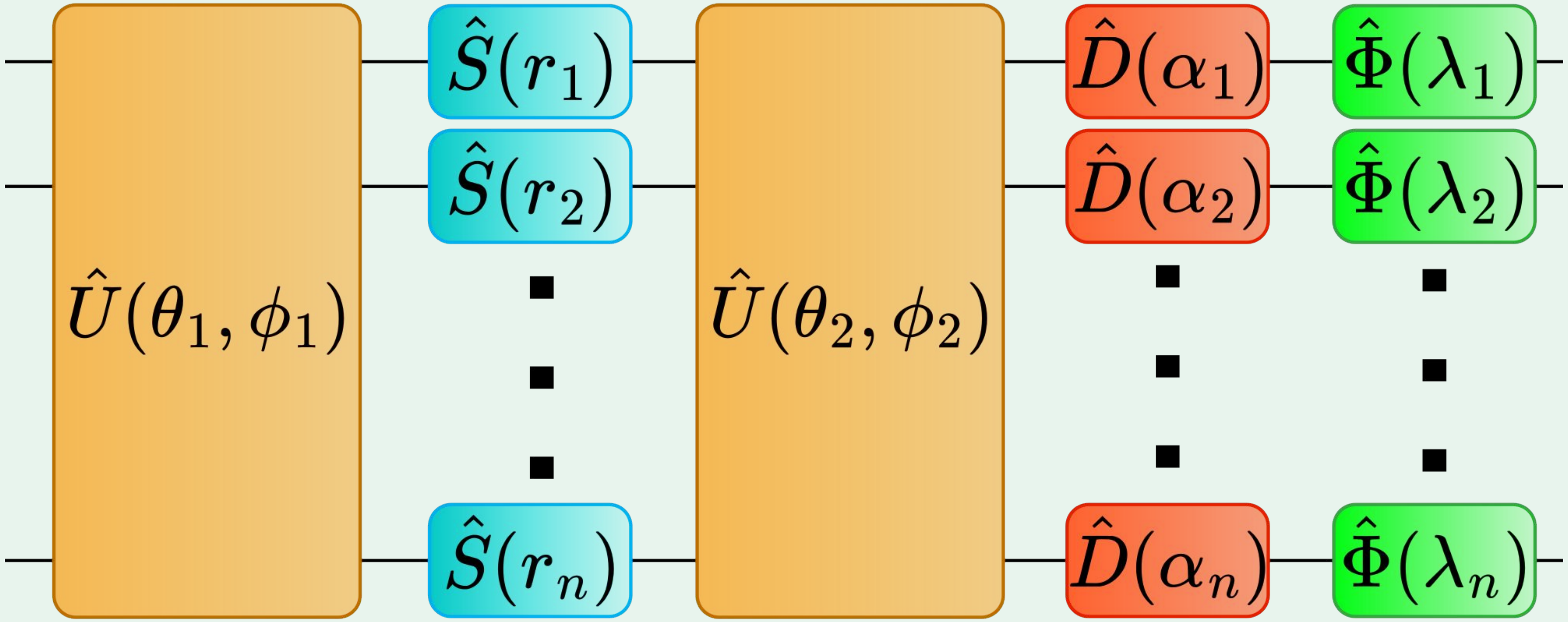
What are the advantages of photonic NNs?



- Light travels at lightspeed 😊. Photonic circuits can operate at terahertz (THz) frequencies, far beyond GHz speeds of CPUs/GPUs.
- Ultra-Low Power Consumption, passive photonic components (like beam splitters and phase shifters) consume little or no power (even factor of 10^{-5} compared to digital circuits)
- Low Latency: photonic circuits can compute almost instantly as light passes through the network
- Massive parallelism: Wavelength Division Multiplexing (WDM); different data on different wavelengths
- Reduced Heat Generation: less heat means better scalability and reduced need for cooling,

CVQNN as QPINN

CVQNN = continuous variable quantum neural networks
QPINN = quantum physics informed NN



arXiV:2503.12244v1

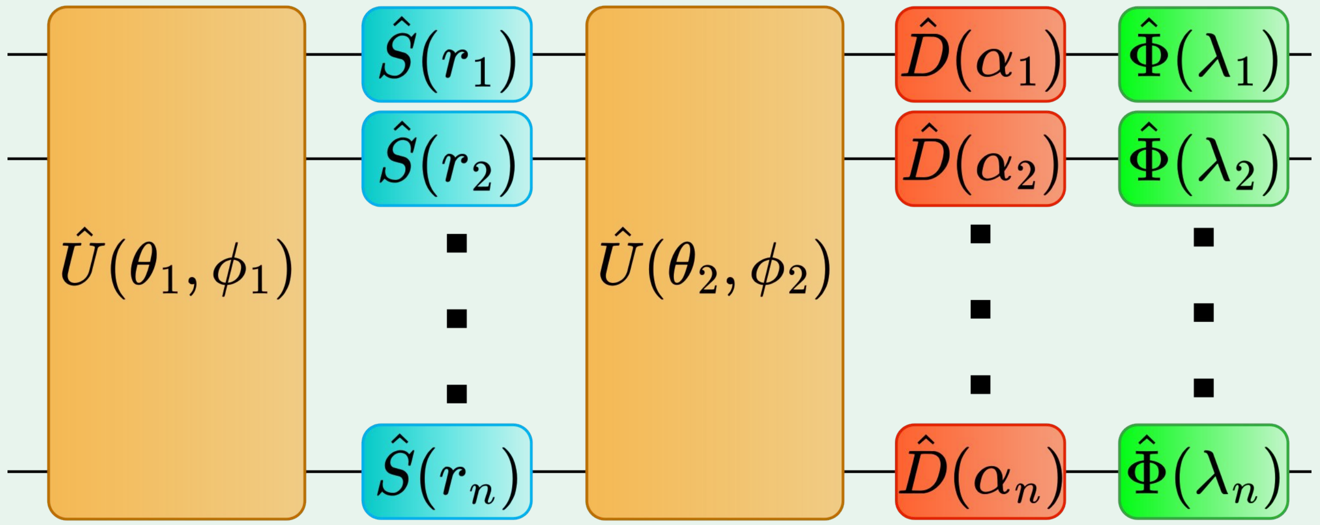
Phys. Rev. Res. 1, 033063 (2019)

- CVQNN can be considered as an analog of classical photonic NNs
- Information is encoded into continuous values of x (position) and p (momentum) in *wave-function* or *phase space* representations
- There are *Gaussian* and gates like rotation $\hat{R}(\phi)$, displacement $\hat{D}(\alpha)$ and squeezing $\hat{S}(r)$ (single-mode), and the beamsplitter $\hat{BS}(\theta)$ (two-mode)
- And non-Gaussian gates: cubic phase gate $\hat{V}(\gamma)$ and the Kerr gate $\hat{K}(\kappa)$

Some benefits of QPINN over classical photonic NN

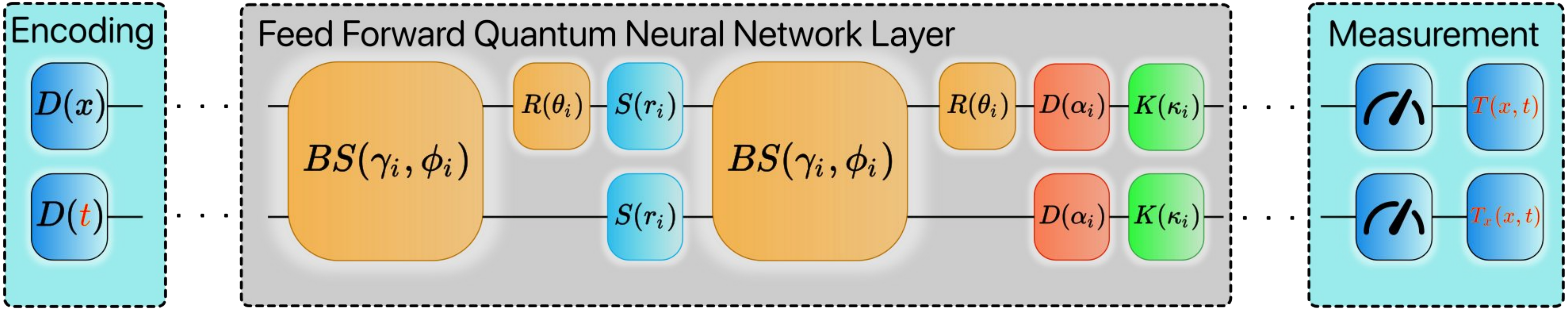
arXiV:2503.12244v1

- **High-dimensional feature space:** exponentially large Hilbert spaces. Even a small number of modes can encode extremely rich features
- **Intrinsic Nonlinearity via Measurement:** measurement serve as a built-in "activation" mechanism without needing physical nonlinear elements



- **Quantum Entanglement:** Entangled photonic modes can capture correlations and patterns across inputs in ways that are hard to simulate classically.
- **Reduced Circuit Depth via Superposition:** A quantum circuit can process a superposition of many input states simultaneously, consequently many-many training points can be processed in parallel
- **Enhanced Expressivity:** Empirically, small-scale quantum circuits have been shown to match or exceed the representational power of much larger classical networks when constrained to the same count of physical components

Quantum photonic layout



- One realization of QPINN for solving the Heat−equation from arXiV:2503.12244v1
- x and t coordinates of the input collocation points (+BC+IC) are encoded using the displacement gates
- In the output $\hat{X}$ is measured, the two output channel corresponds to the solution and its spatial derivative
- The total loss contains the following terms: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{PDE}} + \lambda_2 \mathcal{L}_{\text{IC}} + \lambda_3 \mathcal{L}_{\text{BC}} + \lambda_4 \mathcal{L}_{\text{consis}} + \lambda_5 \mathcal{L}_{\text{Tr}}$
- Extra terms for consistency, i.e. the second output should correspond to the derivative of the first output
- And for trace, i.e. for normalization of the state vector
- Increased weight (λ_2) for the initial condition (60%)

Summary and outlook

- Classical digital, classical photonic, and quantum photonic neural networks can all be used for solving PDEs.
- Each has its own benefits, so it's worth comparing them in detail for specific tasks.
- Classical and quantum photonic setups were presented alongside the well-working classical digital setup.

Summary and outlook

- Classical digital, classical photonic, and quantum photonic neural networks can all be used for solving PDEs.

- Each has its own benefits, so it's worth comparing them in detail for specific tasks.

- Classical and quantum photonic setups were presented alongside the well-working classical digital setup.

- There is no parameter shift rule for the CVQNN layer, so there is no good method for calculating derivatives.
- Interpolation might be better suited to approximate derivatives. ⟶ Zoltán Kolarovszki’s talk
- Work in progress ...

Thank you for your attention!

Email
kovacs.peter@wigner.hun-ren.hu

Slides Carnival

for this presentation template

Pexels, Pixabay, Sketchify

for the photos, graphics, and elements