

# Reconfigurable Integrated Photonic Unitary Neural Networks With Phase Encoding Enabled by In-Situ Training

Shengjie Tang , Cheng Chen , Qi Qin , and Xiaoping Liu , *Member, IEEE*

**Abstract**—Photonic neural networks are emerging as promising computing platforms for artificial intelligence (AI). Particularly, integrated photonic unitary neural networks (IPUNNs) are capable of mitigating gradient vanishing/explosion problems when deeper neural networks are constructed. Furthermore, their optical implementations are also much simpler compared to non-unitary counterparts. Meanwhile, real-valued datasets still dominate AI research and the encoding strategy is critical for IPUNNs' performances. However, there are few studies to compare different encoding strategies of IPUNNs to represent these real-valued datasets and their impacts on IPUNNs' performances. Here, in the scope of encoding strategies for real-valued features, we first compare different schemes, such as phase, amplitude and hybrid encoding using numerical simulations, with benchmarks of decision boundary and image recognition tasks. These encoding strategies of IPUNNs are also compared to non-unitary real-valued neural networks (RVNNs) with trainable biases for the same benchmarks. The results suggest that phase encoding outperforms amplitude and hybrid encoding, and exhibits comparable performances to non-unitary RVNNs. To verify the numerical results, a  $10 \times 10$  IPUNN chip is designed and fabricated. The phase encoding is chosen to be implemented because of its superior performances in numerical studies. We reconfigure the IPUNN chip to perform decision boundary and image recognition tasks by on-chip in-situ training. The experimental results match the simulations well. Our work provides insights for implementing reconfigurable IPUNNs in AI computing.

**Index Terms**—In-situ training, phase encoding, photonic unitary neural network, reconfigurable.

## I. INTRODUCTION

THE rapid development of artificial intelligence (AI) has led to an exponential growth in demand for computing

Received 1 August 2024; revised 26 August 2024; accepted 29 August 2024. Date of publication 3 September 2024; date of current version 13 September 2024. This work was supported in part by the Science and Technology Commission of Shanghai Municipality under Grant 21DZ1101500, in part by startup funding from ShanghaiTech University, in part by Guangdong Provincial Quantum Science Strategic Initiative under Grant GDZX2306001 and Grant GDZX2303001, and in part by startup funding from Shenzhen University and Shenzhen City. (*Corresponding authors: Qi Qin; Xiaoping Liu.*)

Shengjie Tang, Cheng Chen, and Xiaoping Liu are with the School of Physical Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: tangshj@shanghaitech.edu.cn; chencheng@shanghaitech.edu.cn; liuxp1@shanghaitech.edu.cn).

Qi Qin is with the College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China, and with the State Key Laboratory of Radio Frequency Heterogeneous Integration, Shenzhen University, Shenzhen 518060, China, and also with the Quantum Science Center of Guangdong-Hong Kong-Macao Greater Bay Area (Guangdong), Shenzhen 518045, China (e-mail: qi.qin@szu.edu.cn).

Digital Object Identifier 10.1109/JPHOT.2024.3453898

power in recent years [1]. Meanwhile, conventional electronic computing platforms are approaching their physical limits [2], which makes it challenging to continue to keep up with AI's ongoing development. This plight prompts researchers to seek the next-generation high performance computing platforms for AI. Optics has been widely recognized as a promising medium to implement large-scale neural networks in AI due to its intrinsic parallelism, low latency, large bandwidth and high energy efficiency [3]. To overcome the shortcomings of bulk optical components, the programmable integrated photonic neural network using coherent light was proposed as a scalable and phase-stable hardware solution [4]. The essential part of current widely used architectures for coherent integrated photonic neural networks is composed of specific arrays of Mach-Zehnder interferometers (MZIs) with different mesh topologies [5], [6], [7]. These MZI mesh-based architectures realized on photonic integrated circuits (PICs) can perform universal unitary transformations at the speed of light, which have widespread applications in quantum computations [8], [9], [10], [11] and neural networks [4], [12], [13], [14], [15]. In the design of integrated photonic neural networks, the MZI mesh-based architectures have been successfully used to construct real-valued neural networks (RVNNs) [4], [13], [16], where all the weights are real-valued matrices, aligning with the existing mainstream AI models primarily ground on real-valued arithmetic. The photonic complex-valued neural networks proposed by Zhang et al. have the weights comprising non-unitary complex-valued matrices and improve the capability of photonic neural networks to achieve more complex modeling [14]. When mapping an arbitrary real-valued and complex-valued matrix to unitary photonic devices, they need to be decomposed into a product of two unitary matrices and a diagonal matrix according to singular value decomposition (SVD) [5], [17], [18]. Correspondingly, the construction of an arbitrary real-valued and complex-valued weight matrix requires an additional unitary photonic mesh and an array of optical attenuators/amplifiers compared to a pure unitary weight matrix. Moreover, the coherent detection is inevitable in photonic complex-valued neural networks to obtain complex-valued features, which complicates the detection procedure and increases the energy cost of detecting optical signals. As a result, exploiting unitary photonic meshes to construct neural networks with all weights being unitary matrices, namely unitary neural networks (UNNs), has lower hardware and operational complexity compared to RVNNs and complex-valued neural networks.

It is worth noting that UNNs are special instances of RVNNs and complex-valued neural networks because unitary matrices can be implemented in both real and complex domains. For simplicity and clarity of the discussion, the RVNNs (real-valued matrices) and complex-valued neural networks (complex-valued matrices) in this paper are all indicated as non-unitary ones to distinguish from UNNs (unitary matrices).

Recently, UNNs have garnered increased attention owing to their adaptability for construction on PICs without the need for gain and loss mechanisms [19], [20], [21], and their ability to mitigate gradient vanishing/explosion problems in deep neural networks [22], [23]. Despite this, one should also be aware that when confronting highly complex tasks, the performance of UNNs may be degraded to some extent owing to their smaller parameter spaces. The utilization of promising avenues, such as introducing optical biases for unitary photonic meshes [24] and deploying hybrid integration with other efficient photonic computing architectures [25], could alleviate this issue in photonic neural networks, which merits further exploration in future research. Here, we will focus on the integrated photonic neural networks that basically implement UNN models, i.e., integrated photonic unitary neural networks (IPUNNs). Combining the advantages of optics and UNNs makes IPUNNs a potential platform for expediting deep learning computations. Currently, the vast majority of datasets or features for deep learning are based on real-valued representations. Real-valued features are often encoded by the amplitude or power of optical signals in MZI mesh-based architectures [4], [12], [13], [16], but systematic analyses of various encoding strategies and their impacts on IPUNNs' performances are seldom explored. Despite the fact that various encoding and detection methods have been investigated for photonic complex-valued neural networks, the comparative studies are specifically designed for complex-valued features, while only magnitude encoding is adopted for real-valued features [14]. Additionally, when handling images related tasks, a common encoding technique known as the Fourier transform is often utilized. Previous works typically utilize the portion of low-frequency Fourier coefficients as complex-valued input features when dealing with images to reduce the input dimension of IPUNN chips [15], [19], [21], [26]. However, for complex images with numerous rapidly varying spatial features, taking only small subsets of Fourier coefficients around the zero-frequency component will lead to a nonnegligible performance degradation. Besides, the on-chip optical implementation of Fourier transform requires using additional photonic devices [20], [27], which is not conducive to large-scale integration in terms of fabrication tolerance and footprints. Overall, an appropriate encoding scheme is significant for fully exploiting the potential of IPUNNs in AI acceleration.

In this work, numerical simulations are first conducted to investigate the impact of different encoding schemes for real-valued features on the performance of UNNs, using benchmarks of decision boundary and image recognition tasks. Considering that the majority of existing models in the AI field are primarily based on real-valued computations, reference RVNNs with trainable biases are also studied for comparisons. Simulation results show that phase encoding outperforms amplitude and

hybrid encoding in UNN models, demonstrating comparable performances to RVNNs with trainable biases. Furthermore, a  $10 \times 10$  IPUNN chip is designed and fabricated for experimental verifications. On the basis of feature extraction and on-chip in-situ training, we effectively reconfigure the IPUNN chip to specific states capable of performing diverse machine learning (ML) tasks. From the following results presented in this work, we hope to provide a reference and insights for the application and implementation of reconfigurable IPUNNs in AI computing.

## II. NUMERICAL SIMULATION RESULTS

In this section, numerical simulations of several ML benchmarks, decision boundaries for nonlinear datasets and image recognition tasks, are performed to demonstrate the performance of UNNs with different encoding schemes. With an Intel Xeon CPU E5-2680 v4 @ 2.40GHz and a NVIDIA GPU GeForce GTX 1080Ti, all numerical simulations are performed using Python 3.8.10 and TensorFlow 2.11.0 running on a Linux operating system. Based on the previously explored photonic neural network framework [15], we investigate the architectures apt for physical implementations, which comprise incoherent detection based nonlinear activations and multilayer photonic unitary networks. For the numerical simulation of UNNs, we construct trainable unitary layers by multichannel mixing blocks setup [28]. Each  $N \times N$  trainable unitary weight matrix is decomposed into  $N + 1$  cascaded building blocks where each building block consists of a random (but fixed during training)  $N \times N$  unitary matrix and  $N$  trainable and independent phase parameters, which can be expressed as:

$$U(N) = \prod_{k=0}^N V_{(N+1-k)} \Theta(\vec{\theta}_{(N+1-k)}), \quad (1)$$

where  $V_{(N+1-k)}$  is the random  $N \times N$  unitary matrix of the  $(N + 1 - k)^{th}$  building blocks,  $\vec{\theta}_{(N+1-k)} = (\theta_{(N+1-k)}^{(1)}, \theta_{(N+1-k)}^{(2)}, \dots, \theta_{(N+1-k)}^{(N)})$  is the vector with  $N$  trainable and independent phase parameters corresponding to the transfer diagonal matrix  $\Theta(\vec{\theta}_{(N+1-k)}) = \text{diag}[\exp(j\theta_{(N+1-k)})]$  in the  $(N + 1 - k)^{th}$  building blocks.

### A. Decision Boundaries of Nonlinear Datasets

We first simulate decision boundary tasks to intuitively demonstrate the discrepancies in expressivity among various encoding strategies. The benchmarks of decision boundaries are two nonlinear datasets: the 2-classes Moon and 3-classes Spiral used in [29]. The nonlinear datasets have two real-valued features  $x_1, x_2 \in (0, 1)$ . We randomly split the 1000 samples in the Moon dataset into 600 training samples and 400 testing samples. Whereas, the Spiral dataset are randomly divided into 900 training samples and 600 testing samples from 1500 samples in total. As shown in Fig. 1(a), there are two layers of trainable unitary weight matrices in the numerical simulation model for decision boundary tasks.

The input vector  $[x_1, x_2]$  is first projected to a ten-dimensional space by a zero-padding operation, an encoding layer and a

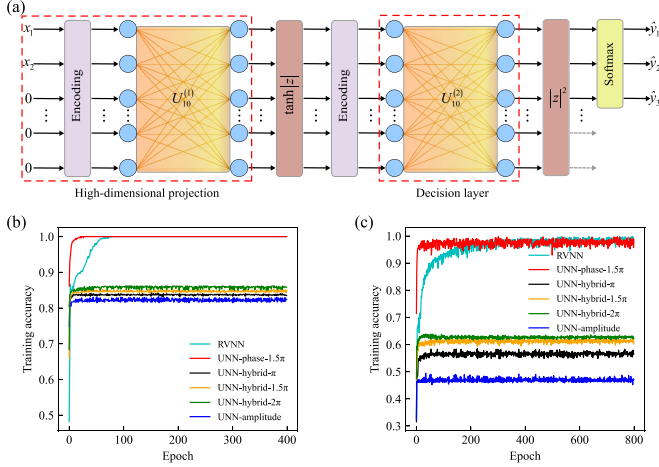


Fig. 1. (a) The architecture diagram of the numerical model for decision boundary tasks. Training accuracies of the (b) Moon and (c) Spiral datasets during training for UNNs with various encoding schemes and the reference RVNN. The optimum phase-product factor of phase encoding is  $1.5\pi$ , while for hybrid encoding, it is set to  $\pi$ ,  $1.5\pi$  and  $2\pi$  for comparisons.

trainable  $10 \times 10$  unitary layer named as  $U_{10}^{(1)}$ . The encoding layer carries out different element-wise encoding schemes for comparisons. This is followed by an activation function:  $\tanh|z|$ , which is applied on the complex output of the first unitary layer to implement nonlinear activation and normalize features to  $[0, 1)$ . Furthermore, these normalized features are then encoded and fed to the second trainable  $10 \times 10$  unitary layer named as  $U_{10}^{(2)}$ . The output of  $U_{10}^{(2)}$  is followed with a nonlinear activation  $|z|^2$ . Both  $|z|$  and  $|z|^2$  can be experimentally realized by a simple incoherent detection. Ten-dimensional outputs are dropped to two ports (for the Moon) or three ports (for the Spiral) and a softmax function is applied to generate probability distribution. For a given arbitrary normalized real-valued feature  $x$ , amplitude encoding is represented as  $A_0|x|$ , where  $A_0$  is a positive constant. By loading features into the phase domain, phase encoding is represented as  $A_0e^{j*x*p\pi}$ , where  $p\pi$  is a phase-product factor used to extend coding to an appropriate phase range. Hybrid encoding integrates the above two operations, namely  $A_0|x|e^{j*x*p\pi}$ . The value of  $A_0$  is considered as 1 for every operation in this paper. A larger phase-product factor will lead to a larger Euclidean distance among entangled samples in the complex space, which could bring a higher classification accuracy. However, if  $2\pi$  is used as a phase-product factor for phase encoding, the discrete points around 0 and 1 that are not included in training and testing sets will tend to be predicted as the same category incorrectly, since  $A_0e^{j*0*2\pi} = A_0e^{j*1*2\pi}$ , which could result in ambiguity and deteriorated generalization performance. For these two nonlinear datasets,  $1.5\pi$  is chosen as a phase-product factor for phase encoding taking both generalization and classification accuracies into account. For comparisons, the phase-product factor of hybrid encoding is set to  $\pi$ ,  $1.5\pi$  and  $2\pi$ .

We also consider the reference RVNN that has the same architecture and configuration (topology and activation functions) as the UNN model depicted in Fig. 1(a), with the exception of

TABLE I  
PERFORMANCES OF THE UNNS WITH VARIOUS ENCODING SCHEMES AND THE REFERENCE RVNN FOR DECISION BOUNDARY TASKS

Model	Moon		Spiral	
	Training	Testing	Training	Testing
RVNN	100.00%	100.00%	99.78%	99.50%
UNN-phase- $1.5\pi$	100.00%	100.00%	99.78%	99.67%
UNN-hybrid- $\pi$	84.67%	79.25%	58.44%	52.33%
UNN-hybrid- $1.5\pi$	85.17%	80.00%	62.44%	57.50%
UNN-hybrid- $2\pi$	86.17%	81.50%	63.78%	59.50%
UNN-amplitude	83.33%	78.00%	49.44%	48.50%

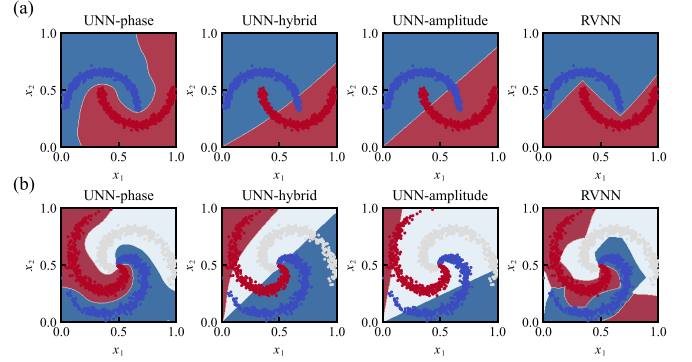


Fig. 2. Simulated decision boundaries of the (a) 2-classes Moon dataset and (b) 3-classes Spiral dataset for UNNs with different encoding schemes and the reference RVNN. The blue and red in data points and regions respectively represent the classes 0, 1 for the Moon dataset, whereas the blue, white and red in data points and regions represent the classes 0, 1, 2 for the Spiral dataset, respectively. Here, the phase-product factor of phase encoding and hybrid encoding is  $1.5\pi$  for visualization.

replacing each  $10 \times 10$  unitary weight matrix by a learnable  $10 \times 10$  real-valued weight matrix and a learnable  $10 \times 1$  real-valued additive bias vector. Note that encoding layers for RVNNs are equivalent to amplitude encoding for all tasks performed in this paper. We use the standard categorical cross-entropy [30] as the loss function (LF) for training. The Adaptive moment estimation (Adam) gradient descent method [31] is used to minimize the LF and update the trainable parameters for all models. The learning rate for the Moon and Spiral dataset is set to 0.005 and 0.008, respectively. Both the UNNs and RVNNs are trained with a batch size of 60. The predicted category of each sample is determined by the index where the maximum output value occurs. Training convergence curves of all numerical models are shown in Fig. 1(b) and Fig. 1(c) for the Moon dataset and the Spiral dataset, respectively. Table I shows the classification accuracies for the Moon and Spiral datasets after various numerical models being trained. It can be seen that phase encoding demonstrates significant superiority compared to other encoding schemes for UNN models.

After the models are trained,  $x_1$  and  $x_2$  are uniformly sampled from 0 to 1 with a step of 0.005 to produce total 40401 discrete points. The 40401 discrete points are then fed into the trained models corresponding to the maximum accuracy of training sets to generate decision boundaries. The predicted decision boundaries are shown in Fig. 2. The blue and red in data points and regions respectively represent the classes 0, 1 for

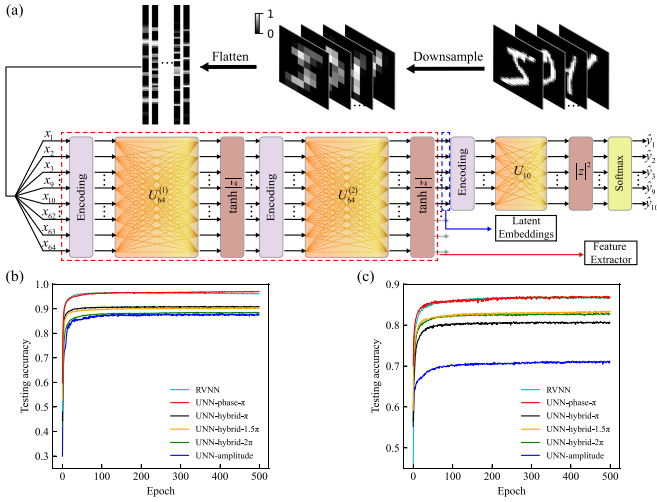


Fig. 3. (a) The architecture diagram of the numerical model for image recognition tasks. Testing accuracies of the (b) MNIST and (c) Fashion-MNIST for UNNs with various encoding schemes and the reference RVNN. The optimum phase-product factor of phase encoding is  $\pi$ , while for hybrid encoding, it is set to  $\pi$ ,  $1.5\pi$  and  $2\pi$  for comparisons.

the Moon dataset, whereas the blue, white and red in data points and regions represent the classes 0, 1, 2 for the Spiral dataset, respectively. These notation methods also apply to the following in-situ training experiments.

As shown by the decision boundaries generated by trained models, phase encoding provides nonlinear boundaries that are capable of separating entangled samples almost perfectly. Even with nonlinear activation functions, the decision boundaries offered by amplitude encoding are almost straight. Hybrid encoding produces certain nonlinear boundaries compared with amplitude encoding, but the overall performance is not improved over pure phase encoding.

### B. Image Recognition

To further verify the performances of aforementioned models in more complex datasets, we also conduct simulations for image recognition tasks. The MNIST (Modified National Institute of Standards and Technology) handwritten digit dataset [32] and the Fashion-MNIST dataset [33], each consisting of 60,000 training images and 10,000 testing images corresponding to ten categories, are utilized as benchmarks. All 60,000 training images are used for training and the numerical model architecture for image recognition tasks is shown in Fig. 3(a).

The simulation is performed with the following procedure. Original  $28 \times 28$  grayscale images are resized to  $8 \times 8$  pixels via down-sampling and then pixel values of down-sampled images are divided by 255 to be normalized into the  $[0,1]$  interval. The down-sampling operation is implemented with “INTER\_AREA” method in OpenCV [34]. Subsequently, each down-sampled image is flattened into a  $64 \times 1$  vector, of which the size is compatible with the commercial photonic computing platform recently reported [35]. Similar to the method used in [14], [36], [37], we also employ a fully connected network to extract low-dimensional features. The fully connected network

TABLE II  
PERFORMANCES OF THE UNNS WITH VARIOUS ENCODING SCHEMES AND THE REFERENCE RVNN FOR IMAGE RECOGNITION TASKS

Model	MNIST	Fashion-MNIST
	Testing accuracy	Testing accuracy
RVNN	96.68%	87.03%
UNN-phase- $\pi$	97.10%	87.06%
UNN-hybrid- $\pi$	90.99%	80.85%
UNN-hybrid- $1.5\pi$	90.32%	83.44%
UNN-hybrid- $2\pi$	88.56%	82.95%
UNN-amplitude	88.19%	71.23%

consists of two trainable  $64 \times 64$  unitary weight matrices, each being followed by a  $\tanh |z|$  activation function. Prior to executing the matrix-vector multiplication by each unitary weight matrix, all features are preprocessed through an encoding layer to execute various encoding schemes for comparisons. All the encoding layers execute the same operation when each model is conducted. The outputs of the feature extractor [depicted by the dashed red box in Fig. 3(a)] are then dropped to ten-dimensional vectors as the latent embeddings. The latent embeddings are encoded once again and fed to a  $10 \times 10$  UNN, emulating our optically implemented IPUNN chip. Similar to the manner used in decision boundary tasks, a nonlinear activation  $|z|^2$  followed by a softmax function is employed at the output and the categorical cross-entropy loss is utilized during training.

The reference RVNN with the same weight matrix dimensions and nonlinear activation functions as the UNNs has also been trained. Wherein, each  $64 \times 64$  unitary weight matrix is replaced by a learnable  $64 \times 64$  real-valued weight matrix added by a learnable  $64 \times 1$  real-valued bias vector. Accordingly, the  $10 \times 10$  layer executes analogous changes with the corresponding dimension. The Adam algorithm is used for training and the learning rate is set to 0.0005 with a batch size of 600. In image recognition tasks, the phase-product factor for phase encoding is chosen as  $\pi$  rather than  $2\pi$  to avoid ambiguity for pixel values around 0 and 1 after being mapped to phase domain. As to hybrid encoding, the phase-product factors of  $\pi$ ,  $1.5\pi$  and  $2\pi$  are studied for comparisons. Convergence plots on all the 10,000 testing images of MNIST and Fashion-MNIST are shown in Fig. 3(b) and (c), respectively. The final recognition accuracies over the complete testing set are provided in the Table II. The UNN with phase encoding outperforms amplitude and hybrid encoding in achieving higher testing accuracy, which shares comparable performances with the RVNNs for all the benchmarks used in numerical simulations. It is important to note that while most RVNNs implemented by MZI arrays based on SVD do not include bias calculations, we also simulate RVNNs containing trainable biases here. This allows for direct and objective comparisons with conventional digital RVNNs and optical RVNNs incorporating optical biases in the future. In fact, bias is likewise crucial for the performance of neural networks. Removing the adjustable bias vectors of RVNNs can result in performance degradation at some level. Therefore, from this perspective, IPUNNs with phase encoding may offer a certain

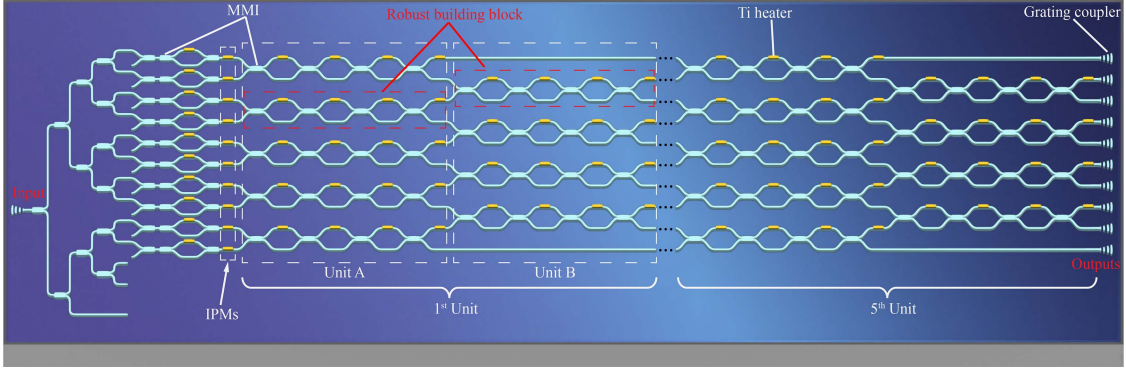


Fig. 4. Schematic illustration of the designed IPUNN chip. Input coherent light is split into 13 branches through four-stage  $1 \times 2$  MMI couplers with an equal optical path, top ten of them are fed into the unitary network. IPMs are used for input preparations. Five cascaded MZI network units form the  $10 \times 10$  unitary transformation architecture and the 2nd to 4th network units are omitted in the figure. Each MZI network unit is constructed with two types of subunits, denoted as unit A and unit B. Only the input and ten output grating couplers are displayed in the figure, while grating couplers of other unused ports are not depicted.

degree of advantages over RVNNs without optical biases in regard to model performances.

### III. DESIGN AND FABRICATION OF THE IPUNN CHIP

To experimentally verify the superior performance of UNNs with the phase encoding scheme, we designed an IPUNN chip which basically operates the reconfigurable  $10 \times 10$  unitary transformation, as schematically shown in Fig. 4. A coherent laser is coupled into the IPUNN chip from the input grating coupler using a fiber array (FA) and then split into 13 branches with an equal optical path length by four-stage  $1 \times 2$  multimode interference (MMI) couplers. The top 10 branches are used as the input ports of the unitary transformation architecture. Input preparation is realized by 10 input phase modulators (IPMs) by encoding real-valued features into phase domain. The architecture of our IPUNN chip obeys to the Clements method [6], but with a difference that each conventional MZI building block is replaced by a robust MZI building block to make the unitary network more robust to combat fabrication imperfections [38], [39]. As depicted by the dashed red box in Fig. 4, the robust MZI building block consists of four tunable phase shifters and four  $2 \times 2$  MMI-based beam splitters. Five cascaded MZI network units, each with 9 robust MZI building blocks, form a UNN architecture with a full-capacity configuration, i.e., the number of input modes is equal to the optical depth [22]. Each MZI network unit comprises two types of subunits, unit A and unit B, as depicted by the wider dashed white boxes in Fig. 4. The IPUNN chip contains 45 robust MZI building blocks and 180 trainable phase shifters in total. Output optical signals are coupled out from ten grating couplers through a FA.

The designed IPUNN chip was fabricated on a silicon-on-insulator (SOI) platform with a 220nm thick top silicon layer and  $2 \mu\text{m}$  thick buried oxide. The silicon waveguide layer is covered by a  $\text{SiO}_2$  upper-cladding. A layer of titanium (Ti) is deposited on the  $\text{SiO}_2$  upper-cladding as resistive heaters to perform thermo-optical modulations. Each heater has an electrical resistance about  $460 \Omega$  and is electrically connected to pads by patterned aluminum metal wires. The fabricated IPUNN chip is adhered and wire-bonded to a printed circuit board, allowing it

to be controlled by external multi-channel digital-to-analog converters (DACs). The packaged IPUNN chip and the micrograph of the fabricated IPUNN chip are shown in Fig. 5(b) and (c), respectively. We characterize the modulation response of IPMs, laying the groundwork for subsequent phase encoding operations in ML tasks. As an instance, the imparted phase dependent on the applied voltage  $v$  of the 5th IPM at 1540 nm wavelength is shown in Fig. 5(a) (measured at the 9th output port). We fit the transmission data using one-term Fourier series model to evaluate the voltage required to induce a phase change of  $2\pi$ , which is given by  $a_0 + a_1 \cos(wv^2) + b_1 \sin(wv^2)$ , where  $a_0$ ,  $a_1$ ,  $b_1$  and  $w$  can be obtained through fitting. Then the voltage of the corresponding modulator for  $2\pi$  phase shift ( $V_{2\pi}$ ) can be calculated by  $V_{2\pi} = \sqrt{\frac{2\pi}{|w|}}$ . Here,  $V_{2\pi}$  of the 5th IPM is approximately 4.93V. Notably, despite an identical design for all the heaters,  $V_{2\pi}$  may vary slightly among them due to the fabrication non-uniformity.

### IV. RECONFIGURE THE IPUNN CHIP TO PERFORM VARIOUS ML TASKS BY ON-CHIP IN-SITU TRAINING

#### A. Procedure and Experimental Setup of In-Situ Training

In order to enable our IPUNN chip to be reconfigured and train various ML tasks in real-time, we employ a forward propagation-based method rather than the commonly used backpropagation algorithm [40] in a conventional computer, in order to achieve the gradient of each trainable parameter. The gradient  $\mathbf{g}$  of the LF  $\mathcal{L}(u)$  evaluated at  $u_0$ , i.e.,  $g_{u_0}$ , is calculated with a high-order finite difference method based on Lagrange interpolation [41]:

$$\mathbf{g}_u = \nabla_u \mathcal{L}(u), \quad (2)$$

$$g_{u_0} = \frac{-\mathcal{L}(u_0 + 2h) + 8\mathcal{L}(u_0 + h) - 8\mathcal{L}(u_0 - h) + \mathcal{L}(u_0 - 2h)}{12h} + O(h^4), \quad (3)$$

where  $h$  denotes a small perturbation of each trainable parameter, and  $O(h^4)$  represents the truncation error with the order of  $h^4$ . Here, (3) indicates that the gradient of each trainable

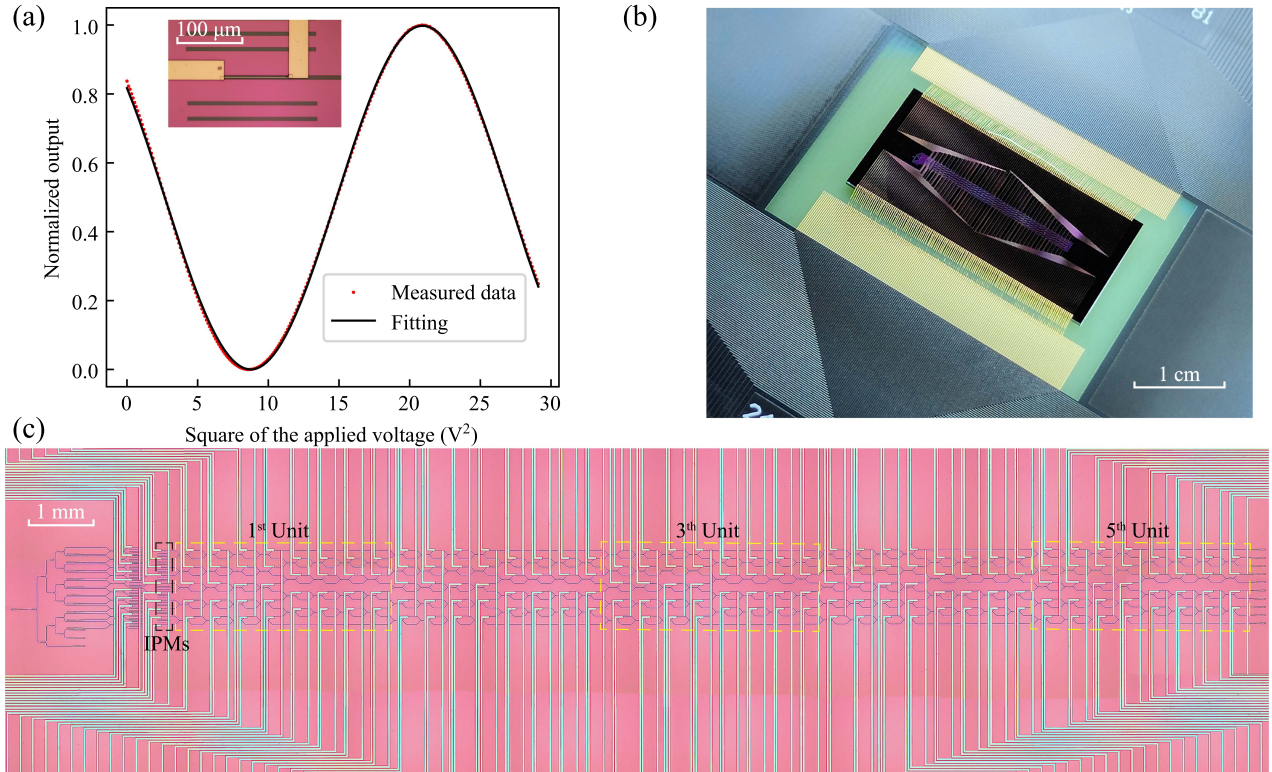


Fig. 5. (a) Normalized transmission response for tuning the 5th IPM. The measurement is done by applying voltage to the IPM, while monitoring the output intensity coupled out from the 9th grating coupler. Ordinal numbers here refer to the case counting from top to bottom. Measured data is then fitted with one-term Fourier series model to evaluate the voltage required for a  $2\pi$  phase shift. The micrograph of the 5th IPM is shown in the inset. (b) Photograph of the IPUNN chip packaged with a printed circuit board. (c) Micrograph of the fabricated IPUNN chip, with the area covering all the photonic devices.

parameter can be evaluated during four forward propagation operations by detecting  $\mathcal{L}(u_0 + 2h)$ ,  $\mathcal{L}(u_0 + h)$ ,  $\mathcal{L}(u_0 - h)$  and  $\mathcal{L}(u_0 - 2h)$ . During the actual experimental process,  $u_0$  corresponds to the current voltage applied on a specific heater while  $h$  is a deviation with respect to  $u_0$ . The perturbation  $h$  must lead to perceivable changes of  $\mathcal{L}(u)$  above the system's background noise, and is set to 0.15 V for all tasks in the experiments, which is determined by several trials. After the gradient  $g_{u_0}$  is obtained, the Adam algorithm is used to update modulation voltages. This in-situ training method offers a viable way to train IPUNNs for various ML tasks in real-time, taking full advantage of the optical acceleration in forward propagation processes. The complete in-situ training process is listed below:

- 1) Initialize all the trainable parameters, i.e., the modulation voltages of all the phase shifters in the unitary transformation architecture.
- 2) In each iteration, encode the input features into phase domain by applying corresponding voltages to IPMs. Implement four consecutive forward propagations with the perturbative parameters  $u_0 + 2h$ ,  $u_0 + h$ ,  $u_0 - h$  and  $u_0 - 2h$  for each trainable phase shifter, measuring the corresponding loss  $\mathcal{L}(u_0 + 2h)$ ,  $\mathcal{L}(u_0 + h)$ ,  $\mathcal{L}(u_0 - h)$  and  $\mathcal{L}(u_0 - 2h)$ .
- 3) Repeat step 2 to traverse samples in the batch of the current iteration. Average the measured loss over the samples in this batch for the four types of perturbations separately,

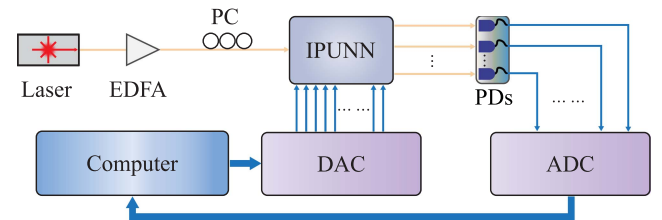


Fig. 6. A schematic of the experimental setup for on-chip in-situ training. The orange and blue lines respectively represent optical and electrical paths. EDFA, erbium-doped fiber amplifier; PC, polarization controller; PDs, photodetectors; DAC, digital-to-analog converter; ADC, analog-to-digital converter.

and then calculate the gradient of the trainable parameter for each phase shifter according to (3).

- 4) Update the modulation voltages of all the trainable phase shifters with the Adam algorithm based on the calculated gradients simultaneously.
- 5) Repeat steps 2 to 4 until the LF converges.

Note that the applied voltages on all trainable modulators range from 0 V to  $V_{2\pi}$  during training regardless of the phase-product factor set for IPMs. The experimental setup for conducting the proposed in-situ training method is shown in Fig. 6. A coherent laser source at 1540 nm wavelength is amplified to 24 dBm by an erbium-doped fiber amplifier (EDFA) to improve signal-to-noise ratio of output signals. Using a polarization controller, the coupling of the light source to the

IPUNN chip is optimized. The output optical signals coupled out from the chip are detected with an array of ten amplified photodetectors (PDs), of which the output voltage signals are digitized using an analog-to-digital converter (ADC) module (National Instruments PXIe-6358) controlled by a computer. The modulation voltage required by each heater is synchronously supplied by multiple DAC modules (National Instruments PXIe-6739) which is also controlled by the same computer. During the on-chip in-situ training process, the IPUNN chip is thermally stabilized on a heat sink attached to a thermoelectric cooler (TEC) in order to alleviate heat accumulation.

### B. Reconfigure the IPUNN Chip to Perform Decision Boundary Tasks

For all samples, we extract the learned ten-dimensional vectors generated from the output of the  $\tanh|z|$  activation in the numerical model shown in Fig. 1(a) and the activated real-valued features are then linearly mapped into phase domain by imparting corresponding phase shifts on the ten IPMs:

$$V^{(i)} = \sqrt{m_i \frac{\phi_{\max}}{2\pi}} V_{2\pi}^{(i)}, \quad (4)$$

where  $m_i$  represents the extracted features in the  $i^{\text{th}}$  dimension,  $\phi_{\max}$  represents the phase-product factor and is set to  $1.5\pi$  for decision boundary tasks.  $V_{2\pi}^{(i)}$  is the modulation voltage required to achieve a  $2\pi$  phase shift for the  $i^{\text{th}}$  IPM and  $V^{(i)}$  is the encoding voltage of the  $i^{\text{th}}$  IPM for the real-valued feature  $m_i$ .

We utilize all the training and testing samples employed in the simulation of decision boundary tasks with a batch size of 30 for on-chip in-situ training. As mentioned in (3), gradient calculation during in-situ training depends on the loss values, which are closely related to the output voltage signals acquired by the ADC. Consequently, output signals necessitate an appropriate magnitude. Here, the output voltage signals proportional to the output optical intensities are multiplied by a common constant to be scaled to a level where the summation of ten-channel signals is around 10V. Based on this scaling level, learning rates of 0.005 and 0.008 are chosen for the Moon and Spiral datasets, respectively. As far as the inference stages are concerned, only the top two (Moon) or three (Spiral) output ports are used for predictions. Nevertheless, output signals from all ten channels, rather than only the top two (Moon) or three (Spiral) output ports, are used for LF calculation during training stages, allowing optical power to gradually concentrate to the top two (Moon) or three (Spiral) output ports along with the evolution of the modulation voltages. The LF is calculated as the cross-entropy between the softmax of scaled ten-channel output signals (acquired by the ADC) and the ten-dimensional one-hot encoding vectors of true labels. In actual experiments, we find that training only the 90 phase shifters (the first two of each robust building block), namely keeping phase shifters of the additional MZIs designed for combating fabrication imperfections passive, is sufficient to effectively reconfigure the IPUNN chip to perform target tasks. This experimental observation is consistent with the theoretical finding in [39].

The on-chip in-situ training convergence processes of decision boundary tasks for the Moon and the Spiral datasets are illustrated in Fig. 7(a) and (b), respectively. The LF of the Moon (Spiral) dataset decreases from 1.08 (1.24) to 0.01 (0.17) with a maximum training accuracy of 100% (99%) and the corresponding testing accuracy is 100% (98.67%). The region of  $x_1, x_2 \in [0, 1]$  is discretized in a step of 0.025 to evenly generate 1681 new samples that are unseen during training to infer decision boundaries. With the input of these unseen samples, the softmax probabilities are calculated for the top two (Moon) or three (Spiral) output ports based on the scaled output voltage signals of corresponding output ports. As an illustrative example for the Moon (Spiral) dataset, the softmax probability distribution of the second (first) output port, calculated based on the scaled output voltage signals in the top two (three) output ports, is shown under three different epochs in Fig. 7(c) [Fig. 7(d)]. The blue region with low probabilities in Fig. 7(c) [Fig. 7(d)] suggests that the output optical power at the second (first) port is suppressed, while the red region with high probabilities implies a concentration of optical power for the input of corresponding samples. Classification results are determined by taking the port with the maximum optical power. The final decision boundaries obtained by in-situ training are shown in the inset of Fig. 7(a) and (b), matching the simulation results well.

### C. Reconfigure the IPUNN Chip to Perform Image Recognition Tasks

We further experimentally validate the performance of UNNs with phase encoding scheme using MNIST and Fashion-MNIST datasets. In our experiments, both of these two datasets use the same configuration in all aspects. Latent embeddings learned by a two-layer unitary fully connected network, as we have already demonstrated in Fig. 3(a), are employed as input features of our IPUNN chip. By applying corresponding modulation voltages to IPMs, these feature embeddings are encoded into the phase domain in accordance to (4), where the phase-product factor  $\phi_{\max}$  is equal to  $\pi$ . As a proof of concept, 500 instances are randomly drawn from the 60,000 training images to implement in-situ training while 300 instances randomly selected from the 10,000 testing images are used to test the trained IPUNN for both two datasets. Similar to decision boundary tasks, the output optical signals are converted to voltage signals by an array of ten PDs and then converted to digital signals via ADCs in real-time. Subsequently, all acquired output voltage signals are scaled to a level with the summation of ten channels reaching around 10V. The cross-entropy between the softmax of the scaled output voltage signals and ten-dimensional one-hot encoding vectors of true labels is calculated as the LF. We train the same 90 phase shifters as in the decision boundary tasks using the Adam algorithm to minimize the LF with a learning rate of 0.01 and a batch size of 25 for image recognition tasks. The recognition result is also determined by the output port with the maximum optical power. Experimental results of the image recognition tasks are shown in Fig. 8, where the convergence processes of the MNIST and Fashion-MNIST are plotted in Fig. 8(a) and (c), illustrating an experimental testing accuracy

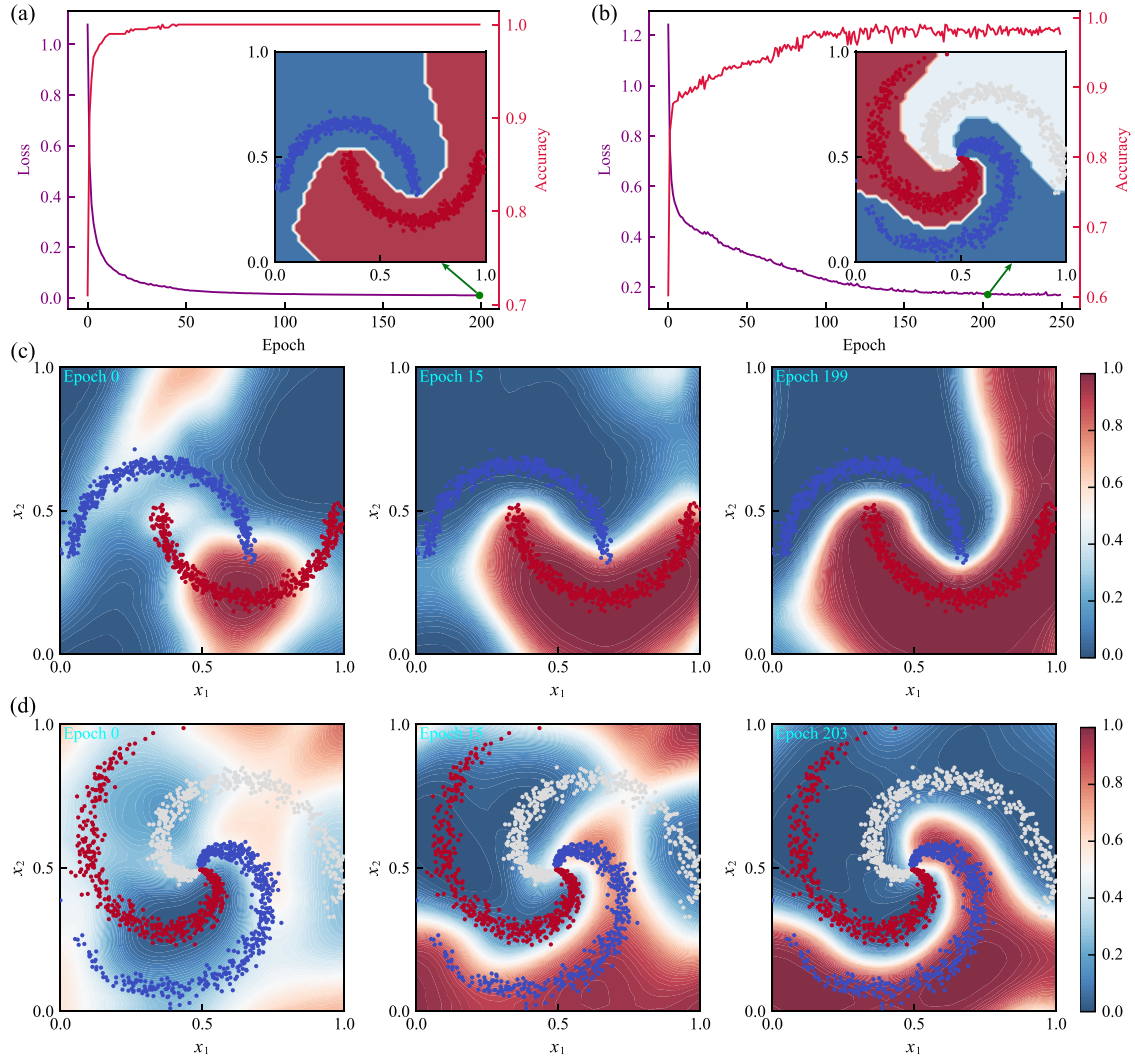


Fig. 7. Experimental results of in-situ training for decision boundary tasks. Convergence processes of the (a) Moon and (b) Spiral with training loss and accuracy are shown. The  $[x_1, x_2]$  region is discretized at an interval of 0.025 to evenly generate 1681 new samples that are unseen during training to infer decision boundaries. With the input of these unseen samples, the softmax probabilities are calculated for the top two (Moon) or three (Spiral) output ports based on the scaled output voltage signals of corresponding output ports. (c) The softmax probability distribution of the second output port under epoch 0, epoch 15 and epoch 199 for the Moon dataset. (d) The softmax probability distribution of the first output port under epoch 0, epoch 15 and epoch 203 for the Spiral dataset. Final predicted decision boundaries shown in the inset of (a) and (b) match the simulation results well.

of 97% and 86.33% after the LF is converged, respectively. The corresponding confusion matrices for the 300 randomly selected testing images are depicted in Fig. 8(b) and (d), respectively.

We analyze the optical intensity distribution coupled out from the ten grating couplers for the testing set after the in-situ training is completed. Thirty test samples are selected for demonstration, with each category comprising three typical instances, whose original images are shown in the Fig. 8(e) and (f) for MNIST and Fashion-MNIST, respectively. The number located at the bottom left corner of each original image obeys to the notation rule of image number-true labels, e.g., the nineteenth image with the label 8 is notated as 19-8. In Fig. 8(g) and (h), the intensity distribution of the ten output ports with the input of abovementioned 30 testing images from MNIST and Fashion-MNIST is displayed, respectively. It clearly demonstrates that the well-trained IPUNN chip efficiently guides the optical signal

to the desired output channel for majority of the testing samples, resulting in the highest energy in the corresponding channel. Moreover, the energy in other non-target channels is suppressed to relatively low levels.

## V. DISCUSSIONS

It is well known that implementing an  $N \times N$  universal unitary matrix optically requires  $N(N-1)/2$  MZI building blocks [5], [6]. To further realize an  $m \times n$  non-unitary matrix  $M$ , it conventionally requires three cascaded parts according to the SVD [17], which is a factorization as  $M = U\Sigma V^\dagger$ , where  $U$  is an  $m \times m$  unitary matrix;  $\Sigma$  represents an  $m \times n$  rectangular diagonal matrix; and  $V^\dagger$  is the Hermitian transpose of the  $n \times n$  unitary matrix  $V$ . Consequently, the expressivity of RVNNs may inherently encompass that of UNNs in general, but at the cost



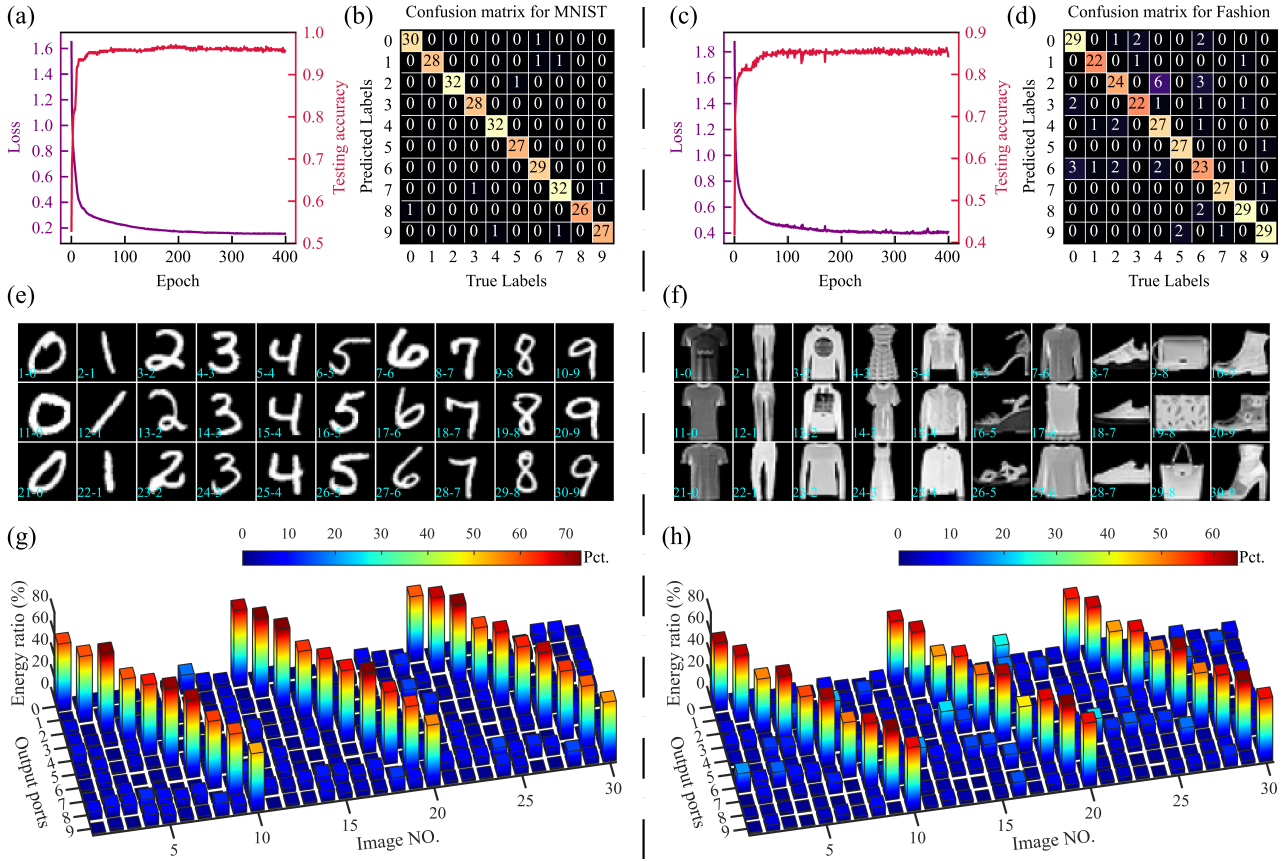


Fig. 8. Experimental results of in-situ training for image recognition tasks, using 500 training images and 300 testing images that are randomly selected from each dataset. Convergence processes of the (a) MNIST and (c) Fashion-MNIST with training loss and testing accuracy are shown. The confusion matrices of the 300 randomly selected testing images from the (b) MNIST and (d) Fashion-MNIST, with a recognition accuracy of 97% and 86.33% being obtained by in-situ training, respectively. Original images of 30 typical testing samples (three per category) in the (e) MNIST and (f) Fashion-MNIST dataset are used for demonstrating output intensity distribution. The output intensity distribution after training is shown in (g) for the testing samples displayed in (e), and in (h) for the testing samples displayed in (f). Pct. denotes percentage. Labels (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) correspond to digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) and fashion products (t-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and boots) for MNIST and Fashion-MNIST, respectively. The number in (e) and (f) located at the bottom left corner of each original image obeys to the notation rule of image number-true labels.

of more complex hardware implementations. For concreteness, when  $m = n = N$ , it takes  $N(N - 1)$  MZI building blocks and  $N$  optical attenuators/amplifiers implementing  $\Sigma$  to build an  $N \times N$  arbitrary real-valued matrix, which could beget more serious performance degradation due to increasing imprecisions [42] compared to constructing a pure  $N \times N$  unitary mesh. In order to reduce the number of required MZIs for RVNNs, Tian et al. proposed a pseudo-real architecture employing the real part of a unitary mesh to represent the target real-valued matrix [43]. However, it demands strict requirements for the inputs to be purely amplitude-modulated, which is challengeable in coherent networks. The  $2 \times 2$  MMI mixers at the output end of the pseudo-real architecture may introduce additional imprecisions caused by fabrication errors. Wu et al. utilized incoherent MZI networks to express  $N \times N$  real-valued matrices with an  $(N + 1) \times (N + 1)$  unitary mesh [44], but it requires  $N$  incoherent light sources modulated by  $N$  intensity modulators and additional  $N$  MZI building blocks compared to  $N \times N$  unitary mesh. Training IPUNNs with phase encoding close to RVNNs conveys that one could achieve non-degraded performance using smaller footprints, lower insertion loss and fewer optical components in specific ML application scenarios.

The performance enhancement of phase encoding may arise from its capacity to induce a certain nonlinearity in UNNs, which could be intuitively visualized from the comparison of various encoding schemes applied in decision boundary tasks, as shown in Fig. 2. From another aspect, similar to the usage of kernel function in supporting vector machine [45], phase encoding could project the original data to a higher-dimensional feature space, making it more easier to be linearly separated. In contrast to the pure phase encoding, hybrid encoding models do not attain a better performance here, mainly because the encoded amplitude and phase information originates from the same real-valued feature, which may result in feature redundancy. This is quite distinct from the complex encoding studied in [14], since the amplitude and phase component of the features calculated by a complex-valued encoder (e.g., learnable complex-valued matrices) are always independent.

In practical applications of IPUNN chips, hybrid encoding and amplitude encoding require intensity modulation for input features, which results in power loss along with reduction in the signal-to-noise ratio. On the other hand, implementing intensity modulation on PICs is commonly realized through modulating Mach-Zehnder modulators or micro-ring modulators.

Nevertheless, these typical intensity modulators often introduce unwanted phase shifts concurrently with intensity modulations. For coherent IPUNNs, it necessitates the extra phase compensation to mitigate this adverse impact on feature encoding. Phase encoding avoids these issues without requiring additional photonic devices and increasing footprints in the waveguide layer. Hence, employing phase encoding in IPUNNs offers significant advantages both from the perspective of the model performance and hardware implementation, increasing potentials for accelerating AI models by IPUNN counterparts.

## VI. CONCLUSION

In summary, we have compared the performance of UNNs with different encoding schemes for real-valued features in several ML tasks. We numerically simulated fully unitary network architectures respectively for decision boundary and image recognition tasks. Simulation results indicate that, the UNNs with phase encoding and the RVNNs with trainable biases exhibit comparable performances for various benchmarks when configurations remain identical, outperforming the other two encoding schemes. Nevertheless, the hardware implementation of UNNs is much simpler than RVNNs in theory. Furthermore, a reconfigurable  $10 \times 10$  IPUNN chip was designed and fabricated to experimentally demonstrate the implementation of UNNs with phase encoding. We effectively reconfigure the IPUNN chip to perform various benchmarks by in-situ training using a high-order finite difference method based on Lagrange interpolation. Good agreement between our simulation and experimental results is observed. It is worth noting that although we demonstrate IPUNNs based on thermo-optical modulation in this work, the computing paradigm presented here can also be extended to other high-speed PIC platforms (e.g., electro-optical modulation on the thin-film lithium niobate platform [46]). Our results implicate that phase encoding-based IPUNN could become a promising computing platform for AI acceleration, bridging the real-valued representations of mainstream AI models and reconfigurable IPUNN architectures in a way that is energy-efficient and conducive to large-scale integration.

## REFERENCES

- [1] N. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," in *Proc. 9th Comput. Limits*, Jun. 14, 2023. [Online]. Available: <https://limits.pubpub.org/pub/wm1lwjce>
- [2] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, no. 7589, 2016, Art. no. 144.
- [3] B. J. Shastri et al., "Photonics for artificial intelligence and neuromorphic computing," *Nature Photon.*, vol. 15, no. 2, pp. 102–114, 2021.
- [4] Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, no. 7, pp. 441–446, 2017.
- [5] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Phys. Rev. Lett.*, vol. 73, pp. 58–61, Jul. 1994.
- [6] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, "Optimal design for universal multiport interferometers," *Optica*, vol. 3, no. 12, pp. 1460–1465, Dec. 2016.
- [7] F. Shokraneh, S. Geoffroy-gagnon, and O. Liboiron-Ladouceur, "The diamond mesh, a phase-error- and loss-tolerant field-programmable MZI-based optical processor for optical neural networks," *Opt. Exp.*, vol. 28, no. 16, pp. 23495–23508, Aug. 2020.
- [8] J. Carolan et al., "Universal linear optics," *Science*, vol. 349, no. 6249, pp. 711–716, 2015.
- [9] J. Wang et al., "Multidimensional quantum entanglement with large-scale integrated optics," *Science*, vol. 360, no. 6386, pp. 285–291, 2018.
- [10] J. M. Arrazola et al., "Quantum circuits with many photons on a programmable nanophotonic chip," *Nature*, vol. 591, no. 7848, pp. 54–60, 2021.
- [11] Y. Chi et al., "A programmable qudit-based quantum processor," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 1166.
- [12] F. Shokraneh, S. Geoffroy-Gagnon, M. S. Nezami, and O. Liboiron-Ladouceur, "A single layer neural network implemented by a  $4 \times 4$  MZI-based optical processor," *IEEE Photon. J.*, vol. 11, no. 6, Dec. 2019, Art. no. 4501612.
- [13] H. Deng and M. Khajavikhan, "Parity-time symmetric optical neural networks," *Optica*, vol. 8, no. 10, pp. 1328–1333, 2021.
- [14] H. Zhang et al., "An optical neural chip for implementing complex-valued neural network," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 457.
- [15] S. Pai et al., "Experimentally realized in situ backpropagation for deep learning in photonic neural networks," *Science*, vol. 380, no. 6643, pp. 398–404, 2023.
- [16] Y. Shi et al., "Nonlinear germanium-silicon photodiode for activation and monitoring in photonic neuromorphic networks," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 6048.
- [17] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1995.
- [18] N. C. Harris et al., "Linear programmable nanophotonic processors," *Optica*, vol. 5, no. 12, pp. 1623–1631, Dec. 2018.
- [19] S. Pai et al., "Parallel programming of an arbitrary feedforward photonic network," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 5, Sep./Oct. 2020, Art. no. 6100813.
- [20] R. Wang et al., "Photonic unitary neural network chip with complex-valued transformation layer," in *2021 Asia Commun. Photon. Conf.*, 2021, pp. 1–3.
- [21] G. Sarantoglou, A. Bogris, C. Mesaritakis, and S. Theodoridis, "Bayesian photonic accelerators for energy efficient and noise robust neural processing," *IEEE J. Sel. Topics Quantum Electron.*, vol. 28, no. 6, Nov./Dec. 2022, Art. no. 6100710.
- [22] L. Jing et al., "Tunable efficient unitary neural networks (EUNN) and their application to RNNs," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 1733–1741.
- [23] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1120–1128.
- [24] Z. Yang, T. Zhang, J. Dai, and K. Xu, "Tunable-bias based optical neural network for reinforcement learning in path planning," *Opt. Exp.*, vol. 32, no. 10, pp. 18099–18112, May 2024.
- [25] X.-K. Li et al., "High-efficiency reinforcement learning with hybrid architecture photonic integrated circuit," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 1044.
- [26] I. A. D. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, "Reprogrammable electro-optic nonlinear activation functions for optical neural networks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 1, Jan./Feb. 2020, Art. no. 7700412.
- [27] K. Takiguchi, T. Kitoh, A. Mori, M. Oguma, and H. Takahashi, "Optical orthogonal frequency division multiplexing demultiplexer using slab star coupler-based optical discrete Fourier transform circuit," *Opt. Lett.*, vol. 36, no. 7, pp. 1140–1142, 2011.
- [28] M. Y. Saygin, I. V. Kondratyev, I. V. Dyakonov, S. A. Mironov, S. S. Straupe, and S. P. Kulik, "Robust architecture for programmable universal unitaries," *Phys. Rev. Lett.*, vol. 124, Jan. 2020, Art. no. 010501.
- [29] G. Cong et al., "On-chip bacterial foraging training in silicon photonic circuits for projection-enabled nonlinear classification," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 3261.
- [30] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 8792–8802.
- [31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*, doi: [10.1038/s41566-021-00796-w](https://doi.org/10.1038/s41566-021-00796-w).
- [32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [33] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
- [34] G. Bradski, "The OpenCV library," *Dr Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–123, 2000.
- [35] C. Ramey, "Silicon photonics for artificial intelligence acceleration: Hotchips 32," in *2020 IEEE Hot Chips 32 Symp.*, 2020, pp. 1–26.
- [36] H. H. Zhu et al., "Space-efficient optical computing with an integrated chip diffractive neural network," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 1044.

- [37] T. Fu et al., "Photonic machine learning with on-chip diffractive optics," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 70.
- [38] D. A. B. Miller, "Perfect optics with imperfect components," *Optica*, vol. 2, no. 8, pp. 747–750, Aug. 2015.
- [39] R. Hamerly, S. Bandyopadhyay, and D. Englund, "Asymptotically fault-tolerant programmable photonics," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 6831.
- [40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [41] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, 2004.
- [42] M. Y.-S. Fang, S. Manipatruni, C. Wierzynski, A. Khosrowshahi, and M. R. DeWeese, "Design of optical neural networks with component imprecisions," *Opt. Exp.*, vol. 27, no. 10, pp. 14009–14029, May 2019.
- [43] Y. Tian et al., "Scalable and compact photonic neural chip with low learning-capability-loss," *Nanophotonics*, vol. 11, no. 2, pp. 329–344, 2022.
- [44] B. Wu et al., "Real-valued optical matrix computing with simplified mzi mesh," *Intell. Comput.*, vol. 2, 2023, Art. no. 0047.
- [45] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.
- [46] D. Zhu et al., "Integrated photonics on thin-film lithium niobate," *Adv. Opt. Photon.*, vol. 13, no. 2, pp. 242–352, Jun. 2021.