Supplementary Materials for

**On-chip Reconfigurable Optical Neural Networks**

Xianmeng Zhao1,2,3,†, Haibin Lv1,†, Cheng Chen1, Shenjie Tang1, Xiaoping Liu1\*, Qin Qi4,5\*

1School of Physical Science and Technology, ShanghaiTech University, Shanghai 201210, China

2Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai 201800, China

3University of Chinese Academy of Sciences, Beijing 100049, China

4College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen, 100049, China

5Shenzhen Key Laboratory of Intelligent Optical Measurement and Detection, Shenzhen, 100049, China

†These authors contributed equally to this work

\*E-mail: [liuxp1@shanghaitech.edu.cn](mailto:liuxp1@shanghaitech.edu.cn); [qi.qin@szu.edu.cn](mailto:qi.qin@szu.edu.cn)

**1 Chip design, characterization, and error analysis**

**1.1 Chip design and characterization**



**Figure S1. Chip design and characterization**. a) Micrograph of a basic on-chip optical unit used in our diffractive optical neural network. b) Micrograph of a Mach–Zehnder interferometer used for testing our thermo-optical phase modulator. c) Experimentally obtained phase modulation curve as a function of the applied voltage.

Our diffractive optical neural network (DONN) is designed and fabricated (by *SiPhotonIC1*) on a commercially-available silicon-on-insulator (SOI) substrate with a 220 nm thick top silicon device layer and buried oxide with 2 µm thickness. This basic unit includes an optical interference unit (OIU) and an optical phase modulator unit (OPMU), as shown in Fig. S1a. The OIU is constructed with a 5 × 5 multi-mode interference (MMI) device with a footprint size of roughly 325 µm × 15 µm. The OPMU consists of five SOI strip waveguides with an identical size of 220 nm × 500 nm. Each of them is individually thermo-optically phase-modulated via a microheater, i.e., a TiN resistive wire that is 100 nm thick, 30 µm wide, and 100 µm in length. It is electrically connected to wire-bonding pads via patterned aluminum metal wires. All the wires are optically isolated from the SOI device layer via a deposited oxide buffer layer. The fabricated chip is first glued on a printed circuit board (PCB) using a thermally conductive paste and then wire-bonded to the PCB. A testing modulator is placed on one of the Mach-Zehnder interferometer (MZI) arms to measure the phase response curve of the fabricated thermo-optic modulators, as shown in Fig. S1b. The MZI is patterned next to the DONN on the same chip. The microheater’s measured electrical resistance is about 210 , and a representative phase curve as a function of the square of the applied voltage is shown in Fig. S1c. Here, the response curve is roughly linear with a value of about 3.8V.

The dimensions of our MMI-based OIU are optimized numerically in COMSOL Multiphysics to maximize transmission for each input channel. The final OIU device has a footprint size of 325 µm × 15 µm, and its transmission matrix *W* is shown below

.

This transmission matrix verifies that each input channel of this OIU can transmit about 98% input power almost evenly into the 5 output channels. The corresponding intensity distribution of this OIU for two representative input channels is depicted in Fig. S2.



**Figure S2. The intensity distribution of our OIU for two different input channels**. a) Middle input channel. b) Top-most input channel.

**1.2 Phase error analysis and the transfer matrix’s completeness of the hidden layer**

****

**Figure S3. Phase error.** The SOI waveguide mode’s propagation phase error versus the waveguide length for different waveguide width deviation width.

As discussed above, the interconnection between adjacent OIUs consists of an OPMU, which are SOI strip waveguides with optical phase modulators on them. The optical mode effective index of these waveguides and the corresponding optical propagation phase are affected by the waveguide geometric parameters. Fig. S3 plots the calculated phase error curve versus the propagation distance for different waveguide-width deviation, , from the targeted waveguide width of 500 nm. It indicates that even nanometer-scale deviation reasonably within fabrication tolerance can cause a large phase error in coherent DONNs, especially with a long propagation distance. This finding means that the fabricated DONN that relies on coherent interference will most likely fail to function without in-situ training.

We can already conclude that the existence of the inevitable phase errors in OPMU, as discussed above, directly affects the practical ability to explicitly configure the DONN to a targeted state. But the phase errors could also come from the imperfect OIU. Fortunately, an appropriate designed DONN architecture can compensate for both of these errors and allows for almost any arbitrary transfer matrix out of the continuous unitary space2. This effect is demonstrated on our layered DONN using the Monte Carlo simulation. The transfer matrix for its hidden layers can be described by following formulate:

Here is a phase vector of the OPMU, the is a Haar-random perturbation, the is a random number for different layers. denotes the perturbation to the designed transfer matrix the OIU, i.e., *W* from Section 1. Given a random unitary matrix , the Monte Carlo simulation algorithm searches for a global minimum of infidelity over the space of phase vectors. The is defined as:

Here *N* is the dimension of the . For example, as shown in Fig. S4, the probability density function (PDF) of the infidelity of the 500 randomly generated unitary matrices that the hidden layer of our DONN needs to express is sharply center at the zero infidelity. This example strongly suggests that our DONN’s hidden layers can express an almost arbitrary transfer matrix out of the continuous unitary space.



**Figure S4. Infidelity** **probability density function (PDF) of the 500 randomly generated unitary matrices**.

**2 On-chip DONN training**

**2.1 Experiment setup**

As shown in Fig. 1c and d of the main text, the input light (Santec TSL-710), having a wavelength of 1550 nm and optical power of 10 dBm, is coupled into the chip through a grating coupler and then split into eight parallel channels by cascaded MMIs. Five channels are used as the input ports for DONN. The DONN’s output channels are coupled into a fiber array via an array of on-chip pitch-matched grating couplers. The coupled power is detected with amplified photodetectors (Thorlabs PDA20CS2), of which the voltage signals are digitized by an analog-to-digital converter (ADC) interface (National Instruments PXIe-6358) controlled by a corresponding personal-computer (PC) based PXI controller. The drive voltage required by the microheater is synchronously supplied by a multi-channel digital-to-analog converter (DAC) (National Instruments PXIe-4322) that is also controlled by the same controller. During experiments, the DONN chip is temperature stabilized on a heat sink to alleviate heat accumulation during a long-time training session.

**2.2 Gradient Calculation**

The gradient of the cost function (*CF*) for the trainable variables is evaluated with Lagrange interpolation method3, which is given by

where isthe gradient evaluated at *x*, and *h* is a small perturbation imposed on each variable *x* for *CF*(***x***). In our case, the variables are the applied voltages of the phase modulators in all the hidden layers. We choose , a value determined after several rounds of trial training, to guarantee that the perturbed *CF* has observable changes above our setup’s noise background.

**2.3 Parameter updating**

In order to overcome the noise of gradients and damp oscillations across ravines, the adaptive moment estimation (Adam) gradient descent method4 is used to update the variables,

Here, *t* is the training epoch, ***m*** is the first moment that resembles the momentum that records the past normalized gradient, ***n*** is the second moment that gives different learning rates for different variables, are constant numbers, are bias-corrected items for the ***m*** and ***n***, *r* is the learning rate equal to 0.03 in our experiment and is a small constant number to avoid the denominator being zero.

**2.4 Training Implementation**

The on-chip training of DONN can be implemented as the following algorithm in pseudocode format.

**Algorithm** On-chip training of DONN

X = : the applied voltages of the phase shifters for the input channels, where is the measured mapping function that maps an applied voltage of our thermo-optical phase shifter to a corresponding phase value, see Fig. S1(c) for example;

Y: true labels;

E: the maximum number of training epoch;

R: learning rate;

*h*: small perturbation;

W: the applied voltages to all the phase shifters in the hidden layers;

**function** TRAIN (X, Y, E, R, *h*):

Initialize W;

**for** (*i* = 1; *i* <= E; *i* = *i* + 1) **do**

predict = forward propagation (X, W);

loss = CF (predict, Y);

**for** (*k* = 1; *k* <= length (W); *k* = *k* + 1) **do**

calculate loss with ;

calculate the *k*-th component of the gradient with Lagrange interpolation method;

**end for**

update W with Adam method;

**end for**

**end function**

**3 Training the DONN as a one-bit full adder**

A one-bit full adder can be considered as a classifier with a vector input , where is the carry from the previous full adder, and are the two addends in the *n*-th bit. We split a one-bit full adder into two steps (step A and step B). The input of the first step, step A, of a full adder is the vector of . The second step input, i.e., step B, is the vector of , where and are the outputs of the first step, respectively. The output of a one-bit full adder is a two-dimensional vector , where is the carry and is the sum.

**3.1 Forward propagation**

Since one-bit full adder is a four-port device (two input and two output), only the two input/output channels are required among the total six input/output channels. For forward propagation, the input data are encoded into optical phases in the first two input channels. Each hidden layer performs the same kind of operation to its input, i.e., the output from its prior layer,

.

Here, is the -th layer output, is the phase shift applied to the OPMU, and the symbol means Hadamard product operation. The output of the last layer is fed to the SoftMax function

where the SoftMax function is defined as:

Then it is normalized with the maximum value in the vector as follows:

where is the *j*-th output vector of the SoftMax function, and  is the corresponding normalized value. Finally, only the first output channels corresponding to the first two elements in are used as the prediction output to further calculate the corresponding cost function.

**3.2 Training results**

In our experiment, the cost function of both step A and step B is defined as the mean squared error, which can be expressed as follows:

Here, is the *j*-th predicted result, is the *j*-th true result, is the weight for *j*-th category, and is the number of input channels. For step A, equal weight for all the output is used, e.g., . For step B, only the weight associated with the targeted category of (1,1) is larger than 1 to achieve equality between the two output ports. The DONN on-chip training algorithm above is then used to train our DONN with this cost function to implement step A and step B separately. The *CF* as a function of the training epoch for step A and step B is shown in Fig. S5. With the final minimized *CF*, our DONN gives results (illustrated in Fig. 2b in the main text) that agree well with the corresponding truth tables in Table S1.



**Figure S5. Training one-bit full adder**.Cost function as a function of the training epoch for step A and step B.

**Table S1. The truth tables for step A, step B, and full adder.**



**3.3 Test result of the as-fabricated one-bit full adder DONN**



**Figure S6. Test results of the as-fabricated one-bit full adder**.

We apply the computer trained parameters that are the modulators’ phase delay values to our as-fabricated DONN. Fig. S6 shows the corresponding results. Neither step A nor Step B is consistent with the truth table (shown in Table S1), which indicates that the as-fabricated DONN without our proposed in-situ training fails to function correctly. This finding is expected from the phase error analysis discussed in Section 1. Therefore, in-situ training is required to tune the DONN to a correct state.

**4 Training the DONN as a vowel classifier**

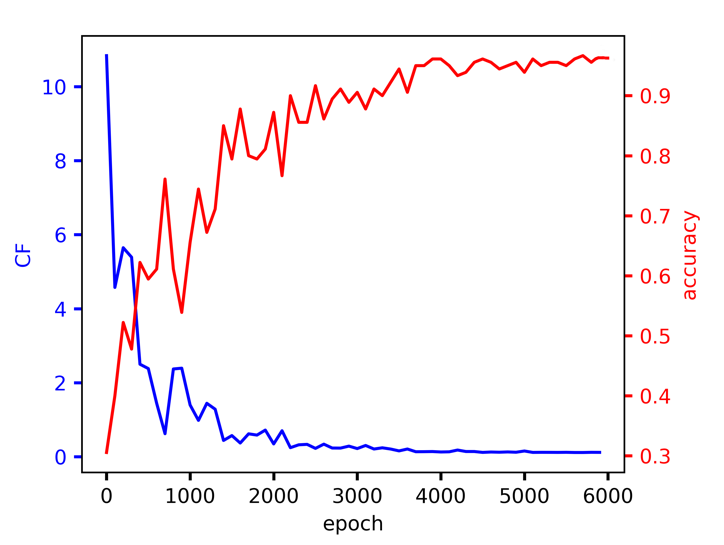
**4.1 Forward Propagation**

The procedure of setting up forward propagation of the vowel classifier in the DONN is the same as the full adder except for the number of input and output ports used. The SoftMax function is also applied to the output of the last layer to get the final classification result

**4.2 Training and Testing on a 64-bit computer**

For training the vowel classifier, the cost function is defined as the cross-entropy function5, which is given by

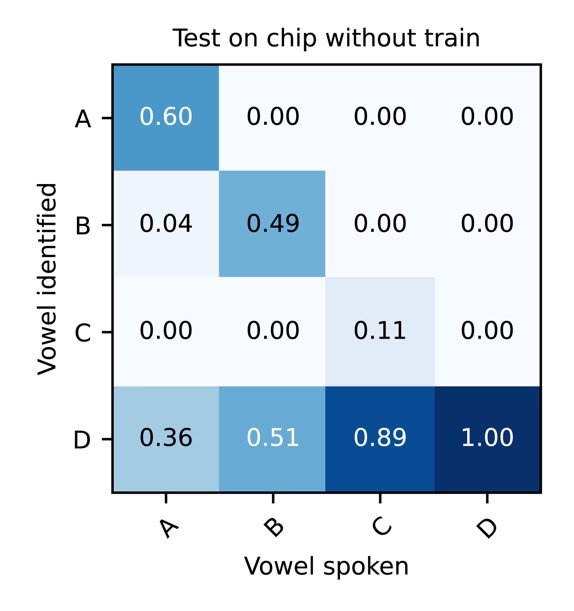
Here, is the *j*-th predicted result, and is the *j*-th true result. Our on-chip DONN’s optical transfer function with trainable variables is numerically implemented in *TensorFlow6* framework and trained*.* The *CF* and accuracy for the training set are shown in Fig. S7.



**Figure S7.** Cost function and accuracy evolution for training on the personal computer.

**4.3 Test result of the as-fabricated vowel classifier DONN**

Same as the one-bit full adder discussed above,we apply the computer trained parameters to the as-fabricated DONN before the proposed in-situ training, and the test result is shown in Fig. S8. Again, it fails to work properly with the vowel classifier’s accuracy of 55%.

****

**Figure S8. Test results of the as-fabricated vowel classifier.**

**5 Training speed and power efficiency**

**5.1 Training speed.**

On the conventional electronic computing hardware, training neural networks typically demands significant computational time and resources. Our proposed architecture for training the DONN on-chip takes advantage of parallel forward propagation of DONN and potentially high-speed modulators, which may enable high-speed training compared to the conventional computer. A rough estimation of the DONN’s performance could be benchmarked with the number of forward-propagations per second. It can be evaluated with

Here, is the bandwidth of the photodetectors, is the bandwidth of phase shifter, is the optical calculation bandwidth determined by the network’s optical path length, which depends on the number of the layers, , and the length of each layer, , are the bandwidth of DAC and ADC. With *N* optical modulators per layer, *m* hidden layers, and *M* train data points, the time for one training one epoch is about . In our experiment, the photodetectors, optical modulators, DACs, and ADCs have roughly 250 kHz bandwidth. The is about 25 GHz in our case, considering the length per layer of 600 μm and seven layers. Hence, our DONN’s performance in our experiment is limited by the bandwidth of DACs and ADCs. In this case, the corresponding training time is estimated to be less than one second per one epoch. This number can be potentially reduced using the fast photodetectors and phase shifters coupled with fast electronic interfaces (e.g., DACs and ADCs operating at 5 GHz and 7-bit precision)7,8. Therefore, an optimistic estimation for the training time could be on the order of 10 for one epoch. On the conventional digital computing hardware, the training time for one epoch is about , where is the floating-point operations per second (FLOPs). For a typical Nvidia Tesla-p100 GPUs, single-precision *R* is about 10 TFLOPs. DONN will start to show comparative computing speed advantages when .

**5.2 Power efficiency**

The energy cost mainly depends on the DACs, ADCs, phase shifters, photodetectors, and light sources. For the DONN, one forward-propagation equals to complete floating-point operations. The power efficiency of DONN, defined as energy per FLOP, is

Here, is the power per DAC, phase shifter, ADC, and photodetector, respectively, and is the power of the light source. With 10 mW per laser, 68 mW per phase shifter, 26 mW per DAC, 72 mW per ADC, 17 mW per photodetector, our DONN’s power efficiency is on the order of 0.01 per FLOP. If the electronic bandwidth for all the components, i.e., DAC, phase shifter, ADC, and photodetector, could be increased9, e.g., to 5 GHz, a rough estimation of the power efficiency would be less than 10 pJ per FLOP. In this scenario, even for a small-scale DONN, the power efficiency is much better than the conventional computing hardware, for instance, 100 pJ per FLOP for GPUs. It suggests that training the DONN on-chip could be a far more power-efficient solution than conventional computing hardware.

Reference:

1 SiPhotonicIC. *For your silicon photonic integrated circuits*, https://siphotonics.com (2020).

2 Saygin, M. Y. *et al.* Robust Architecture for Programmable Universal Unitaries. *Phys Rev Lett* **124**, 010501 (2020).

3 Berrut, J.-P. & Trefethen, L. N. Barycentric lagrange interpolation. *SIAM review* **46**, 501-517 (2004).

4 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

5 Kline, D. M. & Berardi, V. L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications* **14**, 310-318 (2005).

6 Abadi, M. *et al.* Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

7 Sedighi, B., Khafaji, M. & Scheytt, J. C. Low-power 8-bit 5-GS/s digital-to-analog converter for multi-gigabit wireless transceivers. *International Journal of Microwave and Wireless Technologies* **4**, 275-282 (2012).

8 Fang, J. *et al.* A 5-GS/s 10-b 76-mW time-interleaved SAR ADC in 28 nm CMOS. *IEEE Transactions on Circuits and Systems I: Regular Papers* **64**, 1673-1683 (2017).

9 Bangari, V. *et al.* Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs). *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1-13 (2019).