| | |
|---|---|
| Title | Efficient on-chip training of optical neural networks using genetic algorithm |
| Author(s) | Hui Zhang, Jayne Thompson, Mile Gu, Xu Dong Jiang, Hong Cai, Patricia Yang Liu, Yuzhi Shi, Yi Zhang, Muhammad Faeyz Karim, Guo Qiang Lo, Xianshu Luo, Bin Dong, Leong Chuan Kwek, and Ai Qun Liu |

# Efficient On-Chip Training of Optical Neural Networks Using Genetic Algorithm

Hui Zhang[1], Jayne Thompson[2*], Mile Gu[3,4,2*], Xu Dong Jiang[1*], Hong Cai[5], Patricia Yang Liu[1], Yuzhi Shi[1], Yi Zhang[1,6], Muhammad Faeyz Karim[1], Guo Qiang Lo[7], Xianshu Luo[7], Bin Dong[7], Leong Chuan Kwek[1,2,8*], and Ai Qun Liu[1*]

*[1] Quantum Science and Engineering Centre (QSec), Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

*[2] Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, 117583, Singapore*

*[3] Quantum Hub, School of Physical and Mathematical Science, Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

*[4] Complexity Institute and School of Physical and Mathematical Sciences, Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

*[5] Institute of Microelectronics, A*STAR (Agency for Science, Technology and Research), 138634, Singapore*

*[6] School of Mechanical & Aerospace Engineering, Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

*[7] Advanced Micro Foundry, 11 Science Park Road, 117685 Singapore*

*[8] National Institute of Education, 1 Nanyang Walk, 637616 Singapore*

**Corresponding Authors: thompson.jayne2@gmail.com (Jayne Thompson), mgu@quantumcomplexity.org (Mile Gu), exdjiang@ntu.edu.sg (Xu Dong Jiang), cqtklc@nus.edu.sg (Leong Chuan Kwek), eaqliu@ntu.edu.sg (Ai Qun Liu)*

## ABSTRACT

Recent advances in silicon photonic chips have made huge progress in optical computing owing to their flexibility in the reconfiguration of various tasks. Its deployment of neural networks serves as an alternative for mitigating the rapidly increased demand for computing resources in electronic platforms. However, it remains a formidable challenge to train the online programmable optical neural networks efficiently, being restricted by the difficulty in obtaining gradient information on a physical device when executing gradient descent algorithm. Here, we experimentally demonstrate an efficient, physics-agnostic, and closed-loop protocol for training optical neural networks on chip. A gradient-free algorithm, i.e., the genetic algorithm, is adopted. The protocol is on-chip implementable, physical agnostic (No need to rely on characterization and offline modelling), and gradient free. The protocol works for various types of chip structures and is especially helpful to those cannot be analytically decomposed and characterized. We confirm its viability using several practical tasks, including the crossbar switch and the *Iris* classification. Finally, by comparing our physics-agonistic and gradient-free method to the off-chip and gradient-based training methods, we demonstrate the robustness of our system to perturbations such as imperfect phase implementation and photodetection noise. Optical processors with gradient-free genetic algorithms have broad application potentials in pattern recognition, reinforcement learning, quantum computing, and realistic applications (such as facial recognition, natural language processing, and autonomous vehicles).

## KEYWORDS

Optical neural networks, on-chip training, gradient-free, genetic algorithm, deep learning optical computing.

# INTRODUCTION

The integration of precise chip-based optical processors have advantages in precision and compactness comparing to capability of spatial optics[1-2]. They are typically formed by interconnected Mach-Zehnder interferometers (MZIs)[3-6] and are programmed by tuning the relative phases of optical modes in the arms of the MZIs [7-9]. Those optical processors are widely applied to frequency filtering[10], switch networks[11], optical neural networks[12] and quantum optics operations[13-16]. Among them, the optical neural network has attracted a lot of interest, because the frequently repeated matrix-multiplication-accumulation in neural network algorithms[12, 17-18] are passively executed at low energy consumption and high speed. Other effective architectures for implementing the feed-forward part of the neural networks based on passive and active microring resonator (MRR)[19-22] have also been reported.

However, there remains significant challenges in engineering efficient real-time and automatic reconfiguration, for the on-chip training of an optical neural network. Previous implementations of optical neural networks are trained by simulating a model of the system offline on standard computing equipment[12]. This requires explicit knowledge the characterisation of each component inside the chip, whose efficacy fails without an extensive and highly accurate physical model of how they integrate and interact. Meanwhile, the gradient descent training via finite difference method leads to high computational complexity because one has to sample many times to get accurate estimation of the gradient of each of the variables[23]. A possible work around for evaluating gradients was proposed via in-situ gradient measurement with the adjoint variable method (AVM)[24].

These limitations motive consideration of a neuro-evolutionary learning methods[25-26]. Here, we demonstrate experimentally the on-chip training of optical neural networks via genetic algorithm (GA). We collect and analyse the output signals of the processor, adjust them dynamically and optimize the inner-chip parameters, under the routine of GA. The resulting closed-loop system has the following desired features: (a) It is a physics-agnostic method which does not require us to know the detailed chip structures and thus work even for chips with no analytical model. (b) It does not require collection of exponentially large datasets – a issue for existing methods for dealing with chip structures that cannot be analytically characterized[27]. (c) In our protocol, the training accuracy does not rely on an offline modelling function. The environmental perturbations (e.g., thermal crosstalk, photodetection noise) that are hard to control and model, can be expressed in the chip response and effectively cancelled during our gradient-free training process. (d) Our protocol does not require gradient information, which are very challenging to obtain directly from a physical device. We demonstrate a proof-of-principle task of training an optical neural network to classify *Iris* dataset, illustrating the viability of the GA-based training protocol.

## FRAMEWORK OVERVIEW AND EXPERIMENTAL SECTION

**Detailed working flow -** Optical chips have two types of inputs: optical and electrical. Optical input refers to the designated light intensity based on special tasks. Electrical input is the current supplied to the thermal-optic phase modulator, which programs the transformation matrix of the chip. The training goal of the chip is to achieve a desired output intensity distribution $\widetilde{D}$ by optimizing the set of currents that are supplied to the chip, under the given input intensity distribution.

In a genetic algorithm, one begins with a set of individuals called a population. Each individual is a potential solution to the problem that we wish to solve. In our training protocol, the individuals are selected as the supplied currents to the chip, and the characterization of the dependence of chip components to those supplied currents are taken naturally as one part of the training. By this way, we avoid being restricted by the accuracy of modelling the chip, and can make online corrections to the electrical crosstalk, heat crosstalk and unavoidable noise during the training process. Our system is a fully automated with feedback from a closed loop (**Fig. 1a**). All the training steps and the interaction between the electrical interface and the optical chip are executed sequentially without human participation.

The training process is summarized as follows: first, initialize a population $E_{init}: \Phi = E_1, \dots, E_t$, which is composed of *t* randomly generated individuals. Each individual in this population is a collection of the supplied electrical currents and is represented by $E_l: \Phi = I_{l1}^{(\theta)}, \dots, I_{lN}^{(\theta)}, I_{l1}^{(\phi)}, \dots, I_{lN}^{(\phi)}, l \in [1, t]$, where $N = m(m-1)/2$ with *m* denoting the number of modes on the optical chip. Each current then uniquely determines its corresponding phase angle $\theta$ (or $\phi$), where $\theta$ is the internal phase of an MZI and $\phi$ is the external phase. Each individual will produce an output intensity distribution *D* that is acquired by intensity measurement to the chip's output ports. Our training goal is to find an optimum individual that has the closed output intensity distribution to the target one. To evaluate the performance of individuals, we define a fitness function $f(E)$ as the *l*$_2$-norm between the target $\widetilde{D}$ and the measured *D,* as expressed by

$$f(E) = \left\|\widetilde{D} - D\right\| \tag{1}$$

where $f(E) \in [0, \infty)$, meaning the lower the fitness value, the better the performance of evaluated individual. The genetic algorithm will generate new population from the current population by some operators (see **Implementation of Genetic Algorithm**). The new population will replace the current one and the individuals included will be evaluated on chip. By continuously executing new individuals, measuring, and processing, an effective closed loop is formed between the optical chip and the electrical system. The training stops when one of the stopping criteria is met, such as when the limitation of the number of iterations are reached or the fitness value is decreased to the threshold, for instance, e.g., $10^{-3}$.

**Hardware components -** The experimental setup is shown in **Fig. 1b**. The system is comprised of an optical part and an electrical interface. The electrical interface collects, and analyses signals from the optical part and gives control instructions of the supplied voltages to the phase shifters on chip. The reconfigured chip then reacts and produces updating signals to the electrical interface until the training goal is achieved. The optical part is composed of a 1550 nm laser with 12 dBm power, a polarization controller, a multi-channel optical splitter with switch placed at each channel, the integrated optical neural network chip and an array of photodetectors. Two fibre arrays are coupled to the chip at both input and output ports, which enable selection among multiple input ports and output puts. The output signals from the chip are converted to electrical signals by array of photodetectors and collected by a Data Acquisition (DAQ) module, which consists of a gainable transimpedance amplifier (TIA) and a 16-bit analog-to-digital converter (ADC). The classical logic unit analyses the signals and produces instructions to guide the next-step chip configuration, based on the genetic algorithm. A 16-bit digital-to-analog (DAC) circuit works as the performing circuit and supply electrical power to configure the chip.

The hardware control flow is presented as: (a) Inject laser and select the input port by the array of optical switches. (b) Randomly generate a set of initial electrical power $(I_1, \dots I_n)$, where $n = m(m-1)$, $m$ is the total modes of the linear optical circuits. (c) Supply the electrical powers onto phase shifter and collect the output intensity distributions $D = (d_1, \dots d_m)$ using the photodetector array. (d) Analyse the collected distributions and produce new instructions which are updated set of electrical powers, following genetic algorithm. (e) Repeat step (c) and (d) until the stopping criteria satisfied. The fabrication result using Reck scheme design[8] (two layers, each with 6 neurons) are shown in **Fig. 2a**. Two layers are connected by an array of reconfigurable MZIs. Another chip that using Fast design[9, 17] (single layer with 8 neurons) is shown in **Fig. 2b**. The loss of the cross marked by yellow circle is 0.02dB each. The fly in the ointment is that the waveguide routing in this design contains curved paths and shall be optimized to avoid bending losses. The chips adopt thermo-optical phase shifters which are integrated as metallic thin-film tracks and coated on the top of the silicon waveguides. When plugged to electrical power, the phase shifters act as non-ohmic resistive heaters, thus changing the local refractive index of the waveguides[28].

**Implementation of Genetic Algorithm** - The GA is a heuristic algorithm inspired by the theory of evolution: individuals in an ecosystem with high adaptability to environmental variables have higher probability of surviving and reproducing. The average adaptability of population is expected to gradually improve with each generation. The working flow of GA is shown in **Fig. 1c**. The first step is population initialization. Individuals in the population are evaluated on chip and fitness values are returned. We generate new population to replace the current one by three genetic operators, i.e., the selection, the crossover and the mutation. Qualified individuals (those with lower fitness values in our case) are selected from the initial population as parents. Some of the individuals in the parents that have the lowest fitness are chosen as “elite”, which are passed directly to the next population. Children are produced from parents either by combining the vector entries of a pair of parents - crossover or by making random changes to a single parent – mutation.

The current population will be replaced with children to form the next generation. The new generation is implemented onto chip, and chip responses are collected again. We then stop the training process and return the optimum individual when stopping criteria is satisfied. The selection operator adopted in this work is a stochastic uniform selection approach[29]. The crossover ratio is set to be 0.8. The crossover operator creates children by selecting vector entries from a pair of parents and combines them to form a child. Intermediate crossover is adopted which is to create children by taking a weighted average of the parents. As the variables we optimized are currents, a constraint $I \leq 5$ mA is applied. Random changes are brought into the individuals by the operation mutation.

## RESULTS AND DISCUSSION

We implement the proposed training protocol on chip to realize several practical tasks, and significantly we compare our on-chip GA-based training to other methods in terms of computational cost and performance, i.e.,

(a) To program an 1D array of MZIs, which is a typical structure that cannot be characterised because the input and output are the only accessible ports of the whole array[27]. Our training protocol is physics-agnostic and requires no calibration before programming it.

(b) To program a 6-mode optical chip for realizing arbitrary crossbar switch. It is an essential task to demonstrate the reconfigurability of the chip and the practicality of the training protocol[23]. Our protocol is automated, fast and efficient when training an entire reconfigurable optical chip.

(c) To train an optical neural network for *Iris* classification, which is a showcase of how our system works when training a neural network.

(d) To demonstrate that our method cancels environmental permutations such as imperfect phase implementation resulting from thermal crosstalk and photodetection noise[21, 30], in contrast to off-chip and gradient-based training strategies.

***Program a 1D array of MZIs*** - The structure of the array of MZIs is shown in **Fig. 3a**. Each heater inside the structure cannot be independently characterized. Previous work proposed to scan a multi-dimensional fringe based on the array and fit the collected data to mathematical models using a least-square minimization algorithm[27]. The data volume increases exponentially with number of MZIs. Our protocol requires neither pre-characterisation nor fringe collection.

Let us consider a scenario that we have access to an array of two MZIs. The optical input is a light signal injected from the up input port. The training variables are the currents $\{I_1, I_2\}$ supplied to the two heaters. Our training goal is to achieve an intensity distribution $\widetilde{D} = [1,0]$ (normalized) at the output ports. The fitness function is defined in Equation (1). We denote the normalized output intensity from the respective output ports as $|A_1|^2$ and $|A_2|^2$ $(|A_1|^2 + |A_2|^2 = 1)$. **Figure 3b** shows the measured results of $|A_1|^2$ under meshes of $\{I_1, I_2\}$. The training goal can be interpreted visually as climbing to either one of the hills in **Fig. 3b**. The training process is displayed in the contour map by the evolution from initial population (blue crosses) to the final optimal population (red crosses). The final optimum is located around the hill, meaning the training goal is achieved. In this case several hills have equal heights. The optimum converges randomly to one of them, depending partially on the choice of the initial population.

We test the training protocol when the number of MZIs $n$ increases in the array. Here, the training target is to achieve normalized intensity distributions $\widetilde{D} = [\eta, 1-\eta]$, $\eta \in \{0,0.5,1\}$. Training processes are listed for $n$ equals to 5, 7 and 9. The fitness value of each individual in each generation under each $n$ and $\eta$ are reported in **Fig. 3c**. During the training, the fitness value is observed to converge to zero with increasing generations. The required iterations of optimization increase in order with the learning variables. A final best individual can always be found.

***Program a 6-mode chip to arbitrary crossbar switch*** - The task of a 6-mode crossbar switch is to realize arbitrary connectivity between input ports and output ports. It is a typical task for demonstrating the reconfigurability of the chip and the capability of the training protocol. A 4-mode crossbar switch has been demonstrated by the gradient-based training where the gradient is achieved by finite difference method[23] and also on a specially designed chip where each MZI can be trained by its own local feedback loop[11].

Here, we consider the crossbar switch task as to realize a unique routing from the input intensity distribution to the output intensity distribution. The routing is defined randomly from In 1 → Out 3, In 2 → Out 1, In 3 → Out 5, In 4 → Out 2, In 5 → Out 6 and In 6 → Out 4. The target transformation matrix $\tilde{T}$ under this routing can be expressed by

$$\tilde{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

For the actual transformation matrix T implemented on chip, we collect an output intensity distribution $D \in \mathbb{R}^{6\times6}$ for it in the following manner: Each column of D is collected at all the 6 output ports, when light is injected sequentially from the 1st port, the 2nd port, until the 6th port. A total of 36 values are collected and used for the evaluation of fitness function. Suppose the input light from *i<sup>th</sup>* *(i*=1,…,6*)* port is represented by a column vector $A_{in}^{i}$ with the *i*th element being 1, e.g., $A_{in}^{1} = [1{,}0{,}0{,}0{,}0{,}0]^{T}$ or $A_{in}^{4} = [0{,}0{,}0{,}1{,}0{,}0]^{T}$. Since we are performing intensity detection, the output signal is given by

$$T \cdot A_{in}^{i} = A_{out}^{i} \tag{3a}$$

$$I_{out}^{i} = \left|A_{out}^{i}\right|^{2} = [|T_{1i}|^{2}, |T_{2i}|^{2}, |T_{3i}|^{2}, |T_{4i}|^{2}, |T_{5i}|^{2}, |T_{6i}|^{2}]^{T} \tag{3b}$$

where $I_{out}^{i}$ forms the *i<sup>th</sup>* column vector of *D*, hence the output intensity distribution

$$D = [I_{out}^{1}, I_{out}^{2}, I_{out}^{3}, I_{out}^{4}, I_{out}^{5}, I_{out}^{6}] = |T|^{2} \tag{4}$$

Similarly, the expected output intensity distribution $\widetilde{D}$ which corresponds to the target transformation matrix $\tilde{T}$ is $\widetilde{D} = \left|\tilde{T}\right|^{2}$. The fitness function in equation (1) will be applied here for *D* and $\widetilde{D}$. The training process and optimum results are shown in **Fig. 4a**, in which the *x*-axis are the generations, and the *y*-axis are the fitness value of each individual in current generation. Blue dots with fitness value approaching zero represents a qualified individual, while red the opposite. In the initial population, the fitness values of individuals are widely distributed. We randomly take an individual and observe its response on chip. With the number of generations increasing, the fitness values will converge to be prominently close to zero, thus not distinct to show the similarity between $D$ and $\widetilde{D}$. We define a fidelity to evaluate the performance of the selected individual by

$$F(D,\widetilde{D}) = \frac{1}{6}\sum_{i=1}^{6}\sum_{j=1}^{6}\sqrt{D_{ij}\widetilde{D}_{ij}} \quad (5)$$

The fidelity resulted by this individual is *F = 0.30,* as shown in **Fig. 4b**. The fidelity of an individual randomly taken from the 32nd generation is *F = 0.89*, as shown in **Fig. 4c**. In the final generation, each of the individuals possesses a fitness value very close to 0. We show the best individual whose fidelity is *F = 0.98* in **Fig. 4d**, indicating that the optimum individual in the final generation best configures the optical chip and achieves the target. The statistic information of individuals in the final generation is shown in **Fig. S1**. The currents supplied to the heaters by most individuals are converges to some mean with a standard deviation of 0.13 mA (the maximum supply current is 6.17 mA). The optimum individual is picked from this generation and has the most desired fitness value. The convergence of the generation average and the best individual in the 60 iterations is shown in **Fig. 4e**. The evolution of the output intensity distribution is monitored at each output port during the optimization process as shown in **Fig. 4f**. The six subplots show the average light intensity among all the individuals with each generation. The average crosstalk of the crossbar switch under the optimum set of currents are -23dB. Since our training objective is the entire transformation matrix that contains 36 elements, the trained crossbar switch is capable of routing multiple inputs simultaneously, and accurately delivering the intensity of each input port to the desired output port (see **Fig. S5**).

The algorithm works at any formulations of the optical processor, such as the Fast design chip as reported in **Fig. 2b**. The results of training a balanced design and an FFT design to realize arbitrary crossbar switch are shown in **Figs. S2** and **S3**, respectively. In training an optical circuit with generic topologies, we treat the entire physics-agnostic system as a black box, where we can only access the trainable inputs (i.e., the supplied currents on heaters) and the outputs (i.e., the output intensity measured at each output port). The time complexity of training scales with the number of trainable input variables. It was shown intuitively in Fig. **3c** that the more trainable variables included, the more training iterations required to reach a convergence. Concretely, for an *N*-mode Reck scheme, the time complexity is $O(N(N-1))$, while for an FFT design, the time complexity is $O(Nlog_2N)$. Unlike the Reck scheme that realize universal unitary transform, the computational resources saving in FFT design comes at the expense of implementing the matrix in a subset instead of the entire unitary space. When the target matrix cannot be analytically decomposed for FFT design, our GA training method could help to find a closest substitute on the chip.

We implemented another randomly generalized intensity transformation matrix (T-matrix) by on-chip training. The target distribution is randomly generated as the norm of a unitary matrix. The comparison between target distribution and the measured output intensity distribution under the optimum set of variables is shown in **Fig. S4**. The fidelity $F$ = 0.93 is reported. In implementing the random generalized intensity transformation matrix on multiple inputs, these multiple inputs have the same initial phase.

***Iris classification*** **–** Here, as an example, we train the optical neural network to classify *Iris* dataset. The classification task is for each sample in the dataset, to predict which of the three labels ("1": setosa, "2": versicolor, or "3": virginica) it belongs to. Each instance in the dataset has four features. The dataset is divided into training set and testing set by a ratio of 0.8:0.2. The testing set are unknown to the training process and are evaluated to demonstrate the generalization capability of the trained model. All the testing accuracy in this article refers to the accuracy obtained on the testing set.  The optical inputs of the neural network on-chip are coherent and are encoded by amplitude according to the four-dimensional input vectors of the *Iris* dataset. Concretely, a single laser is injected into the chip. Then the single light signal is divided into four paths by a chain of MZIs on chip (as marked with red blocks in **Fig. 2a** for ease of understanding), simultaneously the amplitude of the light signal in each path is modulated according to the four real-valued features in a one-to-one correspondence, through deliberately configuring the phase shifters of the MZIs. The relative phase of the four light signals is stringently maintained as identical, without being vulnerable to any phase fluctuations (e.g., in fibres), because only a single coherent laser is injected into the chip, and all the light signals are split from it and modulated based on the same phase reference. Therefore, all ports of the neural network layer are excited with light at the same time.

For each four-dimensional input vector, a three-dimensional output vector will be returned. In our implementation, a single complex-valued layer $L^{4\times3} \in \mathbb{C}$ is adopted. The complex-valued matrix is singularly decomposed to two unitary matrices and a diagonal matrix, which are implemented using a single chip. Here under our training protocol, the size of the initial population is 20. We evaluate the fitness function on each individual in the population. The overall fitness function is defined as the sum of the fitness value (i.e., the cross entropy loss) of each sample $f = -\frac{1}{n}\sum_{k=1}^{n}\widetilde{D}_k log({D_k}^T)$, where $n$ = 120 is the number of training sample, $\widetilde{D}_k \in \mathbb{R}^{1\times 3}$ is the ground truth vector, and $D_k \in \mathbb{R}^{1\times 3}$ is the estimated probability distribution – the result of applying a *SoftMax* function to the output intensity distribution. For each sample $k$ in training sets with known labels, if $k$ has label "1", $\widetilde{D}_k = [1{,}0{,}0]$, else if $k$ has label "2", $\widetilde{D}_k = [0{,}1{,}0]$, otherwise if $k$ belongs to label "3", $\widetilde{D}_k = [0{,}0{,}1]$. Correspondingly, the prediction label is determined by the *Argmax* manner, which returns the index of the maximum value in the output vector. The training process and monitored intensity at each of the three output ports are shown in **Fig. 6a**. By randomly selecting an exemplary sample from each species, we show the training process for classifying each of the three species. Here each of the three training samples shown has its own weight matrices. The ultimate goal of the *Iris* task is to find a common weight matrix that classifies the three species. We accomplish the goal by displaying the training and testing curves of 100 generations in **Fig. 6b**. The highest accuracy obtained by the optimum set of parameters is 94.2% on the training dataset and correspondingly 93.3% on the testing dataset, showing that the generalized model fits well to unknown data. As a result, the on-chip training method by genetical algorithms are good at exploring a large and complex space in an intelligent way to solve the problem of training chip-base optical neural networks.

***The comparison to off-chip and gradient-based methods*** – As illustrated in the above sections, our training method for an optical neural network is on-chip and gradient-free. Here, we compare our method to two variations, the off-chip training method and the gradient-based method, respectively. The off-chip training method is widely accepted in existing optical neural networks[4, 12], in which the training stage is executed on electrical computer and only the inference stage is implemented on chip. The detailed scheme is as follows: in the training stage, one trains the objective task on the electrical computer and obtain the weight matrices; in the inference stage, the numerical matrices are decomposed into phase values, and each phase value is converted into a supply power through the calibration curve of corresponding component on the chip. This method requires to establish an accurate analytical model for the chip structure as well as to calibrate the relationship between phase and electrical inputs. Modelling accuracy is restricted because it is challenging to account for various sources of noise, such as calibration errors, thermal crosstalk, photodetection noise and environmental influence, leading to inaccurate phase modulation and ultimately the reduced learning efficacy.

We evaluate the impact of imperfect phase implementation (by simulating thermal crosstalk) and photodetection noise on inference accuracy. The objective task is still the *Iris* classification. **Figure 6a** shows how accuracy is impacted by inference noise, when the network is trained off-chip with 1% Gaussian noise. It is suggested that introducing a certain level of noise into off-chip training would help enhance the model's robustness to noise in physical implementation[21, 30]. We also simulate the on-chip GA-based training in **Fig. 6b**, showing that it effectively compensates for the imperfect phase realization and enhances robustness to photodetection noise. Our on-chip method presents unparalleled advantages in that (1) it utilizes the most fundamental variables – the supply power directly for training, avoiding the complicated calibration process; (2) the feedforward part of the network is implemented on real chip, regarding the hard-to-model factors (e.g., thermal crosstalk among heaters) as part of learning, effectively compensating for the gap between computer-based training and chip-based inference, which typically presents in off-chip method due to the restricted modelling accuracy.

We do not distinguish between gradient-based and gradient-free algorithms in off-chip training, because the training is implemented entirely by electrical computers and does not impose any difference in the inference stage on the physical chip. However, in on-chip training, it is necessary to compare the efficiency and computational cost of the gradient-free and gradient-based methods. To the best of our knowledge, the practical implementation of gradient-based training method on-chip remains an immense hurdle, due to the difficulty of obtaining accurate gradient information on an actual physical chip. There are two ways to calculate the gradient, namely (1) though analytical means via a mathematical mode, and (2) through numerical estimates from experimental data. The analytical method requires calculus to derive a direct formula for the gradient, thereby requiring the explicit analytical model of the physical chip. As established above, this is generally impractical as such a model will need to account for the entire set of free parameters and all sources of noises.

Numerical estimates involve estimating the finite difference $\frac{df(x)}{dx} = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$, where $f(x)$ is the cost function. This would involve (a) recording the multivariable $\vec{x} = [x_1, x_2, \ldots x_n]$, and its on-chip evaluation of cost function $f(\vec{x})$; b) for $x_1$ in $\vec{x}$, add an increment to it and evaluate $f(\vec{x}')$ on current $\vec{x}' = [x_1 + h, x_2, \ldots x_n]$; c) compute $\frac{f(\vec{x}\prime)-f(x)}{h}$ and recover $x_1$ to its original value; d) repeat the step b) and c) for each of the other variables, $x_2, x_3, \ldots$. While this is technically possible, it is both time-consuming and unreliable. Even most naively, this process grows linearly with the number of free parameters. To perform only a single update, we must thus measure the chip as many times as the number of parameters, which is a *N* square complexity with the chip size *N*. In addition, the gradient is numerical unstable as it involves estimating the difference between two very similar numbers – especially as we push $h \ll 1$ when $f$ has a significant second derivative. Whereas in our GA method, the measurement time equals to the number of individuals in a generation, which is a predefined constant value (e.g., 20 as in *Iris* task) and not related to the number of free parameters.

We compare these two methods in their power consumption and latency, as shown in **Table 1**. Each phase shifter on our chip consumes $P = 4.68$ mW at its full load of generating a phase shift of $2\pi$. The photodetector in use takes up $T_1 = 4$ us per measurement. Our thermo-optic phase modulator is 10kHz, yielding a time consumption of $T_2 = 0.1$ ms. The fastest phase modulator under RC-limitation should have a speed of 50 GHz. Each layer of a neural network with N neurons would require $2N^2$ phase shifters (Concretely, an arbitrary $N \times N$ weight matrix is singularly decomposed to two unitary matrices which has $N(N\text{-}1)$ phase shifters each and a diagonal matrix which has $2N$ phase shifters). Therefore, in each iteration of updating all the free parameters, the power consumption is $P \times 2N^2 \times M$ for GA method, where $M$ is the number of individuals, while for numerical gradient method is $P \times 2N^2 \times 2N^2$ because during each evaluation, besides the one under evaluation, all other phase shifters are also served as standing costs. The latency in GA method is $(T_1 + T_2) \times M$, while in numerical gradient method is $(T_1 + T_2) \times 2N^2$. When evaluating each individual in GA, the modulation of all the phase shifters is accomplished simultaneously. Our GA-based training method also exhibits a faster convergence, as shown in **Fig. 6d**. In terms of the blind classification accuracy achieved after convergence, the GA method outperforms numerical gradient method (93.3% vs. 83.3%), whose confusion matrices are shown in **Figs. S6a** and **S6b**, respectively. The weight matrices (trained models) of the two methods are the outcomes (after convergence) from the two curves as shown in **Fig. 6d**, respectively. The testing set are unknown to the training process. The GA method exhibits better generalization capability, compared to the numerical gradient method.

## CONCLUSIONS

We proposed and demonstrated an optical processor embedded with automatic electrical interface, for implementing the gradient-free genetic algorithm. In the training procedure, the feed forward part of the algorithm, which is costly in digital computers, is realized passively on chip by light interference. We demonstrate on-chip training by several proof-of-principle tasks, including training the optical neural network for classification of *Iris* dataset. Our training protocol does not require the knowledge of the chip structures. Therefore, the protocol works for any type of chip structures (i.e., chip with Reck design and Fast design which are demonstrated by our experiments,) even those cannot be analytically decomposed and characterised (i.e., array of MZIs). Because the protocol is physical-agnostic and does not rely on characterisation and modelling, the accuracy is not restricted by the offline modelling of the chip. Factors that affect the accuracy of characterization, such as thermal crosstalk, photodetection noise and other inevitable physical noise, will be incorporated into the training process and effectively eliminated. Most significantly, the protocol does not require gradient information, thus overcomes the difficulty in obtaining gradient information on physical devices.

Although gradient-based algorithms are still the most effective for training large-scale networks on electrical computers, the GA-style and gradient-free algorithms (Particle swarm optimization, Bayesian optimization, to name a few) could be quite advantageous in hybrid systems (e.g., optical neural network whose feedforward part and training part are based on different hardware platforms), where gradient estimations become difficult. The proposed GA-based training can achieve high accuracy on large datasets (e.g. the MNIST and the CIFAR-10 as shown in **Figs. S7** and **S8**, respectively), and is promising to be applied in reinforcement learning[31-32], the reconstruction of unitary transformation[33], training of hybrid quantum computer[34-36]. The scale of network that can be trained with GA can be up to four million free parameters[37] (equivalent to an optical network composed of 60 layers, each layer is 256 × 256), far exceeding the current fabrication capability of optical circuits which is up to 100 modes[38].

## ASSOCIATED CONTENT

**Supplementary information** This material is available free of charge via the internet at http://pubs.acs.org. Brief descriptions in non-sentence format listing the contents of the files supplies as Supporting information.

> The statistic information of individuals in the final optimal generation; The results of realizing crossbar switch on a 4-mode optical processor following balanced design; The results of realizing a crossbar switch on an 8-mode optical processor following Fast design; Programming a 6-mode chip to realize a random T-matrix; Demonstrating the crossbar switch with multiple inputs; Blind classification with GA-based method and numerical gradient-based method; GA-based training for handwriting digit classification; GA-based training for dataset CIFAR-10.

## AUTHOR INFORMATION

The authors declare no competing interests.

## ACKNOWLEDGEMENTS

This work was supported by the Singapore Ministry of Education (MOE) Tier 3 grant (MOE2017-T3-1-001), the Singapore National Research Foundation (NRF) National Natural Science Foundation of China (NSFC) joint grant (NRF2017NRF-NSFC002-014), the Singapore National Research Foundation under the Competitive Research Program (NRF-CRP13-2014-01), the NRF Fellowship reference no. NRF-NRFF2016-02.

## REFERENCES

(1) Miller, D. A., Silicon photonics: Meshing optics with applications. *Nat. Photonics* **2017,** *11*, 403-404.

(2) Pérez, D.; Gasulla, I.; Capmany, J., Programmable multifunctional integrated nanophotonics. *Nanophotonics* **2018,** *7*, 1351-1371.

(3) Miller, D. A., Self-configuring universal linear optical component. *Photonics Res.* **2013,** *1*, 1-15.

(4) Carolan, J.; Harrold, C.; Sparrow, C.; Martín-López, E.; Russell, N. J.; Silverstone, J. W.; Shadbolt, P. J.; Matsuda, N.; Oguma, M.; Itoh, M., Universal linear optics. *Science* **2015,** *349*, 711-716.

(5) Pérez, D.; Gasulla, I.; Crudgington, L.; Thomson, D. J.; Khokhar, A. Z.; Li, K.; Cao, W.; Mashanovich, G. Z.; Capmany, J., Multipurpose silicon photonics signal processor core. *Nat. Commun.* **2017,** *8*, 636.

(6) Perez, D.; Gasulla, I.; Fraile, F. J.; Crudgington, L.; Thomson, D. J.; Khokhar, A. Z.; Li, K.; Cao, W.; Mashanovich, G. Z.; Capmany, J., Silicon photonics rectangular universal interferometer. *Laser Photonics Rev.* **2017,** *11*, 1700219.

(7) Clements, W. R.; Humphreys, P. C.; Metcalf, B. J.; Kolthammer, W. S.; Walmsley, I. A., Optimal design for universal multiport interferometers. *Optica* **2016,** *3*, 1460-1465.

(8) Reck, M.; Zeilinger, A.; Bernstein, H. J.; Bertani, P., Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **1994,** *73*, 58-61.

(9) Flamini, F.; Spagnolo, N.; Viggianiello, N.; Crespi, A.; Osellame, R.; Sciarrino, F., Benchmarking integrated linear-optical architectures for quantum information processing. *Sci. Rep.* **2017,** *7*, 15133.

(10) Zhuang, L.; Roeloffzen, C. G.; Hoekman, M.; Boller, K.-J.; Lowery, A. J., Programmable photonic signal processor chip for radiofrequency applications. *Optica* **2015,** *2*, 854-859.

(11) Ribeiro, A.; Ruocco, A.; Vanacker, L.; Bogaerts, W., Demonstration of a 4× 4-port universal linear circuit. *Optica* **2016,** *3*, 1348-1357.

(12) Shen, Y.; Harris, N. C.; Skirlo, S.; Prabhu, M.; Baehr-Jones, T.; Hochberg, M.; Sun, X.; Zhao, S.; Larochelle, H.; Englund, D., Deep learning with coherent nanophotonic circuits. *Nat. Photonics* **2017,** *11*, 441-446.

(13) Tillmann, M.; Dakić, B.; Heilmann, R.; Nolte, S.; Szameit, A.; Walther, P., Experimental boson sampling. *Nat. photonics* **2013,** *7*, 540-544.

(14) Wilkes, C. M.; Qiang, X.; Wang, J.; Santagati, R.; Paesani, S.; Zhou, X.; Miller, D. A.; Marshall, G. D.; Thompson, M. G.; O'Brien, J. L., 60 dB high-extinction auto-configured Mach–Zehnder interferometer. *Opt. Lett.* **2016,** *41*, 5318-5321.

(15) Harris, N. C.; Steinbrecher, G. R.; Prabhu, M.; Lahini, Y.; Mower, J.; Bunandar, D.; Chen, C.; Wong, F. N.; Baehr-Jones, T.; Hochberg, M., Quantum transport simulations in a programmable nanophotonic processor. *Nat. Photonics* **2017,** *11*, 447-452.

(16) Shadbolt, P. J.; Verde, M. R.; Peruzzo, A.; Politi, A.; Laing, A.; Lobino, M.; Matthews, J. C.; Thompson, M. G.; O'Brien, J. L., Generating, manipulating and measuring entanglement and mixture with a reconfigurable photonic circuit. *Nat. Photonics* **2012,** *6*, 45-49.

(17) Jing, L.; Shen, Y.; Dubcek, T.; Peurifoy, J.; Skirlo, S.; LeCun, Y.; Tegmark, M.; Soljačić, M. In *Tunable efficient unitary neural networks (EUNN) and their application to RNNs*, Proceedings of the 34th International Conference on Machine Learning, JMLR. org: 2017; pp 1733-1741.

(18) Fang, M. Y.-S.; Manipatruni, S.; Wierzynski, C.; Khosrowshahi, A.; DeWeese, M. R., Design of optical neural networks with component imprecisions. *Opt. Express* **2019,** *27*, 14009-14029.

(19) Cheng, Z.; Ríos, C.; Pernice, W. H.; Wright, C. D.; Bhaskaran, H., On-chip photonic synapse. *Sci. Adv.* **2017,** *3*, e1700160.

(20) Tait, A. N.; De Lima, T. F.; Zhou, E.; Wu, A. X.; Nahmias, M. A.; Shastri, B. J.; Prucnal, P. R., Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* **2017,** *7*, 7430.

(21) Mehrabian, A.; Miscuglio, M.; Alkabani, Y.; Sorger, V. J.; El-Ghazawi, T., A Winograd-based Integrated Photonics Accelerator for Convolutional Neural Networks. *IEEE J. Sel. Top. Quantum Electron.* **2019,** *26*, 6100312.

(22) Miscuglio, M.; Sorger, V. J., Photonic tensor cores for machine learning. *Appl. Phys. Rev.* **2020,** *7*, 031404.

(23) Zhou, H.; Zhao, Y.; Wang, X.; Gao, D.; Dong, J.; Zhang, X., Self-learning photonic signal processor with an optical neural network chip. *arXiv preprint* **2019,** *arXiv:1902.07318*.

(24) Hughes, T. W.; Minkov, M.; Shi, Y.; Fan, S., Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* **2018,** *5*, 864-871.

(25) Koza, J. R.; Poli, R., Genetic programming. In *Search methodologies*, Springer: 2005; pp 127-164.

(26) Zhang, T.; Wang, J.; Dan, Y.; Lanqiu, Y.; Dai, J.; Han, X.; Sun, X.; Xu, K., Efficient training and design of photonic neural network through neuroevolution. *Opt. Express* **2019,** *27*, 37150-37163.

(27) Qiang, X.; Zhou, X.; Wang, J.; Wilkes, C. M.; Loke, T.; O'Gara, S.; Kling, L.; Marshall, G. D.; Santagati, R.; Ralph, T. C., Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. *Nat. Photonics* **2018,** *12*, 534-539.

(28) Zheng, S. N.; Zou, J.; Cai, H.; Song, J.; Chin, L.; Liu, P.; Lin, Z.; Kwong, D.; Liu, A. Q., Microring resonator-assisted Fourier transform spectrometer with enhanced resolution and large bandwidth in single chip solution. *Nat. Commun.* **2019,** *10*, 2349.

(29) Baker, J. E. In *Reducing bias and inefficiency in the selection algorithm*, Proceedings of the second international conference on genetic algorithms, 1987; pp 14-21.

(30) Milanizadeh, M.; Aguiar, D.; Melloni, A.; Morichetti, F., Canceling thermal cross-talk effects in photonic integrated circuits. *J. Lightwave Technol.* **2019,** *37*, 1325-1332.

(31) Stanley, K. O.; Clune, J.; Lehman, J.; Miikkulainen, R., Designing neural networks through neuroevolution. *Nat. Mach. Intell.* **2019,** *1*, 24-35.

(32) Such, F. P.; Madhavan, V.; Conti, E.; Lehman, J.; Stanley, K. O.; Clune, J., Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint* **2017,** *arXiv:1712.06567.*

(33) Spagnolo, N.; Maiorino, E.; Vitelli, C.; Bentivegna, M.; Crespi, A.; Ramponi, R.; Mataloni, P.; Osellame, R.; Sciarrino, F., Learning an unknown transformation via a genetic approach. *Sci. Rep.* **2017,** *7*, 14316.

(34) Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A., A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf.* **2019,** *5*, 45.

(35) Zhu, D.; Linke, N. M.; Benedetti, M.; Landsman, K. A.; Nguyen, N. H.; Alderete, C. H.; Perdomo-Ortiz, A.; Korda, N.; Garfoot, A.; Brecque, C., Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **2019,** *5*, eaaw9918.

(36) Beer, K.; Bondarenko, D.; Farrelly, T.; Osborne, T. J.; Salzmann, R.; Scheiermann, D.; Wolf, R., Training deep quantum neural networks. *Nat. Commun.* **2020,** *11*, 808.

(37) Such, F. P.; Madhavan, V.; Conti, E.; Lehman, J.; Stanley, K. O.; Clune, J., Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567* **2017**.

(38) Paesani, S.; Ding, Y.; Santagati, R.; Chakhmakhchyan, L.; Vigliar, C.; Rottwitt, K.; Oxenløwe, L. K.; Wang, J.; Thompson, M. G.; Laing, A., Generation and sampling of quantum states of light in a silicon chip. *Nat. Phys.* **2019,** *15*, 925-929.

# FIGURE LEGENDS

**Figure 1.** The optical processor with electrical interface for implementing GA-based training protocol. (a) Interaction between the optical chip and the electrical interface. (b) An integrated hybrid optical processor comprising an optical neural network circuit and a classical logic unit. (c) The flow of genetic-algorithm-based training mechanism on an optical processor. Exemplary crossover and mutation operators are displayed.

**Figure 2.** Fabrication results of a 6-mode 2-layered optical neural network integrated on silicon photonic chip. (a) A 6-mode two layered linear optical processor. Most results shown in manuscript are obtained by testing on this chip. (b) An 8-mode non-universal linear optical processor fabricated following the butterfly-like design as inspired by Fast Fourier Transform. The design is tested for demonstrating that the physics-agnostic attribute of our training method.

**Figure 3.** Program a 1D array of MZIs to realize designed intensity distribution. (a) Diagram of the array of MZIs. (b) The normalized output intensity under meshes of $\{I_1, I_2\}$, considering the case where we have access to an array of two MZIs. In the contour map, the blue crosses are the initial population, and the red crosses are the final optimum results. (c) The training process when testing scaling number of MZIs. The target $\eta$ is set as 0, 0.5 and 1. Numbers of phase shifters in test are 5, 7 and 9.

**Figure 4.** Program a 6-mode chip to arbitrary crossbar switch. (a) The process of training the optical processor by showing the fitness value of each individual (y-axis) in each generation (x-axis). As the generations increase, the fitness value approaches zero. (b) In the initial population, the fitness values of individuals are widely distributed. We randomly take individuals from the initial population and show the corresponding chip response. (c) Chip response of an individual randomly selected during the training. (d) The final optimum result, which achieves a fidelity of 0.98. (e) The convergence of mean fitness value and best fitness value during training is shown. (f) The evolution of the output intensity distribution monitored at each output port during the optimization process.

**Figure 5.** On-chip training of an optical neural network for *Iris* classification. (a) Training the chip to classify an exemplary sample from each species. The learning process and monitored outputs are as shown. (b) The on-chip training process. The highest accuracy obtained by the optimum set of parameters is 94.2% on the training dataset. The corresponding accuracy on testing dataset are 93.3%.

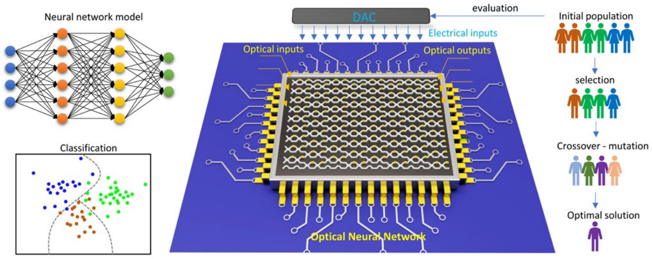
**Figure 6.** The comparison of our on-chip training method to the off-chip training method, by simulating the impact of imperfect phase implementation and photodetection noise. (a) Off-chip training with 1% Gaussian noise. (b) On-chip GA-based training. Our on-chip method effectively compensates the imperfect phase implementation and enhances the robustness to noise. (d) The comparison of the training curves of our GA-based and gradient-free method to the gradient-based method. Both are implemented on-chip, and our method exhibits a much faster convergence.

**Table 1: The comparison of GA method and numerical gradient method in power consumption and latency**

| | Scaling | | Merits under size | |
|---|---|---|---|---|
| | GA method | Numerical gradient | N=4 | N=10 |
| Power consumption | $2N^2 \times M \times P$ | $2N^2 \times 2N^2 \times P$ | 2.99 W *vs.* 4.79 W | 18.72 W *vs.* 187.2 W |
| Latency | $(T_1 + T_2) \times M$ | $(T_1 + T_2) \times 2N^2$ | 2.08 ms *vs.* 3.33 ms | 2.08 ms *vs.* 20.8 ms |

($P$ = 4.68 mW, $T_1$ = 4 us and $T_2$ = 0.1 ms in our setup, assume $M$ = 20)

## For Table of Contents Use Only



Title: Efficient On-Chip Training of Optical Neural Networks Using Genetic Algorithm

Authors: Hui Zhang, Jayne Thompson, Mile Gu, Xudong Jiang, Hong Cai, Patricia Yang Liu, Yuzhi Shi, Yi Zhang, Muhammad Faeyz Karim, Guo Qiang Lo, Xianshu Luo, Bin Dong, Leong Chuan Kwek, and Ai Qun Liu

Brief synopsis: An efficient, physics-agnostic, and closed-loop protocol for training optical neural networks on chip is demonstrated experimentally. The gradient-free training scheme based on genetic algorithm performs well on various types of optical circuits, especially those that are difficult to calibrate.