FLOPS: EFficient On-Chip Learning for OPtical Neural Networks Through Stochastic Zeroth-Order Optimization

Jiaqi Gu[1*], Zheng Zhao[1], Chenghao Feng[1], Wuxi Li[2], Ray T. Chen[1], and David Z. Pan[1†]

[1]ECE Department, The University of Texas at Austin [2]Xilinx Inc.
[*]jqgu@utexas.edu; [†]dpan@ece.utexas.edu

Abstract— **Optical neural networks (ONNs) have attracted extensive attention due to its ultra-high execution speed and low energy consumption. The traditional software-based ONN training, however, suffers the problems of expensive hardware mapping and inaccurate variation modeling while the current on-chip training methods fail to leverage the self-learning capability of ONNs due to algorithmic inefficiency and poor variation-robustness. In this work, we propose an on-chip learning method to resolve the aforementioned problems that impede ONNs' full potential for ultra-fast forward acceleration. We directly optimize optical components using stochastic zeroth-order optimization on-chip, avoiding the traditional high-overhead back-propagation, matrix decomposition, or *in situ* device-level intensity measurements. Experimental results demonstrate that the proposed on-chip learning framework provides an efficient solution to train integrated ONNs with 3$\sim$4$\times$ fewer ONN forward, higher inference accuracy, and better variation-robustness than previous works.**

I. Introduction

As Moore's Law winds down, it becomes challenging for electronic digitals to meet with escalating computational demands of machine learning applications. In recent years, the optical neural network (ONN) has been demonstrated to be an emerging neuromorphic platform with ultra-low latency, high bandwidth, and high energy efficiency, which becomes a promising alternative to its traditional electronic counterpart. Computationally-intensive operations in neural networks, e.g., matrix multiplication, can be efficiently realized by optics at the speed of light [1]–[4]. Shen, *et al.* [1] demonstrated a classical integrated fully-optical neural network platform to implement a multi-layer perceptron. Based on matrix singular value decomposition (SVD) and unitary parametrization [5], [6], the weight matrices are mapped onto cascaded Mach-Zehnder interferometer (MZI) arrays to realize ultra-fast neural network inference with over 100 GHz photo-detection rate and near-zero energy consumption [7].

However, training methodologies for integrated ONNs still have limited efficiency and performance these days. The mainstream approach is to offload the training process to pure software engines, obtain pre-trained weight matrices using classical back-propagation (BP) algorithm, and then map the trained model onto photonic hardware through matrix decomposition and unitary parametrization [1], [2]. This traditional method benefits from off-the-shelf deep learning toolkits for easy training of weight matrices under a noise-free environment. Nevertheless, this inevitably bears several downsides in efficiency, performance, and robustness. First, pure software-based ONN training is still bottlenecked by the performance of digital computers, which suffers inefficiency when emulating the actual ONN architectures given the expensive computational cost required in matrix decomposition and parametrization. Hence, there exist great potentials to fully utilize ultra-fast photonic chips to accelerate the training process as well. Second, photonic chips could be exposed to various non-ideal effects [1], [8]–[10], e.g., device manufacturing variations, limited phase encoding precision, and thermal cross-talk, thus the ONN model trained by pure software methods could suffer severe performance degradation and poor variation-robustness for lack of accurate non-ideality modeling. Simply mapping the back-propagation algorithm onto photonic chips is conceptually straightforward but may not fully exploit the self-learning capability of ONN chips. Back-propagation is technically difficult to be implemented on chip, especially considering the expensive hardware overhead and time-consuming matrix decomposition procedure. Also, decoupling the ONN hardware and software training process disables potential online learning applications.

A brute-force phase tuning algorithm is proposed and adopted in [1], [11] to perform ONN on-chip training. Each MZI phase is individually perturbed by a small value and then updated based on the function evaluation results. This greedy parameter search algorithm is intractable when tuning a large number of phases and may not be robust enough to handle non-ideal variations. To mitigate the inefficiency issue of the above brute-force algorithm, an *in situ* adjoint variable method (AVM) [12] is applied to perform ONN on-chip training. This inverse design method directly computes the gradient w.r.t. MZI phases by propagating through the chip back-and-forth for several times. However, it strictly assumes the photonic system is fully-observable and requires light field intensity measurement on each device, which is technically challenging to scale to larger systems. Evolutionary algorithms, e.g., genetic algorithm (GA) and particle swarm optimization (PSO), are applied to train ONNs by emulating the biological evolution process [13] with a large population.

The above on-chip training protocols are only demonstrated to handle small photonic systems with $<$100 MZIs, while in this work, we propose a novel framework `FLOPS` that extends the learning scale to $\sim$1000 MZIs with higher training efficiency, higher inference accuracy, and better robustness to non-ideal thermal variations. Compared with the previous state-of-the-art methods [1], [11], [13], our on-chip learning framework `FLOPS` has the following advantages.

- Efficiency: our learning method leverages stochastic zeroth-order optimization with a parallel mini-batch based gradient estimator and achieves 3$\sim$4$\times$ fewer ONN forward than previous methods.
- Accuracy: our proposed optimization algorithm is extended with a light-weight second-stage learning procedure `SparseTune` to perform sparse phase tuning, achieving further accuracy boost while the efficiency superiority still maintains.
- Robustness: our method is demonstrated to improve the test accuracy of ONNs under thermal cross-talk and produces better variation-robustness than previous methods.

The remainder of this paper is organized as follows. Section II introduces the ONN architecture and previous training methods. Section III discusses details of our proposed framework. Section IV analyzes `FLOPS` in the presence of thermal variations. Section V evaluates the accuracy, efficiency, and robustness improvement of `FLOPS` over previous works, followed by the conclusion in Section VI.

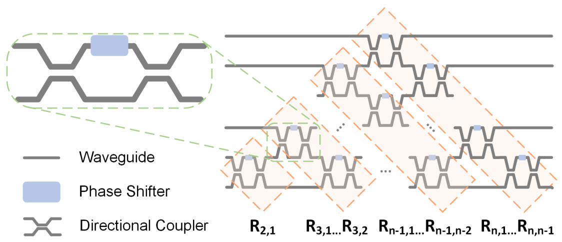


Fig. 1: Schematic of an MZI triangular array and a closeup view of the MZI structure.

II. Preliminaries

In this section, we will introduce the architecture of integrated ONNs, current ONN training methods, and background knowledge about stochastic zeroth-order optimization with gradient estimation.

A. ONN Architecture and Training Methods

The ONN is a photonic implementation of artificial neural networks. It implements matrix multiplication with optical inference units (OIU) consisting of cascaded MZIs, shown in Fig. 1. Each MZI is a reconfigurable photonic component that can produce interference of input light signals as follows [1],

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{1}$$

The phase shift ϕ produced in the coupling region of an MZI can be achieved by an optical phase shifter. The triangular structure of an MZI array shown in Fig. 1 is proven to realize a unitary matrix based on unitary group parametrization [1], [5],

$$\boldsymbol{U}(n) = \boldsymbol{D} \prod_{i=n}^{2} \prod_{j=1}^{i-1} \boldsymbol{R}_{ij}. \tag{2}$$

where the 2-dimensional planar rotator $\boldsymbol{R_{ij}}$ is an identity matrix except that entries at (i,i), (i,j), (j,i), (j,j) are $\cos\phi$, $\sin\phi$, -$\sin\phi$, $\cos\phi$, respectively. $\boldsymbol{D}$ is a matrix where only its main diagonal has -1 or 1. Each $\boldsymbol{R}_{ij}$ can be implemented by an MZI in the OIU. We denote all parametrized phases as a phase vector $\boldsymbol{\Phi}$. Since an $m \times n$ weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ can be decomposed via singular value decomposition (SVD) $\boldsymbol{W} = \boldsymbol{U\Sigma V^*}$, where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{V}^* \in \mathbb{R}^{n\times n}$ are square unitary matrices and $\boldsymbol{\Sigma}$ is an $m \times n$ non-negative diagonal matrix, we can map $\boldsymbol{U}$ and $\boldsymbol{V}^*$ onto two OIUs and $\boldsymbol{\Sigma}$ onto optical attenuators/amplifiers to realize vector-matrix multiplication.

A training procedure is needed to determine the configuration of each MZI. Traditionally, this can be achieved by pure software training based on back-propagation (BP) algorithm, and then map those pre-trained matrices onto MZI arrays through SVD and Eq. (2). This BP-based training decouples the software and hardware implementation, leading to efficiency limitation by electronic computers and robustness issues due to inaccurate variation modeling. Instead of training the weight matrices, a possible alternative approach is to learn MZI configurations on chip directly. As the parametrization procedure is differentiable, we can derive the theoretical derivative of an unitary matrix $\boldsymbol{U}$ w.r.t. a phase ϕ_{ij} [1],

$$\frac{\partial \boldsymbol{U}}{\partial \phi_{ij}} = \boldsymbol{D}\boldsymbol{R}_{n1}\boldsymbol{R}_{n2}\boldsymbol{R}_{n3}\cdots\frac{\partial \boldsymbol{R}_{ij}}{\partial \phi_{ij}}\cdots\boldsymbol{R}_{31}\boldsymbol{R}_{32}\boldsymbol{R}_{21}. \tag{3}$$

However, the computational cost of this theoretical derivative is prohibitive to be involved in the BP-based training.

To utilize the ultra-fast speed of ONN forward propagation, a brute-force method [1], [11] that iteratively updates each phase is proposed for MZI tuning. After $\mathcal{O}(|\boldsymbol{\Phi}|)$ number of function queries, all phases are updated once. Adjoint variable method (AVM) [12] is a recently proposed inverse design method that computes the exact gradient based on *in situ* intensity measurement, which is intrinsically unscalable as it is based on expensive device-level signal detection. Evolutionary algorithms, e.g., PSO, are also applied to search for optimal MZI configurations in the phase space [13]. However, a high-dimensional parameter space could lead to optimization inefficiency and unsatisfying optimality as they typically require a large population and inevitably faces pre-mature issues.

B. Optimization with Zeroth-Order Gradient Estimation

Zeroth-order optimization (ZOO) has received much attention recently [14], [15] as it enables effective optimization when the explicit gradient of the objective function is infeasible, e.g., reinforcement learning and black-box adversarial attack on DNNs. ZOO gains much momentum as it is capable of handling higher-dimensional problems than traditional methods, e.g., Bayesian optimization, and can be integrated with extensive state-of-the-art first-order optimization algorithms with the true gradient being replaced by the approximated zeroth-order gradient [14], [15]. ZOO uses a Gaussian function approximation to estimate the local value of the original function $f(\theta)$ as follows,

$$f_\sigma(\theta) = \mathbb{E}_{\Delta\theta}\Big[f(\theta+\Delta\theta)\Big] = \frac{1}{\sigma\sqrt{(2\pi)^d}}\int f(\theta+\Delta\theta)e^{\frac{-||\Delta\theta||_2^2}{2\sigma^2}}d\Delta\theta, \tag{4}$$

where $\Delta\theta$ is a perturbation sampled from Gaussian distribution $\mathcal{N}(0,\sigma^2)$. Nesterov *et al.* [16] proves a bound of the difference between the true gradient and its Gaussian approximation if the original function $f(\theta)$ is β-smooth,

$$||\nabla f_\sigma(\theta) - \nabla f(\theta)|| \leq \frac{\sigma}{2}\beta(d+3)^{3/2}, \quad \forall\theta \in \mathbb{R}^d. \tag{5}$$

Based on this, we can approximate the true gradient $\nabla f(\theta)$ through estimating this surrogate gradient $\nabla f_\sigma(\theta)$ with the above error bound. To estimate $\nabla f_\sigma(\theta)$, a symmetric difference quotient method is typically used,

$$\hat{\nabla} f(\theta) = \frac{1}{2\sigma^2}(f(\theta+\Delta\theta) - f(\theta-\Delta\theta))\Delta\theta. \tag{6}$$

Such a sampling-based first-order oracle is an unbiased estimator of $f_\sigma(\theta)$. Passing this zeroth-order gradient estimation to a first-order optimization algorithm, e.g., gradient descent, we can perform optimization only with function queries.

III. On-Chip ONN Training based on Zeroth-order Gradient Estimation

A. Phase Domain Characterization

To better characterize the optimization problem of ONN on-chip training, we illustrate details of our optimization domain. Back-propagation based methods optimize in the *weight matrix domain*, while on-chip learning methods optimize in the latent *phase domain*. Bridged by SVD and unitary parametrization (Eq. (2)), the above two domains can switch to each other equivalently. However, co-optimization between those two domains is theoretically difficult. First, two domains are fully coupled and the transformation is highly-nonlinear and not element-wise. Any change in a phase represents a high-dimensional rotation of the weight matrix, thus leading to perturbation of all entries in the weight matrix, and vice versa. Besides, the phase domain is not an unbounded space as the weight domain but a high-dimensional hypercube with a valid phase shift range of [-π, π). Though this validity constraint can be guaranteed by projection onto a feasible set, it will cause optimization performance penalty. Since

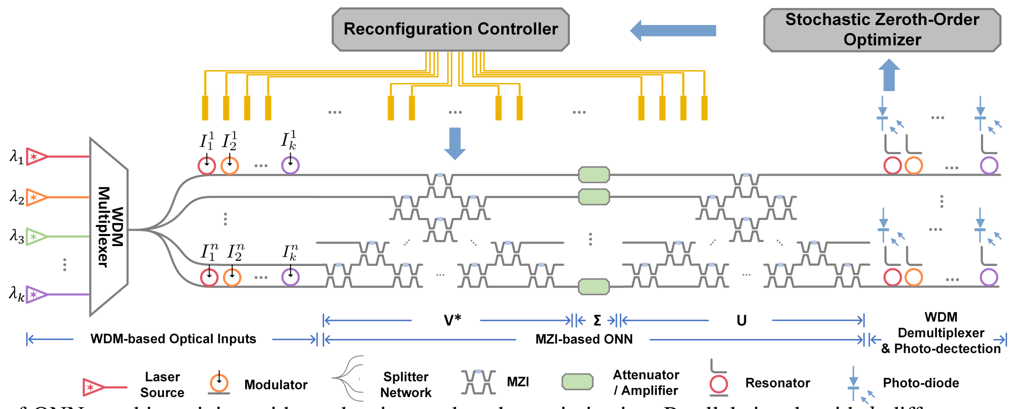


Fig. 2: Framework of ONN on-chip training with stochastic zeroth-order optimization. Parallel signals with k different wavelengths are shown.

phases will intrinsically wrap around to the valid range, the solution space can also be viewed as a periodically expanded space, such that the validity constraint can be relaxed, shown in Fig. 4. This feature is leveraged in our optimization algorithm in later sections. The above analysis on phase domain characteristics casts both theoretical and practical difficulty on possible co-optimization between the weight matrix and the phase domain, which provides a strong motivation for us to design an efficient on-chip learning method directly in the phase space.

B. *On-Chip Learning with Zeroth-Order Gradient Estimation*

As shown in Fig. 2, an ONN consists of cascaded MZIs configured with external controls. We denote all programmable MZI phases as $\mathbf{\Phi}$. During ONN on-chip training, the final objective function $\mathcal{L}$ is optimized based on the following gradient descent formulation,

$$\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \alpha\, \hat{\nabla}_{\mathbf{\Phi}}\mathcal{L}, \tag{7}$$

where $\hat{\nabla}_{\mathbf{\Phi}}\mathcal{L}$ is the zeroth-order estimation whose expectation approximates the true gradient $\nabla_{\mathbf{\Phi}}\mathcal{L}$ with an error bound shown in Eq.(5). Similar to Eq.(6), the stochastic zeroth-order gradient can be evaluated on a mini-batch as,

$$\hat{\nabla}_{\mathbf{\Phi}}\mathcal{L} = \frac{1}{2\sigma^2|\mathcal{S}|}\sum_{i=0}^{|\mathcal{S}|}\big(\mathcal{L}(x_i;\mathbf{\Phi}+\mathbf{\Delta\Phi}_i) - \mathcal{L}(x_i;\mathbf{\Phi}-\mathbf{\Delta\Phi}_i)\big)\mathbf{\Delta\Phi}_i, \tag{8}$$

where x_i is one example in a mini-batch $\mathcal{S}$; $\mathbf{\Delta\Phi}_i$ is a random high-dimensional perturbation sampled from a multivariate Gaussian distribution $\mathcal{N}(0,\sigma^2)$. However, the optimization performance of stochastic gradient descent based methods highly depends on the accuracy of gradient estimation [15]. High variance in gradient estimation could generally lead to a slow convergence rate and degraded solution optimality. In this section, we will focus on the adopted techniques and theoretical analysis to improve ONN on-chip training efficiency and accuracy by reducing the computational and sampling complexity as well as minimizing the gradient estimation error.

1) On-Chip Learning Efficiency Improvement: High efficiency is one of the major targets when performing ONN on-chip training. To efficiently leverage the ultra-fast ONN hardware platform to estimate the zeroth-order gradient, we propose to use a parallel mini-batch based asymmetric gradient estimator as follows,

$$\begin{aligned}\mathcal{L}_{\mathcal{S}}(x;\mathbf{\Phi}) &= \frac{1}{|\mathcal{S}|}\sum_{i=0}^{|\mathcal{S}|-1}\mathcal{L}(x_i;\mathbf{\Phi}),\\ \hat{\nabla}_{\mathbf{\Phi}}\mathcal{L} &= \frac{1}{\sigma^2}\big(\mathcal{L}_{\mathcal{S}}(x;\mathbf{\Phi}+\mathbf{\Delta\Phi}) - \mathcal{L}_{\mathcal{S}}(x;\mathbf{\Phi})\big)\mathbf{\Delta\Phi},\end{aligned} \tag{9}$$

where $\mathcal{L}_{\mathcal{S}}(\cdot)$ averages the loss over a mini-batch. Two reasons account for the superior efficiency of the proposed estimator. First, a sample-efficient asymmetric estimator replaces its symmetric counterpart Eq. (8) [16] to achieve fewer function queries. Second, this estimation is performed in parallel with a fixed perturbation $\mathbf{\Delta\Phi}$ used for all examples within a mini-batch $\mathcal{S}$. Thus, this method is at least $|\mathcal{S}|$ times more efficient than the single-example based method (Eq. (8)). Specifically, it eliminates expensive averaging operations over $|\mathcal{S}|$ length-d gradient samples, while only a cheap averaging operation on scalar loss functions is required. Moreover, our method shares the same $\mathbf{\Delta\Phi}$ in a mini-batch, leading to $|\mathcal{S}|$ times fewer Gaussian variable samples than its single-example based counterpart.

From the perspective of hardware implementation, this parallel gradient estimation can be achieved by a readily available wavelength-division multiplexing (WDM) technique that enables fully parallel optical signal processing [17], [18]. As shown in Fig. 2, a mini-batch of input data, e.g., 16 or 32, can be encoded into parallel optical signals with $k = |\mathcal{S}|$ different wavelengths and then input into the ONN chip through the same waveguides. Different output wavelengths can be filtered and separated by corresponding WDM de-multiplexer, and finally detected by photodiode arrays in the end.

Gradient estimation based on Eq. (9) costs only two function queries, thus can lead to convergence issues due to a large variance, especially with a larger dimension d. In the following section, we will focus on variance analysis and reduction techniques.

2) On-Chip Learning Accuracy Improvement: To reduce the gradient estimation variance, we adopt its sample average with $Q+1$ function queries as follows,

$$\hat{\nabla}_{\mathbf{\Phi}}\mathcal{L} = \frac{1}{Q\sigma^2}\sum_{q=0}^{Q-1}\big(\mathcal{L}_{\mathcal{S}}(x;\mathbf{\Phi}+\mathbf{\Delta\Phi}_q) - \mathcal{L}_{\mathcal{S}}(x;\mathbf{\Phi})\big)\mathbf{\Delta\Phi}_q, \tag{10}$$

where the sampling factor Q is the number of independent perturbations used to calculate the sample average of gradient estimation.

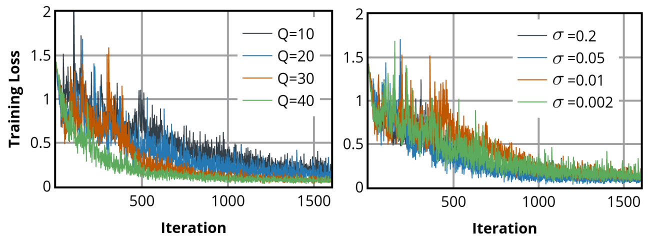
We show the variance bound of this parallel mini-batch based asymmetric zeroth-order gradient estimator. First, a standard assumption of stochastic gradient descent is given on the upper bound of the variance of the stochastic gradient on a mini-batch [16]. If the original function $\mathcal{L}$ is β-smooth, we can assume

$$\mathbb{E}_{\mathcal{S}}[||\nabla_{\mathbf{\Phi}}\mathcal{L}_{\mathcal{S}} - \nabla_{\mathbf{\Phi}}\mathcal{L}||^2] \le \sigma_s^2. \tag{11}$$

Based on the above assumption, the variance upper bound of stochastic zeroth-order gradient estimator [16] is derived as,

$$\mathbb{E}_{\mathcal{S},\mathbf{\Delta\Phi}}[||\hat{\nabla}_{\mathbf{\Phi}}\mathcal{L} - \nabla_{\mathbf{\Phi}}\mathcal{L}_{\sigma}||^2] \le \mathcal{O}\Big(\frac{\sigma^2\beta^2 d^3}{Q} + \frac{\sigma_s^2 d}{|\mathcal{S}|Q} + \frac{||\nabla_{\mathbf{\Phi}}\mathcal{L}||^2 d}{Q}\Big). \tag{12}$$

The above theoretical conclusion implies that increasing the sampling factor Q can effectively minimize the gradient estimation



(a) (b)

Fig. 3: (a) Training curve with different sampling factor Q; (b) training curve with different sampling variances σ. A 3-layer ONN with configuration of 8-16-16-4 is used, where 16 represents 16 neurons at that hidden layer.

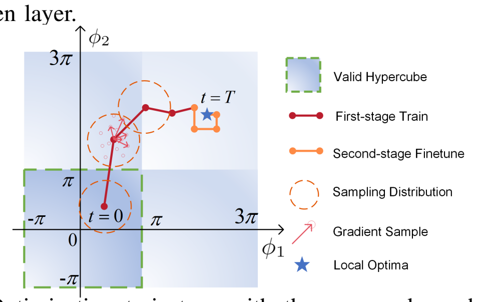


Fig. 4: Optimization trajectory with the proposed on-chip training algorithm in the relaxed, periodic phase space.

variance as illustrated in Fig. 3a. Besides, a smaller sampling variance σ theoretically reduces the first term of the variance upper bound, but it is not sensitive within a wide range as observed in Fig. 3b. Hence, during ONN on-chip training, the sampling factor, and mini-batch size are major tunable hyperparameters that can achieve a trade-off between gradient estimation error and function query complexity under certain ONN forward budget.

This zeroth-order gradient estimation based method quickly explores in the phase space till a roughly converged solution, as shown in Fig. 4, but it may still have some accuracy degradation due to stochastic gradient sampling error (Eq.(12)). If more function queries are allowed to recover the accuracy loss, we propose to extend this algorithm with `SparseTune`, a light-weight fine-tuning procedure based on random coordinate-wise phase tuning. The complete two-stage on-chip learning algorithm is described in Alg. 1. At this second-stage `SparseTune`, a randomly selected subset of phases $\{\phi_i\}_{i=1}^{M}$ with cardinality M are sequentially tuned for each individual coordinate, shown in the second part of optimization trajectory in Fig. 4. This sparse tuning costs more function queries per iteration than the first stage as $M > Q$, but it is overall more efficient than brute-force training as $M \ll d$ and $T_f \ll T$. It is effective as it leverages the sparsity assumption in the parameter space [19] for variance reduction, thus can boost the optimization performance with less expensive parameter sweeping.

IV. Robust ONN Learning with *in situ* Thermal Variation

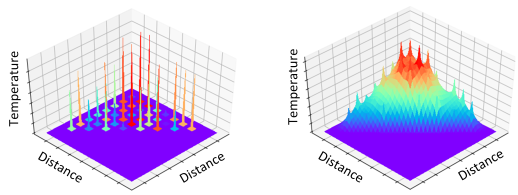
As a high-performance analog neuromorphic platform, ONN inevitably encounters robustness issues that could lead to possible accuracy degradation, where thermal cross-talk is among one of the most critical concerns [1], [9]. In this section, we will justify the robustness advantages of our on-chip learning method over traditional software training with thermal cross-talk.

Thermo-optic phase shifters are widely used to configure the MZI arrays on the ONN chip. Cross-talk exists among nearby devices, e.g., between two phase shifters or between phase shifters and waveguides, by influencing their relative refractive index n, which is difficult to

Algorithm 1 ONN On-Chip Training With Zeroth-Order Optimization

Input: ONN forward function $\mathcal{L}(\cdot)$, initial MZI phases $\mathbf{\Phi}^0$, training dataset X, initial learning rate α^0, total iterations T, starting iteration for `SparseTune` T_f, cardinality of finetuned phases M, and initial tuning step size $\delta\phi^0$;
Output: Converged phases $\mathbf{\Phi}^{T-1}$;

```
 1: for $t \leftarrow 0 \cdots T_f - 1$ do                                   ▷ First stage training
 2:     $\mathcal{L}_{\mathcal{S}}(x^t; \mathbf{\Phi}^t), \quad x^t \sim X$        ▷ ONN forward on a mini-batch
 3:     $\{\mathbf{\Delta\Phi}_0^t, \cdots, \mathbf{\Delta\Phi}_{Q-1}^t\} \sim \mathcal{N}(0, (\sigma^t)^2 \mathbb{G}^t)$        ▷ Sample $\Delta\mathbf{\Phi}$
 4:     $\hat{\nabla}_{\mathbf{\Phi}^t}\mathcal{L} = \frac{1}{Q(\sigma^t)^2}\sum_{q=0}^{Q-1}\left(\mathcal{L}_{\mathcal{S}}(x^t; \mathbf{\Phi}^t + \mathbf{\Delta\Phi}_q^t) - \mathcal{L}_{\mathcal{S}}(x^t; \mathbf{\Phi}^t)\right)\mathbf{\Delta\Phi}_q^t$
 5:     $\hat{\mathbf{\Phi}}^t \leftarrow \mathbf{\Phi}^t - \alpha^t \hat{\nabla}_{\mathbf{\Phi}^t}\mathcal{L}$        ▷ Phase updating
 6:     $\alpha^{t+1} = \texttt{Update}(\alpha^t)$        ▷ Learning rate decay
 7: for $t \leftarrow T_f \cdots T - 1$ do        ▷ Second stage sparse tuning
 8:     Randomly sample a mini-batch $x^t$ from $X$
 9:     Randomly select a set of phases $\{\phi_i\}_{i=1}^{M} \subseteq \mathbf{\Phi}^t$
10:     for each phase $\phi_i \in \{\phi_i\}_{i=1}^{M}$ do
11:         if $\mathcal{L}(x^t; \phi_i^t + \delta\phi^t) < \mathcal{L}(x^t; \phi_i^t)$ then
12:             $\phi_i^{t+1} \leftarrow \phi_i^t + \delta\phi^t$
13:         else
14:             $\phi_i^{t+1} \leftarrow \phi_i^t - \delta\phi^t$
15:     $\delta\phi^{t+1} = \texttt{Update}(\delta\phi^t)$
```



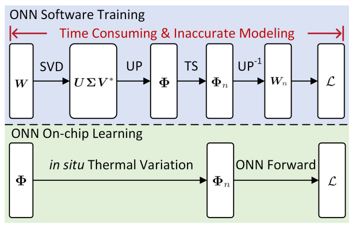
(a) (b)

Fig. 5: Thermal variation simulation for a 9×9 MZI triangular array based on Poisson's equation. (a) Initial heat source distribution; (b) steady normalized temperature distribution.

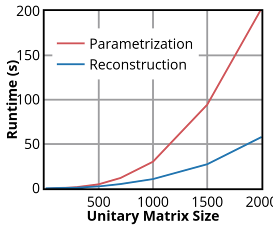
accurately model in an efficient way for several reasons. First, due to heat propagation, the temperature at any point on the chip is fully correlated with others, which can be ideally modeled with a Poisson's equation. Solving the steady temperature distribution of the whole chip will be time-consuming during training. Second, The heat source is not a single point but has a heat distribution along the physical dimensions of the device, which means different segments of the phase shifter will have different values of refractive index n under different temperatures. Hence the phase shift induced is given by the integral along the device dimensions. Third, the thermal impact from the phase shifters to neighboring waveguides is more complicated, as waveguides have different shapes, e.g., line, curves, circles. An accurate cross-talk model requires prohibitive computation that is rather challenging to consider as the chip scales up. In the present of thermal cross-talk $\mathcal{T}(\cdot)$, ONN on-chip training can be formulated as optimizing in the projected phase space,

$$\mathbf{\Phi}^* = \underset{\mathbf{\Phi}}{\operatorname{argmin}}\, \mathcal{L}(x; \mathcal{T}(\mathbf{\Phi})), \tag{13}$$

To resolve this robustness issue when optimizing Eq. (13), thermal variation can be modeled during ONN training. Back-propagation based software training may encounter severe efficiency and effectiveness issues, as shown in Fig. 6a. Modeling thermal variations during software training is time-consuming and inaccurate. It requires computationally-intensive SVD, unitary parametrization UP, and its inverse reconstruction UP^{-1} to switch between the weight matrix domain $\boldsymbol{W}$ and the corresponding phase domain $\mathbf{\Phi}$. To accurately obtain the projected phases $\mathbf{\Phi}_n = \mathcal{T}(\mathbf{\Phi})$, thermal variation with cross-talk simulation needs to be injected in each training iteration. This



(a)



(b)

Fig. 6: (a) Comparison between software training and on-chip learning. UP, TS, UP^{-1} represent unitary parametrization, thermal simulation, and inverse unitary reconstruction, respectively. (b) Runtime cost of unitary parametrization and inverse reconstruction.

whole procedure suffers from inefficiency and poor scalability given the high computational cost and runtime cost of unitary parametrization (Eq. (2)) and accurate thermal simulation. Our proposed on-chip learning method can inherently avoid those expensive domain transfer and inaccurate variation modeling via phase domain optimization with *in situ* thermal variation. Thus it can improve the ONN robustness with much higher learning efficiency than software training.

V. Experimental Results

To evaluate the effectiveness and efficiency of our proposed ONN on-chip learning algorithm, we compare inference accuracy and the number of ONN forward propagation with 1) brute-force phase tuning (BFT) [1], [11] algorithm, 2) particle swarm optimization (PSO) based on-chip training [13] algorithm, 3) our proposed algorithm with stochastic zeroth-order gradient estimation (FLOPS), and 4) our proposed algorithm with a second-stage sparse tuning (FLOPS+). Experiments are conducted on a Vowel Recognition dataset [20] to perform vowel phoneme classification. We implement all methods in PyTorch with an NVIDIA GTX 1080 GPU and an Intel Core i7-3770 CPU. We use two 3-layer ONN configurations in our experiments: 1) 8-16-16-4 and 2) 10-24-24-6, where 10 and 24 represent the input length and the number of neurons, respectively. We adopt a learning rate α=2 with exponential decaying rate of 0.985, a sampling standard deviation σ=0.002, and a mini-batch size $|\mathcal{S}|$=32, which is technically realizable by modern WDM techniques.

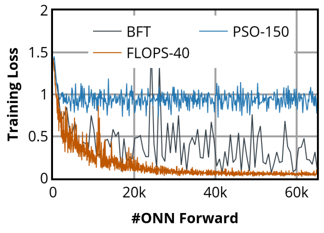
A. ONN Training Method Comparison

In the comparison experiments, the brute-force on-chip phase tuning method (BFT) [1], [11] sequentially perturbs $|\mathbf{\Phi}| = d$ phases with a decaying perturbation $\delta\phi$, compare the perturbed loss function with the original one, and update each phase according to,

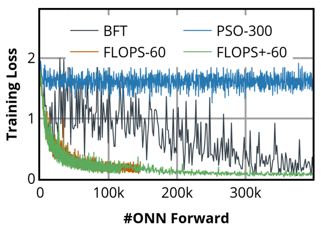
$$\phi_i \leftarrow \begin{cases} \phi_i + \delta\phi, & \text{if } \mathcal{L}(x, \phi_i + \delta\phi) < \mathcal{L}(x, \phi_i) \\ \phi_i - \delta\phi, & \text{if } \mathcal{L}(x, \phi_i + \delta\phi) \geq \mathcal{L}(x, \phi_i) \end{cases} \tag{14}$$

The particle swarm optimization (PSO) initializes a population of P phase solutions $\{\mathbf{\Phi}_p\}_{p=1}^{P}$, and iteratively updates the population towards a randomly mixed direction between their globally best-performing solution and individually best solution after P function evaluations. Note that as mentioned in Section I, the adjoint variable method (AVM) [12] is not compared because it requires expensive *in situ* light intensity measurement in the device-level such that it is technically intractable to realize on larger systems.

Based on the training curve comparison in Fig. 7, we notice a slow and unstable convergence for BFT method. We cut off the plot after certain function queries for clear demonstration, while BFT actually



(a)



(b)

Fig. 7: (a), (b) are training curve comparisons among different methods with ONN configurations of 8-16-16-4, and 10-24-24-6 respectively. *BFT* is trained for 50 epochs, and other methods are trained for 200 epochs.

takes an extremely long time to converge. For PSO-based ONN on-chip training [13], we adopt an experimentally optimal set of hyper-parameters for fair comparison, where initial velocity is within [-2,2], inertia weight w=0.5, individual cognitive constant c_1=0.5, and social constant c_2=1. PSO stagnates at a poor saddle point in an early stage, which is hard to overcome since only zeroth-order oracle information is used [19]. Our proposed algorithm (FLOPS) quickly explores the phase space along the estimated zeroth-order gradient directions towards a relatively low training loss with cheap function query complexity. Extended with sparse tuning procedure, shown in Fig. 7b, FLOPS+ takes longer to converge but effectively boosts the ONN learning performance, such that the accuracy gap is minimized compared to the best result.

Besides inference accuracy, we give theoretical analysis and practical measurements for the learning efficiency of the above four methods. BFT sequentially sweeps over all phases $\mathbf{\Phi} \in \mathbb{R}^d$, leading a query complexity of $\mathcal{O}(T\lambda d)$, where λ is the average number of function query at each tuning step. Assuming either case in Eq.(14) happens with the same probability, we estimate λ as 3/2. PSO method performs practically well on small-scale ONN training (<100 MZIs) [13], but a population-related complexity $\mathcal{O}(TP)$ makes it query-inefficient when optimizing a larger number of phases. FLOPS is sample-efficient with a query complexity of $\mathcal{O}(TQ)$, where Q is practically much lower than either P or λd. FLOPS+ offers a controllable approach to trade off between efficiency and performance. It costs more ONN forward as $\mathcal{O}((T - T_f)Q + T_f\lambda M)$, while better solution optimality can be obtained, and the training efficiency advantages still hold. To validate the potential and scalability of FLOPS, we estimate the runtime of on-chip training methods and BP-based software training with a 3-layer example ONN configuration of 200-500-500-10. As analyzed in Section. IV, BP-based software training suffers from computational inefficiency due to expensive domain transfer between $\boldsymbol{W}$ and $\mathbf{\Phi}$. The runtime breakdown for each software training iteration is estimated as,

$$\begin{aligned} t_{sw} &\approx t_{fp} + t_{svd} + t_{up} + t_{rec} + t_{bp} \\ &\approx 70ms + 200ms + 20s + 10s + 20ms \approx 30s \end{aligned} \tag{15}$$

where forward t_{fp}, backward t_{bp}, and SVD t_{svd} take around 300 ms. Unitary parametrization t_{up} and its inverse reconstruction t_{rec} takes the majority of the runtime. Figure. 6b shows that t_{up} and t_{rec} grow rapidly as the matrix size scales up. This runtime cost could be unaffordable once thermal simulation is added. For the same ONN configuration, the runtime estimation of on-chip training is [21],

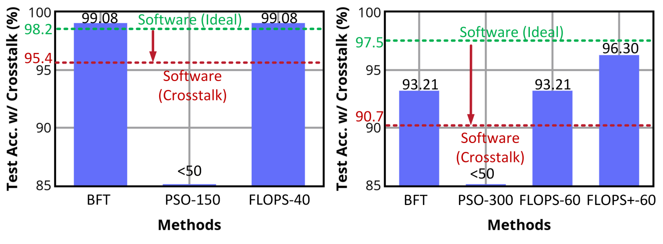
$$\begin{aligned} t_{oc} &\approx t_{prog} + t_{opt} + t_{pd} + t_{ad} + t_{iter} \\ &\approx 10\mu s + 1000ps + 20ps + 1ns + 1ms \approx 1ms, \end{aligned} \tag{16}$$

where t_{prog} is the thermal constant time for programming MZIs, t_{opt} is the propagation latency of optics through the 3-layer MZI

TABLE I: On-chip training methods comparison in terms of inference accuracy and number of ONN forward on a Vowel Recognition dataset.

ONN Setup	Methods	Test Accuracy	#ONN Forward
8-16-16-4($\|\mathbf{\Phi}\| = 448$)	BFT [1], [11]	99.08%	268.8 K (4.1)
	PSO-150 [13]	56.89%	256.0 K (3.9)
	FLOPS-40	99.08%	65.6 K (1.0)
10-24-24-6($\|\mathbf{\Phi}\| = 960$)	BFT [1], [11]	99.38%	864.0 K (3.9)
	PSO-300 [13]	43.83%	720.0 K (3.3)
	FLOPS-60	95.06%	219.6 K (1.0)
	FLOPS+-60	98.17%	405.1 K (1.8)

PSO-150 represents a population of 150; *FLOPS-40* sets Q to 40. *FLOPS+-40*, *FLOPS+-60* are extended with `SparseTune` with M=200 and 400 respectively. Normalized number of ONN forward is also shown for efficiency comparison.



(a) (b)

Fig. 8: (a), (b) are accuracy comparison under thermal cross-talk with ONN configurations of 8-16-16-4 and 10-24-24-6 respectively. The gap between *Software (Ideal)* and *Software (Crosstalk)* shows the accuracy drop caused by thermal cross-talk.

arrays, t_{pd} is the photo-detection time with WDM de-multiplexing, t_{ad} accounts for the analog-to-digital conversion time, and t_{iter} adds the computation overhead for gradient calculation and phase updates. Given the sampling factor $Q \ll \frac{30s}{1ms}$, our proposed ONN on-chip learning method potentially benefits from order of magnitude learning efficiency improvement over the traditional software-based training.

B. On-chip Training under in situ Thermal Variation

To demonstrate the robustness of our proposed learning method in the presence of device thermal variation, we evaluate the impact of thermal cross-talk among MZIs on the inference accuracy. Given that the phase shift is proportional to the temperature $\mathbf{\Delta\Phi} \propto \Delta T$ [21], [22], all phase shifts will increase since thermal cross-talk slows down the heat dissipation. We show the inference accuracy under thermal cross-talk for different methods in Fig. 8.

The pure software learning method achieves high accuracy under variation-free case ($\sim$98%), but degrades by $\sim$5% when the thermal variation is considered. The brute-force phase tuning (`BFT`) method demonstrates higher inference accuracy than pure software training but still suffers from inefficiency due to a considerable amount of function queries. Particle swarm optimization (`PSO`) generally shows unsatisfying robustness against thermal variation, leading to less than 50% accuracy for both ONN setups. Our proposed method `FLOPS` naturally considers the thermal non-ideality during on-chip learning and demonstrates better robustness and better function query efficiency than previous works. After fast exploration of `FLOPS`, an extra `SparseTune` is considered to trade off between efficiency and performance. At the cost of more function queries, our two-stage zeroth-order optimization method `FLOPS+` achieves the best accuracy and robustness under thermal cross-talk while still more efficient than previous methods.

VI. Conclusion

In this work, we propose a solution to enable efficient and robust ONN on-chip learning based on stochastic optimization with zeroth-order gradient estimation. A parallel mini-batch based asymmetric gradient estimator `FLOPS` is adopted to leverage the ultra-fast parallel photonic chips to improve training efficiency as well as learning performance. Extended with a light-weight sparse phase tuning `SparseTune`, a two-stage `FLOPS+` is introduced to further boost the accuracy under thermal variation while still maintaining better efficiency than previous works. We give a theoretical analysis of the variance bound of `FLOPS`, function query complexity, and runtime comparison with other methods. Experimental results on a Vowel Recognition dataset with two ONN setups are demonstrated. Compared with the brute-force method and PSO-based method, our proposed framework `FLOPS` provides a 3$\sim$4$\times$ more efficient on-chip learning protocol with better inference accuracy and robustness to thermal variation.

Acknowledgment

The authors acknowledge the Multidisciplinary University Research Initiative (MURI) program through the Air Force Office of Scientific Research (AFOSR), contract No. FA 9550-17-1-0071, monitored by Dr. Gernot S. Pomrenke.

References

[1] Y. Shen, N. C. Harris, S. Skirlo *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, 2017.
[2] Z. Zhao, D. Liu, M. Li *et al.*, "Hardware-software co-design of slimmed optical neural networks," in *Proc. ASPDAC*, 2019.
[3] J. Gu, Z. Zhao, C. Feng *et al.*, "Towards area-efficient optical neural networks: an FFT-based architecture," in *Proc. ASPDAC*, 2020.
[4] C. Feng, Z. Zhao, Z. Ying *et al.*, "Compact design of on-chip elman optical recurrent neural network," in *Proc. CLEO*, 2020.
[5] M. Reck, A. Zeilinger, H. Bernstein *et al.*, "Experimental realization of any discrete unitary operator," *Physical review letters*, 1994.
[6] A. Ribeiro, A. Ruocco, L. Vanacker *et al.*, "Demonstration of a 4×4-port universal linear circuit," *Optica*, 2016.
[7] L. Vivien, A. Polzer, D. Marris-Morini *et al.*, "Zero-bias 40gbit/s germanium waveguide photodetector on silicon," *Opt. Express*, 2012.
[8] Z. Zhao, J. Gu, Z. Ying *et al.*, "Design technology for scalable and robust photonic integrated circuits," in *Proc. ICCAD*, 2019.
[9] M. Milanizadeh, D. Aguiar, A. Melloni, and F. Morichetti, "Canceling thermal cross-talk effects in photonic integrated circuits," *J. Light. Technol.*, 2019.
[10] J. Gu, Z. Zhao, C. Feng *et al.*, "ROQ: A noise-aware quantization scheme towards robust optical neural networks with low-bit controls," in *Proc. DATE*, 2020.
[11] H. Zhou, Y. Zhao, X. Wang *et al.*, "Self-learning photonic signal processor with an optical neural network chip," *arXiv*, 2019.
[12] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica*, 2018.
[13] T. Zhang *et al.*, "Efficient training and design of photonic neural network through neuroevolution," *arXiv*, 2019.
[14] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," *SIOPT*, 2013.
[15] S. Liu, B. Kailkhura, P.-Y. Chen *et al.*, "Zeroth-order stochastic variance reduction for nonconvex optimization," in *Proc. NeurIPS*, 2018.
[16] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, 2017.
[17] C. Sun, M. T. Wade, Y. Lee, J. S *et al.*, "Single-chip microprocessor that communicates directly using light," *Nature*, 2015.
[18] C. Feng, Z. Ying, Z. Zhao *et al.*, "Wavelength-division-multiplexing-based electronic-photonic network for high-speed computing," in *Proc. SPIE, Smart Photonic and Optoelectronic Integrated Circuits XXII*, 2020.
[19] K. Balasubramanian and S. Ghadimi, "Zeroth-order nonconvex stochastic optimization: Handling constraints," *Arxiv*, 2019.
[20] D. Deterding, "Speaker normalisation for automatic speech recognition," PhD thesis, University of Cambridge, 1989.
[21] E. Timurdogan, Z. Su, C. V. Poulton *et al.*, "Aim process design kit (AIMPDKv2.0): Silicon photonics passive and active component libraries on a 300mm wafer," in *Optical Fiber Communication Conference*, 2018.
[22] N. C. Harris *et al.*, "Efficient, compact and low loss thermo-optic phase shifter in silicon," *Opt. Express*, 2014.