Efficient On-Chip Learning for Optical Neural Networks Through Power-Aware Sparse Zeroth-Order Optimization

Jiaqi Gu[1], **Chenghao Feng** [1], **Zheng Zhao** [2], **Zhoufeng Ying** [3], **Ray T. Chen** [1], **David Z. Pan** [1]
[1] University of Texas at Austin
[2] Synopsys, Inc. [3] Alpine Optoelectronics, Inc.
{jqgu, fengchenghao1996, zhengzhao, zfying}@utexas.edu, {chen, dpan}@ece.utexas.edu

Abstract

Optical neural networks (ONNs) have demonstrated record-breaking potential in high-performance neuromorphic computing due to its ultra-high execution speed and low energy consumption. However, current learning protocols fail to provide scalable and efficient solutions to photonic circuit optimization in practical applications. In this work, we propose a novel on-chip learning framework to release the full potential of ONNs for power-efficient *in situ* training. Instead of deploying implementation-costly back-propagation, we directly optimize the device configurations with computation budgets and power constraints. We are the first to model the ONN on-chip learning as a resource-constrained stochastic noisy zeroth-order optimization problem, and propose a novel mixed-training strategy with two-level sparsity and power-aware dynamic pruning to offer a scalable on-chip training solution in practical ONN deployment. Compared with previous methods, we are the first to optimize over 2,500 optical components on chip. We can achieve much better optimization stability, 3.7×-7.6× higher efficiency, and save >90% power under practical device variations and thermal crosstalk.

Introduction

As Moore's Law slows down, it becomes challenging for traditional electronics to further satisfy the escalating computational demands of machine learning tasks given clock frequency limitation and power density constraints. Recently, the emerging optical neural network (ONN) has attracted increasing attention due to its ultra-high execution speed and order-of-magnitude higher energy efficiency compared to electronics. In resource-constrained applications, ONNs become a promising alternative to accelerate machine learning workloads (Shen et al. 2017; Ramey et al. 2020; Ying et al. 2020a; Feng et al. 2020a; Ying et al. 2020b). Computationally-intensive operations in neural networks, e.g., matrix multiplication, can be finished within the light propagation delay in one shot (Shen et al. 2017; Zhao et al. 2019b; Gu et al. 2020c,d; Feng et al. 2020c; Miscuglio and Sorger 2020; Liu et al. 2019; Zokaee et al. 2020; Zhao et al. 2019a). With optical interconnects to reduce communication and memory transaction cost, a fully-optical neural engine provides a fundamental solution to break through the NN performance bound. Shen, *et al.* (Shen et al. 2017) demonstrated an integrated fully-optical neural chip to implement a multi-layer perceptron based on singular value decomposition (SVD) (Reck et al. 1994; Ribeiro et al. 2016). The weight matrices are mapped onto cascaded Mach-Zehnder interferometer (MZI) meshes to realize ultra-fast neural computing with over 100 GHz photo-detection rate and near-zero energy consumption (Shen et al. 2017; Vivien et al. 2012).

However, training methodologies for integrated ONNs still lack a scalable and efficient solution so far. The mainstream approach offloads the training and simulation process to electrical computers using classical back-propagation (BP) (Shen et al. 2017; Zhao et al. 2019b), which is inefficient in circuit simulation and inaccurate in device noise modeling. Hence, there exist great potentials to offload the learning process on photonic circuits. Back-propagation is technically challenging to be implemented on a chip given the expensive hardware overhead and time-consuming gradient computation.

A brute-force phase tuning algorithm is proposed and adopted in (Shen et al. 2017; Zhou et al. 2019) to perform ONN on-chip training via sequential device tuning, which is intractable as circuits scale up. To mitigate the inefficiency issue of the above brute-force algorithm, an *in situ* adjoint variable method (AVM) (Hughes et al. 2018) is applied to directly compute the gradient w.r.t. MZI phases via inverse design. However, it is challenging to be scaled to larger systems as the fully-observable circuits is a technically impractical assumption. Evolutionary algorithms, e.g., genetic algorithm (GA) and particle swarm optimization (PSO), are introduced to train ONNs by population evolution (Zhang et al. 2019). A stochastic zeroth-order optimization framework `FLOPS` (Gu et al. 2020a) has been proposed to improve the ONN learning efficiency by 3-5 times via random-sampling-based zeroth-order gradient estimation.

However, previous works have the following disadvantages: 1) nontrivial Gaussian sampling cost, 2) divergence issues due to high variance, 3) high energy consumption, and 4) hardware-unfriendly weight update step size. In this work, we propose a novel mixed-training framework that enables scalable on-chip optimization with more stable convergence, higher training efficiency, and much lower power consumption under non-ideal environment. Compared with previous state-of-the-art (SOTA) methods, our mixed-training framework has the following advantages.

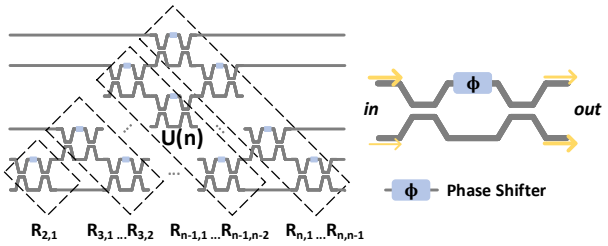


Figure 1: MZI triangular array $\boldsymbol{U}(n)$ and the MZI structure.

- Efficiency: our mixed-training strategy achieves 3∼7× fewer ONN forward and much lower computation complexity than SOTA ONN on-chip learning methods.
- Robustness: our method adopts a novel optical device mapping with mixed-active/passive regions to protect ONNs from device variations and thermal crosstalk, leading to better noise-tolerance than previous solutions.
- Stability: our stochastic zeroth-order sparse coordinate descent optimizer (`SZO-SCD`) outperforms SOTA zeroth-order optimizers with more stable convergence and better performance in on-chip accuracy recovery.
- Scalability: our proposed optimizer leverages two-level sparsity in on-chip training, extending the ONN learning scale to >2,500 MZIs.
- Power: we propose a lightweight power-aware dynamic pruning technique, achieving >90% lower power consumption with near-zero accuracy loss or computation overhead.

Preliminary

In this section, we will introduce the architecture of integrated ONNs, prior work in ONN on-chip training, and background knowledge about stochastic zeroth-order optimization.

ONN Architecture and Training Methods

The integrated optical neural network (ONN) is a hardware platform that implements artificial neural networks with silicon-photonics. As a case study, we focus on an ONN architecture based on singular value decomposition (SVD) (Shen et al. 2017). It decomposes an $m \times n$ weight matrix using SVD, i.e., $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. The diagonal matrix $\boldsymbol{\Sigma}$ can be simply implemented by on-chip attenuators, e.g., single-port Mach-Zehnder interferometers (MZIs), to perform signal scaling. The unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}^*$ can be realized by a cascaded MZI triangular array (Reck et al. 1994), shown in Fig. 1. A Reck-style (Reck et al. 1994) unitary group reconstruction is given by,

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

where $\boldsymbol{D}$ is a diagonal matrix with ± 1 on its diagonal entries, and the 2-dimensional planar rotator $\boldsymbol{R}_{ij}(\phi_{ij})$ is an n-dimensional identity matrix where entries on (i,i), (i,j), (j,i), (j,i) are $\cos\phi_{ij}$, $\sin\phi_{ij}$, $-\sin\phi_{ij}$, $\cos\phi_{ij}$, respectively. Each rotator $\boldsymbol{R}_{ij}$ can be implemented by a 2×2 MZI that produces unitary interference of input light signals with a phase shift ϕ in its inner arm as follows (Shen et al. 2017),

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{2}$$

To train ONNs, the traditional procedure trains the weight matrix $\boldsymbol{W}$ using gradient back-propagation and then maps it to photonic circuits through SVD and unitary group parametrization (Reck et al. 1994), which is inefficient and hardware-agnostic. Later, several ONN on-chip learning protocols are proposed to perform *in situ* circuit optimization. To solve the problem, a straightforward approach is to compute the gradient w.r.t each MZI configuration given by,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \phi_{ij}} &= \sum \left((\nabla_y \mathcal{L}\, x^T \boldsymbol{\Sigma} \boldsymbol{V}^*) \odot \frac{\partial \boldsymbol{U}}{\partial \phi_{ij}} \right) \\ \frac{\partial \boldsymbol{U}}{\partial \phi_{ij}} &= \boldsymbol{D}\boldsymbol{R}_{n1}\boldsymbol{R}_{n2}\boldsymbol{R}_{n3} \cdots \frac{\partial \boldsymbol{R}_{ij}}{\partial \phi_{ij}} \cdots \boldsymbol{R}_{31}\boldsymbol{R}_{32}\boldsymbol{R}_{21}. \end{aligned} \tag{3}$$

On edge computing platforms, this analytical Jacobian is computationally-prohibitive and noisy, which is intractable in practical deployment. Later, a brute-force phase tuning method is proposed (Shen et al. 2017; Zhou et al. 2019) using finite-difference-based gradient estimation. Adjoint variable method (AVM) (Hughes et al. 2018) is proposed to model the circuit state as a partial-differential-equation-controlled linear system, and directly measures the exact gradient via *in situ* light intensity measurement. Evolutionary algorithms, e.g., particle swarm optimization and genetic algorithm, are demonstrated to train MZIs on chip (Zhang et al. 2019). A stochastic zeroth-order gradient descent based method `FLOPS` (Gu et al. 2020a) has been proposed to improve the training efficiency by 3-5× compared with previous methods.

Stochastic Zeroth-Order Optimization

To solve optimization problems when analytical Jacobian is infeasible to compute, zeroth-order optimization (ZOO) plays an significant role, e.g., black-box adversarial attacks, policy-gradient-based reinforcement learning, and circuit parameter optimization (Chen et al. 2017, 2019; Ghadimi and Lan 2013; Liu et al. 2018; Gorbunov et al. 2020; Zhao et al. 2020; Tu et al. 2019; Wang, Qian, and Yu 2018). Various ZOO methods have been proposed with mathematically-proven convergence rate, including stochastic gradient descent with Nesterov's acceleration (Nesterov and Spokoiny 2017), zeroth-order coordinate-wise Adam `ZOO-ADAM` and Newton's method `ZOO-Newton` (Chen et al. 2017), zeroth-order adaptive momentum method `ZOO-AdaMM` (Chen et al. 2017), stochastic three-points (Bibi et al. 2020), stochastic momentum three-points (Gorbunov et al. 2020), etc. Most zeroth-order optimizers have a convergence rate dependent on the dimensionality, which intrinsically makes them less efficient and less scalable than higher-order optimizers. In this work, we explore two-level sparsity in stochastic zeroth-order optimization to enable scalable, stable, and efficient ONN on-chip training.

Problem Formulation and Analysis

Before discussing our proposed on-chip learning framework, we give a formulation to the resource-limited ONN learning problem. In practical ONN applications, the ultimate target is

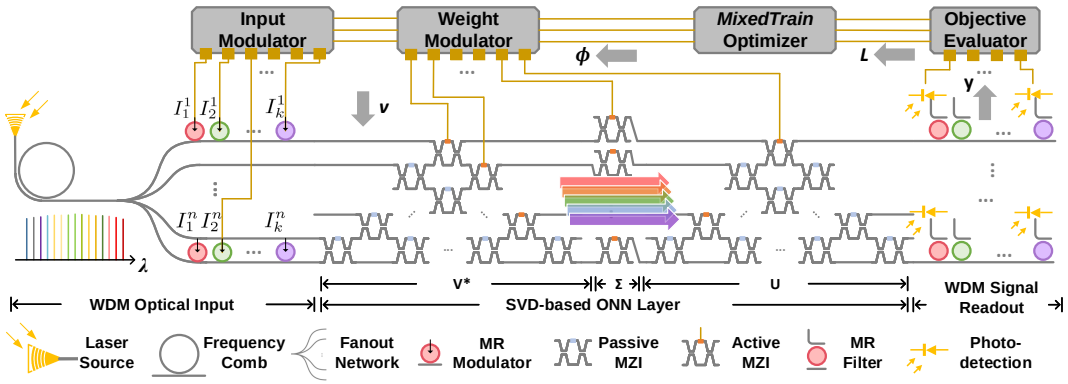


Figure 2: Schematic of ONN on-chip learning framework with stochastic zeroth-order mixed-training.

to leverage photonic neural chip to complete machine learning tasks with high accuracy, low latency, and low energy consumption, under environmental changes and device-level variations. The optimization variables are optical device configurations, i.e., phase shift for all MZIs $\boldsymbol{\Phi}$, including those in the unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}^*$ and the diagonal matrix $\boldsymbol{\Sigma}$. The objective is the task-specific loss function. We are the first to formulate the ONN on-chip learning as a resource-limited accuracy recovery problem in the unitary space,

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}\sim\mathcal{R}} \mathcal{L}_{\mathcal{D}^{trn}}(\boldsymbol{W}(\boldsymbol{\Phi})), \tag{4}$$

$$\text{s.t. } \boldsymbol{W}(\boldsymbol{\Phi}) = \boldsymbol{U}(\boldsymbol{\Phi}^U)\boldsymbol{\Sigma}(\boldsymbol{\Phi}^{\boldsymbol{S}})\boldsymbol{V}^*(\boldsymbol{\Phi}^V),$$

$$\boldsymbol{U}(\boldsymbol{\Phi}^U) = \boldsymbol{D}^U \prod_{i=N}^{2}\prod_{j=1}^{i-1} \boldsymbol{R}_{ij}(\phi_{ij}^U),$$

$$\boldsymbol{V}^*(\boldsymbol{\Phi}^V) = \boldsymbol{D}^V \prod_{i=M}^{2}\prod_{j=1}^{i-1} \boldsymbol{R}_{ij}(\phi_{ij}^V),$$

$$\|\boldsymbol{\Sigma}(\boldsymbol{\Phi}^S)\|_\infty < m,$$

$$\boldsymbol{\Phi} \in [0, 2\pi),$$

$$\texttt{Power}(\boldsymbol{\Phi}) \le \tilde{P}, \int_t \texttt{Power}(\boldsymbol{\Phi}^t)\,\mathrm{d}t \le \tilde{E},$$

$$\mathcal{C}(\nabla_{\boldsymbol{\Phi}}\mathcal{L}) \gg \tilde{C} \gg \mathcal{C}(\mathcal{L}), \mathcal{C} \le \tilde{C},$$

where $\mathcal{D}^{trn}$ is the training set. In each layer, the weight matrix $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ is constructed by $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{V}^*$, where $\boldsymbol{U}$ and $\boldsymbol{V}^*$ are constrained in the Stiefel manifold, and the ℓ_∞-norm of the diagonal matrix $\boldsymbol{\Sigma}$ is bounded by a empirically largest signal scaling range m. The optimization parameters $\boldsymbol{\Phi}$ are constrained in a hypercube within 0 degree and 2π degree. The photonic device programming power has to honor a maximum power budget $\tilde{P}$ during ONN inference. Also, the total energy used to program ONN devices during on-chip training is bounded by an energy budget $\tilde{E}$. The last constraint is the computation budget for the optimizer, which can not afford to calculate the Jacobian $\nabla_{\boldsymbol{\Phi}}(\mathcal{L})$ shown in Eq. (3), but the objective evaluation is ultra-fast with optics.

To solve the optimization problem on this SVD-based architecture, we directly optimize the decomposed matrices $\boldsymbol{U}$ and $\boldsymbol{V}^*$ within the Stiefel manifold. Previous work proposed Riemannian optimization (Huang et al. 2018), unitary regularization (Zhao et al. 2019b), and unitary projection (Gu et al. 2020b) to satisfy the unitary constraints. In this work, we optimize the phases $\boldsymbol{\Phi}$ in the Reck-style unitary parametrization space to achieve minimum computation complexity. For the diagonal matrix, we optimize it as $\texttt{diag}(\boldsymbol{\Sigma}) = m(\cos\phi_0^S, \cdots, \cos\phi_{\min(M,N)-1}^S)$, such that the optimization variables can be unified with $\boldsymbol{\Phi}^U$ and $\boldsymbol{\Phi}^V$. To facilitate power optimization, we do not use the periodic phase space relaxation (Gu et al. 2020a). Instead, we wrap the phase within the valid hypercube by $\phi = (\phi \mod 2\pi)$ at each iteration avoid unnecessary power once the updated phase exceeds 2π. To meet the computation budget, we will introduce a lightweight technique to handle power and energy constraints. The computation budget of the resource-constrained platform can be satisfied by manual searching in the optimizer design space, where lightweight zeroth-order optimization methods will be promising candidates.

Proposed ONN On-Chip Learning Framework

In the practical ONN deployment, apart from the basic constraints listed in Eq. (4), non-ideal environment and device noises are necessary to be considered in the learning framework. Therefore, we propose a mixed-training strategy to efficiently solve this noisy learning problem with all the aforementioned constraints in Fig. 3.

Scalable Mixed-Training Strategy

To enable efficient ONN on-chip learning on practical networks and dataset, we propose a mixed-training strategy to reduce the optimization dimensionality and minimize tunable devices for better convergence and lower power consumption. Specifically, we assume a model is pre-trained and prepared for edge ONN deployment. Then, our target is to implement the pre-trained model on practical ONN engines while recovering the accuracy given non-ideal environment and device variations. The naive solution is to deploy a fully active ONN where all optical devices are thermo-optically tunable with maximum learnability (Gu et al. 2020a). However, This leads to high control complexity, power consumption, and non-ideal thermal crosstalk. In this work, we propose a mixed-training strategy that integrates passive and active ONNs to balance efficiency, robustness, and learnability, shown in

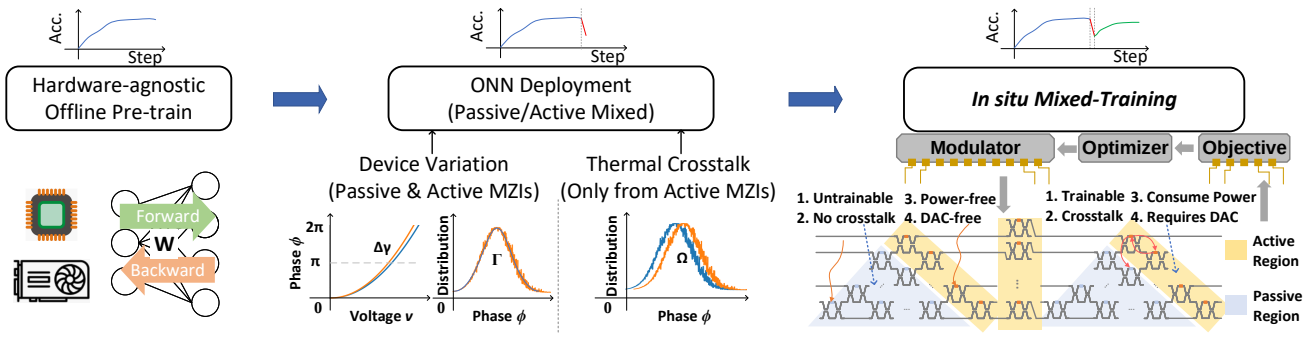


Figure 3: Mixed-training flow in the practical ONN deployment.

Fig. 3. Now we introduce three stages in the entire ONN on-chip mixed training flow.

Hardware-unaware Pre-training For the MZI-based ONN architecture, Hardware-unaware training based on back-propagation is firstly performed with an ideal computational model on electrical digital platforms, e.g., GPUs and CPUs, to obtain target device configurations.

ONN Deployment with Mixed Active/Passive Regions We proposed to deploy the ideally-trained model on the photonic circuits using a mixed passive/active design, where most parameters in two unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}^*$ are fixed by using passive optical devices. Only the diagonal matrix $\boldsymbol{\Sigma}$ and a small fraction ($\alpha \ll 1$) of phases in two unitary matrices are implemented by active devices to enable its adaptability and learnability. We denote fixed phases in the passive region as $\mathcal{P}$ and tunable phases in the active region as $\mathcal{A}$.

In practical applications, device variations, e.g., phase shifter γ coefficient drift, and thermal crosstalk among MZIs will be present, leading to output perturbation and thus accuracy loss, shown in Fig. 3. The phase shifter variations come from environmental temperature changes or manufacturing errors. Under variations, the power-to-phase-shift factor γ of both active and passive phase shifters will drift from the ideal value as $\gamma^v = \gamma + \Delta\gamma$, where we assume the noise is sampled from a truncated Gaussian distribution $\Delta\gamma \in \mathcal{N}(0, \sigma_\gamma^2)$. Given that the phase shift is proportional to the device-related coefficient as $\phi \propto \gamma$, the noisy phase shift is denoted as $\phi^v = \phi\gamma^v/\gamma$. For N phase shifters, this variation is described as a diagonal perturbation matrix $\boldsymbol{\Phi^v} = \boldsymbol{\Gamma\Phi}$. In terms of thermal crosstalk, the correlated heat distribution among thermo-optic devices leads to an increase in the steady temperature. In the heat steady state, the mutual correlation of phases within N noisy phase shifters $\boldsymbol{\Phi}^v$ can be described by a coupling matrix as $\boldsymbol{\Phi}^c = \boldsymbol{\Omega\Phi^v}$,

$$\begin{pmatrix} \phi_0^c \\ \phi_1^c \\ \vdots \\ \phi_{N-1}^c \end{pmatrix} = \begin{pmatrix} \omega_{0,0} & \omega_{0,1} & \cdots & \omega_{0,N-1} \\ \omega_{1,0} & \omega_{1,1} & \cdots & \omega_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{N-1,0} & \omega_{N-1,1} & \cdots & \omega_{N-1,N-1} \end{pmatrix} \begin{pmatrix} \phi_0^v \\ \phi_1^v \\ \vdots \\ \phi_{N-1}^v \end{pmatrix}$$
$$\text{s.t. } \omega_{i,j} = 1, \quad \forall\, i = j$$
$$\omega_{i,j} = 0, \quad \forall\, i \neq j \text{ and } \phi_j \in \mathcal{P} \tag{5}$$
$$0 \leq \omega_{i,j} < 1, \quad \forall\, i \neq j \text{ and } \phi_j \in \mathcal{A}.$$

The diagonal factor $\omega_{i,j}, i = j$ is the self-coupling coefficient, which is typically set to 1. $\omega_{i,j}, i \neq j$ is the mutual coupling coefficient (Milanizadeh et al. 2019). As a physical fact, only active devices are thermal aggressors that perturb adjacent devices, while passive devices do not impact their neighbors since they have zero heat dissipation. Hence mutual coupling factors $\omega_{i,j}, i \neq j$ are set to 0 if ϕ_j represents a passive phase shifter. We can unify the γ noise with the crosstalk as $\boldsymbol{\Phi^c} = \boldsymbol{\Omega\Gamma\Phi}$. Therefore, the objective is re-written as,

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}\sim\mathcal{R}} \mathcal{L}_{\mathcal{D}^{trn}}(\boldsymbol{W}(\boldsymbol{\Omega\Gamma\Phi})). \tag{6}$$

Mixed-training with Stochastic Zeroth-Order Sparse Coordinate Descent (SZO-SCD) In this stage, we introduce stochastic zeroth-order sparse coordinate descent (SZO-SCD) to tune the active devices for *in situ* accuracy recovery. Since the pre-trained model is roughly converged, the ZO-gradient-based method (Gu et al. 2020a) will suffer from divergence issues due to its gradient estimation variance. In contrast, our SZO-SCD optimizer is more suitable for near-convergence fine-tuning in the phase space. In iteration t, only a fraction ($s \ll 1$) of active devices $\boldsymbol{\Phi}_s = \{\phi_0, \cdots, \phi_{s|\mathcal{A}|-1}\} \subseteq \mathcal{A}$ are selected for coordinate descent as follows,

$$\phi_i^{t+1} \leftarrow \arg\min_{\phi_i} \{\mathcal{L}_{\mathcal{I}^t}(\phi_i^t + \delta\phi), \mathcal{L}_{\mathcal{I}^t}(\phi_i^t - \delta\phi)\}. \tag{7}$$

The mini-batch evaluation of $\mathcal{L}_{\mathcal{I}}(\cdot)$ can be processed in parallel by using WDM (Tan, Grieco, and Fainman 2014; Feng et al. 2020b) shown in Fig. 2.

The advantages of the mixed-training strategy with SZO-SCD lie in several aspects. First, since the method reduces the tunable parameters of an $N \times N$ weight matrix from N^2 to $N + s\alpha N^2$ per iteration, the optimization efficiency will be considerably improved. Second, the passive ONN part consumes nearly zero energy, leading to approximately (1-α) power saving. Third, the optimization dimensionality is reduced from the full $\mathcal{O}(N^2)$ space to a sparse subspace, which accelerates the convergence of our zeroth-order learning algorithm with a slimmed computation demand. Fourth, this method has $\mathcal{O}(1)$ computation complexity and $\mathcal{O}(1)$ memory complexity per iteration, which is nearly the cheapest optimizer in the design space.

Power-aware Dynamic Pruning

On resource-limited edge applications, low power consumption will be a preferable feature to enhance endurance. Since

Algorithm 1 SZO−SCD with Power-aware Dynamic Pruning

Require: ONN forward function $\mathcal{L}(\cdot)$, phases $\mathbf{\Phi}^0$ after ONN deployment, training dataset $\mathcal{D}^{trn}$, total iterations T, Active set $\mathcal{A}$, sparsity of fine-tuned phases s, initial tuning step size $\delta\phi^0 > 0$, and power awareness $p \in [0, 1]$, power estimator $\texttt{power}(\cdot)$;

Ensure: Converged phases $\mathbf{\Phi}^{T-1}$;

1: **for** $t \leftarrow 0 \cdots T-1$ **do**
2: Randomly sample a mini-batch $\mathcal{I}^t$ from $\mathcal{D}^{trn}$
3: Randomly select $\mathbf{\Phi}_s^t = \{\phi_0^t, \cdots, \phi_{s|\mathcal{A}|-1}^t\} \subseteq \mathcal{A}$ without replacement
4: **for** $\phi_i^t \leftarrow \phi_0^t, \cdots, \phi_{s|\mathcal{A}|-1}^t$ **do**
5: **if** $\mathcal{L}_{\mathcal{I}^t}(\phi_i^t + \delta\phi^t) < \mathcal{L}_{\mathcal{I}^t}(\phi_i^t)$ **then**
6: $\phi_i^{t+1} \leftarrow \phi_i^t + \delta\phi^t$
7: **else**
8: **if** $\texttt{power}(\phi_i^t - \delta\phi^t) > \texttt{power}(\phi_i^t)$ **then**
9: $b \sim \mathcal{B}(p)$ ▷ Sample from Bernoulli distribution with probability p to take 1
10: $\phi_i^{t+1} \leftarrow \phi_i^t - b \cdot \delta\phi^t$
11: **else**
12: $\phi_i^{t+1} \leftarrow \phi_i^t - \delta\phi^t$
13: $\delta\phi^{t+1} = \texttt{Update}(\delta\phi^t)$ ▷ Step size decay

the power of active phase shifters is proportional to the phase shift $P \propto \phi$, we use the phase shift $\phi \in [0, 2\pi)$ as a fast device tuning power estimator. A straightforward approach to handle this power constraint is to use Lagrangian relaxation to add the power constraint in the objective as follows,

$$\begin{aligned} \mathbf{\Phi}^* &= \arg\min_{\mathbf{\Phi}\sim\mathcal{R}} \mathcal{L}_{\mathcal{D}^{trn}}(\boldsymbol{W}(\mathbf{\Omega\Gamma\Phi})) + \lambda P(\mathbf{\Phi}) \\ P(\mathbf{\Phi}) &= \sum_{\phi\in\mathbf{\Phi}} (\phi \mod 2\pi), \end{aligned} \tag{8}$$

and solve it using alternating direction multiplier method (ADMM). However, the dual update for power optimization will cause convergence issues, which will be shown in our later experiment. To implicitly consider power constraints in the optimization, we propose a power-aware dynamic pruning technique to further boost the power efficiency with stable convergence. The detailed power-aware optimization algorithm is described in Alg. 1. In lines 8-12, the optimizer will prune backward steps with probability p if the objective increases in the positive direction and the power consumption increases in the negative direction. The intuition behind this efficient power-aware pruning is that our SZO−SCD only queries the zeroth-order oracle, such that the step-back is not guaranteed to be a descent direction. This uncertainty enables us to embed a power-constraint handling mechanism to dynamically prune step-backs that do not have a descent guarantee but lead to a certain power increase. The probabilistic power-awareness factor p also provides a parametric approach to balance between power and solution quality.

Experimental Results

Experiments are conducted on a vowel classification dataset (Deterding 1989), MNIST (LeCun 1998), FashionMNIST (Xiao, Rasul, and Vollgraf 2017), and CIFAR-10 (Krizhevsky, Hinton et al. 2009) for image classification. As a proof-of-concept demonstration, those datasets are standard and practical for ONNs. We implement all methods in PyTorch with an NVIDIA Quadro RTX 6000 GPU and an Intel Core i7-9700 CPU. We adopt a step size $\delta\phi$=0.02 with an epoch-wise exponential decaying rate of 0.985 and a mini-batch size of 32. The upper bound m set for $\mathbf{\Sigma}$ is 3. Following a common setting, the std. of phase variation $\sigma(\gamma)$ is set to 2e-3 and the mutual-coupling factor ω is set to 2e-3 only for adjacent MZIs. Rectified linear units (ReLU) (He et al. 2015) with an upper limit 4 are used as the nonlinearity.

Evaluation on Mixed-Training Strategy

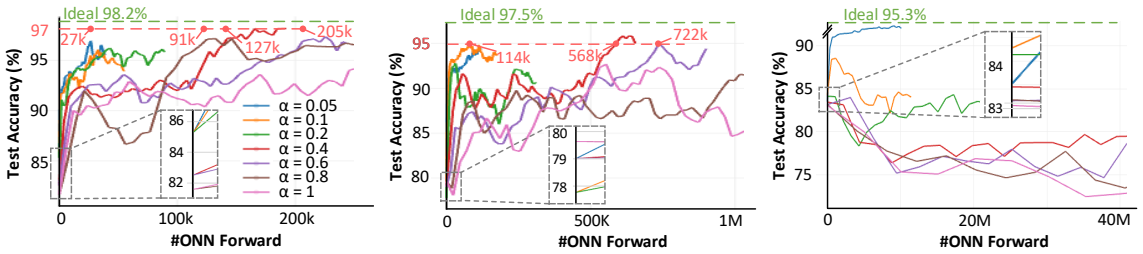
Figure 4 demonstrate the inference accuracy curve after on-chip deployment using our mixed-training strategy. Active phases are randomly selected from all phases. With $\Delta\gamma \in \mathcal{N}(0, 0.002)$ phase shifter variations and ω=2e-3 thermal crosstalk between adjacent devices, the initial accuracy varies among different mixed-training sparsity values. A larger mixed-training sparsity can provide protection to the PIC from thermal crosstalk since more passive devices generate fewer thermal noises. The convergence of those curves shows that only a small fraction (5%-15%) of devices is necessary to perform on-chip learning for accuracy recovery, while simply tuning every parameter has the lowest efficiency and effectiveness among all settings. When α is set to 5%-15%, we observe the fastest convergence speed, leading to 3.7×-7.6× higher training efficiency, i.e., fewer function queries, than ours with a large α. This mixed-training strategy makes it possible to recover the accuracy of larger-scale ONNs with much fewer function queries and lower power.

Evaluation on the Sparsity of SZO−SCD

Figure 5 demonstrates how different sparsity s influences the on-chip learning performance under a given mixed-training sparsity $\alpha = 0.15$. Among all sparsity values, 60% is the best tuning sparsity that leads to the fastest convergence speed on small datasets. In contrast, on MNIST, since the dataset variance is much larger than Vowel Recognition, a highly-sparse learning strategy (s >0.9) is more suitable to balance the variance and generalization in the stochastic optimization. In other words, overly-greedy optimization caused by small s values is harmful to stochastic learning. Note that a higher sparsity, e.g., s >0.98, will lead to accuracy loss since the variance is too large for the optimizer to converge, which is not shown on the figure for brevity.

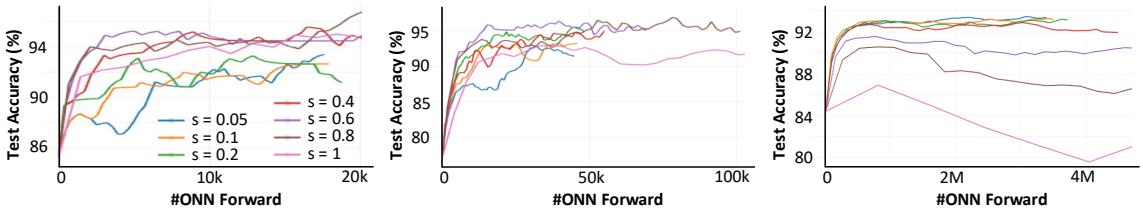
Compare with Other Zeroth-Order Optimizers

To validate the efficiency of our proposed SZO−SCD, we compare a variety of state-of-the-art ZO optimizers on different sparsity in Table. 1. Proper α and s are adopted to obtain a good trade-off between accuracy and efficiency. Learning rates reported are empirically most suitable values with equal parameter searching efforts for all methods. The comparison results provide several important insights. First, in high-dimensional ONN parameter space, the gradient-based methods, e.g., FLOPS, generally show poor performance and unstable convergence due to gradient estimation variance.



(a) (b) (c)

Figure 4: Test accuracy with different mixed-training sparsity α. (a) MLP (8-16-16-4) on Vowel Recognition, (b) MLP (10-24-24-6) on Vowel Recognition, and (c) MLP (8×8-24-24-10) on MNIST. Close-up views show the accuracy after deployment.



(a) (b) (c)

Figure 5: Evaluation with different sparsity s in SZO−SCD. α is set to 0.15 for all models. (a) 8-16-16-4 on Vowel Recognition dataset. (b) 10-24-24-6 on Vowel Recognition dataset. (c) (8×8)-24-24-10 on MNIST dataset.

Even for ZOO−ADAM and ZOO−Newton that adaptively adjust the step size, they suffer from divergence in the phase-domain optimization unless a descent in the objective can be partially guaranteed like our proposed coordinate descent method. Second, the two-level sparsity is indeed necessary to achieve stable convergence and good model generalization. FLOPS with two-level sparsity coincides with (Ohta et al. 2020) and shows better convergence than the dense counterpart. Third, gradient-based methods require an arbitrarily tiny step size (<1e-3) for gradient estimation and weight updating, which is not practical given limited device control resolution. In contrast, our method only needs a medium step size, corresponding to 8-bit control precision, a more hardware-friendly configuration in analog neuromorphic computing. Fourth, interestingly, the stochastic coordinate-wise three-points method (STP) leads to worse inference accuracy than our method due to its overly-greedy updating mechanism that potentially harms generalization as follows,

$$\phi_i^t \leftarrow \arg\min_{\phi_i} \{\mathcal{L}(\phi_i^{t-1}), \mathcal{L}(\phi_i^{t-1} + \delta\phi), \mathcal{L}(\phi_i^{t-1} - \delta\phi)\}. \quad (9)$$

Table. 2 further shows the average performance on different datasets with CNNs. Overall, our proposed mixed-training strategy with SZO−SCD shows the best convergence and accuracy with the smallest computation and memory cost.

Evaluation on the Power-Aware Dynamic Pruning

We evaluate the effectiveness of our proposed power-aware dynamic pruning technique in Fig. 6. Different power awareness values lead to slightly different inference accuracy after convergence. However, fully-power-aware (p=1) pruning can cut down 30%-50% power compared with the power-unaware version (p=0). Compared with the naive ONN deployment without mixed-training, our power-aware mixed-training can save a total ∼90% power. This lightweight pruning method not only reduces the power in inference $\mathcal{P}$, shown in the final power value at the end of the curve, it also saves the training energy $\int_t \mathcal{P}\ \mathrm{d}t$ indicated by the area under the power curve. We also compare with ADMM to show the superiority of our dynamic pruning technique in Table 3. The Lagrangian-relaxation-based formulation and ADMM-based optimization algorithm are not suitable for power-aware ONN on-chip learning. A small λ in the dual update step has a trivial influence on the total power, while a large λ leads to unstable convergence. In contrast, our proposed method can provide stable power constraint handling with a parametric mechanism to achieve a trade-off between accuracy and power.

Evaluation on CNNs and Different Datasets

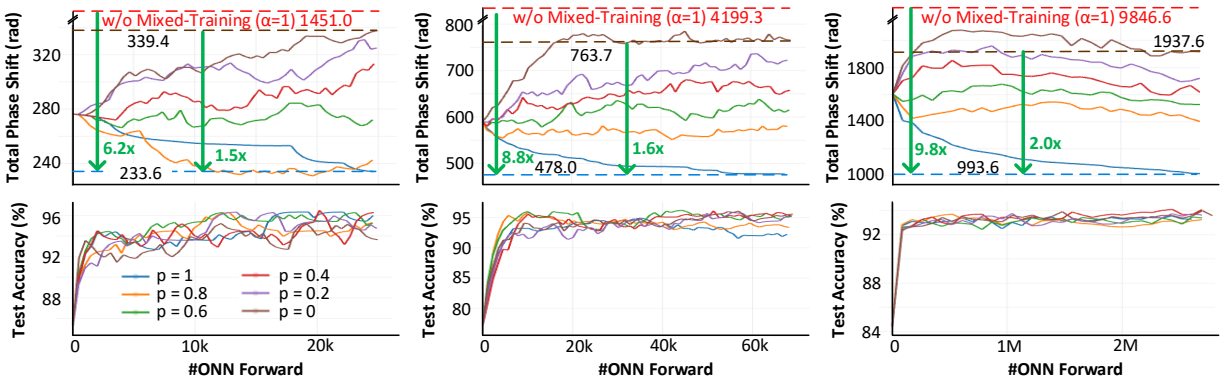
We further evaluate the effectiveness of our proposed mixed-training strategy with sparse tuning on convolutional neural networks (CNNs). We use *im2col* algorithm to implement convolution with general matrix multiplication (GEMM). Table 4 shows the accuracy recovery results and power improvement on three different datasets compared with w/o mixed-training or power handling. On three practical datasets, our proposed methods demonstrate stable accuracy recovery for optical CNN architectures under device variations while reducing the total inference power by >95%.

Conclusion

In this work, we propose a scalable ONN on-chip learning framework to efficiently perform *in situ* accuracy recovery with dynamic power optimization. We are the first to formulate ONN on-chip learning problem with device non-ideality

Table 1: Comparison with SOTA ZO optimizers in terms of optimizer cost per iteration, ONN query complexity per iteration, and memory complexity. *lr* is the step size. We evaluate on MNIST with a 3-layer optical MLP (64-24-24-10). T is the total iteration. d is the total number of variables (d=2,350). The sampling factor Q is set to 60 as used in FLOPS (Gu et al. 2020a).

Optimizer	α	s	lr	Computation	#ONN forward	Memory	Best Acc.
ZOO-ADAM (Chen et al. 2017)	1	1	1e-3	$\mathcal{O}(d)$	$2Td$ $(4700T)$	$\mathcal{O}(d)$	diverge
ZOO-ADAM (Chen et al. 2017)	0.15	0.1	1e-3	$\boldsymbol{\mathcal{O}(\alpha sd)}$	$2T\alpha sd$ $(70.5T)$	$\mathcal{O}(\alpha d)$	88.1%
ZOO-Newton (Chen et al. 2017)	1	1	1e-3	$\mathcal{O}(d)$	$3Td$ $(7050T)$	$\boldsymbol{\mathcal{O}(1)}$	diverge
ZOO-Newton (Chen et al. 2017)	0.15	0.1	1e-3	$\boldsymbol{\mathcal{O}(\alpha sd)}$	$3T\alpha sd$ $(105.75T)$	$\boldsymbol{\mathcal{O}(1)}$	diverge
STP (Bibi et al. 2020)	1	1	2e-2	$\mathcal{O}(d)$	$2Td$ $(4700T)$	$\boldsymbol{\mathcal{O}(1)}$	diverge
STP (Bibi et al. 2020)	0.15	0.1	2e-2	$\boldsymbol{\mathcal{O}(\alpha sd)}$	$2T\alpha sd$ $(70.5T)$	$\boldsymbol{\mathcal{O}(1)}$	90.2%
FLOPS (Gu et al. 2020a)	1	1	1e-1	$\mathcal{O}(Qd)$	TQ $(60T)$	$\mathcal{O}(d)$	diverge
FLOPS (Gu et al. 2020a)	0.15	0.1	1e-1	$\mathcal{O}(Qd)$	TQ $(60T)$	$\mathcal{O}(\alpha sd)$	83.5%
SZO-SCD (Proposed)	0.15	0.1	2e-2	$\boldsymbol{\mathcal{O}(\alpha sd)}$	$\boldsymbol{1.5T\alpha sd}$ $\boldsymbol{(52.88T)}$	$\boldsymbol{\mathcal{O}(1)}$	**93.5%**



(a) (b) (c)

Figure 6: Estimated power and inference accuracy evaluation with different power awareness p. The mixed-training sparsity α is selected as 0.15. The sparsity s for SZO-SCD is set to 0.6 for (a) and (b), and is set to 0.1 for (c).

Table 2: Average accuracy(std.) among different optimizers over 3 runs. The CNN setting is 16×16-c8s2-c6s2-10 for MNIST, 32×32-c8s2-c8s2-10 for FMNIST, and 32×32-c8s2-c8s2-c8s2-10 for CIFAR-10. *c8s2* is 8 kernels with size 3×3 and stride 2. α and s are set to 0.05 and 0.1 for all optimizers.

Optimizer	MNIST	FMMIST	CIFAR-10
ZOO-Adam	88.51%(0.10)	68.16%(0.13)	diverge
ZOO-Newton	diverge	67.60%(0.23)	diverge
STP	93.74%(0.30)	75.43%(3.86)	diverge
FLOPS	diverge	67.27%(0.19)	diverge
SZO-SCD	94.88%(0.26)	82.63%(0.04)	51.35%(0.86)

Table 3: Comparison among ADMM-based power-constrained optimization and our proposed dynamic power-aware pruning. Power is estimated by the total phase shifts of active MZIs. λ is the weight for the relaxed power penalty term. The model configuration of the 3-layer optical MLP is 64-24-24-10, and the dataset is downsampled MNIST. α and s are set to 0.15 and 0.1 respectively.

Method	Hyperparameter	Power (rad)	Test Accuracy
ADMM	$\lambda = 0.05 \sim 0.3$	2076.6∼2367.1	92.8%∼79.0%
ADMM	$\lambda > 0.3$	-	diverge
Proposed	$p = 0 \sim 1$	1937.6∼993.6	93.9%∼93.1%

Table 4: Power reduction on different datasets with CNNs (same as Table. 2). *DAcc.* and *RAcc.* represent deployed and recovered accuracy, respectively. *PR-Ours* and *PR-FLOPS* are power reduction compared to ours(p=0) and FLOPS.

Dataset	α	s	p	DAcc.	RAcc.	PR-Ours.	PR-FLOPS
MNIST	0.05	0.1	1	87.4%	95.5%	98.8%	97.6%
FMNIST	0.05	0.1	1	65.7%	82.6%	95.6%	98.1%
CIFAR-10	0.05	0.1	1	36.0%	52.5%	96.7%	96.7%

and power constraints. A mixed-training strategy with sparse coordinate descent SZO-SCD is proposed to explore two-level sparsity in ONN deployment and optimization, leading to better training efficiency and robustness. A lightweight dynamic power-aware pruning is proposed to implicitly optimize power during *in situ* learning with near-zero computational cost or accuracy loss. Compared with SOTA ONN on-chip learning methods, our proposed framework boosts the efficiency by 3.7×-7.6× with better crosstalk-robustness, 2× better scalability, and over 10× better power efficiency.

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

Bibi, A.; Bergou, E. H.; Sener, O.; Ghanem, B.; and Richtarik, P. 2020. A Stochastic Derivative-Free Optimization Method with Importance Sampling: Theory and Learning to Control. In *Proc. AAAI*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models. In *Proc. AISec*, 15–26.

Chen, X.; Liu, S.; Xu, K.; Li, X.; Lin, X.; Hong, M.; and Cox, D. 2019. Zo-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. In *Proc. NeurIPS*, 7204–7215.

Deterding, D. 1989. Speaker Normalisation for Automatic Speech Recognition. PhD thesis, University of Cambridge.

Feng, C.; Ying, Z.; Zhao, Z.; Gu, J.; et al. 2020a. Wavelength-division-multiplexing (WDM)-based integrated electronic–photonic switching network (EPSN) for high-speed data processing and transportation. *Nanophotonics* .

Feng, C.; Ying, Z.; Zhao, Z.; et al. 2020b. Wavelength-division-multiplexing-based electronic-photonic network for high-speed computing. In *Proc. SPIE, Smart Photonic and Optoelectronic Integrated Circuits XXII*.

Feng, C.; Zhao, Z.; Ying, Z.; Gu, J.; Pan, D. Z.; and Chen, R. T. 2020c. Compact design of On-chip Elman Optical Recurrent Neural Network. In *Proc. CLEO*.

Ghadimi, S.; and Lan, G. 2013. Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. *SIAM Journal on Optimization* .

Gorbunov, E.; Bibi, A.; Sener, O.; Bergou, E. H.; and Richtárik, P. 2020. A Stochastic Derivative Free Optimization Method with Momentum. In *Proc. ICLR*.

Gu, J.; Zhao, Z.; Feng, C.; Li, W.; Chen, R. T.; and Pan, D. Z. 2020a. FLOPS: Efficient On-Chip Learning for Optical Neural Networks Through Stochastic Zeroth-Order Optimization. In *Proc. DAC*.

Gu, J.; Zhao, Z.; Feng, C.; Zhu, H.; Chen, R. T.; and Pan, D. Z. 2020b. ROQ: A Noise-Aware Quantization Scheme Towards Robust Optical Neural Networks with Low-bit Controls. In *Proc. DATE*.

Gu, J.; Zhao, Z.; Feng, C.; et al. 2020c. Towards Area-Efficient Optical Neural Networks: an FFT-based architecture. In *Proc. ASPDAC*.

Gu, J.; Zhao, Z.; Feng, C.; et al. 2020d. Towards Hardware-Efficient Optical Neural Networks: Beyond FFT Architecture via Joint Learnability. *IEEE TCAD* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*.

Huang, L.; Liu, X.; Lang, B.; Yu, A. W.; Wang, Y.; and Li, B. 2018. Orthogonal Weight Normalization: Solution to Optimization over Multiple Dependent Stiefel Manifolds in Deep Neural Networks. In *Proc. AAAI*.

Hughes, T. W.; Minkov, M.; Shi, Y.; and Fan, S. 2018. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* .

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images .

LeCun, Y. 1998. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Liu, S.; Kailkhura, B.; Chen, P.-Y.; Ting, P.; Chang, S.; and Amini, L. 2018. Zeroth-Order Stochastic Variance Reduction for Nonconvex Optimization. In *Proc. NeurIPS*.

Liu, W.; Liu, W.; Ye, Y.; Lou, Q.; Xie, Y.; and Jiang, L. 2019. HolyLight: A Nanophotonic Accelerator for Deep Learning in Data Centers. In *Proc. DATE*.

Milanizadeh, M.; Aguiar, D.; Melloni, A.; and Morichetti, F. 2019. Canceling Thermal Cross-Talk Effects in Photonic Integrated Circuits. *J. Light. Technol.* .

Miscuglio, M.; and Sorger, V. J. 2020. Photonic tensor cores for machine learning. *Applied Physics Review* .

Nesterov, Y.; and Spokoiny, V. 2017. Random Gradient-Free Minimization of Convex Functions. *Foundations of Computational Mathematics* .

Ohta, M.; Berger, N.; Sokolov, A.; and Riezler, S. 2020. Sparse Perturbations for Improved Convergence in Stochastic Zeroth-Order Optimization. *arXiv preprint arXiv:2006.01759* .

Ramey, C.; et al. 2020. Silicon Photonics for Artificial Intelligence Acceleration. In *Proc. HotChips*.

Reck, M.; Zeilinger, A.; Bernstein, H.; et al. 1994. Experimental realization of any discrete unitary operator. *Physical review letters* .

Ribeiro, A.; Ruocco, A.; Vanacker, L.; et al. 2016. Demonstration of a 4×4-port universal linear circuit. *Optica* .

Shen, Y.; Harris, N. C.; Skirlo, S.; et al. 2017. Deep learning with coherent nanophotonic circuits. *Nature Photonics* .

Tan, D. T. H.; Grieco, A.; and Fainman, Y. 2014. Towards 100 channel dense wavelength division multiplexing with 100GHz spacing on silicon. *Opt. Express* .

Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2019. AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks. In *Proc. AAAI*.

Vivien, L.; Polzer, A.; Marris-Morini, D.; et al. 2012. Zero-bias 40Gbit/s germanium waveguide photodetector on silicon. *Opt. Express* .

Wang, H.; Qian, H.; and Yu, Y. 2018. Noisy Derivative-Free Optimization With Value Suppression. In *Proc. AAAI*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *Arxiv* .

Ying, Z.; Feng, C.; Zhao, Z.; Dhar, S.; et al. 2020a. Electronic-photonic Arithmetic Logic Unit for High-speed Computing. *Nature Communications* .

Ying, Z.; Feng, C.; Zheng Zhao, J. G.; et al. 2020b. Sequential logic and pipelining in chip-based electronic-photonic digital computing. *IEEE Photonics Journal* .

Zhang, T.; et al. 2019. Efficient training and design of photonic neural network through neuroevolution. *arXiv* .

Zhao, P.; Chen, P.-Y.; Wang, S.; and Lin, X. 2020. Towards Query-Efficient Black-Box Adversary with Zeroth-Order Natural Gradient Descent. In *Proc. AAAI*.

Zhao, Z.; Gu, J.; Ying, Z.; et al. 2019a. Design Technology for Scalable and Robust Photonic Integrated Circuits. In *Proc. ICCAD*.

Zhao, Z.; Liu, D.; Li, M.; et al. 2019b. Hardware-software Co-design of Slimmed Optical Neural Networks. In *Proc. ASPDAC*.

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

Zokaee, F.; Lou, Q.; Youngblood, N.; et al. 2020. LightBulb: A Photonic-Nonvolatile-Memory-based Accelerator for Binarized Convolutional Neural Networks. In *Proc. DATE*.