Toward Hardware-Efficient Optical Neural Networks: Beyond FFT Architecture via Joint Learnability

Jiaqi Gu, *Student Member, IEEE*, Zheng Zhao, Chenghao Feng, *Student Member, IEEE*, Zhoufeng Ying, *Member, IEEE*, Mingjie Liu, *Student Member, IEEE*, Ray T. Chen, *Fellow, IEEE*, and David Z. Pan, *Fellow, IEEE*

***Abstract*— As a promising neuromorphic framework, the optical neural network (ONN) demonstrates ultrahigh inference speed with low energy consumption. However, the previous ONN architectures have high area overhead which limits their practicality. In this article, we propose an area-efficient ONN architecture based on structured neural networks, leveraging optical fast Fourier transform for efficient computation. A two-phase software training flow with structured pruning is proposed to further reduce the optical component utilization. Experimental results demonstrate that the proposed architecture can achieve 2.2–3.7× area cost improvement compared with the previous singular value decomposition-based architecture with comparable inference accuracy. A novel optical microdisk-based convolutional neural network architecture with joint learnability is proposed as an extension to move beyond Fourier transform and multilayer perception, enabling hardware-aware ONN design space exploration with lower area cost, higher power efficiency, and better noise-robustness.**

***Index Terms*—Hardware-efficient, nanophotonics, neural network hardware, optical computing, performance optimization.**

I. Introduction

Deep neural networks (DNNs) have demonstrated superior performance in a variety of intelligent tasks, for example convolutional neural networks (CNNs) on image classification [1] and recurrent neural networks on language translation [2]. Multilayer perceptrons (MLPs) are among the most fundamental components in modern DNNs, which are typically used as regression layers, classifiers, embedding layers, attention layers, etc. However, it becomes challenging for traditional electrical digital von Neumann schemes to support escalating computation demands owing to speed and energy inefficiency [3]–[7]. To resolve this issue, significant efforts have been made on hardware design of neuromorphic computing frameworks to improve the computational speed of neural networks, such as electronic architectures [8]–[10] and photonic architectures [11]–[15]. Among extensive neuromorphic computing systems, optical neural networks (ONNs) distinguish themselves by ultrahigh bandwidth, ultralow latency, and near-zero energy consumption. Even though ONNs are currently not competitive in terms of area cost, they still offer a promising alternative approach to microelectronic implementations given the above advantages.

Recently, several works demonstrated that MLP inference can be efficiently performed at the speed of light with optical components, e.g., spike processing [11] and reservoir computing [16]. They claimed a photodetection rate over 100 GHz in photonic networks, with near-zero energy consumption if passive photonic components are used [17]. Based on matrix singular value decomposition (SVD) and unitary matrix parametrization [18], [19], Shen *et al.* [3] designed and fabricated a fully ONN that achieves an MLP with Mach–Zehnder interferometer (MZI) arrays. Once the weight matrices in the MLP are trained and decomposed, thermo-optic phase shifters (PSs) on the arms of MZIs can be set up accordingly. Since the weight matrices are fixed after training, this fully ONN can be completely passive, thus minimizes the total energy consumption. However, this SVD-based architecture is limited by high photonic component utilization and area cost. Considering a single fully connected layer with an $m \times n$ weight matrix, the SVD-based ONN architecture requires $\mathcal{O}(m^2 + n^2)$ MZIs for implementation. Another work [20] proposed a slimmed ONN architecture ($T\Sigma U$) based on the previous one [3], which substitutes one of the unitary blocks with a sparse tree network. However, its area cost improvement is limited. Therefore, this high hardware complexity of the SVD-based ONN architecture has become the bottleneck of its hardware implementation.

In addition to hardware implementation, recent advances in neural architecture design and network compression techniques have shown significant reduction in computational

Manuscript received March 14, 2020; revised June 13, 2020 and August 3, 2020; accepted September 18, 2020. Date of publication September 29, 2020; date of current version August 20, 2021. This work was supported in part by the Multidisciplinary University Research Initiative Program through the Air Force Office of Scientific Research under Contract FA 9550-17-1-0071, monitored by Dr. Gernot S. Pomrenke. The preliminary version has been presented at the ACM/IEEE Asian and South Pacific Design Automation Conference (ASP-DAC) in 2020. This article was recommended by Associate Editor J. Xu. *(Corresponding author: Jiaqi Gu.)*

Jiaqi Gu, Chenghao Feng, Mingjie Liu, Ray T. Chen, and David Z. Pan are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78731 USA (e-mail: jqgu@utexas.edu).

Zheng Zhao is with the Design Group, Synopsys Inc., Mountain View, CA 94043 USA.

Zhoufeng Ying is with the Silicon Photonics Department, Alpine Optoelectronics, Fremont, CA 94538 USA.

Digital Object Identifier 10.1109/TCAD.2020.3027649

cost. For example, structured neural networks (SNNs) [21] were proposed to significantly reduce computational complexity and thus, become amenable to hardware. Besides, network pruning offers another powerful approach to slimming down neural networks by cutting off insignificant neuron connections. While nonstructured pruning [22] produces random neuron sparsity, group sparsity regularization, [23] and structured pruning [9] can lead to better network regularity and hardware efficiency. However, readily available pruning techniques are rather challenging to be applied to the SVD-based architecture due to some issues, such as accuracy degradation and hardware irregularity. The gap between hardware-aware pruning and the SVD-based architecture gives another motivation for a pruning-friendly ONN architecture.

In this article, we propose a new ONN architecture that improves area efficiency over previous ONN architectures. It leverages optical fast Fourier transform (OFFT) and its inverse (OIFFT) to implement SNNs, achieving lower optical component utilization. It also enables the application of structured pruning given its architectural regularity. The proposed architecture partitions the weight matrices into block-circulant matrices [24] and efficiently performs circulant matrix multiplication through OFFT/OIFFT. We also adopt a two-phase software training flow with structured pruning to further reduce photonic component utilization while maintaining comparable inference accuracy to previous ONN architectures. We extend this architecture to a hardware-efficient optical CNN design with joint learnability, and demonstrate its superior power efficiency and noise-robustness compared with Fourier transform-based design. The main contributions of this work are as follows.

1) We propose a novel, area-efficient ONN architecture with OFFT/OIFFT, and exploit a two-phase software training flow with structured pruning to learn hardware-friendly sparse neural networks that directly eliminate part of OFFT/OIFFT modules for further area efficiency improvement.
2) We experimentally show that pruning is challenging to be applied to previous ONN architectures due to accuracy loss and retrainability issues.
3) We experimentally demonstrate that our proposed architecture can lead to an area saving of 2.2–3.7× compared with the previous SVD-based ONN architecture, with negligible inference accuracy loss.
4) We extend our ASP-DAC version of ONN architecture [25] to a novel design for microdisk (MD)-based frequency-domain optical CNNs with high parallelism.
5) We propose a trainable frequency-domain transform structure and demonstrate it can be pruned with high sparsity and outperforms traditional Fourier transform with less component count, higher power efficiency, and better noise-robustness.

The remainder of this article is organized as follows. Section II introduces the background knowledge for our proposed architecture. Section III presents details about the proposed ONN architecture and software pruning flow. Section IV analytically compares our hardware utilization with the SVD-based architecture. Section V demonstrates an extension to optical CNN with trainable transform structures. Section VI reports the experimental results for our proposed ONN architecture and its CNN extension, followed by the conclusion in Section VII.

II. Preliminaries

In this section, we introduce the background knowledge for our proposed architecture. We discuss principles of cirulant matrix representation and its fast computation algorithms in Section II-A and illustrate structured pruning techniques with Group Lasso regularization in Section II-B.

A. FFT-Based Circulant Matrix Computation

Unlike the SVD-based ONNs which focus on classical MLPs, our proposed architecture is based on SNNs with circulant matrix representation. SNNs are a class of neural networks that are specially designed for computational complexity reduction, whose weight matrices are regularized using the composition of structured submatrices [21]. Among all structured matrices, circulant matrices are often preferred in recent SNN designs.

As an example, we show an $n \times n$ circulant matrix $\boldsymbol{W}$ as follows:

$$\begin{bmatrix} w_0 & w_{n-1} & \cdots & w_1 \\ w_1 & w_0 & \cdots & w_2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{n-1} & w_{n-2} & \cdots & w_0 \end{bmatrix}.$$

The first column vector $\boldsymbol{w} = [w_0, w_1, \ldots, w_{n-1}]^{\mathrm{T}}$ represents all independent parameters in $\boldsymbol{W}$, and other columns are just its circulation.

According to [24], circulant matrix-vector multiplication can be efficiently calculated through fast Fourier transform (FFT). Specifically, given an $n \times n$ circulant matrix $\boldsymbol{W}$ and a length-n vector $\boldsymbol{x}$, $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ can be efficiently performed with $\mathcal{O}(n \log n)$ complexity as

$$\boldsymbol{y} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{w}) \odot \mathcal{F}(\boldsymbol{x})) \quad (1)$$

where $\mathcal{F}(\cdot)$ represents n-point real-to-complex FFT, $\mathcal{F}^{-1}(\cdot)$ represents its inverse (IFFT), and $\odot$ represents complex vector element-wise multiplication (EM).

SNNs benefit from high computational efficiency while maintaining comparable model expressivity to classical NNs. Theoretical analysis [26] shows that SNNs can approximate arbitrary continuous functions with arbitrary accuracy given enough parameters, and are also capable of achieving the identical error bound to that of classical NNs. Therefore, based on SNNs with circulant matrix representation, the proposed architecture features low computational complexity and comparable model expressivity.

B. Structured Pruning With Group Lasso Penalty

The proposed ONN architecture enables the application of structured pruning to further save optical components while maintaining accuracy and structural regularity. Structured pruning trims the neuron connections in NNs to mitigate computational complexity. Unlike ℓ_1 or ℓ_2 norm regularization, which produces arbitrarily appearing zero elements, structured pruning with Group Lasso regularization [9], [27] leads to zero entries in groups. This coarse-grained sparsity is more friendly to hardware implementation than nonstructured sparsity. The formulation of Group Lasso regularization term is given as follows:

$$\boldsymbol{L}_{GL} = \sum_{g=0}^{G} \sqrt{1/p_g}\,\|\beta_g\|_2 \tag{2}$$

where G is the total number of parameter groups, β_g is the parameter vector in the gth group, $\|\cdot\|_2$ represents ℓ_2 norm, p_g represents the vector length of β_g, which accounts for the varying group sizes. Intuitively, the ℓ_2 norm penalty $\|\beta_g\|_2$ encourages all elements in the gth group to converge to 0, and the group-wise summation operation is equivalent to group-level ℓ_1 norm regularization, which contributes to the coarse-grained sparsity. Leveraging the structured pruning together with Group Lasso regularization, our proposed architecture can save even more photonic components.

III. Proposed Architecture

In this section, we will discuss details about the proposed architecture and pruning method. In the first part, we illustrate five stages of our proposed architecture. In the second part, we focus on the two-phase software training flow with structured pruning.

A. Proposed Architecture

Based on SNNs, our proposed architecture implements a structured version of MLPs with circulant matrix representation. A single layer in the proposed architecture performs linear transformation via block-circulant matrix multiplication $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. Consider an n-input, m-output layer, the weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ is partitioned into $p \times q$ submatrices, each being a $k \times k$ circulant matrix. To perform tiled matrix multiplication, the input $\boldsymbol{x}$ is also partitioned into q segments $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{q-1})$. Thus, $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ can be performed in a tiled way

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{p-1} \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{q-1} \boldsymbol{W}_{0j}\boldsymbol{x}_j \\ \sum_{j=0}^{q-1} \boldsymbol{W}_{1j}\boldsymbol{x}_j \\ \vdots \\ \sum_{j=0}^{q-1} \boldsymbol{W}_{p-1j}\boldsymbol{x}_j \end{pmatrix}. \tag{3}$$

The ith segment $y_i = \sum_{j=0}^{q-1} \boldsymbol{W}_{ij}\boldsymbol{x}_j$ is the accumulation of q independent circulant matrix multiplications. Each $\boldsymbol{W}_{ij}\boldsymbol{x}_j$ can be efficiently calculated using the fast computation algorithm mentioned in (1). Based on the aforementioned equations, we realize block-circulant matrix multiplication $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ in five stages: 1) splitter tree (ST) stage to split input optical signals for reuse; 2) OFFT stage to calculate $\mathcal{F}(\boldsymbol{x})$; 3) EM stage to calculate $\mathcal{F}(\boldsymbol{w_{ij}}) \odot \mathcal{F}(\boldsymbol{x_j})$ as described in (1); 4) OIFFT stage to calculate $\mathcal{F}^{-1}(\cdot)$; and 5) combiner tree (CT) stage to accumulate partial multiplications to form the final results.

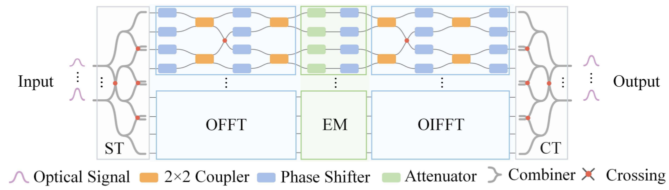


Fig. 1. Schematic diagram of a single layer of the proposed architecture. All adjacent PSs on the same waveguide are already merged into one PS.

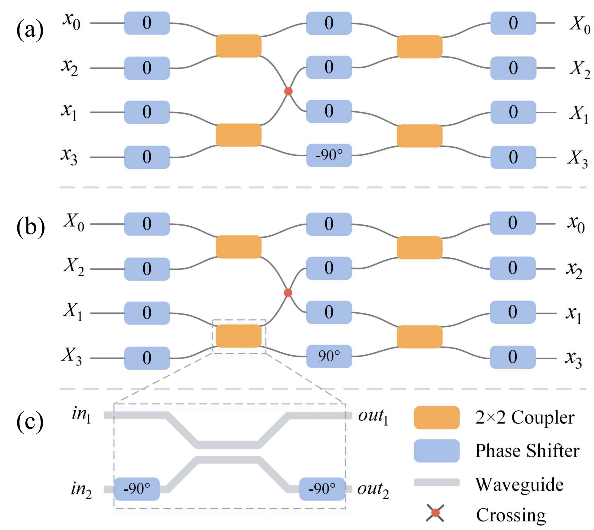


Fig. 2. Schematics of (a) 4-point OFFT, (b) 4-point OIFFT, and (c) 2×2 coupler. Note that PSs shown above are not merged for structural completeness consideration.

$\mathcal{F}(\boldsymbol{w}_{ij})$ can be precomputed and encoded into optical components, thus there is no extra stage to physically perform it. The schematic of our proposed architecture is shown in Fig. 1. Details of the above five stages will be discussed in the rest of this section.

1) OFFT/OIFFT Stages: To better model the optical components used to implement the OFFT/OIFFT stages, we introduce a unitary FFT as

$$\boldsymbol{X}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \boldsymbol{x}_n e^{-i\frac{2\pi kn}{N}}, \quad k = 0, 1, \ldots, N-1. \tag{4}$$

We denote this special operation as $\widehat{\mathcal{F}}(\cdot)$ and its inverse as $\widehat{\mathcal{F}}^{-1}(\cdot)$, to distinguish from the original FFT/IFFT operations. Equivalently, we rewrite the circulant matrix multiplication with the above new operations

$$\boldsymbol{y} = \widehat{\mathcal{F}}^{-1}\big(\mathcal{F}(\boldsymbol{w}) \odot \widehat{\mathcal{F}}(\boldsymbol{x})\big). \tag{5}$$

This unitary FFT operation can be realized with optical components. We first give a simple example for the optical implementation of a 2-point unitary FFT. As shown in (7), the transformation matrix of a 2-point unitary FFT can be decomposed into three transform matrices. They can be directly mapped to a 3-dB directional coupler (DC) with two $-\pi/2$ PSs on its lower input/output ports. The transfer matrix

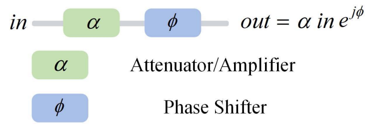


Fig. 3. Complex number multiplication realized by cascaded attenuator/amplifier and PS.

of a 50/50 optical DC is given by

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & j\\ j & 1\end{pmatrix}. \tag{6}$$

The transfer function of a PS is $\text{out} = \text{in} \cdot e^{j\phi}$. For brevity, we refer to this cascaded structure as a 2×2 coupler, which is shown in Fig. 2(c)

$$\begin{aligned}\begin{pmatrix}\text{out}_1\\ \text{out}_2\end{pmatrix} &= \frac{1}{\sqrt{2}}\begin{pmatrix}\text{in}_1 + \text{in}_2\\ \text{in}_1 - \text{in}_2\end{pmatrix}\\ &= \underbrace{\begin{pmatrix}1 & 0\\ 0 & -j\end{pmatrix}}_{\text{output phase shifter}} \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix}1 & j\\ j & 1\end{pmatrix}}_{\text{directional coupler}} \underbrace{\begin{pmatrix}1 & 0\\ 0 & -j\end{pmatrix}}_{\text{input phase shifter}} \begin{pmatrix}\text{in}_1\\ \text{in}_2\end{pmatrix}.\end{aligned} \tag{7}$$

Based on 2×2 couplers and PSs, larger-sized OFFT/OIFFT can be constructed with a butterfly structure. The schematics of a simple 4-point OFFT and OIFFT are shown in Fig. 2(a) and (b). Extra 0-degree PSs are inserted for phase tuning purpose.

This butterfly structured OFFT may have scalability issues because the number of waveguide crossings (CRs) will increase rapidly when the number of point gets larger. However, this unsatisfying scalability will not limit our proposed architecture for two reasons. First, only small values of k, e.g., 2, 4, 8, will be adopted to balance hardware efficiency and model expressivity. Second, input and output sequences can be reordered to avoid unnecessary waveguide crossings, as shown in Fig. 2.

2) EM Stage: In the EM stage, complex vector EMs will be performed in the Fourier domain as $\alpha e^{\phi} \cdot I_{\text{in}} e^{\phi_{\text{in}}} = \alpha\, I_{\text{in}} e^{\phi_{\text{in}}+\phi}$, where I_{in} and ϕ_{in} are magnitude and phase of input Fourier light signals, respectively. Leveraging the polarization of light, we use optical attenuators (ATs) or amplification materials/optical on-chip amplifiers with a scaling factor α to realize modulus multiplication $\alpha \cdot I_{\text{in}}$ and PSs with ϕ phase shift for argument addition $e^{j(\phi+\phi_{\text{in}})}$, which is shown in Fig. 3.

3) ST/CT Stage: We introduce tree-structured splitter/combiner networks to realize input signal splitting and output signal accumulation, respectively. To reuse input segments $\boldsymbol{x}_j$ in multiple blocks, optical splitters (SPs) are used to split optical signals. Similarly, to accumulate partial multiplication results, i.e., $\boldsymbol{y}_i = \sum_{j=0}^{q-1} \boldsymbol{W}_{ij}\boldsymbol{x}_j$, we adopt optical combiners (CBs) for signal addition. Given that SPs can be realized by using combiners in an inversed direction, we will focus on the CT structure for brevity.

The transfer function of an N-to-1 CB is:

$$\text{out} = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \text{in}_l. \tag{8}$$

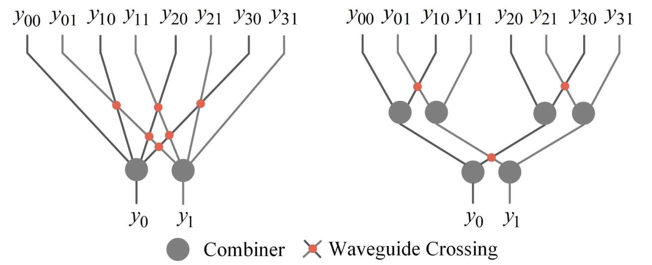


Fig. 4. Comparison between direct combining (left) and CT (right) with 4 length-2 vectors accumulated.

Accumulating q length-k vectors by simply using k q-to-1 combiners introduces a huge number of waveguide crossings which may cause intractable implementation difficulty. Also, combiners with more than two ports are still challenging for manufacturing. In order to alleviate this problem, we adopt a tree-structured combiner network, shown in Fig. 4. This CT consists of $k(q-1)$ combiners and reduces the number of waveguide crossings to $k(k-1)(q-1)/2$. Given that combiners will cause optical intensity loss by a factor of $1/\sqrt{N}$ as shown in (8), we assume there will be optical amplifiers added to the end to compensate this loss.

In terms of cascading multiple layers, our proposed FFT-based MLP is fully optical, such that the output optical signals can be directly fed into the next layer without optical-electrical-optical (O-E-O) conversion. At the end of the last layer, photo-detection is used for signal readout, and the phase information of the outputs are removed, which can be fully modeled during our training process without causing any accuracy loss.

B. Two-Phase Training Flow With Structured Pruning

Structured pruning can be applied to our proposed architecture during training given its architectural regularity. We propose a two-phase software training flow with structured pruning to train a more compact ONN. We first pre-train the model with the Group Lasso regularization term to explore a good initialization. Then we progressively prune the weight blocks by forcing some groups to 0 based on a increasing threshold T such that the corresponding hardware modules can be completely eliminated. Meanwhile we finetune the model to recover accuracy.

IV. THEORETICAL ANALYSIS ON PROPOSED ARCHITECTURE

In this section, we analyze the hardware utilization and compare with previous architectures.

We derive a theoretical estimation of hardware utilization of the proposed architecture, the SVD-based architecture [3], and the slimmed $T\Sigma U$-based architecture [20]. By comparing the hardware component utilization, we show that theoretically our proposed architecture costs fewer optical components than the SVD-based architecture and $T\Sigma U$-based architecture. The comparison results are summarize the in Table I for clear demonstration.

Algorithm 1 Two-Phase Training Flow With Structured Pruning

Input: Initial parameter $\boldsymbol{w}^0 \in \mathbb{R}^{p\times q\times k}$, pruning threshold T, initial training timestep t_{init}, and learning rate α;
Output: Converged parameter $\boldsymbol{w}^t$ and a pruning mask $\boldsymbol{M} \in \mathbb{Z}^{p\times q}$;

```
M ← 1                                        ▷ Initialize pruning mask to all 1
for t ← 1, ..., t_init do                    ▷ Phase 1: Initial training
    L^t(w^{t-1}) ← L^t_base(w^{t-1}) + λ · L^t_GL(w^{t-1})
    w^t ← w^{t-1} − α · ∇_w L^t(w^{t-1})
end for
while w^t not converged do                   ▷ Phase 2: Structured pruning
    for all w^{t-1}_{i,j} ∈ w^{t-1} do
        if ||w^{t-1}_{ij}||_2 < T then
            M[i, j] ← 0                      ▷ Update pruning mask
        end if
    end for
    ApplyDropMask(M, w^{t-1})
    L^t(w^{t-1}) ← L^t_base(w^{t-1}) + λ · L^t_GL(w^{t-1})
    w^t ← w^{t-1} − α · ∇_w L^t(w^{t-1})
    UpdateThreshold(T)                       ▷ Smoothly increase threshold
end while
```

TABLE I
SUMMARY OF HARDWARE COMPONENT COST ON AN $m \times n$ LAYER IN SVD-BASED ONN AND OUR PROPOSED ARCHITECTURE (SIZE-k CIRCULANT BLOCKS). MOST AREA-CONSUMING COMPONENTS ARE CONSIDERED. PS AND DC REPRESENT PS AND DC

	#DC	#PS
SVD-ONN	$m(m-1)+n(n-1)+\max(m,n)$	$\frac{m(m-1)+n(n-1)}{2}$
$T\Sigma U$-ONN	$m(m-1)+2n+\max(m,n)$	$\frac{m(m-1)+2n}{2}$
Our ONN	$\frac{mn(\log_2 k+1)}{k}$	$\frac{mn(2\log_2 k+1)}{k}$

For simplicity, we convert all area-costly components, i.e., 2×2 couplers, MZIs, and attenuators, to 3-dB DCs and PSs. Specifically, one 2×2 coupler can be taken as one DC and two PSs, and one MZI can be taken as two DCs and one PS. Since an attenuator can be achieved by a single-input DC with appropriate transfer factor, we count one attenuator as one DC.

Given an n-input, m-output layer, the SVD-based implementation requires $m(m-1)/2+n(n-1)/2$ MZIs, and $\max(m, n)$ attenuators to realize the weight matrix. Therefore, with the aforementioned assumption, the total number of components it costs is given by

$$\begin{aligned}\#\mathrm{DC}_{\mathrm{SVD}} &= m(m-1)+n(n-1)+\max(m,n)\\ \#\mathrm{PS}_{\mathrm{SVD}} &= m(m-1)/2+n(n-1)/2.\end{aligned} \tag{9}$$

For the slimmed $T\Sigma U$-based ONN architecture [20], one unitary matrix is replaced by a compact sparse tree network consisting of n MZIs. Therefore, the component utilization of $T\Sigma U$-based ONN is given by

$$\begin{aligned}\#\mathrm{DC}_{T\Sigma U} &= m(m-1)+2n+\max(m,n)\\ \#\mathrm{PS}_{T\Sigma U} &= m(m-1)/2+n.\end{aligned} \tag{10}$$

For our architecture, each $k\times k$ circulant matrix costs k attenuators and corresponding components required by k-point OFFT/OIFFT. The following formulation gives the number of components for a k-point OFFT/OIFFT:

$$\begin{aligned}\#\mathrm{DC}_{\mathtt{OFFT}}(k) &= 2\times\#\mathrm{DC}_{\mathtt{OFFT}}(k/2)+k/2=\frac{k}{2}\log_2 k\\ \#\mathrm{PS}_{\mathtt{OFFT}}(k) &= k\big(\log_2 k+1\big).\end{aligned} \tag{11}$$

A phase shift is physically meaningful only when it is within $(-2\pi, 0]$ as phases can wrap around. Hence, multiple successive PSs on the same segment of a waveguide can be merged as one PS, which can be seen when comparing Figs. 1 and 2. Then, the total number of components used in our design to implement an $m\times n$ weight matrix with size-k circulant submatrices is given by

$$\begin{aligned}\#\mathrm{DC}_{\mathtt{Ours}}(k) &= \frac{m}{k}\times\frac{n}{k}\times\big(2\times\#\mathrm{DC}_{\mathtt{OFFT}}(k)+k\big)\\ &= \frac{mn}{k}\big(\log_2 k+1\big)\\ \#\mathrm{PS}_{\mathtt{Ours}}(k) &= \frac{m}{k}\times\frac{n}{k}\times\big(2\times\#\mathrm{PS}_{\mathtt{OFFT}}(k)-k\big)\\ &= \frac{mn}{k}\big(2\log_2 k+1\big).\end{aligned} \tag{12}$$

In practical cases, k will be set to small values, such as 2, 4, and 8. Given arbitrary values of m and n, the proposed architecture costs theoretically fewer optical components than the SVD-based architecture.

We also give a qualitative comparison with incoherent microring resonator-based ONNs (MRR-ONNs). There are two MRR-ONN variants. The first one is based on all-pass mircroring (MR) resonators [29]. The second one proposed later is based on the differential add-drop MR resonators [30]. We assume an $M\times N$ matrix multiplication in the following tasks. Since the physical dimensions of MRs are smaller than couplers and PSs in general, thus a lower area cost can be expected for MRR-ONNs compared with ours. However, in terms of model expressivity, all-pass MRR-ONN is much less than the other two, since it only supports positive weights. Add-drop MRR-ONN and our architecture can support a full-weight range without positive limitation. In terms of robustness, MRR-ONNs are less robust since the MR resonators are more sensitive to device variations and environmental changes than PSs. Especially for add-drop MRR-ONN, its differential structure amplifies the noise on the MR transmission factor by 2 times on its represented weight. Thus, less robustness can be expected for MRR-ONNs. Furthermore, in terms of power consumption, our architecture can benefit from structured sparsity to obtain a much lower power, which will be shown in Section VII. In contrast, for MRR-ONNs, even though a group of weights get pruned to zero values, the corresponding MR resonators are not idle [29], [30], which means its power consumption can barely benefit from pruning techniques. Therefore, from the above qualitative analysis, though our architecture demonstrates a relatively larger footprint than MRR-ONNs, we outperform them in terms of model expressivity, robustness, and power.

V. EXTENSION TO OPTICAL CNN WITH LEARNABLE TRANSFORMATIONS

To demonstrate the applicability of the proposed architecture, we extend this architecture to a compact frequency-domain MD-based optical CNN with joint learnability, where the convolutional kernels and frequency-domain transforms are jointly optimized during hardware-aware training.

A. Microdisk-Based Frequency-Domain CNN Architecture

Given the 2-D nature of photonic integrated chips (PICs), currently we only demonstrate optical designs for MLPs. Previous solutions to accelerate CNNs are based on kernel sliding, convolution unrolling, and time multiplexing [31], [32]. At each time step, the input feature chunks and corresponding convolutional kernels are flattened as a 1-D vector and fed into the ONNs to perform vector dot-product. Another solution to solve this is to use *im2col* algorithm [29], [33], that transforms convolution to general matrix multiplication (GEMM). Convolutional kernels and input features are reshaped as matrix-matrix multiplication, which can be directly mapped on ONNs. Such implementation is inherently inefficient as overlapped convolutional patterns will create a huge amount of data redundancy in the unrolled feature maps. In this work, we proposed to achieve CNNs with a new ONN architecture equipped with learnable transformation structures. Fig. 5 demonstrates our proposed optical MD-based CNN architecture featured by kernel sharing, learnable transformation, and augmented frequency-domain kernel techniques. Multichannel input feature maps are encoded onto multiple wavelengths and input into the learnable frequency-domain transforms, then split into multiple branches through the fanout network for parallel multikernel processing. Frequency-domain convolution is performed in the MD-based kernel banks and the final results are transformed back to the real domain via the reversed transforms. Note that we do not include a detailed discussion on the pooling operations since they are not the computationally intensive parts in NNs. For example, optical comparators can be used to achieve max-pooling. Average-pooling can be implemented by a fixed-weight convolution engine based on combiner-tree networks. Multiple layers can be cascaded through O-E-O conversion. The phase information loss during photo-detection can be fully modeled during training without harming the model expressivity, which is actually a competitive substitute for rectified linear unit (ReLU) activation in the complex NN domain [34]. All of our experiments in later sections model this phase removal during training, which shows that this nonideality induced by photo-detection does not cause any accuracy loss. We will introduce details of the principles of the designed optical CNN in the following section.

B. Kernel Weight Sharing

Modern CNN architectures, e.g., inception architecture [35], adopts weight sharing to reduce the number of parameters in the convolutional layers. For example, a 5×5 2-D convolution involves 25 parameters. It can be replaced by two cascaded lightweight 1×5 and 5×1 convolutions, which only contain ten unique variables. Such a strategy trains a low-rank convolutional kernel and can benefit its photonic implements as it can be directly applicable to 2-D PICs, which is visualized in Fig. 6.

C. Learnable Frequency-Domain Convolution

Spatial domain convolution requires to slide the receptive field of convolutional kernels across the input features. This could induce hardware implementation difficulty and inefficiency as time multiplexing increases the latency and control complexity of photonic convolution. we solve this issue by a parametrized frequency-domain convolution method. As mentioned before, we decompose the 2-D convolution as row-wise and column-wise 1-D convolutions through weight sharing. For brevity, we focus on the column-wise frequency-domain convolution in the following discussion. The same principle also applies to the row-wise convolution. The column-wise convolution can be formulated as:

$$\boldsymbol{w} * \boldsymbol{x} = \mathcal{T}^{-1}(\mathcal{T}(\boldsymbol{w}; \boldsymbol{\phi}) \odot \mathcal{T}(\boldsymbol{x}; \boldsymbol{\phi}); \boldsymbol{\phi}) \tag{13}$$

where $\mathcal{T}(\cdot; \boldsymbol{\phi})$ is the learnable frequency-domain projection, and $\boldsymbol{\phi}$ represents the trainable parameters in it. This parametrized transformation enlarges the parameter space to compensate for the model expressiveness degradation induced by kernel weight sharing. Considering the learnable transform as a high-dimensional unitary rotation, it is not necessary to adopt an inverse transform pair to limit the exploration space. To enable the maximum learnability of our trainable transform structure, we relax the inverse transform to a reversed transform

$$\boldsymbol{w} * \boldsymbol{x} = \mathcal{T}_r\big(\mathcal{T}(\boldsymbol{w}; \boldsymbol{\phi}) \odot \mathcal{T}(\boldsymbol{x}; \boldsymbol{\phi}); \boldsymbol{\phi}_r\big) \tag{14}$$

where $\mathcal{T}_r$ has a reversed butterfly structure but is not constrained to be the inverse of $\mathcal{T}$.

We now discuss how our proposed trainable transform structures can move beyond Fourier transform, thus enable hardware-aware learnability. Fourier transform is a complex domain transformation that is mathematically designed for frequency component extraction. However, the Fourier transform is not necessary to be the best-performed transformation that can be used in CNNs. Other manually designed unitary transforms are also experimentally demonstrated to have a similar ability for signal integration and extraction [36]. Hence, we upgrade the fixed transformation structure to an adaptive structure where all PSs are trainable. As mentioned in Section IV, PSs in the same segment of waveguide can be merged into one PS. Therefore, to avoid redundant trainable PSs, we redesign the learnable basic block, as shown in Fig. 7. For the original transformation, two PSs ϕ_1 and ϕ_2 are placed on the input port of the DC. The transfer function of a learned basic block can be formulated as

$$\begin{aligned}\mathcal{T}(2) &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}\begin{pmatrix} e^{j\phi_1} & 0 \\ 0 & e^{j\phi_2} \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} \cos\phi_1 + j\sin\phi_1 & -\sin\phi_1 + j\cos\phi_1 \\ -\sin\phi_2 + j\cos\phi_2 & \cos\phi_2 + j\sin\phi_2 \end{pmatrix}.\end{aligned} \tag{15}$$

In the reversed transformation structure, the basic block is the same as used in the original transforms since the inverse basic block requires a conjugate transposed transfer function which is not implementable with this basic block. Based on this basic block, we recursively build a trainable N-length transform with a butterfly structure, which can be described as $\log_2 N$ stages of projection, $\log_2 N - 1$ stages of permutation, and a final extra group of PSs. The original transformation, shown in Fig. 7(a),

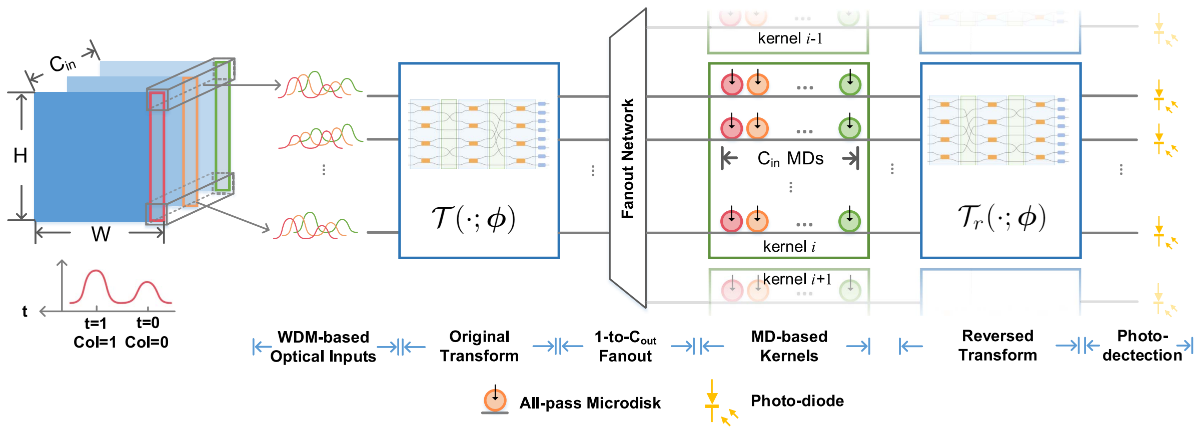


Fig. 5. Architecture of an MD-based optical convolutional layer with trainable frequency-domain transforms. Columns of input features are fed into the architecture in different time steps. Multiple kernels are implemented with multiple photonic chiplets to achieve higher parallelism.

can be formulated as

$$\mathcal{T}(N) = \mathcal{D}\,\mathcal{B}_{\log_2 N-1}(N) \prod_{i=0}^{\log_2 N-2} \mathcal{P}_i(N)\mathcal{B}_i(N) \tag{16}$$

where $\mathcal{B}_i(N)$ the ith stage of butterfly projection, $\mathcal{P}_i(N)$ is the ith stage signal permutation, and the diagonal matrix $\mathcal{D}$ represents the final extra column of PSs. The butterfly projection operator $\mathcal{B}(N)$ is a diagonal matrix with a series of $\mathcal{T}(2)$ as its diagonal submatrices

$$\mathcal{B}(N) = \begin{pmatrix} \mathcal{T}_0(2) & 0 & \cdots & 0 \\ 0 & \mathcal{T}_1(2) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathcal{T}_{N/2-1}(2) \end{pmatrix}. \tag{17}$$

The index permutation operator $\mathcal{P}_i(N)$ can be expressed as a size-N identity matrix with reordered rows. As shown in $\mathcal{P}_0$ and $\mathcal{P}_1$ in Fig. 7, the green entries represent 1, and other blank entries represent 0. Note that the permutation operators in the reversed structure is simply the reversed counterparts in the original structure, i.e., $\mathcal{P}_{i,\text{ori}}(N) = \mathcal{P}^{\text{T}}_{i,\text{rev}}(N)$. The reversed learnable transformation, shown in Fig. 7(b), is designed to have reversed butterfly structure which can be derived as follows:

$$\mathcal{T}_r(N) = \mathcal{D}\left(\prod_{i=0}^{\log_2 N-2} \mathcal{B}_{r,i}(N)\mathcal{P}_{r,i}(N)\right)\mathcal{B}_{r,\log_2 N-1}(N). \tag{18}$$

Note that the reversed transform is not guaranteed to be inverse to the original transform, which requires particular phase configurations discussed later.

Compared with its MZI-based counterparts, this trainable butterfly transformation structure has a constrained projection capability as only a limited set of unitary matrices can be implemented by it [37] and [38]. As shown in unitary group parametrization, a full N-dimensional unitary space $\boldsymbol{U}(N)$ has $N(N-1)/2$ independent parameters, while the butterfly structure substitutes part of parametrized unitary matrices with fixed permutation operators. Hence, based on full 2-D unitary matrices $\boldsymbol{U}(2)$, the butterfly structure has $2N\log_2 N$ independent parameters. Our proposed learnable block $\mathcal{T}(2)$ is a reduced version of $\boldsymbol{U}(2)$, as it only covers half of the full 2-D planar rotation space. The pruned transform space $\mathcal{T}^*(2)$ can be expressed as the conjugate transpose of $\mathcal{T}(2)$, which is not implementable without waveguide crossings

$$\mathcal{T}^*(2) = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & -j \\ -j & 0 \end{pmatrix}\begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}\begin{pmatrix} e^{j\phi_1} & 0 \\ 0 & e^{j\phi_2} \end{pmatrix}. \tag{19}$$

Equivalently, our learnable transformation structure has $N\log_2 N$ free parameters.

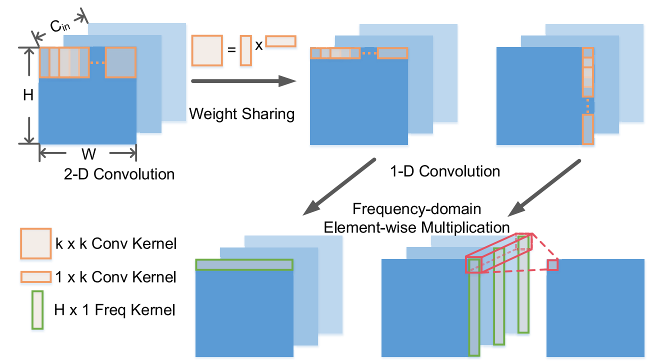


Fig. 6. 2-D convolutional kernel decomposition using weight sharing and frequency-domain transformation.

D. Microdisk-Based Augmented Kernels

To enable highly parallel CNN architecture with reinforced model expressiveness, we propose MD-based augmented convolutional kernels with multilevel parallelism across input features, input channels, and output channels.

In our design, each 2-D convolutional layer consists of two cascaded 1-D frequency-domain convolutions along columns and rows. We will focus on the column-wise convolution, and the same architecture applies to its row-wise counterpart with an extra matrix transposition operation. We denote the input feature map as $\boldsymbol{I} \in \mathbb{R}^{C_{\text{in}}\times H\times W}$, which C_{in}, H, W represent the number of input channel, spatial height, and spatial width, respectively. At time step t, the corresponding column $\boldsymbol{I}_{:,t,:} \in \mathbb{R}^{C_{\text{in}}\times H\times 1}$ will be input into the optical

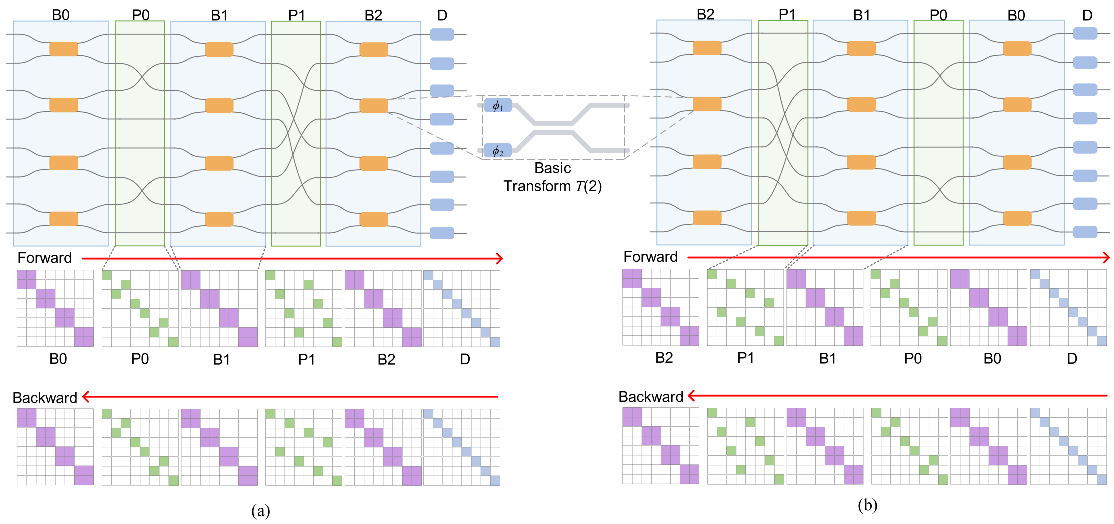


Fig. 7. (a) Original learnable frequency-domain transformation structure. (b) Reversed learnable transformation structure.

CNN. Different input channels are encoded by different wavelengths $\{\lambda_0, \lambda_1, \ldots, \lambda_{C_{\text{in}}-1}\}$. Through the wide-band learnable transformation structure, we obtain the frequency-domain features $\mathcal{T}(\boldsymbol{I}_{:,t,:}; \phi)$. This stage enables parallel transformation across the input channels. Then the optical signals carrying those features will be split into C_{out} planes for data reuse. Such a multidimensional ONN design can be supported by state-of-the-art integration technology with multiple photonic chiplets [39]. In the MD-based convolution stage, $C_{\text{out}} \times C_{\text{in}} \times H$ all-pass MDs are used to implement the frequency-domain kernels $\boldsymbol{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times H}$. Given that the working principle of MD is primarily optical signal magnitude modulation, our augmented kernels are trainable only in the magnitude space without phase modulation. Each convolutional core is designed to perform the convolution of one output channel. This MD-based convolution is different from the previous EM stage consisting of attenuators and PSs. First, all pass MDs can only perform configurable magnitude modulation of the input signals with fixed phase responses, which means the augmented kernels will not expand over the entire complex space. Here, we give the transfer function of an MD

$$
\begin{aligned}
I_{\text{out}} &= W \cdot I_{\text{in}} \\
\cos\theta &= \frac{a^2 + r^2 - W\big(1 + r^2 a^2\big)}{2(1 - W)ar} \\
\phi_{\text{out}} &= \pi + \theta + \arctan \frac{r\sin\theta - 2r^2 a \sin\theta\cos\theta + ra^2\sin\theta}{(a - r\sin\theta)(1 - ra\cos\theta)}
\end{aligned} \tag{20}
$$

where I_{in} is the magnitude of the input light, $I_{\text{out}}, \phi_{\text{out}}$ are magnitude and phase of the output optical signal, θ, a, r are the phase, self-coupling coefficient, and coupling loss factor of an MD, respectively. W is the transmitivity of the MD which corresponds to the trained augmented kernel weight. Typically, parameter a and r are very close to 1. Our proposed architecture enables another level of parallelism across output channels. Given that different convolutional kernels share the same input features, multiple MD convolution cores, and reversed transform structures will share one original transform structure for hardware reuse and highly parallel convolution.

A higher modeling capacity is enabled by our augmented kernel technique. Instead of training spatial kernels $\boldsymbol{w}$, we explicitly train the latent weights $\boldsymbol{W}$ in the frequency domain without performing $\mathcal{T}(\boldsymbol{w}; \boldsymbol{\phi})$ during training. The augmented latent weights $\boldsymbol{W}$ will not meet the conjugate symmetry constraint as its spatial-domain counterparts are not real-valued. Hence, this enables a potentially infinite solution space in the spatial kernel space with various kernel sizes and shapes.

We briefly discuss the scalability of this when mode-division (WDM)-based highly parallel architecture. WDM plays an important role in the high parallelism of our proposed frequency-domain optical CNN. Currently, the widely acknowledged maximum number of wavelength in the single-mode dense-WDM (DWDM) is over 200 [40]–[42]. WDM multiplexing is further considered, higher parallelism can be supported given the current technology. This means in our architecture has enough parallelism to support most modern CNN architectures.

E. Discussion: Exploring Inverse Transform Pairs in Constrained Unitary Space

In manually designed frequency-domain convolution algorithms, domain transformation will be designed to be inverse, e.g., FFT and IFFT. This implies an inverse constraint between two mutually reversed transform structures $\mathcal{T}$ and $\mathcal{T}_r$. To be able to realize trainable inverse transform pairs, we add unitary constraints to our learnable transform structures

$$
\mathcal{T}_r\big(\cdot, \boldsymbol{\phi}_r\big) = \mathcal{T}^{-1}(\cdot; \boldsymbol{\phi}). \tag{21}
$$

Inverse constraints typically can be addressed via adding a regularization term in training

$$
\mathcal{L}_{\text{inv}} = \|\boldsymbol{U}_r \boldsymbol{U} - \boldsymbol{I}\|_2. \tag{22}
$$

However, this requires explicit transfer matrices of $\mathcal{T}$ and $\mathcal{T}_r$ to compute this regularization term [43], which is memory-intensive and computational expensive as indicated by (17)

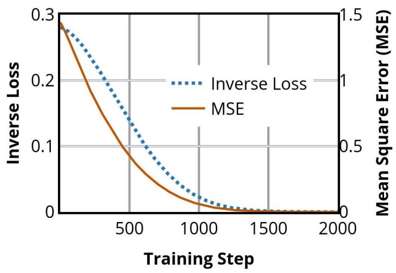


Fig. 8. Training curve of inverse loss $\mathcal{L}_{\text{inv}}$ and mean square error between trained phase configurations and theoretical 4-point OFFT settings.

and (18). We propose an efficient regularization method to exert inverse constraint

$$\mathcal{L}_{\text{inv}} = \|\mathcal{T}_r(\mathcal{T}(\boldsymbol{e})) - \boldsymbol{e}\|_2, \quad \boldsymbol{e} \in \mathbb{C}^N \tag{23}$$

where $\boldsymbol{e}$ is the orthonormal bases of N-dimensional complex space. Notice that if $\mathcal{T}_r(\mathcal{T}(\boldsymbol{e})) = \boldsymbol{e}$, then for any $\boldsymbol{x} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{e}$ the following statement holds:

$$\mathcal{T}_r(\mathcal{T}(\boldsymbol{x})) = \mathcal{T}_r\Big(\mathcal{T}\Big(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{e}\Big)\Big) = \boldsymbol{\alpha}^{\mathrm{T}}\mathcal{T}_r(\mathcal{T}(\boldsymbol{e})) = \boldsymbol{x}. \tag{24}$$

Thus, transforms $\mathcal{T}$ and $\mathcal{T}_r$ are inverse transforms once the regularization loss reaches 0. This surrogate method reduce the computation complexity from $\mathcal{O}(N^2 \log_2 N)$ in (16) to $\mathcal{O}(N \log_2 N)$, where diagonal matrix multiplication with $\mathcal{B}(N)$ is simplified by 2×2 submatrix multiplication with $\mathcal{T}(2)$.

Using our proposed inverse pair regularization method, we show that our trainable transform $\mathcal{T}$ can efficiently learn Fourier transform by setting $\mathcal{T}_r$ as OIFFT. Fig. 8 demonstrates that the trainable transform will quickly converge to the theoretical OFFT as the mean square error between trained phase settings and target PS settings reduces to 0 when the loss converges.

F. Discussion: Hardware-Aware Pruning for Trainable Transforms

In this section, we demonstrate that our proposed trainable transform has excellent compatibility with hardware-aware pruning techniques. Compared to the fixed manual design of frequency-domain transforms, e.g., OFFT, we can further boost the hardware efficiency by eliminating a subset of phase shifter columns inside the trainable transforms. With this fine-grained structured pruning, we can improve the area, power, and noise-robustness since phase shifters contribute to nearly 50% of the total area and majority of the total power and noise. We adopt a phase-wrapping Group Lasso regularization similar to (2) together with incremental pruning technique to slim the trainable transforms targeted at lower area cost and lower power consumption. The proposed phase-wrapping Group Lasso (PhaseGL) is formulated as

$$\begin{aligned} \boldsymbol{L}_{\text{PhaseGL}} &= \sum_{g=0}^{G} \sqrt{1/p_g}\left\|\phi_g - \phi_g^*\right\|_2 \\ \phi_{g,i}^* &= \begin{cases} 0, & \phi_{g,i} \in [0, \pi), \quad 0 \le i < p_g \\ 2\pi, & \phi_{g,i} \in [\pi, 2\pi), \quad 0 \le i < p_g \end{cases} \end{aligned} \tag{25}$$

where ϕ_g is a column of PSs and this regularization term encourages phases toward their corresponding prunable targets ϕ_g^*. G is the total columns of PSs, which is $(\log_2 N + 1)$ for a length-N transform. Once the group lasso of a column falls below a threshold $T_{\mathcal{T}}$, the entire column of PSs are pruned. The ratio of pruned columns to all PS columns is called transform sparsity ($\mathcal{T}$ sparsity), defined as

$$s_{\mathcal{T}} = \frac{\left|\left\{\phi_g \middle| \sqrt{1/p_g}\left\|\phi_g - \phi_g^*\right\| < T_{\mathcal{T}}\right\}\right|}{G}.$$

Our proposed regularization and pruning strategy improves area cost as an entire column of PSs are pruned to save chip area in the actual layout. Furthermore, power consumption and noise robustness can also be improved as a majority of power consumption and noises are from trainable transform structures [20], [43], [44].

TABLE II
HARDWARE COST SUMMARY ON THE PROPOSED MD-BASED OPTICAL CNN ARCHITECTURE. THE INPUT FEATURE MAP IS OF SIZE $H \times W \times C_{\text{in}}$, THE NUMBER OF OUTPUT CHANNELS IS C_{out}, AND THE SPARSITY OF THE LEARNABLE TRANSFORMS IS $s_{\mathcal{T}} \in [0, 1]$. FOR SIMPLICITY, WE ASSUME $H = W$, WHICH IS A WIDELY USED CONFIGURATION FOR MOST CNNS. GIVEN THE ULTRACOMPACT FOOTPRINT OF AN MD, E.G., $5 \times 5\ \mu m^2$ [47], WE COUNT 100 MDS AS ONE DC IN THE AREA ESTIMATION. THE ROW-WISE AND COLUMN-WISE CONVOLUTIONS ARE BOTH COUNTED IN THIS TABLE

Structure	Hardware Cost
$\mathcal{T}$	$H\log_2 H$ DCs + $2s_{\mathcal{T}}H(1+\log_2 H)$ PSs
Kernel	$2HC_{in}C_{out}$ MDs$\approx \frac{H}{50}C_{in}C_{out}$ DCs
$\mathcal{T}_r$	$H\log_2 HC_{out}$ DCs + $2s_{\mathcal{T}}H(1+\log_2 H)C_{out}$ PSs
Total	$\approx H(\log_2 H + \frac{C_{in}}{50})C_{out}$ DCs + $2s_{\mathcal{T}}H(1+\log_2 H)C_{out}$ PSs

G. Discussion: Hardware Cost of the Proposed MD-Based Optical CNN

We give a summary on the hardware component usage of the proposed MD-based optical CNN architecture in Table II. Our architecture shares the original transform among multiple kernels to save area. Our proposed pruning technique can regularly sparsify the transform structures for further area reduction. The MD-based convolution stage is very compact since the footprint of an MD is two-order-of-magnitude smaller than a DC. In contrast, the SVD-based ONN costs $H(C_{\text{out}}^2 + C_{\text{in}}^2 \times K^4)$ DCs and $H(C_{\text{out}}^2/2 + C_{\text{in}}^2 \times K^4/2)$ PSs to achieve the same latency with our architecture, i.e., H forwards to finish a convolutional layer, where K is the spatial kernel size. For example, if we set $H = 64$, $C_{\text{in}} = C_{\text{out}} = 32$, $K = 3$, $s_{\mathcal{T}} = 0.5$, our architecture uses $> 370\times$ fewer DCs and $> 180\times$ fewer PSs than the single-wavelength SVD-based ONN. If SVD-based ONNs also use WDM techniques for higher parallelism with the same number of wavelength as ours, i.e., 32, we still outperform theirs by $11.6\times$ fewer DCs and $5.6\times$ fewer PSs. Hence, our frequency-domain CNN architecture outperforms previous MZI-ONNs with higher computational efficiency and better scalability by a large margin.

TABLE III
OPTICAL COMPONENT SIZES USED IN THE AREA ESTIMATION

Optical Component	Length (μm)	Width (μm)
3-dB Directional Coupler [3]	54.4	40.3
Thermo-optic Phase Shifter [44]	60.16	0.50
2-to-1 Optical Combiner [48]	20.00	3.65
Waveguide Crossing [49]	5.9	5.9

VI. EXPERIMENTAL RESULTS

We conduct numerical simulations for functionality validation and evaluate our proposed architecture on the handwritten digit recognition dataset (MNIST) [49] with various network configurations. Quantitative evaluation shows that our proposed architecture outperforms the SVD-based and $T\Sigma U$-based ONN architectures in terms of area cost without any accuracy degradation. We further evaluate our proposed MD-based optical CNN architecture and demonstrates its superior power reduction and robustness improvement on MNIST and FashionMNIST [50] dataset.

A. Simulation Validation

To validate the functionality of our proposed architecture, we conduct optical simulations on a 4×4 circulant matrix-vector multiplication module using Lumerical INTERCONNECT tools. First, we encode a 4×4 identity weight matrix into our architecture and input 4 parallel optical signals to validate its functionality. For brevity, we plot several different representative cases in Fig. 9(a). It shows that our designed architecture can correctly realize identity projection. Further, we randomly generate a length-4 real-valued weight vector $\boldsymbol{w} = (0.2, -0.1, 0.24, -0.15)$ to represent a circulant matrix, and encode $\mathcal{F}(\boldsymbol{w}) = (0.19e^{0j}, 0.064e^{-2.246j}, 0.69e^{0j}, 0.064e^{2.246j})$ into attenuators and PSs in the EM stage. The simulation results in Fig. 9(b) shows good fidelity ($< 1.2\%$ maximum relative error) to the ground truth results.

B. Comparison Experiments on FFT-Based ONNs

To evaluate our proposed ONN architecture, we conduct a comparison experiment on a machine learning dataset MNIST [28], and compare the hardware utilization, model expressivity among four architectures: 1) SVD-based architecture [3]; 2) $T\Sigma U$-based architecture [20]; 3) ours without pruning; and 4) ours with pruning.

We implement the proposed architecture with different configurations in PyTorch and test the inference accuracy on a machine with an Intel Core i9-7900X CPU and an NVIDIA TitanXp GPU. We set λ to 0.3 for the Group Lasso regularization term, initialize all trainable weights with a Kaiming-normal initializer [51], adopt the Adam optimizer [52] with initial learning rate $= 1 \times 10^{-3}$ and a step-wise exponential-decay learning rate schedule with decay rate $= 0.9$. We use the ideal ReLUs activation function as nonlinearity. All NN models are trained for 40 epochs with a mini-batch size of 32 till fully converged. The structured sparsity for our proposed FFT-based MLP is defined as the percentage of pruned parameters in all parameters, i.e., $|\{w| \|\boldsymbol{w}_{ij}\|_2 < T\}|/|\boldsymbol{w}|$. We call it block sparsity.

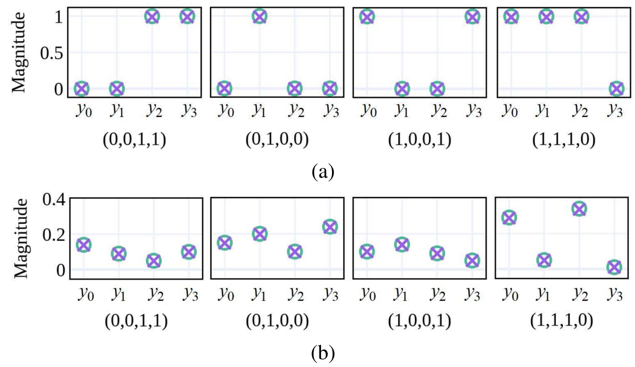


Fig. 9. (a) Simulated output intensities (crosses) and ground truth (circles) of a 4×4 identity circulant matrix-vector multiplication. (b) Simulated output intensities (crosses) and ground truth (circles) of a 4×4 circulant matrix-vector multiplication, with $\boldsymbol{w} = (0.2, -0.1, 0.24, -0.15)$. E.g., $(0, 0, 1, 1)$ is the input signal.

For a fair comparison, all architectures are trained with the same hyper-parameters and have similar test accuracy in each experiment configuration. To estimate the component utilization and area cost, we adopt exactly the same type of photonic devices in all architectures, as listed in Table III, and accumulate the area of each optical component for approximation. Placement or routing information is not considered in our estimation.

In Table IV, the first column indicates different neural network configurations. The $T\Sigma U$-based architecture adopts a unique training methodology and claims to have small accuracy degradation ($< 1\%$) [20], thus we assume it has approximately the same accuracy as the SVD-based architecture. In the $T\Sigma U$-based architecture, the total number of MZIs used to implement an $m \times n$ weight matrix is bounded by $n(n+1)/2$.

Among various network configurations, our proposed architecture outperforms the SVD-based architecture and the $T\Sigma U$-based architecture with lower optical component utilization and better area cost. We normalize all areas to our architecture with pruning applied and show the normalized area comparison in Fig. 10. Consistent with analytical formulations in Section IV, the experimental results show that, as the difference between input and output channels for each layer in the original MLPs gets larger, our proposed architecture can save a larger proportion of optical components. Furthermore, ablation experiments on our structured pruning method validate the effectiveness of the proposed two-phase training flow. It can save an extra 30–50% optical components with negligible model expressivity loss.

C. Comparison Among Different Trainable Transform Settings

As mentioned in previous sections, we extend our ONN architecture to MD-based CNNs with trainable frequency-domain transforms. We will demonstrate several experimental evaluations on our proposed MD-based CNN architecture.

TABLE IV
COMPARISON OF INFERENCE ACCURACY AND HARDWARE UTILIZATION ON MNIST DATASET WITH DIFFERENT CONFIGURATIONS. FOR EXAMPLE, CONFIGURATION (28 × 28)-1024(8)-10(2) INDICATES A 2-LAYER NEURAL NETWORK, WHERE THE FIRST LAYER HAS 784 INPUT CHANNELS, 1024 OUTPUT CHANNELS WITH SIZE-8 CIRCULANT MATRICES, AND SO ON

	Network Configurations	Block Sparsity	#Parameter	Accuracy	#DC	#PS	Area (cm^2)
Model 1	SVD [3]: (28×28)-400-10	0.00	318 K	98.49%	934 K	467 K	20.62
	TΣU [20]: (28×28)-400-10	0.00	318 K	98.49%	777 K	388 K	17.15
	Ours w/o Prune: (28×28)-1024(8)-10(2)	0.00	105 K	98.32%	412 K	718 K	9.33
	Ours w/ Prune: (28×28)-1024(8)-10(2)	0.40	63 K	98.26%	244 K	425 K	5.53
Model 2	SVD [3]: (14×14)-70-10	0.00	14 K	96.93%	48 K	24 K	1.07
	TΣU [20]: (14×14)-70-10	0.00	14 K	96.93%	44 K	22 K	0.97
	Ours w/o Prune: (14×14)-256(4)-10(2)	0.00	14 K	96.93%	40 K	67 K	0.90
	Ours w/ Prune: (14×14)-256(4)-10(2)	0.45	8 K	96.91%	22 K	36 K	0.49
Model 3	SVD [3]: (28×28)-400-128-10	0.00	366 K	98.58%	967 K	483 K	21.35
	TΣU [20]: (28×28)-400-128-10	0.00	366 K	98.58%	794 K	396 K	17.52
	Ours w/o Prune: (28×28)-1024(8)-128(4)-10(2)	0.00	134 K	98.53%	501 K	868 K	11.34
	Ours w/ Prune: (28×28)-1024(8)-128(4)-10(2)	0.39	81 K	98.43%	289 K	517 K	6.77
Model 4	SVD [3]: (14×14)-160-160-10	0.00	59 K	97.67%	141 K	70 K	3.10
	TΣU [20]: (14×14)-160-160-10	0.00	59 K	97.67%	91 K	45 K	2.00
	Ours w/o Prune: (14×14)-256(4)-256(8)-10(2)	0.00	22 K	97.67%	73 K	123 K	1.64
	Ours w/ Prune: (14×14)-256(4)-256(8)-10(2)	0.37	14 K	97.52%	47 K	79 K	1.05

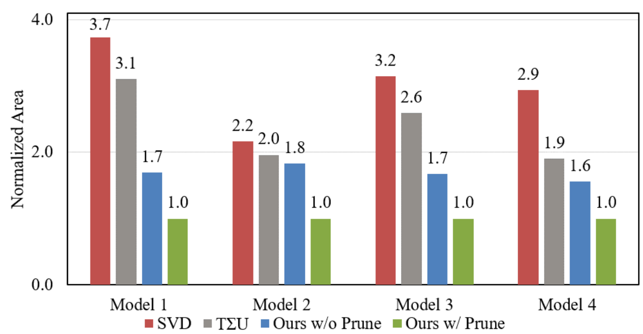


Fig. 10. Normalized area comparison with different model configurations. *Model* 1–4 refer to Table IV. SVD refers to [3] and $T\Sigma U$ refers to [20].

TABLE V
ACCURACY COMPARISON AMONG FOUR TRAINABLE TRANSFORM SETTINGS. THE MODEL IS 16 × 16-C16-BN-MAXPOOL5-F32-F10.

Settings	`AllFree`	`Shared`	`Inverse`	`InvShared`
Test Accuracy	96.88%	96.13%	96.41%	96.40%

First, we discuss how different transform settings impact the CNN performance. Recall that each 2-D frequencydomain convolution involves total four trainable transforms, denoted as $\mathcal{T}_{\text{row}}; \mathcal{T}_{\text{row;r}}; \mathcal{T}_{\text{col}}; \mathcal{T}_{\text{col;r}}$. We evaluate the performance of four different transform settings on MNIST dataset: 1) four transforms are trained independently (`AllFree`); 2) column-wise and row-wise convolutions share the same transform as $\mathcal{T}_{\text{row}} = \mathcal{T}_{\text{col}}, \mathcal{T}_{\text{row},r} = \mathcal{T}_{\text{col},r}$ (`Shared`); 3) reversed transforms are constrained to be close to the inverse transform as $\mathcal{T}_{\text{row},r} \approx \mathcal{T}_{\text{row}}^{-1}, \mathcal{T}_{\text{col},r} \approx \mathcal{T}_{\text{col}}^{-1}$ (`Inverse`); and 4) transforms are shared between column-wise and row-wise convolutions and the inverse constraints are applied (`InvShared`). Table V shows the comparison results.

Based on the results, we observe that the inverse constraint and shared transform produces no benefits in terms of inference accuracy. Training the original and reversed transforms across row-wise and column-wise convolutions independently offers the best results. Thus, we will use `AllFree` transform settings for our experiments.

D. Comparison With Hardware-Aware Transform Pruning

To jointly optimize classification accuracy and hardware cost in terms of area, power, and robustness, we perform hardware-aware pruning assisted by phase-wrapping Group Lasso regularization to our proposed trainable transforms. The weight for L_{PhaseGL} is 0.05, and we set ten epochs for the first pretraining phase and 40 epochs for incremental structured pruning.

1) Power Consumption Evaluation: We calculate the energy cost by summing all phase shifts as they are proportional to power consumption, and show the energy saved by our pruned transforms in Table VI. We also evaluate the power consumption by applying pruned trainable transform in our block-circulant matrix-based MLP architecture. The block sparsity, transform sparsity $\mathcal{T}$ sparsity, power consumption, and area cost are estimated in Table VII. Therefore, our energy-saving and area-efficient ONN architecture is more suitable for resource-constrained applications, e.g., edge computing and online learning tasks [53], [54].

2) Variation-Robustness Evaluation: To evaluate the noise-robustness of the frequency-domain transform, we inject device-level variations into PSs to introduce phase programming errors and demonstrate the accuracy and its variance under different noise intensities σ on MNIST and FashionMNIST dataset. Specifically, we inject Gaussian noise $\Delta\gamma \sim \mathcal{N}(0, \sigma^2)$ into the γ coefficient of each PS to perturb its phase response $\phi_n = (\gamma + \Delta\gamma)v^2$, where γ is calculated by the voltage that can produce π phase shift as $\gamma = \pi/v_\pi^2$ and we adopt 4.36 V as the typical value of v_π [3], [45]. Fig. 11 shows that $\sim$ 80% structured sparsity can be achieved by our phase-wrapping pruning method, and our pruned trainable transform outperforms the OFFT structure with over 80%

TABLE VI

TRANSFORM SPARSITY ($\mathcal{T}$ SPARSITY) AND POWER CONSUMPTION COMPARISON AMONG OPTICAL FFT AND OUR TRAINABLE TRANSFORM WITH HARDWARE-AWARE PRUNING ON MNIST AND FASHIONMNIST DATASET. $\mathcal{T}$ SPARSITY REPRESENTS HOW MANY COLUMNS OF PSs ARE PRUNED IN OUR TRAINABLE FREQUENCY-DOMAIN TRANSFORMS. THE POWER CONSUMPTION ASSUMES MAXIMUM PARALLELISM ACROSS OUTPUT CHANNELS, THUS, ONE ORIGINAL TRANSFORM AND C_{out} REVERSED TRANSFORMS ARE COUNTED FOR EACH LAYER. FOR THE MNIST DATASET, WE ADOPT THE ONN CONFIGURATION AS 16×16-C16-BN-RELU-MAXPOOL5-F32-RELU-F10, AND FOR THE FASHIONMNIST DATASET WE SET THE ONN CONFIGURATION AS 16×16-C24-BN-RELU-MAXPOOL6-F64-RELU-F10. THE POWER CONSUMPTION IS ESTIMATED BY THE SUM OF PHASE SHIFTS GIVEN THAT THE PHASE SHIFT IS PROPORTIONAL TO THE THERMAL TUNING POWER, I.E., $\phi \propto v^2$. OTHER POWER CONSUMPTION SOURCES, E.G., INSERTION LOSS, ARE NOT CONSIDERED FOR SIMPLICITY

Dataset	Transform	OFFT	Trainable (Pruned)
MNIST [28]	$\mathcal{T}$ Sparsity	0%	88.2%
	Normalized Power	100%	18.4%
FashionMNIST [50]	$\mathcal{T}$ Sparsity	0%	88.4%
	Normalized Power	100%	15.5%

TABLE VII

COMPARISON OF BLOCK SPARSITY, FREQUENCY-DOMAIN TRANSFORM ($\mathcal{T}$) SPARSITY, NORMALIZED POWER CONSUMPTION, AND ESTIMATED AREA (cm^2) AMONG 1) SVD-BASED ONN; 2)$T\Sigma U$-BASED ONN; 3) OPTICAL FFT; 4) OUR TRAINABLE TRANSFORM WITHOUT PRUNING TRANSFORMS; AND 5) OUR TRAINABLE TRANSFORM WITH HARDWARE-AWARE PRUNING ON MNIST DATASET. SVD-BASED AND $T\Sigma U$-BASED ONN CONFIGURATION IS $28 \times 28 - 400 - 10$, AND OURS IS $28 \times 28 - 1024(8) - 10(2)$. ALL ONNs HAVE A SIMILAR INFERENCE ACCURACY WITH A 0.5% ACCURACY DISCREPANCY AMONG ALL ARCHITECTURES. BLOCK SPARSITY IS FOR PRUNED CIRCULANT BLOCKS. $\mathcal{T}$ SPARSITY IS FOR PRUNED TRAINABLE FREQUENCY-DOMAIN TRANSFORMS. THE POWER CONSUMPTION IS NORMALIZED TO SVD-BASED ONN, WHICH IS ESTIMATED BY THE SUM OF ALL PHASE SHIFTS GIVEN THAT THE PHASE SHIFT IS PROPORTIONAL TO THE THERMAL TUNING POWER, I.E., $\phi \propto v^2$

Architecture	Block Sparsity	$\mathcal{T}$ Sparsity	Power	Area (cm^2)
SVD-based [3]	-	-	100%	20.62
$T\Sigma U$-based [20]	-	-	83.1%	17.15
Ours-OFFT [25]	0.40	0.00	98.9%	5.53
Ours-Trainable	0.71	0.00	79.9%	2.54
Ours-Trainable	0.66	0.96	9.9%	2.99

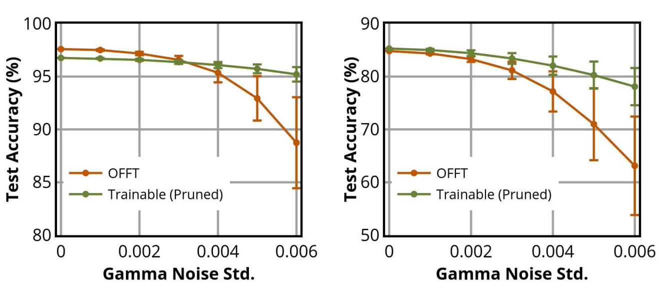


Fig. 11. Robustness comparison among OFFT and pruned trainable transform on MNIST and FashionMNIST dataset. The error bar is drawn to show the $\pm 1\sigma$ accuracy variance from 20 runs. For MNIST dataset, we adopt the ONN configuration as 16×16-C16-BN-ReLU-MaxPool5-F32-ReLU-F10, and for FashionMNIST dataset we set the ONN configuration as 16×16-C24-BN-ReLU-MaxPool6-F64-ReLU-F10.

power reduction and much better robustness under various noise intensities.

We also evaluate the robustness on our circulant-matrix-based MLP architecture. Our FFT-based MLP and trainable transform-based architecture show superior robustness with over 97% accuracy on MNIST due to their structured sparsity and blocking design, while the SVD-based ONN drops below 90% due to severe error accumulation.

VII. CONCLUSION

In this work, we proposed a hardware-efficient ONN architecture. Our proposed ONN architecture leverages block-circulant matrix representation and efficiently realizes matrix-vector multiplication via optical fast Fourier transform, saving $2.2-3.7\times$ area cost compared to prior work. Our proposed two-phase training flow performs structured pruning to our architecture and further improves hardware efficiency with negligible accuracy degradation. We extend the proposed architecture to an optical MD-based frequency-domain CNN, and propose a trainable transform structure to enable a larger design space exploration. We demonstrate structured pruning to our trainable transform structures and it achieves less component usage, over 80% power reduction in CNNs, over 90% power reduction in MLPs, and much better variation-robustness under device-level noises than prior work.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. NIPS*, 2012, pp. 1106–1114.

[2] T. Mikolov *et al.*, “Recurrent neural network based language model,” in *Proc. Interspeech*, 2010, pp. 1045–1048.

[3] Y. Shen *et al.*, “Deep learning with coherent nanophotonic circuits,” *Nat. Photon.*, vol. 11, pp. 441–446, Jun. 2017.

[4] Z. Ying *et al.*, “Electronic-photonic arithmetic logic unit for high-speed computing,” *Nat. Commun.*, vol. 11, p. 2154, May 2020.

[5] C. Feng *et al.*, “Wavelength-division-multiplexing (WDM)-based integrated electronic—Photonic switching network (EPSN) for high-speed data processing and transportation,” *Nanophotonics*, to be published.

[6] C. Feng *et al.*, “Integrated WDM-based optical comparator for high-speed computing,” in *Proc. CLEO*, 2020, pp. 704–706.

[7] M. Miscuglio *et al.*, “Million-channel parallelism Fourier-optic convolutional filter and neural network processor,” in *Proc. CLEO*, 2020, pp. 1–8.

[8] S. K. Esser *et al.*, “Convolutional networks for fast, energy-efficient neuromorphic computing,” *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 41, pp. 11441–11446, 2016.

[9] Y. Wang *et al.*, “Group scissor: Scaling neuromorphic computing design to large neural networks,” in *Proc. DAC*, 2017, pp. 1–6.

[10] Y. Zhang, X. Wang, and E. G. Friedman, “Memristor-based circuit design for multilayer neural networks,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 677–686, Feb. 2018.

[11] A. N. Tait, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, “Broadcast and weight: An integrated network for scalable photonic spike processing,” *J. Lightw. Technol.*, vol. 32, no. 21, pp. 4026–4041, Nov. 1, 2014.

[12] J. Bueno *et al.*, “Reinforcement learning in a large-scale photonic recurrent neural network,” *Optica*, vol. 5, no. 6, p. 756, 2018.

[13] C. Feng, Z. Zhao, Z. Ying, J. Gu, D. Z. Pan, and R. T. Chen, “Compact design of on-chip elman optical recurrent neural network,” in *Proc. CLEO*, 2020, pp. 1–8.

[14] F. Zokaee, Q. Lou, N. Youngblood, W. Liu, Y. Xie, and L. Jiang, “LightBulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks,” in *Proc. DATE*, 2020, pp. 1438–1443.

[15] M. Miscuglio and V. J. Sorger, “Photonic tensor cores for machine learning,” *Appl. Phys. Rev.*, vol. 7, no. 3, 2020, Art. no. 031404.

[16] D. Brunner *et al.*, “Parallel photonic information processing at gigabyte per second data rates using transient states,” *Nat. Commun.*, vol. 3, p. 1364, Jan. 2013.

[17] L. Vivien *et al.*, “Zero-bias 40 Gbit/S Germanium waveguide photodetector on silicon,” *Opt. Exp.*, vol. 20, no. 2, pp. 1096–1101, 2012.

[18] M. Reck, A. Zeilinger, H. Bernstein, and P. Bertani, “Experimental realization of any discrete unitary operator,” *Phys. Rev. Lett.*, vol. 73, no. 1, pp. 58–61, 1994.

[19] A. Ribeiro, A. Ruocco, L. Vanacker, and W. Bogaerts, “Demonstration of a 4×4-port universal linear circuit,” *Optica*, vol. 3, no. 12, p. 1348, 2016.

[20] Z. Zhao *et al.*, “Hardware-software co-design of slimmed optical neural networks,” in *Proc. ASPDAC*, 2019, pp. 705–710.

[21] Z. Li *et al.*, “Efficient recurrent neural networks using structured matrices in FPGAs,” in *Proc. ICLR Workshop*, 2018, p. 238.

[22] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural networks,” in *Proc. NIPS*, 2015, pp. 2–9.

[23] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, “A sparse-group lasso,” *J. Comput. Graph. Stat.*, vol. 22, no. 2, pp. 231–245, 2013.

[24] O. Grandstrand, *Innovation and Intellectual Property Rights*. Oxford, U.K.: Oxford Univ. Press, 2004.

[25] J. Gu *et al.*, “Towards area-efficient optical neural networks: An FFT-based architecture,” in *Proc. ASPDAC*, 2020, pp. 1–3.

[26] L. Zhao, S. Liao, Y. Wang, Z. Li, J. Tang, and B. Yuan, “Theoretical properties for neural networks with weight matrices of low displacement rank,” in *Proc. ICML*, 2017, pp. 4082–4090.

[27] J. Friedman, T. Hastie, and R. Tibshirani, “A note on the group lasso and a sparse group lasso,” 2010. [Online]. Available: arXiv:1001.0736.

[28] Y. LeCun. (1988). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/ exdb/mnist/

[29] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, “Holylight: A nanophotonic accelerator for deep learning in data centers,” in *Proc. DATE*, 2019, pp. 1483–1488.

[30] A. N. Tait *et al.*, “Neuromorphic photonic networks using silicon photonic weight banks,” *Sci. Rep.*, vol. 7, p. 7430, Aug. 2017.

[31] V. Bangari *et al.*, “Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs),” in *Proc. IEEE JSTQE*, 2020, pp. 277–368.

[32] H. Bagherian *et al.*, “On-chip optical convolutional neural networks,” 2018. [Online]. Available: arxiv.abs/1808.03303

[33] S. Xu, J. Wang, R. Wang, J. Chen, and W. Zou, “High-accuracy optical convolution unit architecture for convolutional neural networks by cascaded acousto-optical modulator arrays,” *Opt. Exp.*, vol. 27, no. 14, pp. 19778–19787, 2019.

[34] W. Uijens, “Activating frequencies: Exploring non-linearities in the Fourier domain,” M.S. thesis, School Elect. Eng., Math. Comp. Sci., Delft Univ. Technol., Delft, The Netherlands, 2018.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. CVPR*, 2016, pp. 2818–2826.

[36] R. Zhao, Y. Hu, J. Dotzel, C. D. Sa, and Z. Zhang, “Building efficient deep neural networks with unitary group convolutions,” in *Proc. CVPR*, 2019, pp. 11303–11312.

[37] L. Jing *et al.*, “Tunable efficient unitary neural networks (EUNN) and their application to RNNs,” in *Proc. ICML*, 2017, pp. 1–8.

[38] M. Y.-S. Fang, S. Manipatruni, C. Wierzynski, A. Khosrowshahi, and M. R. DeWeese, “Design of optical neural networks with component imprecisions,” *Opt. Exp.*, vol. 27, no. 10, 2019, Art. no. 14009.

[39] R. Meade *et al.*, “TeraPHY: A high-density electronic-photonic chiplet for optical I/O from a multi-chip module,” in *Proc. IEEE OFC*, 2019, pp. 1–3.

[40] D. T. H. Tan, A. Grieco, and Y. Fainman, “Towards 100 channel dense wavelength division multiplexing with 100 Ghz spacing on silicon,” *Opt. Exp.*, vol. 22, no. 9, pp. 10408–10415, 2014.

[41] C. Feng *et al.*, “Wavelength-division-multiplexing-based electronic-photonic network for high-speed computing,” in *Proc. SPIE Smart Photon. Optoelectron. Integr. Circuits XXII*, 2020, Art. no. 011284.

[42] J. Yu and X. Zhou, “Ultra-high-capacity DWDM transmission system for 100G and beyond,” *IEEE Commun. Mag.*, vol. 48, no. 3, pp. 56–64, Mar. 2010.

[43] T. Dao, A. Gu, M. Eichhorn, A. Rudra, and C. Ré, “Learning fast algorithms for linear transforms using butterfly factorizations,” in *Proc. ICML*, 2019, pp. 1517–1524.

[44] N. C. Harris *et al.*, “Efficient, compact and low loss thermo-optic phase shifter in silicon,” *Opt. Exp.*, vol. 22, pp. 10487–10493, Oct. 2014.

[45] J. Gu, Z. Zhao, C. Feng, H. Zhu, R. T. Chen, and D. Z. Pan, “ROQ: A noise-aware quantization scheme towards robust optical neural networks with low-bit controls,” in *Proc. DATE*, 2020, pp. 1586–1589.

[46] Z. Zhao, J. Gu, Z. Ying, C. Feng, R. T. Chen, and D. Z. Pan, “Design technology for scalable and robust photonic integrated circuits,” in *Proc. ICCAD*, 2019, pp. 1–7.

[47] E. Timurdogan *et al.*, “AIM process design kit (AIMPDKv2.0): Silicon photonics passive and active component libraries on a 300mm wafer,” in *Proc. Opt. Fiber Commun. Conf.*, 2018, pp. 1–3.

[48] Z. Sheng *et al.*, “A compact and low-loss MMI coupler fabricated with CMOS technology,” *IEEE Photon. J.*, vol. 4, no. 6, pp. 2272–2277, Dec. 2012.

[49] Y. Zhang, A. Hosseini, X. Xu, D. Kwong, and R. T. Chen, “Ultralow-loss silicon waveguide crossing using bloch modes in index-engineered cascaded multimode-interference couplers,” *Opt. Lett.*, vol. 38, no. 18, p. 3608, 2013.

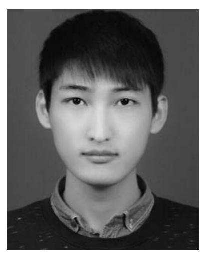
[50] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[51] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. ICCV*, 2015, pp. 1026–1034.

[52] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015, pp. 1–6.

[53] J. Gu, Z. Zhao, C. Feng, W. Li, R. T. Chen, and D. Z. Pan, “FLOPS: Efficient on-chip learning for optical neural networks through stochastic zeroth-order optimization,” in *Proc. DAC*, 2020, pp. 1–4.

[54] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, “Training of photonic neural networks through *in situ* backpropagation and gradient measurement,” *Optica*, vol. 5, no. 7, p. 864, 2018.



Jiaqi Gu (Student Member, IEEE) received the B.E. degree in microelectronic science and engineering from Fudan University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA, under the supervision of Prof. D. Z. Pan.

His current research interests include machine learning, algorithm and architecture design, optical neuromorphic computing for AI acceleration, and GPU acceleration for VLSI physical design automation.

Mr. Gu has received the Best Paper Reward at ASP-DAC’20 and the Best Paper Finalist at DAC’20.

Zheng Zhao received the B.S. degree in automation from Tongji University, Shanghai, China, in 2012, the M.S. degree in electrical and computer engineering from Shanghai Jiao Tong University, Shanghai, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2020.

After her Ph.D. degree, she joined Synopsys Inc., Mountain View, CA, USA, as a Senior Research and Development Engineer II.

Chenghao Feng (Student Member, IEEE) received the B.S. degree in physics from Nanjing University, Nanjing, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA.

His research interests include silicon photonics devices and system design for optical computing and interconnect in integrated photonics.

Zhoufeng Ying (Member, IEEE) received the B.E. and M.E. degrees in optical engineering from Nanjing University, Nanjing, China, in 2014 and 2016, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2020.

After his Ph.D. degree, he joined Alpine Optoelectronics, Fremont, CA, USA, as a Senior Silicon Photonics Designer.

Mingjie Liu (Student Member, IEEE) received the B.S. degree from Peking University, Beijing, China, in 2016, and the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Texas at Austin, Austin, TX, USA. His current research interests include applied machine learning for design automation, and physical design automation for analog and mixed-signal integrated circuits.

Ray T. Chen (Fellow, IEEE) received the B.S. degree in physics from the National Tsing Hua University, Hsinchu, Taiwan, in 1980, and the M.S. degree in physics and the Ph.D. degree in electrical engineering from the University of California at Oakland, Oakland, CA, USA, in 1983 and 1988, respectively. He is the Keys and Joan Curry/Cullen Trust Endowed Chair with the University of Texas at Austin (UT Austin), Austin, TX, USA. He is the Director of the Nanophotonics and Optical Interconnects Research Lab, Microelectronics Research Center. He is also the Director of the AFOSR MURI-Center for Silicon Nanomembrane involving faculty from Stanford University, UIUC University, Rutgers University, and UT Austin. In 1992, he joined UT Austin to start the optical interconnect research program. From 1988 to 1992, he worked as a Research Scientist, a Manager, and the Director of the Department of Electro-Optic Engineering, Physical Optics Corporation, Torrance, CA, USA. From 2000 to 2001, he served as the CTO, the Founder, and the Chairman of the Board of Radiant Research, Inc., where he raised $18 million A-Round funding to commercialize polymer-based photonic devices involving more than 20 patents, which were acquired by Finisar in 2002, a publicly traded company in the Silicon Valley (NASDAQ:FNSR). He also serves as the Founder and the Chairman of the Board of Omega Optics Inc. since its initiation in 2001. Omega Optics has received over $5 million in research funding. His research work has been awarded over 145 research grants and contracts from sponsors, such as Army, Navy, Air Force, DARPA, MDA, NSA, NSF, DOE, EPA, NIST, NIH, NASA, the State of Texas, and private industry. The research topics are focused on four main subjects: 1) nanophotonic passive and active devices for bio- and EM-wave sensing and interconnect applications; 2) thin-film-guided-wave optical interconnection and packaging for 2-D and 3-D laser beam routing and steering; 3) true-time-delay wideband phased array antenna; and 4) 3-D printed microelectronics and photonics. Experiences garnered through these programs are pivotal elements for his research and further commercialization. His group at UT Austin has reported its research findings in more than 970 publications, including over 100 invited papers and 74 patents.

Dr. Chen was the recipient of the 1987 UC Regent's Dissertation Fellowship and the 1999 UT Engineering Foundation Faculty Award, for his contributions in research, teaching, and services. He received the Honorary Citizenship Award in 2003 from the Austin city council for his contribution in community service. He was also the recipient of the 2008 IEEE Teaching Award, the 2010 IEEE HKN Loudest Professor Award, and the 2013 NASA Certified Technical Achievement Award for contribution on moon surveillance conformable phased array antenna. During his undergraduate years at the National Tsing Hua University, he led the 1979 university debate team to the Championship of the Taiwan College-Cup Debate Contest. He has chaired or been a program committee member for more than 130 domestic and international conferences organized by IEEE, SPIE (The International Society of Optical Engineering), OSA, and PSC. He has served as an editor, co-editor, or coauthor for over 20 books. He has also served as a consultant for various federal agencies and private companies and delivered numerous invited talks to professional societies. He is a Fellow of OSA and SPIE.

Dr. Chen has supervised 39 postdocs and graduated 53 PhD students from his research group at UT Austin. Many of them are currently professors in the major research universities in the world.

David Z. Pan (Fellow, IEEE) received the B.S. degree from Peking University, Beijing, China, in 1992, and the M.S. and Ph.D. degrees from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 1998 and 2000, respectively.

From 2000 to 2003, he was a Research Staff Member with IBM T. J. Watson Research Center, Armonk, NY, USA. He is currently the Silicon Laboratories Endowed Chair in Electrical Engineering, Austin, TX, USA. He has published over 380 journal articles and refereed conference papers, and is the holder of eight U.S. patents. He has graduated over 35 Ph.D./postdoc students at UT Austin who are holding key academic and industry positions. His research interests include electronic His research is mainly focused on cross-layer design for manufacturing, reliability, security, machine learning and hardware acceleration, design/CAD for analog/mixed signal designs and emerging technologies.

He has served as a Senior Associate Editor of *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, an Associate Editor of IEEE Design & Test, IEEE Transactions on CAD, IEEE Transactions on VLSI, IEEE Transactions on CAS-I, IEEE Transactions on CAS-II, IEEE CAS Society Newsletter, *Science China Information Sciences*, and *Journal of Computer Science and Technology*. He has served in the Executive and Program Committees of many major conferences, including DAC, ICCAD, ASPDAC, and ISPD. He has served as the General Chair of ICCAD 2019 and ISPD 2008, Program Chair of ICCAD 2018 and ASPDAC 2017, and DAC 2014 Tutorial Chair. He is elected to the ACM/SIGDA Executive Committee in 2018 and serves as the Award Chair.

Dr. Pan has received a number of prestigious awards for his research contributions, including the 2013 SRC Technical Excellence Award, DAC Top 10 Author in Fifth Decade, DAC Prolific Author Award, ASP-DAC Frequently Cited Author Award, 19 Best Paper Awards (ISPD 2020, ASP-DAC 2020, DAC 2019, GLSVLSI 2018, VLSI Integration 2018, HOST 2017, SPIE-AL 2016, ISPD 2014, ICCAD 2013, ASPDAC 2012, ISPD 2011, IBM Research Pat Goldberg Memorial Best Paper Award 2010 in CS/EE/Math, ASPDAC 2010, DATE 2009, ICICDT 2009, SRC Techcon 2015, 2012, 2007 and 1998), Communications of the ACM Research Highlights (2014), ACM/SIGDA Outstanding New Faculty Award (2005), NSF CAREER Award (2007), UCLA Engineering Distinguished Young Alumnus Award (2009), UT Austin RAISE Faculty Excellence Award (2014), IBM Faculty Award four times, SRC Inventor Recognition Award three times, Cadence Academic Collaboration Award (2019), and a number of international CAD contest awards, among others. His students have won many awards, including the First Place of ACM Student Research Competition Grand Finals in 2018, ACM/SIGDA Student Research Competition Gold Medal (twice), ACM Outstanding PhD Dissertation in EDA (twice), EDAA Outstanding Dissertation Award (twice), and so on. He is a Fellow of IEEE and SPIE.