

SCATTER: Algorithm-Circuit Co-Sparse Photonic Accelerator with Thermal-Tolerant, Power-Efficient In-situ Light Redistribution

Ziang Yin¹, Nicholas Gangi², Meng Zhang², Jeff Zhang¹, Rena Huang², Jiaqi Gu^{1†}

¹Arizona State University, ²Rensselaer Polytechnic Institute

†jiaqigu@asu.edu

ABSTRACT

Photonic computing has emerged as a promising solution for accelerating computation-intensive artificial intelligence (AI) workloads. However, limited reconfigurability, high electrical-optical conversion cost, and thermal sensitivity limit the deployment of current optical analog computing engines to support power-restricted, performance-sensitive AI workloads at scale. Sparsity provides a great opportunity for hardware-efficient AI accelerators. However, current dense photonic accelerators fail to fully exploit the power-saving potential of algorithmic sparsity. It requires sparsity-aware hardware specialization with a fundamental re-design of photonic tensor core topology and cross-layer device-circuit-architecture-algorithm co-optimization aware of hardware non-ideality and power bottleneck. To trim down the redundant power consumption while maximizing robustness to thermal variations, we propose SCATTER, a novel algorithm-circuit co-sparse photonic accelerator featuring dynamically reconfigurable signal path via thermal-tolerant, power-efficient *in-situ* light redistribution and power gating. A power-optimized, crosstalk-aware dynamic sparse training framework is introduced to explore row-column structured sparsity and ensure marginal accuracy loss and maximum power efficiency. The extensive evaluation shows that our cross-stacked optimized accelerator SCATTER achieves a 511× area reduction and 12.4× power saving with superior crosstalk tolerance that enables unprecedented circuit layout compactness and on-chip power efficiency. Our code is open sourced at [link](#).

1 INTRODUCTION

The quest for efficient and high-performance artificial intelligence (AI) solutions has propelled the development of photonic computing. By harnessing the unique properties of light, photonic accelerators offer unmatched speed and energy efficiency, particularly for resource-constrained AI applications [3, 6, 7, 19–21, 23, 24, 30]. Photonic tensor cores (PTCs) are the fundamental building blocks of these optical AI accelerators, and various designs have been demonstrated for matrix-vector multiplication or convolution using either coherent interference [7, 20, 24, 29, 30] or incoherent intensity modulation with multi-wavelength accumulation [6, 21, 23].

Despite the ultra-fast speed and high throughput, the widespread adoption of photonic accelerators is hampered by several critical challenges: *thermal robustness*, *non-trivial power bottleneck from signal conversion* between electrical and optical domains, and hardware reconfigurability [11, 18, 26, 28, 29, 31]. **❶ Thermal Variation Robustness.** Thermo-optic devices, chosen for their compactness and low insertion loss, are used for phase/magnitude modulation but are prone to thermal crosstalk, which significantly degrades computational accuracy. Solutions often compromise chip density, require noise modeling [11, 18, 31], or rely on on-chip calibration [10, 12, 16], which induces hardware overhead and specific to individual chips. A more general, thermal variation-tolerant architecture supporting standard model compression is needed. **❷ E-O/A-D Conversion**

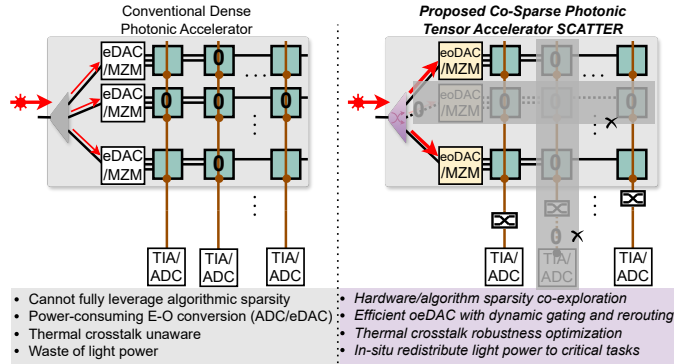


Figure 1: Our proposed SCATTER architecture co-explores circuit/algorithm sparsity with power efficiency and robustness co-optimization compared to generic dense tensor cores.

Power. In electronic-photonic heterogeneous accelerators, signal conversion between optical and electrical domains is a significant power bottleneck [19, 29, 32]. High-speed, high-resolution digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) dominate on-chip power and area. Researchers have tried reducing power by lowering sampling frequency (<1 GHz) and bit resolutions (<4-bit) with minimal accuracy loss [11, 26, 32]. However, achieving a balance between *resolution*, *speed*, and *low power/area cost* remains challenging. **❸ Reconfigurability.** Reprogrammability is crucial for versatile photonic accelerators. Current PTC designs are specialized for dense MVM with fixed circuit topologies, lacking flexibility to exploit sparsity in modern AI models. An efficient co-sparse architecture should optimize more than just setting weights to zero. A reconfigurable PTC that adaptively reroutes and gates its signal path to support algorithmic sparsity would significantly enhance flexibility and efficiency.

To address these fundamental roadblocks, *for the first time*, we present a dynamically reconfigurable photonic accelerator SCATTER that features native support for algorithm/hardware co-sparsity with cross-layer power/thermal robustness co-optimization. **❶ To boost the thermal variation robustness**, we optimize device spacing, exploit circuit sparsity, and employ *in-situ* power gating to minimize crosstalk. **❷ To boost power efficiency**, we explore power-optimized photonic devices, hybrid electrical-optical DAC designs, and architectural hardware sharing. Our power-aware dynamic sparse training framework explicitly targets power minimization during sparsity exploration. **❸ To enable flexible hardware reconfiguration**, we introduce an on-chip tunable light rerouter to dynamically redistribute the optical power to efficiently support structured row-column weight sparsity that directly translates to power reduction and noise suppression.

The major contributions of this paper are as follows:

- **In-situ Light Redistribution** – *For the first time*, we introduce *in-situ* light redistribution mechanism for reconfigurable photonic tensor cores, achieving power-optimized, thermal-robust algorithm-circuit co-sparsity. We dynamically reroute light power to focus on critical computations and suppress variation-induced

errors from pruned components, enhancing efficiency and signal-to-noise ratio.

- **Dynamic Reconfigurable Architecture** – We introduce on-chip optical rerouter input/output power gating to enable fine-grained signal path control as a native primitive for circuit sparsity, achieving superior efficiency for multi-core photonic AI accelerators.
- **Cross-layer Power/Area Minimization** – We integrate power-optimized photonic devices, hybrid electrical-optical DAC designs, dynamic power-gated input/readout circuitry, and automated architecture exploration, realizing 511× area reduction and 12.4× power saving compared to dense designs built on foundry devices.
- **Power-Robustness Co-Optimized Sparsity** – Our one-shot, hardware-aware dynamic sparse training learns structured weight sparsity masks while optimizing accuracy, power efficiency, and thermal crosstalk tolerance.

2 BACKGROUND

2.1 Dense/Sparse Optical Neural Networks

Various photonic neural network designs encode inputs and weights to light magnitude/phase and circuit transmission, performing ultra-fast matrix multiplication [3, 6, 7, 19–21, 23, 24, 29, 30]. However, most prior work focuses on dense photonic tensor cores (PTCs) with fixed topologies, limiting compatibility with the algorithmic sparsity in modern AI models. Some pruning techniques for optical neural networks have shown power reduction by pruning phase shifters in MZI arrays [1, 8, 9], but they fail to fully leverage structured sparsity. The challenge remains to dynamically reconfigure circuit connectivity and signal paths for co-exploration of algorithmic and circuit sparsity, optimizing power and robustness.

2.2 Structured Sparsity and Dynamic Sparse Training

Modern neural networks often exhibit intrinsic sparsity, offering opportunities for memory and computational savings through pruning [4, 22, 25, 27]. Unlike unstructured sparsity with arbitrary zero entries in the matrix, structured sparsity, where clusters of elements are pruned in hardware-aware patterns, is particularly advantageous for efficient implementation [4, 17, 22].

To automate structured sparsity exploration while minimizing accuracy loss, we adopt state-of-the-art dynamic sparse training (DST) [2, 14, 15]. Unlike traditional approaches that first train a dense model, DST maintains a sparse model throughout training, iteratively pruning and regrowing connections. This one-shot approach streamlines the neural architecture search process. We will develop our power/robustness sparsity optimization based on the flexible DST framework.

3 PROPOSED CO-SPARSE PHOTONIC ACCELERATOR SCATTER

We introduce SCATTER, a multi-core dynamic photonic accelerator, shown in Fig. 2. SCATTER is designed to overcome the limitations of traditional photonic accelerators with these key features: ❶ phase-agnostic incoherent photonic tensor cores for robust tensor computing; ❷ shared input modulation modules and readout circuitry to balance area, power, and control flexibility; ❸ in-situ tunable rerouter for light redistribution; ❹ hardware gating to support structured row-column sparsity with enhanced power efficiency and thermal crosstalk robustness; ❺ co-optimized devices, circuits, and architecture configurations with maximum efficiency and thermal variation tolerance.

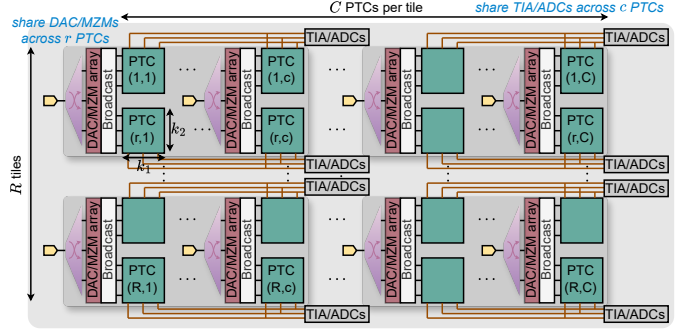


Figure 2: Dynamic multi-core photonic accelerator architecture with R tiles and C PTCs per tile. Each PTC is of size $k_1 \times k_2$. Input modulation modules are shared by r PTCs across different tiles. Readout circuitry is shared by c PTCs in a tile.

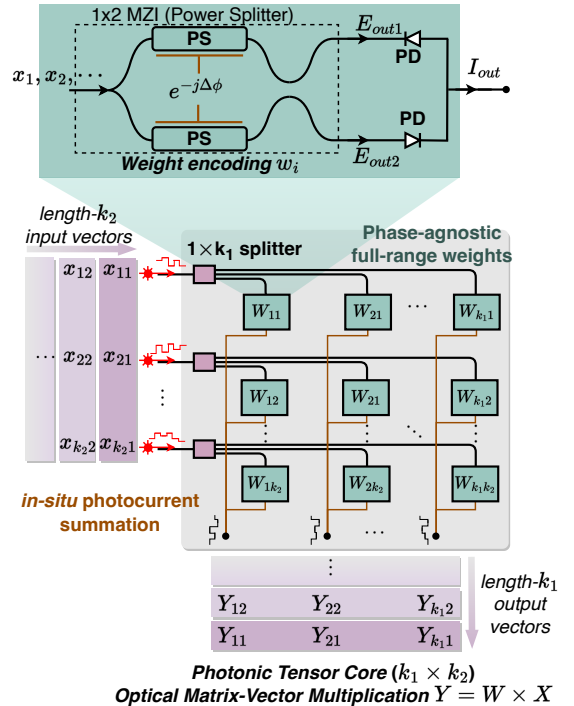


Figure 3: Schematic of phase-agnostic incoherent PTC.

In this section, we will detail the core innovations of SCATTER’s hardware/algorithm co-sparse design and our comprehensive cross-layer co-optimization methodology.

3.1 Accelerator Architecture Overview

3.1.1 Phase-Insensitive Differential Photonic Tensor Cores. To avoid the phase instability issue of coherent PTCs, e.g., unitary MZI mesh [20], and dynamic crossbar arrays [29], and thermal sensitivity issue of narrow-band resonance-based PTCs, e.g., MRR weight banks [21], we introduce a phase-agnostic full-range PTC architecture. This architecture forms the foundation of our sparsity optimization techniques. Figure 3 illustrates a $k_1 \times k_2$ PTC. The length- k_2 input vector x is encoded as light intensity and broadcast to k_1 columns via $1 \times k_1$ even splitter. Each crossbar node is a full-range multiplication engine consisting of a 1×2 MZI power splitter and balanced photodetectors (BPD).

Partial products are accumulated along each column through photocurrent aggregation. Formally, the tensor core operation is

$$y = Wx; y_i = \sum_j W_{ij}x_j;$$

$$\begin{pmatrix} E_{out1} \\ E_{out2} \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix} \begin{pmatrix} e^{-j\Delta\phi} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} x \quad (1)$$

$$W_{ij}x = |E_{out1}|^2 - |E_{out2}|^2 = \left(2 \cos^2 \left(\frac{\Delta\phi + \phi_b}{2}\right) - 1\right)x,$$

where $\Delta\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the MZI arm phase difference. The default phase bias ϕ_b is $\frac{\pi}{2}$. Differential outputs from the power splitter and BPDs enable full-range weight representation. For input vectors, since they are quantized within a certain range, we can adopt non-negative isomorphic transformation before deployment to guarantee positive-only input x with a certain bias [13]. The weight matrices and input vectors are normalized to ensure they are implementable by the modulation coefficient and light intensity, and the output results are scaled back with the normalization factor. Note that neither phase coherence nor thermal feedback, like in MZI and MRR arrays, is needed due to intensity encoding and broadband non-resonance devices.

3.2 Power, Area, and Robustness Analysis

We present a thorough power analysis of photonic computing engines, which will guide our optimization strategies. We assume the multi-core architecture has R tiles and C cores per tile, operating at frequency f . The input modulation module is shared across r PTCs across different tiles, and the readout circuitry is shared across c cores within a tile.

3.2.1 On-chip Power Modeling. We break down the key contributors to on-chip power consumption:

For input modulation, high-speed b_{in} -bit DACs consume a large amount of power. The input modulation power is estimated as

$$P_{in} = \frac{RCk_2}{r} (P_{mod} + P_{eDAC}(b_{in}, f)), \quad (2)$$

$$P_{mod} = P_{mod,static} + E_{mod}f, \quad P_{eDAC}(b_{in}, f) = P_{0,eDAC} \frac{2^{b_{in}}}{b_{in} + 1} f,$$

where $P_{mod,static}$ is the static power of the MZM, E_{mod} is the dynamic energy per full-range modulation (J/bit), $P_{0,eDAC}$ is the reported eDAC power working at its designed sampling rate and precision. Note that the eDAC power scales linearly with frequency f and exponentially with resolution b_{in} . Reducing the eDAC power is crucial for system energy efficiency.

For weight encoding, phase shifters in MZI splitters, low-speed ($f_w \ll f$) b_w -bit weight DACs, and BPDs contribute to the weight encoding power:

$$P_{wgt} = RCk_1k_2(P_{MZI} + 2P_{PD}), \quad P_{PS}^{ij} = \mathcal{P}(|\Delta\phi|, l_s), \quad (3)$$

where the MZI static power dominates. This power is a function of the absolute phase difference $|\Delta\phi| = |\Delta\phi^{up} - \Delta\phi^{lo}|$ and MZI arm spacing l_s . The simulated power function $\mathcal{P}(\cdot)$ is shown in Fig. 4(c).

The readout circuitry also consumes significant power, especially for the high-speed ADCs. The total readout power is,

$$P_{out} = \frac{RCk_1}{c} (P_{TIA} + P_{ADC}(b_o, f)), \quad P_{ADC}(b_o, f) = P_{0,ADC} \cdot b_o f, \quad (4)$$

where the ADC power dominates the readout power, which scales linearly with output precision and sampling frequency. Hence, the total on-chip power is $P = P_{in} + P_{wgt} + P_{out}$. Note that off-chip laser and low-speed weight DACs are not included in this model.

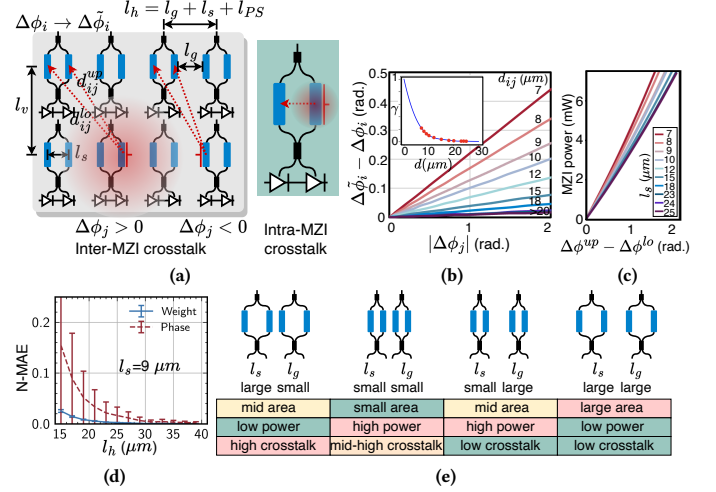


Figure 4: (a) Inter- and intra-MZI thermal crosstalk are modeled by distance-related coupling coefficients γ . (b) Lumerical HEAT simulation is used to sweep various phase shifter spacings and fit a numerical crosstalk model. (c) Larger arm spacing l_s reduces the required MZI power to realize the same phase difference. (d) Larger MZI spacing l_h reduces normalized mean-absolute error (N-MAE) on phases and weights. (e) Impact of arm spacing and MZI spacing on area, power, and crosstalk.

3.2.2 Area Modeling. Each crossbar node area is

$$A_{node} = (l_s + w_{PS}) \times (l_Y + l_{PS} + l_{DC}), \quad (5)$$

based on our designed phase shifter and layout, $(l_Y + l_{PS} + l_{DC}) = 115\mu\text{m}$ and $w_{PS} = 6\mu\text{m}$. The $k_1 \times k_2$ array area is

$$A_{PTC,wgt} = ((k_2 - 1)l_o + l_Y + l_{PS} + l_{DC}) \times ((k_1 - 1)l_h + l_s + w_{PS}). \quad (6)$$

Considering the multi-core architecture with an input modulation sharing factor r and readout sharing factor c , the total on-chip area is estimated as

$$A = RC(A_{PTC,wgt} + k_2A_{MMI} + 2k_1k_2A_{PD}) + \frac{RC}{r}(k_2A_{DAC} + k_2A_{MZM} + A_{rerouter}) + \frac{RC}{c}(k_1A_{ADC} + k_1A_{TIA}), \quad (7)$$

where the PD arrays are placed in separate regions to avoid thermal noises, A_{MMI} is the $1 \times k_1$ MMI splitter, $A_{rerouter}$ is the area corresponding to the compact folded layout shown in Fig. 5. Note that laser and weight DACs are off-chip and are not included here.

3.2.3 Thermal Crosstalk Modeling. Thermo-optic MZI power splitters experience intra-MZI and inter-MZI thermal crosstalk, which leads to power penalty and accuracy degradation. In a k_2 -row and k_1 -column PTC, we model the impact from other MZIs to the i -th MZI as follows,

$$\Delta\tilde{\phi}_i = \Delta\phi_i + \sum_{j \neq i}^{k_1 k_2} \Delta\gamma_{ij} |\Delta\phi_j| = \Delta\phi_i + \sum_{j \neq i}^{k_1 k_2} (\gamma_{ij}(d_{ij}^{up}) - \gamma_{ij}(d_{ij}^{lo})) |\Delta\phi_j|, \quad (8)$$

where $\Delta\phi_i$ is the target phase shift, and $\Delta\gamma_{ij}$ is the crosstalk coefficient between the i -th and j -th MZIs. This coefficient accounts for the differential working mode and depends on the distance between the aggressor and victim phase shifters. γ_{ij} is a function of center distance d_{ij} between the aggressor phase shifter in the j -th MZI and the victim phase shifter in the i -th MZI. d_{ij}^{up} and d_{ij}^{lo} represent the distance w.r.t the upper and lower arm of the victim MZI, respectively.

This distance is calculated dynamically based on the sign of the aggressor's phase shift $\Delta\phi_j$. It will heat up the upper arm to realize a positive $\Delta\phi \in [0, \pi/2]$ and heat up the lower arm to create a negative

$\Delta\phi \in [0, -\pi/2]$. We formulate the phase-dependent distance as

$$\begin{aligned} d_{ij}^{up} &= \sqrt{[(R(j) - R(i))l_v]^2 + [(C(j) - C(i))h - l_s \mathbf{I}_{\Delta\phi_j < 0}]^2}, \\ d_{ij}^{lo} &= \sqrt{[(R(j) - R(i))l_v]^2 + [(C(j) - C(i))h + l_s \mathbf{I}_{\Delta\phi_j \geq 0}]^2}, \end{aligned} \quad (9)$$

where the indicator function $\mathbf{I}_{\Delta\phi_j < 0}$ is 1 with negative $\Delta\phi_j$, and $C(\cdot)$ and $R(\cdot)$ are the column and row index of j -th MZI.

We use Lumerical HEAT and MODE simulations to characterize the relationship between γ and d , shown in Fig. 4(a). Thermal profiles were imported into MODE to determine the effective indices and $\Delta\phi$ of the upper and lower arms based on the thermo-optic coefficient of silicon. With the same spacing, the crosstalk factor $\gamma \propto \frac{\Delta\phi_i}{\Delta\phi_j}$ is constant, which indicates that γ is only a function of spacing. The crosstalk coefficient decays exponentially with increasing distance. We fit this relationship with a piecewise function (5th-order polynomial and exponential function),

$$\gamma(d) = \left(\sum_{i=0}^5 p_i d^i \right) \mathbf{I}_{d < 23} + a_0 e^{-a_1 d} \mathbf{I}_{d \geq 23}, \quad (10)$$

where the coefficients are $[p_0, \dots, p_5] = [1, -1.76e-1, 9.9e-3, -8.30e-6, -1.56e-5, 3.55e-7]$, $[a_0, a_1] = [0.217, 0.127]$. The curve fitting fidelity R^2 are 0.999 for the polynomial part and 0.998 for the exponential part.

3.3 Power, Area, and Robustness Co-Optimization

We introduce synergistic optimization approaches across device, circuit, architecture, and algorithm levels, guided by our in-depth efficiency and robustness analysis.

3.3.1 Device-Level: Power-Efficient Footprint-Compact MZI. Foundry-provided MZI switch (Foundry-MZI) consumes $P_\pi = 30$ mW for π phase shift and a large footprint consumption ($550 \mu\text{m}$ in length). We design a low-power MZI switch (LP-MZI) with compact size ($115 \mu\text{m}$ in length) and 50% lower power $P_\pi = 15$ mW. The phase shifter width w_{PS} and phase shifter spacing l_s between two arms impact the switching power and device footprint. The phase difference between the upper and lower arm of an MZI is $\Delta\phi^{up} - \Delta\phi^{lo}$. If the upper arm is heated up, the intra-MZI crosstalk will increase $\Delta\phi^{lo}$ and thus diminish the phase difference. It results in a power penalty required to realize the same $\Delta\phi$. We show how MZI power P_{MZI} changes with arm spacing and phase difference between arms in Fig. 4(c). Later, we will explore optimal settings to balance power and area and show our efficiency advantage over standard foundry devices.

Figure 6 illustrates the power-area-accuracy design space of a 16×16 PTC. We carefully select device spacing configurations (l_s, l_g) to balance power, area, and accuracy. For a dense PTC, to meet the accuracy target (e.g., <1% drop), we set $l_s = 9 \mu\text{m}$, $l_g = 5 \mu\text{m}$ by minimizing the power-area product. Importantly, we'll demonstrate how sparsity techniques can relax these design constraints and allow for an even more compact layout, leading to improvements in both power and area efficiency.

3.3.2 Circuit-Level: In-situ Tunable Light Redistribution for Column Sparsity. High-speed DACs and drivers for input modulation consume significant power. To maximize efficiency, we strategically zero out length- rk_1 column vectors in the $rk_1 \times ck_2$ weight chunk, enabling us to shut down the weight MZIs on those pruned rows. However, due to non-idealities (e.g., phase bias deviation, crosstalk, and noises), simply removing power from pruned weight MZIs can still lead to non-zero weights, which induces computing errors.

We propose a dynamically reconfigurable sparse tensor core with in-situ light redistribution. This is the key to fully leveraging sparsity

benefits. To illustrate this, we first express the vector dot-product result with crosstalk and noise as:

$$y = \sum_j^{k_2} (\tilde{w}_j x_j + \delta n_{PD}), \quad (11)$$

where δn_{PD} is random photocurrent noises from PDs (we set it to 0.01), and \tilde{w} is the weight under crosstalk. Given a weight column sparsity mask $m^c = [m_1^c, \dots, m_{k_2}^c] \in \{0, 1\}^{k_2}$, we assume there are k_2' nonzero elements in m^c .

Based on the above assumption, we compare two conventional design approaches and highlight our superiority with *in-situ* light redistribution technique in Fig. 5.

Weight Pruning Only. In Fig. 5(Left), the input light is evenly split via a balanced splitter tree without shutting down modulators. Though weight MZI power is removed, the DAC/MZM power is wasted. More importantly, pruned paths still contribute to the final photocurrent, leading to leakage errors:

$$y_{\text{prune}} = \sum_{j, m_j^c=1}^{k_2} (\tilde{w}_j x_j) + \sum_{j, m_j^c=0}^{k_2} (\delta w_j \cdot x_j) + \sum_j^{k_2} \delta n_{PD}, \quad (12)$$

where δw is the error due to non-idealities, e.g., weight MZI crosstalk, random phase noises, and limited extinction ratio (ER) of MZIs, defined as the ratio between maximum and minimum transmission.

Weight Pruning + Input Gating (IG). In Fig. 5(Middle), the power supply of the high-speed DACs and MZMs for pruned ports are gated. While this saves some power, light still leaks through the high-speed MZMs (due to a limited extinction ratio). Further, light power on pruned paths is completely wasted without contributing to useful computation. The dot-product result is as follows

$$y_{\text{IG}} = \sum_{j, m_j^c=1}^{k_2} (\tilde{w}_j x_j) + \sum_{j, m_j^c=0}^{k_2} (\delta w_j \cdot \delta x_j) + \sum_j^{k_2} \delta n_{PD}, \quad (13)$$

where δw and δx are nonzero errors due to non-ideal variations. Note that column pruning with input MZM gating has no reduction in the PD noises and still suffers from leakage errors, as in the second term.

Pruning + Input Gating + Light Redistribution (LR). As shown in Fig. 5(Right), our proposed solution upgrades the passive even splitter tree to an *in-situ* tunable light rerouter to dynamically redistribute light power from unused ports to active ports.

This boosts the intensity on active ports by a factor of k_2/k_2' . The TIA gain will be reduced by a ratio of k_2'/k_2 to recover the same range. Then, the result becomes

$$y_{\text{IG+LR}} = \frac{k_2'}{k_2} \left(\sum_{j, m_j^c=1}^{k_2} \left(\frac{k_2}{k_2'} \tilde{w}_j x_j \right) + \sum_j^{k_2} \delta n_{PD} \right) = \sum_{j, m_j^c=1}^{k_2} \tilde{w}_j x_j + \frac{k_2'}{k_2} \sum_j^{k_2} \delta n_{PD}. \quad (14)$$

Light redistribution has two main advantages: it eliminates the leakage errors on pruned ports and effectively reduces the photocurrent noise by a factor of $\frac{k_2'}{k_2}$. For example, with a 20% column sparsity, light redistribution will have a 7 dB higher SNR. Optical power redistribution through light path reconfiguration can be realized by cascading a number of MZI power splitters. Later, our power-efficient sparse training algorithm will find the optimal column sparsity mask m^c that minimizes the power consumption of the rerouter given a defined column sparsity.

3.3.3 Circuit-Level: On-chip TIA/ADC Gating for Row Sparsity. To maximize power savings from sparsity, we enable dynamic TIA/ADC gating in Fig. 7. Our architecture accumulates the photocurrent from c PTCs per tile as analog-domain partial product summation and

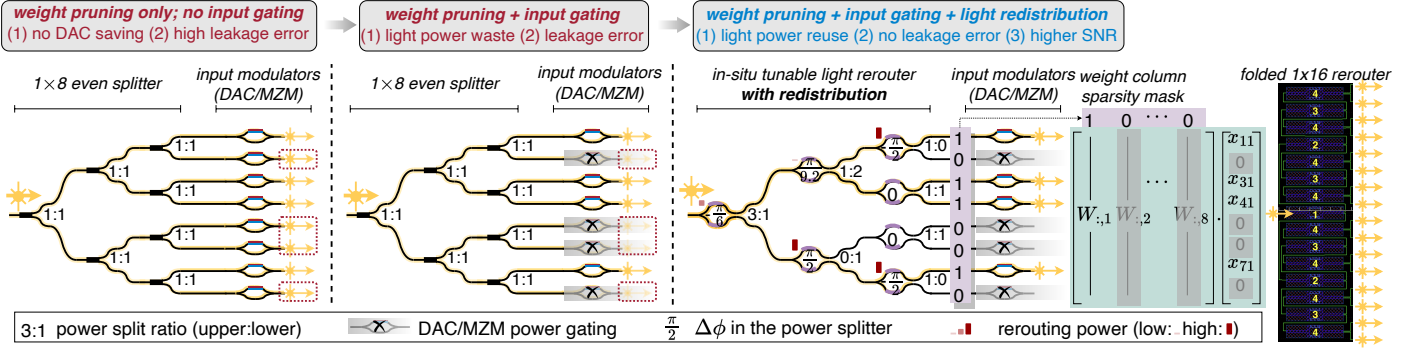


Figure 5: Weight block column-wise sparsity can be supported by on-chip light rerouter with in-situ tunable light splitting ratios. Here, we show an 8×8 block as an example. Input gating helps save significant high-speed DAC and input modulation power while reducing leakage error in pruned paths. Light redistribution eliminates leakage errors and provides light power to unpruned computing engines with higher optical SNR. Different from the tree structure in the schematic, a folded rerouter layout is designed to save area. Refocusing can effectively reduce computing N-MAE errors compared to standard weight pruning.

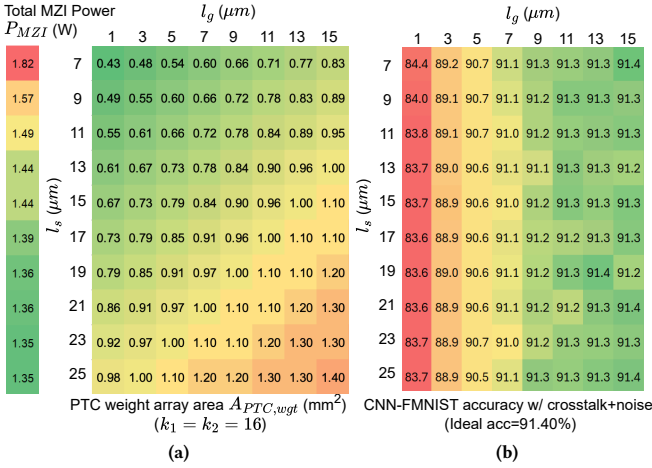


Figure 6: (a) Area and power of weight MZI array ($k_1 = k_2 = 16$) with different arm spacing l_s and MZI gap l_g . (b) Accuracy under variations on CNN FashionMNIST with different spacing.

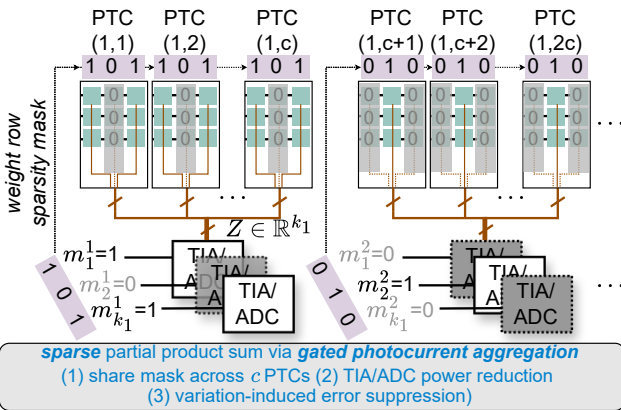


Figure 7: Weight block matrix row-wise sparsity with output TIA/ADC gating (OG).

shares the same TIA/ADC array among them. To fully exploit the benefits of power gating, we focus on coarse-grained structured sparsity where entire length- ck_2 row vectors in the $rk_1 \times ck_2$ weight chunk

are pruned. This allows us to shut down corresponding TIA/ADC for energy reduction and eliminate any leakage, crosstalk, and PD noises.

3.3.4 Circuit-Level: Efficient Hybrid Electronic-Optic DAC. High-speed DACs are a major power bottleneck and often limit resolution and signal-to-noise ratio (SNR). As shown in Fig. 8, generating 6-bit PAM signals at 5 GHz with a single electronic DAC (eDAC) incurs a high power cost $\frac{64}{7}P_{0,eDAC}$ and suffers from low SNR due to overlapped symbols. We introduce a hybrid electronic-optic DAC (eoDAC) that realizes weighted signal modulation both in the electrical and optical domains. The modulation actuators are partitioned into nonuniform segments, and each can be modulated with a low-bit eDAC. There exist fundamental trade-offs among eDAC area, number of IO pads, eDAC power, SNR, and manufacturability. Figure 8 shows different hybrid eoDAC settings. We find that an optimal design partitions the phase shifter into two segments (with an 8:1 length ratio) controlled by 3-bit eDACs. It can approximately realize a 6-bit PAM signal via two 3-bit modulators, e.g., 010001 = $2^3 \cdot (010) + 001$. The length ratio can be customized based on the actual MZM response. This setting requires twice the independent IO pads but saves 2.3× DAC power with significant SNR improvement. Further partitioning (e.g., pure optical DAC) offers negligible power benefits while increasing area, layout, and manufacturing complexity.

3.3.5 Algorithm-Level: Power/Crosstalk-Aware Dynamic Sparse Training. We adapt the SoTA DST algorithm to automatically select the structured sparsity patterns aware of accuracy, power, and crosstalk. Unlike conventional pruning methods, which start from a dense pre-trained model, we initialize a sparse model and dynamically explore sparsity patterns, balancing accuracy, power, and robustness.

Crosstalk/Power-Minimized Initialization. Assume a model has L convolutional (CONV) layers, and sparsity is not applied to the first CONV layer and the last linear layer. For the l -th CONV layer, the weight matrix is of size $W^l \in \mathbb{R}^{C_o C_i K K}$. After $im2col$, the unfolded weight matrix will be padded and partitioned into a 6-D tensor $W^l \in \mathbb{R}^{p \times q \times r \times c \times k_1 \times k_2}$, where $p = \lceil \frac{C_o}{rk_1} \rceil$ and $q = \lceil \frac{C_i K^2}{ck_2} \rceil$. With a given sparsity s (percentage of nonzero elements), we need to first assign the layer-wise sparsity (s^1, \dots, s^L) to match the target sparsity and then initialize the sparsity mask for the l -th layer $m^l = (m^{(l,c)}, m^{(l,r)})$, which contains a column mask $m^c \in \{0, 1\}^{p \times q \times r \times 1 \times k_1 \times 1}$ and a row mask $m^r \in \{0, 1\}^{p \times q \times 1 \times c \times 1 \times k_2}$.

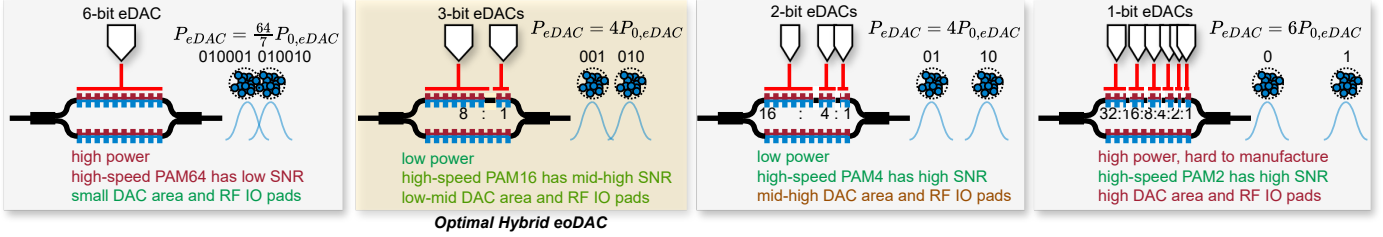


Figure 8: Hybrid electrical-optical DAC (eoDAC) with optimal settings gives the best power, area, manufacturability, and SNR.

For simplicity, we assign the same sparsity for all layers. Better strategies in the literature can be applied. With a target sparsity s_l , we assume the same sparsity pattern for each $k_1 \times k_2$ weight block.

Since the horizontal spacing $l_h < 20 \mu\text{m}$ is much smaller than the vertical spacing $l_v = 120 \mu\text{m}$, crosstalk primarily occurs between weights in the same column vector. We initialize the row mask m^r with an interleaved pattern (e.g., 101010...) to minimize crosstalk based on the guidance from Fig. 9(a). The row mask requires up to 50% row sparsity, which contains the maximum 0's needed to eliminate crosstalk. If the target sparsity is high ($s > 0.5$), we allocate all sparsity to the row mask for maximum crosstalk reduction. E.g., $s^r = 0.75$ and $rk_1 = 8$ lead to a row mask of 11111010. If $s < 0.5$, we initialize column mask m^c with a column sparsity of $s^c = s/s^r$ followed by power optimization. For each $rk_1 \times ck_2$ block, among ck_2 column vectors, we find the best group of $[ck_2 s^c]$ vectors among all $\binom{ck_2}{[ck_2 s^c]}$ combinations with the lowest power.

How to Calculate Power Metric for a Mask? Power P is estimated based on Section 3.2.1. In SCATTER, the sparsity mask indicates power gating on corresponding weight MZIs, input modulation modules, and read-out TIA/ADC arrays to save power and reduce leakage noises. The power of the light rerouter can be calculated by the power splitting ratio derived from the column mask. E.g., with a mask $m^c = 10110010$, the root splitter's ratio is $up:lo=(1+0+1+1)/(0+0+1+0)=3:1$, and the phase is $\Delta\phi = 2\arccos(\sqrt{\frac{up}{lo+up}}) - \phi_b$. If $up+lo=0$, we set $\Delta\phi = 0$. Then, its power can be obtained by using the $\mathcal{P}(|\Delta\phi|, l_s)$ function. Similarly, the weight MZIs' power is dynamically calculated with the actual weights and defined arm spacing.

Power-Aware Pruning Procedure. For simplicity, we only discuss one layer and remove the layer index l in all notations. As shown in Alg. 1, for every ΔT step, if $s^c < 1$, we fix the row mask and explore column sparsity pattern with one pruning and one growth stage. We define a death rate α that controls the percentage of unpruned weights to be pruned in the current step. We apply step-wise CosineDecay to death rate $\alpha^{t+1} \leftarrow \frac{\alpha^t}{2} (1 + \cos(\frac{t\pi}{T_{end}}))$ for stable exploration and convergence, where T_{end} is 80% of total training steps. The remaining 20% of the training steps will keep the same mask and resume accuracy.

The number of weights to be pruned is $D = \alpha \sum(m^r \odot m^c)$. The mask update procedure contains the following stages: ❶ **Determine number of columns to prune.** Since the row sparsity pattern is fixed and identical for all blocks, we can calculate the number of column vectors to prune as $n_c = D / (\sum m^r / (pq))$ to match the scheduled death rate. ❷ **Select small-magnitude column vector candidates.** Unpruned column vectors are sorted based on their ℓ_2 -norm, $\|w\|_2$. The smallest $n_c + \Delta m$ vectors form a pruning candidate pool. Δm is the selection margin (e.g., set to 2) to leave space for power optimization. ❸ **Select low-power column vector candidates.** Among $n_c + \Delta m$ candidate vectors, we enumerate all $\binom{n_c + \Delta m}{n_c}$ combinations (up to a maximum combination in case there are too many candidates) and

Algorithm 1 Power/Crosstalk-Aware Dynamic Sparse Training

Input: Loss function \mathcal{L} , neural network $f_{bin}(\cdot)$ with b_{in} -bit activation quantization, weight bitwidth b_w , input data X , target y , learning rate η , total steps T , steps per epoch ΔT , epoch to stop prune&grow T_{end} , pruning margin Δm , initial death rate α^0 .

Output: Converged parameter W and sparsity mask m ;

- 1: ---Crosstalk/Power-Minimized Initialization---
- 2: $s^r = \max(s, 0.5)$, $m^{(l,r)} = \text{InterleavedOnes}(s^r)$ \triangleright Min-crosstalk m^r
- 3: $s^c = s/s^r$, $m^{(l,c)} = \text{argmin}_{m^{(l,c)}} P(m^{(l,c)})$ \triangleright Min-power m^c
- 4: **for** $t \leftarrow 1 \dots T$ **do**
- 5: $W^l \leftarrow W^l \odot m^{(l,r)} \odot m^{(l,c)}$ \triangleright Inplace apply sparsity mask
- 6: $W^l \leftarrow W^l - \eta \nabla_{W^l} \mathcal{L}(f_{bin}(Q_{b_w}(W), X), y)$
- 7: **if** $t \bmod \Delta T == 0$ and $t < T_{end}$ **then**
- 8: $\alpha = \frac{\alpha^0}{2} (1 + \cos(\frac{t\pi}{T_{end}}))$ \triangleright Schedule death rate
- 9: ---Stage 1: Update Sparsity Mask with Pruning---
- 10: $D^l = [\alpha \sum(m^{(l,r)} \odot m^{(l,c)})]$
- 11: $n_c^l = D^l / (\sum m^{(l,r)} / (p^l q^l))$
- 12: Select $(n_c^l + \Delta m)$ column vectors with smallest ℓ_2 -norm.
- 13: Further select the lowest power and least crosstalk column vectors.
- 14: $m^{(l,c)} \leftarrow m^{(l,c)} \& m_{\text{death}}^{(l,c)}$
- 15: ---Stage 2: Update Sparsity Mask with Growth---
- 16: $n_c^l = (s^l p^l q^l r c k_1 k_2 - \sum(m^{(l,r)} \odot m^{(l,c)})) / (\sum(m^{(l,r)} / (p^l q^l)))$
- 17: $m^{(l,c)} \leftarrow m^{(l,c)} \mid m_{\text{grow}}^{(l,c)}$ \triangleright Similar procedure to select column vectors with large gradient norm and lowest power to grow

select the combination that minimizes the overall power consumption P . We find the death mask m_{death}^c where 1 represents newly pruned columns and update the column mask $m^c \leftarrow m^c \& m_{\text{death}}^c$.

Power-Aware Growth Procedure. To maintain sparsity while exploring patterns, we grow (resume) roughly the same number of weights that were pruned. The number of column vectors to be resumed n_c is calculated based on the target sparsity s and the number of nonzero elements per column, $n_c = (spqrck_1k_2 - \sum(m^r \odot m^c)) / (\sum(m^r / (pq)))$. The column vector selection procedure is based on gradient magnitude for accuracy, i.e., $\|\frac{\partial \mathcal{L}}{\partial w}\|_2$. The same power minimization procedure applies to resume low-power column vectors. At the end of the growth stage, we obtain a growth column mask m_{grow}^c , where 1 represents resumed columns. We then update the sparsity mask, i.e., $m^c \leftarrow m^c \mid m_{\text{grow}}^c$.

4 EXPERIMENTAL RESULTS

4.1 Experiment Setup

Dataset and Models. We evaluate our method on a three-layer CNN (C64K3-C64K3-C64K3-Pool5-FC10) on Fashion-MNIST, VGG-8 on CIFAR-10, and ResNet-18 CIFAR-100 for image classification.

Training Settings. We pre-train CNN for 50 epochs with an Adam optimizer with a 2E-3 learning rate (lr), a cosine decay scheduler, 1E-4 weight decay, and data augmentation (random crop and flip)

Table 1: Optimal device spacing on a dense network ($s=1$) with high accuracy under crosstalk and noises ($\sim 1\%$ drop than ideal accuracy 91.4%) and minimum power-area product (PAP). Average power P_{avg} is evaluated on CNN-FashionMNIST For a dense accelerator, the optimal settings are $l_s = 9\mu\text{m}$ and $l_g = 5\mu\text{m}$.

l_s (μm)	l_g (μm)	Acc (%) \uparrow	P_{avg} (W) \downarrow	A (mm^2) \downarrow	PAP \downarrow
7	5	91.03	23.21	17.33	402.2
8	5	91.11	22.06	17.81	393.0
9	5	91.10	20.58	18.30	376.6
10	5	91.02	20.26	18.79	380.5
11	5	91.00	19.70	19.27	379.8

on Fashion-MNIST. Other models are trained for 200 epochs with an SGD-momentum optimizer (lr of 0.02 for ResNet, 0.002 for VGG8). We use learned stepsize quantization-aware training [5]. we employ $b_w=8$ -bit symmetric signed per-tensor quantization for weights and $b_{in}=6$ -bit for activations. For DST, we adopt an initial death rate of $\alpha^0=0.5$, $T_{\text{end}}=80\%$ total training steps. We update masks per epoch. No noise-aware training is applied, which is orthogonal to our method. **Architecture Settings.** We configure our architecture to have $R = 4$ tiles with $C = 4$ cores per tile. Each PTC is of size $k_1 = k_2 = 16$ working at clock frequency $f=5$ GHz. We assume the same device cost as prior work [29]. For the MZI power splitter, we have two options: the one from foundry has 30 mW P_π with 156.25 μm in width and 550 μm in length (Foundry-MZI); our optimized low-power MZI (LP-MZI) has a length of 115 μm and width of $l_s + w_{PS} = 9 + 6 = 15$ μm and a power profile shown in Fig. 4(c) ($P_\pi = 15.02$ mW).

Evaluation Metrics. We evaluate the total accelerator area (A), total energy $E_{\text{tot}} = \sum_i^L \sum_j^P \sum_k^Q (P_{i,j}^k \cdot \text{Cyc}_{i,j}^k / f)$, calculated by accumulating each PTC’s power over its execution runtime across all layers and all weight chunk, and average power $P_{\text{avg}} = \frac{E_{\text{tot}}}{\text{Cyc}_{\text{tot}}/f}$. We compare the area-energy efficiency (TOPS/W/mm²) for efficiency evaluation. To clarify, since a fine-grained row-column sparse model consumes the same cycle as a dense model, i.e., it still takes 1 cycle to map a $rk_1 \times ck_2$ weight block onto our accelerator regardless of row/column sparsity, allowing us to use power-area product (PAP) to guide the optimization, which is equivalent to TOPS/W/mm² given the same speed (lower PAP means higher TOPS/W/mm²). Note that the memory latency of loading sparse/dense weights is often hidden by optimized SRAM design [29]. Hence, we do not show throughput/speed in our results. We evaluate ideal accuracy and accuracy with crosstalk and random noises. Since the last layer is sensitive to error and pruning, we protect the last linear layer by mapping the weights to non-adjacent columns of MZIs to eliminate crosstalk. Note that we focus on the robustness and efficiency benefits from our circuit sparsity and light redistribution techniques on crossbar-style photonic tensor cores. Comparing the case-study architecture with other PTC designs or electronic digital accelerators is out of scope.

4.2 Ablation Study

We first explore different spaces to find optimal device/architecture settings and validate the effectiveness of our proposed techniques.

4.2.1 Optimal Device Spacing. Table 1 shows the trade-offs between spacing, area, power, and robustness across different MZI device spacing with a dense network ($s = 1$). Based on Fig. 6(a), we determine the most efficient arm spacing is l_s is 9 μm , minimizing PAP. To ensure $<1\%$ accuracy drop, the minimum MZI horizontal gap l_g is conservatively set to 5 μm . In later experiments, we show sparsity and power gating enable further shrinking of the device spacing down to $l_g=1\mu\text{m}$ with superior crosstalk tolerance.

Table 2: Evaluate accuracy and inference average power on CNN-FashionMNIST with different sparsity, architecture sharing factor r and c , and three sparsity.

r	c	Sparsity=0.8		Sparsity=0.6		Sparsity=0.4	
		P_{avg} (W) \downarrow	Acc (%) \uparrow	P_{avg} (W) \downarrow	Acc (%) \uparrow	P_{avg} (W) \downarrow	Acc (%) \uparrow
1	1	17.94	91.92	17.22	91.71	17.99	92.08
2	2	12.28	91.86	11.26	91.73	12.50	91.69
4	4	8.052	91.78	7.343	91.76	9.350	91.85

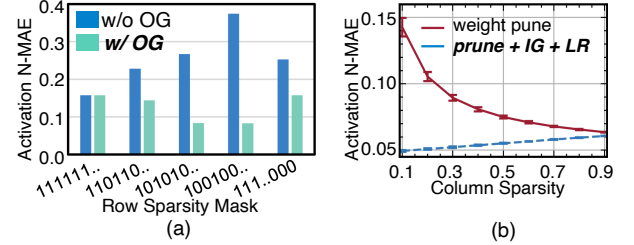


Figure 9: Thermal variation-induced activation error (N-MAE) on a 64-channel 3x3 CONV layer. (a) Output TIA/ADC gating (w/ OG) with row sparsity masks with interleaved 1’s can effectively reduce crosstalk-induced error. We use $l_s = 9\mu\text{m}$ and $l_g = 5\mu\text{m}$. (b) Input gating and light redistribution (IG+LR) can effectively suppress the error due to crosstalk and output noises.

4.2.2 Architecture Sharing Factor and Sparsity Granularity. With a predefined PTC size $k_1 \times k_2 = 16 \times 16$, we need to decide the optimal architecture sharing factor r and c to balance area, power, and accuracy. Table 2 explores the impact of input/readout sharing factors (r , c) and sparsity. A sharing factor $r = c = 4$ offers the best power efficiency with minimum accuracy drop, corresponding to pruning/growth of length-64 weight row/column vectors. With a larger accelerator scale, i.e., larger R and C , we will keep the same sharing factor to maintain the pruning granularity.

4.2.3 Light Redistribution and Power Gating. Figure 9(a) shows different row sparsity patterns and their impacts on thermal variation-induced activation error. Without TIA/ADC gating, sparse rows lead to even higher errors since zero elements still suffer from leakage and crosstalk errors. With the proposed gating, the crosstalk-induced error will be eliminated, and noises will also be reduced. Figure 9(b) investigates the effectiveness of light gating and redistribution. With lower sparsity (more zeros), the SNR will be largely increased with a significant fidelity boost. In our SCATTER system, we will enable OG+IG+LR together for the best thermal variation tolerance.

4.2.4 Progressive Power-Area Optimization. Figure 10 illustrates the step-by-step impact of our optimizations towards orders-of-magnitude power/area reduction. The baseline is chosen to be a dense network with foundry MZI switches without architectural hardware sharing ($r=c=1$). A conservative device spacing $l_g=20$ μm is adopted to avoid thermal crosstalk issues. ❶ As we replace the foundry MZI with our compact low-power LP-MZI device design, the chip area can be reduced by 279 \times with 41.1% average power saving. ❷ We further squeeze the tensor core layout with our optimal device spacing, i.e., $l_s=9$ μm and $l_g=5$ μm , which gives a merely 5.7% power penalty due to intra-MZI crosstalk but leads to 23.3% area saving. Note that this aggressive shrinking of gap l_g causes severe inter-MZI crosstalk, leading to large accuracy degradation for a dense network. ❸ The architectural sharing of input modulation and readout circuitry largely amortizes the DAC/ ADC cost, which further reduces the chip area

Table 3: Evaluation of ideal accuracy, accuracy with thermal variation (w/ TV), resumed accuracy with input light gating (IG) + output TIA/ADC gating (OG) + light redistribution (LR), and single-image inference energy consumption. CNN uses $s = 0.3$, and VGG8/ResNet18 use $s = 0.4$. Device spacing settings and accelerator area are shown in the upper left corner of the table. To clarify, the area adopts eoDAC, which is 0.704 mm^2 larger than the numbers shown in Fig. 10(⑦).

		$l_g=1\mu\text{m}$ Area=12.37 mm^2	$l_g=3\mu\text{m}$ Area=13.44 mm^2	$l_g=5\mu\text{m}$ Area=14.20 mm^2	Energy (mJ)
DensePTC	Ideal Acc	Acc w/ TV	Acc w/ TV	Acc w/ TV	
CNN-FMNIST	91.40	84.00	89.10	90.70	0.59
VGG8-CIFAR10	88.02	59.23	76.05	81.54	3.17
ResNet18-CIFAR100	66.46	44.12	57.84	60.94	24.06

		Acc w/ TV	Acc w/ TV +IG+OG+LR	Acc w/ TV	Acc w/ TV +IG+OG+LR	Acc w/ TV	Acc w/ TV +IG+OG+LR	Energy (mJ)
SCATTER	Ideal Acc	Acc w/ TV	Acc w/ TV +IG+OG+LR	Acc w/ TV	Acc w/ TV +IG+OG+LR	Acc w/ TV	Acc w/ TV +IG+OG+LR	
CNN-FMNIST	91.56	91.23	91.26	91.24	91.21	91.31	91.30	0.14
VGG8-CIFAR10	85.64	63.49	82.04	72.78	82.04	77.23	82.24	1.78
ResNet18-CIFAR100	59.18	0.51	57.40	0.86	57.40	0.51	57.46	11.18

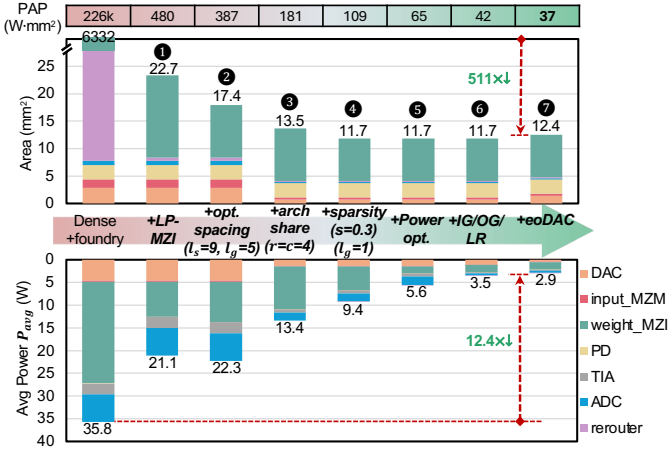


Figure 10: Significant power-area-product reduction can be achieved by progressively adding our proposed cross-layer optimization and algorithmic-circuit co-sparsity techniques. Detailed power/area breakdown has been presented.

by 39.9% and saves 22% power compared to dedicated DACs/ADCs for each PTC. ④ In order to handle the thermal effect from shrinking the l_g , we add $s=0.3$ algorithm-circuit co-sparsity to the accelerator. It allows us to turn off 70% of the weight MZIs with 30% power saving. With such interleaved row sparsity patterns, thermal crosstalk is mostly eliminated if output TIA/ADC gating (OG) is applied, which enables extremely narrow MZI spacing down to $l_g=1 \mu\text{m}$ with 13.3% smaller chip area. ⑤ During dynamic sparse training, we enable power-aware pruning/growth to select low-power column masks. ⑥ With input/output gating and light redistribution, we not only suppress most of the thermal variations but also actively turn off the unused DACs/MZMs and TIAs/ADCs. Also, power optimization helps to locate the least-power rerouter configurations. ⑦ At the final step, we upgrade the traditional 6-bit eDAC with our optimized hybrid eoDAC comprised of two 3-bit eDACs and a two-segment MZM. We trade $2\times$ the DAC area for $2.28\times$ power reduction, which overall boosts the system power-area product by another 12%.

4.3 Main Results

We compare the accuracy, power, and area on 2 settings and 3 benchmarks: (1) dense model and (2) SCATTER with power-optimized sparsity in Table 3. Both settings adopt the best configurations from Fig. 10.

Models are evaluated under different l_g with and without thermal variations. As the l_g decreases from $5 \mu\text{m}$ to $1 \mu\text{m}$, we can see a clear accuracy drop due to thermal variations for dense models. With a row-column sparsity $s=0.3\sim 0.4$, we observe some improvement on small benchmarks but much worse results on VGG8 and ResNet18. **Key Insights:** ① Sparsity itself does not naturally boost the thermal robustness. A sparse ResNet18 suffers from complete malfunction under crosstalk. This is expected based on our previous analysis in Fig. 9. ② Dense models degrade with smaller l_g due to crosstalk, while SCATTER resumes accuracy, when sparsity meets *in-situ* power gating and light redistribution (IG+OG+LR). ③ Sparsity can enable extremely narrow MZI spacing $l_g=1 \mu\text{m}$ to save chip real estate by another 12.9%. With input/output power gating, the single-image inference energy on three benchmarks can be reduced by an average of 52.9%. Our experiments demonstrate that SCATTER’s hardware/algorithm co-design significantly improves power efficiency and enables more compact photonic accelerators while maintaining thermal crosstalk robustness.

5 CONCLUSION AND DISCUSSION

In this work, we introduce SCATTER, the first dynamically reconfigurable photonic tensor core architecture featuring cross-layer optimization for power, area, and thermal robustness. Our *in-situ* light redistribution and power gating enable fine-grained signal path control, facilitating algorithm-circuit sparsity co-exploration for significant power reduction and thermal variation suppression. Our power/crosstalk-aware dynamic sparse training framework automatically explores thermally robust, low-power sparsity masks tailored to SCATTER hardware. We integrate synergistic optimization with customized compact low-power photonic devices, hybrid electrical-optical DACs, and optimal circuit/architecture design space exploration to maximize efficiency. Compared to dense photonic accelerators based on standard foundry devices, SCATTER can save chip area by $511\times$ and on-chip power consumption by $12.4\times$, maintaining deployment accuracy even with significant thermal crosstalk and chip noise. This framework’s dynamic reconfiguration and flexible signal path control establish a crucial design principle for next-generation reconfigurable photonic AI systems, pushing the boundaries of compute density and energy efficiency. Thermal crosstalk suppression via hardware/algorithm co-sparsity can be applied to other crossbar-type photonic tensor core designs, offering a generalizable and versatile co-design solution for reliable and efficient photonic AI computing systems.

REFERENCES

- [1] Sanmitra Banerjee, Mahdi Nikdast, Sudeep Pasricha, and Krishnendu Chakrabarty. 2022. Pruning Coherent Integrated Photonic Neural Networks Using the Lottery Ticket Hypothesis. In *2022 IEEE Comput. Soc. Annu. Symp. VLSI ISVLSI*. 128–133.

- [2] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. 2021. Chasing Sparsity in Vision Transformers: An End-to-End Exploration. In *Proc. NeurIPS*.
- [3] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman. 2020. Silicon Photonics Codesign for Deep Learning. *Proc. IEEE* (2020).
- [4] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. 2020. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proc. IEEE* (2020).
- [5] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. 2020. Learned Step Size Quantization. In *Proc. ICLR*.
- [6] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Raja, Junqiu Liu, David Wright, Abu Sebastian, Tobias Kippenberg, Wolfram Pernice, and Harish Bhaskaran. 2021. Parallel convolutional processing using an integrated photonic tensor core. *Nature* (2021).
- [7] Chenghao Feng, Jiaqi Gu, Hanqing Zhu, Zhoufeng Ying, Zheng Zhao, et al. 2022. A Compact Butterfly-Style Silicon Photonic-Electronic Neural Chip for Hardware-Efficient Deep Learning. *ACS Photonics* 9, 12 (2022), 3906–3916.
- [8] Jiaqi Gu, Chenghao Feng, Hanqing Zhu, et al. 2022. SqueezeLight: A Multi-Operand Ring-Based Optical Neural Network with Cross-Layer Scalability. *IEEE TCAD* (2022).
- [9] Jiaqi Gu, Zheng Zhao, Chenghao Feng, et al. 2020. Towards Area-Efficient Optical Neural Networks: an FFT-based architecture. In *Proc. ASPDAC*.
- [10] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Wuxi Li, Ray T. Chen, and David Z. Pan. 2020. FLOPS: Efficient On-Chip Learning for Optical Neural Networks Through Stochastic Zeroth-Order Optimization. In *Proc. DAC*.
- [11] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Hanqing Zhu, Ray T. Chen, and David Z. Pan. 2020. ROQ: A Noise-Aware Quantization Scheme Towards Robust Optical Neural Networks with Low-bit Controls. In *Proc. DATE*.
- [12] Jiaqi Gu, Hanqing Zhu, Chenghao Feng, Zixuan Jiang, Ray T. Chen, and David Z. Pan. 2021. L2ight: Enabling On-Chip Learning for Optical Neural Networks via Efficient in-situ Subspace Optimization. In *Proc. NeurIPS*.
- [13] Manos Kirtas, Nikolaos Passalis, Nikolaos Pleros, and Anastasios Tefas. 2023. Non-negative isomorphic neural networks for photonic neuromorphic accelerators. *ArXiv abs/2310.01084* (2023).
- [14] Mike Lasby, Anna Golubeva, Utku Evci, Mihai Nica, and Yani Ioannou. 2024. Dynamic Sparse Training with Structured Sparsity. In *Proc. ICLR*.
- [15] Junjie Liu, Zhe Xu, Runbin Shi, Ray C. C. Cheung, and Hayden K.H. So. 2020. Dynamic Sparse Training: Find Efficient Sparse Network From Scratch With Trainable Masked Layers. In *Proc. ICLR*.
- [16] Haotian Lu, Sanmitra Banerjee, and Jiaqi Gu. 2024. DOCTOR: Dynamic On-Chip Temporal Variation Remediation Toward Self-Corrected Photonic Tensor Accelerators. *IEEE Journal of Lightwave Technology* (2024).
- [17] Sangkug Lym and Mattan Erez. 2020. FlexSA: Flexible Systolic Array Architecture for Efficient Pruned DNN Model Training.
- [18] Asif Mirza, Febin Sunny, et al. 2022. Silicon Photonic Microring Resonators: A Comprehensive Design-Space Exploration and Optimization Under Fabrication-Process Variations. *IEEE TCAD* 41, 10 (2022), 3359–3372.
- [19] Bhavin J. Shastri, Alexander N. Tait, et al. 2021. Photonics for Artificial Intelligence and Neuromorphic Computing. *Nature Photonics* (2021).
- [20] Yichen Shen, Nicholas C. Harris, Scott Skirlo, et al. 2017. Deep Learning with Coherent Nanophotonic Circuits. *Nature Photonics* (2017).
- [21] Alexander N. Tait, Thomas Ferreira de Lima, Ellen Zhou, et al. 2017. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* (2017).
- [22] Wei Wen, Chumpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning Structured Sparsity in Deep Neural Networks. In *Proc. NIPS*.
- [23] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G. Nguyen, Sai T. Chu, Brent E. Little, Damien G. Hicks, Roberto Morandotti, Arnan Mitchell, and David J. Moss. 2021. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* (2021).
- [24] Zhihao Xu, Tiankuang Zhou, Muzhou Ma, ChenChen Deng, Qionghai Dai, and Lu Fang. 2024. Large-scale photonic chiplet Taichi empowers 160-TOPS/W artificial general intelligence. *Science* 384, 6692 (2024), 202–209.
- [25] Jie-Fang Zhang, Ching-En Lee, Chester Liu, Yakun Sophia Shao, Stephen W. Keckler, and Zhengya Zhang. 2021. SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference. *IEEE J. Solid-State Circuits* 56, 2 (Feb. 2021), 636–647.
- [26] Meng Zhang, Dennis Yin, Nicholas Gangi, Amir Begović, Alexander Chen, Zhao-ran Rena Huang, and Jiaqi Gu. 2024. TeMPO: Efficient Time-Multiplexed Dynamic Photonic Tensor Core for Edge AI with Compact Slow-Light Electro-Optic Modulator. [arXiv:2402.07393](https://arxiv.org/abs/2402.07393) [cs.ET]
- [27] Zhekai Zhang, Hanrui Wang, Song Han, and William J. Dally. 2020. SpArch: Efficient Architecture for Sparse Matrix Multiplication. In *Proc. HPCA*.
- [28] Zheng Zhao, Jiaqi Gu, Zhoufeng Ying, et al. 2019. Design Technology for Scalable and Robust Photonic Integrated Circuits. In *Proc. ICCAD*.
- [29] Hanqing Zhu, Jiaqi Gu, Hanrui Wang, Zixuan Jiang, Zhekai Zhang, Rongxin Tang, Chenghao Feng, Song Han, et al. 2024. Lightning-Transformer: A Dynamically-Operated Photonic Tensor Core for Energy-Efficient Transformer Accelerator. In *Proc. HPCA*.
- [30] H.H. Zhu, J. Zou, H. Zhang, et al. 2022. Space-efficient optical computing with an integrated chip diffractive neural network. *Nature Commun.* (2022).
- [31] Ying Zhu, Grace Li Zhang, Bing Li, et al. 2020. Countering Variations and Thermal Effects for Accurate Optical Neural Networks. In *Proc. ICCAD*.
- [32] Farzaneh Zokaee, Qian Lou, Nathan Youngblood, et al. 2020. LightBulb: A Photonic-Nonvolatile-Memory-based Accelerator for Binarized Convolutional Neural Networks. In *Proc. DATE*.