

Opto-ViT-v2: Noise-Resilient On-Chip Fine-Tuning for Photonic Near-Sensor Vision Transformer Accelerators

Xuming Chen
xuming.chen@case.edu
Case Western Reserve University
USA

Deniz Najafi
dn339@njit.edu
New Jersey Institute of Technology
USA

Mehrdad Morsali
mm2772@njit.edu
New Jersey Institute of Technology
USA

Chengwei Zhou
chengwei.zhou@case.edu
Case Western Reserve University
USA

Zahra Ghanaatianjobzari
Zahra.Ghanaatian@colostate.edu
Colorado State University
USA

Mahdi Nikdast
mahdi.nikdast@colostate.edu
Colorado State University
USA

Shaahin Angizi
shaahin.angizi@njit.edu
New Jersey Institute of Technology
USA

Gourav Datta
gourav.datta@case.edu
Case Western Reserve University
USA

ABSTRACT

Silicon-photonics (SiPh) accelerators have emerged as a promising platform for Vision Transformer (ViT) inference, achieving high throughput and energy efficiency by performing matrix multiplications on microring-resonator (MRR) banks. Extending these platforms to support on-chip fine-tuning, however, has remained an open problem: backpropagation demands large activation storage that exceeds typical on-chip buffer capacity, weight write-back to MRR banks after each gradient step is prohibitively slow, and device-level noise corrupts both forward and backward computation. We present **Opto-ViT-v2**, the first framework that enables parameter-efficient fine-tuning (PEFT) on a near-sensor SiPh ViT accelerator. We propose a tensorized low-rank decomposition of the weight increments that separates pretrained weights (executed on the optical cores) from a small set of trainable factors (as few as 8K parameters for ViT-Base) that reside and are updated entirely in the electronic domain. This decomposition dramatically reduces both the activation storage required during backpropagation and the number of weight write-back operations, making on-chip training feasible within the memory and tuning constraints of the photonic architecture. We further introduce a *gradient-accumulated sparse head* that freezes low-importance classifier weights via one-shot top- k gradient masking, reducing the head training budget by $\sim 40\%$. To evaluate training under realistic hardware conditions, we develop the first system-level noise model for photonic on-chip training, characterizing how MRR crosstalk, thermal drift, and laser amplitude noise propagate through both forward and backward passes. Calibrated against >200 fabricated MRR devices, this model enables evaluation of PEFT methods under photonic noise and reveals that low-rank factor updates exhibit greater robustness than full fine-tuning or per-layer low-rank adaptation under identical noise conditions. Evaluated on VTAB-1K (19 tasks) and FGVC few-shot benchmarks using a full-stack device-to-application co-design framework, Opto-ViT-v2 recovers within 0.3-0.8% of clean-software accuracy under measured photonic noise, achieving >100 KFPS/W throughput and enabling, for the first time, on-chip domain adaptation at the photonic sensor edge.

1 INTRODUCTION

Vision Transformers (ViTs) [1] have become the dominant backbone for tasks spanning image classification, object detection, and video understanding, owing to their scalable self-attention mechanism and strong transfer-learning capability. The standard deployment paradigm pretrains a ViT on a large-scale dataset (e.g., ImageNet-21K) and then fully fine-tunes it on each downstream task. This paradigm, however, poses two compounding challenges for resource-constrained edge systems. First, the dense matrix multiplications in multi-head self-attention (MHSA) and feed-forward network (FFN) blocks are computationally intensive, demanding accelerator-class hardware even for inference. Second, adapting the model to new domains requires updating tens of millions of parameters, a process that is memory- and energy-prohibitive on embedded platforms.

Silicon-photonics (SiPh) accelerators offer an attractive path toward addressing the inference challenge. By encoding weights onto microring-resonator (MRR) banks and performing multiply-accumulate (MAC) operations via wavelength-division multiplexing (WDM), architectures such as CrossLight [2], Lightator [3], and Opto-ViT [4] have demonstrated inference throughput exceeding 100 KFPS/W with significant energy savings over digital baselines. These systems stream pretrained weight matrices onto MRR arrays layer by layer and execute the ViT inference pipeline through a hybrid electronic-photonic dataflow.

The second challenge, on-chip fine-tuning, remains unsolved for photonic platforms. Three fundamental constraints make backpropagation-based training difficult on SiPh hardware. (i) *Activation storage*: Backpropagation requires intermediate activations from the forward pass to compute parameter gradients during the backward pass. Storing them on-chip alongside the weight parameters far exceeds the buffer memory capacity of current photonic architectures; recomputing them from scratch for each layer during the backward pass, while possible, multiplies the number of optical core invocations and is prohibitively expensive. (ii) *Weight write-back*: In standard fine-tuning, updated weights must be written back to

the MRR banks after each gradient step. Repeated over many mini-batches and epochs, this write-back process dominates latency and energy, fundamentally limiting training throughput. (iii) *Device noise*: Photonic MAC operations are subject to fabrication variability, thermal crosstalk between adjacent MRRs, and laser relative intensity noise, all of which corrupt both forward activations and backward gradient signals, degrading training convergence.

Parameter-efficient fine-tuning (PEFT) methods offer a path forward by dramatically reducing the number of trainable parameters. Methods such as LoRA [5], visual prompt tuning (VPT) [6], and adapters [7] have shown that updating fewer than 1% of a ViT’s parameters can match or exceed full fine-tuning accuracy on diverse downstream tasks [8–10]. Recent work on tensorized decomposition of weight increments [8] has pushed this further, achieving competitive accuracy with as few as 0.01% of trainable parameters by exploiting both intra-layer and cross-layer redundancy. However, no prior work has investigated how to map PEFT training onto photonic hardware, or whether the reduced parameter footprint translates into concrete architectural benefits under the activation storage, write-back, and noise constraints of SiPh accelerators.

In this paper, we present **Opto-ViT-v2**, the first framework enabling on-chip PEFT on a near-sensor silicon-photon ViT accelerator, targeting real-world vision tasks. We show that a tensorized low-rank decomposition of weight increments is not merely parameter-efficient in the software sense, but directly addresses all three photonic training bottlenecks: the small trainable factors require orders of magnitude less activation caching during backpropagation, eliminate MRR weight write-back entirely (since pretrained weights remain static on the optical cores), and confine all parameter updates to the electronic domain where they are immune to photonic noise. Our contributions are as follows.

(1) First on-chip photonic PEFT via hardware–algorithm co-design. We develop a tensorized low-rank decomposition strategy co-designed with the Opto-ViT-v2 framework: pretrained weights $\mathbf{W}_0$ are streamed onto MRR banks during forward and backward passes, while a compact set of trainable factors resides entirely in the electronic processing unit. By keeping $\mathbf{W}_0$ fixed, we eliminate MRR retuning and costly weight write-backs. Factor gradients are computed via small ($d \times r$, $r \times r$) operations without forming full $d \times d$ gradients, significantly reducing compute and memory overhead. We further design the complete forward–backward dataflow, including optical scheduling, activation compression leveraging low-rank structure, and gradient checkpointing aligned with the Opto-ViT-v2 memory hierarchy.

(2) Gradient-accumulated sparse head. While tensorized decomposition significantly reduces the backbone’s trainable parameters, the task-specific classification head comes to dominate the budget, accounting for over 90% of the total trainable parameters. Applying low-rank factorization to the head degrades accuracy because the classifier must produce sharp, class-discriminative logits that aggressive rank reduction blurs. We propose a complementary approach that accumulates gradients over multiple mini-batches to derive robust importance estimates, and then retains only the top- k parameters (by gradient magnitude) as trainable while freezing the rest. This *gradient-accumulated sparse head* reduces the overall training budget by $\sim 40\%$ with negligible accuracy loss.

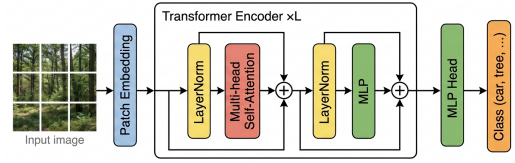


Figure 1: Architecture overview of a standard ViT encoder that processes patch embeddings through L MHSA-FFN layers with layer normalization and residuals.

(3) Full-stack evaluation with photonic training noise model.

The first system-level noise model for photonic on-chip training is designed, capturing how MRR crosstalk, thermal drift, and laser amplitude noise propagate through both forward and backward passes. Calibrated against measurements from >200 fabricated MRR devices, the model enables realistic evaluation of PEFT methods under photonic noise. Using a full-stack device-to-application framework, we show that Opto-ViT-v2 recovers within 0.3–0.8% of noise-free accuracy while achieving >100 KFPS/W, and that low-rank factor updates exhibit improved robustness compared to full fine-tuning and per-layer adaptation.

2 BACKGROUND & RELATED WORK

Vision Transformer Architecture: As shown in Fig. 1, A Vision Transformer (ViT) [1] partitions an input image into non-overlapping patches, projects each patch into a d -dimensional embedding, and processes the resulting sequence through L encoder blocks. Each block consists of a multi-head self-attention (MHSA) module followed by a feed-forward network (FFN), with layer normalization and residual connections around each sub-layer. In MHSA, the input $\mathbf{X} \in \mathbb{R}^{N \times d}$ (where N is the number of patches) is projected into queries, keys, and values via weight matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$, which are further divided into N_h heads of dimension $d_k = d/N_h$. The self-attention output for head i is

$$\text{Head}_i(\mathbf{X}) = \text{softmax}\left(\frac{\mathbf{X}\mathbf{W}_Q^{(i)}\mathbf{W}_K^{(i)\top}\mathbf{X}^\top}{\sqrt{d_k}}\right)\mathbf{X}\mathbf{W}_V^{(i)}\mathbf{W}_O^{(i)\top} \quad (1)$$

and the outputs of all heads are concatenated and linearly projected. The FFN applies two fully connected layers with a GELU nonlinearity: $\text{FFN}(\mathbf{X}) = \text{GELU}(\mathbf{X}\mathbf{W}_{\text{up}})\mathbf{W}_{\text{down}}$, where $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d \times 4d}$ and $\mathbf{W}_{\text{down}} \in \mathbb{R}^{4d \times d}$. For ViT-Base/16, $L=12$, $d=768$, $N_h=12$, and $d_k=64$, yielding approximately 86M total parameters. Each encoder block contains four $d \times d$ matrices in the MHSA module ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$) and two matrices in the FFN ($\mathbf{W}_{\text{up}}, \mathbf{W}_{\text{down}}$), with the FFN matrices subdivided into $d \times d$ blocks as described in [8]. This structural uniformity across layers is important for the tensorized decomposition we employ in Section 3.

Silicon-Photonic ViT Acceleration: Silicon-photonic (SiPh) accelerators perform MACs in the optical domain by encoding inputs as light amplitudes and weights as microring resonator (MRR) transmission coefficients. With wavelength-division multiplexing (WDM), multiple multiplications occur in parallel across wavelengths, and balanced photodetectors accumulate results. MRR weights are programmed via thermo- or electro-optic tuning, shifting resonance to modulate overlapping optical signals. Prior SiPh accelerators target specific DNN settings: LightBulb [11] for binarized

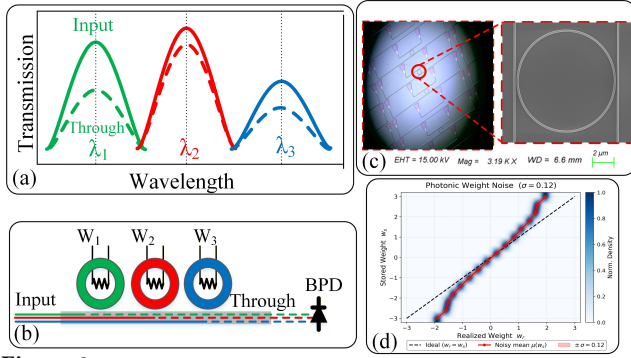


Figure 2: (a) Spectra at the MRR input and through ports after parameter imprinting via tuning. Adjusting the resonant wavelength causes part of the signal to couple into the ring while the rest passes to the through port, encoding the parameter in the output. (b) Multiple MRRs in one arm imprint weight values across different wavelengths of the input signal. (c) Fabricated SiPh IC with over 200 identical MRR cell layouts with SEM image. (d) Photonic weight noise.

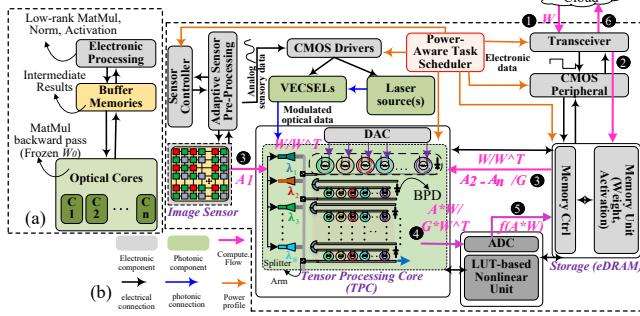


Figure 3: (a) Opto-ViT-v2 operations, and (b) photonic architecture and corresponding interface signals for Opto-ViT-v2.

CNNs (limited by ADC power), HolyLight [12] and CrossLight [2] for reduced-precision convolutions, ROBIN [13] for robustness, Lightator [3] for near-sensor processing, TRON [14] for transformer blocks, and Opto-ViT [4] for hybrid photonic–electronic ViTs with WDM parallelism and pruning. However, none support on-chip training.

Parameter-Efficient Fine-Tuning for ViTs: Parameter-efficient transfer learning (PETL) adapts pretrained models by updating only a small subset of parameters, reducing storage and compute. Adapter methods insert lightweight bottlenecks [7, 9, 15], while prompt-based methods prepend learnable tokens [6, 10]. Low-rank methods like LoRA [5] decompose each weight update as $\Delta W = sAB$, but operate independently per layer. Tensorized methods address this by jointly decomposing all weights. FacT [8] stacks all ViT weight matrices into a tensor $\mathcal{W} \in \mathbb{R}^{12L \times d \times d}$, where $12L$ denotes the total number of $d \times d$ matrices across L layers (4 from MHSA and 8 from FFN per layer). The update ΔW is then factorized (e.g., Tensor-Train), sharing factors across layers. At rank $r=4$, FacT-TT uses only $\sim 8K$ parameters ($\sim 0.01\%$ of ViT-Base) while achieving competitive accuracy.

3 PROPOSED METHOD: OPTO-VIT-V2

Fig. 3 provides an overview of the proposed Opto-ViT-v2 node architecture, highlighting its electronic and photonic building blocks, which are *co-designed and co-optimized* to realize a standalone,

reconfigurable, and energy-aware optical vision sensor that seamlessly integrates sensing and processing. *The design intrinsically supports granularity-configurable, low-bit-width implementations of MHSA, patch embedding and linear layers in ViTs. This enables flexible trade-offs between power consumption and accuracy.* Opto-ViT-v2 consists of the following steps.

In step 1, the weights are received from the cloud via an RF transceiver and written to on-chip memory 2. In step 3, the frozen weights are streamed layer by layer onto the Tensor Processing Core (TPC) via the memory controller. The TPC comprises 64 waveguide arms, each supporting 32 wavelength channels (aligned with d_k), with MRRs performing parallel multiplication and BPDs accumulating results. Input activations are captured by a power-aware CMOS image sensor and modulated directly onto VCSEL-generated optical signals, avoiding the higher cost of MRR-based input encoding. In step 4, MAC operations execute in parallel on the TPC, with results fed to a programmable LUT-based nonlinear unit for activation functions and their derivatives. In step 5, TPC outputs are digitized via ADCs and written to the activation memory, with compressed intermediates cached for the backward pass. In step 6, trained factors can be absorbed into the weights for zero-overhead inference, and results are transmitted to the cloud if required.

We present three contributions that together form a complete on-chip fine-tuning framework: Section 3.1 maps the tensorized low-rank decomposition onto the TPC and electronic unit, deriving the forward and backward dataflows that keep W_0 frozen on the optical cores while confining all trainable computation to the electronic domain. Section 3.2 introduces a gradient-accumulated sparse head that reduces the training budget by $\sim 40\%$. Section 3.3 develops the first system-level noise model for photonic training, characterizing how MRR device noise in both optical passes propagates into the factor gradients computed electronically in Section 3.1.

3.1 Hardware Mapping of Tensorized Low-Rank Training

Training neural networks on analog or mixed-signal hardware is fundamentally challenging. Prior work on analog in-memory computing (AIMC) [16] and ReRAM-based transformer accelerators [17] has identified weight write-back overhead, limited update precision, and intermediate data buffering as primary obstacles. In SiPh accelerators, the analogous bottlenecks are MRR retuning latency, buffer memory capacity, and photonic MAC noise. These constraints make standard backpropagation with full weight updates infeasible on the Opto-ViT-v2 platform.

As shown in Fig. 4, our approach is to decompose the fine-tuning weight update into a heavy frozen component executed on the optical cores and a light trainable component confined to the electronic domain. We adopt the tensorized Tensor-Train decomposition [8], in which the weight of the i -th sub-layer during fine-tuning is expressed as $W_i = W_{0,i} + s \cdot U \Sigma_i V^T$, where $W_{0,i} \in \mathbb{R}^{d \times d}$ is the frozen pretrained weight, $U \in \mathbb{R}^{d \times r}$ and $V \in \mathbb{R}^{d \times r}$ are global factors shared across all $12L$ sub-layers, $\Sigma_i \in \mathbb{R}^{r \times r}$ is specific to sub-layer i , s is a scaling hyperparameter, and $r \ll d$. Only U , V , and $\{\Sigma_i\}_{i=1}^{12L}$ are trainable. For ViT-Base ($L=12$, $d=768$) at rank $r=4$, the total trainable parameter count is $2dr + 12Lr^2 = 2 \cdot 768 \cdot 4 + 144 \cdot 16 = 8,448$, occupying fewer than 8.5 KB at 8-bit precision.

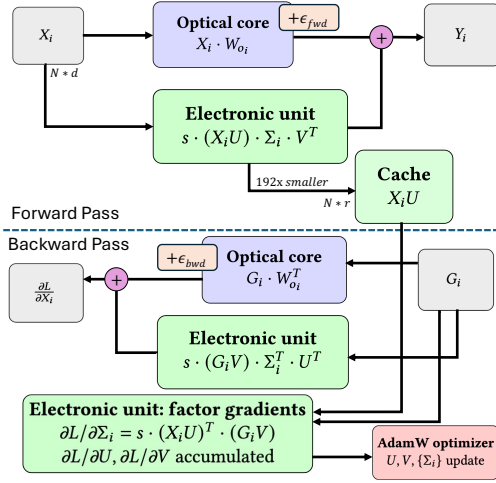


Figure 4: Opto-ViT-v2 per-layer training dataflow. Optical cores (blue) handle the heavy W_0 multiplications in both passes with photonic noise (amber); the electronic unit (green) computes the lightweight factor chain and all factor gradients using the compressed cached intermediate $X_i U$ (192 $\times$ SRAM reduction). W_0 is never retuned

We now derive how this decomposition maps onto the Opto-ViT-v2 hardware for both the forward and backward passes, and present the five-core scheduling strategy.

3.1.1 Forward Pass. The forward pass through sub-layer i computes the output Y_i from input $X_i \in \mathbb{R}^{N \times d}$:

$$Y_i = \underbrace{X_i W_{0,i}}_{\text{optical}} + s \cdot \underbrace{(X_i U) \Sigma_i V^T}_{\text{electronic}} \quad (2)$$

The first term is a $N \times d$ by $d \times d$ matrix multiplication requiring Nd^2 MACs. This is the dominant computational cost and is executed on the optical cores, with $W_{0,i}$ streamed onto the MRR banks layer by layer in 32-element chunks following the partitioning strategy described in [4]. Since $W_{0,i}$ is never modified, the MRR banks serve as a static weight store throughout the entire fine-tuning process, and no retuning or write-back is required. The second term decomposes into a chain of three small matrix multiplications executed on the electronic processing unit: (a) $X_i U$: $N \times d$ by $d \times r \rightarrow N \times r$ (Ndr MACs); (b) $(X_i U) \Sigma_i$: $N \times r$ by $r \times r \rightarrow N \times r$ (Nr^2 MACs); (c) result $\times V^T$: $N \times r$ by $r \times d \rightarrow N \times d$ (Ndr MACs). For $N=196$, $r=4$, and $d=768$, the electronic factor chain requires $2Ndr + Nr^2 \approx 1.2M$ MACs, compared to $Nd^2 \approx 115M$ MACs for the optical term (<1.1% overhead), and can run in parallel.

Compressed activation caching. A key architectural benefit of the tensorized decomposition is the reduction in activation storage during training. Standard backpropagation caches the full input $X_i \in \mathbb{R}^{N \times d}$ at each sub-layer for use in the gradient computation. With $12L=144$ sub-layers in ViT-Base, the total activation footprint is $144 \times 196 \times 768 \times 2B \approx 42$ MB, far exceeding the buffer memory of Opto-ViT-v2. In our framework, the factor gradient for Σ_i (Eq. (4) below) depends only on the projected intermediate $X_i U \in \mathbb{R}^{N \times r}$, not the full X_i . We therefore cache only this small product during

the forward pass. For $r=4$, this compresses per-layer activation storage from $196 \times 768 = 150,528$ values to $196 \times 4 = 784$ values, a **192 $\times$ reduction**. Across all 144 sub-layers, the total cached footprint drops to $144 \times 784 \times 2B \approx 220$ KB, which comfortably fits in the electronic unit's SRAM.

3.1.2 Backward Pass. The backward pass must accomplish two tasks: propagate the gradient signal backward through the network, and compute the gradients of the loss with respect to the trainable factors.

Gradient propagation. Let $G_i \triangleq \partial \mathcal{L} / \partial Y_i \in \mathbb{R}^{N \times d}$ denote the upstream gradient at sub-layer i . The gradient with respect to the sub-layer input is

$$\frac{\partial \mathcal{L}}{\partial X_i} = \underbrace{G_i W_{0,i}^T}_{\text{optical}} + s \cdot \underbrace{(G_i V) \Sigma_i^T U^T}_{\text{electronic}} \quad (3)$$

The first term is a dense $N \times d$ by $d \times d$ multiplication executed on the optical cores. Since $W_{0,i}$ is frozen, its transpose $W_{0,i}^T$ is precomputed offline and streamed onto the MRR banks during the backward pass, incurring only the standard layer-by-layer streaming latency with no additional tuning overhead. The second term is a chain of small electronic multiplications: $G_i V$ ($N \times r$), then $\times \Sigma_i^T$ ($N \times r$), then $\times U^T$ ($N \times d$), contributing negligible overhead relative to the optical term. **Factor gradient: per-layer core Σ_i .** The gradient with respect to the per-layer factor is computed entirely in the electronic domain:

$$\frac{\partial \mathcal{L}}{\partial \Sigma_i} = s \cdot \underbrace{(X_i U)^T}_{r \times N, \text{ cached}} \underbrace{(G_i V)}_{N \times r} \quad (4)$$

The left operand $(X_i U) \in \mathbb{R}^{N \times r}$ was cached during the forward pass. The right operand $G_i V \in \mathbb{R}^{N \times r}$ is a small electronic multiplication. The result is an $r \times r$ matrix: for $r=4$, a 4×4 output from two 196×4 operands. This is computed independently for each of the 144 sub-layers.

Factor gradient: shared factors U and V. The gradient accumulates contributions from all sub-layers:

$$\frac{\partial \mathcal{L}}{\partial U} = s \sum_{i=1}^{12L} X_i^T \underbrace{G_i V \Sigma_i^T}_{\triangleq Z_i \in \mathbb{R}^{N \times r}} \quad \frac{\partial \mathcal{L}}{\partial V} = s \sum_{i=1}^{12L} G_i^T \underbrace{(X_i U) \Sigma_i}_{\in \mathbb{R}^{N \times r}} \quad (5)$$

Each per-layer term Z_i reuses the $G_i V$ already computed for Eq. (4). The product $X_i^T Z_i \in \mathbb{R}^{d \times r}$ requires the full $X_i \in \mathbb{R}^{N \times d}$, which was not cached (only $X_i U$ was stored). During the backward pass, as layers are processed in reverse order, X_i can be reconstructed by a single forward evaluation of the preceding sub-layer on the optical cores using the stored layer-boundary checkpoint. This adds one extra optical core invocation per layer but avoids storing the full $N \times d$ activations for all layers simultaneously. Similarly, for V , $(X_i U) \Sigma_i$ reuses the cached $X_i U$ and the small Σ_i . The product $G_i^T (\cdot) \in \mathbb{R}^{d \times r}$ is well within the electronic unit's capacity.

Optimizer step. After accumulating the factor gradients across all layers, the AdamW optimizer is applied to U , V , and $\{\Sigma_i\}$ entirely in the electronic domain. The optimizer state (first and second moments) adds $2 \times 8,448 = 16,896$ values, and the total SRAM footprint for factors plus optimizer state is under 50 KB.

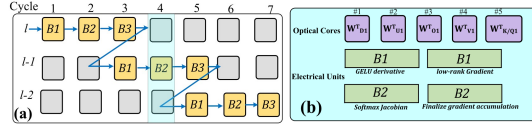


Figure 5: (a) Pipelined processing of backward stages of different layers. (b) Optical and Electronic Hardware allocation for backward stage in Cycle 4

3.1.3 Five-Core Scheduling. The Opto-ViT-v2 architecture provides five optical processing cores (C1–C5), each with 32 WDM wavelength channels across 64 waveguide arms. We propose a *forward-backward pipelined* scheduling strategy that reuses the same physical cores for both passes.

Forward phase (layers $l = 1 \rightarrow L$). Each encoder layer is processed following the Opto-ViT-v2 attention pipeline [4]. In stage F1, cores C1, C2, and C3 are simultaneously tuned with $\mathbf{W}_Q$, $\mathbf{W}_K^T/\sqrt{d_k}$, and $\mathbf{X}^T$, computing the decomposed attention score $(\mathbf{Q}\mathbf{W}_K^T)\mathbf{X}^T$ without buffering the intermediate $\mathbf{K}$ (Eq. (2) in [4]). In stage F2, the electronic unit computes softmax and C4–C5 process the value projection and output via $\mathbf{W}_V$ and the softmax result. The FFN sub-layer is then handled by streaming $\mathbf{W}_{up}$ and $\mathbf{W}_{down}$ onto C1–C2 (two cores suffice since the FFN has two sequential linear layers). Throughout the forward phase, the electronic unit computes the factor contributions $s(\mathbf{X}_i\mathbf{U})\Sigma_i\mathbf{V}^T$ in parallel with the optical computation and caches $\mathbf{X}_i\mathbf{U}$ for each sub-layer.

Backward phase (layers $l = L \rightarrow 1$). Processing reverses the layer order. For each layer, we define three stages that overlap optical and electronic operations to maximize core utilization as shown in Fig. 5.

Stage B1 (FFN backward): Cores C1–C2 are tuned with the pre-computed transposes $\mathbf{W}_{down}^T$ and $\mathbf{W}_{up}^T$ to propagate the gradient through the FFN block via Eq. (3). Concurrently, the electronic unit computes the GELU derivative and the FFN factor gradients (Eqs. (4)–(6)) using the cached $\mathbf{X}_i\mathbf{U}$ intermediates and the upstream gradient $\mathbf{G}_i$.

Stage B2 (MHSA backward): Cores C3–C5 are tuned with $\mathbf{W}_Q^T$, $\mathbf{W}_K^T$, and the attention backward weights $(\mathbf{W}_Q^T, \mathbf{W}_K^T)$ to propagate the gradient through the MHSA block. The softmax Jacobian is computed electronically in parallel. Cores C1–C2, now free from stage B1, can assist by handling additional MHSA weight transposes if needed for the four attention sub-layers.

Stage B3 (overlap with next layer): While the optical cores begin streaming weights for layer $l-1$'s backward pass, the electronic unit finalizes factor gradient accumulation for layer l and, if needed, recomputes $\mathbf{X}_i$ for Eq. (5) by executing a single forward sub-layer evaluation on a temporarily available core (C4 or C5).

This staggered pipeline ensures that the optical cores and Electronic units are rarely idle simultaneously. The key enabler is that all weight transposes $\mathbf{W}_{0,i}^T$ are precomputed and stored in the same weight memory used for $\mathbf{W}_{0,i}$ during the forward phase, so the backward phase incurs no additional tuning latency beyond the standard layer-by-layer streaming.

3.2 Gradient-Accumulated Sparse Head

The tensorized decomposition reduces backbone trainable parameters to $\sim 8K$ for ViT-Base, but the task-specific classification head $\mathbf{W}_{head} \in \mathbb{R}^{d \times C}$ (where C is the number of classes) requires $d \cdot C$ parameters. For $C=100$ (as in CIFAR-100 or many VTAB-1K tasks),

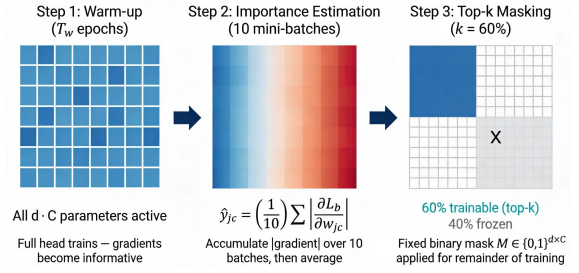


Figure 6: Gradient-Accumulated Sparse Head Training.

this is $768 \times 100 = 76,800$ parameters, which is $9\times$ the backbone budget and dominates the total trainable count. Applying low-rank factorization to the head is ineffective: the classifier must map from a shared d -dimensional representation to C class-specific logits, and aggressive rank reduction blurs inter-class boundaries, causing significant accuracy drops.

We propose an orthogonal approach: *gradient-accumulated sparse training* of the head. The method determines a fixed sparsity mask early in training and applies it for the remainder, proceeding in three steps.

Step 1: Warm-up. We train the full head (all $d \cdot C$ parameters active) for T_w epochs, allowing the logit space to reach a coarse alignment with the target task. During this phase, the head parameters settle into a neighborhood where gradient magnitudes become informative of long-term parameter importance.

Step 2: Importance estimation. After warm-up, we accumulate the absolute gradient magnitude $|\partial \mathcal{L} / \partial w_{jc}|$ for each head parameter w_{jc} over $B_{acc}=10$ consecutive mini-batches, then average:

$$\bar{g}_{jc} = \frac{1}{10} \sum_{b=1}^{10} \left| \frac{\partial \mathcal{L}_b}{\partial w_{jc}} \right| \quad (6)$$

Accumulating over 10 batches provides sufficient statistical stability to distinguish consistently important parameters from those with transiently large gradients due to mini-batch noise, while adding negligible overhead to the training loop.

Step 3: Top- k masking. We rank head parameters by $\bar{g}_{jc}$ and construct a binary mask $\mathbf{M}$ that retains the top- k fraction as trainable, while freezing the rest for the remainder of training. The update becomes $w_{jc} \leftarrow w_{jc} - \eta M_{jc} \partial \mathcal{L} / \partial w_{jc}$.

In practice, retaining $k=60\%$ yields negligible accuracy loss. The total trainable parameters are reduced to $\sim 54K$ (backbone + head). On the photonic hardware, the fixed mask means that 40% of the head gradient computations and the corresponding optimizer state are eliminated for the entire training run after the brief warm-up and estimation phase, reducing both SRAM footprint and backward compute cost.

3.3 Photonic Noise Model for Training

Existing SiPh accelerator studies [2–4, 13] characterize MRR device noise exclusively for the inference setting. In this work, we extend the noise model developed for the Opto-ViT-v2 inference pipeline [4] to cover the full training loop, incorporating the same device-level noise sources into both the forward and backward optical passes.

3.3.1 Noise Sources. Prior photonic noise modeling frameworks primarily treat device non-idealities, including fabrication variation, thermal drift, and laser noise, as *additive or multiplicative stochastic perturbations* with activation-dependent variance proxies derived at the MAC or logit level [18]. While such models capture aggregate noise statistics, they assume *independent, zero-mean perturbations* and do not explicitly model spatial thermal interactions or weight-dependent effects. In contrast, we focus on **thermo-optic crosstalk** as a dominant source of structured, correlated noise in dense MRR arrays. Our device-level characterization shows that heater-induced temperature variations propagate laterally across waveguides, introducing *deterministic and spatially coupled resonance shifts* that depend on both **input optical power** and **inter-ring spacing**. Unlike prior stochastic models, this behavior cannot be captured by independent Gaussian noise. To address this, we model the effective resonance of each MRR as:

$$\lambda_i^{\text{act}} = \lambda_i^{\text{LUT}} + \underbrace{\alpha_i(P) \rho(d_i)}_{\text{thermal crosstalk}} + \mu_{r,i} + \epsilon_i, \quad (7)$$

where $\alpha_i(P)$ captures the **input power-dependent thermal shift**, and $\rho(d_i)$ models **distance-dependent heat diffusion** across neighboring rings. This formulation explicitly captures *weight-dependent noise*, since the effective perturbation depends on the programmed detuning and optical intensity at each MRR. Importantly, thermal drift also alters spectral spacing, leading to **dynamic, input-dependent inter-channel crosstalk**: $\phi_{ij}^{\text{act}} = \frac{\delta_i^2}{(\lambda_i^{\text{act}} - \lambda_j^{\text{act}})^2 + \delta_i^2}$, which couples neighboring channels through thermally shifted resonances.

3.3.2 Forward-Pass Noise. Under the proposed thermo-optic crosstalk model, the optical computation at sub-layer i is perturbed by structured, weight-dependent noise:

$$\hat{\mathbf{Y}}_i^{\text{opt}} = \mathbf{X}_i \mathbf{W}_{0,i} + \epsilon_i^{\text{fwd}} \quad (8)$$

where ϵ_i^{fwd} arises from thermally induced, spatially correlated perturbations in the MRR resonances and the resulting inter-channel crosstalk. Unlike conventional i.i.d. noise models, this perturbation is *input- and weight-dependent*, since it is governed by the local optical power and programmed de-tuning. The noisy optical output is combined with the clean electronic factor contribution $s(\mathbf{X}_i \mathbf{U}) \Sigma_i \mathbf{V}^T$ and propagated through subsequent nonlinear and residual operations, forming the input $\mathbf{X}_{i+1}$. As a result, cached intermediates (e.g., $\mathbf{X}_i \mathbf{U}$) implicitly accumulate forward noise across layers.

3.3.3 Backward-Pass Noise. The backward propagation $\mathbf{G}_i \mathbf{W}_{0,i}^T$ is executed on the same optical cores and is subject to the same structured noise:

$$\left. \frac{\partial \mathcal{L}}{\partial \mathbf{X}_i} \right|_{\text{opt}} = \mathbf{G}_i \mathbf{W}_{0,i}^T + \epsilon_i^{\text{bwd}} \quad (9)$$

where ϵ_i^{bwd} follows the same weight- and input-dependent structure as ϵ_i^{fwd} , but is statistically independent due to different operands and operating conditions. Noise accumulates along the backward chain, such that gradients at earlier layers are affected by both forward-pass corruption (in activations) and backward-pass perturbations (in gradient propagation).

	#p	C100	Cal	DTD	Flw	Pets	SVHN	Sun	Cam	Eur	Res	Ret	Avg
Full	85.9M	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	78.6
Adapter [7]	217K	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9
AdaptFormer [9]	217K	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	76.3	82.1
LoRA [5]	355K	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	81.3
NOAH [19]	421K	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	81.9
FacT-TT ₄ [8]	68K	69.4	88.5	70.6	98.8	90.0	83.3	53.7	83.9	95.1	81.5	75.4	80.9
FacT-TT ₁₆ [8]	96K	71.3	89.6	70.7	98.9	91.0	87.8	54.6	85.2	95.5	83.4	75.7	82.2
ViT-B (CL)	60K	43.4	85.7	64.7	97.1	86.6	76.4	49.3	79.3	90.3	76.4	74.3	74.9
ViT-B ₄ (FIT)	44K	67.0	88.1	68.1	98.6	89.2	86.5	51.5	83.8	95.2	82.8	75.4	80.6
ViT-B ₁₆ (FIT)	72K	67.4	88.8	68.6	98.8	89.3	89.2	51.2	86.1	95.8	84.1	76.1	81.4

Table 1: Results on the VTAB-1K benchmark under hardware-aware noise. ‘#p’ denotes the average number of trainable parameters, including both the backbone and the classification head. CL refers to classification-layer-only fine-tuning.

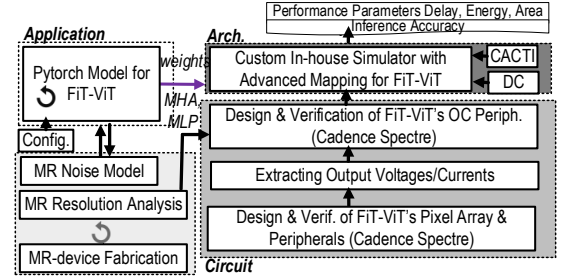


Figure 7: Proposed bottom-up evaluation framework.

3.3.4 Impact on Factor Gradients. While factor gradient computations are performed electronically, their inputs are corrupted: (i) $\mathbf{X}_i \mathbf{U}$ carries accumulated forward noise, and (ii) $\mathbf{G}_i \mathbf{V}$ carries accumulated backward noise. Consequently, the electronic unit computes exact updates on *noisy operands*, which is the primary mechanism through which structured photonic noise influences learning. All noise parameters are calibrated from measured MRR behavior and injected into both forward and backward optical computations during training, enabling realistic evaluation of noise impact on parameter-efficient fine-tuning.

4 EXPERIMENTAL RESULTS

4.1 Evaluation Framework

We develop a comprehensive bottom-up framework spanning device, circuit, architecture, and application levels (Fig. 7).

At the device level, MRRs were fabricated and calibrated to achieve 8-bit precision, with over 200 identical instances integrated on a $10 \times 10 \text{ mm}^2$ chip (Fig. 2(c)) to characterize process variations. Thermo-optic behavior and thermal crosstalk were analyzed using coupled Ansys Lumerical HEAT and FDE simulations. A silicon waveguide ($500, \text{nm} \times 220, \text{nm}$) with a heater positioned $2, \mu\text{m}$ above was modeled, with a secondary waveguide placed at separations of $2\text{--}10, \mu\text{m}$ to study spatial thermal coupling. Sweeping heater power ($0\text{--}6, \text{mW}$), we extracted temperature distributions, effective index changes (using a thermo-optic coefficient of $1.8 \times 10^{-4}, \text{K}^{-1}$), and resulting phase and wavelength shifts. Thermal crosstalk was quantified via the relative temperature rise in the secondary waveguide. These measurements were co-simulated with interface CMOS circuits in Cadence Spectre.

At the circuit level, pixel arrays and peripherals were implemented using the 45nm NCSU PDK [20], enabling extraction of input-output characteristics. Building on this, all Opto-ViT-v2 components (excluding memory) were designed and validated using

Model	#param	FGVCAircraft				OxfordPets				Food101				StanfordCars				Flowers102			
		2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16
Adapter [7]	239K	11.9	21.5	32.1	45.0	76.3	85.3	87.5	87.8	48.6	60.0	66.2	71.1	12.3	23.8	40.0	60.0	96.4	97.4	98.9	99.3
AdaptFormer [9]	239K	12.0	21.6	34.2	48.9	74.9	83.9	86.3	87.5	48.3	59.8	66.3	71.4	12.3	26.5	48.5	65.2	96.4	97.3	98.9	99.3
LoRA [5]	377K	12.1	21.8	34.5	48.7	76.2	85.6	86.4	87.4	48.6	59.5	66.1	71.4	12.3	26.4	48.2	67.8	96.5	97.3	98.7	99.4
NOAH [19]	417K	12.3	22.2	34.5	49.2	78.2	84.9	87.4	89.4	51.5	64.1	71.9	74.7	12.2	25.5	48.5	68.2	97.0	97.5	98.9	99.5
FacT-TT ₁₆ [8]	143K	15.7	24.5	36.8	50.8	82.2	87.5	89.8	90.0	53.8	64.5	71.5	75.0	16.2	29.8	52.5	74.3	90.0	97.0	98.9	99.3
ViT-B ₄ (FiT)	58K	17.2	27.1	37.6	51.1	80.8	86.3	90.1	91.0	47.8	61.4	68.8	74.1	13.9	23.6	40.7	57.8	97.1	98.1	98.8	98.9
ViT-B ₁₆ (FiT)	86K	17.2	28.3	40.4	56.1	82.4	86.8	89.4	90.0	47.8	59.8	66.8	72.6	14.1	26.7	50.0	69.5	97.6	97.8	98.7	98.9

Table 2: Few-shot results from 2 to 16 shots on FGVC benchmarks under hardware-aware noise. ‘#param’ denotes the average number of trainable parameters, including both the backbone and the classification head. FacT-TT rank-16 results are obtained under noise-free conditions, while FiT method evaluates the same setting with a 40% sparse head under hardware-induced noise.

Cadence Spectre and Synopsys Design Compiler [21], and integrated into our in-house simulator for holistic latency and energy evaluation across configurations.

At the application level, we evaluate our method on the Visual Task Adaptation Benchmark (VTAB-1K) [22] and several fine-grained visual classification (FGVC) datasets. VTAB-1K comprises 19 classification tasks with 1,000 training samples each, covering natural, specialized, and structured domains, and is widely used for evaluating parameter-efficient transfer learning under distribution shifts. We further evaluate on five FGVC benchmarks: Food-101 [23], Stanford Cars [24], Oxford Flowers-102 [25], FGVC-Aircraft [26], and Oxford-IIIT Pets [27], following the standard few-shot protocol. In order to report the top-1 accuracy of them, all models are initialized from an ImageNet-21K pretrained ViT-Base, with backbone weights quantized to 8-bit and kept frozen; only the proposed FacT parameters and classification head are updated.

To evaluate hardware-aware robustness, we inject structured photonic noise during both forward and backward propagation. Following the thermo-optic crosstalk model in Section 3.3, weight perturbations are obtained via a lookup-table (LUT) that maps programmed weights to their *input- and weight-dependent* resonance shifts. This captures the deterministic component of thermal drift arising from optical power and spatial coupling. On top of this, we model residual stochastic variations as zero-mean Gaussian noise with standard deviation $\sigma = 0.12$, consistent with our post-trimming device-level measurements. Input activations are similarly perturbed to reflect optical intensity fluctuations, using Gaussian noise with $\sigma = 0.03$. These perturbations are propagated through both forward and backward passes, ensuring that gradients are computed under the same structured, hardware-consistent noise conditions. For fine-tuning, we use AdamW with a learning rate of 5×10^{-3} and weight decay of 10^{-4} . A cosine schedule is applied with a 5-epoch warmup at 10^{-6} , followed by 95 epochs of training.

4.2 Opto-ViT-v2 Results Analysis

Table 1 and Table 2 present the performance of Opto-ViT-v2 on VTAB-1K and FGVC benchmarks under hardware-aware photonic noise. The results show that Opto-ViT-v2 maintains strong adaptation capability while operating under strict parameter and hardware constraints.

Accuracy analysis. Opto-ViT-v2 with rank-16 achieves competitive performance across both VTAB-1K and FGVC tasks, with accuracy comparable to state-of-the-art PEFT methods. This indicates that the proposed tensorized low-rank adaptation retains most of the representational capacity of full fine-tuning, even under structured photonic noise. The rank-4 configuration shows a moderate drop in accuracy, but still consistently outperforms the classification-layer-only baseline. Nevertheless, the results demonstrate that a small number of shared factors is sufficient to capture task-specific adaptation because it still have comparable result under the tested VTAB-1K datasets

On FGVC benchmarks, Opto-ViT-v2 performs particularly well in low-shot regimes, retaining the few-shot accuracy compared to FacT-TT’s result. It suggests that the structured low-rank parameterization together with sparse updates introduces an implicit regularization effect. This helps prevent overfitting when training data is limited, while still enabling effective adaptation.

Parameter efficiency and robustness. Opto-ViT-v2 achieves these results with substantially fewer trainable parameters than prior methods. While existing approaches require 200K–300K parameters, Opto-ViT-v2 operates with only 44K–72K parameters, corresponding to a $3\times$ – $8\times$ reduction.

Importantly, the total training cost is not dominated by the backbone alone, but also by the classification head. In many VTAB and FGVC tasks, the head accounts for the majority of trainable parameters. The sparse head of Opto-ViT-v2 helps remove low-importance parameters and reduces the head size by approximately 40% with minimal impact on accuracy. This significantly lowers both memory footprint and update cost, making the method more suitable for on-chip training. Meanwhile, Opto-ViT-v2 remains robust under hardware noise. By constraining updates to a low-dimensional factor space and eliminating unnecessary head updates, the method reduces sensitivity to noisy gradients and activations. As a result, Opto-ViT-v2 maintains accuracy close to noise-free baselines despite the presence of structured photonic perturbations.

Sparse Head and Noise Ablation Study. The effect of sparse-head design and hardware noise is further validated by Table 3. Increasing the retaining ratio from 0.4 to 0.6 improves accuracy, while further increasing to 0.8 provides only marginal gains, indicating that most useful parameters are concentrated in a subset of the head. This confirms that the proposed 0.6 retaining ratio strategy effectively reduces parameter count.

Effect of Head Retaining Ratio							
Retaining ratio	# Head Params	CIFAR100	Pets	RESISC45	Camelyon	CLEVR-Count	KITTI
0.4	10K	65.5	89.2	80.6	82.9	74.4	77.9
0.6	15K	67.0	89.2	82.8	83.8	73.5	78.2
0.8	20K	67.9	90.0	83.7	83.6	73.4	78.3

Effect of Fabrication Noise σ_{fab}							
σ_{fab}	# Head Params	CIFAR100	Pets	RESISC45	Camelyon	CLEVR-Count	KITTI
0.1		67.0	89.2	82.8	83.8	73.5	78.2
0.2	15K	64.2	88.5	81.2	82.4	69.6	77.4
0.3		58.1	85.5	78.3	81.8	62.0	76.1

Table 3: Ablation study on head retaining ratio and fabrication noise in several VTAB1K datasets. For sparse head analysis, σ_{fab} is fixed at 0.1. For noise analysis, retaining ratio is fixed at 0.6. “# Head Params” denotes the average number of trainable classifier-head parameters.

As fabrication noise increases, accuracy degrades across all tasks, with more pronounced drops on complex datasets. This reflects the impact of noise on both forward activations and backward gradients. Nevertheless, Opto-ViT-v2 degrades gracefully, demonstrating that the proposed low-rank and sparse-update design improves stability under realistic hardware conditions.

4.3 Energy and Latency Analysis

We evaluate one full training epoch for a ViT-Base-style backbone under the VTAB-1K setting using the proposed customized evaluation framework designed to capture forward propagation, backward propagation, gradient/factor computation, activation caching or recomputation, and parameter update/write-back overheads. We assume a ViT-Base-like configuration with embedding dimension $d = 768$, depth $L = 12$, and $12L = 144$ trainable backbone matrices after tensorization into the unified sub-layer representation used by FacT-style factorization. For VTAB-1K, we assume 1000 training samples per task and task-specific classifier dimensions C , while keeping the same backbone and hardware configuration across tasks. The number of optimization steps per epoch is therefore $\lceil 1000/B \rceil$, where B is the batch size.

For each model, we decompose epoch cost into the following phases:

$$E_{\text{epoch}} = E_{\text{fwd}} + E_{\text{bwd}} + E_{\text{grad/update}} + E_{\text{update/tune}} + E_{\text{mem/recompute}} + E_{\text{head}} \quad (10)$$

$$T_{\text{epoch}} = T_{\text{fwd}} + T_{\text{bwd}} + T_{\text{grad/update}} + T_{\text{update/tune}} + T_{\text{mem/recompute}} + T_{\text{head}} \quad (11)$$

These terms are accumulated over all optimization steps in one epoch.

Compared Architectures. We evaluate four photonic training architectures along with an FPGA baseline, differing in how matrix multiplications and parameter updates are mapped across optical and electronic domains. In the **[O:E]** model, forward and backward matrix multiplications are executed optically, while full-rank gradient computation and weight updates are performed electronically, avoiding optical write-back but incurring large gradient and memory costs. The proposed **[O:E_L]** model extends this design by adopting a low-rank factorization, where only compact factors are trained electronically, significantly reducing gradient dimensionality, activation storage, and update overhead. In contrast, the **[O:O]** model represents a fully optical training approach in which

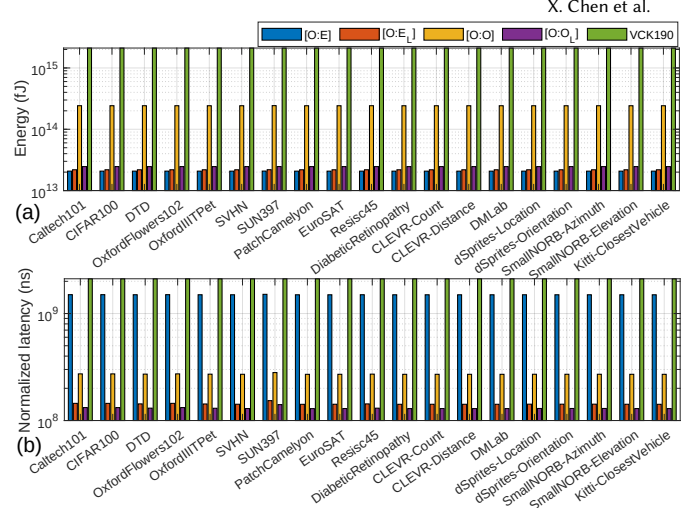


Figure 8: Per-task (a) total epoch energy and (b) Normalized total latency.

both propagation and full-rank weight updates are implemented in optical cores, requiring extensive microring tuning and write-back operations, leading to high update energy and latency. The **[O:O_L]** model combines optical execution with low-rank training, keeping pretrained weights fixed on optical cores while updating only a small set of low-rank parameters. Finally, we include a Xilinx VCK190 FPGA baseline, using its reported energy efficiency (1.42 KFPS/W) for total energy comparison.

Results. As shown in Fig. 8, the fully optical model **[O:O]** incurs the highest cost due to updating the entire backbone in the optical domain. The number of tunable parameters scales as $12Ld^2 \approx 8.5 \times 10^7$, resulting in $\sim 1.7 \times 10^8$ optical write operations per step (including W and W^T), making update/tuning the dominant energy and latency bottleneck. In contrast, the low-rank design **[O:O_L]** reduces trainable parameters to $2dr + 12Lr^2 = 8448$ ($d=768, r=4$), i.e., $\sim 10^4 \times$ fewer parameters, and correspondingly reduces programming events to $\sim 1.7 \times 10^4$, yielding orders-of-magnitude savings in update energy and latency. A similar trend holds for electronic-update variants. While **[O:E]** avoids optical write-back, it still incurs full-rank gradient computation and large activation storage. The proposed **[O:E_L]** reduces gradient complexity from $O(d^2)$ to $O(dr)$ and shrinks cached activations by $\sim 192 \times$ (from 2.17×10^7 to 1.13×10^5 elements), leading to consistently lower energy and latency. Across VTAB tasks, variation is small since costs are dominated by the shared ViT backbone. Overall, these results show that weight updates—not forward/backward passes—are the primary bottleneck in photonic training, and that low-rank factorization is key to scalable efficiency.

The average epoch energy breakdown in Fig. 9(a) shows that forward and backward propagation dominate due to shared optical execution. However, the **Update/Tuning** phase exhibits the largest variation. The fully optical **[O:O]** design incurs the highest update energy from $\sim 1.7 \times 10^8$ programming events per step, while **[O:O_L]** reduces this to $\sim 1.7 \times 10^4$, achieving an $\sim 10^4 \times$ reduction. Electronic-update models eliminate optical tuning, with **[O:E_L]** further lowering gradient energy via low-rank factorization. Memory/recomputation energy is also reduced due to a $\sim 192 \times$

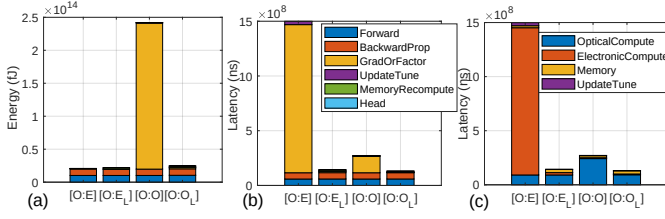


Figure 9: Average epoch (a) phase and (b) Latency by phase and (c) Latency by resource.

decrease in cached activations. Latency trends in Fig. 9(b) mirror energy, with forward/backward passes dominating the baseline. However, **Update/Tuning** dominates latency in full optical training. **[O:O]** suffers from sequential programming scaling with $O(d^2)$, while **[O:O_L]** reduces this to only 8448 parameters, yielding orders-of-magnitude lower latency. Electronic-update models **[O:E]** and **[O:E_L]** avoid optical programming entirely, achieving the lowest update latency, with **[O:E_L]** benefiting from reduced gradient complexity. From a resource perspective, optical computation dominates forward/backward latency across all models. However, **Update/Tuning** is the primary bottleneck in **[O:O]** due to $\sim 10^8$ writes, while **[O:O_L]** reduces this by $\sim 10^4\times$. Electronic computation remains modest but increases in **[O:E]** and **[O:E_L]** due to gradient updates. Memory latency is also reduced in low-rank models, benefiting from the $\sim 192\times$ reduction in activation storage.

5 CONCLUSION

This work presents Opto-ViT-v2, a hardware–algorithm co-designed framework for parameter-efficient on-chip fine-tuning on silicon-photonics ViT accelerators. By combining tensorized low-rank adaptation with a gradient-accumulated sparse head, FiT-ViT significantly reduces training parameters while preserving adaptation capability. Opto-ViT-v2 achieves competitive accuracy with fewer parameters and remains robust under structured photonic noise, while reducing activation storage and eliminating costly weight write-back. This demonstrates that low-dimensional and selective updates enable stable and efficient on-chip learning on photonic hardware.

REFERENCES

- [1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.
- [2] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. CrossLight: A cross-layer optimized silicon photonic neural network accelerator. In *Proc. Design Automation Conference (DAC)*, pages 1069–1074, 2021.
- [3] Mehrdad Morsali, Brendan Reidy, Deniz Najafi, Sepehr Tabrizchi, Mohsen Imani, Mahdi Nikdast, Arman Roohi, Ramtin Zand, and Shaahin Angizi. Lightator: An optical near-sensor accelerator with compressive acquisition enabling versatile image processing. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024.
- [4] Mehrdad Morsali, Chengwei Zhou, Deniz Najafi, Sreetama Sarkar, Pietro Mercati, Navid Khoshavi, Peter Beerel, Mahdi Nikdast, Gourav Datta, and Shaahin Angizi. Opto-ViT: Architecting a near-sensor region of interest-aware vision transformer accelerator with silicon photonics. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, 2025.
- [5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. International Conference on Learning Representations (ICLR)*, 2022.
- [6] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proc. European Conference on Computer Vision (ECCV)*, 2022.
- [7] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proc. International Conference on Machine Learning (ICML)*, 2019.
- [8] Shibo Jie and Zhi-Hong Deng. FacT: Factor-tuning for lightweight adaptation on vision transformer. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 1060–1068, 2023.
- [9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [10] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022.
- [11] Farzaneh Zokaee, Qian Lou, Nathan Youngblood, Weichen Liu, Yuan Xie, and Lei Jiang. LightBulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks. In *Proc. Design, Automation and Test in Europe Conference (DATE)*, pages 1438–1443, 2020.
- [12] Weichen Liu, Wenyang Liu, Yun Ye, Qian Lou, Yuan Xie, and Lei Jiang. HolyLight: A nanophotonic accelerator for deep learning in data centers. In *Proc. Design, Automation and Test in Europe Conference (DATE)*, pages 1483–1488, 2019.
- [13] Febin P. Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. ROBIN: A robust optical binary neural network accelerator. *ACM Transactions on Embedded Computing Systems (TECS)*, pages 1–24, 2021.
- [14] Salma Afifi, Sajjad Moazeni, and Sudeep Pasricha. TRON: Transformer neural network acceleration with non-coherent silicon photonics. In *Proc. Great Lakes Symposium on VLSI (GLSVLSI)*, pages 15–21, 2023.
- [15] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proc. European Chapter of the Association for Computational Linguistics (EACL)*, 2021.
- [16] Malte J Rasch, Fabio Carta, Omobayode Fagbohunge, and Tayfun Gokmen. Fast and robust analog in-memory deep neural network training. *Nature Communications*, 15(1):7133, 2024.
- [17] Xiaoxuan Yang, Bonan Yan, Hai Li, and Yiran Chen. ReTransformer: ReRAM-based processing-in-memory architecture for transformer acceleration. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, 2020.
- [18] Xuming Chen, Deniz Najafi, Chengwei Zhou, Pietro Mercati, Arman Roohi, Mohsen Imani, Mahdi Nikdast, Shaahin Angizi, and Gourav Datta. Light-bound transformers: Hardware-anchored robustness for silicon-photonics computer vision systems. *arXiv preprint arXiv:2604.04330*, 2026.
- [19] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022.
- [20] North Carolina State University. FreePDK45: An open-source predictive process design kit. <https://eda.ncsu.edu/freePDK/freePDK45/>, 2011. Accessed: 2026.
- [21] Synopsys, Inc. Synopsys design compiler, product version 14.9.2014, 2014.
- [22] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. The visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.
- [23] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.
- [24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2013.
- [25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2008.
- [26] Subhransu Maji, Juho Kannala, Esa Rahtu, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. Technical report, arXiv preprint arXiv:1306.5151, 2013.
- [27] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.