

Sparse coherent photonic processor for solving eigenmode problems

Zheyuan Zhu^{1*}, Andrew B. Klein¹, Dewan Saiham¹, Guifang Li, and Shuo Pang¹

1. CREOL, The College of Optics and Photonics, University of Central Florida, 4304 Scorpius St., Orlando, FL, USA, 32816

zyzhu@knights.ucf.edu

Abstract—We present a photonic processing primitive based on coherent mixing with high-speed programmable electronic interface for sparse matrix-vector multiplications. The precision of the photonic primitive is sufficient for solving waveguide eigenmodes, which has been verified on an FPGA emulator of the primitive.

Keywords—analog computing, matrix multiplication, eigen solver, photonic processing.

I. INTRODUCTION

In recent years, the demand for computing power in artificial intelligence (AI)-enabled industrial applications has spurred the research in fast and high-efficiency photonic computing platforms. These platforms typically consist of photonic memories and passive interconnects, which are designed to match the topologies of various artificial neural network (ANN) structures, including fully connected, convolutional, and recurrent ANNs [1]–[3]. Analog photonic networks with pre-trained, limited-precision weights have demonstrated the potential for lower power consumption than electronic counterparts in inference tasks, due to the intrinsically passive photonic multiply-accumulate (MAC) processes with static weights [4].

However, existing photonic processors lack the flexibility for performing basic linear algebra subprograms (BLAS), which are common in numerical analysis and scientific computing. These routines typically require dynamically changing weight matrices, which pose challenges in speed and power consumption when updating them on a photonic-memory-based computing platform [5]. In addition, most BLAS routines in scientific computing, such as solving partial differential equations, involve large yet sparse matrices. Converting the large sparse matrix with millions of elements into a dense matrix in photonic memory is not only uneconomical, but in most cases, impossible. Here we present a coherent photonic processing primitive with direct modulations on both the inputs and the weights. A peripheral electronic interface with custom firmware provides flexibility for performing real-time BLAS routines with support for sparse matrix. In this work, we demonstrate the operation of our analog photonic primitive, and its potential in solving eigenmodes in waveguide structures using the data format compatible with the analog photonic primitive.

II. COHERENT PHOTONIC PROCESSING PRIMITIVE

The core building block of the coherent photonic processing primitive, shown in Fig. 1 (a), is an interference and coherent detection unit. The electric fields representing the weight and input are generated by their respective electro-optic modulators

(JDSU OC-192), coherently mixed, and detected by a pair of balanced photodiodes (Thorlabs BX1BA). The output differential photocurrent is proportional to the product $E_A E_X$. The coherent photonic primitive can additionally incorporate a multitude of dimensions available in optical signals, including wavelength, spatial modes, time, and beam locations, to parallelize and accumulate the scalar multiplications.

The photonic primitive is controlled through high-speed electronic interface to provide the capability of deploying various computing applications in real time. Figures 1 (b, c) show an example of the primitive running in time-division multiplexing (TDM) mode in conjunction with a custom FPGA firmware for streaming the input matrix / vector elements and accumulating the partial sums. In this example running at 0.5GBd symbol rate, the photonic primitive accepts a stream of signed inputs (Fig. 1 (b)) within $[-2, 2]$, and calculates the partial sums ranging from -4 to 4 . The statistics (Fig.1 (c)) of all 1600 output symbols in this stream exhibits good linearity among all possible output levels. The signal-to-noise ratio of the maximum output level is 19.5, indicating that more than 16 levels within the dynamic range of the photonic processor can be accurately digitized.

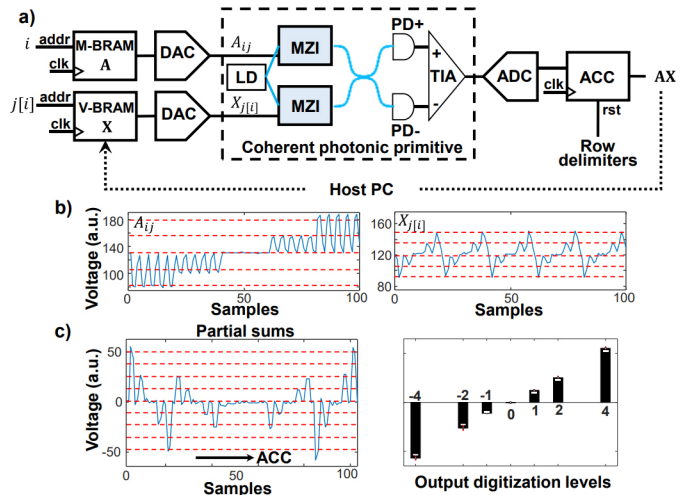


Fig. 1. (a) A coherent photonic primitive for real-time sparse matrix-vector multiplication. (b) Experimental input streams representing the matrix and vector elements. (c) Partial sums and their statistics from the primitive's output. The error bars indicate the standard deviation of each output digitization level within 1600 symbols.

TDM provides the flexibility to avoid encoding and caching the full matrix as optical signals in sparse matrix-vector multiplication. The vector $\mathbf{X}$ and the non-zero elements in the

large sparse matrix $\mathbf{A}$ are loaded onto separate block memory (BRAM) regions, M-BRAM, and V-BRAM, inside the FPGA. The index, $j[i]$, of the non-zero element $A[j[i]]$ in the sparse matrix drives the address port of V-BRAM to produce an input stream of corresponding vector elements $X[j]$. The FPGA accumulates the digitized partial sums and resets the accumulator at the end of each matrix row.

III. EIGENMODE SOLVER ON PHOTONIC PRIMITIVE

In optical waveguides, the electric field E of a transverse electric (TE) or transverse magnetic (TM) eigenmode satisfies the following characteristic equations [6]

$$\begin{aligned} \frac{1}{k_0^2} \frac{\partial}{\partial x} \left(\frac{1}{\varepsilon_r} \frac{\partial}{\partial x} (\varepsilon_r E) \right) + \frac{1}{k_0^2} \frac{\partial^2}{\partial y^2} E + \varepsilon_r E &= \bar{\beta}^2 E \text{ (TE)} \\ \frac{1}{k_0^2} \frac{\partial^2}{\partial x^2} E + \frac{1}{k_0^2} \frac{\partial}{\partial y} \left(\frac{1}{\varepsilon_r} \frac{\partial}{\partial y} (\varepsilon_r E) \right) + \varepsilon_r E &= \bar{\beta}^2 E \text{ (TM)} \end{aligned} \quad (1)$$

Here ε_r is the relative permittivity, k_0 is the wavenumber in vacuum, and $\bar{\beta}$ is the effective index of the mode. After discretizing the cross section of the waveguide structure with grid size Δx and Δy along the horizontal and vertical directions respectively, both characterization equations for TE and TM mode (1) can be expressed in discrete form as

$$\mathbf{A}(\varepsilon_r) \mathbf{E} = \bar{\beta}^2 \mathbf{E}. \quad (2)$$

Here the matrix $\mathbf{A}(\varepsilon_r)$ is a second-order finite difference operator in 2D that depends on the distribution of ε_r .

The fundamental eigenmode that has the highest effective refractive index $\bar{\beta}$ can be solved using a fixed-point power iteration algorithm [7] tailored to the coherent photonic primitive, shown as Algorithm I. Starting with a random and quantized initial guess $\mathbf{E}_0$, the algorithm recurrently calculates the fixed-point sparse matrix-vector multiplications with signed 8-bit or lower-precision integer format, targeting the available dynamic range of the photonic primitive. The normalization step in the vanilla power iteration is substituted with the adaptive exponent adjustment step [8] on host PC to prevent overflow elements after each matrix-vector multiplication.

Algorithm I: Fixed-point power iteration for solving eigenmodes

Initialize $\mathbf{E}_0$ with random values

for $k=1$ to K :

$\mathbf{E}_k \leftarrow \mathbf{A} \mathbf{E}_{k-1}$ //fixed-point FPGA or photonic primitive

$e_k = \text{ceil}(\log_2(\max|\mathbf{E}_k|))$ //host PC

Adjust the exponent of $\mathbf{E}_k$ to e_k

Prior to the deployment of eigenmode solver on photonic hardware, we first emulated the fixed-point coherent sparse photonic primitive using DSP and BRAM slices available on an AMD Zynq-7000 system-on-chip device. Additional FIFO and transceiver blocks are included in FPGA firmware to handle PCIe communication with a host PC, following the architecture in [8].

The fixed-point eigenmode solver was tested with a silicon-on-oxide (SOI) waveguide structure at $\lambda=1550\text{nm}$, shown in Fig. 2 (a). The Si waveguide ($\varepsilon_r = 12$) is 250nm in thickness, 450nm in width, and is sandwiched between SiO_2 ($\varepsilon_r = 2.1$) and air. With a grid size $\Delta x = \Delta y = 25\text{nm}$, the entire solution region

consists of 48×64 pixels. The matrix $\mathbf{A}$ is 3072×3072 in size with 15136 non-zero elements, averaging ~ 5 elements per row due to the use of 5-stencil finite difference operator.

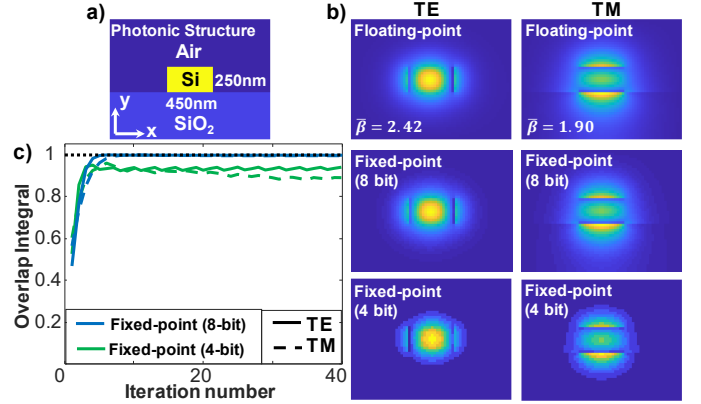


Fig. 2. Fixed-point eigenmode solutions for a photonic structure in (a). (b) TE and TM modes from MATLAB floating-point eigen solver, and fixed-point eigen solvers (8-bit and 4-bit) on FPGA emulator. (c) Overlap integral as a function of the power iteration step.

Figure 2 (b) shows the fundamental TE and TM modes from MATLAB's double-precision floating-point eigen solver "eigs", which we considered as the ground truth, as well as the solutions from fixed-point eigen solver on FPGA emulator with 8- and 4-bit precisions. We evaluate the accuracy of the eigenmode solution using overlap integral η between the fixed-point $\hat{\mathbf{E}}$ and ground truth $\mathbf{E}_t$, defined in

$$\eta = \frac{|\sum \mathbf{E}_t^* \hat{\mathbf{E}}|^2}{\sum |\mathbf{E}_t|^2 \sum |\hat{\mathbf{E}}|^2}. \quad (3)$$

Fig. 2 (c) plots the overlap integral as a function of iteration step k for both TE and TM mode solutions. Even with signed 4-bit precision, the fixed-point eigenmode solutions are reasonably close to the analytical ones with 94.7% for TE mode, and 93.2% for TM mode, in overlap integral.

IV. SUMMARY

We have demonstrated a flexible coherent photonic processing primitive for large-scale sparse matrix-vector multiplications. We have verified the deployment of an eigenmode solver using signed 4-bit fixed-point format that can be supported by the dynamic range (from -16 to 16) of the coherent sparse photonic primitive.

REFERENCES

- [1] J. Feldmann *et al.*, Parallel convolutional processing using an integrated photonic tensor core, *Nature*, 589, 52–58 (2021)
- [2] A. N. Tait *et al.*, Silicon Photonic Modulator Neuron, *Phys. Rev. Appl.*, 11, 064043 (2019)
- [3] R. Hamerly *et al.*, Large-Scale Optical Neural Networks Based on Photoelectric Multiplication, *Phys. Rev. X*, 9, 021032 (2019)
- [4] B. J. Shastri *et al.*, Photonics for artificial intelligence and neuromorphic computing, *Nat. Photonics*, 15, 102–114 (2021)
- [5] C. Rios *et al.*, In-memory computing on a photonic platform, *Sci. Adv.*, 5, eaau5759 (2019)
- [6] M. S. Wartak, *Computational photonics: an introduction with MATLAB* 9781107005, (Cambridge University Press, 2013)
- [7] J. W. Demmel, *Applied Numerical Linear Algebra* (Society for Industrial and Applied Mathematics, 1997)
- [8] Z. Zhu *et al.*, Fixed-point iterative linear inverse solver with extended precision, *Sci. Rep.*, 13, 5198 (2023)