Time-multiplexed Weight Sharing of Photonic Neural Networks

Jiawei Zhang,[1,*] Eli A. Doris,[1] Weipeng Zhang,[1] Yusuf O. Jimoh,[1] Bhavin J. Shastri,[2] and Paul Prucnal[1]

[1] *Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA*
[2] *Department of Physics, Engineering Physics & Astronomy, Queen's University, Kingston, Ontario K7L 3N6, Canada*
[*] *jiawei.zhang@princeton.edu*

Abstract: Silicon photonic neural networks often rely on thermal tuning of weights, limiting scaling for deep learning tasks. We demonstrate a time-domain weight sharing method that can reduce reconfigurations and energy by 6.6x without compromising accuracy. © 2024 The Author(s)

Integrated photonic neural networks (PNNs) have promised superior performance for deep learning tasks, in terms of low latency, high bandwidth, and energy efficiency [1]. Photonic tensor core processors with large scale, rapid weight updates are highly desirable. However, most state-of-the-art silicon PNNs still rely on slow thermal index-tuning mechanisms for weight configuration, as this method provides the widest tuning range and effectively compensates for fabrication variations [2]. Additionally, executing on-chip matrix-vector multiplication (MVM) operations typically requires iterative reconfiguration of all weights since the practical maximum size of physically-implemented matrices is often much smaller than those encountered in deep learning tasks. This imposes significant latency and energy overhead, as thermally reconfiguring weights takes much longer than the optical signal bandwidth capacity. Recent advancements have been made in alternative material platforms such as thin-film lithium niobate [3] and dual thermal-PN weights [4], but these either rely on technologies that are much less mature than silicon photonics or require doubling electrical I/O and control complexity. Consequently, saving latency and energy in this regime is still of practical interest.

Here, we propose and demonstrate a time-domain weight sharing approach that reduces the need for weight reconfigurations within a classification. In this work, we specifically focus on implementing dense MVMs on microring resonators (MRRs). In MRR-based PNNs, MVMs are implemented by modulating input data onto different optical wavelengths, thermally tuning MRR arrays to perform weighting, and summing via photodetectors [5]. Due to space and control limitations, the size of on-chip MRR weight banks (denoted as $m \times n$) are typically significantly smaller than the weight matrices in most NN fully-connected (FC) layers, requiring iterative reconfiguration of the MRR weight matrix ($\mathbf{W}_{\mathrm{MRR}} \in \mathbb{R}^{m \times n}$) at a slow timescale ($\sim$ ms) for each input vector $\mathbf{x}_i$ (as shown in Fig. 1a). This is referred to as time-domain multiplexing (TDM). In contrast, our approach significantly "compresses" the weight matrix in the FC layer by enforcing randomly selected weights to share the same values (denoted as $\mathbf{W}'_{\mathrm{MRR}}$) [6]. As shown in Fig. 1b, this can reduce reconfigurations when the MRR weight matrix is fixed to be $\mathbf{W}'_{\mathrm{MRR}}$ and the input data are rearranged ($\mathbf{x}_i \rightarrow \mathbf{x}'_i$) such that those sharing the same set of weights are sent through sequentially. Since the signals are modulated on a much faster timescale ($\sim$ ns), reducing reconfigurations in this way cuts down on the dominant time expenditure and therefore reduces total classification energy.

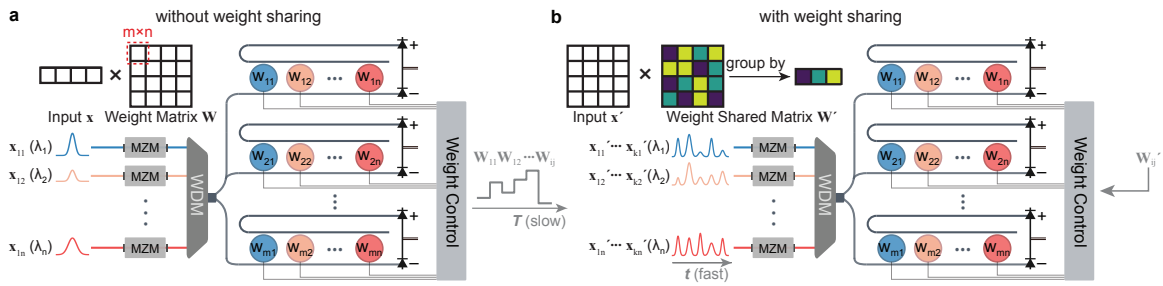

Fig. 1. Implementation of MVMs on MRR-based PNNs. **a** Without weight sharing. **b** With time-multiplexed weight sharing approach. MZM: Mach-Zehnder modulator. WDM: wavelength division multiplexer.

Our approach is validated on a two-layer convolutional neural network (CNN) for standard Modified National Institute of Standards and Technology (MNIST) classification dataset, which contains 60,000 training images and 10,000 testing images of 28×28 pixels. As shown in Fig. 2a, our CNN comprises of a convolution layer with three 3×3 kernels (activated by the ReLU function), a 2×2 max-pooling layer, and an FC layer with a size of 507×10. The 507×10 weight matrix in the FC layer is decomposed into $169 \times 5 = 845$ sub-matrices to be fitted into a 2×3 MRR-based PNN. Notably, the large weight matrix is effectively compressed by our weight sharing approach, with the compression ratio γ denoted as the ratio between its original size and the compressed size. Fig. 2b shows our experimental setup, where the time-multiplexed input signals are sent into the 2×3 MRR weight bank by an arbitrary waveform generator (Keysight M8196a). The weights are pre-trained on software, and mapped onto the reconfigurable MRRs by applying tuning currents via co-packaged current sources (Analog Device LTC2662) for the metal heaters on top of MRRs. Subsequently, the results of these decomposed MVMs are captured by on-chip photodetectors and accumulated digitally by Python-coded software.

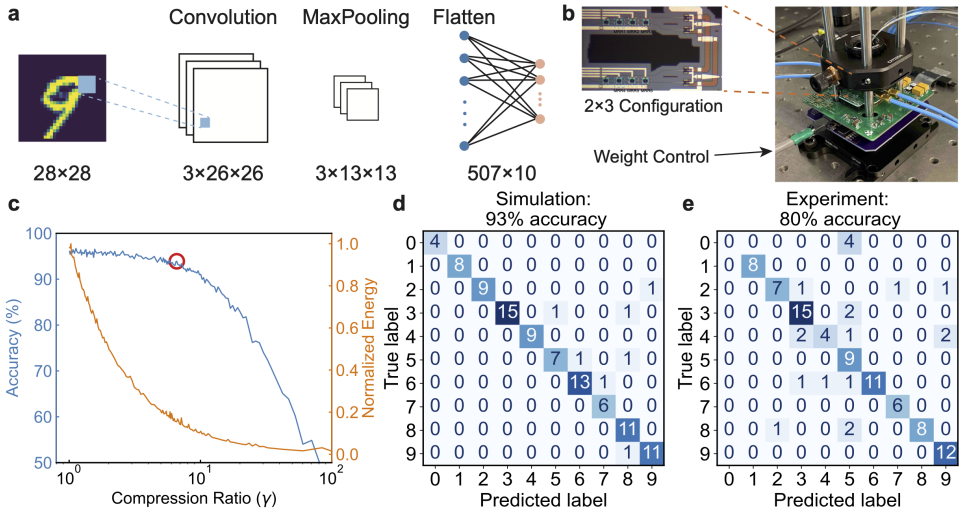

Fig. 2. Demonstration on MNIST dataset. **a** Two-layer CNN architecture. **b** Experimental setup for implementing the "compressed" FC layer. **c** Simulation result of testing accuracy, normalized energy consumption versus compression ratio γ. **d** Simulated, **e** Experimentally measured confusion matrices for 100 sample MNIST images when the compression ratio $\gamma = 6.6$.

We first simulate the tradeoff between the compression ratio γ and testing accuracy, repeating each 10 times to ensure consistency. As illustrated in Fig. 2c, the FC layer achieves significant compression while maintaining accuracy. As anticipated, the classification energy decreases linearly with fewer reconfiguration iterations. For instance, when $\gamma = 6.6$, the MRR reconfigurations are reduced from 845 times to 128 times, resulting in an 84.2% reduction in classification energy while preserving a 93% accuracy in simulation (Fig. 2d) and 80% accuracy (Fig. 2e) in experiments. The performance drop from simulation to experiment is primarily attributed to inaccurate weight mapping caused by offline training on software, which can be mitigated by diverse approaches [7] beyond the scope of this work.

In conclusion, our proposed approach significantly reduces required weight reconfigurations, leading to lower classification energy in PNNs. We envision that our proof-of-concept demonstrations will provide a foundational methodology to address the latency and energy challenges associated with the limited MVM scale in photonic computing. Future work will explore integrating this method with on-chip III-V lasers and charge integrators, as well as evaluating the compatibility of time-multiplexing with processing RF analog signals.

References

1. Shastri, B., *et al.*, *Nat. Photon.* **15**, 102–114 (2021)
2. Huang, C., *et al.*, *APL Photonics* **5** (2020)
3. Lin, Z., *et al.*, *Nat. Commun.* **15**, 9081 (2024)
4. Zhang, W., *et al.*, *CLEO* 1–2 (2024)
5. Tait, A., *et al.*, *JSTQE* **22**, 312–325 (2016)
6. Chen W., *et al.*, *ICML*, 2285–2294 (2015)
7. Xu, T., *et al.*, *Optica* **11**, 1039–1049 (2024)