

# Reduced rank photonic computing accelerator

SAMARTH AGGARWAL,<sup>1</sup>  BOWEI DONG,<sup>1</sup> JOHANNES FELDMANN,<sup>1,2</sup> NIKOLAOS FARMAKIDIS,<sup>1</sup>   
WOLFRAM H. P. PERNICE,<sup>3</sup> AND HARISH BHASKARAN<sup>1,\*</sup> 

<sup>1</sup>Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, UK

<sup>2</sup>Sallience Labs, Oxford, UK

<sup>3</sup>Kirchhoff-Institute for Physics, Heidelberg University, Heidelberg, Germany

\*harish.bhaskaran@materials.ox.ac.uk

Received 17 January 2023; revised 23 June 2023; accepted 6 July 2023; published 4 August 2023

Use of artificial intelligence for tasks such as image classification and speech recognition has started to form an integral part of our lives. Facilitation of such tasks requires processing a huge amount of data, at times in real time, which has resulted in a computation bottleneck. Photonic cores promise ultra-fast convolutional processing by employing broadband optical links to perform parallelized matrix–vector multiplications (MVMs). Yet the scalability of photonic MVMs is limited by the footprint of the system and energy required for programming the weights, which scale with the matrix dimensionality ( $M \times N$ ). One approach is to reduce the number of hardware matrix weights required, which would allow for less aggressive scaling of the hardware. In this paper, we propose and experimentally demonstrate precisely such a hardware photonic architecture with reduced rank of operation, significantly improving on scalability and decreasing the system complexity. We employ the reduced photonic matrix with reconfigurable optical weights in image processing tasks where we demonstrate the ability to achieve edge detection and classification with 33% reduction in the conventional  $3 \times 3$  kernel matrix and with no detectable loss of accuracy. While our demonstration is in photonics, this architecture can be universally adapted to MVM engines, and offers the potential for fast, scalable computations at a lower programming cost.

Published by Optica Publishing Group under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

<https://doi.org/10.1364/OPTICA.485883>

## 1. INTRODUCTION

Deep convolutional neural networks (DNNs) have been extensively applied in complex tasks including speech recognition [1], object classification, and image filtering [2–5]. However, such convolutional neural networks (CNNs) are computationally intensive and take up the majority of memory and power consumption. This is particularly problematic in traditional von Neumann computing architectures, which are limited by the shuttling efficiency between the memory and processing units. To circumvent this limitation, various in-memory computing hardware designs have been proposed to improve data movement [6–8]. With Moore's law slowing down, photonic in-memory computational cores have been demonstrated with high data throughput and energy efficiency superior to their electronic counterparts [9–17]. However, current photonic implementations face technological challenges related to the scalability of kernel sizes, which is required to implement modern CNN models.

Over the past few decades, various powerful CNN models such as ResNet50 and ResNet18 with tens of millions of trainable parameters have been proposed. These large CNN models require large scale matrix–vector multiplication (MVM) operations. However, such CNNs are usually over-parameterized [18], and therefore, the redundancy can be reduced by rank reduction

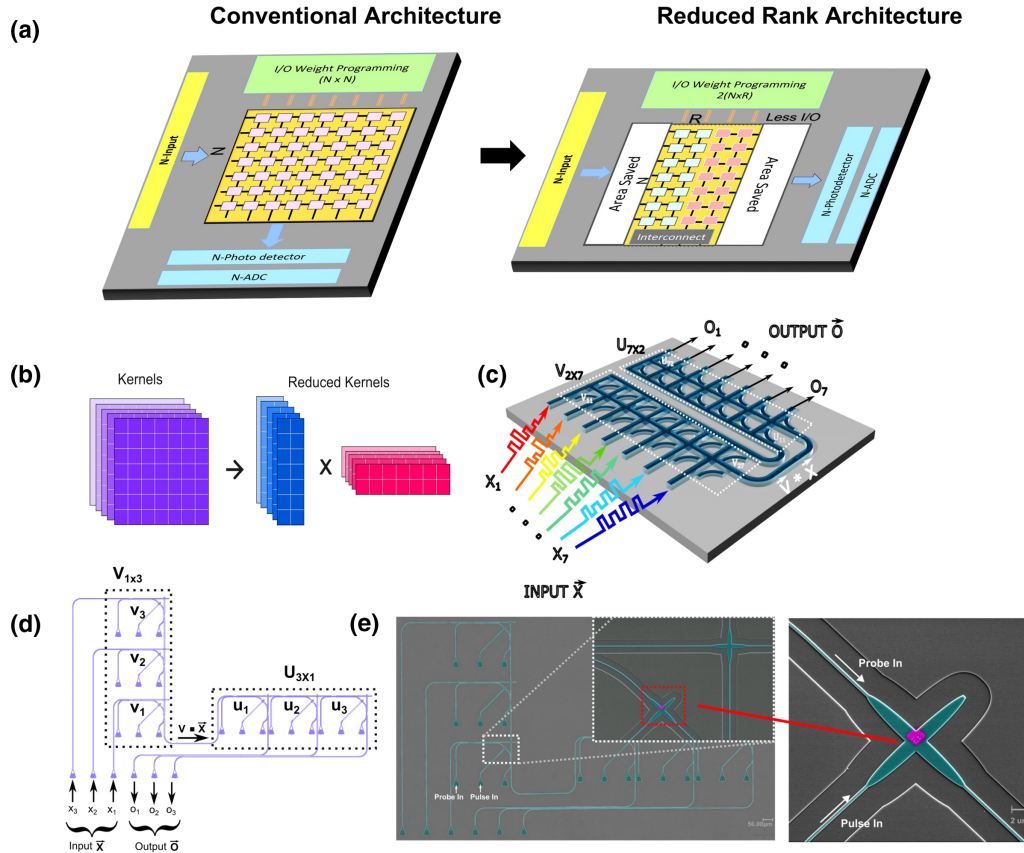
operations such as matrix factorization. While such rank reduction operations have already been shown to achieve highly efficient CNN tasks in software [19–21], hardware implementations of such rank reduction methods are power hungry with huge footprints [22–24]. Such implementations have been demonstrated using traditional Mach–Zehnder interferometer (MZI) mesh networks [25,26]. However, such MZI mesh designs use large phase shifters. Such phase shifters allow a wide range of applications but at the cost of a larger footprint.

In this work, we combine the benefits of high throughput, high computational density photonic in-memory computing, and computational redundancy reduction of low rank matrix factorization to propose a novel MVM accelerator hardware, thereby overcoming the scalability challenges. We implement this on a silicon photonic platform using phase change photonic memory cells to achieve image filtering and image classification tasks and achieve accuracy close to software-based implementations.

## 2. METHODS

### A. Architecture Design

We introduce our reduced rank architecture in Fig. 1(a). Conventional MVM photonic accelerators (size  $N \times N$ ) consist



**Fig. 1.** Reduced rank architecture. (a) Schematic of photonic chip computing comparing the benefits of our reduced rank architecture over conventional crossbar array architecture in terms of chip area and reduced I/O. (b) Kernels are factorized using rank reduction into two smaller matrices with reduced parameters. (c) Schematic of the photonic implementation of proposed architecture. Input signals  $x_1$ - $x_7$ , encoded onto different wavelengths, are used as input vectors with two subsequent MVM results in the final output  $O$ . (d) Schematic diagram of a  $3 \times 3$  equivalent device used in the experiments, showing signal flow. (e) False colored optical image of the fabricated device equivalent to a  $3 \times 3$  matrix but with 33% reduction in size. (Inset) False colored SEM image of a unit cell showing waveguide interconnects and “active” PCM memory. Zoomed-in false colored SEM image showing  $1 \mu\text{m} \times 1 \mu\text{m}$  PCM (AIST) deposited on waveguide.

of an  $N$  input vector, which is processed through the computation core consisting of  $N^2$  weight elements. Each weight element is programmed to a corresponding weight, controlled using an  $N^2$  input/output (I/O) control system. This increases both the footprint and energy budget of the system. The kernel matrix can be factorized and represented as a product of two reduced rank matrices as demonstrated in Fig. 1(b), resulting in an overall decrease in kernel parameters. In our reduced rank architecture, we leverage the benefits of this factorization process to gain benefits in terms of footprint and reduced number of I/O required to control weight parameters and thus overall lower energy budget and higher computation density.

In Fig. 1(c), we illustrate a schematic of integrated photonic implementation of the reduced rank photonic architecture. The schematic represents a computational core equivalent to a  $7 \times 7$  matrix but with 40% reduction in size. The design is based on a crossbar array design, as explained in previous work [11]. Our architecture leverages the advantages of non-volatile photonics using phase change materials (PCMs) for high energy efficient and high throughput operation. Without loss of generality and for an easier understanding of our design, we take an example of the MVM of a  $7 \times 7$  matrix. Matrix  $\mathbf{W}_{7 \times 7}$  is factorized into two smaller matrices  $\mathbf{U}_{7 \times 2}$  and  $\mathbf{V}_{2 \times 7}$ , of reduced rank of “2,” resulting

in a net 40% reduction in size of the matrix. The MVM operation  $\mathbf{W} \cdot \vec{X}$  is mathematically equivalent to two inner products  $\mathbf{U} \cdot (\mathbf{V} \cdot \vec{X})$ .

To carry out the multiplication, input vector  $\vec{X}$  ( $x_1$ - $x_7$ ) is pre-processed and normalized to  $[0,1]$ . Input vectors  $\vec{X}$  ( $x_1$ - $x_7$ ) are amplitude modulated and encoded on different wavelengths of the light  $\lambda_1 - \lambda_7$  (shown as different colors of in the schematic) and used as input signals to our photonic network. As discussed above, in our architecture, the input signal undergoes two MVMs. First, input vector  $\vec{X}$  is multiplied by matrix  $\mathbf{V}_{2 \times 7}$ . The resultant vector is then subsequently multiplied by matrix  $\mathbf{U}_{7 \times 2}$ , thereby undergoing the second MVM.

For accurate computations, it is important to route light equally to the individual weight cells; therefore, for the first matrix  $\mathbf{V}$ , the input signal is split equally using directional couplers along the row of the matrix. Each signal then gets multiplied by the corresponding weight, stored in a photonic memory cell. Subsequently, the results of these multiplications are accumulated (added) on to a common bus waveguide (along the column) using directional couplers, which completes the first MVM ( $\mathbf{V} \cdot \vec{X}$ ). This is then the input vector for the second multiplication, and the second output is computed in the same manner. Therefore, our proposed architecture performs two multiply-accumulate (MAC) operations one

after the other in a single shot, which yields the complete MVM product  $\vec{O}$ .

To demonstrate this, we implement a reduced rank processor equivalent to a  $3 \times 3$  matrix but with a 33% reduction. Figure 1(d) illustrates the signal flow schematic of the actual device used in the subsequent experiments. Figure 1(e) illustrates a false colored optical image of the fabricated device. The inset is a false colored SEM image of a single photonic memory cell using PCMs; this consists of a waveguide crossing to route signals across other memory cells and the  $1 \times 1 \mu\text{m}$  area of an in-memory active weight cell, which in our case is the PCM  $\text{Ag}_3\text{In}_4\text{Sb}_{76}\text{Te}_{17}$  (AIST).

## B. Setting Weights

Our weights use in-memory computing concepts described elsewhere [27]; our matrix memory elements use PCM-based memory cells [28]. PCMs have different refractive indices depending on their phases. For conventional PCMs such as GST and AIST, the crystalline phase is absorptive, and the amorphous phase is more transparent [29]. Therefore, by varying the fraction of the crystalline to amorphous phase, different intermediate transmission states can be achieved. Pre-determined weights can be programmed within the matrix by sending a high energy optical “pulse” [Fig. 1(e)]; we use a single shot programming technique to program weights as described in our previous work [30]. Individual memory cells can be encoded to better than 5-bit accuracy optically using 15 ns optical pulses and varying switching pulse amplitudes, as shown in Supplement 1 Fig. S2. To set any weight in the range of [0,1], the crystalline state (lowest memory level) is mapped to a “0,” and the fully amorphous state (highest memory level) is mapped to a “1,” and therefore, an intermediate amorphous state will correspond to a number between zero and one. Similarly, to set a negative number, say in range  $[-1, 1]$ , “−1” can be mapped to the crystalline state (lowest memory level) and “+1” can be mapped to the amorphous state (highest memory level); therefore, all numbers between −1 and +1 can be mapped to intermediate memory levels.

## C. Device Fabrication

The fabrication of photonic waveguides is on silicon on insulator (SOI) wafers, with a 220 nm silicon device layer and 3  $\mu\text{m}$  buried oxide. The waveguides are patterned using electron-beam lithography using a positive photoresist (AR-P 6200). The patterned waveguide is then partially dry-etched (etch depth 120 nm). AIST is deposited using radio frequency (RF) sputtering after patterning, followed by lift-off on the waveguides.

## D. Experimental Setup

We use two optical lines: to set the weight (programming line) and to carry out computation (inference line). The programming line consists of a continuous wave (CW) C-band laser from Santec (TSL-550). To send optical pulses, we modulate the CW laser using an electro-optic modulator (EOM). Using a pulse generator (Tektronix AFG3011C) and EOM, we shape the CW laser to obtain ns pulses. Optical pulses are then amplified using an erbium-doped fiber amplifier (EDFA) (Pritel FA-15) and injected into the chip with an inline polarizer (Thorlabs FPC032) to control polarization. The inference line consists of a supercontinuum light source as an input signal, which is split using a  $1 \times 4$  dense

wavelength division multiplexing (DWDM) module. We modulate individual input signals using a variable optical attenuator (VOA) from Thorlabs (V1550A) controlled by a data acquisition module (DAQ). The computed results of the MVM are recorded using a  $3 \times 200$  kHz photodetector (2011-FC-M) from Newport. All photodetectors are connected to a computer using DAQ to record data.

## 3. RESULTS

### A. Image Filtering Task

To demonstrate the utility of our architecture, we test the performance of our photonic hardware to carry out image filtering tasks using an edge detection filter. We use vertical and horizontal edge

detection filters described mathematically as matrix  $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$

and  $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ , respectively, on an image of the Taj Mahal,

Figs. 2(a)–2(c). The input image is pre-processed and flattened as input vectors  $\vec{X}$  ( $x_1$ – $x_3$ ) to be fed through our photonic network with the filter as the kernel  $\vec{W} = \vec{U} * \vec{V}$ . The vertical edge detection filter matrix is factorized to two smaller matrices of reduced

rank “1”  $\vec{U} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$  and  $\vec{V} = [1 \ 0 \ -1]$ ; similarly, the horizontal

edge filter is reduced to  $\vec{U} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$  and  $\vec{V} = [1 \ 1 \ 1]$ .

Experimentally, input vectors are normalized to [0,1] and optically encoded using VOAs. The optically encoded information is then multiplied by individual weights of the kernel and accumulated at the photodetector, therefore performing a MAC operation. The weights of the kernel between  $[-1, 1]$  are achieved by mapping the lowest memory level as “−1” and highest memory level as “+1”; therefore, any number in the range is represented by the corresponding intermediate memory state. We use a 10% switching contrast of a PCM memory cell, between lowest and highest levels and therefore intermediate transmission to represent numbers in a range of  $[-1, 1]$ .

We then combine the result for horizontal and vertical edge filters in Fig. 2(d), resulting in edge detection. To assess the computational accuracy, the error rate of the expected and measured output of over 40,000 MVM operations is plotted in Fig. 2(e). We obtain a normal distribution of the error with a mean error rate of −0.087 and a standard deviation of 0.10.

To show the ubiquity of our network we implement another edge detection filter, this time a vertical Sobel filter, described

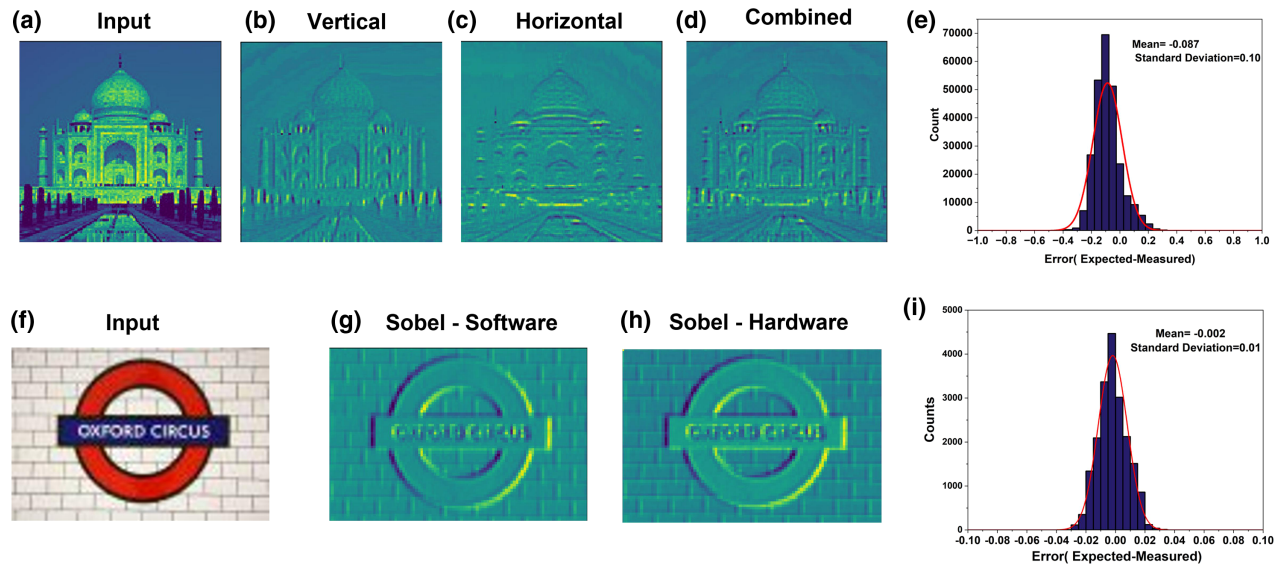
mathematically as  $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$  to detect the left edge on a logo of

an image of a London tube station signage [Fig. 2(f)]. We reduce the Sobel filter weight matrix  $\vec{W}$  into two low rank matrices  $\vec{U}$  and

$\vec{V}$ , where  $\vec{U} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$  and  $\vec{V} = [1 \ 0 \ -1]$ .

To improve the error rate, we switch PCM memory cells to about 40% contrast to represent kernel weights between  $[-2, 2]$ . We compare the performance of our photonic hardware to





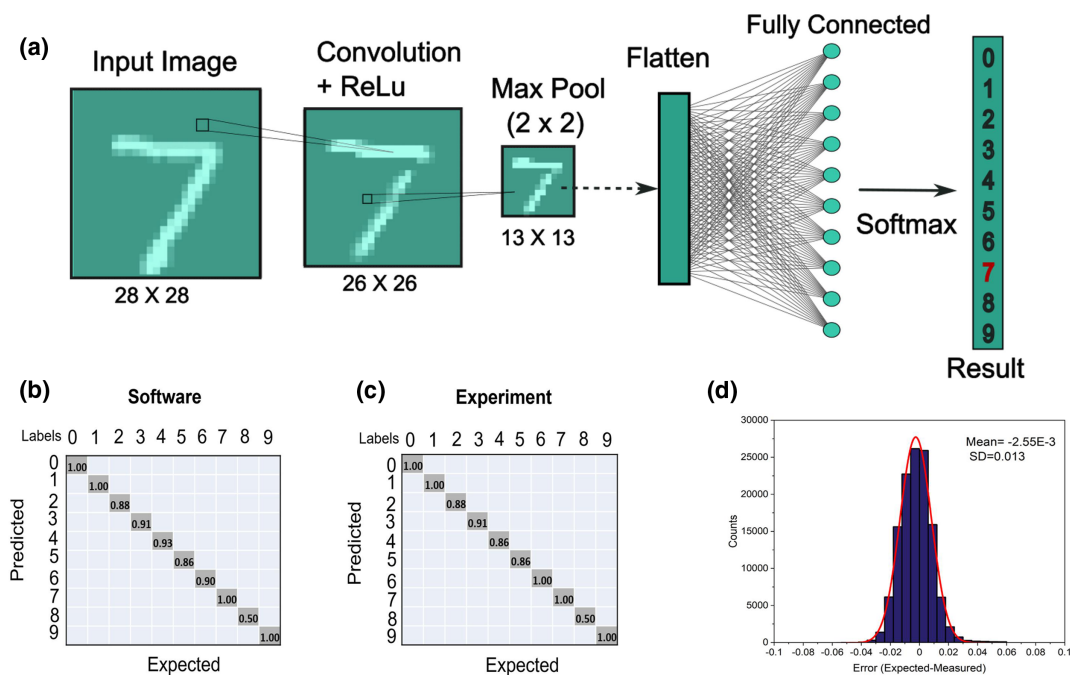
**Fig. 2.** Image filtering. (a) Input image of Taj Mahal is used by pre-processing as an input vector for edge detection experiments. (b), (c) Successful edge detection of vertical and horizontal filters, respectively, using the reduced rank photonic architecture. (d) Combining the edge detection results from (b) and (c) showing highlighted edges. (e) Computing error of expected versus measured value showing a Gaussian distribution with standard deviation of 0.1. (f) Sobel filter is used for edge detection on an image of London tube station. (g) Edge detection results on software are compared with (h) one processed on photonic hardware, showing excellent conformity of the two. (i) Computing error of expected versus measured value showing a normal distribution with standard error deviation of 0.01.

highlight the left edge by comparing the results with that obtained on software, Figs. 2(g) and 2(h), therefore showing an excellent performance of our photonic hardware. As previously shown, we assess the computation accuracy by calculating the error of measured and expected MVM results of over 18,000 operations in Fig. 2(i) and obtain a normal error distribution. It is worth noting that increasing the switching contrast from 10% to 40% helps

in reducing errors by one order of magnitude. We obtain a mean error of  $2 \times 10^{-3}$  with a standard deviation of 0.01. Therefore, for subsequent experiments we use a switching contrast of 40%.

## B. Classification Task

We then build a CNN model to demonstrate an image classification task on the MNIST digit dataset. Figure 3(a) shows the



**Fig. 3.** Image classification using convolutional neural network. (a) Schematic of the convolutional neural network used to classify MNIST handwriting digit. We use our reduced rank photonic architecture with kernel size equivalent to  $3 \times 3$ , but with 33% reduction. (b), (c) Confusion matrix of image classification task for the model on software and experimentally obtained on our photonic hardware, showing excellent agreement. (d) Error estimation of over 120 KMVM by calculating expected versus measured results. A normal distribution fit reveals a standard error deviation of 0.013.



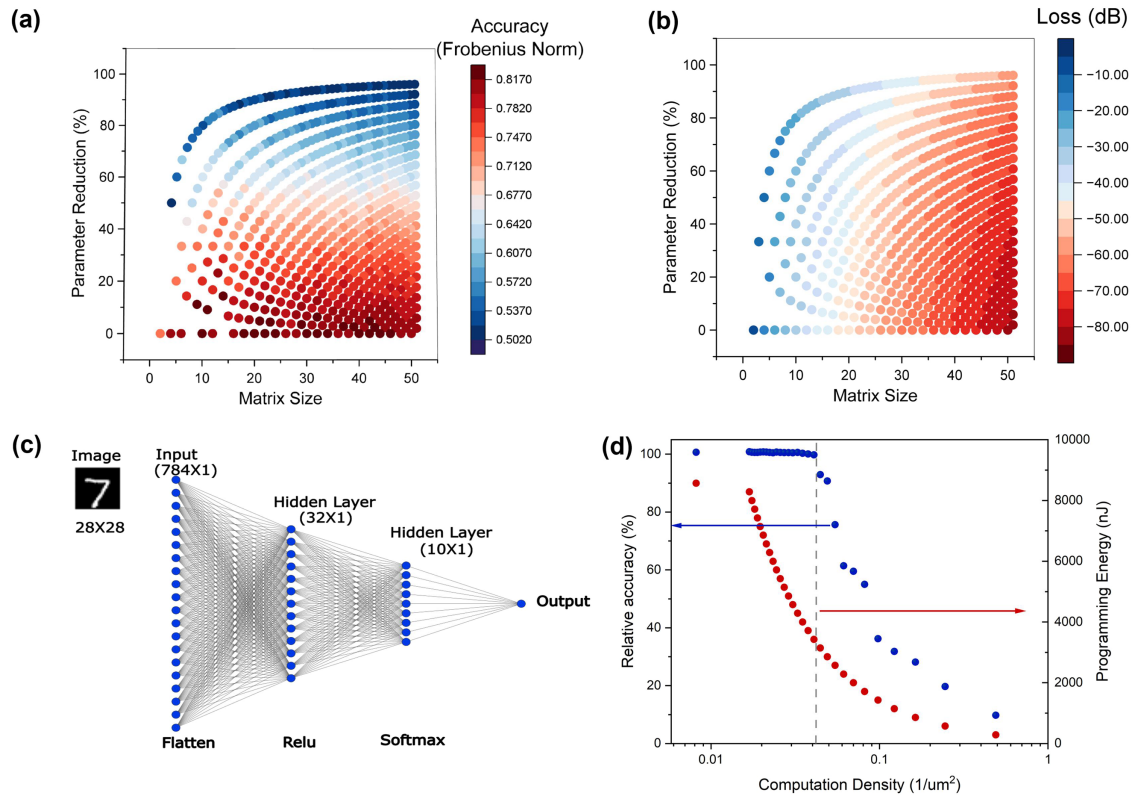
schematic of our CNN model using our low rank matrix photonic core, which is a reprogrammable hardware-based convolution layer. The model is trained on software to achieve 96% accuracy. Our photonic hardware is then programmed with the appropriate weights (by setting the state of the phase change photonic memories) and is then used as the convolution layer. Our photonic kernel is set to the appropriate state as described previously. We use 100 images from the MNIST dataset as a testing dataset. As before, the images are flattened as input vectors normalized to  $[0, 1]$  for multiplication. Figures 3(b) and 3(c) show the confusion matrices for the image classification task obtained on software and experimentally on our photonic hardware. We achieve 94% accuracy in the classification task experimentally, which matches closely to accuracy obtained on software, suggesting that our approach works very well. In Fig. 3(d), we plot a histogram of computation errors for over 120,000 MVM operations. A Gaussian distribution fit reveals a mean error of  $-2.55 \times 10^{-3}$  with a standard deviation of 0.013, approaching software-based accuracy.

### C. Future Projections

In this work, we have demonstrated that this technique works well for small matrices and show excellent performance vis-à-vis software. We further evaluate the benefits of our technique for larger matrices. As one would expect, the source of error in computation is due to the matrix factorization process itself. Mathematically, it is not feasible to have a reduced rank matrix to represent the original weight matrix  $W_{m \times n}$ . As explained before, weight matrix

$W_{m \times n}$  can be reduced to two matrices  $U_{m \times r}$  and  $V_{r \times n}$ , where  $r$  is the targeted reduced rank of the matrix. The total number of parameters after rank reduction is  $mr + rn$ ; therefore, for any matrix size reduction advantages,  $r$  is chosen such that  $r < \frac{mn}{m+n}$ . The target rank  $r$  is varied, which is equivalent to reducing the number of parameters in the weight matrix. In Fig. 4(a), we simulate the effect of parameter reduction (reducing the rank of the matrix) on the accuracy of representing the original matrix. As one would expect, reducing the number of parameters results in a reduction of accuracy. We use this to estimate the system level loss of our photonic network for a matrix of size  $N \times N$ ; we assume directional coupler losses of  $-0.1$  dB and waveguide crossing loss of  $-0.13$  dB, as estimated in previous work [11]. Reducing the size of the matrix through rank reduction results in a smaller size array and hence lower optical losses as shown in Fig. 4(b).

For practical applications of this technique, one needs to optimize accuracy and optical losses in the system. To find an optimum balance of parameter reduction, we calculate the effect of reducing the number of weight parameters on the accuracy of the image classification task. We simulate a feed-forward DNN with two hidden layers and over 25,000 trainable parameters, as illustrated in Fig. 4(c). The model is trained for MNIST handwritten digit classification with over 91% accuracy. Weight matrix  $W$  is then extracted from the trained model. We then factorize the weight matrix, using a rank reduction technique. To account for negative numbers in the trained matrix, we use a semi-negative matrix factorization algorithm [31]. The relative accuracy of the classification task from a rank reduced matrix network is plotted in Fig. 4(d) as



**Fig. 4.** (a) Effect of reducing the number of parameters on the accuracy of matrix factorization for different sizes of matrices. Here the accuracy is defined as the difference of Frobenius norm of the original weight matrix and reduced rank matrix. Increasing the reduction factor results in lower accuracy. (b) Effect of parameter reduction of matrix on overall optical loss of the proposed photonic architecture. Parameter reduction results in lower losses in the system. (c) Schematic of a feed-forward model used for MNIST handwritten digit recognition used for simulation in (d). (d) Effect of parameter reduction on the relative accuracy and programming energy of the system. Parameter reduction results in over  $3\times$  gain in computation density due to smaller footprint without any loss in accuracy for the classification task with additional benefits of  $2.5\times$  gain in terms of programming energy.

a function of parameter reduction. Remarkably, we observe that even with a 60% reduction in parameters, there is no significant loss in accuracy; this results in  $> 3\times$  improvement in computation density.

Assuming the weight matrix based on PCM photonic memories, switching energy for single shot programming is assumed to be 350 pJ for a single memory cell. In terms of system level energy requirements, a reduction in training parameters results in over  $2.5\times$  improvement in programming energy; thus, our architecture scales very favorably in terms of system level energy.

## 4. CONCLUSIONS

We have demonstrated a novel MVM architecture for applications in neural networks to reduce kernel sizes without incurring significant losses in accuracy. We experimentally implement this architecture on a silicon photonic platform using phase change reconfigurable weights. By applying the proposed architecture to image filtering and image classification tasks, we experimentally demonstrate  $3\times 3$  kernel equivalency with a kernel reduction of 33%. In spite of the matrix size reduction and reduced complexity of the system, we detect higher accuracy in the computation due to the smaller number of required weights and resulting variability. We estimate that for up to a 60% reduction in kernel size, there is no significant loss of information for the MNIST classification task. The proposed architecture thus not only improves scalability, complexity, and programming energy, but is also adaptable and can be readily implemented in diverse neuromorphic computing architectures.

**Funding.** Engineering and Physical Sciences Research Council (EP/W022931/1, EP/T023899/1, EP/R001677/1); Horizon 2020 Framework Programme (101017237); HORIZON EUROPE European Innovation Council (101046878, 101098717); Clarendon Fund.

**Acknowledgment.** The authors acknowledge helpful discussions with A. Ne and R. Saphal.

**Author Contributions.** SA conceived the original experiments and carried out experimental work with help from BD, JF, and NE. WP and HB helped analyze results and write the paper; HB led the work. SA and HB wrote the manuscript with substantial input from all authors.

**Disclosures.** JF, WHPP, and HB are co-founders and shareholders in Saliency Labs Ltd.

**Data availability.** The data that support the findings of this work are available from the corresponding author upon reasonable request.

**Supplemental document.** See Supplement 1 for supporting content.

## REFERENCES

- O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.* **22**, 1533–1545 (2014).
- K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016), Vol. **2016**.
- S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.* **2016**, 1 (2016).
- D. Cireřan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.* **32**, 333–338 (2012).
- T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *IEEE 2nd International Conference on Big Data Analysis (ICBDA)* (IEEE, 2017), pp. 721–724.
- D. Shin and H. J. Yoo, "The heterogeneous deep neural network processor with a non-von Neumann architecture," *Proc. IEEE* **108**, 1245–1260 (2020).
- A. Sebastian, M. Le Gallo, and E. Eleftheriou, "Computational phase-change memory: beyond von Neumann computing," *J. Phys. D* **52**, 443002 (2019).
- P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature* **577**, 641–646 (2020).
- X. Li, N. Youngblood, W. Zhou, J. Feldmann, J. Swett, S. Aggarwal, A. Sebastian, C. D. Wright, W. Pernice, and H. Bhaskaran, "On-chip phase change optical matrix multiplication core," in *IEEE International Electron Devices Meeting (IEDM)* (2020), pp. 7.5.1–7.5.4.
- J. Y. S. Tan, Z. Cheng, X. Li, N. Youngblood, U. E. Ali, C. D. Wright, W. H. P. Pernice, and H. Bhaskaran, "Monadic Pavlovian associative learning in a backpropagation-free photonic network," *Optica* **9**, 792–802 (2020).
- J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, and H. Bhaskaran, "Parallel convolutional processing using an integrated photonic tensor core," *Nature* **589**, 52–58 (2021).
- N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Neuromorphic photonic networks using silicon photonic weight banks," *Sci. Rep.* **7**, 7430 (2017).
- Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**, 441–446 (2017).
- X. Xu, M. Tan, B. Corcoran, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, D. G. Hicks, R. Morandotti, A. Mitchell, and D. J. Moss, "11 TOPS photonic convolutional accelerator for optical neural networks," *Nature* **589**, 44–51 (2021).
- H. Zhou, J. Dong, J. Cheng, W. Dong, C. Huang, Y. Shen, Q. Zhang, M. Gu, C. Qian, H. Chen, Z. Ruan, and X. Zhang, "Photonic matrix multiplication lights up photonic accelerator and beyond," *Light Sci. Appl.* **11**, 30 (2022).
- G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. B. Miller, and D. Psaltis, "Inference in artificial intelligence with deep optics and photonics," *Nature* **588**, 39–47 (2020).
- X.-Y. Xu, X.-L. Huang, Z.-M. Li, J. Gao, Z.-Q. Jiao, Y. Wang, R.-J. Ren, H. P. Zhang, and X.-M. Jin, "A scalable photonic computer solving the subset sum problem," *Sci. Adv.* **6**, eaay5853 (2020).
- J. Gan, W. Wang, and K. Lu, "Compressing the CNN architecture for in-air handwritten Chinese character recognition," *Pattern Recognit. Lett.* **129**, 190–197 (2020).
- N. Kozyrskiy and A.-H. Phan, "CNN acceleration by low-rank approximation with quantized factors," *arXiv*, arXiv:2006.08878 (2020).
- M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," *arXiv*, arXiv:abs/1405.3 (2014).
- J. H. Luo, J. Wu, and W. Lin, "ThiNet: a filter level pruning method for deep neural network compression," in *Proceedings of the IEEE International Conference on Computer Vision* (2017), Vol. **2017**, pp. 5058–5066.
- S. Pai, O. Solgaard, S. Fan, and D. A. B. Miller, "Scalable and self-correcting photonic computation using balanced photonic binary tree cascades," *arXiv*, arXiv:2210.16935 (2022).
- C. Feng, J. Gu, H. Zhu, Z. Ying, Z. Zhao, D. Z. Pan, and R. T. Chen, "A compact butterfly-style silicon photonic-electronic neural chip for hardware-efficient deep learning," *ACS Photon.* **9**, 3906–3916 (2022).
- S. Banerjee, M. Nikdast, S. Pasricha, and K. Chakrabarty, "Pruning coherent integrated photonic neural networks," *IEEE J. Sel. Top. Quantum Electron.* **29**, 6101013 (2023).
- J. Gu, Z. Zhao, C. Feng, M. Liu, R. T. Chen, and D. Z. Pan, "Towards area-efficient optical neural networks: an FFT-based architecture," in *25th Asia and South Pacific Design Automation Conference (ASP-DAC)* (2020), pp. 476–481.
- M. Milanizadeh, F. Toso, G. Ferrari, T. Jonuzi, D. A. B. Miller, A. Melloni, and F. Morichetti, "Coherent self-control of free-space optical beams with integrated silicon photonic meshes," *Photon. Res.* **9**, 2196–2204 (2021).
- C. Rios, N. Youngblood, Z. Cheng, M. le Gallo, W. H. P. Pernice, C. D. Wright, A. Sebastian, and H. Bhaskaran, "In-memory computing on a photonic platform," *Sci. Adv.* **5**, eaau5759 (2019).

28. C. Ríos, M. Stegmaier, P. Hosseini, D. Wang, T. Scherer, C. D. Wright, H. Bhaskaran, and W. H. P. Pernice, "Integrated all-photonic non-volatile multi-level memory," *Nat. Photonics* **9**, 725–732 (2015).
29. Z. Gong, F. Yang, L. Wang, R. Chen, J. Wu, C. P. Grigoropoulos, and J. Yao, "Phase change materials in photonic devices," *J. Appl. Phys.* **129**, 030902 (2021).
30. X. Li, N. Youngblood, C. Ríos, Z. Cheng, C. D. Wright, W. H. Pernice, and H. Bhaskaran, "Fast and reliable storage using a 5 bit, nonvolatile photonic memory cell," *Optica* **6**, 1–6 (2019).
31. C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 45–55 (2010).