Large-Scale Optical Hardware for Neural Network Inference Acceleration

by

Liane Bernstein

B. Eng., Polytechnique Montréal (2016)
S. M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

© 2024 Liane Bernstein. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Liane Bernstein
Department of Electrical Engineering and Computer Science
January 11, 2024

Certified by: Dirk R. Englund
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Large-Scale Optical Hardware for Neural Network Inference Acceleration

by

Liane Bernstein

Submitted to the Department of Electrical Engineering and Computer Science
on January 11, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

Artificial deep neural networks (DNNs) have revolutionized tasks such as automated classification and natural language processing. To boost accuracy and handle more complex workloads, DNN model sizes have grown exponentially over the last decade, outpacing improvements in digital electronic microprocessor efficiency. This mismatch limits DNN performance and contributes to soaring data center energy costs. Optical hardware for deep learning (optical neural networks, or ONNs) can theoretically increase DNN processing efficiency; however, the feasibility of large-scale, fully programmable and reconfigurable ONNs has not yet been comprehensively shown in experiments.

This thesis reports our demonstrations of ONNs that classify ∼1000-element input vectors using standard DNN layers in inference without hardware modeling or retraining. In a first project, we used digital optical links to replace copper wires for transmitting and copying data to electronic multipliers. Our experimental implementation showed an MNIST classification accuracy within <0.6% of the digital electronic ground truth. We estimated that this 'digital ONN' could reduce energy consumption for long data transfer lengths, but not in tightly packed electronic multiplier arrays. Therefore, in a second project, we expanded upon this work by performing reconfigurable optical multicast and analog optoelectronic weighting to compute DNN layer outputs in a single shot. Our proof-of-concept system yielded an MNIST classification accuracy of 96.7% (boosted to 97.3% with weight fine-tuning) with respect to the ground-truth accuracy of 97.9%. We calculated that a near-term optimized version of this system could lower energy consumption and latency by 1-2 orders of magnitude compared to a state-of-the-art digital electronic systolic array. These findings suggest a paradigm shift towards optoelectronic DNN accelerators with lower resource utilization that could enable the next generation of deep learning.

Thesis supervisor: Dirk R. Englund
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

I am very grateful to the many wonderful people in my life for their guidance and support throughout my PhD journey.

I would first like to thank my PhD advisor, Professor Dirk Englund. Dirk's enthusiasm and excitement about physics and engineering are truly infectious. He introduced me to the field of optical hardware for deep learning, and I have appreciated all the new ideas he has shared with me that push the boundaries of computing. He encouraged me when I had doubts and, through his guidance, helped me move forward in various projects. I am thankful for the opportunity he gave me to be in his group. I would also like to express gratitude to my other committee members, Professors Joel Emer and Jelena Notaros. I was both a student and a teaching assistant in Joel's class on hardware for machine learning (co-taught with Professor Vivienne Sze), and Joel taught me a tremendous amount about electronic deep neural network accelerators. I thank him for his mentorship, characterized by kindness and a genuine desire to see me succeed. I had the pleasure of attending the inaugural semester of Jelena's first-of-its-kind silicon photonics class, in which I learned a great deal about the subject in both theory and experiment. Jelena also provided me with invaluable feedback when she served on my RQE committee, and I thank her for being a reliable and considerate presence on my thesis committee.

Thanks also to my collaborators Alex Sludds, Ryan Hamerly, Professor Vivienne Sze, Sivan Trajtenberg-Mills and Chris Panuski. I am particularly grateful to Alex for our friendship and for our long days in the lab working on the DONN, and to Chris for spending numerous hours discussing SLMs with me. I also appreciate my colleagues Saumil Bandyopadhyay, Sri Krishna Vadlamani, Mihika Prabhu, Eric Bersin, Hugo Larocque, Camille Papon, Ian Christen, Stefan Krastanov, Isaac Harris, Cole Brabec, Jacques Carolan, Jordan Goldstein, Dalia Ornelas-Huerta, Carlos Errando-Herranz, Michael Walsh, Sophia Duan and Valeria Saggio, not just for their brilliance, but also for their friendship and camaraderie. Saumil and Sri, thanks for all our technical and non-technical discussions en route to the coffee shop and for our conference shenanigans. Hugo, thanks for bearing with my monologues during our long drives up to Canada. Mihika and Eric, thanks for all your advice over the years and our good times together. I am also grateful to my graduate counselor,

Professor Roger Mark, as well as EECS Graduate Officer Professor Leslie Kolodziejski for their encouragement and our insightful discussions. Thanks also to our administrative staff Janice Balzer, David Barnett, Janet Fischer and Kathy McCoy who helped me navigate various logistical hurdles and enabled many research projects and collaborations. Lastly, thanks to facilities staff Bill Gibbs and Matt McGlashing for help setting up the labs.

The work in this thesis would not have been possible without my incredible friends. Even in my lowest moments, you never stopped believing in me. You inspire me every day through your generosity, thoughtfulness and joie-de-vivre. Alina LaPotin, Angelina Nou, Jen Yarin, Una Nattermann, Jacob Baron, Matthew Tien, Yan Teng, Dima Kochkov, Ann Young, Alex Young, Misha Badov, Dan Lin, Bilkit Githinji, Jake Teitelbaum, Andee Wallace, Koe Inlow, Emma Lee, Ariana Wermer-Colan, Tess Carter, Jonah Weil, Avilash Cramer, Félix Beaudry, Mélissa Lacasse, Joanna Garcia, Zoe Guan, Caitlin Peotto, Lamia Ateshian and Maddie Sutula: you all have such big hearts, and I feel extremely lucky to have you in my life. We have been on so many wonderful adventures together, and I look forward to many more. Special thanks to Alina for having been an amazing roommate, climbing partner and overall person – whether you are down the hall or in a remote cabin in the Colorado mountains, I know I can always count on you!

To my Order of the White Rose family, especially Édith Ducharme, Nathalie Provost and Michèle Thibodeau-DeGuire, I am so lucky to know you – your kindness and your strength are incredibly inspiring to me.

Lastly, a special thanks to my siblings, cousins, aunts, uncles, nieces, and most of all, my parents. Marie Béland, Mark Bernstein and Louise Parker, you have been my biggest cheerleaders not just during my PhD, but throughout the rest of my life as well. I cannot thank you enough for your ever-enduring love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The limits of digital electronics for deep learning

Improved processing power in digital electronic computers has occurred so consistently over the past 50 years that it is practically taken for granted. Empirical formulas were devised to describe these advances, with Moore's Law in 1965 stating that the number of transistors per chip doubles every year, with a lower cost per component [20] (revised to every two years in 1975 [21]) and Dennard scaling in 1974 stating that power dissipation remains constant despite an increase in the number of components per area [22]. These projections held remarkably true for decades. Now, though transistors continue to be scaled down, we are reaching the physical limits of transistor size – within a small number of atoms per transistor, and at increased fabrication cost. Moreover, limits on power delivery and heat dissipation are causing single-thread performance and clock frequency to plateau, breaking with Dennard scaling [23]. In other words, further miniaturization of general-purpose electronic chips can no longer be counted on for additional gains in computational capacity and efficiency.

These challenges are of particular concern in resource-intensive tasks like processing deep neural networks (DNNs). DNNs are a category of artificial intelligence that has revolutionized fields like translation [8], automated image classification [1] and medical prediction and diagnosis [24]. They have recently come to the forefront of public consciousness, as they represent the foundation of generative models like ChatGPT [19], [25], which can output conversational text or even artwork of a given style in response to a user's prompt. Huge

datasets and DNN models have enabled these breakthroughs, with larger DNN model sizes tending to yield better accuracies [26], [27]. In fact, the number of parameters in DNN models has increased exponentially over the last decade, as shown in Fig. 1.1. The resulting mismatch between DNN processing requirements and hardware capabilities limits DNN performance, and causes rising energy consumption and greenhouse gas emissions [28]. To bridge this gap between flatlining general-purpose digital electronic processors and the ever-increasing computational demands of modern DNNs, we need a paradigm shift.

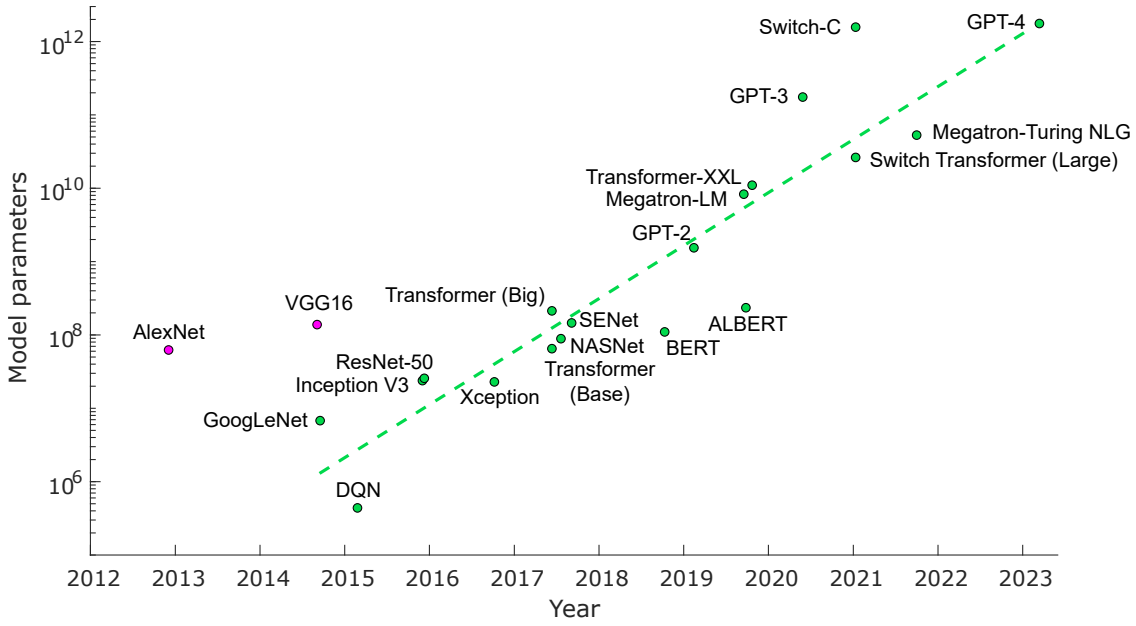

Figure 1.1: Number of parameters, i.e., weights, in recent landmark DNNs [1]–[19][1,2,3]. Larger models tend to require more compute power, notably in fully connected layers. (The number of multiplications in each DNN is more representative of its computational burden, but is not always reported.) The two outlying nodes (pink) are now considered overparameterized – subsequently, efforts have been made to reduce DNN sizes. Despite these efforts, model sizes continue to grow exponentially.[4]

[1]References dated by first release, e.g., on arXiv.

[2]The number of parameters in GPT-4 are estimated in Ref. [29].

[3]Switch Transformers are designed to be sparse, so not all parameters are used in each inference.

[4]I have updated this figure from our 2021 paper [30] to include more recent networks.

1.2 A shift from electronics to optics?

One option to improve electronic hardware performance is to tailor it to the specific task of processing DNNs. While such circuits can yield significant increases in efficiency, they will ultimately be bounded by the same physical limits as general-purpose digital electronic chips. Instead, we can change the way we do computing entirely: by using optics in concert with electronics for DNNs. Thanks especially to the inherent parallelism and low loss in optical data transfer and copying, optics is very well-suited to DNN hardware acceleration (so-called 'optical neural networks' or ONNs). I will show in this thesis that optics can realize fully programmable and reconfigurable hardware for deep learning, implementing standard DNN models at scale.

1.3 Thesis overview

This thesis describes my work toward demonstrating large-scale ONNs, where I focus on 3D architectures that have high connectivity between tightly packed spatial modes. The thesis is divided as follows:

- Chapter 2 is an introduction to DNNs and specialized hardware for DNNs, including summaries of the state of the art in both electronics and optics.
- Chapter 3 is a description of the elements of 3D optical computing.
- Chapter 4 reports our 3D digital optical scheme for data transfer and copying from transmitters to digital electronic multipliers.
- Additional performance gains are achievable if computation is performed in optics alongside optical data routing. Chapter 5 demonstrates an analog single-shot-per-layer ONN that can reduce DNN latency and energy consumption for applications that require ultra-fast computation, at the cost of slightly reduced accuracy.
- Lastly, in Chapter 6, I conclude by summarizing the results of this thesis as well as suggesting future research directions.

Chapter 2

Artificial Deep Neural Networks

This chapter serves as a high-level overview of DNNs as well as state-of-the-art DNN accelerator architectures.

2.1 Introduction to deep neural networks

An artificial deep neural network (DNN) is a computer program that learns patterns in datasets to make predictions (or *inferences*) about new data. Because DNN calculations are so resource-intensive, they are mostly performed 'in the cloud', i.e., in data centers that contain digital electronic supercomputers. In these data centers, 80-90% of machine learning tasks are in inference rather than training (the step in which the DNN model parameters are learned) [31]. This statistic can be understood with the intuition that while training a model requires much more compute time and energy than a single inference, a model only has to be trained once (excluding minor model updates or fine-tuning) before it is deployed to perform many inferences. Inference acceleration will thus be the focus of this thesis.

A DNN is composed of a sequence of layers, where the outputs of each layer are weighted sums of the inputs followed by a nonlinear function. A weighted summation is equivalent to a linear transformation, which can be represented as either a convolution of an input matrix with a learned kernel or a multiplication of an input vector by a learned matrix – matrix-vector multiplication (MVM). During the training step, the elements inside the kernel or matrix (the *weights*) are iteratively updated by comparing the DNN outputs to the known

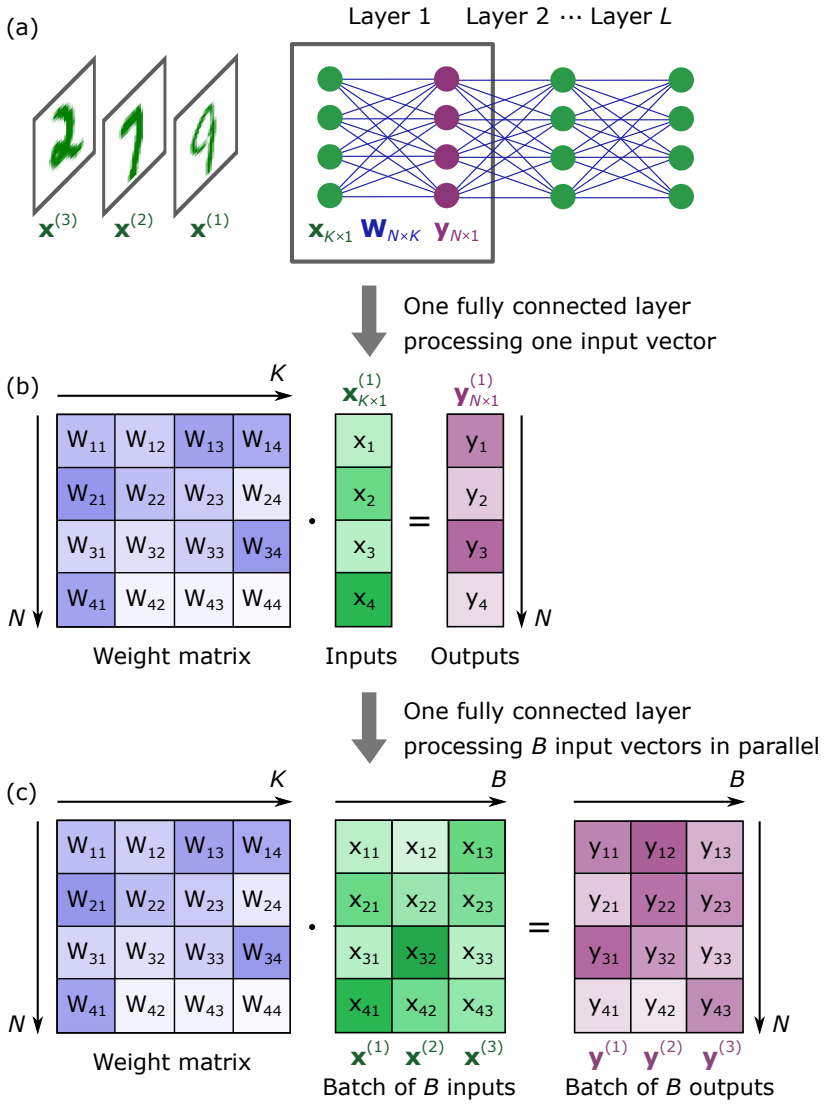

Figure 2.1: Fully connected neural network (FC-NN). **a**, An FC-NN classifies input images $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$ using L layers. **b**, The output activation vector $\mathbf{y}_{N\times1}$ of the first layer is the product of the weight matrix $\mathbf{W}_{N\times K}$ with the input vector $\mathbf{x}_{K\times1}$ (input image reshaped from 2D to 1D). Nonlinearity not shown for ease of visualization. **c**, Images processed in a batch: one layer becomes a matrix-matrix product of weight matrix $\mathbf{W}_{N\times K}$ with activation matrix $\mathbf{X}_{K\times B}$.

correct values in the training set.

Figure 2.1 shows a DNN composed of three MVMs (i.e., two 'hidden layers', defined as the number of output vectors in the network, excluding the final outputs). MVM layers are called 'fully connected' (FC) because each output element is the weighted sum of all of the inputs. Within an FC layer (Fig. 2.1b), a matrix $\mathbf{W}_{N\times K}$ of weights multiplies a vector $\mathbf{x}_{K\times 1}$ of input values ('input activations', of length K, where often, $K \approx 1{,}000$ in modern DNN workloads [27]). This vector is either originally a 1D input or a flattened multidimensional matrix (e.g., a 2D input image). The MVM yields a vector of output activations ($\mathbf{y}_{N\times 1}$), which is then fed through a nonlinear function. A common nonlinear function is the rectified linear unit (ReLU), where $\text{ReLU}(y_n) = \max(0,\, y_n)$. ReLU is easily implemented in digital electronics, for example, by a comparator. The vectors can also be processed in B-sized batches, where the inputs are represented by a matrix $\mathbf{X}_{K\times B}$ (Fig. 2.1c). The FC layer then becomes a matrix-matrix multiplication ($\mathbf{W}_{N\times K} \cdot \mathbf{X}_{K\times B}$). While batching tends to enable higher throughput, it can increase latency as there may be a delay in collecting inputs if they are streamed in.

MVM is the core operation in multilayer perceptrons, recommendation models, recurrent neural networks and large language models such as Transformers. Together, these tasks represent a large fraction of modern data center inference workloads[1] (e.g., 82% in Google data centers [32]). While DNN models may also include convolutions (CONV), as discussed above, convolutions can be recast into matrix-matrix products, e.g., with a Toeplitz matrix [33] or using a patching technique [34] where the CONV feature map matrices are unrolled into a larger 2D matrix product. For these reasons, this thesis will focus on DNNs composed of MVMs. I will, however, note that mapping convolution to matrix multiplication can come at the cost of reduced efficiency because of redundant data in the unrolled matrices. I will come back to this point later (Section 5.9.4) when I discuss potential extensions of our work on optical MVM to the direct implementation of convolutions through the spatial Fourier transform properties of lenses.

[1] A workload is a computational task such as an MVM of a given size within a DNN.

2.2 Hardware for deep neural networks

General-purpose hardware like a CPU is optimized to be able to quickly switch between a wide variety of tasks, like writing a document or browsing the web. CPUs tend to perform operations serially with repeated calls to memory. While they can be used to run DNNs, they are not very efficient, since the energy required for a single memory access can be two orders of magnitude higher than one multiplication [33], [35]. By contrast, DNN hardware accelerators leverage the regular structure of DNN layers to reduce memory accesses and increase parallelism.

Before I discuss current state-of-the-art hardware, I will describe dataflows and hardware evaluation criteria. A dataflow defines the way data are routed and processed through the hardware. It characterizes the order of operations, such as how the data are fed into the hardware and which data are stored in local memories near multipliers. Two common dataflows are weight-stationary and output-stationary, to which I have provided a simplified introduction below. See Refs. [35] and [33] for a more thorough explanation.

2.2.1 Dataflows

Weight stationary

Figure 2.2 shows the flow of data through a simple weight-stationary MVM accelerator. The activation vector $\mathbf{x}$ is streamed through an array of processing elements (PEs) that perform the basic operation of MVM, one multiply-accumulate (MAC). Each PE stores an element of the weight matrix W_{nk}. The weights can be kept fixed (i.e., stationary) in a small memory (register) of the PE, as long as the weight matrix fits onto the hardware – otherwise, the weights need to be tiled and updated over time, as I will explain below. Then, each row is summed while the outputs are streamed out.

Output stationary

Figure 2.3 shows an output-stationary configuration. Each weight column flows into the PE array at the same time as its corresponding input activation. The PEs multiply the weight

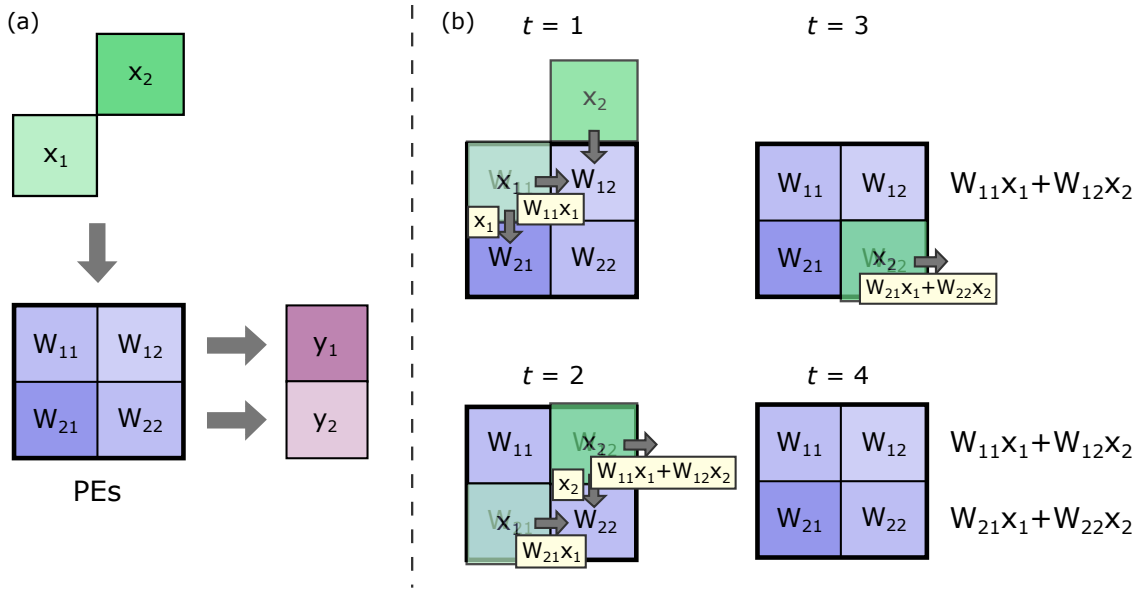

Figure 2.2: Flow of data in weight-stationary MVM accelerator. Shown here: systolic array. **a**, Inputs **x** are streamed in to the array of processing elements (PEs). Each PE contains a register that stores a weight. **b**, Example processing steps through time, where inputs and partial sums are passed from one PE to the next at each time step (yellow boxes).

values with the activations, then the next weight column and activation value are streamed in. The PEs perform the next multiplications and accumulate in-place. In other words, the partial products are kept local to the PEs until all the columns of the weight matrix have been processed, and the output vector **y** can then be read out. If there are not enough PEs to store all outputs, we need to resort to tiling here as well.

As seen in Fig. 2.3, this dataflow can process an equally-sized matrix with fewer PEs: N PEs for an $N \times K$-sized weight matrix, as opposed to $N \times K$ PEs in the weight-stationary case. However, there can be tradeoffs in latency, as will be discussed later.

Other dataflows

Other dataflow such as input stationary or row stationary also exist. In an input-stationary dataflow, the inputs are kept local to the PEs. In the row-stationary case, a small memory within the PE keeps CONV filter rows and input feature map rows stationary for optimal convolutional reuse. Others have flexible dataflows, where the dataflow is selected to be best-suited to a workload. Each option has trade-offs depending on the hardware architecture

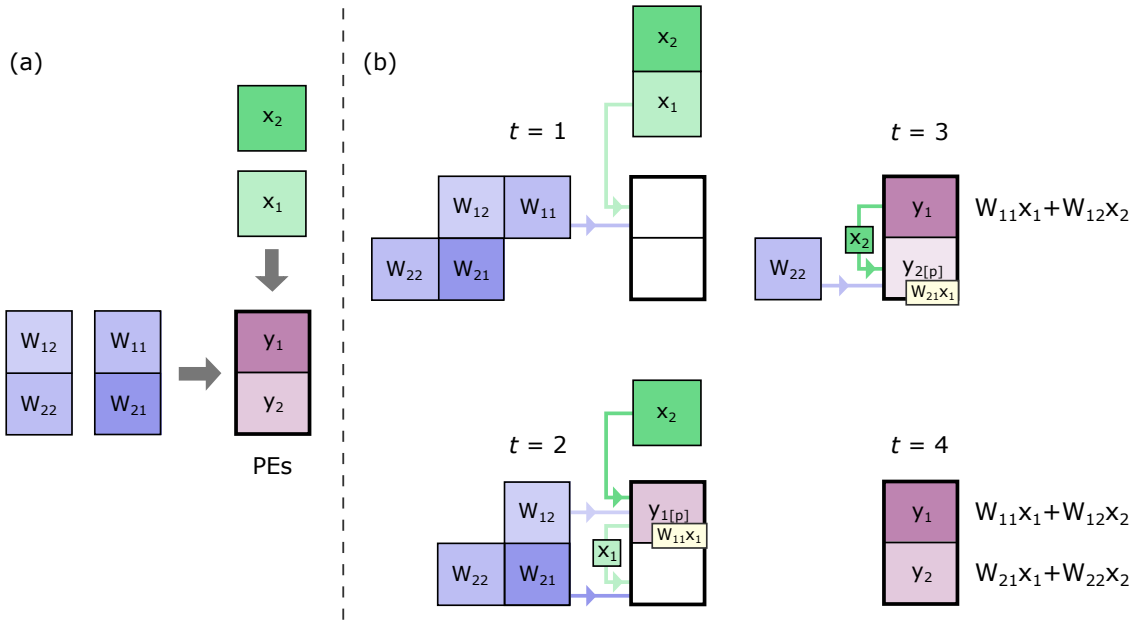

Figure 2.3: Example output-stationary dataflow for MVM. **a**, Weights and inputs flow into the PEs to be multiplied and accumulated. The PEs store the partial sums (denoted by the subscript 'p'). **b**, Example processing steps, where an element of the activation vector x_k flows into the PEs with one column of the weight matrix at each time step, with in-place accumulation.

and task, e.g., whether the hardware is optimized for CONV versus MVM, or whether to prioritize low latency over high throughput.

Tiling

When the DNN layer is too large to 'fit' in the PE memories in a specified dataflow, it needs to be tiled, i.e., split into blocks that are computed one at a time, saved to memory, then recombined later. For example, if there are $2N \times K$ weights and K activations to be processed in a weight-stationary dataflow on hardware that has $N \times K$ PEs with registers that can hold one weight each, then the outputs can be computed in two tiles.

When the layer dimensions are much larger than the number of PEs or when the data shapes cannot easily be factored to fit well onto the hardware (as is often the case), tiling can become quite complex. Tools such as Timeloop [36] can find the optimal mapping of data onto the hardware in these cases.

2.2.2 Criteria for evaluation of DNN accelerators

The key criteria used in evaluating hardware for DNNs are latency, throughput, energy consumption, area and accuracy. Performance values are often normalized per MAC to be able to compare differently sized hardware.

- Latency: the time to process one input vector through a full network or layer, from start to finish.
- Throughput: the number of input vectors or MACs per second that are output by the system. Because operations can be stacked and executed in parallel for data at different stages through the processing pipeline, throughput is greater than 1/latency (unless there is no pipelining).
- Energy or power consumption: can be measured in energy or power expended per inference or per MAC. This value should include all peripheral logic, and is ideally measured at the wall plug.
- Chip area: used as a measure of cost of the hardware.

- Accuracy: low-precision or analog hardware solutions can yield lower accuracy than their full-precision digital electronic counterparts because of noise that can arise from random fluctuations. Therefore, accuracy should be reported alongside the intrinsic accuracy of a model, which is the accuracy of performing a task (e.g., classification) with error-free arithmetic. The next section presents datasets commonly used to report classification accuracy.

The workload will influence these metrics, since it affects the hardware utilization. Intuitively, if a DNN layer size does not align well with the PE arrangement, there will be unused portions of the hardware during DNN computation. The unused PEs can burn power without providing useful computation or require data to travel additional distances over unactivated parts of the hardware. Recent benchmarking methods such as MLPerf [37] have been developed to standardize reporting metrics on representative workloads to try to ensure fair comparisons between hardware.

2.2.3 Datasets to test accuracy of DNN hardware

Standardized datasets are used to compare the accuracy of different DNN hardware. A common benchmark is the Modified National Institute of Standards and Technology (MNIST) handwritten digit dataset [38]. It is comprised of 10,000 test images and 60,000 training images of 28×28-pixel greyscale hand-drawn numbers in ten classes (the numbers 0 to 9). The goal is to determine which number was drawn in each image. While classifying this dataset is generally considered a fairly easy task, it is a typical initial proof of principle for emerging DNN hardware. Fashion-MNIST [39] is a dataset containing greyscale images of fashion items (e.g., a dress, a sneaker, etc.) that was designed as a more challenging drop-in for MNIST. Each image is also of size 28×28 pixels, with 60,000 training images and 10,000 test images. Still harder tasks are language processing (e.g., SQuAD [40]) or the classification of complex images divided into more classes, like QuickDraw [41], CIFAR-100 [42] or ImageNet [43].

2.2.4 State-of-the-art electronic DNN accelerators

State-of-the-art commercial DNN accelerators are specialized digital electronic circuits that optimize parallel processing of highly structured data. Examples include graphics processing units (GPUs) [44], field-programmable gate arrays (FPGAs) [45] or tensor processing units (TPUs, which are weight-stationary systolic arrays). The TPU v1 [46] started out as an inference-only accelerator, but later models [32] also support training. These hardware solutions aim to reduce data movement and increase parallelism through a large number of multipliers with minimal idle time. They also often support low-precision or mixed-precision computing, which is less resource-consuming with fewer bits per operation [33]. When models are trained accordingly (e.g., with 'quantization-aware training'), the resulting loss of accuracy can be small compared with the full-precision ground truth [47]. Additionally, these accelerators can exploit the sparsity of DNNs, i.e., the fact that many values in DNNs are zeros: the ReLU nonlinearity sets any negative activations to zero, and weights with small values can be set to zero with little impact on accuracy in a process called 'pruning'. Intuitively, because any number multiplied by zero equals zero, the multiplication process can be bypassed entirely for many elements in sparse matrices – hardware optimized to process sparse matrices uses this concept to reduce resource consumption.

While these architectural improvements in digital electronic DNN hardware have yielded large efficiency gains for DNNs, we are reaching the limits of digital electronics in terms of performance gains at the device level, as mentioned in Chapter 1. Furthermore, wiring constraints prevent full connectivity in a large PE array, resulting in message passing between PEs, which increases latency. For these reasons, analog electronic circuits such as memristor crossbar arrays [48] have been proposed, as they can sum partial products along a wire with low latency by Kirchhoff's current law. However, they are bounded in speed, energy consumption and accuracy by parasitic capacitance, resistance, and current leakage. There also remain challenges in demonstrating endurance (maximum number of cycles before performance degradation) and repeatable fabrication of large arrays [49].

2.2.5 Optical DNN accelerators

Another option to boost DNN processor performance is to use optics for fast and potentially massively parallel linear algebra.

Integrated optical schemes can compute with ultra-low latency [50]–[53] (all weight-stationary), but they are confined to 2D planar space as they perform data routing and fan-out with integrated on-chip waveguides. With the resulting $\mathcal{O}(K)$-depth circuits, limits on control, multiplexing, component area and on-chip losses constrain scalability to a vector length K of hundreds of elements (with only $K \approx 10$ demonstrated experimentally, e.g., in Ref. [53]). Thus, they seem most appropriate for edge computation with small models, assuming the best-case scenario where their peripheral hardware requirements can be miniaturized using integrated optical sources and co-packaged electronics.

A free-space optical architecture can reach greater connectivity than integrated schemes thanks to its out-of-plane data routing and copying that ultimately allow for tighter packing of weighting elements and detectors. Given the prevalence of megapixel-scale optoelectronic components (e.g., cameras and displays) [54], [55], '3D' data transfer can occur between potentially millions of spatial modes, resulting in high scalability. One downside of note, however, is that free-space optical components can be bulkier in volume than their 2D counterparts, so they are best-suited for data-center-type applications.

3D optical MVM accelerators were among the first to be investigated by pioneers in the field [56]–[59], but they did not achieve widespread adoption for two reasons. First, competition from digital electronic computers limited their application space [60]. Second, they suffered from a lack of flexibility, as experimental demonstrations employed fixed weighting masks or large modulators and were small in scale (input vectors with $K \approx 10$ here as well).

Recently, theoretical analyses showing the potential of ultra-low-energy computing (e.g., our theory paper [34] led by Dr. Ryan Hamerly), have driven a resurgent interest in 3D optical processors for DNNs. However, experimental proof-of-concept realizations of 3D ONNs have been limited to a single vector-vector dot product per time step [61] or have relied on device-specific training to offset systematic errors (e.g., [62], with $K = 100$ and matrix size up to $K \times N = 100 \times 25$). Others (e.g. [63], [64]) have required all-electronic neural

network layers in post-processing that can perform a significant amount of computation [65] at the cost of increased energy and latency. Diffractive and convolutional systems [66]–[68] have yielded promising results in the classification of datasets with $K\sim$1,000-10,000, but they include complex physical models of the hardware in their digital electronic model training. Furthermore, due to sensitivity to hardware imperfections, these models also have to be retrained with measured outputs to achieve high accuracy. For example, without retraining, the classification accuracy of the diffractive neural network by Zhou et al. falls to 63.9% on the MNIST dataset [66]. These systems thus require an entirely new deep learning framework since they do not process standard DNN models, as well as additional infrastructure for retraining, impeding large-scale use in data centers. A summary of these points alongside demonstrated classification accuracies is reported in Table 2.1.

Table 2.1: Experimentally demonstrated classification accuracies of prior proof-of-concept 3D optical DNN accelerators

Ref.	Dataset	Standard training?[1]	# weights in digital back-end layer(s)[2]	Accuracy (%)	
				Basic[3]	Retraining with hardware outputs[4]
[62]	MNIST	No	0	N/A	88.0
			250	N/A	92.7
[63]	MNIST	Yes	5040, 350	N/A	93.1
[64]	QuickDraw[5]	Yes	40	N/A	79.0
[66]	MNIST	No	0	63.9	96.0
			6M[6]	N/A	96.6
	Fashion-MNIST		6M[6]	N/A	84.6
[67]	MNIST	No	256	92	98
[68]	MNIST	No	0	81[7]	-
	Fashion-MNIST		0	73	-

[1] Standard training includes FC and CONV layers and common augmentation techniques that add noise or distort the inputs to avoid overfitting and help the model generalize.

[2] All-digital-electronic FC layers after the optical system. 0: there are none; multiple values: multiple layers. Note that an FC-NN where the inputs are cropped to 23 × 23 pixels and connected directly to the outputs (i.e., one layer of 5290 weights) can achieve 92.6% classification accuracy on the MNIST test set.

[3] Run inference without adjusting the weights based on outputs from experimental setup. N/A means this accuracy was not reported.

[4] Includes adaptive training or fine-tuning based on outputs from the experimental setup.

[5] The authors hand-picked 10 classes and removed 'inappropriate' images.

[6] Architecture unclear, but Suppl. Table S2 in Ref. [66] seems to indicate >6 million digital electronic operations.

[7] The authors report correct classification of 88% of images picked from within the 92% correctly classified images by the digital model, so the accuracy could higher than 81% if the originally misclassified images by the electronic hardware were correctly classified by the optical setup. There were 50 test images for each dataset.

In conclusion, while these optical systems hold promising theoretical performance improvements over digital electronics in terms of energy consumption and latency, the potential for 'plug-and-play' replacement of prior accelerators remains to be demonstrated.

Chapter 3

Elements of 3D Optical Computation

Before delving into our optical DNN accelerators, in this chapter, I describe the key elements of 3D optical computation, from displays, to optical propagation and data routing, to detectors. Displays can project inputs or impart weights or fan-out patterns onto signals, making them critical components in our ONNs. Data transmission and copying occur through optical propagation in free space. Detectors convert the signal from optics back to electronics in order to perform the summation, nonlinearity, and any electronic processing that might be required after the DNN layer.

3.1 Displays

Digital micromirror devices (DMDs)

A 2D array of micromirrors (digital micromirror device or DMD) is conceptually the simplest display. Each pixel is a μm-scale mirror that pivots between 'on' and 'off' states on a hinge. Electrodes control the motion of each mirror via the electrostatic force. DMDs thus provide pixel-wise binary amplitude modulation of an incident light field with up to ∼10 million pixels at relatively high switching speeds (up to 10s of kHz) [69]. They can achieve greyscale modulation if the detector integrates over many on-off cycles. DMDs are used in a wide variety applications such as projectors [69], optical traps [70] and holography [71].

Liquid-crystal-on-silicon spatial light modulators (LCoS SLMs)

Liquid crystals (LCs) are made up of elongated molecules that align to an applied electric field. When the molecules are ordered, the resulting material is birefringent, which means that the refractive index experienced by incident light depends on its polarization. The LC has two axes, called 'ordinary' and 'extraordinary'. The refractive index along the extraordinary axis, n_e, is voltage-dependent, while the refractive index along the perpendicular ordinary axis is constant with voltage. When light passes through a liquid crystal, the phase shift $\Delta\phi$ experienced by its polarization component that is parallel to the extraordinary axis is:

$$\Delta\phi(V) = \tfrac{2\pi}{\lambda}(n_e(V) - n_o)a \tag{3.1}$$

where V is voltage, λ is wavelength, n_e is induced extraordinary refractive index (modulated by V), n_o is ordinary refractive index and a is liquid crystal thickness [72]. In theory, the polarization component along the ordinary axis does not experience a phase shift (though in practice, we found the phases to be somewhat coupled in interferometric measurements; see Appendix).

A liquid-crystal-on-silicon spatial light modulator (LCoS SLM) is composed of up to millions of μm-scale LC cells that are individually electrically addressable and updatable at ∼10s to 100s of Hz. LCoS SLMs can therefore perform independent pixel-wise phase modulation of one polarization component of an incident optical field. They can be used in what I will call 'phase' or 'amplitude' modes. In phase mode, the polarization of the incident light is parallel to the extraordinary axis of the device. In amplitude mode, the incident light is instead polarized to 45°, e.g., by adding a half-wave plate before the SLM. The SLM pixels can then be thought of as individual variable wave plates, or polarization rotators. If a polarizer is then added after the SLM to reject one polarization component, the phase SLM can effectively be used as an array of high-precision (∼8-bit) amplitude modulators. SLMs also exist in reflective or transmissive modes. Transmissive SLMs have lower fill factors since they need electronic circuitry between pixels and therefore, have lower diffraction efficiency. We therefore used reflective SLMs in our setups.

Single-spatial-mode source arrays

An array of high-speed sources can also be used as a display. For example, μm-scale vertical-cavity surface-emitting lasers (VCSELs) [73] or micro-light-emitting diodes (μLEDs) [74], [75] can be packed into a dense array of single-spatial-mode emitters. While VCSEL arrays has been restricted to tens or hundreds of elements [76], [77], increasing array sizes and laser wall-plug efficiencies are active areas of research. Large-scale μLED arrays, on the other hand, are becoming commercially available [78], and high-speed arrays of photonic crystal cavities are also being investigated [79]. VCSEL arrays can exhibit temporal coherence, meaning that they have the potential to maintain a consistent phase relationship over time, whereas μLED are temporally incoherent. While certain optical MVM configurations require temporal coherence [34], [80], others may benefit from sources with lower temporal coherence to reduce fringes in interference effects from stray light.

3.2 Data transmission and copying (*fan-out*)

In this section, I will describe how with free-space optics, we can: 1) transmit inputs and 2) copy data in a reconfigurable multicast to the detector plane. I will first review optical propagation through lenses and diffractive elements. Lenses can passively transfer a signal from one 2D plane to another. In our hardware configurations, this mapping of light from the input to output plane happens through a series of optical Fourier transforms, as will be explained below. I will also show how we can achieve 2D multicast (*fan-out*) by generating a 2D spot array with a diffractive element, then optically convolving this spot array with our image in the output plane (again, using the Fourier transform property of lenses). To clarify, these Fourier transforms and convolutions are not performing any MAC operations here, but are the means to the end of the 2D multicast. The multicast can be tailored to match a DNN layer's shape, and is independent of the activations and weights – it only has to be updated if the DNN model shape changes. As fast, low-energy data transmission and copying are important bottlenecks in electronics, these concepts are key to our optical deep learning accelerators.

3.2.1 Fourier transform by a lens

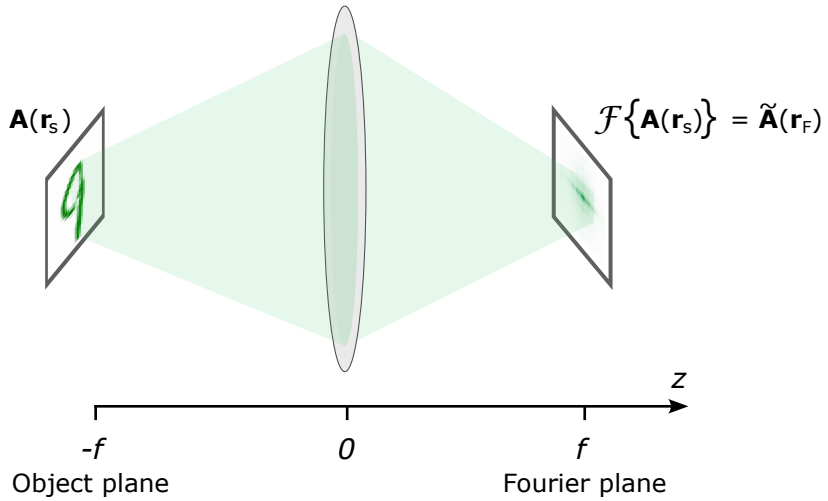

Figure 3.1: Fourier transform by a lens of input field $\mathbf{A}(\mathbf{r}_\mathrm{s})$ in the object plane to $\mathscr{F}\{\mathbf{A}(\mathbf{r}_\mathrm{s})\} = \tilde{\mathbf{A}}(\mathbf{r}_\mathrm{F})$ in the Fourier plane. The optical axis (light propagation direction) is perpendicular to the 2D object and Fourier planes. These planes are at distances -f and f from the lens along the optical axis, respectively, where f is the lens's focal length. $\mathbf{r}_\mathrm{s}$ is the position within the object plane, $\mathbf{r}_\mathrm{F} = 2\pi \cdot \mathbf{r}_\mathrm{s}/(\lambda \cdot f)$ is the same in the Fourier plane.

As Fig. 3.1 illustrates, a lens performs a spatial Fourier transform $\mathscr{F}\{\cdot\}$ of the 2D optical field distribution $\mathbf{A}(\mathbf{r}_\mathrm{s})$ from the object plane to the Fourier plane to yield $\tilde{\mathbf{A}}(\mathbf{r}_\mathrm{F}) = \mathscr{F}\{\mathbf{A}(\mathbf{r}_\mathrm{s})\}$ [81]. The locations of these planes along the optical axis (z) as well as the feature sizes are defined by the lens's key property: its focal length f. Mathematically, $\mathbf{r}_\mathrm{F} = 2\pi \cdot \mathbf{r}_\mathrm{s}/(\lambda \cdot f)$, where λ is the wavelength of the light.

If, for example, the input field in the object plane is a point source (modeled as a Dirac delta function), then the Fourier plane is fully illuminated. In reality, of course, the point source is not infinitesimally small, but rather follows a distribution such as a Gaussian of $1/\mathrm{e}^2$ diameter $2w_0$. Then, the ideal distribution in the Fourier plane is also a Gaussian, but of $1/\mathrm{e}^2$ diameter $2\lambda f/(\pi w_0)$ (not accounting for aberrations introduced by imperfect lenses), which can be much larger than the initial Gaussian. A lens can therefore act as one type of fan-out element in itself, since it takes one small data point and broadcasts it to a much greater area.

3.2.2 Imaging

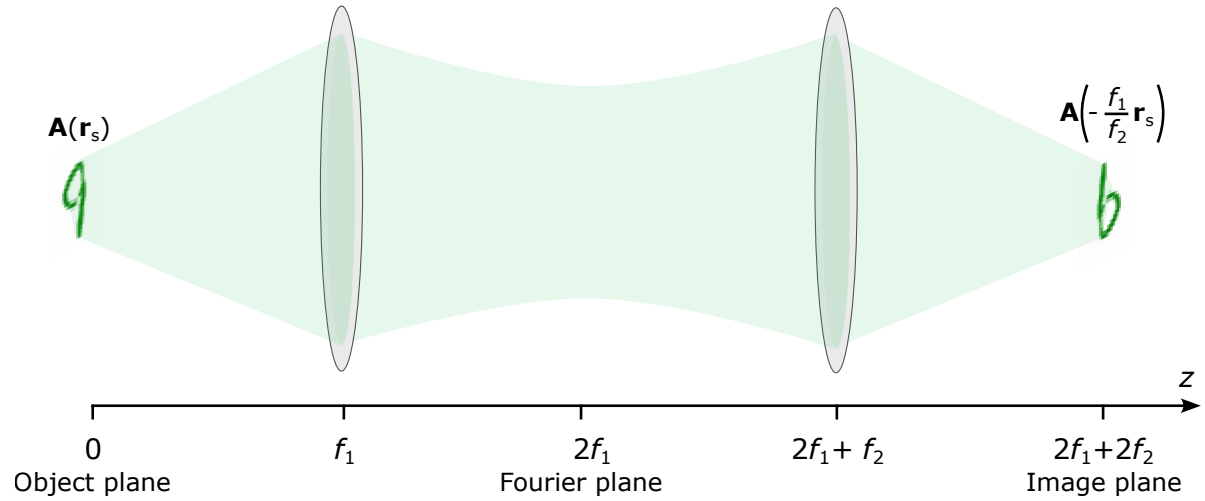

Figure 3.2: Imaging by a $4f$ lens system of input field $\mathbf{A}(\mathbf{r}_s)$ to $\mathbf{A}(-(f_1/f_2)\mathbf{r}_s)$ in the image plane. The image is flipped and magnified by f_2/f_1.

We can add another lens with focal length f_2 after the initial lens of focal length f_1, which gives us the Fourier transform of the Fourier transform of the input field (see Fig. 3.2). In other words, we have routed the input array in the object plane to the output image plane, unmodified except for a magnification factor of $-f_2/f_1$. (We can use this magnification factor to tune the size of the output array to match our desired detector or weight array pixel widths.) This arrangement of lenses is called a $4f$ system because there is a distance of four focal lengths along the optical axis from the object plane to the image plane.

3.2.3 Spot array generation

Another concept that will be used in our fan-out scheme is spot array generation by a phase mask (a diffractive optical element or DOE, see Fig. 3.3). Spot arrays appear in a wide variety of fields, such as the control of positions of atoms in quantum computing [82] and nanopatterning [83].

With a uniformly illuminated phase object in the object plane, the optical field in the Fourier plane will be the Fourier transform of the phase distribution imparted by the object. In the case of a diffraction grating, for example, interference maxima appear in the Fourier

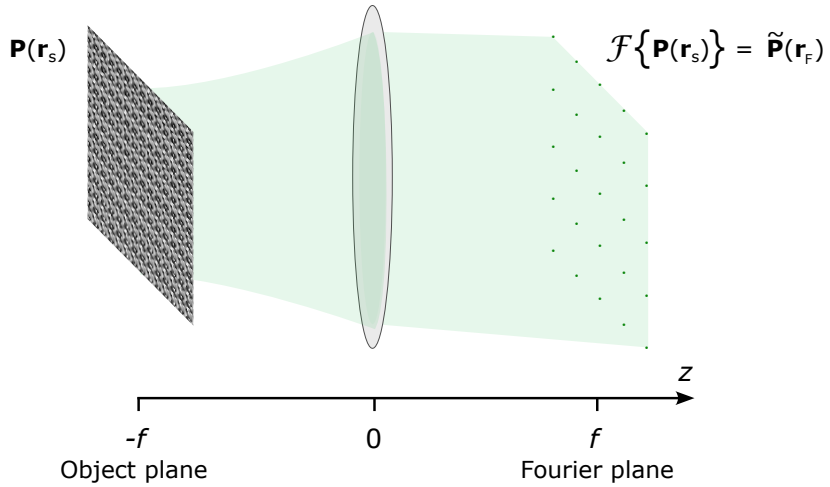

Figure 3.3: Spot array generation in the Fourier plane by a uniformly illuminated phase mask in the object plane. Phase mask calculated by fixed-phase weighted Gerchberg-Saxton algorithm.

plane, yielding spots at locations determined by the grating equation. However, these spots do not have uniform intensities. An array of microlenses can also generate an array of spots, which has in fact been used in a recent ONN demonstration [64]. While these optical elements have the advantage of being passive, neither a grating nor a microlens array can be reconfigured or create spots at arbitrary locations.

Instead, we can use a programmable LCoS SLM to display a reconfigurable phase mask, as described above. The phase offsets within this mask can be calculated via the Gerchberg-Saxton algorithm [84] to generate large-scale, high-uniformity spot arrays. From the starting point of an array of random values, this algorithm computationally iterates through Fourier transforms and inverse Fourier transforms until it reaches the target distribution in the Fourier plane (more on this below). In a modified form, called the fixed-phase weighted Gerchberg-Saxton (WGS) algorithm and reported in Ref. [85], a normalizing correction is applied at each iteration to improve spot intensity uniformity. The phase is also fixed in Fourier space after some number of iterations T for faster convergence. Ref. [85] achieved intensity uniformities of >98% in arrays of >1,000 spots at arbitrary locations (measured in experiment) after also including camera feedback into the correction term. The pseudocode below illustrates the fixed-phase WGS algorithm, followed by an explanation.

```
I_s = ones(num_SLM_pixels) # input laser field (ones means flat)
phi_s = rand(num_SLM_pixels) * 2*pi
target = grid(intensities) # spots at desired locations & intensities
past_corr = grid(ones) # spots at desired locations, intensities = 1

for t in range(T_total):
    F = fft2( sqrt(I_s) * exp(1i*phi_s) )
    I_F = abs(F)^2
    corr = sqrt(mean(I_F)/I_F[target>0]) * past_corr[target>0]
    past_corr = corr
    if t < T:
        phi_F = angle(F)
    S = ifft2(corr * sqrt(target) * exp(1i * phi_F) )
    phi_s = angle(S)

phi_s = mod(phi_s, 2*pi) * 255/(2*pi) # display on 8-bit SLM
```

The phase pattern ϕ_S is initialized to a random 2D matrix, which is multiplied by the input light field incident on the SLM (square root of the intensity I_s, assuming a uniform incident phase), then Fast Fourier Transformed (FFT'ed) to simulate the optical field distribution in the Fourier plane (F). The amplitude of F is replaced by the target amplitude (i.e., the spot array), where the target spot locations need to be transformed by the grid() function to real space, accounting for lens focal length, wavelength and SLM pixel size. This new distribution is multiplied by the normalizing correction, which is the mean of the amplitude of F divided by the amplitudes of F at the desired spot locations. This step is followed by an inverse FFT to yield S, whose phase ϕ_s is kept as the new SLM pattern, replacing the random initialization. The amplitude is once again swapped out, but this time, for input field $\sqrt{I_s}$, which concludes the first iteration. This process is repeated in a loop for some number of iterations T_{total} or while an error criterion has not been met. If the number of iterations reaches $t > T$ before convergence, where T is a fixed hyperparameter (typically between 10 and 20), then ϕ_F, the phase of F, is held constant. Once the phase map to

display on the SLM has been determined by this method, it should be summed with the flatness correction map at the source wavelength provided by the manufacturer. The map can be further refined with experimental data by including feedback from the camera into the correction term.

If we want to use a passive component instead, the LCoS SLM can be replaced by a fixed metasurface fabricated to impart the appropriate phase shifts onto input light. While this option can lower energy consumption, it is generally not reconfigurable. An exception may be a surface made of phase-change materials that can have local phase offsets set by laser pulses, though transmission efficiency, bit precision (number of levels), and repeatability remain open questions [86].

3.2.4 Image replication

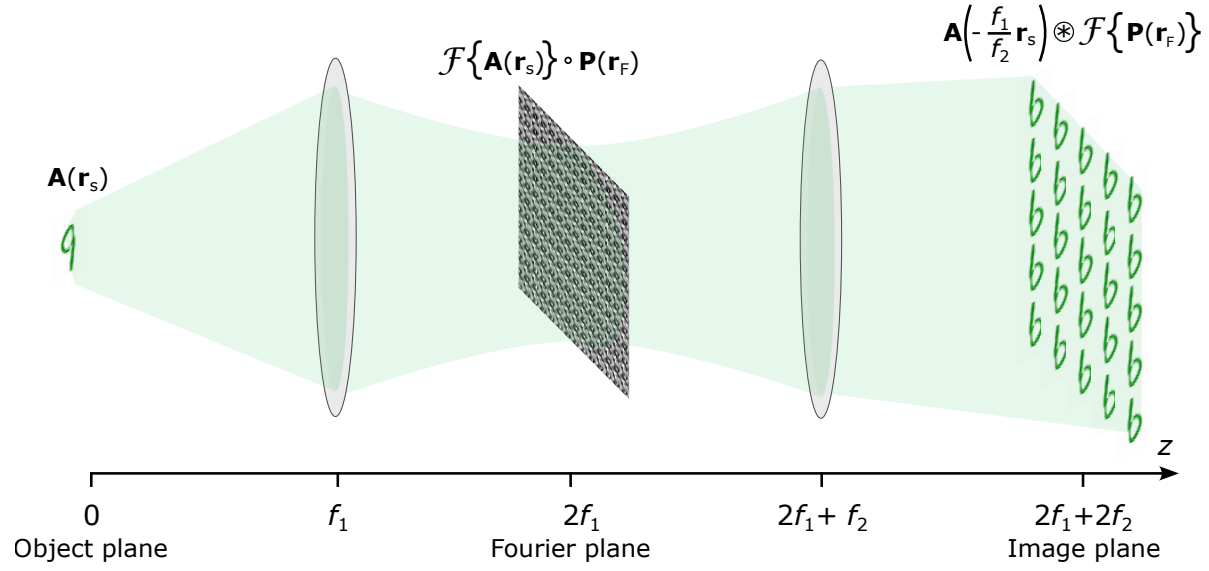

Figure 3.4: With a spot array generation pattern in the Fourier plane of a $4f$ system, an input is replicated in the image plane (2D multicast thanks to 3D optical propagation).

By combining imaging with spot array generation, we can replicate inputs from the object plane to the image plane (see Fig. 3.4). An input field in the object plane is spatially Fourier transformed by a lens to the Fourier plane. The mask $\mathbf{P}(\mathbf{r}_s)$ displayed by an LCoS SLM in the Fourier plane then imparts the phases of the spot array generation pattern onto the Fourier transformed input field (pixel-wise multiplication by phase pattern). In the image

plane, by Fourier convolution theory, this product becomes a convolution – we obtain the convolution of the Fourier transforms of the individual distributions in the Fourier plane. Therefore, in the image plane, we have convolved the image of the input field with the spot array, i.e., produced image replicas.

Conceptually, we can think of the spots as individual Dirac delta functions. We know that the convolution of an arbitrary function with a Dirac delta returns the same function offset to the Dirac delta's location in the 2D plane. This means that with this optical convolution, our output image is copied to the location of each spot, yielding our 2D multicast. The spot array generation pattern defines the number of image replicas and their locations in the image (detector) plane, and it is independent of the input data. When we use this concept in our ONN, we will see that it is also independent of the weights. Therefore, the spot array generation pattern only needs to be updated if the DNN model *shape* changes.

3.3 Detectors

Photodetectors convert the optical signal back to the electronic domain for further processing or readout. Small, μm-scale photodetectors (PDs) [54], [87]–[90] can be highly efficient, reaching responsivities of ≈ 0.2 A/W [90] and GHz speeds. While these detectors have lower sensitivities than a conventional camera, they are optimized to reduce overall energy consumption and increase speed. They can be used in conjunction with a small transimpedance amplifier (TIA) [91] or without amplification, in a 'receiverless' configuration [54].

Chapter 4

Digital Optical Neural Network

This chapter is adapted from work[1] *reported in Ref. [30].*

In this first project, we introduced an ONN that encodes digital inputs and weights into reconfigurable on-off optical pulses for high-efficiency data transmission, but not computation. In other words, incoherent optical paths replace electrical on-chip interconnects, but not multipliers. Prior research into digital optical interconnects focused on integrated point-to-point connections [92], [93], free-space point-to-point transmission [94], [95], and small-scale free-space multicast [96]. These ideas would be difficult to scale in DNNs since they incur significant overhead in number of components and introduce compounded component losses. Here, by contrast, in our 'digital optical neural network' (DONN), three-dimensional free-space optical elements passively transmit and copy data (fan-out) from memory to large-scale electronic multiplier arrays for the specific application of matrix multiplication. The length-independence of this optical data routing enables scalable systems, where single transmitters are fanned out to many receivers with fast and energy-efficient links.

We first illustrate the DONN architecture and discuss a possible implementation. Then, in a proof-of-concept experiment, we demonstrate that digital optical transmission and fan-out with cylindrical lenses has little effect on the classification accuracy of the MNIST handwritten digit dataset (<0.6%). The primary cause of this drop in accuracy is crosstalk,

[1] This project was done in close collaboration with Dr. Alexander Sludds, who wrote most of the data acquisition and training code and co-developed the concept and energy analysis.

which is the unwanted contamination of one signal by another, in this case through blurring of the optical outputs. Because crosstalk is deterministic, it can be compensated: with a simple correction scheme, we reduce our bit error rates by two orders of magnitude. Alternatively, crosstalk can be greatly reduced through optimized optical design.

We also compare the energy consumption of digital optical interconnects (including light source energy) against that of electronic interconnects over distances representative of logic, multi-chiplet interconnects and multi-chip interconnects in a 7 nm CMOS node. Multiple chips [97] or partitioned chips [98], [99] are regularly employed to process large networks since they can ease electronic constraints and improve performance over a monolithic equivalent through greater mapping flexibility [100], at the cost of increased communication energy. Our calculations show an advantage in data transmission costs for distances ≥ 5 μm (roughly the size of the basic computational element: an 8-bit MAC unit). The DONN thus scales favorably with respect to very large DNN accelerators: the DONN's optical communication cost for an 8-bit MAC, i.e., the energy to transmit two 8-bit values, remains constant at ∼3 fJ/MAC, whereas multi-chiplet systems have much higher electrical interconnect costs (∼1,000 fJ/MAC), and multi-chip systems have a higher energy consumption still (∼30,000 fJ/MAC). Thus, the efficient optical data distribution provided by the DONN architecture can enable continued growth of DNN performance through increased model sizes and greater connectivity.

4.1 Concept

Figure 4.1 shows our DONN scheme, which is output stationary. In the matrix-matrix multiplication $\mathbf{W}\cdot\mathbf{X}$, optical elements transfer and fan out input activation and weight bits to electronic multipliers, where each element X_{kb} is fanned out N times, and W_{nk} is fanned out B times. At the first time step, the input activation matrix transmitters fan out the first bit of each of the elements in the first row of the input activation matrix $X_{1b}, \forall b \in \{1...B\}$ to the PEs. Simultaneously, a row of weight matrix transmitters broadcast the weight bits W_{n1} to each PE. The photons from these input activation and weight bits generate photoelectrons in the detectors, producing the voltages required at the inputs of electronic multipliers (either

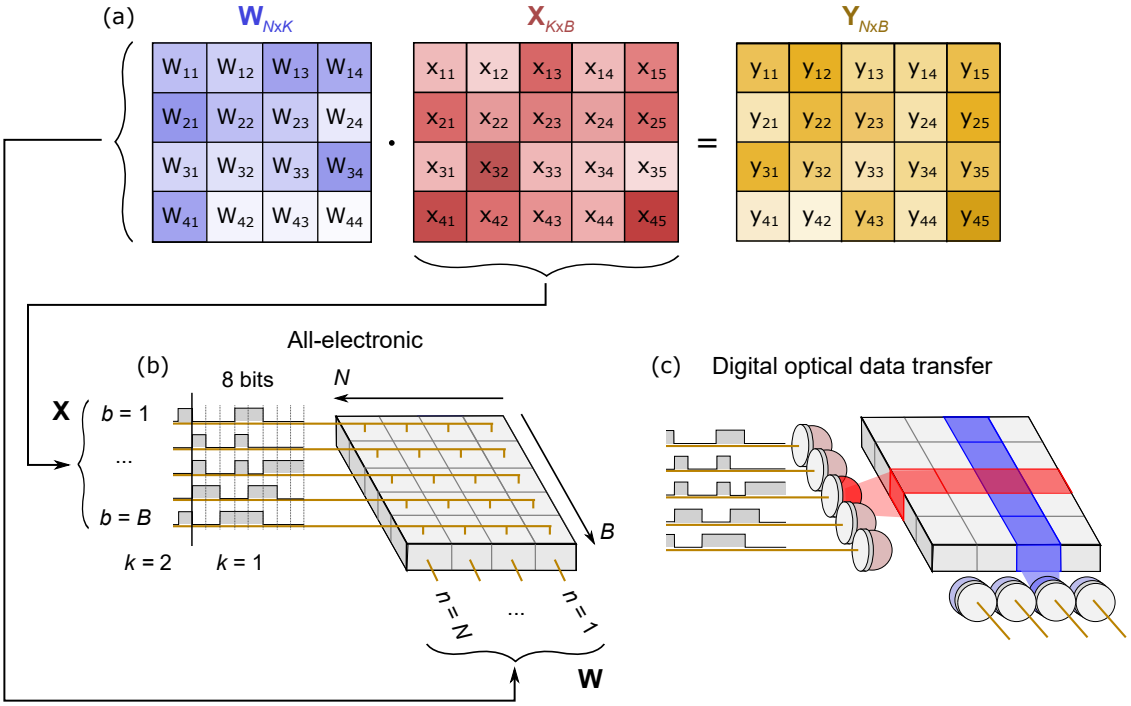

Figure 4.1: Digital electronic and optical implementations of MVM. **a**, Matrix representation of one layer of an FC-NN with B-sized batching (nonlinearity not shown). **b**, Example bit-serial multiplier array, with output-stationary accumulation across k. Fan-out of $\mathbf{X}$ across $n \in \{1...N\}$; fan-out of $\mathbf{W}$ across $b \in \{1...B\}$. The size of the multiplier array (PEs) is equal to the size of the output matrix ($\mathbf{Y}$). All-electronic version with fan-out by copper wires (for clarity, fan-out of $\mathbf{W}$ not illustrated). **c**, Digital optical neural network version, where $\mathbf{X}$ and $\mathbf{W}$ are fanned out with optics and transmitted to an array of photodetectors. Each pixel contains two photodetectors, where the input activations and weights can be separated by, e.g., polarization or wavelength filters. Each photodetector pair is directly connected to a multiplier in close proximity.

0 V for a '0' or 0.8 V for a '1'). The input and weight bits can be discriminated by being encoded into different wavelengths or polarization states. After 8 time steps, a multiplier has received 2×8 bits (8 bits for the input activation and 8 bits for the weight), and the electronic multiplication occurs as it would in an all-electronic system. The activation-weight product is completed, and is added to the locally stored partial sum. The entire matrix-matrix product is therefore computed in $8 \times K$ time steps. Instead of this bit-serial implementation, bits can also be encoded spatially, using a bus of parallel transmitters and receivers. The trade-off between added energy and latency in bit-serial multiplication versus increased area from photodetectors for a parallel multiplier can be analyzed for specific applications and CMOS nodes.

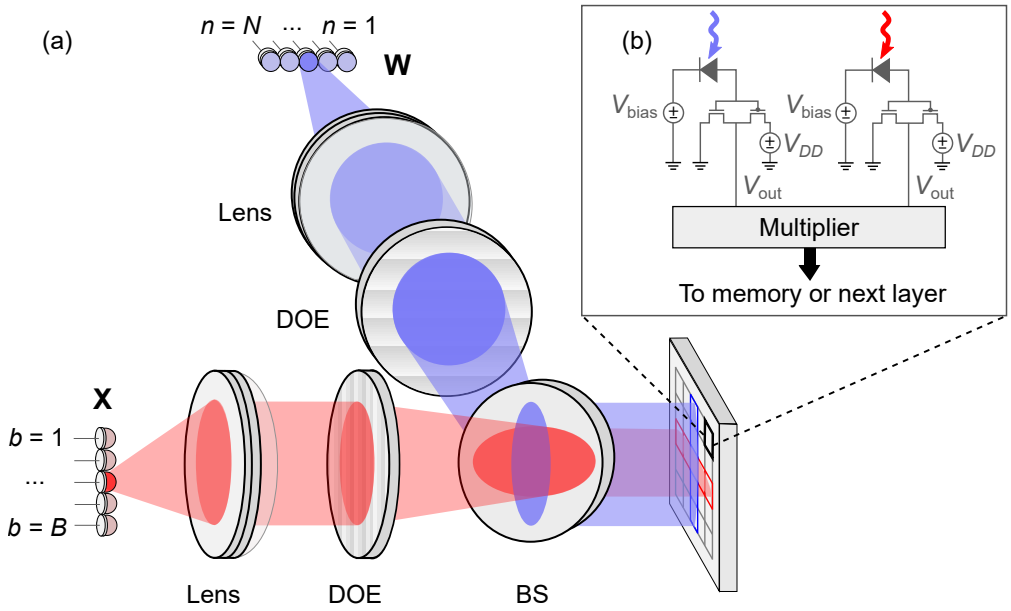

Figure 4.2: Digital optical neural network. **a**, Digital inputs and weights are transmitted electronically to an array of light sources (red and blue, respectively, illustrating different paths). Single-mode light from a source is collimated by a spherical lens, then focused to a 1D spot array by a diffractive optical element (DOE). Single source illuminated for illustrative purposes, but light from all sources transmitted in parallel. A 50:50 beamsplitter brings light from the inputs and weights into close proximity on a custom CMOS receiver. **b**, Example circuit (by A. Sludds) with 2 photodetectors (biased by voltage V_{bias}) per PE: 1 for input activations; 1 for weights. Received bits (V_{out}) proceed to multiplier, then memory or next layer.

We illustrate an example DONN implementation in Fig. 4.2. The optical sources in the linear arrays can be selected from among those described in Chapter 3, e.g., VCSELs or

μLEDs. Each source emits a cone of light into free space, which is collimated by a spherical lens. A DOE (can be a cylindrical lens, metasurface, or LCoS SLM paired with another lens) focuses the light from each input to a 1D spot array on a 2D receiver. The input activations and weights are brought into close proximity using a beamsplitter. 'Receiverless' photodetectors [54] convert the individual optical bits to the electrical domain. An electronic multiplier then multiplies each 8-bit input activation with its corresponding weight. The output is either saved to memory, or routed directly to another DONN that implements the next layer of computation. Note that the data distribution pattern need not be confined to regular rows and columns, as DOEs can be reconfigurable (see Chapter 3). Furthermore, since free-space optical elements and free-space propagation are highly efficient, most length- or receiver-number-dependent losses can be attributed to imperfect focusing, e.g., from optical aberrations far from the optical axis. These effects can be mitigated through judicious optical design. We assume for the remainder of our analysis in this chapter that energy is length-independent.

4.2 Experimental setup

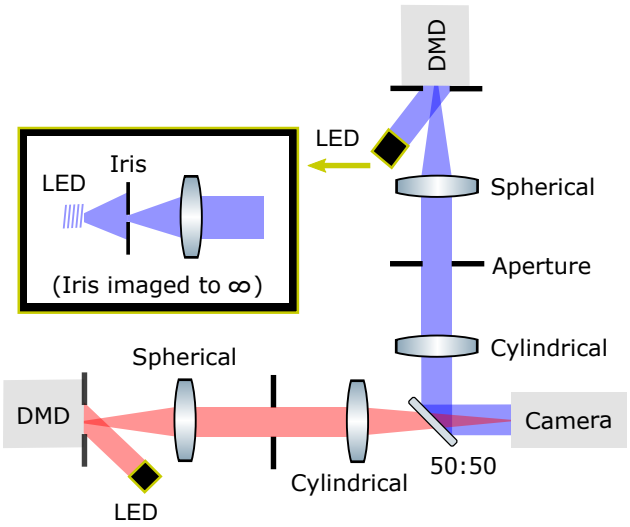

Figure 4.3: Our experimental implementation of the digital optical neural network. Digital micromirror devices (DMDs) illuminated by LEDs act as stand-ins for high-speed sources; cylindrical lenses are fan-out elements in the input activation and weight arms.

We used the DONN implementation shown in Fig. 4.3 to test digital optical data transmission and fan-out for DNNs. Digital micromirror devices (DMDs, Texas Instruments DLP3000, DLP4500) illuminated by spatially-filtered and collimated LEDs (Thorlabs M625L3, M455L3) act as stand-ins for the two linear source arrays. For the input activations/weights, each 10.8 μm-long mirror in one DMD column/row reflects red/blue light toward the detector ('1') or a beam dump ('0'). Then, $f = 100$ mm spherical lenses and $f = 100$ mm cylindrical achromatic lenses image each DMD pixel to an entire row/column of superpixels of a color camera (Thorlabs DCC3240C). Each camera superpixel is made up of four pixels of size $(5.3\ \mu\text{m})^2$: two green, one red and one blue. The Thorlabs camera acquisition program applies a 'de-Bayering' interpolation to automatically extract color information for each sub-pixel; this interpolation causes blurring, and therefore it increases crosstalk.

4.2.1 Image processing

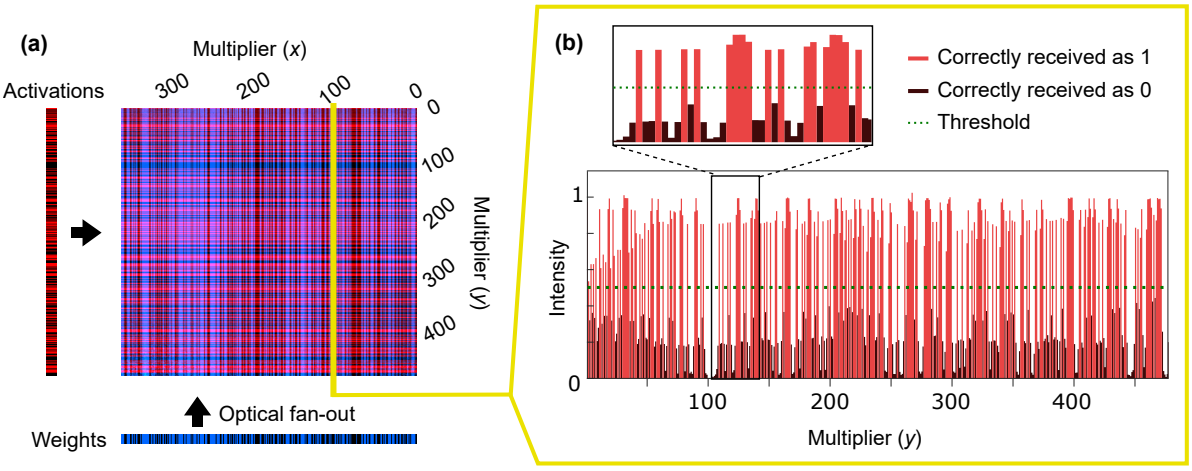

Figure 4.4: Background-subtracted and normalized camera output from our proof-of-concept free-space digital optical neural network experiment. Random vectors of '1's and '0's displayed on DMDs for characterization. **a**, Full received 2D image. **b**, One randomly chosen column of the image: pixels received as '1' in red and '0' in black. Bits are correctly received if they are on the appropriate side of the threshold (though ideally, '0' bits would have 0 intensity, and '1' bits would be saturated at 1).

To process each image received by the camera, we subtracted the background, normalized, then thresholded the outputs by a fixed value for each channel. We acquired normalization and background curves with all DMD pixels in the 'on' and 'off' states, respectively. This

background subtraction and normalization could be implemented on the receiver chip by biasing each pixel by some fixed voltage. If the detected intensity was above the threshold value, it was labeled a '1'; below threshold, a '0'. Figure 4.4 shows an example of a background-subtracted and normalized image, captured on the camera when the digital micromirror devices (DMDs) displayed random vectors of '1's and '0's. The error that causes the '0' values to receive >0 optical intensity is primarily crosstalk, as will be discussed later.

4.2.2 Crosstalk correction

Since crosstalk is deterministic, it can be compensated by post-processing. To illustrate this principle, we evaluated the performance of a simple crosstalk correction scheme, in which we multiplied each line of an image detected on the camera by a tridiagonal crosstalk reduction matrix:

$$\begin{bmatrix} \overline{I_{1n}} \\ \overline{I_{2n}} \\ \vdots \\ \\ \end{bmatrix} = \begin{bmatrix} 1 & -\xi & 0 & & \\ -\xi & 1 & -\xi & & \\ 0 & -\xi & 1 & \ddots & \\ & & \ddots & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} I_{1n} \\ I_{2n} \\ \vdots \\ \\ \end{bmatrix} \tag{4.1}$$

where ξ is a constant value per channel and $\overline{I_{:n}}$ is the corrected line of the camera image, which is subsequently renormalized. ξ was measured to be $\sim$0.19 and $\sim$0.18 for the red and blue arms, respectively, from a calibration image of alternating '1's and '0's transmitted by the DMDs. The results of the crosstalk correction are shown in Fig. 4.5. We quantify the quality of the correction by calculating the bit error rate in the next section. In the Discussion (Sec. 4.5), we propose extensions and alternatives to this scheme that do not require post-processing.

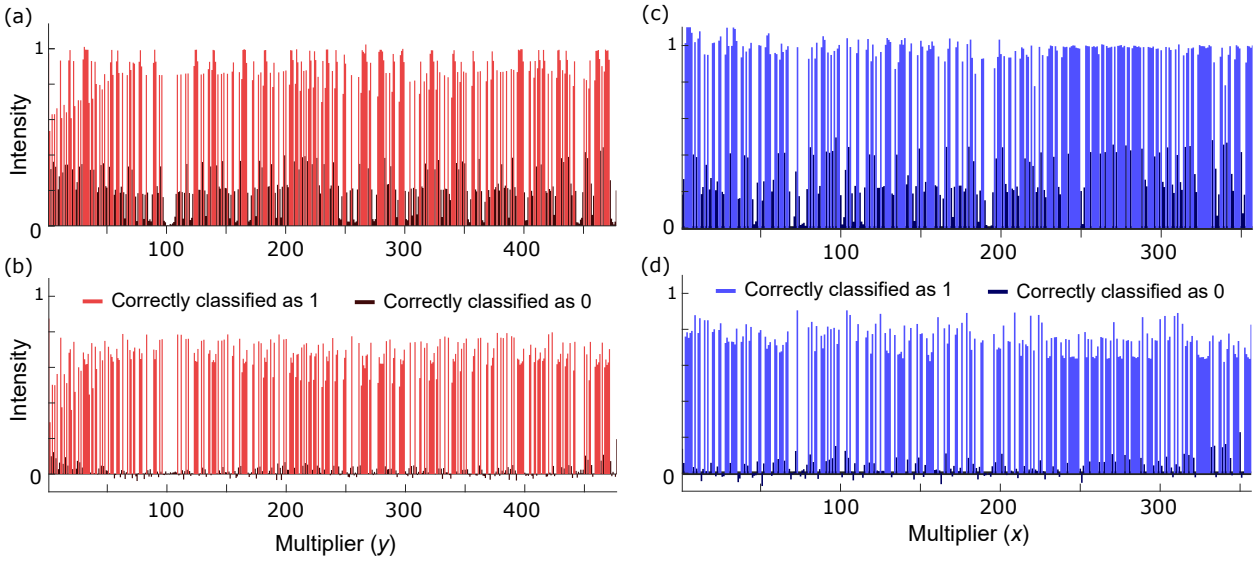

Figure 4.5: Randomly chosen lines from the received image shown in Fig. 4.4. One column from the red channel before (**a**) and after (**b**) crosstalk correction. **c-d**, Same, but for the blue channel.

4.3 Measured accuracy

4.3.1 Bit error rate

In our first experiment, we determined the bit error rate of our system. We compared the parsed values from the camera with the known values transmitted by the DMDs, and defined the bit error rate as the number of incorrectly received bits divided by the total number of received bits. The camera's de-Bayering algorithm, optical aberrations and misalignment caused some crosstalk between pixels, as can be seen in Figs. 4.4 and 4.5. Using a region of 357×477 superpixels on the camera, we calculated bit error rates (in a single image) of 1.2×10^{-2} and 2.6×10^{-4} for the blue and red channels, respectively. When we confined the region of interest to 151×191 superpixels (centered and less aberrated), the bit error rate (averaged over 100 different trials, i.e., 100 pairs of input vectors) was 4.4×10^{-3} and 4.6×10^{-5} for the blue and red arms.

We also measured the bit error rate after crosstalk correction. The bit error rates for the blue and red channels then respectively dropped to 2.9×10^{-3} and 0 for the 357×477-pixel single image and 2.6×10^{-5} and 0 for the 151×191-pixel, 100-image average. In other words,

after crosstalk correction, there were no errors in the red channel, and the errors in the blue channel dropped by two orders of magnitude in the centered region of interest.

4.3.2 DNN inference

Next, we experimentally tested the DONN's effect on the accuracy of classification of 500 MNIST images using FC-NNs with one and two hidden layers of 100 activations each. The MNIST images were downsampled from 28×28 to 7×7, which were then flattened to 1D 49-element vectors. The weights were trained on the MNIST training set with categorical cross-entropy as the loss function and dropout between layers to prevent overfitting. The DMDs displayed the activations and pre-trained weights, which propagated through the optical system to the camera. Then, as described before, the images were background subtracted and normalized, and the CPU multiplied each input activation with each weight, accumulated the partial products, and applied the nonlinear function (ReLU). The CPU then fed the outputs back to the input activation DMD, and the weights were updated for the next layer of computation.

As reported in Table 4.1, we measured a 0.6% drop in classification accuracy for the DONN versus the ground truth values (or 3 additional incorrectly classified images) with a three-layer FC-NN. In the case of the DNN with a single hidden layer ('2-layer' case), we achieved similar results: a 0.4% drop in accuracy, or 2 misclassified images. No crosstalk error correction was applied to these results to illustrate the worst-case impact on accuracy.

Table 4.1: MNIST classification accuracy of DONN (no crosstalk correction applied) versus all-electronic ground truth with FC-NNs of different depths

	2 layers	3 layers
Electronic (ground truth)	95.8%	96.4%
DONN	95.4%	95.8%

4.4 Energy analysis of an optimized system

In this section, we compare the theoretical interconnect energy consumption of an optimized implementation of the DONN with its all-electronic equivalent, where interconnects are il-

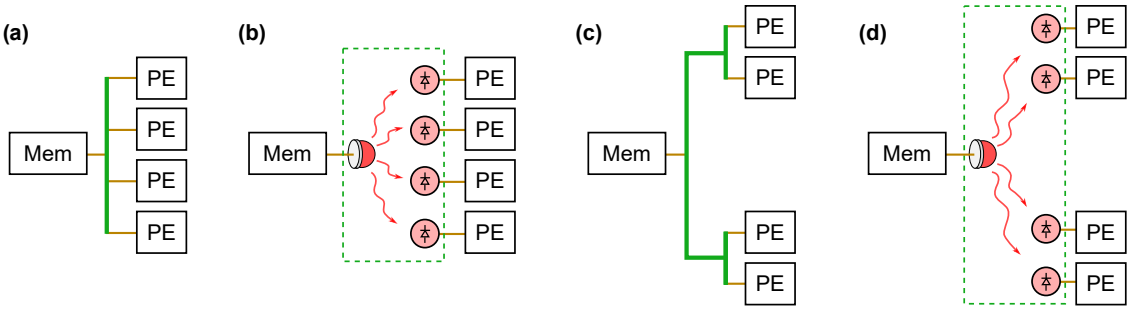

Figure 4.6: Fan-out of one bit from memory (Mem) to multiple processing elements (PEs). **a**, Fan-out by electrical wire to a row of PEs in a monolithic chip. **b**, DONN equivalent of monolithic chip, where green wire is replaced by optical paths. **c**, Fan-out by electrical wire to blocks of PEs divided into chiplets, or separated by memory and logic. **d**, DONN equivalent of fan-out to PEs in multiple blocks (energetically equivalent to **b**).

lustrated in green in Fig. 4.6. We assume a 7 nm CMOS process for both cases (close to the state of the art in 2020, when we performed this analysis). The interconnect energy to transmit one bit, which must include any optical source inefficiencies, is the energy required to charge all parasitic capacitances of the system. To determine the interconnect energy per MAC (E_{comm}, in fJ/MAC), we simply multiply the energy per bit by 16.

4.4.1 Digital electronic data transmission

In the electronic case (E_{elec}), a long wire transports data to a row of multipliers using low-cost repeaters, which we assume consume negligible energy (see Supplementary Note 6 of our paper [30]). The interconnect energy is thus the energy to charge the wire and an inverter, which is representative of the input to a multiplier:

$$E_{\text{elec}}/\text{bit} = \tfrac{1}{4}\left(\tfrac{C_{\text{wire}}}{\mu\text{m}} \cdot L_{\text{wire}} + C_{\text{T}}\right) \cdot V_{DD}^2 \tag{4.2}$$

where V_{DD} is the supply voltage, $C_{\text{wire}}/\mu\text{m}$ is the wire capacitance per micrometer, L_{wire} is the wire length between two multipliers and C_{T} is the inverter capacitance. The wire has a large parasitic capacitance, but also produces an effective electrical fan-out. Interconnects consume energy predominantly when a load capacitance, such as a wire, is charged from a low (0 V) to a high ($\sim$1 V) voltage, i.e., in a $0 \rightarrow 1$ transition. If we assume a low leakage

current, maintaining a value of '1' or '0' (i.e., $1 \rightarrow 1$) consumes little additional energy. To switch a wire from a '1' to a '0', the wire is simply discharged to ground. Assuming a random distribution of '0' and '1' bits, we therefore include a factor of $1/4$ in equation (4.2) to account for this dependence on switching activity.

4.4.2 Digital optical data transmission

In the DONN, a light source replaces the wire for fan-out and detectors are at the inputs to the multipliers. The low-capacitance receiverless detectors in the DONN do not require amplifiers [54]. The energetic requirements of these detectors contrast with those of conventional optical receivers, which aim to maximize sensitivity to the optical input field, rather than minimize the energetic cost of the system as a whole. The DONN's minimum energy consumption corresponds to the optical energy required to generate a voltage swing of 0.8 V on the load capacitance (i.e., the photodetector, C_{det}, and an inverter, C_{T}), all divided by the source's power conversion efficiency (wall-plug efficiency, η_s):

$$E_{\mathrm{DONN}}/\mathrm{bit} = \tfrac{1}{2} \cdot \tfrac{h\nu}{\eta_s} \cdot n_{\mathrm{p}} \tag{4.3}$$

where $h\nu$ is the photon energy which must be greater than or equal to the bandgap E_{g} of the detector material, and the number of photons per bit, n_{p}, is determined by:

$$n_{\mathrm{p}} = \frac{(C_{\mathrm{det}} + C_{\mathrm{T}}) \cdot V_{DD}}{e} \tag{4.4}$$

As in the all-electronic case, we assume low leakage on the receiverless photodetector and a random distribution of bits. Here, however, photons are generated for every '1' and therefore, the switching activity factor is $1/2$ instead of $1/4$.

4.4.3 Comparison

The parameters used to calculate the energy consumption in the electronic and optical cases are summarized in Table 4.2. We chose silicon as an example material, and set the photon energy equal to the bandgap $h\nu/e = E_{\mathrm{g}}$. C_{det} is a theoretical approximation of the capaci-

Table 4.2: Parameters in interconnect energy estimates

$C_{\text{wire}}/\mu\text{m}$	~0.2 fF/μm [54], [101], [102]
C_{T}	~0.1 fF [54], [103]
C_{det}	0.1 fF [54]
$h\nu/e$	1.12 eV
η_s	~0.5 [104], [105]
A_{det}	1 μm × 1 μm [54]
$L_{\text{wire intra-chiplet}}$	5-8 μm†
$L_{\text{wire inter-chiplet}}$	2.5 mm [98]
$L_{\text{wire inter-chip}}$	~5 cm [106]
V_{DD}	0.8 V [107]
E_{MAC}*	25 fJ/MAC [107], [108]

†We assume a square multiplier and scale reported 8-bit multiplier areas in a 45 nm node [109]–[111] to a 7 nm node with scaling factors from Ref. [107]. A MAC unit comprises both an 8-bit multiplier and a 32-bit adder, so we are placing a lower bound on the minimum length of L_{wire}. Recent work [112] optimizes MAC units for DNNs, and reports a 337 μm^2 area in a 28 nm node, where the MAC unit comprises an 8-bit multiplier and a 32-bit adder. Extrapolated to a 7 nm node with a fourth-order polynomial fit of the scaling factors from Ref. [107], the MAC unit is of size $(7\ \mu\text{m})^2$, which falls within the 5-8 μm range. *E_{MAC}, the energy required for one digital multiply-accumulate, shown for reference.

tance of a receiverless cubic photodetector with surface area $A_{\text{det}} = (1 \times 1)\ \mu\text{m}^2$ [54]. Several past examples of small CMOS integrated detectors in older CMOS nodes [88], [113] showcase the feasibility of receiverless detectors. The optical source power conversion efficiency (η_s) is a measured value for VCSELs [104], [105]. C_{T} is an approximation for the capacitance of an inverter [54], [103]. L_{wire} is the distance between MAC units in various scenarios: with abutted MAC units (intra-chiplet), between chiplets (inter-chiplet) and between chips (inter-chip).

Putting these values into equations 4.2 and 4.3, we find interconnect energies shown in Fig. 4.7. The optical communication energy in the DONN is $E_{\text{comm}} \approx 3$ fJ/MAC, independent of length (limited by the photodetector and inverter capacitances). On the other hand, the electrical interconnect energy scales from $E_{\text{comm}} = 3$–4 fJ/MAC for inter-multiplier communication for abutted MAC units, to ~1,000 fJ/MAC for inter-chiplet interconnects, to ~30,000 fJ/MAC for inter-chip interconnects. The crossover point where the optical interconnect energy drops below the electrical energy occurs when $L_{\text{wire}} \geq 5$ μm. In Fig. 4.7, we have also included the optical communication energy per MAC with a large, commercial

photodiode, which illustrates the need for receiverless photodetectors in advanced CMOS processes. In the future, plasmonic photodetectors may lower the capacitance further than 0.1 fF [114].

4.5 Discussion

With minimal impact on accuracy, the DONN can yield an energy advantage over all-electronic accelerators with long wire lengths for digital data transfer. In our proof-of-concept experiment, we performed inference on 500 MNIST images with 2- and 3-layer FC-NNs and found a <0.6% drop in accuracy with respect to the ground truth implementation on CPU.

We attribute these errors to crosstalk from imperfect alignment and the camera's Bayer filter. A simple crosstalk correction scheme lowered measured bit error rates by two orders of magnitude, allowing us to transmit bits with 100% measured fidelity in the input activation arm (better-aligned than the weight arm). Crosstalk can thus be mitigated and possibly eliminated through post-processing. However, to maximize energy efficiency, post-processing should be avoided, for example, by employing charge-sharing among the detector pixels to implement equation (4.1). Alternatively, we could simply reduce crosstalk through improved system design – better lenses and the absence of the de-Bayering algorithm will help significantly. We could also choose to space the PEs further apart or shrink the active region of the detectors to improve the ratio of signal at a given pixel to noise from neighboring pixels.

In the hypothetical regime where error due to crosstalk is negligible, the remaining noise sources are shot and thermal ('kT/C') noise. Intuitively, shot and thermal noise are also present in an all-electronic system, and the number of photoelectrons at the input to an inverter in the DONN (∼1000 photons/bit) is equal to the number of electrons at the input to an inverter in electronics. Therefore, if these noise sources do not significantly affect accuracy in the all-electronic case, they will not affect the DONN either [54]. For mathematical validation, see our paper [30].

In our energy estimates for a near-term optimized DONN, we contrasted the data delivery costs in optics versus a comparable electronic system. We found that in the worst case, when multipliers are abutted in a multiplier array, optical transmitters have a similar interconnect

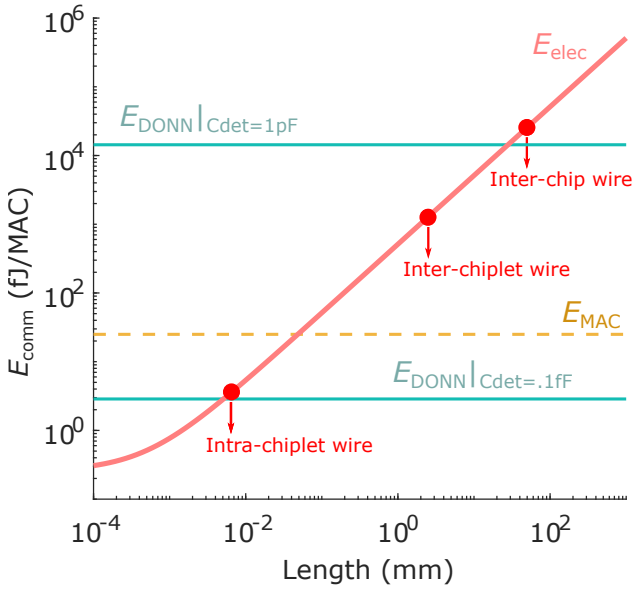

Figure 4.7: DONN analysis: energy required to transmit 16 bits (communication energy per 8-bit MAC, i.e., E_{comm}). Electronic data transfer energy (E_{elec}) increases with wire length, whereas optical data transfer energy (E_{DONN}) remains constant. Optical data transfer evaluated for two detector capacitances: $C_{\text{det}} = 1$ pF for large, commercially-available photodiodes [115]; and $C_{\text{det}} = 0.1$ fF for emerging receiverless, $(1\ \mu\text{m})^3$-sized cubic detectors in modern CMOS processes [54]. Below $C_{\text{det}} = 0.1$ fF, the capacitance of the overall receiver becomes limited by the capacitance of the CMOS inverter. Energy of one digital 8-bit multiply-accumulate operation ($E_{\text{MAC}} = 25$ fJ/MAC) also shown for reference.

energy cost compared to copper wires in a 7 nm node. The regime where the DONN shows important gains over copper interconnects is in architectures with increased spacing between computation units. As problems scale beyond the capabilities of existing single electronic chips, multiple chiplets or chips perform DNN tasks in concert. In the multi-chiplet and multi-chip cases, the costs to transmit two 8-bit values in electronics (∼1,000 fJ/MAC and ∼30,000 fJ/MAC, respectively) are therefore significantly larger than that of an 8-bit MAC (25 fJ/MAC) [107], [108]. On the other hand, in optics, the interconnect cost (∼3 fJ/MAC, including source energy) remains an order of magnitude smaller than the MAC cost. Since multi-chiplet and multi-chip systems offer a promising approach to increasing throughput on large DNN models, optical connectivity can further these scaling efforts by reducing inter-chiplet and inter-chip communication energy by orders of magnitude. In practice, the DONN's scaling will be limited by the amount of optical power that a source can produce for sufficient photons at the receiver array, though this restriction can be circumvented by tiling sources and receiver arrays.

In terms of the DONN's area, we assume the added chip area at the receiver is negligible, since the area of a photodetector $A_{\text{det}} = 1\ \mu\text{m}^2$ is ∼50× smaller than a MAC unit of size $(L_{\text{wire intra-chiplet}})^2$. Overall packaged volume, while likely larger than an all-electronic system, is not as important in many practical applications (e.g., workstations, servers, data centers) because chip area sets fabrication cost. Furthermore, in data centers, space is required between chips for heat sinks and airflow, and the addition of lenses need not increase this volume significantly.

Lastly, optimized optical devices do not restrict the clock speed of the system since their bandwidths can be >10 GHz. In fact, the clock speed of a digital electronic system is generally limited to ∼1 GHz due to thermal dissipation requirements; it could be improved in the DONN, since greater component spacing for thermal management would not increase energy consumption.

Because length-independent data distribution is a tool currently unavailable to digital system designers, relaxing electronic constraints on locality can open new avenues for DNN accelerator architectures. For example, memory can be devised such that numerous pieces of memory are located far away from the point of computation and reused many times spatially,

with a small fixed cost for doing so. Designers can then lay out smaller memory blocks with higher bandwidth, lower energy consumption, and higher yield. If memory and computation are spatially distinct, we have the added benefit of allowing for more compact memories that consume less energy and area, e.g., DRAM, which is fabricated with a different process than typical CMOS to achieve higher density than on-chip memories. Furthermore, due to its massive fan-out potential, the DONN can reduce overhead by minimizing a system's reliance on a memory hierarchy and also amortize the cost of weight delivery to multiple clients running the same neural network inference on different inputs. Additionally, some newer neural network models require irregular connectivity (e.g., graph neural networks, which show state-of-the-art performance on recommender systems, but are restricted in size due to insufficient compute power [116], [117]). These systems have arbitrary connections with potentially long wire lengths between MAC units, representing different edges in the graph. The DONN can implement these links without incurring additional costs in energy from a complex network-on-chip in electronics. Yet another instance of greater distance between multipliers is in higher-bit-precision applications, as in training, which require larger MAC units.

4.6 Conclusion

In summary, the DONN implements transmission and fan-out of data with an energy cost per MAC that is independent of length and number of receivers. This property is key to scaling deep neural network accelerators, where increasing the number of processing elements for greater throughput in all-electronic hardware typically implies higher data communication costs due to longer electronic path length. The DONN does not require digital-to-analog conversion and is therefore less prone to error propagation. It is also reconfigurable, in that the weights and input activations can be easily updated, and the fan-out locations could be updated to suit a given DNN with an adjustable DOE. We find that optical data transfer begins to save energy when the spacing of MAC units is >5 μm. More broadly, further gains can be expected through the relaxation of electronic system architecture constraints.

Chapter 5

Single-Shot Optical Neural Network

This chapter is adapted from work reported in Ref. [118].

The output-stationary architecture of the DONN described in the previous chapter works well for inference performed on large batches of inputs; however, the input vectors are streamed in sequentially over time, resulting in a latency that scales with the input vector length. The DONN is therefore not optimized for inference tasks that have very stringent requirements on latency, e.g., translation and autonomous driving, or newer applications in astrophysics [119] and the control of fusion reactors [120]. A weight-stationary architecture, on the other hand, can produce an output vector in one shot – but only if it can simultaneously multicast the input vector to all required multiplier units, if there is sufficient PE connectivity for single-step accumulation and readout, and if there are enough PEs to store the entire weight matrix. Additionally, while the DONN can achieve performance gains in the transmission of data over large distances, it does not exploit the ability of optics to take on some of the computational burden of DNNs. With these considerations in mind, we designed the 'single-shot optical neural network', where optoelectronic components perform reconfigurable data fan-out *and* element-wise analog weighting.

As discussed in Chapter 2, weight-stationary accelerators have previously been developed, but with various limitations. On the electronics side, in systems like the TPU [46], constraints on wiring require that multipliers pass partial products from one multiplier to the next at each clock cycle, which prohibits single-shot MVM (though pipelining allows for high

throughput). In optics, 2D integrated schemes are restricted in scale, and while 3D systems can theoretically achieve higher scaling in terms of number of PEs, high-accuracy proof-of-principle demonstrations are lacking. Past works have, for example, required fully digital electronic back-end layers and completely new training paradigms.

Our work demonstrates a fully programmable 3D ONN capable of single-shot-per-layer inference at large scale. A source array encodes an input vector into analog optical intensities. Free-space optical components (LCoS SLM and lenses) copy and distribute the input light to optoelectronic weighting elements. Using standard DNN models and without retraining, we show low loss of classification accuracy on the MNIST, Fashion-MNIST and QuickDraw datasets in a proof-of-concept demonstration with input vector length $K = 784$. With up to 49× optical fan-out, our system performs up to 38,416 multiplications per time step. Dynamic reconfigurability of the inputs, fan-out and weighting elements in our experiment allow for potential model updates. We also experimentally determine the physical upper bound to throughput of our system (∼0.9 exaMAC/s) by measuring classification accuracy versus input source spectral width. Lastly, we estimate that a near-term optimized single-shot ONN can outperform digital electronic accelerators in latency and energy consumption by processing a complete million-element DNN layer in ∼10 ns with ∼10 fJ/MAC, while maintaining ∼petaMAC/s throughput.

5.1 Concept

Figure 5.1 illustrates the single-shot ONN's architecture at a high level. As explained in Chapter 2, an FC-NN layer's output vector ($\mathbf{y}$) is the product of a matrix of pre-learned weights ($\mathbf{W}$) with the input vector ($\mathbf{x}$), then cascaded into a nonlinear activation function. In Fig. 5.1b-c, the input vector $\mathbf{x}$ and the rows of the weight matrix $\mathbf{W}$ are both recast into blocks, and then $\mathbf{x}$ is projected onto each row of $\mathbf{W}$. (In the case of a 2D input image, $\mathbf{x}$ is already in block form.) With the inputs then local to their corresponding weights, all required element-wise multiplications are completed simultaneously. After block-wise summation and the nonlinearity, the computation of $\mathbf{y}$ is complete.

Free-space optics can passively realize this data routing, replication and weighting (Fig. 5.1d).

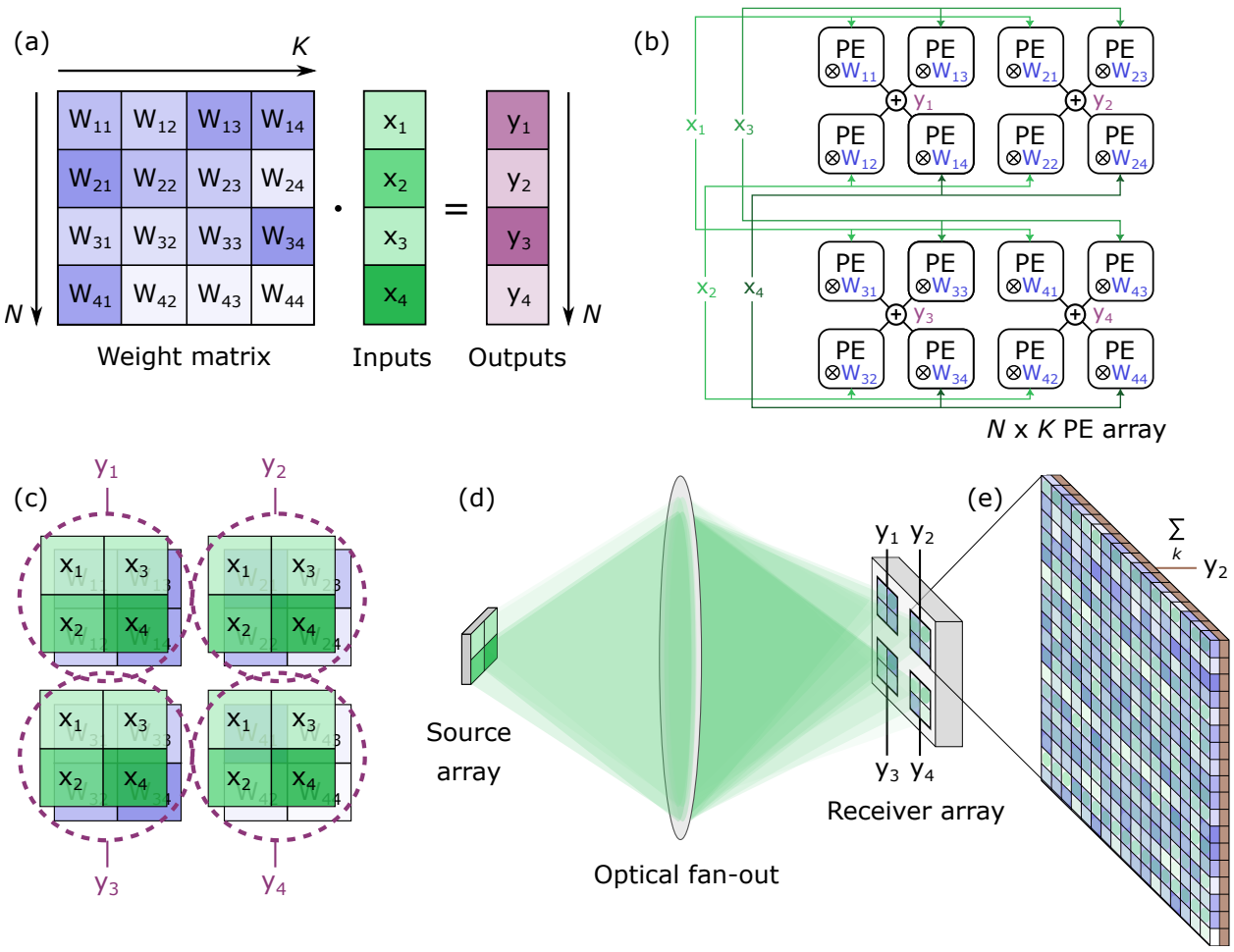

Figure 5.1: Analog, single-shot computation of a fully connected neural network layer. **a**, One layer of an FC-NN: weight matrix $\mathbf{W}_{N\times K}$ multiplies input activation vector $\mathbf{x}_{K\times 1}$ followed by a nonlinearity (e.g., ReLU) to produce $\mathbf{y}_{N\times 1}$. **b**, Architectural depiction of the single-shot ONN: each PE stores a weight value, the number of PEs is equal to the weight matrix size, and every input activation is simultaneously multicast to N PEs. **c**, Alternative visualization showing that $\mathbf{x}$ is block-encoded and fanned out over the rows of $\mathbf{W}$. With block-wise summation, $\mathbf{W}\cdot\mathbf{x}$ is computed in one time step. **d**, Optical implementation: K-element source array encodes $\mathbf{x}$ into analog optical intensities and is replicated and imaged onto N receiver blocks, with electronics for summation and the nonlinearity. Additional sources broadcast outputs to the next layer, e.g., a duplicate of the same hardware. **e**, Free-space optics enable high-density, 3D information transfer, with K $\sim 10^3$ inputs incident on up to $\sim 10^3$ weighting elements per block above electrically connected photodetectors.

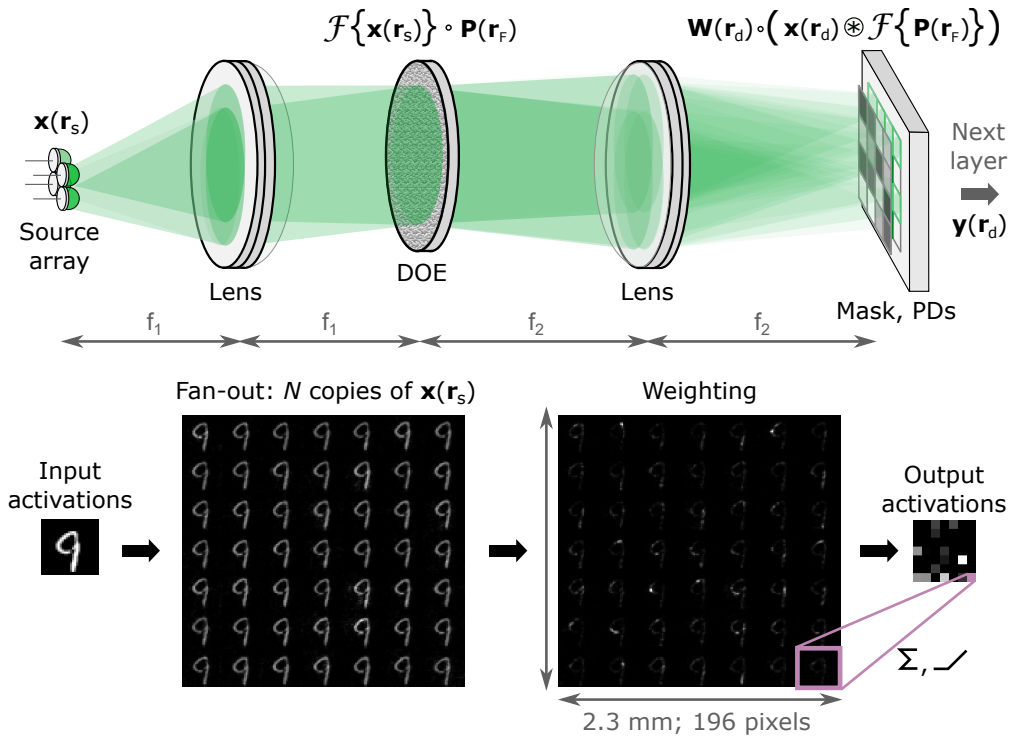

Figure 5.2: Single-shot optical neural network. Source array (wavelength λ, object plane) encodes inputs $\mathbf{x}(\mathbf{r}_s)$ into analog optical intensities at transverse spatial positions $\mathbf{r}_s$. A diffractive optical element (DOE, Fourier plane) performs element-wise multiplication of the spatial Fourier transform of $\mathbf{x}(\mathbf{r}_s)$ with fan-out phase pattern $\mathbf{P}(\mathbf{r}_F)$, where $\mathbf{r}_F = 2\pi \cdot \mathbf{r}_s/(\lambda \cdot f_1)$. Optoelectronic weighting elements (image plane) perform element-wise products between the weight matrix and replicated (multicast) input activations: $\mathbf{W}(\mathbf{r}_d) \circ (\mathbf{x}(\mathbf{r}_d) \circledast \tilde{\mathbf{P}}(\mathbf{r}_d))$, where $\mathbf{r}_d = -(f_1/f_2)\,\mathbf{r}_s$ and $\tilde{\mathbf{P}}(\mathbf{r}_d) = \mathscr{F}\{\mathbf{P}(\mathbf{r}_F)\}$ is the spot array. Electronics sum K photodetector outputs per block by Kirchhoff's current law. Experimental fan-out and weighting data shown.

Figure 5.2 illustrates our approach in more detail, showing how we can combine the concepts described in Chapter 3 to achieve single-shot MVM. A source array encodes the input activations into the single-spatial-mode analog amplitudes of light pulses $\mathbf{x}(\mathbf{r_s})$, where $\mathbf{r_s}$ is the transverse spatial position in the 2D object plane. In the Fourier plane, a diffractive optical element (DOE) such as an LCoS SLM displays a spot array generation phase pattern for reconfigurable fan-out. Following Fourier convolution theory, the spot array in the image plane generated by the DOE is optically convolved with the image of the input pattern, which yields N copies of the input pattern (i.e., 2D multicast – see Chapter 3). Reconfigurable weighting elements, e.g., LCoS SLM pixels plus a polarizer, then attenuate the intensity of each replicated input pixel proportionally to each weight value in $\mathbf{W}$. µm-scale PDs [87], [90] directly below the weighting elements can convert the signal to analog electronics for block-wise summation by Kirchhoff's current law. Another option for summation is 'optical fan-in' [61], where a single large PD can replace the block of small, electrically connected PDs. An amplifier per block reads out the accumulated charge and an electronic post-processing unit performs the nonlinearity to yield $\mathbf{y}$. An output source array (e.g., same components as the input sources) with one source per block can then be the input to the next layer, e.g., a replica of the hardware used for the previous layer.

5.2 Experimental setup

We verified the impact of single-shot analog optical data encoding, fan-out and weighting on DNN classification accuracy. Figure 5.3 shows our experimental implementation of Fig. 5.2. Illuminated by a continuous-wave (CW) laser, three LCoS SLMs display a complete activation image (SLM #1), fan-out pattern (SLM #2) and full weight matrix (SLM #3) for up to $28 \times 28 \times 49 = 38,416$ multiplications per frame. (The pattern on SLM #2 can be updated to match a layer's shape.) Additional polarizers and the iris reduce stray light.

The first SLM (Meadowlark, AVR Optics P1920-400-800-HDMI-T, pixel width 9.2 µm) displays an input image $\mathbf{x}$. The incident light is polarized to 45° after rotation by the half-wave plates. A polarizing beamsplitter (PBS) rejects the unrotated polarization from the SLM, i.e., we are using it in 'amplitude mode'. We use every second pixel for the activation

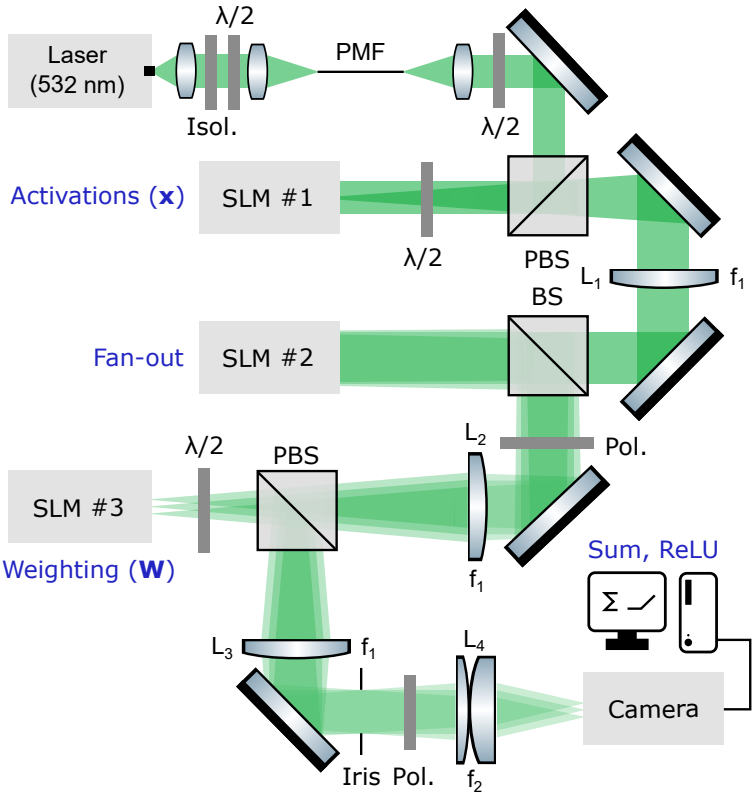

Figure 5.3: Proof-of-concept implementation of single-shot optical neural network. Collimated laser light is incident on a spatial light modulator (SLM #1, object plane) with 45° polarization after half-wave plate (λ/2). SLM #1 with polarizing beamsplitter (PBS) encodes $\mathbf{x}$ by pixel-wise intensity modulation. SLM #2 (Fourier plane) imparts fan-out phase pattern. Achromatic lenses of focal lengths $f_1 = 250$ mm and $f_2 = 145$ mm image $\mathbf{x}$ from SLM #1 to SLM #3 for weighting ($\mathbf{W}$) and from SLM #3 to camera. Digital computer controls the hardware, sums each block and implements nonlinearity.

display, turning every other pixel 'off' such that the input pattern is effectively multiplied by a grating. We then block the zero order in the Fourier plane to reduce background, such as reflection from the backplane. In subsequent layers with fewer input activations (shorter $\mathbf{x}$ vector), we use every eighth pixel of the SLM to further reduce crosstalk.

The incident light on the second SLM (Hamamatsu X10468-04, pixel width 20 μm) is horizontally polarized (along the extraordinary axis); this SLM is in the Fourier plane and adds a variable phase delay to each pixel to impart a fan-out phase pattern onto the signal. We used WGS (without camera feedback) to determine the phase patterns [85], as described in Section 3.2.3. Each pattern only needed to be calculated once per network size, as it does not depend on the weight or input activation values. The third SLM (also Meadowlark, AVR Optics P1920-400-800-HDMI-T), in the image plane, is used in 'amplitude mode', similarly to the first SLM for element-wise multiplication of the replicated inputs by the weights. Telescopes of achromatic lenses transmit the replicated input activations to the image planes for 1:1 mapping from SLM #1 to SLM #3 to the camera (Thorlabs DCC3240M, pixel width 5.3 μm). The lens positions along the optical axis are fine-tuned with linear stages.

As a stand-in for an analog electronic circuit, a digital electronic computer performs the per-block summation and ReLU. The electronic computer also takes the output pixel values that should be negatively weighted and multiplies them by -1, as the SLM can only apply the absolute values of the weights. In practice, this negative weighting can be implemented with an analog switch or two PDs per receiver pixel (see section 5.9.3). The layer outputs, which are always positive because of the ReLU nonlinearity, are then fed back to SLM #1 as inputs to the next layer, and the weights are updated for the next layer of computation. All neural network layers are thus implemented on our ONN, with the fan-out pattern on SLM #2 held constant for a full classification experiment, the weights on SLM #3 updated between layers, and the inputs on SLM #1 refreshed at every time step with each new input image. Instrument control and image processing were performed in MATLAB.

Lens selection

SLM #1 must be perfectly pixel matched to SLM #3 as well as the camera to minimize data transmission error and to reduce the number of calibration steps. (The different pixel

size of SLM #2 is taken into account in the calculation of the phase pattern by WGS.) We needed to select lenses and position them appropriately. Because SLMs #1 and #3 have the same pixel dimensions, lenses L_1 and L_2 should have the same focal length f_1 (see Fig. 5.3). With a camera pixel width of 5.3 μm and SLM #3 pixel width of 9.2 μm, the ratio of focal lengths f_2/f_1 of lenses L_4 and L_3 should equal 5.3/9.2.

In absolute values, the focal lengths should be long enough to accommodate all the required components, and longer focal lengths cause fewer aberrations and are easier to align because the lens curvature is reduced. Furthermore, the focal length of L_1 should yield a collimated beam from one pixel in SLM #1 that covers most of the area of SLM #2 without too much clipping. On the other hand, shorter focal lengths are desirable for system compactness. With $f_1 = 250$ mm, chosen as a compromise length, we need $f_2 = 144$ mm.

Because stock lenses do not come in this exact focal length, we combined two lenses for L_4 and adjusted the positions of lenses L_1 and L_3 with z stages to achieve the desired pixel matching. An added advantage of combining two lenses (e.g., two achromatic doubles) is that they tend to produce lower optical aberrations than a single lens. With lenses of focal lengths 180 mm and 750 mm, the combined focal length, assuming ideal lenses, is $1/(1/180\text{ mm} + 1/750\text{ mm}) = 145$ mm.

While compensating for incorrect focal length by tuning lens positions is not ideal because it can result in field curvature and increase aberrations, the offset is small enough here to be acceptable, as we will verify later. We can use simple ABCD matrices [121] to find the correct positions of the lenses, or ray-tracing software like Zemax, which also allows us to verify the sensitivity to alignment, the ordering of lenses within L_4 and that rays will be contained within a superpixel, despite aberrations, across the whole field of view. For this experiment, with simple cemented achromatic doublets (Thorlabs ACT508-250-A, AC508-180-A, and ACT508-750-A), all the simulated rays in Zemax were within the Airy disk, which was smaller than the size of a superpixel (2×2 camera pixels).

5.3 System calibration

5.3.1 SLM

The LCoS SLM calibrations from the manufacturers are set to map the input values of 0 to 255 linearly to a 0 to 2π phase shift for normally incident 532 nm light. Since we use the SLMs from Meadowlark (i.e., SLMs #1 and #3) to modulate amplitude and not phase, we needed to recalibrate (Fig. 5.4). Our SLM model has 2048 voltage settings that produce $>2\pi$ phase modulation. In the calibration step, we displayed a uniform array for each voltage value and averaged the output intensities received on the camera. We then fit the inputs versus averaged outputs with a 9th-order polynomial. We used this fit to replace the manufacturer lookup table for the SLM. The input activation values and weights are then restricted to $\sim$7 bits of precision since the values of interest are confined to a small region of voltages (depicted by green arrows in Fig. 5.4**a**).

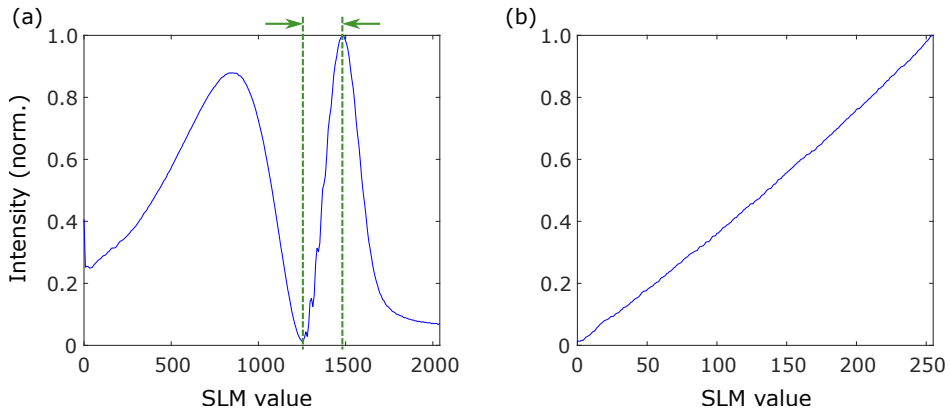

Figure 5.4: SLM #1 calibration. SLM #1 displays values of 0 to 255 with SLMs #2 and #3 set to maximum transmission. These plots show averaged camera outputs **a**, before calibration with fitted region indicated by green dashed lines containing the minimum and maximum outputs and **b**, after calibration. The SLM values can only be programmed to integers from 0 to 255, but there are 2048 voltage values available through the custom lookup table during calibration.

Because a wide region of SLM #3 is illuminated, local non-uniformities cause different blocks (subimages) to require slightly different calibrations to achieve a linear weight display (Fig. 5.5). Therefore, we calculated a refined fit per subimage for SLM #3, where we once again displayed a uniform array at each frame, but here, stepped through the display values

from the global lookup table that we determined in the previous step. We then fitted 8th-order polynomials to the displayed values versus averaged outputs per subimage and adjusted the displayed weight values accordingly in the experiments (Figs. 5.6 and 5.7).

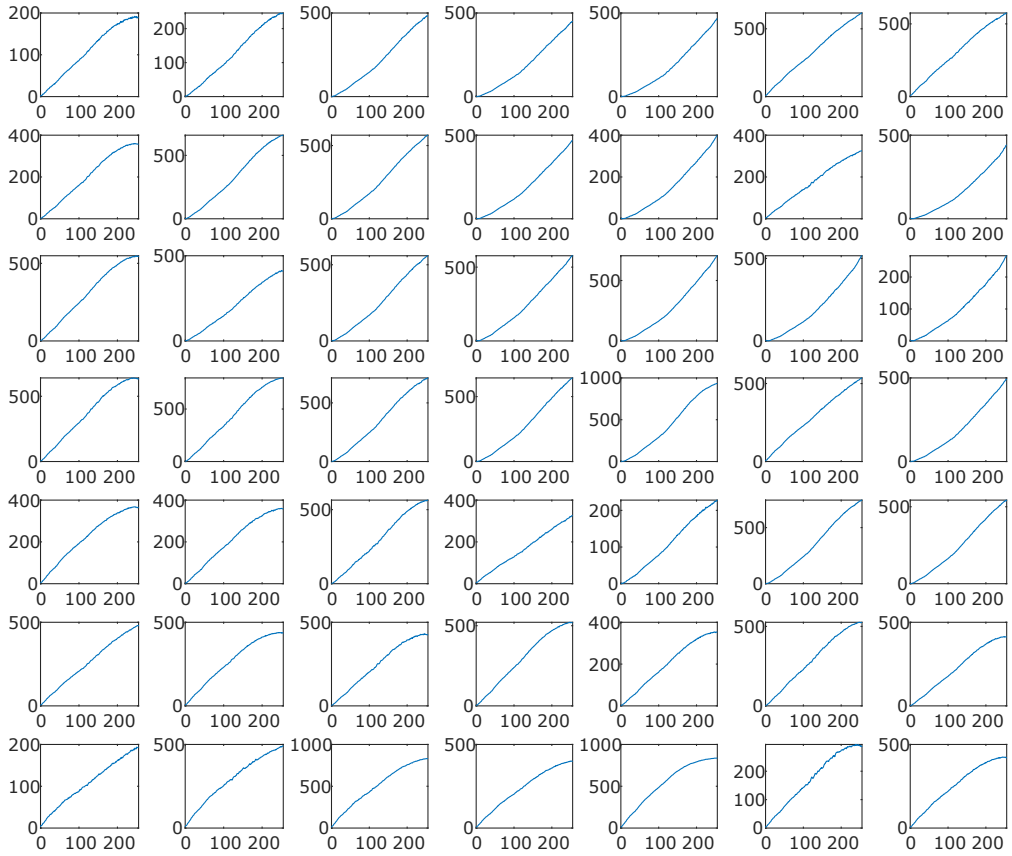

Figure 5.5: After initial calibration of lookup table, outputs averaged per subimage versus displayed value on SLM #3, with 7×7 fan-out. Ideal behavior would be a linear relationship in each subimage.

5.3.2 Image processing

We performed simple processing of the output images from the camera. To reduce the impact of stray light (e.g., room lights) on the system, we acquired a background with SLMs #1 and #3 set to all zeros, and subtracted this background from every output frame. We also performed 2×2 pixel binning. Furthermore, the fan-out pattern displayed on SLM #2 does not yield subimages of equal intensities on the camera. In layer 1, we compensated local non-uniformities by normalizing by a smoothed and background-subtracted calibration map, which we acquired by setting all activations and weights to a constant value. We also divided

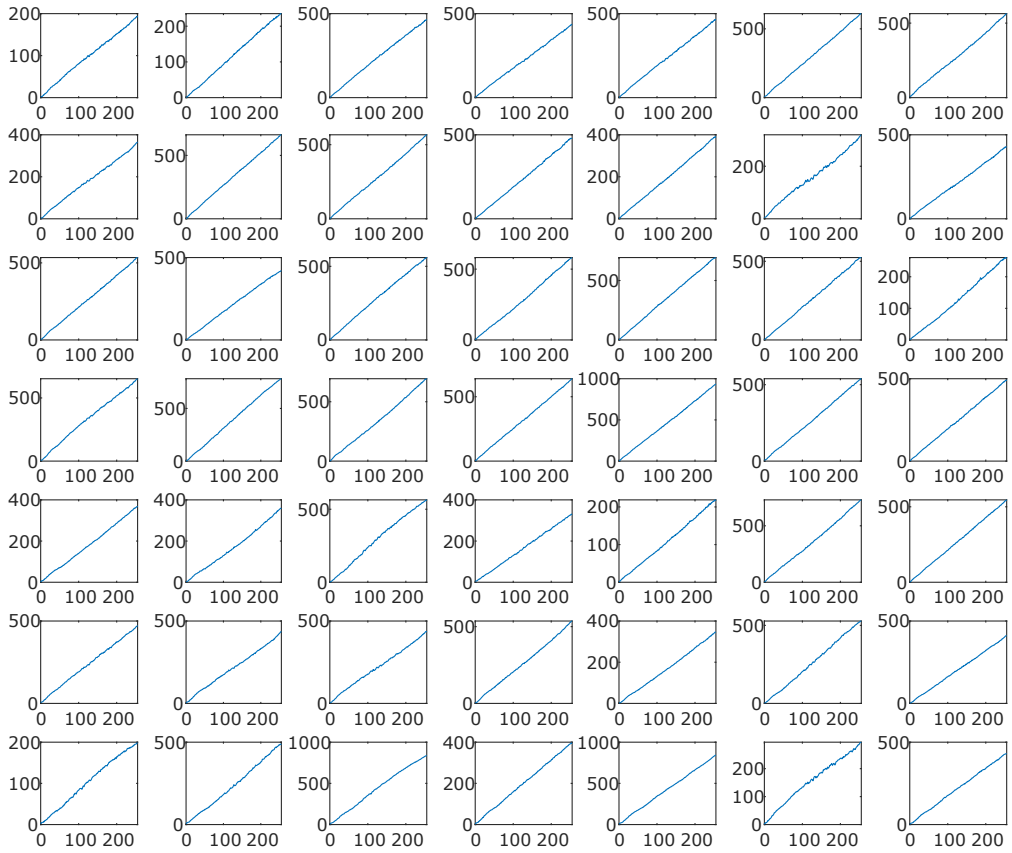

Figure 5.6: Same as Fig. 5.5, but where inputs to each subimage were adjusted with refined 8th-order polynomial fit from values acquired in Fig. 5.5.

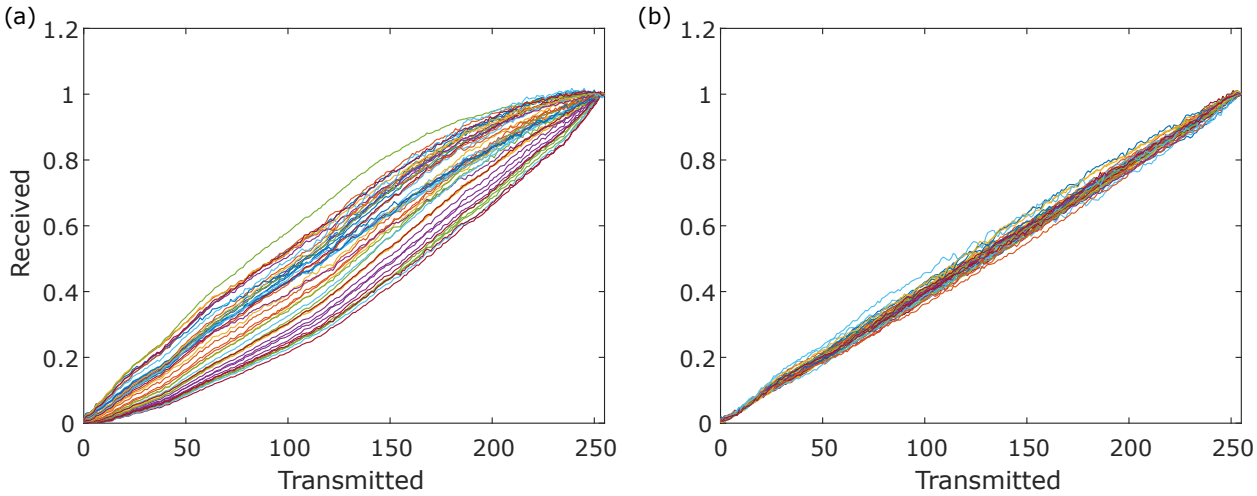

Figure 5.7: All curves from (**a**) Fig. 5.5, and (**b**) Fig. 5.6 collapsed onto the same set of axes, normalized per subimage.

the outputs by the mean intensity per subimage of 100 randomly selected images from the validation set. In subsequent layers, we only divided the summed subimage outputs by the averaged intensity of a calibration map acquired with the inputs and weights set to 255. We had a reduced effective number of bits for the dimmer subimages since we did not make use of the full dynamic range of the camera. Therefore, we fanned out the inputs to an extra row to replace the dimmest subimages and the subimage that overlaps with the zero order.

These processing steps can ultimately be eliminated in an optimized design. The background subtraction can be implemented with a bias voltage (or the computation can be performed in a dark room). The spot uniformity could be improved by adjusting the pattern on SLM #2 with a feedback algorithm [79], [85], which would eliminate the need for normalization. Lastly, we can improve the diffraction efficiency to reduce the optical power in the zero order – then, the extra fan-out images would not be required.

5.4 Deep neural networks

5.4.1 Datasets and DNN layer shapes

We experimentally verified the impact of single-shot analog optical data encoding, fan-out and weighting on the classification accuracy of the MNIST, Fashion-MNIST and QuickDraw datasets. We used one- and two-hidden-layer fully connected neural networks (FC-NNs) with $784 \rightarrow N(\rightarrow N) \rightarrow 10$ activations, where $N \in \{25, 36, 49, 98\}$. (The $N = 98$ layers were computed in two tiles, i.e., time steps.) For QuickDraw, a NumPy random number generator selected 10 out of the possible 345 classes for our experiment, with 10,000 images per class for the training set and 1,000 images per class for the validation and test sets. The ReLU nonlinearity was implemented electronically between layers.

5.4.2 DNN training

We trained the FC-NNs on a standard digital electronic computer using only FC layers and standard layers commonly used to reduce overfitting, namely Gaussian noise and dropout. We used the PyTorch library in Python to train on 50,000 training images for the MNIST and

Fashion-MNIST datasets, and 100,000 images for QuickDraw. 10,000 different images were reserved for validation sets to fine-tune the network hyperparameters and optical setup. Each dataset was normalized by its standard deviation, and L2 regularization with L2 = 0.0001 and 10% dropout were applied to each layer. Gaussian noise was also added to each activation value at every layer for 'noise-aware training'. The standard deviation of the noise was set to $0.25 \cdot \delta$ (except for the 2-hidden-layer, 98-activation-per-layer network, where our inference noise model indicated that $0.35 \cdot \delta$ would better maintain accuracy), where δ is the standard deviation of an activation value across a batch. The activation function on the final layer was softmax. The Adam optimizer minimized the categorical cross-entropy loss function for up to 200 epochs with a batch size of 100 and learning rate of 0.001. The exact number of epochs was selected as the number that gave the highest validation accuracy during training.

5.4.3 Weight fine-tuning

We used the weights trained as described in the previous section to perform inference on our optical system, without modification and without changing our optical setup beyond the calibration detailed above. This direct use of the weights (which we call the 'basic' configuration) came at the cost of a small decrease in accuracy – see below. We also investigated iterative weight fine-tuning (similar to Ref. [66]) to help bridge this gap between the accuracy of the optical hardware and digital electronics. To do so, we loaded the pre-trained weights of the first layer onto SLM #3 and transmitted the training and validation sets through the optical setup. We then fine-tuned the weights of the subsequent layers using the pre-trained weights as a starting point, by training for a few additional epochs (up to 10, where the exact number was determined using the validation set). In the three-layer case, we then loaded the updated weights of the second layer onto SLM #3 and performed the second layer of inference, before finally repeating the fine-tuning process for the last layer.

5.5 System characterization

The optical system is subject to several sources of noise that decrease its data transmission and computing accuracy, including:

- Crosstalk between neighboring pixels on the weighting SLM and camera from imperfect imaging optics and fan-out;
- Stray light on the detector from reflections from the optical components (lenses, wave plates, beamsplitters and backplanes of the SLMs);
- Limited SLM precision ($\sim$7 bits) since we use a constrained set of phases for amplitude modulation;
- Limited camera precision in certain dimmer subimages ($\sim$6 bits) due to the combination of the camera's 8-bit depth and overall dynamic range, i.e., requirement to not saturate the camera for brighter subimages;
- Camera noise, i.e., read and thermal noise;
- SLM flicker noise;
- Laser noise, e.g., shot noise.

To evaluate the impact of these noise sources, we characterized the overall accuracy of received versus transmitted data in our system. Figure 5.8 shows histograms of measured intensities I versus transmitted inputs x, weights W and element-wise products $W \cdot x$ for the MNIST classification task using the FC-NN with $784 \rightarrow 49 \rightarrow 10$ activations (without weight fine-tuning). To evaluate $I(x)$, we displayed a different test image at each time step for 100 steps on SLM #1, fanned out 49$\times$ with SLM #2, set all SLM #3 pixels to a constant, uniform value, and recorded the outputs. Similarly, to measure $I(W)$, we displayed the layer weights on SLM #3 and set all SLM #1 input pixels to a constant, uniform value. For $I(W \cdot x)$, SLM #1 encoded 100 MNIST test images while SLM #3 displayed the full weight matrix. The distribution of outputs $I(W \cdot x)$ versus ground-truth values $W \cdot x$ for layer 1 broadens further from zero, but most values $W \cdot x$ are small, with 97% of the total counts in the range $W \cdot x \in [-0.15, 0.15]$ (center three columns). The distributions for layer 2 are narrower than those for layer 1 due to increased pixel spacing made possible by fewer input activations, which lowers crosstalk.

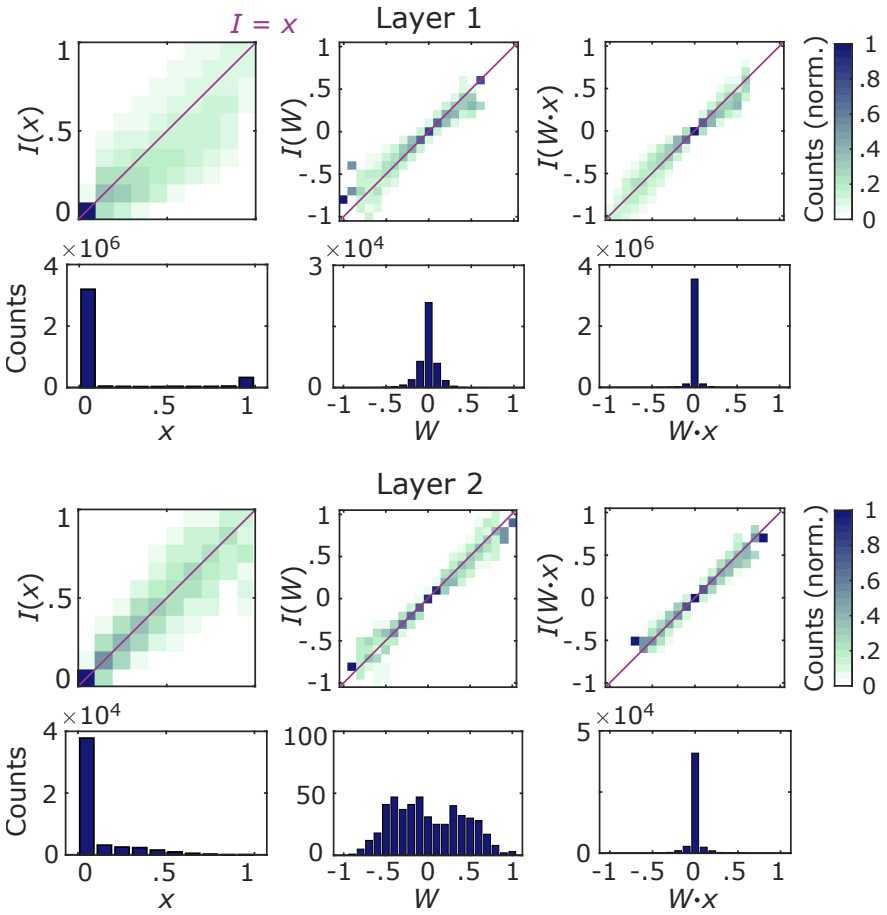

Figure 5.8: With 49× fan-out, histograms of received intensities I (normalized) versus ground-truth values over all pixels for activations x, weights W and element-wise products $W \cdot x$ from 100 random MNIST test images with an FC-NN with $784 \rightarrow 49 \rightarrow 10$ activations. Each column normalized by the sum of the column (with sums shown in 1D histograms). Full weight matrix displayed and held constant on SLM #3. Fan-out phase pattern on SLM #2 also constant for the duration of the experiment.

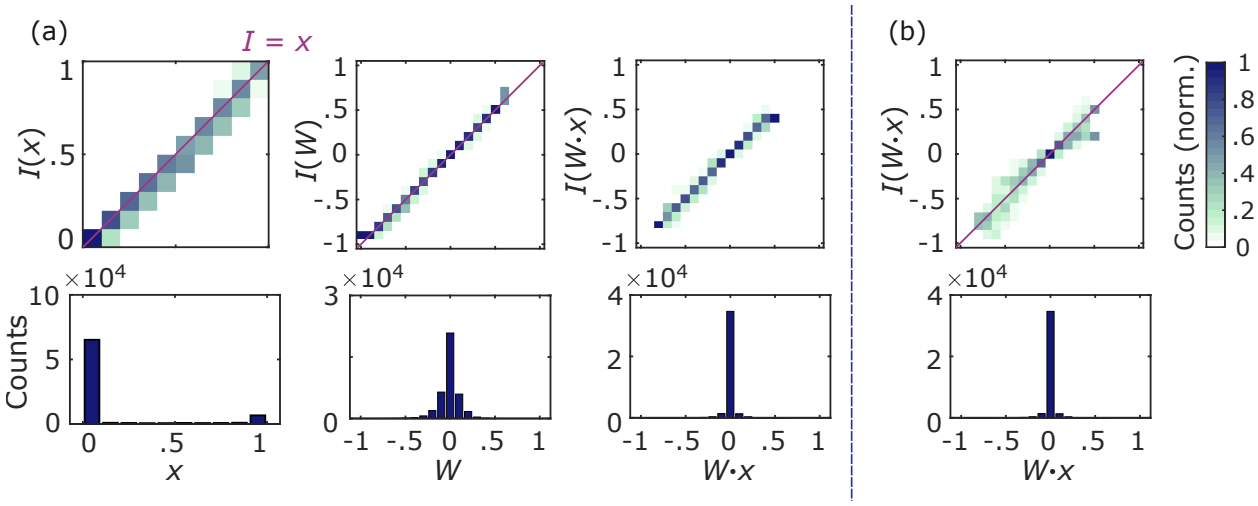

Figure 5.9: **a**, Without fan-out (one data point displayed at a time), histograms of received intensities I (normalized) versus ground-truth values for activations x in 100 random MNIST test images, weights W from the first layer of an FC-NN with $784 \rightarrow 49 \rightarrow 10$ activations and corresponding element-wise products $W \cdot x$ for a random MNIST test image. Data in $I(W \cdot x)$ renormalized twice during long acquisition to account for potential long-term laser intensity fluctuations. Each column normalized by the sum of the column (with sums shown in 1D histograms). This single-pixel configuration was used only for characterization. **b**, For comparison, $I(W \cdot x)$ of same MNIST image with 49× fan-out (one-shot weighting).

We also characterized transmission and weighting of one data point at a time (no fan-out) through the full setup to determine the accuracy without crosstalk and with reduced stray light in the system (fewer illuminated pixels). For this measurement, we used a single pixel on SLM #1, set SLM #2 to 'all on', and set SLM #3 to a single weight value per time step. Figure 5.9 shows histograms of received intensities in one camera superpixel versus transmitted values. Errors here are due to laser and camera noise, limited precision of the optoelectronic components and SLM flickering. The overall errors are greatly reduced with respect to Fig. 5.8, suggesting that crosstalk and stray light are the most important sources of error in the system. The relative impact on DNN classification accuracy will be discussed in Section 5.6.1 below.

5.6 DNN accuracy

We performed inference using the datasets and training methods described in Section 5.4. We calculated the ground-truth accuracies on a digital electronic computer with full precision,

multiplying the pre-trained weight matrices with the input activation vectors of each layer. The ReLU nonlinearity was applied at each layer's output (except the final layer). No noise was added to the ground-truth inference computation.

5.6.1 Simulation

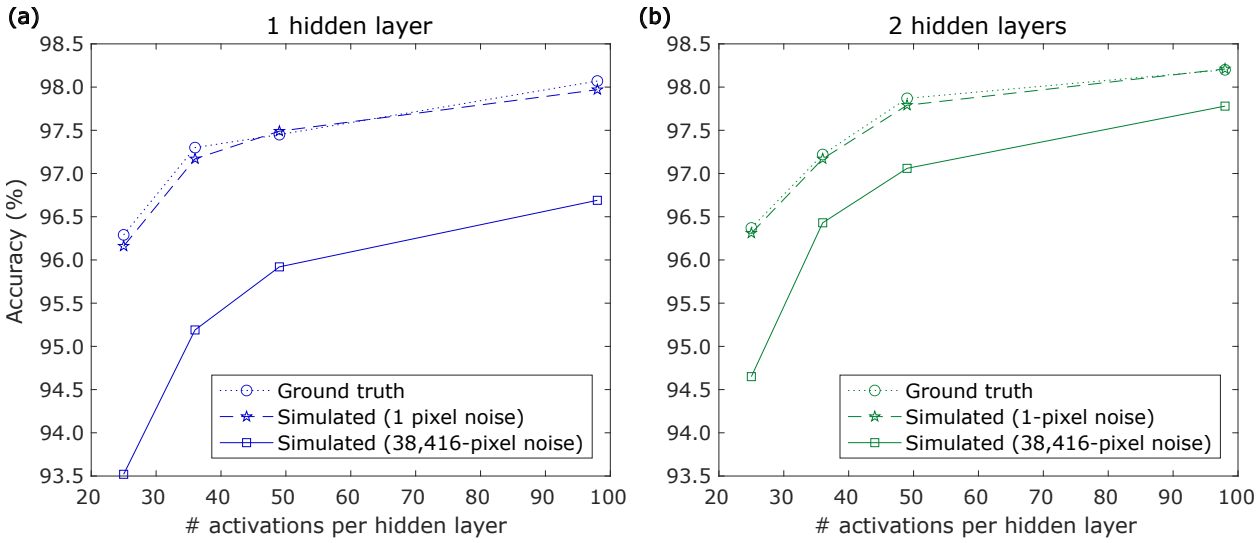

Figure 5.10: Simulated classification accuracies of 10,000 previously unseen MNIST test images with added noise in networks of shape **a**, $784 \to N \to 10$ and **b**, $784 \to N \to N \to 10$, where N is the number of activations per hidden layer (on the horizontal axis). Gaussian noise replicating our characterization data with full fan-out (38,416-pixel noise) and without fan-out (1 pixel noise) was included in each element-wise product in inference on a digital electronic computer.

Before examining our experimental results, here we describe simulations on a standard digital electronic computer of MNIST inference accuracy in the presence of noise. Each element-wise product in the inference calculation was multiplied by and summed with random Gaussian noise. The Gaussian distributions were centered at zero, with standard deviations determined from the characterization data in the previous section. We used the acquired element-wise products $I(W \cdot x)$, which we first normalized to minimize the error with respect to the ground truth element-wise products $W \cdot x$. (A scaling factor does not affect the classification accuracy in our setup since the inference calculation is simply a series of matrix products and ReLU nonlinearities.) Then, we plotted histograms of the element-wise error, $W \cdot x - I(W \cdot x)$, across different bins of ground-truth products (can be thought of as 1D

cross-sections of Fig. 5.9**a**-**b**). Fitting a Gaussian to each histogram, we could estimate the standard deviation of the error as a function of the ground-truth element-wise product value. We could then generate noise with the same statistical properties in our digital electronic model.

In Fig. 5.10, we report the simulated MNIST classification accuracies of FC-NNs of varying shapes and with different noise characteristics. First, we used the noise statistics from the complete single-shot optical system, with 38,416 multiplications per time step. For comparison, we also simulated the classification accuracy with the statistics from the single-pixel-per-time-step transmission (greatly reduced stray light and crosstalk). In this latter case, we found near-equivalent accuracy to the digital electronic ground truth.

5.6.2 Experiment

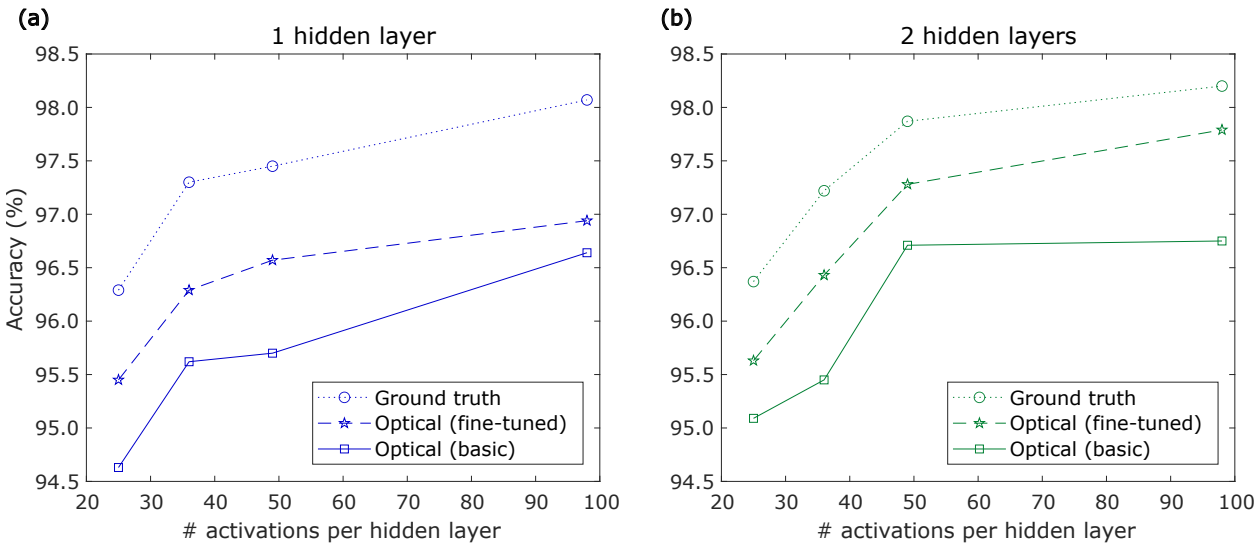

Figure 5.11: Experimentally obtained classification accuracies of 10,000 previously unseen MNIST test images with networks of shape **a**, $784 \to N \to 10$ and **b**, $784 \to N \to N \to 10$, where N is the number of activations per hidden layer (horizontal axis). Inference on optical setup using unaltered pre-trained weights (basic), fine-tuned weights based on hardware outputs (fine-tuned), and all-electronic inference (ground-truth).

Figure 5.11 reports our optical system's classification accuracies of the MNIST hand-written digit dataset with FC-NNs of varied shapes and depths. All MNIST classifications were within 1.8% of the ground truth accuracy. Weight fine-tuning reduced this error to 0.4–1.1%. Table 5.1 shows a subset of the data from Fig. 5.11, as well as our classification

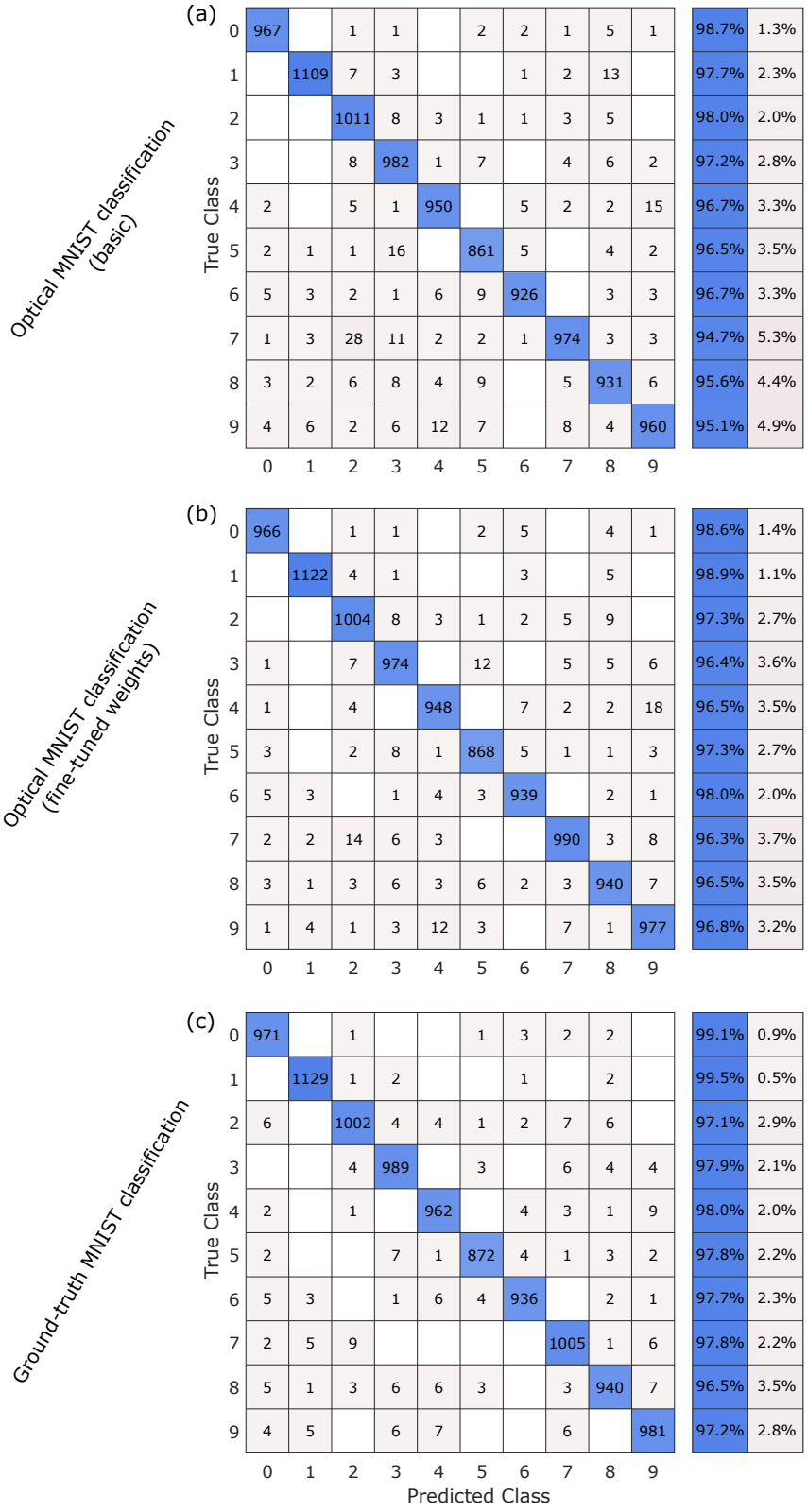

Figure 5.12: Example confusion matrices for classification of 10,000 previously unseen MNIST test images with an FC-NN with $784 \rightarrow 49 \rightarrow 49 \rightarrow 10$ activations using: **a**, our optical system (96.7% accuracy), **b**, our optical system with fine-tuned weights (97.3% accuracy) and **c**, a standard digital electronic computer (97.9% accuracy).

Table 5.1: Selected single-shot optical neural network accuracies versus ground truth

Dataset	# hidden layers	# acts. per hidden layer	ONN accuracy		Ground-truth accuracy
			Basic	Fine-tuned	
MNIST	1	36	95.6	96.3	97.3
MNIST	2	49	96.7	97.3	97.9
MNIST	2	98[1]	96.8	97.8	98.2
Fashion-MNIST	2	36	83.3	85.7	87.1
QuickDraw[2]	2	36	79.0	80.5	82.6

[1] Each layer computed in two shots (tiles).
[2] 10 classes selected at random by NumPy's random number generator.

results of the more challenging Fashion-MNIST and QuickDraw datasets with a moderately sized network. With our largest single-shot-per-layer network, a two-hidden-layer FC-NN with $784 \rightarrow 49 \rightarrow 49 \rightarrow 10$ activations, we obtained a 96.7% inference accuracy on 10,000 previously unseen MNIST test images without fine-tuning or retraining, compared with the ground-truth (all-electronic) accuracy of 97.9%. With fine-tuning, we were able to boost our classification accuracy to 97.3% for the same DNN shape. Figure 5.12 shows the associated confusion matrices for this network, where our optical hardware's classification error was close to the ground truth for all digits (within 2.1% except for the digit '7', which was often mistaken for a '2' or '3'). To show the potential performance of the hardware with larger networks, we also included a classification with 98 activations per hidden layer, where we calculated each layer in two shots (i.e., tiles). The accuracy of our network was even higher at 96.8% accuracy without fine-tuning or retraining and 97.8% accuracy with fine-tuning, compared with a 98.2% ground truth.

Repeatability

With random noise fluctuations between trials, there is some variability in the accuracy of our system. To test the repeatability of our experiments, we classified the first 1,000 MNIST test images with networks of two different sizes ten times in a row (see Fig. 5.13). The accuracy varies by up to 0.5% for the larger network we tested and 0.7% for the smaller network.

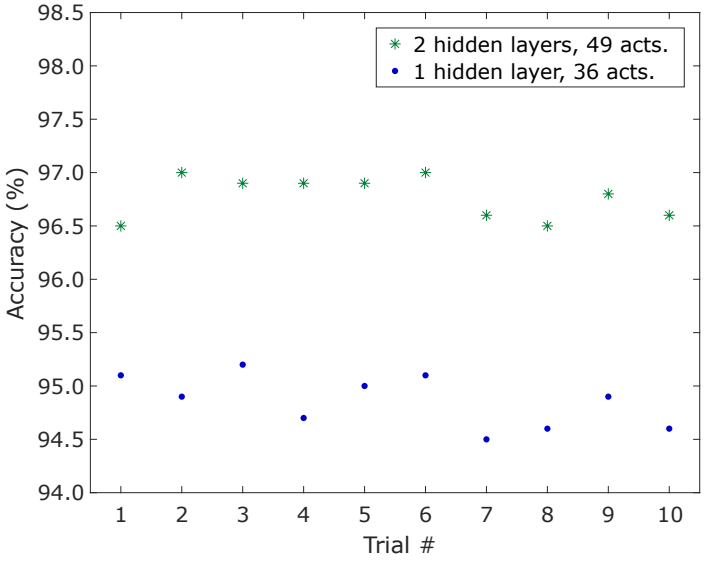

Figure 5.13: Repeatability experiment showing classification accuracies of the first 1,000 previously unseen MNIST test images (initially shuffled). Different trials use the same inputs and weights. Two networks were tested: $784 \to 49 \to 49 \to 10$ and $784 \to 36 \to 10$.

5.7 Optical limit to throughput

Next, we investigated the optical limit to throughput of our system. The clock rate is fundamentally limited by source bandwidth, where a broader laser spectrum can produce shorter pulses. But a wide spectrum yields blurred outputs: by the spatial Fourier transform relationship between the Fourier and image planes, the distance between the center of each replicated input pattern and the optical axis is linear in wavelength (see below). Therefore, the classification error increases with bandwidth. We measured this error by repeating the MNIST classification experiment with two hidden layers of 25 activations each with a supercontinuum source (SuperK EXW-12 from NKT with a VARIA tunable filter) in the place of the CW laser. We classified 1,000 randomly selected images from the test set at different source spectral widths ($2\sigma_\lambda$: twice the RMS width of the spectrum, i.e., twice the standard deviation, to account for the spectrum's irregular shape).

5.7.1 Error model: accuracy vs. optical bandwidth

In this section, we model classification accuracy as a function of optical source bandwidth. We first investigate the wavelength dependence of the outputs.

The height and width of each individual input replica stays constant with wavelength since SLM #1 is imaged to the camera, and any chromatic aberration would cause slight blurring from a shift in the position of the focal plane rather than a change in magnification. There will also be a small change in contrast for SLMs #1 and #3 since the calibration performed at $\lambda_0 = 532$ nm will deviate a little bit from the desired linear relationship. For SLM #2, in phase mode, there is a slight loss of phase contrast for wavelengths not equal to 532 nm as well. The main source of error, however, arises because the fan-out spot pattern on the camera, $\tilde{\mathbf{P}}(\mathbf{r}_\mathrm{d})$, is the spatial Fourier transform of the pattern on SLM #2, $\mathbf{P}(\mathbf{r}_\mathrm{F})$, with an argument of $\mathbf{r}_\mathrm{d} = 2\pi \cdot \mathbf{r}_\mathrm{F}/(\lambda \cdot f_2)$. Therefore, at different wavelengths, though the spatial Fourier transform function remains the same, the argument changes:

$$\mathbf{r}_\mathrm{d}(\lambda) = (\lambda_0/\lambda) \cdot \mathbf{r}_\mathrm{d}(\lambda_0). \tag{5.1}$$

The distance between the zero order and each diffracted spot (i.e., the center of each input replica) thus increases linearly with wavelength under the paraxial approximation. The overlapping, offset images spanning the full wavelength band of the source are then blurred on the receiver.

To simulate this blurring, we first measured the broadband spectra of our supercontinuum laser at different filter bandwidth settings with a custom spectrometer available in our laboratory. We also corrected these spectra to account for our system's wavelength response. To determine the response function, we set the VARIA filter attachment on the SuperK to the narrowest bandwidth. We then swept its center wavelength, every 10 nm from 450 nm to 630 nm, and measured the summed intensity on the camera with SLMs #1 and #3 set to a uniform display. (Our setup has low transmission for wavelengths <460 nm or >560 nm, primarily due to the dielectric mirror of SLM #2.) We performed a similar measurement with the supercontinuum laser connected directly to the spectrometer, where we measured and integrated the filtered spectra at the narrowest bandwidth setting for different center

wavelengths. We then estimated our system's wavelength response as the summed camera intensity per wavelength divided by the spectrometer's summed intensity per center wavelength. We could then correct the broadband spectra measured on the spectrometer through multiplication by this response function (interpolated to cover the full wavelength band).

For each sampled wavelength λ in a corrected spectrum (every 0.7 nm), we calculated the expected locations of the input replicas from a linear change in magnification (of λ/λ_0) of the original spot pattern $\tilde{\mathbf{P}}(\mathbf{r}_\mathrm{d}(\lambda_0))$. We then set the intensity of all input replicas for the wavelength λ to the intensity of the optical spectrum at λ. The sum of the predicted replicated inputs over the entire spectrum then produces the simulated blurred images, which we used as inputs to our two-hidden-layer DNN in inference on a digital electronic computer. The inputs to every layer were blurred following the same procedure. We also calculated the blurring and accuracy degradation from Gaussian spectra to model broader bandwidths (which our experiment did not support due to the dielectric mirror of SLM #2).

5.7.2 Experiment to determine maximum optical bandwidth

We ran the classification experiment with a network of size $784 \to 25 \to 25 \to 10$ on our optical setup, with results shown in Fig. 5.14. The measured experimental error is ϵ_exp, and the simulated error is ϵ_sim with our corrected supercontinuum spectra and ϵ_Gauss with broader ideal Gaussians. Due to noise in the optical setup, for a given spectrum, $\epsilon_\mathrm{exp} > \epsilon_\mathrm{sim}$, but ϵ_exp follows a similar trend to ϵ_Gauss (shifted to lower $2\sigma_\lambda$). ϵ_exp doubles from 5.4% with the CW diode to 11% at $2\sigma_\lambda = 21$ nm, which we define as the widest acceptable source bandwidth for preserved accuracy. The Fourier transform of the corresponding source spectrum yields a pulse of full width at half maximum ∼0.02 ps. Therefore, given a transform-limited source in an optimized implementation, the maximum throughput in the first layer is the number of multiply-accumulate operations (19,600) divided by the minimum pulse length, which yields ∼0.9 exaMAC/s. However, a complete, practical system will be limited by modulator, detector and readout speeds, as we describe in greater depth in the next section.

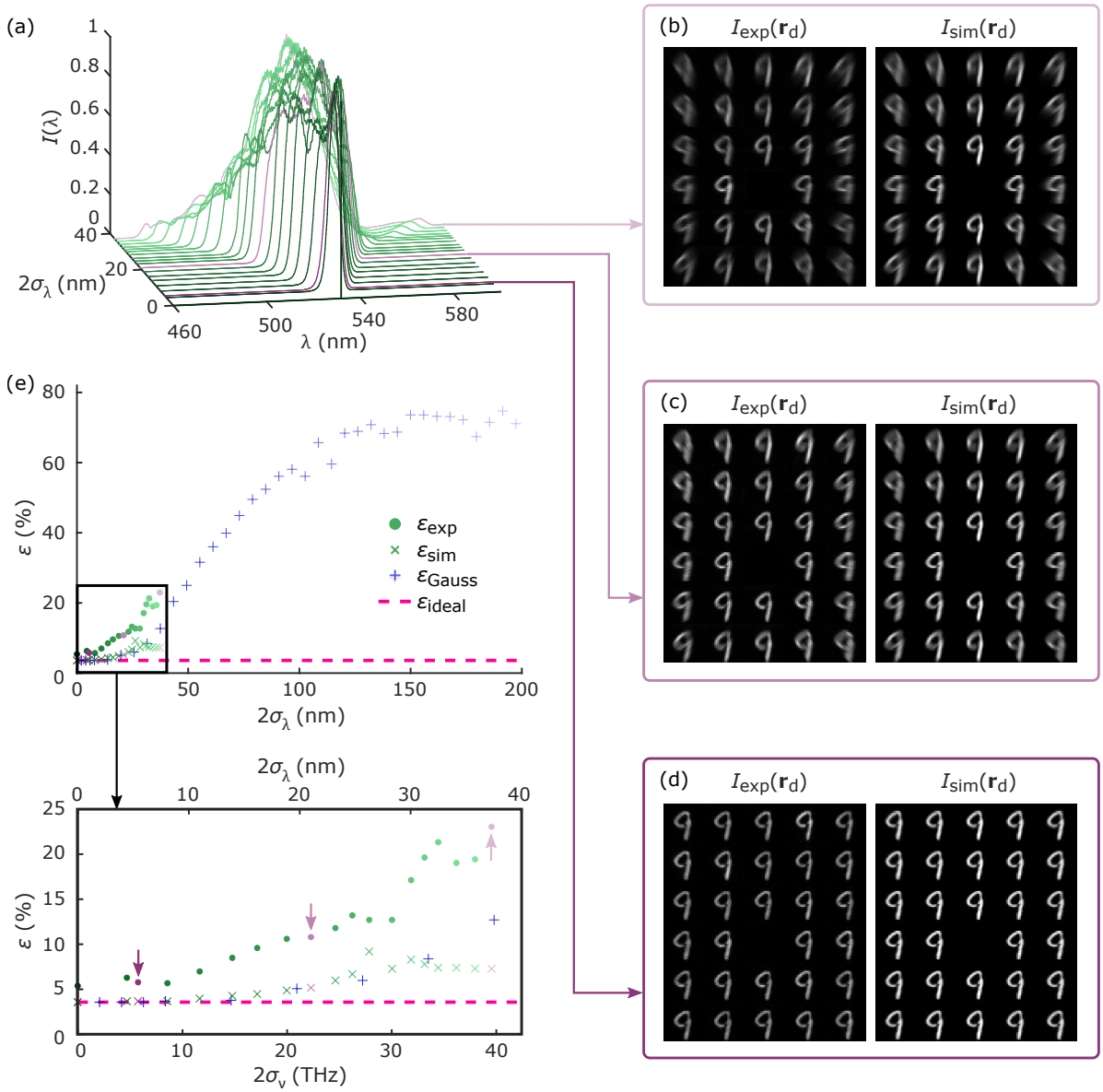

Figure 5.14: Experiment to determine maximum optical throughput of single-shot ONN: MNIST classification with $784 \rightarrow 25 \rightarrow 25 \rightarrow 10$ FC-NN using filtered supercontinuum laser as source in setup shown in Fig. 5.3. **a**, Laser spectra. **b-d**, Example blurred images from source spectral widths of 37 nm, 21 nm and 5.4 nm. Left images acquired on camera; right images simulated from corrected supercontinuum spectra. Element [4,3] overlaps with zero order from SLM #2 and is cut from the images and replaced with element [6,4] in DNN experiments. **e**, Classification error of 1,000 previously unseen MNIST test images versus source spectral width ($2\times$ RMS width, i.e., $2\sigma_\lambda$) measured experimentally (ϵ_{exp}) and simulated from supercontinuum (ϵ_{sim}) and Gaussian (ϵ_{Gauss}) spectra; ground truth error without blurring for reference (ϵ_{ideal}). Arrows indicate results from spectra shown in purple ($2\sigma_\lambda = 37$ nm, 21 nm and 5.4 nm).

5.8 Performance of a near-term optimized system

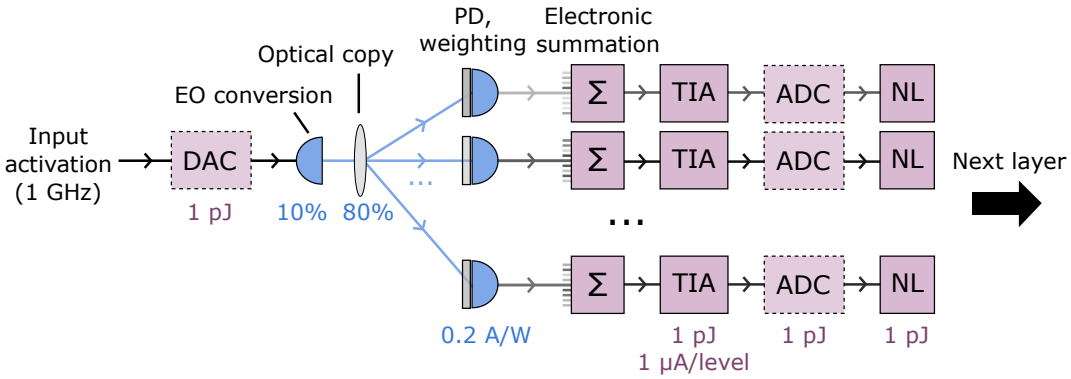

Figure 5.15: Path of one input activation through the single-shot ONN in an optimized setup. VCSEL or μLED converts signal from electronic to optical domain (EO conversion). Optical copy is a reconfigurable diffractive optical element. Each photodetector (PD) includes a weighting element and is electrically connected to K other PDs for analog electronic summation. Transimpedance amplifier (TIA) reads out the analog output signal from each block, which the DAC then converts to the digital domain. Nonlinearity is a simple comparator. Each of the K input activations goes through these processing steps simultaneously such that after one pass, the matrix-vector product is complete.

Our proof-of-concept experiment demonstrates accuracy and scaling and is not optimized for speed or energy consumption, as would be the case with custom transmitter and receiver arrays. Because of the input SLM and the readout camera, our system operates on the order of 10 Hz (the fan-out and weighting SLMs can remain static and are not restrictive, since the weighting SLM is updated just between layers and the fan-out SLM pattern is only changed when the model shape changes). The components in our experiment have a correspondingly high energy consumption (each <10 W).

In the next iteration of the system, the aim should be to demonstrate high throughput and low energy consumption. In this section, we estimate the efficiency metrics for an optimized near-term, CMOS-compatible system, where an input activation follows the path shown in Fig. 5.15. This system includes digital-to-analog converters (DAC), a high-speed source array with K elements, LCoS SLMs for fan-out and weighting, PDs for optical-to-electrical (OE) conversion, analog electronic summation, and for each block, amplification by a transimpedance amplifier (TIA), analog-to-digital conversion (ADC) and a nonlinearity (NL). We assume a matrix size of 1 million elements ($N = K = 1{,}000$), as described below,

since megapixel cameras and SLMs are readily available.

5.8.1 Maximum DNN layer size

The number of pixels in the weighting SLM and camera limit the maximum weight matrix size ($N \times K$) that can be applied in one pass through our system. The number of pixels in the fan-out SLM (DOE) in the Fourier plane dictates the number of spots that can be generated in the image plane, which in turn, sets the maximum output vector length N. In Ref. [85], with a 1.3-megapixel SLM, 1500 uniform spots were experimentally generated (which translates to $N = 1500$ in our scheme).

Considering these constraints, with our megapixel spatial light modulators, our setup can theoretically perform million-element matrix-vector multiplication, with $N = K = 1000$. Our system should therefore be able to accommodate some of the largest DNN layer sizes currently in use, such as those found in Transformers [8]. In other DNN accelerators that have limited scalability, matrices must be fetched from memory over multiple time steps, subjecting them to the same 'memory wall' that currently bottlenecks digital electronics.

5.8.2 Latency and throughput

The optimized single-shot ONN's latency is defined by the sum of the latencies of each of the components encountered by one input activation. The DAC, light source, TIA and ADC take ∼1 ns each (a standard computer clock operates at ∼GHz). In terms of the nonlinearity, Ref. [122] demonstrates an optical ReLU function that operates in sub-ps time scales and also surveys digital electronic, analog electronic and optoelectronic implementations of ReLU with latencies ≤ ns. With a photon time of flight of <10 ns, the system thus operates with a latency on the order of ∼10 ns for a full matrix-vector computation, independent of vector length (as long as the weight matrix fits onto the hardware). In weight-stationary digital electronics (e.g., systolic arrays like Google's TPU [46]), on the other hand, where inputs are message-passed across the weight matrix due to wiring constraints, the latency is at least $N+K$ clock cycles for an $(N \times K)$-sized MVM. Similarly, in output-stationary architectures [30], [34], [80], latency scales with K, as the inputs are streamed in over time. If $K = N = 1000$,

our proposed near-term optical processor then outperforms these architectures by two orders of magnitude.

In throughput, because operations are pipelined to compute 10^6 MACs every ∼1 ns (assuming 100% utilization), the system can compute ∼10^{15} MAC/s — a throughput on the order of petaMAC/s. Emerging high-speed photodetectors [88], [89] and modulators, e.g., plasmonic electro-optic modulators [123], slow-light silicon modulators [124] or thin-film lithium niobate [125], could potentially achieve even higher throughput in the future. These estimates are summarized in Table 5.2.

Table 5.2: Latency and throughput scaling of different architectures for computation of one DNN layer

Dataflow	Latency[4] (ns)	Throughput[4,5]
Weight-stationary (near-term)[1]	∼10	$N \times K$/1 ns → ∼petaMAC/s
Weight-stationary (physical limit)[2]	∼10	19,600/.02 ps → ∼exaMAC/s
Systolic array (e.g., TPU [46])[3]	$N + K \rightarrow$ 2,000	$N \times K$/1 ns → ∼petaMAC/s
Output-stationary (e.g., [34])	$K \rightarrow$ 1,000	$N \times K$/1 ns → ∼petaMAC/s

[1] Our proposed near-term optimized single-shot ONN (CMOS-compatible).
[2] From our maximum throughput measurements.
[3] In practice, $N = K = 256$ (throughput ∼100 teraMAC/s) in the TPU due to wiring and utilization constraints [46].
[4] Assuming $N = K = 1000$ and 100% utilization.

5.8.3 Energy consumption

In energy consumption, digital electronic DNN accelerators are limited primarily by data movement to ∼0.1-1 pJ/MAC [46], [98], depending on the implementation, process technology and workload. This energy value includes peripheral logic and memory access, as well as ∼25 fJ for the MAC operation alone (from Ref. [108], scaled to a 7 nm node [107]).

The energy of our single-shot ONN, on the other hand, can be on the order of ∼10 fJ/MAC. The energy consumption for a complete matrix-vector multiplication that comprises $N \times K$ MACs is the sum of the DAC, SLM, TIA, ADC and nonlinearity energies, plus the photon energy required to discriminate 256 levels on the TIA:

$$E_{\text{total}} = N \cdot \frac{1}{\eta_s \cdot \eta_d \cdot \eta_{\text{PD}}} \cdot 2^{n_b} \cdot \xi \cdot t + K \cdot E_{\text{DAC}} + 2 \cdot E_{\text{SLM}} \; + N \cdot (E_{\text{TIA}} + E_{\text{ADC}} + E_{\text{NL}}) \quad (5.2)$$

where $\eta_s \approx 10\%$ is the source wall-plug efficiency [73], [126], $\eta_d \approx 80\%$ is the optical efficiency

of the DOE, $\eta_{\text{PD}} \approx 0.2$ A/W is the PD responsivity [90], $n_b = 8$ bits, $\xi \approx 1$ μA is the TIA sensitivity at 1 GHz [91], $t = 1$ ns is the integration time, i.e., computer clock cycle time, E_{SLM} is the energy consumed by the SLM in one clock cycle (<10 nJ) and each of the remaining component energies (TIA [91], ADC [127], [128], DAC [128], [129], NL [122]) are 1 pJ per operation or less. The optical efficiency of the DOE, η_{d}, depends first on the light utilization and diffraction efficiency of the SLM (both ∼95% in commercial LCoS SLMs such as our Hamamatsu X10468). Second, it is also determined by the fraction of optical energy in the desired spots generated by the fan-out pattern (defined by the Fourier transform relation), which can be >90% for ∼100-1000× fan-out [85], [130]. The product of these two factors yields >80%. For $N = K = 1000$, the energy per MAC ($E_{\text{total}}/(N \cdot K)$) is therefore on the order of ∼10 fJ/MAC. These parameters and calculations are summarized in Table 5.3.

This overall energy per MAC, including electrical-to-optical and optical-to-electrical conversion, is 1-2 orders of magnitude lower than digital electronic accelerators. In fact, it is similar to the cost of one digital electronic MAC, before even considering the expensive data movement in digital electronic accelerators.

Table 5.3: Parameters in energy calculation

Symbol	Parameter	Value[2]	Fan-out	Energy/MAC[4]
η_{s}	Laser wall-plug efficiency	10% [73], [126]		
η_{d}	Optical efficiency of DOE	>80% [85], [130]		
η_{PD}	PD responsivity	0.2 A/W [90]		
n_b	Effective number of bits	8		
ξ	TIA sensitivity	1 μA [91]		
t	Clock cycle time	1 ns		
E_{DAC}	Energy per DAC conversion	1 pJ [129],[128][3]	N	1 pJ/N → 1 fJ/MAC
E_{SLM}	Energy of LCoS SLM	(<10 W)·t	$N \times K$	<10 nJ/($N \times K$) → <10 fJ/MAC
E_{optical}	Optical energy per block[1]	$\frac{1}{\eta_{\mathrm{s}} \cdot \eta_{\mathrm{d}} \cdot \eta_{\mathrm{PD}}} \cdot 2^{n_b} \cdot \xi \cdot t \approx 10$ pJ	K	10 pJ/K → 10 fJ/MAC
E_{TIA}	Energy of TIA	1 pJ [91]	K	1 pJ/K → 1 fJ/MAC
E_{ADC}	Energy per ADC conversion	2 pJ [127], [128]	K	2 pJ/K → 2 fJ/MAC
E_{NL}	Energy of nonlinearity	<1 pJ [122]	K	<1 pJ/K → <1 fJ/MAC
E_{total}	Energy of full system			∼10 fJ/MAC

[1] Optical energy for 2^{n_b} distinguishable levels by the TIA.
[2] Demonstrated in the literature.
[3] DAC within ADC.
[4] Assuming $N = K = 1000$.

Table 5.4: Projected chip area

Element	Area (mm^2)	Number of elements	Total area (mm^2)
Weighting element[1]	$1.4 \cdot 10^{-5}$ [131]	10^6	14
TIA	0.0022 [91]	10^3	2.2
ADC	0.0016 [132]	10^3	1.6
NL	0.001 [133]	10^3	1
DAC	<0.0016 [132][2]	10^3	<1.6
VCSEL	.01 [134][3]	10^3	10
Area of full system			30

[1] Includes PD (e.g., PDs in Refs. [54], [88]), since weighting element is placed on top of each PD.
[2] DAC within ADC.
[3] Aperture of 6 μm, pitch of 100 μm in demonstrated array.

5.8.4 Chip area

In terms of footprint, with bulk optics, the single-shot ONN consumes more overall volume than digital electronic accelerators, but as discussed for the DONN, this volume does not contribute to fabrication cost and can be used for air cooling between racks. The system can also be condensed with shorter-focal-length, aberration-corrected lenses.

The total chip area of the optimized system is calculated in Table 5.4. The component areas have been demonstrated experimentally in the literature, in CMOS technology nodes larger than the state of the art – therefore, the TIA, ADC, NL, DAC and VCSEL areas could likely be further miniaturized. The weighting elements are limited not only by the liquid crystal cell size, but also by optical spot size. The diffraction limit is sub-μm for green light, so a pixel size of a few μm can maintain low crosstalk in a real system that includes imperfections (i.e., aberrations and misalignment). The weighting elements consume the most area in this projected system, whose total area of ∼30 mm^2 is similar to the area consumed by 1000 × 1000 electronic MAC units without peripheral logic: in a digital electronic weight-stationary array, each MAC unit has an area of (5-8 μm)^2 (8-bit multiplier [109], [110], [112] scaled to a 7 nm node [107]), for a total million-element area of 25-64 mm^2.

5.9 Discussion

We introduced a scalable optical neural network that can compute DNN layer outputs in a single shot. In our proof-of-concept experiment, we demonstrated low loss of classification

accuracy in inference on the MNIST, Fashion-MNIST and QuickDraw datasets with analog optical data encoding, fan-out and weighting in all layers of FC-NNs of varied sizes. We used standard DNN models, without having to rely on all-digital output layers in inference or physical hardware models in training to achieve high accuracy. We also used a filtered supercontinuum source to find that a minimum pulse duration of ∼0.02 ps maintains the MNIST classification error within a factor of two (using a 2-hidden-layer FC-NN with 25 activations per hidden layer) – yielding a near-exascale throughput limit. We also estimated the practical limitations of a near-term system with the same architecture, but using optimized transmitter and receiver chips. We calculated its latency, throughput, energy consumption and chip area. Below, we describe paths forward for the next generation of optical experiments, including potential increases in accuracy and DNN layer size.

5.9.1 Improving accuracy

To improve accuracy in our experimental demonstration, we can reduce stray light and crosstalk in the system. As we saw in our measurements and simulations (Sections 5.5 and 5.6.1, Figs. 5.8, 5.9 and 5.10), they are the main drivers of error in DNN inference: we measured the amount of noise present when transmitting a single data point per time step through our optical setup, and when we injected this noise into MNIST classification simulations, we showed near-equivalent accuracies to the noiseless ground truth. These simulations indicate that once we reduce crosstalk and spurious reflections, we will significantly boost inference accuracy. Some paths toward this goal are to use higher fill-factor SLMs with reduced reflections from their backplanes (e.g., with a dielectric mirror), aberration-corrected lenses that can focus spots more tightly in the image planes to lower crosstalk (see, for example, the Supplement of our theory paper [34]), and better optical coatings which can decrease unwanted interference fringes.

Aided by these modifications, our system will be able to process wider DNN layers, up to approximately $N = K = 1000$, limited by the number of pixels in the camera and SLMs. In our experimental demonstration, we found the current maximum layer size to be restricted by signal-to-noise ratio (noise as discussed above and insufficient laser power). Our higher-accuracy results with the two-tile, 98-activation-per-hidden-layer networks point to

the possibility of accuracy improvements with larger networks. We can also add biases and use deeper DNNs, i.e., a greater number of hidden layers, to continue to lower classification error. Additionally, to preserve accuracy for broader source spectra, we could modify our DNN model with simulated blurred images as inputs during the training phase.

5.9.2 Further performance improvements

To further reduce energy consumption of the single-shot ONN beyond the proposed optimized setup, the fan-out and weighting LCoS SLMs could be replaced with emerging SLM schemes [79] or, as the DOE and weights are only updated when the model changes, with elements that have zero static power consumption, e.g., with an array of optical phase change material cells [135]–[137], a fixed phase mask or MEMS modulators [138]. With an increase in source and detector efficiencies [54], the overall energy consumption could then reach the single-femtojoule-per-MAC regime. DAC and ADC costs can also be eliminated with analog nonlinearities [49], [122], as analog output activations from one layer can be directly used as inputs to the next layer. The inputs to the first layer may also be in the analog domain, e.g., as they are read out from a sensor.

As networks are becoming steadily larger (e.g., GPT-3, with fully connected layers with input vectors up to a length of 10,000 elements [16]), the system may need to be further scaled up beyond $K \approx 1000$. One way to increase the number of matrix elements and outputs is to use SLMs with more pixels, such as the commercially available 10-megapixel SLM in Ref. [131]. Another option could be to use multiple SLMs and combine their outputs with a series of beamsplitters. Lastly, at the cost of decreasing energy efficiency and increasing latency, the inputs can be tiled and the weights updated over different time steps with a small local electronic memory at each weight pixel. In state-of-the-art nodes, an SRAM cell's area is $0.021\mu m^2$ [139]. Therefore, $\sim$80 digital values of 8 bits each can fit into a CMOS chip area of $(3.74\mu m)^2$, which is the area of an LC pixel in high-resolution SLMs [131]. This trade-off space can be evaluated and optimized in future work, e.g., using the mapping tool Timeloop [36].

5.9.3 Negative weights

A number of solutions can implement negative weighting. For example, all the PDs in a block can be connected by two wires instead of one; charge from the negatively weighted pixels would be directed into the second wire with an analog switch. The output from the 'negative' wire can then be subtracted from the output of the 'positive' wire. Because this subtraction only occurs once per block, its cost is amortized by a factor of K and is therefore small with respect to the other costs of the system. Another possibility for negative weighting is to use two PDs per receiver pixel, where one PD pushes charge into the block's wire in the case of a positive weight, and another pulls charge in the case of a negative weight. The weight value of the unused photodetector is set to maximum extinction. Lastly, the weights can be shifted to all positive values, as described by Wang et al. [61].

5.9.4 Potential extension to convolutional layers

As discussed in Chapter 2, convolutional layers can be used in DNNs alongside MVM, especially in image and video processing. While convolutions can be recast as matrix multiplications, we could instead implement them directly onto our hardware, and thus, avoid the costs of conversion and data-redundancy-related inefficiencies.

In our MVM scheme, the DOE in the Fourier plane displays a fan-out phase pattern that yields a spot array in the image plane (the plane of the weight mask and camera). The N generated spots are spaced by the width of one input image, such that the replicated images do not overlap. Each replicated image pixel can then be individually weighted by the weight mask. For convolution, the phase pattern can be modified such that the spot distance in the image plane is equal to the pixel width, and each spot intensity is equal to a value in the convolutional kernel. Then, each replicated image in the image plane is weighted by the value of the convolutional kernel pixel, and detectors passively sum the overlapping images, achieving the desired convolution. The inputs should not be mutually coherent to average out any interference effect in the summation. The weights are restricted to positive values in this case.

5.9.5 Investigation into different 3D ONN architectures

The weight-stationary architecture of the single-shot ONN is by no means the only possibility for an analog ONN – a system's architecture can be optimized to best suit a given application. For example, we described an analog, output-stationary 3D ONN in our theory paper led by R. Hamerly [34] (the 'HD-ONN', which relies on homodyne detection to perform MACs) and demonstrated small-scale output-stationary ONNs in works led by A. Sludds [140] and Z. Chen [80]. Our preliminary efforts towards a larger-scale HD-ONN experiment are reported in the Appendix. These output-stationary ONNs accumulate partial products over time, utilizing temporal rather than spatial multiplexing in the K dimension (like the DONN, but in the analog domain). Therefore, fewer transmitters and receivers are required compared with the single-shot ONN to process an MVM of equal size, presenting a path to further increase scaling. Future work could explore the trade-off space of lower component area at the cost of higher latency.

5.10 Conclusion

In in this chapter, I presented a single-shot-per-layer inference machine and demonstrated low loss of classification accuracy with analog, reconfigurable optical matrix-vector multiplication. Our plug-and-play hardware can be used with standard networks without data preprocessing or retraining, which, coupled to its CMOS manufacturability, makes it a viable, near-term candidate to overcome the latency and energy bottlenecks of state-of-the-art electronics, at scale.

Chapter 6

Summary and Outlook

This thesis has explored the potential of optical neural networks (ONNs) to overcome the limitations of digital electronics in deep learning. Our work was motivated by the need for efficient data transfer and parallel processing in DNNs, for which optics is inherently well-adapted. The focus of this thesis was on novel feasibility experiments of ONNs at a larger scale and with higher accuracy than had previously been demonstrated. This chapter summarizes our results and proposes related future research directions.

6.1 Overview of contributions

The major contributions of this work are as follows:

- Digital Optical Neural Network (DONN): We described and experimentally validated digital optical data transfer and copying tailored to DNNs. The aim of this project was to open a potential avenue toward addressing the scalability challenges inherent in digital electronic interconnects. In the DONN, optical elements transmit and replicate binary on/off optical pulses of input activations and weights to an array of electronic multipliers. In our experimental demonstration with fixed 1D fan-out by a cylindrical lens, we found these added optical components to have minimal impact on DNN accuracy. Our analysis showed that this configuration of optical interconnects is not energetically favorable over digital electronics for a tightly packed multiplier array.

However, thanks to its length-independent data routing, the DONN can provide an advantage in multi-chiplet or multi-chip modules that are common when scaling digital electronic hardware. As such, the DONN can allow more freedom to designers of digital electronic accelerators by permitting memory or processing elements to be arbitrarily spatially located for scale-up or scale-out.

- Single-shot optical neural network: Building on the DONN, the main project of this thesis was the single-shot ONN. This weight-stationary architecture is highly suitable for applications that require minimal latency, especially when the inputs to be classified start out in the analog optical domain (e.g., machine vision, astrophysics, etc.). Similarly to the DONN, the single-shot ONN performs data transmission and replication optically. But where the DONN only uses optics for digital communication, the single-shot ONN also weights the inputs with analog optoelectronic components. We experimentally demonstrated reconfigurable 2D multicast and weighting in a system that could process DNN inference layers in one shot. We showed reliable DNN classification with standard layers and without prior knowledge of the hardware. This work illustrates the single-shot ONN's potential for accurate DNN computation that can be tailored to a particular DNN shape. Alongside our calculations of possible 1-2 orders of magnitude improvements in latency and energy consumption over a digital electronic systolic array, these results highlight the feasibility of a paradigm shift from all-electronic accelerators to high-efficiency optoelectronic DNN processors.

6.2 Future Research Directions

The findings reported in this thesis suggest a number of directions for future research, including:

- Closely integrating optical and electronic components for full-system demonstrations of high-efficieny ONNs;

- Reduction of crosstalk to improve accuracy and scale up DNN model size in ONN experiments;

- Extensions such as exploiting DNN model sparsity and spectral multiplexing in ONNs to further increase scaling;
- Acceleration of training as well as inference.

Integration of optics with electronics: The development of optimized hybrid systems with optical and electronic components is necessary for high-efficiency demonstrations of both the DONN and the single-shot ONN. Specifically, this integration involves the fabrication of high-speed transmitter and receiver arrays to improve the overall performance metrics of our ONNs, particularly in terms of speed and power consumption. An important note that is key to the practical application of the systems is ensuring their CMOS manufacturability.

Crosstalk reduction (improving accuracy): The reduction of crosstalk remains a critical challenge in our demonstrations of accuracy and reliability. Future work should concentrate on improving the optical design to enhance the fidelity of data transfer, striving to match ONN accuracy to the digital electronic ground truth. Potential paths forward would be to investigate higher fill-factor SLMs, optimizing fan-out patterns, charge-sharing schemes for crosstalk reduction, aberration-corrected lenses, and lower-reflection optical coatings. These improvements are expected not only to boost accuracy on relatively simple workloads like MNIST handwritten digit classification, but also to enhance the system's ability to process wider DNN layers and deepen the network with additional hidden layers to perform more complex tasks.

Extensions and scale-up: Future research could focus on further quantifying and optimizing the energy, latency and throughput benefits of ONNs, especially in the case of $<100\%$ utilization, using specialized mapping tools. Additionally, extending the single-shot ONN system to directly compute convolutional layers, rather than recasting them as matrix multiplications, could enhance the processing efficiency of convolutional neural networks as well. Spectral multiplexing could further increase scalability by allowing for the computation of multiple layer or channel outputs simultaneously. Another possibility for scale-up could be to use an analog output-stationary architecture, like the one described in the Appendix, at the cost of increased latency. Lastly, sparsity within DNN models can be exploited to further boost efficiency. For example, a transmitter array for the input activations would

be well-suited to input sparsity (e.g., after ReLU in the previous layer) as sending a '0' would not consume energy when input photons are not generated. Structured sparsity with grouped weight removal could also reduce the required optical power if the fan-out phase mask is reconfigured appropriately.

Training: Other works (e.g., [53], [62]) have proposed the use of ONNs to accelerate training. While training also relies on matrix multiplication and therefore could potentially also be made more efficient with optical hardware, because of the frequent required weight updates, the efficiency gains with respect to digital electronics need to be analyzed further.

6.3 Conclusion

The development of ONNs, as demonstrated by the DONN and the single-shot optical neural network, shows promise in addressing the scalability and efficiency challenges faced by current DNN hardware technologies. The path forward includes refining these optical systems though co-integration of optics and electronics, accuracy improvements, and the other avenues toward scale-up described above. The application of ONNs may further be expanded to a broader range of domains, potentially enabling complex tasks in other fields such as non-convex optimization (e.g., Ising problems), signal processing and other machine-learning tasks where MVM also dominates energy consumption and latency. With more efficient processors available to computer scientists, we could see new computing possibilities open up, facilitating the next generation of artificial intelligence.

Appendix A

Towards Large-Scale Demonstration of HD-ONN

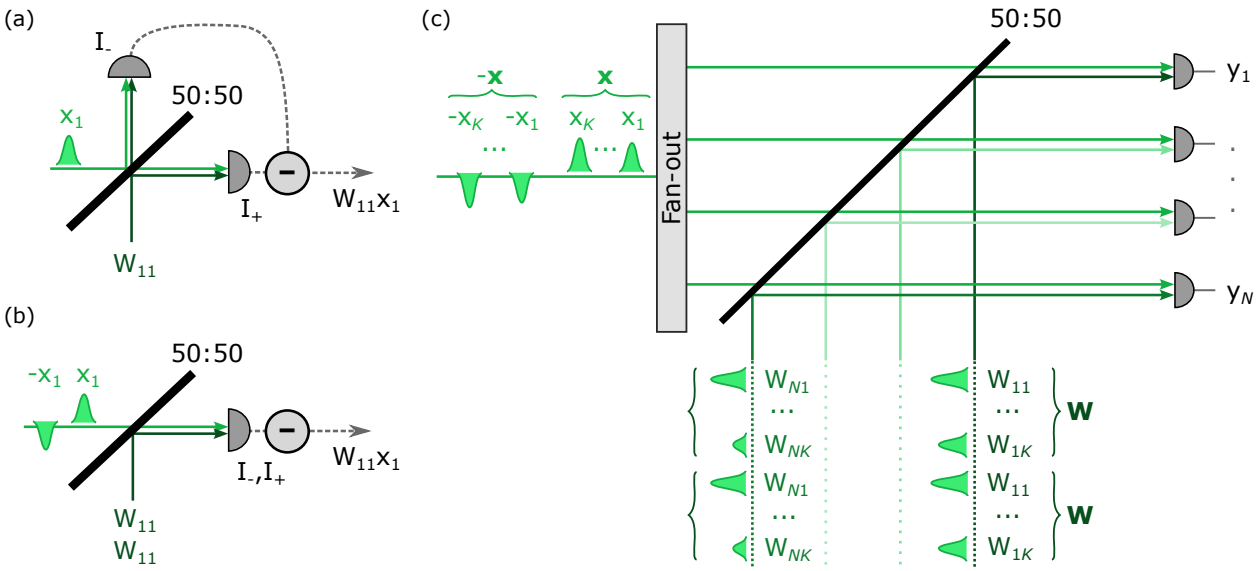

Figure A.1: Time-multiplexed homodyne optical neural network (HD-ONN). **a**, Homodyne detection of incident optical fields of amplitudes x_1 and W_{11} yields the cross term $W_{11} \cdot x_1$ when output intensity I_- is subtracted from I_+. **b**, The same output can be obtained with a single detector in two time steps, with the phase of x_1 flipped in the second time step. **c**, Matrix-vector multiplication: different spatial channels transmit each weight row, and the elements of the input vector are fanned out and broadcast to overlap with the weights on the PD array for passive accumulation over $2K$ time steps. The weights can also be fanned out to be reused for matrix-matrix multiplication (in a 3D architecture like the DONN).

In this Appendix, I outline our work towards a large-scale demonstration of the homodyne

optical neural network (HD-ONN) introduced in our paper [34] led by R. Hamerly. As I describe below, some challenges remain to achieve high-accuracy DNN classification results.

The HD-ONN is output-stationary and relies on optical interference to perform element-wise products. I will first summarize the building blocks of this scheme. In standard balanced homodyne detection (Fig. A.1**a**), coherent light beams at the input ports of a 50:50 beamsplitter with amplitudes x_1 and W_{11} interfere, and the outputs of the two PDs are $I_+ = \frac{1}{2}|W_{11} + x_1|^2$ and $I_- = \frac{1}{2}|W_{11} - x_1|^2$. The difference of the photocurrents $I_+ - I_-$ is proportional to the product $W_{11}x_1$ (our desired weight-activation multiplication). Because inputs and weights are encoded into field amplitudes rather than intensities, x_1 and W_{11} can take on positive or negative values.

To reduce system complexity and eliminate electronic wiring between PDs, we can achieve balanced homodyne detection using a single PD (Fig. A.1**b**). In this case, the input activation x_1 and weight W_{11} are transmitted twice, and a phase modulator applies a phase shift to the second copy of x_1, effectively multiplying its amplitude by -1. The photocurrents I_+ and I_- are then separated in time rather than space, are read out separately and are subsequently subtracted. A full matrix-vector product can be computed with temporal and spatial multiplexing of these element-wise products (Fig. A.1**c**).

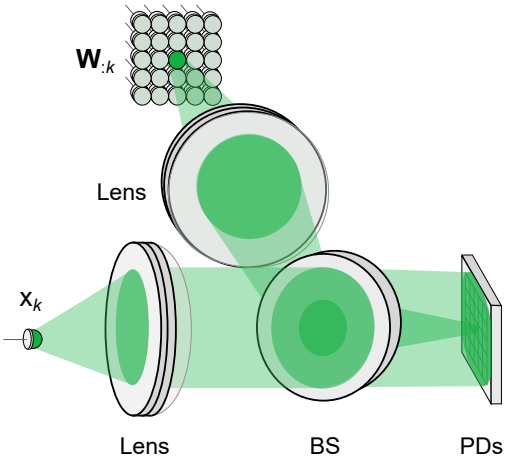

Figure A.2: Implementation of HD-ONN with activation fan-out, but no weight fan-out. Analog inputs and weights encoded into field amplitudes of coherent light. Single-mode light from an input activation source is collimated by a spherical lens to illuminate the full array of PDs. An array of weight sources is imaged to the PDs (light from all sources transmitted in parallel). A 50:50 beamsplitter (BS) allows the k-th fanned out input activation to interfere with the k-th transmitted weight column in one time step.

The HD-ONN can be implemented experimentally in a similar setup to the DONN, but with coherent light sources – compare Fig. A.2 to Fig. 4.2. The main difference is that the digital electronic multipliers and adders are not necessary in the HD-ONN since multiplications are performed through optical interference, with PDs passively accumulating the partial products. Another change from the DONN is that in this HD-ONN scheme, we chose to fan out only the activations and not the weights (though we could fan out both the activations and weights by using linear arrays of transmitters for matrix-matrix multiplication). The reason for this decision was that our goal in this demonstration was to maximize the weight matrix size by using a 2D megapixel SLM. This array of weights is imaged to the PD array, and one input activation is transmitted and fanned out at each time step, which yields a vector-scalar product per time step. Passive accumulation of the vectors gives us our desired matrix-vector product (see Fig. A.7**a** for matrix visualization of these operations).

With this implementation, we aimed to show the potential for computation of DNN layers with millions, or even billions, of weights. Compared with the demonstration in Ref. [80], in which an array of 24 VCSELs transmits the weights at high speed, our system would be slower, but could operate at a much larger scale. Figure A.3 illustrates our experimental setup, where the LCoS SLM (Meadowlark P1920-400-800-HDMI-T, pixel width 9.2 μm) encodes one weight column $\mathbf{W}_{:k}$ per time step, and a modulator encodes one activation x_k per time step. The camera is the Zelux from Thorlabs with 3.45 μm-wide pixels. The modulator can be, e.g., an electro-optic modulator (EOM) or a variable retarder (can be thought of as a single-pixel liquid-crystal SLM). If using an EOM, photorefractive and DC drifts need to be taken into account – a feedback loop based on output power can determine the appropriate set point. In an optimized version of the setup, the activations and weights would both be encoded by high-speed transmitters such as the VCSELs in Ref. [80], and would both be fanned out to maximize data reuse and energy savings.

Similarly to the single-shot ONN, we use a PBS and half-wave plate with the LCoS SLM such that when we have a weight value of '0', there is no light transmitted. We decided to operate in this mode rather than standard phase mode to try to reduce error at low activation and weight values. This point will be made clearer in the following subsections,

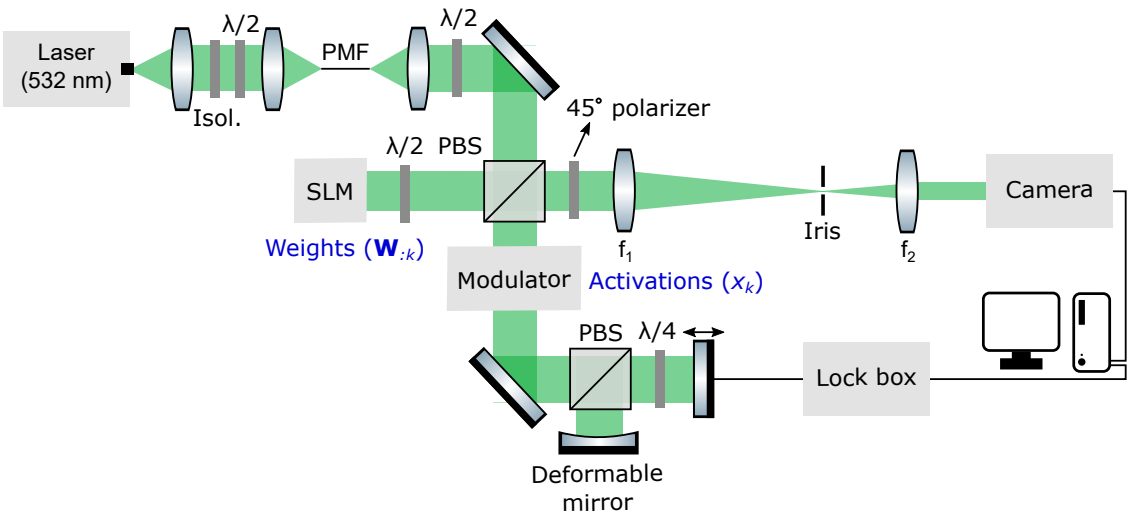

Figure A.3: Proof-of-concept implementation of HD-ONN. As in the single-shot ONN, collimated laser light with 45° polarization after half-wave plate (λ/2) is incident on an LCoS SLM. The SLM encodes one column of the weight matrix per time step, $\mathbf{W}_{:k}$. Relay lenses image the SLM to the camera with $f_1 = 200$ mm and $f_2 = 75$ mm for 1:1 pixel matching. The modulator (e.g., amplitude EOM or variable retarder) encodes one activation x_k at a time. Input activations and weights interfere when projected onto the same polarization with the 45° polarizer. The lock box (PID loop) performs active path length stabilization with a piezo mirror. Digital computer controls hardware and implements nonlinearity.

as we look into the Jones matrices that describe the system.

Weight arm (SLM)

We assume that the phase imparted by the SLM that is parallel to its extraordinary axis $\phi_{//}$ is modifiable, but the perpendicular phase $\phi_{\perp}$ is fixed. $\phi_{//}$ and $\phi_{\perp}$ are position-dependent. (We will define the weight W with respect to $\phi_{//}$ later.) Below, we calculate the Jones vector that describes the resulting components of the field amplitude at the camera, defined with respect to the SLM's axes (that also correspond to horizontal and vertical polarizations on the optical table). The starting point for the computation is just after the first pass through the PBS; the Jones matrices are in the opposite order of the optical propagation. We assume a simplified transformation by the SLM for illustration of the scheme, though we note that an SLM can have coupled phases and off-diagonal elements.

Incident perpendicular polarization → half waveplate @ 22.5°→ SLM →

half waveplate @ -22.5°→ PBS → polarizer @ 45°

$$\begin{bmatrix} E_{\text{out}//} \\ E_{\text{out}\perp} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} -e^{i\phi_{//}} & 0 \\ 0 & e^{i\phi_\perp} \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{A.1}$$

$$= \frac{1}{4}\begin{bmatrix} -e^{i\phi_{//}} + e^{i\phi_\perp} \\ -e^{i\phi_{//}} + e^{i\phi_\perp} \end{bmatrix} \tag{A.2}$$

The intensity on the camera (summing both polarizations) is thus:

$$I = |E|^2 \tag{A.3}$$

$$= 2 \cdot \tfrac{1}{4}(-e^{i\phi_{//}} + e^{i\phi_\perp}) \cdot \tfrac{1}{4}(-e^{-i\phi_{//}} + e^{-i\phi_\perp}) \tag{A.4}$$

$$= \tfrac{1}{4}(1 - \cos(\phi_{//} - \phi_\perp)) \tag{A.5}$$

The SLM can be calibrated for a linear response versus display value.

Input activation arm

In the input activation arm, we assume pure amplitude modulation A with a phase ξ that describes the path-length mismatch between the input and weight arms. ξ is position-dependent because the non-flatness of the SLM causes a wavefront mismatch between the two arms. ξ is also time-dependent, with air currents and vibrations that can cause path length fluctuations in the input activation and weight arms. The Jones vector that describes the output from the activation arm is therefore:

$$\begin{bmatrix} E_{\text{out}//} \\ E_{\text{out}\perp} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} A \cdot e^{i\xi} \\ A \cdot e^{i\xi} \end{bmatrix} \tag{A.6}$$

with a measured intensity on the camera (again, summing both polarizations) of $I = \frac{1}{2}A^2$.

Interference of weight and activation arms

In this section, we will examine how the light from the input activation and weight arms interferes on the camera. We will first look at how we can correct path length mismatches

across the camera field of view. After that, we will see how light modulated by an LCoS SLM pixel in the weight arm interferes with amplitude-modulated light from the input arm to yield an element-wise product.

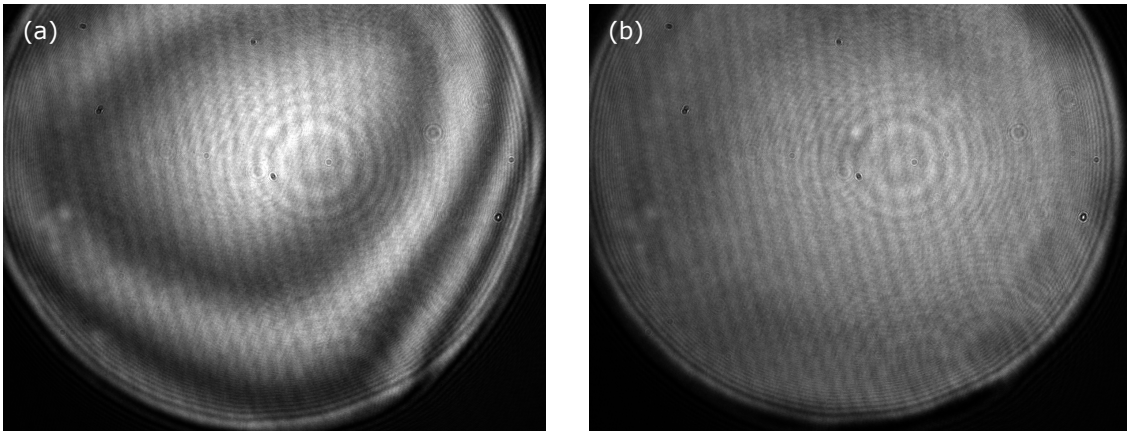

Figure A.4: Interference fringes observed on camera with SLM pixels and modulator set to maximum transmission. Deformable mirror calibrates the wavefront in the activation arm to match the weight arm. **a**, mirror unactuated and **b**, actuated.

We want to eliminate the phase mismatch between the two arms ξ, where we will first remove its spatial dependence using a deformable mirror. Then, through active path length monitoring and correction, we will set ξ to a constant value with respect to time. Figure A.4 shows a camera image obtained from interference of the input activation and weight arms with the SLM and modulator set to constant, uniform values. The wavefront error can be seen by the fringe pattern in Fig. A.4**a**. By adding a deformable mirror to the input activation arm that matches the surface shape of the SLM, we can eliminate these fringes (Fig. A.4**b**). ξ is now time-dependent, but is uniform across the field of view. Next, we can set ξ to a fixed value with active path length stabilization. We added a mirror glued to a piezoelectric stack to the activation arm (which I will call 'piezo mirror' going forward). A proportional–integral–derivative controller (LaseLock, TEM Messtechnik) uses the output signal from the camera to set the position of the mirror along the optical axis[1]. Figure A.5 shows the resulting intensity on the camera over time with the SLM and modulator set to a constant, uniform value with and without active stabilization. The required calibration measurements to set the mirror position can be interspersed in the data acquisition.

[1]Thanks to Dr. Zaijun Chen for help selecting and setting up the lock box.

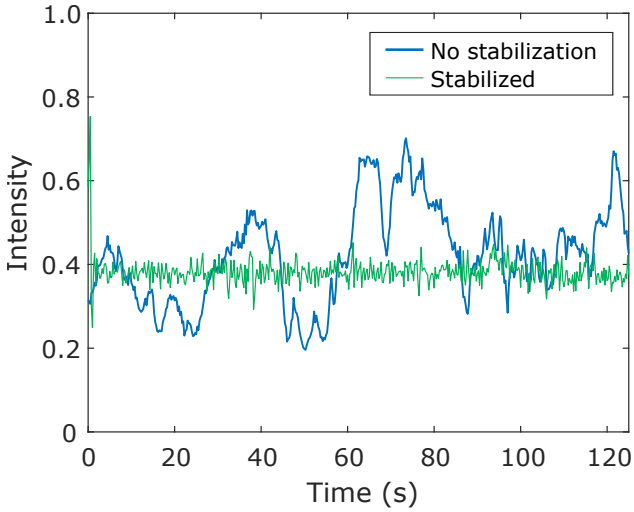

Figure A.5: Active path length stabilization in activation-weight interferometer using piezo mirror connected to lock box. Intensity is the sum of pixel values in a central region on the camera. Weight SLM and input activation modulator set to constant, uniform values.

The intensity measured on the camera with the interfering light from both arms (assuming A is real) is:

$$I = 2 \cdot (-\tfrac{1}{4}e^{i\phi_{//}} + \tfrac{1}{4}e^{i\phi_{\perp}} + \tfrac{1}{2}A \cdot e^{i\xi})(-\tfrac{1}{4}e^{-i\phi_{//}} + \tfrac{1}{4}e^{-i\phi_{\perp}} + \tfrac{1}{2}A \cdot e^{-i\xi}) \tag{A.7}$$

$$= \tfrac{1}{4} + \tfrac{1}{2}A^2 - \tfrac{1}{4}\cos(\phi_{//} - \phi_{\perp}) - \tfrac{1}{2}A \cdot \cos(\phi_{//} - \xi) + \tfrac{1}{2}A \cdot \cos(\phi_{\perp} - \xi)) \tag{A.8}$$

Defining a new phase $\phi = \phi_{//} - \phi_{\perp}$ and setting $\xi = \phi_{\perp} + \pi/2$ with the piezo mirror, we have:

$$I(\phi, A) = \tfrac{1}{2}A^2 + \tfrac{1}{4}(1 - \cos\phi) - \tfrac{1}{2}A \cdot \sin\phi \tag{A.9}$$

Flipping the sign of ϕ (or A):

$$I(-\phi, A) = \tfrac{1}{2}A^2 + \tfrac{1}{4}(1 - \cos\phi) + \tfrac{1}{2}A \cdot \sin\phi \tag{A.10}$$

Then, we have our desired product between the activation (A) and the weight ($W = \sin\phi$) by subtracting equation A.9 from equation A.10:

$$I(-\phi, A) - I(\phi, A) = A \cdot \sin\phi \tag{A.11}$$

The sinusoidal relationship can be taken into account in the SLM lookup table during calibration.

As a sanity-check of equation A.9, if $A = 0$, we get back the expression we expect from equation A.5:

$$I(-\phi, A = 0) = I(\phi, A = 0) = \tfrac{1}{4}(1 - \cos\phi) \tag{A.12}$$

If $\phi = 0$:

$$I(\phi = 0, A) = \tfrac{1}{2}A^2 \tag{A.13}$$

If the weights and activations are both small, then we approach zero photons on the detector. Smaller values tend to have higher transmission accuracy, as we saw with the single-shot ONN, so this scheme is preferable over a phase-only approach, where we would have to take the difference of two larger intensity values to find an output near zero and we would have compounding errors.

Initial system characterization

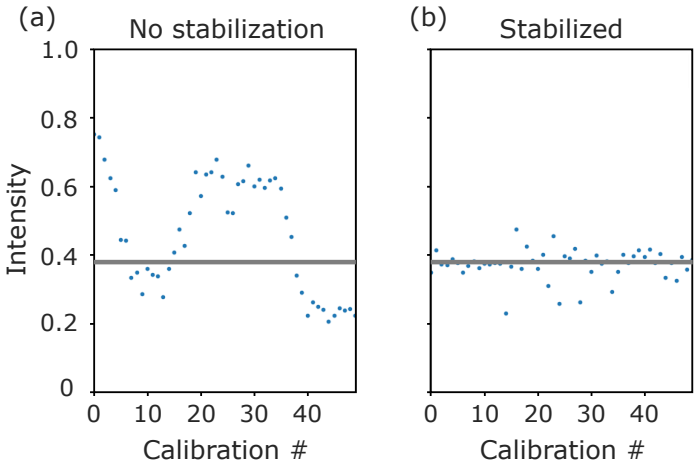

Figure A.6: Active path length stabilization similar to Fig. A.5, but here, stabilization occurred during data collection (interspersed). Specifically, weight and input activation data are displayed on the SLM and modulator, respectively, then 9 calibration frames are acquired for the lock box to set the piezo mirror position. Intensity shown here is the averaged intensity of the center of the 9th calibration frame, i.e., the last frame before data acquisition.

Figure A.5 showed that active path length stabilization by a piezo mirror reduces intensity fluctuations in the interferometric signal. However, we found these fluctuations to be worse when we tried to perform the stabilization between MNIST data transmission frames (Fig. A.6). An alternative solution could be to use a dedicated area of the SLM for stabilization feedback, which could be split from the main signal with a beamspitter and aperture and detected on a separate PD connected to the lock box.

Before looking further into refining the path length stabilization, we wanted to check that the general concept of the experiment would yield accurate DNN classifications. Therefore, to eliminate the path length stabilization requirement, we displayed both the activations and the weights on the same LCoS SLM (Fig. A.7). We have 1:1 imaging from the SLM to the camera, but we can blur the received activation and weight pixels on the camera by closing the iris such that their light interferes in a middle pixel. The light from two interfering pixels follows an almost identical optical path, so any path length fluctuations will happen in tandem. This experiment does not demonstrate optical fan-out, but has the advantage of decoupling different sources of error to allow us to test the basic concepts of multi-pixel interference and negative weighting by an LCoS SLM in 'amplitude mode' for coherent DNN matrix computations. We can flip the phase of the activations (or the weights) to get 'negative' images, and subtract these 'negatives' to keep just the interference term.

We initially encountered difficulties with the phase settings of the SLM and the variable retarder, as $\phi_{\perp}$ and $\phi_{//}$ turn out to be coupled, as I mentioned earlier. We found that this coupling has a smaller effect when we restricted the amplitude to small values (near zero, where the phase flips from positive to negative, i.e., where the bright and dark interference fringes swap). Figure A.8 shows initial characterization data from the simplified experiment. The received element-wise dot products roughly follow the ideal dotted red line, which is promising. However, the received products do not follow a linear relationship to the ground truth, deviating further from ideal behavior when several pixels are used simultaneously (Fig. A.8**b**). This issue can potentially be addressed by refining the system calibration.

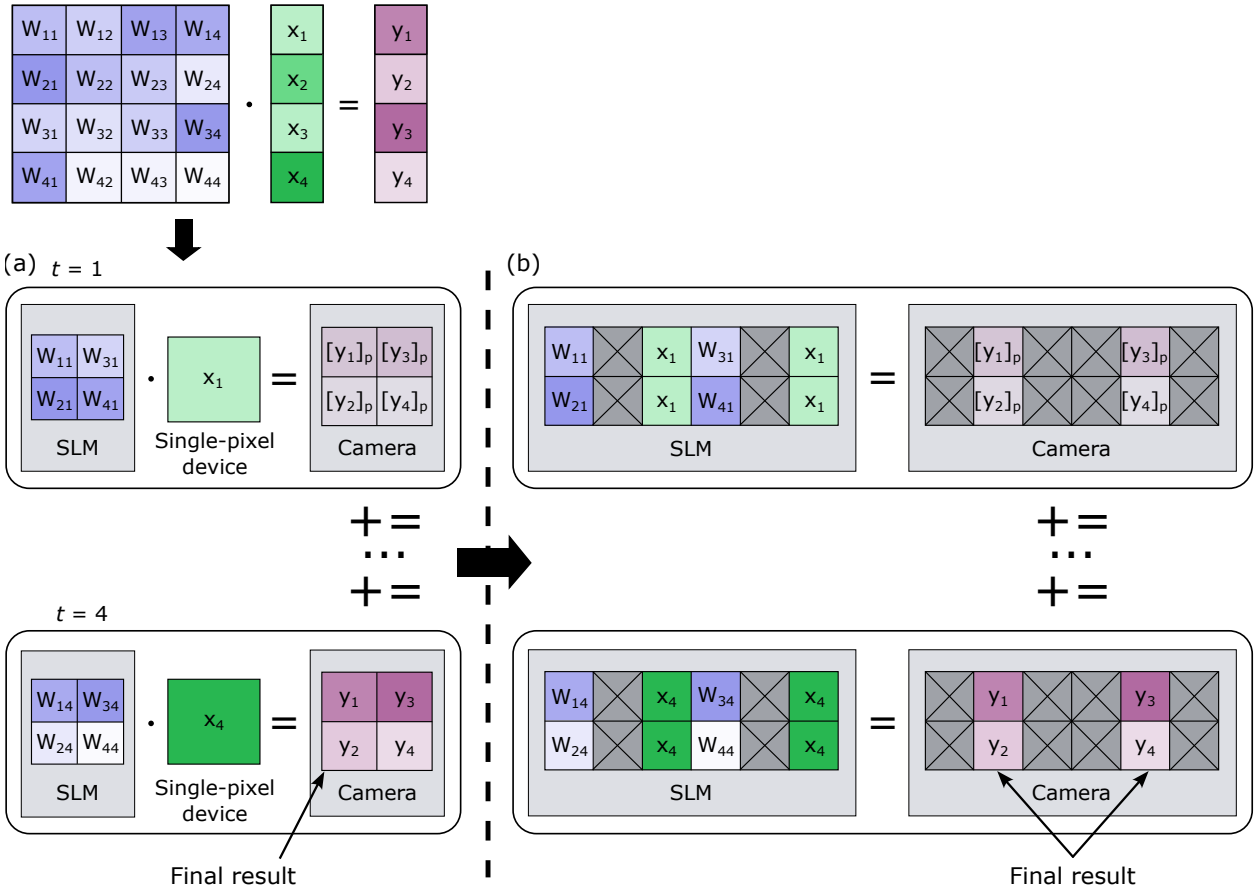

Figure A.7: Accumulation of partial products (denoted by subscript 'p') in HD-ONN. **a**, Matrix processing steps in full experiment. **b**, Simplified experiment where inputs and weights are displayed on the same SLM. Data from the greyed-out pixels are discarded. When the SLM is imaged to the camera, partial closure of the iris in the Fourier plane causes blurring, meaning that each weight overlaps (interferes) with the input in a middle pixel.

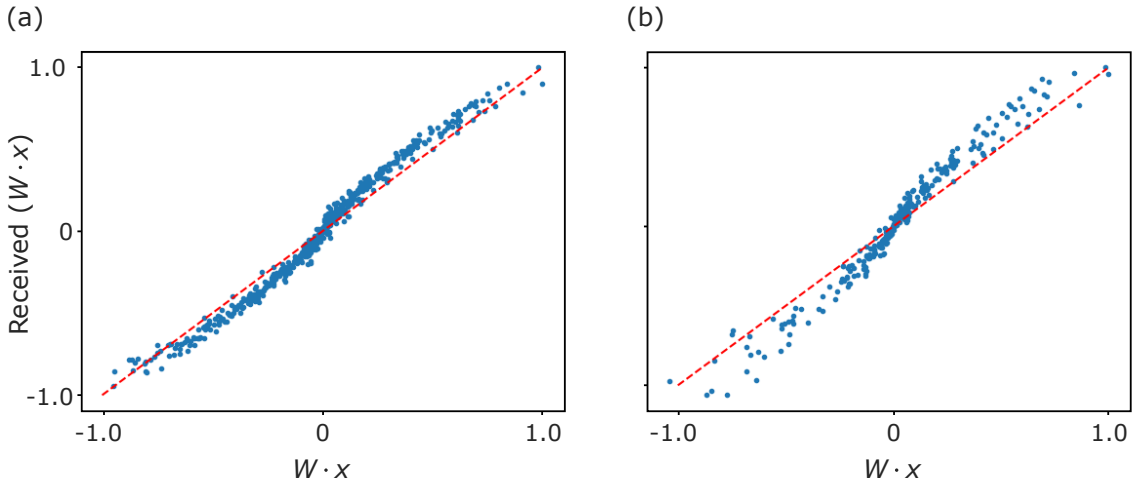

Figure A.8: Received element-wise products versus ground truth with simplified HD-ONN experiment with activations and weights displayed on the same LCoS SLM. **a**, 1 random weight and 1 random input activation displayed per time step for 500 time steps. **b**, 12 random weights and 1 random input activation displayed per time step for 20 time steps.

Future work

This work is not yet complete. The experimental challenges described above need to be addressed, notably path length stabilization and accuracy in computing element-wise products. To do so, a better model of the SLM's polarization response may be required. It may be worth attempting to run the scheme using the SLM in pure 'phase' mode, with the polarization of the light illuminating the SLM parallel to the extraordinary axis. Future steps can then include testing deep neural network inference at a large scale.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014. Preprint at: https://arxiv.org/abs/1409.1556.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, 2014. Preprint at: https://arxiv.org/abs/1409.4842.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[9] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710. DOI: 10.1109/CVPR.2018.00907.

[10] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional Transformers for language understanding*, 2018. Preprint at: https://arxiv.org/abs/1810.04805.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *Technical report, OpenAI Blog*, 2019. [Online]. Available: https://openai.com/blog/better-language-models.

[13] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, *Megatron-LM: Training multi-billion parameter language models using model parallelism*, 2019. Preprint at: https://arxiv.org/abs/1909.08053.

[14] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *ALBERT: A lite BERT for self-supervised learning of language representations*, 2019. Preprint at: https://arxiv.org/abs/1909.11942.

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2019. Preprint at: https://arxiv.org/abs/1910.10683.

[16] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. Preprint at: https://arxiv.org/abs/2005.14165.

[17] W. Fedus, B. Zoph, and N. Shazeer, *Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity*, 2021. Preprint at: https://arxiv.org/abs/2101.03961.

[18] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, *et al.*, *Using DeepSpeed and Megatron to train Megatron-Turing NLG 530b, the world's largest and most powerful generative language model*, 2022. Preprint at: https://arxiv.org/abs/2201.11990.

[19] OpenAI, *GPT-4 technical report*, 2023. Preprint at: https://arxiv.org/abs/2303.08774.

[20] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[21] G. E. Moore, "Progress in digital integrated electronics," *IEEE Electron Devices Meeting*, vol. 21, pp. 11–13, 1975.

[22] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of solid-state circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[23] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl, "There's plenty of room at the Top: What will drive computer performance after Moore's law?" *Science*, vol. 368, no. 6495, eaam9744, 2020. DOI: 10.1126/science.aam9744.

[24] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019. DOI: 10.1038/s41591-018-0316-z.

[25] OpenAI, *ChatGPT*, Accessed: 2023-10-27, 2023. [Online]. Available: https://chat.openai.com/.

[26] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, pp. 216–222, 2018. DOI: 10.1038/s41928-018-0059-3.

[27] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, *Scaling laws for neural language models*, 2020. Preprint at: https://arxiv.org/abs/2001.08361.

[28] J. McDonald, B. Li, N. Frey, D. Tiwari, V. Gadepally, and S. Samsi, "Great power, great responsibility: Recommendations for reducing energy for training language models," in *Findings of the Association for Computational Linguistics*, ser. Association for Computational Linguistics, Seattle, WA, USA, 2022, pp. 1962–1970. DOI: 10.18653/v1/2022.findings-naacl.151.

[29] M. Schreiner, *GPT-4 architecture, datasets, costs and more leaked*, Accessed: 2023-10-25. [Online]. Available: https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/.

[30] L. Bernstein, A. Sludds, R. Hamerly, V. Sze, J. Emer, and D. Englund, "Freely scalable and reconfigurable optical hardware for deep learning," *Scientific Reports*, vol. 11, no. 1, p. 3144, 2021. DOI: 10.1038/s41598-021-82543-3.

[31] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, *Carbon emissions and large neural network training*, 2021. Preprint at: https://arxiv.org/abs/2104.10350.

[32] N. Jouppi, G. Kurian, S. Li, *et al.*, "TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA, New York, NY, USA: Association for Computing Machinery, 2023, pp. 1–14. DOI: 10.1145/3579371.3589350.

[33] V. Sze, Y. Chen, T. Yang, and J. Emer, *Efficient Processing of Deep Neural Networks*. Morgan & Claypool, 2020. DOI: 10.2200/S01004ED1V01Y202004CAC050.

[34] R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-scale optical neural networks based on photoelectric multiplication," *Phys. Rev. X*, vol. 9, p. 021 032, 2019. DOI: 10.1103/PhysRevX.9.021032.

[35] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017. DOI: 10.1109/JPROC.2017.2761740.

[36] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A systematic approach to DNN accelerator evaluation," in *IEEE international symposium on performance analysis of systems and software (ISPASS)*, 2019, pp. 304–315. DOI: 10.1109/ISPASS.2019.00042.

[37] V. J. Reddi, C. Cheng, D. Kanter, *et al.*, *MLPerf inference benchmark*, 2020. Preprint at: https://arxiv.org/abs/1911.02549.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[39] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms*, 2017. Preprint at: https://arxiv.org/abs/1708.07747.

[40] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264.

[41] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg, *The Quick, Draw! AI experiment*, 2016. [Online]. Available: https://quickdraw.withgoogle.com/.

[42] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009, Technical report.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[44] J. Choquette, W. Gandhi, O. Giroux, N. Stam, and R. Krashinsky, "NVIDIA A100 tensor core GPU: Performance and innovation," *IEEE Micro*, vol. 41, no. 2, pp. 29–35, 2021. DOI: 10.1109/MM.2021.3061394.

[45] E. Nurvitadhi, G. Venkatesh, J. Sim, *et al.*, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, New York, NY, USA: Association for Computing Machinery, 2017, pp. 5–14. DOI: 10.1145/3020078.3021740.

[46] N. P. Jouppi, C. Young, N. Patil, *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12. DOI: 10.1145/3079856.3080246.

[47] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 187, pp. 1–30, 2018. DOI: 10.5555/3122009.3242044.

[48] 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*, vol. 577, no. 7792, pp. 641–646, 2020. DOI: 10.1038/s41586-020-1942-4.

[49] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Applied Physics Reviews*, vol. 7, no. 3, p. 031 301, 2020. DOI: 10.1063/1.5143815.

[50] Y. Shen, N. C. Harris, S. Skirlo, *et al.*, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics*, vol. 11, pp. 441–446, 2017. DOI: 10.1038/nphoton.2017.93.

[51] A. 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.*, vol. 7, no. 1, p. 7430, 2017. DOI: 10.1038/s41598-017-07754-z.

[52] J. Feldmann, N. Youngblood, M. Karpov, *et al.*, "Parallel convolutional processing using an integrated photonic tensor core," *Nature*, vol. 589, pp. 52–58, 2021. DOI: 10.1038/s41586-020-03070-1.

[53] S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, and D. Englund, *Single chip photonic deep neural network with accelerated training*, 2022. Preprint at: https://arxiv.org/abs/2208.01623.

[54] D. A. B. Miller, "Attojoule optoelectronics for low-energy information processing and communications," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 346–396, 2017. DOI: 10.1109/JLT.2017.2647779.

[55] S. Maktoobi, L. Froehly, L. Andreoli, X. Porte, M. Jacquot, L. Larger, and D. Brunner, "Diffractive coupling for photonic networks: How big can we go?" *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–8, 2019. DOI: 10.1109/JSTQE.2019.2930454.

[56] J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms," *Opt. Lett.*, vol. 2, no. 1, pp. 1–3, 1978. DOI: 10.1364/OL.2.000001.

[57] R. A. Athale and W. C. Collins, "Optical matrix–matrix multiplier based on outer product decomposition," *Applied Optics*, vol. 21, no. 12, pp. 2089–2090, 1982. DOI: 10.1364/AO.21.002089.

[58] D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Optics Letters*, vol. 10, no. 2, pp. 98–100, 1985. DOI: 10.1364/OL.10.000098.

[59] K. Wagner and D. Psaltis, "Multilayer optical learning networks," *Applied Optics*, vol. 26, no. 23, pp. 5061–5076, 1987. DOI: 10.1364/AO.26.005061.

[60] J. W. Goodman, "4 decades of optical information processing," *Optics and Photonics News*, vol. 2, no. 2, pp. 11–15, 1991. DOI: 10.1364/OPN.2.2.000011.

[61] T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon, "An optical neural network using less than 1 photon per multiplication," *Nature Communications*, vol. 13, p. 123, 1 2022. DOI: 10.1038/s41467-021-27774-8.

[62] J. Spall, X. Guo, and A. I. Lvovsky, "Hybrid training of optical neural networks," *Optica*, vol. 9, no. 7, pp. 803–811, 2022. DOI: 10.1364/OPTICA.456108.

[63] H. Zheng, Q. Liu, Y. Zhou, I. I. Kravchenko, Y. Huo, and J. Valentine, "Meta-optic accelerators for object classifiers," *Science Advances*, vol. 8, no. 30, eabo6410, 2022. DOI: 10.1126/sciadv.abo6410.

[64] T. Wang, M. M. Sohoni, L. G. Wright, M. M. Stein, S.-Y. Ma, T. Onodera, M. G. Anderson, and P. L. McMahon, "Image sensing with multilayer nonlinear optical neural networks," *Nature Photonics*, vol. 17, pp. 408–415, 2023. DOI: 10.1038/s41566-023-01170-8.

[65] L. Bottou, C. Cortes, J. S. Denker, *et al.*, "Comparison of classifier methods: A case study in handwriting digit recognition," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Jerusalem, Israel: IEEE, 1994, pp. 77–82. DOI: 10.1109/ICPR.1994.576879.

[66] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nature Photonics*, vol. 15, no. 5, pp. 367–373, 2021. DOI: 10.1038/s41566-021-00796-w.

[67] M. Miscuglio, Z. Hu, S. Li, J. K. George, R. Capanna, H. Dalir, P. M. Bardet, P. Gupta, and V. J. Sorger, "Massively parallel amplitude-only Fourier neural network," *Optica*, vol. 7, no. 12, pp. 1812–1819, 2020. DOI: 10.1364/OPTICA.408659.

[68] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018. DOI: 10.1126/science.aat8084.

[69] Texas Instruments, *DLP products*, Accessed: 2023-11-16. [Online]. Available: https://ti.com/dlp-chip/overview.html.

[70] G. Gauthier, I. Lenton, N. M. Parry, M. Baker, M. J. Davis, H. Rubinsztein-Dunlop, and T. W. Neely, "Direct imaging of a digital-micromirror device for configurable microscopic optical potentials," *Optica*, vol. 3, no. 10, pp. 1136–1143, 2016. DOI: 10.1364/OPTICA.3.001136.

[71] B. Lee, D. Yoo, J. Jeong, S. Lee, D. Lee, and B. Lee, "Wide-angle speckleless DMD holographic display using structured illumination with temporal multiplexing," *Optics Letters*, vol. 45, no. 8, p. 2148, 2020. DOI: 10.1364/OL.390552.

[72] C. Rosales-Guzmán and A. Forbes, *How to shape light with spatial light modulators*. Bellingham, WA, USA: SPIE Press, 2017. DOI: 10.1117/3.2281295.

[73] A. Skalli, J. Robertson, D. Owen-Newns, M. Hejda, X. Porte, S. Reitzenstein, A. Hurtado, and D. Brunner, "Photonic neuromorphic computing using vertical cavity semiconductor lasers," *Optical Materials Express*, vol. 12, no. 6, pp. 2395–2414, 2022. DOI: 10.1364/OME.450926.

[74] M. S. Wong, S. Nakamura, and S. P. DenBaars, "Progress in high performance III-nitride micro-light-emitting diodes," *ECS Journal of Solid State Science and Technology*, vol. 9, no. 1, p. 015 012, 2019. DOI: 10.1149/2.0302001JSS.

[75] N. B. Hassan, F. Dehkhoda, E. Xie, J. Herrnsdorf, M. J. Strain, R. Henderson, and M. D. Dawson, "Ultrahigh frame rate digital light projector using chip-scale LED-on-CMOS technology," *Photonics Research*, vol. 10, no. 10, pp. 2434–2446, 2022. DOI: 10.1364/PRJ.455574.

[76] T. Heuser, M. Pflüger, I. Fischer, J. A. Lott, D. Brunner, and S. Reitzenstein, "Developing a photonic hardware platform for brain-inspired computing based on 5×5 VCSEL arrays," *Journal of Physics: Photonics*, vol. 2, no. 4, p. 044 002, 2020. DOI: 10.1088/2515-7647/aba671.

[77] A. Krishnamoorthy, K. Goossen, L. Chirovsky, R. Rozier, P. Chandramani, S. Hui, J. Lopata, J. Walker, and L. D'Asaro, "16 × 16 VCSEL array flip-chip bonded to CMOS VLSI circuit," *IEEE Photonics Technology Letters*, vol. 12, no. 8, pp. 1073–1075, 2000. DOI: 10.1109/68.868012.

[78] Z. Chen, S. Yan, and C. Danesh, "MicroLED technologies and applications: Characteristics, fabrication, progress, and challenges," *Journal of Physics D: Applied Physics*, vol. 54, no. 12, p. 123 001, 2021. DOI: 10.1088/1361-6463/abcfe4.

[79] C. L. Panuski, I. Christen, M. Minkov, *et al.*, "A full degree-of-freedom spatiotemporal light modulator," *Nature Photonics*, vol. 16, no. 12, pp. 834–842, 2022. DOI: 10.1038/s41566-022-01086-9.

[80] Z. Chen, A. Sludds, R. Davis III, *et al.*, "Deep learning with coherent VCSEL neural networks," *Nature Photonics*, vol. 17, no. 8, pp. 723–730, 2023. DOI: 10.1038/s41566-023-01233-w.

[81] H. A. Haus, *Waves and fields in optoelectronics*. Prentice-Hall, 1984.

[82] D. Bluvstein, H. Levine, G. Semeghini, *et al.*, "A quantum processor based on coherent transport of entangled atom arrays," *Nature*, vol. 604, no. 7906, pp. 451–456, 2022. DOI: 10.1038/s41586-022-04592-6.

[83] D. Pavlov, S. Gurbatov, S. I. Kudryashov, P. A. Danilov, A. P. Porfirev, S. N. Khonina, O. B. Vitrik, S. A. Kulinich, M. Lapine, and A. A. Kuchmizhak, "10-million-elements-per-second printing of infrared-resonant plasmonic arrays by multiplexed laser pulses," *Optics Letters*, vol. 44, no. 2, pp. 283–286, 2019. DOI: 10.1364/OL.44.000283.

[84] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.

[85] D. Kim, A. Keesling, A. Omran, H. Levine, H. Bernien, M. Greiner, M. D. Lukin, and D. R. Englund, "Large-scale uniform optical focus array generation with a phase spatial light modulator," *Opt. Lett.*, vol. 44, no. 12, pp. 3178–3181, 2019. DOI: 10.1364/OL.44.003178.

[86] Y. Zhang, J. B. Chou, J. Li, *et al.*, "Broadband transparent optical phase change materials for high-performance nonvolatile photonics," *Nature Communications*, vol. 10, no. 1, p. 4279, 2019. DOI: 10.1038/s41467-019-12196-4.

[87] H. Zimmermann, A. Marchlewski, W. Gaberl, I. Jonak-Auer, G. Meinhardt, and E. Wachmann, "Blue-enhanced PIN finger photodiodes in a 0.35-µm SiGe BiCMOS technology," *IEEE Photonics Technology Letters*, vol. 21, no. 22, pp. 1656–1658, 2009. DOI: 10.1109/LPT.2009.2031245.

[88] S. Latif, S. E. Kocabas, L. Tang, C. Debaes, and D. A. B. Miller, "Low capacitance CMOS silicon photodetectors for optical clock injection," *Applied Physics A*, vol. 95, no. 4, pp. 1129–1135, 2009. DOI: 10.1007/s00339-009-5122-5.

[89] Y. Gao, H. Cansizoglu, K. G. Polat, *et al.*, "Photon-trapping microstructures enable high-speed high-efficiency silicon photodiodes," *Nature Photonics*, vol. 11, no. 5, pp. 301–308, 2017. DOI: 10.1038/nphoton.2017.37.

[90] B. Fahs, A. J. Chowdhury, Y. Zhang, J. Ghasemi, C. Hitchcock, P. Zarkesh-Ha, and M. M. Hella, "Design and modeling of blue-enhanced and bandwidth-extended PN photodiode in standard CMOS technology," *IEEE Transactions on Electron Devices*, vol. 64, no. 7, pp. 2859–2866, 2017. DOI: 10.1109/TED.2017.2700389.

[91] N. Mehta, C. Sun, M. Wade, S. Lin, M. Popovic, and V. Stojanović, "A 12Gb/s, 8.6µApp input sensitivity, monolithic-integrated fully differential optical receiver in CMOS 45nm SOI process," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, IEEE, Lausanne, Switzerland, 2016, pp. 491–494. DOI: 10.1109/ESSCIRC.2016.7598348.

[92] A. V. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. E. Cunningham, "Computer systems based on silicon photonic interconnects," *Proceedings of the IEEE*, vol. 97, no. 7, pp. 1337–1361, 2009. DOI: 10.1109/JPROC.2009.2020712.

[93] N. Mehta, S. Lin, B. Yin, S. Moazeni, and V. Stojanović, "A laser-forwarded coherent transceiver in 45-nm SOI CMOS using monolithic microring resonators," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 1096–1107, 2020. DOI: 10.1109/JSSC.2020.2968764.

[94] J. Xue, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, P. Liu, M. Huang, *et al.*, "An intra-chip free-space optical interconnect," in *37th Int. Symp. Computer Architecture (ISCA)*, Saint-Malo, France, 2010, pp. 94–105. DOI: 10.1145/1816038.1815975.

[95] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer, "FireFly: A reconfigurable wireless data center fabric using free-space optics," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 319–330. DOI: 10.1145/2619239.2626328.

[96] J. Bao, D. Dong, B. Zhao, Z. Luo, C. Wu, and Z. Gong, "FlyCast: Free-space optics accelerating multicast communications in physical layer," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 97–98, 2015. DOI: 10.1145/2829988.2790002.

[97] J. Fowers, K. Ovtcharov, M. Papamichael, *et al.*, "A configurable cloud-scale DNN processor for real-time AI," in *ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018, pp. 1–14. DOI: 10.1109/ISCA.2018.00012.

[98] Y. S. Shao, J. Clemons, R. Venkatesan, *et al.*, "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, Columbus, OH, USA: Association for Computing Machinery, 2019, pp. 14–27. DOI: 10.1145/3352460.3358302.

[99] J. Yin, Z. Lin, O. Kayiran, M. Poremba, M. Shoaib Bin Altaf, N. Enright Jerger, and G. H. Loh, "Modular routing design for chiplet-based systems," in *ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018, pp. 726–738. DOI: 10.1109/ISCA.2018.00066.

[100] A. Samajdar, J. M. Joseph, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, "A systematic methodology for characterizing scalability of DNN accelerators using SCALE-Sim," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Boston, MA, USA, 2020, pp. 58–68. DOI: 10.1109/ISPASS48437.2020.00016.

[101] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, 2011. DOI: 10.1109/MM.2011.89.

[102] W. J. Dally, C. T. Gray, J. Poulton, B. Khailany, J. Wilson, and L. Dennison, "Hardware-enabled artificial intelligence," in *IEEE Symposium on VLSI Circuits*, Honolulu, HI, USA, 2018, pp. 3–6. DOI: 10.1109/VLSIC.2018.8502368.

[103] Zheng, P., Connelly, D., Ding, F., and Liu, T.-J. K., "FinFET evolution toward stacked-nanowire FET for CMOS technology scaling," *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 3945–3950, 2015. DOI: 10.1109/TED.2015.2487367.

[104] K. Iga, "Vertical-cavity surface-emitting laser: Its conception and evolution," *Japanese Journal of Applied Physics*, vol. 47, no. 1R, pp. 1–10, 2008. DOI: 10.1143/JJAP.47.1.

[105] R. Jäger, M. Grabherr, C. Jung, R. Michalzik, G. Reiner, B. Weigl, and K. J. Ebeling, "57% wallplug efficiency oxide-confined 850 nm wavelength GaAs VCSELs," *Electronics Letters*, vol. 33, no. 4, pp. 330–331, 1997. DOI: 10.1049/el:19970193.

[106] C. Chao and B. Saeta, "Cloud TPU: Codesigning architecture and infrastructure," in *Hot Chips*, vol. 31, 2019.

[107] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, 2017. DOI: 10.1016/j.vlsi.2017.02.002.

[108] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, IEEE, 2014, pp. 10–14. DOI: 10.1109/ISSCC.2014.6757323.

[109] H. Saadat, H. Bokhari, and S. Parameswaran, "Minimally biased multipliers for approximate integer and floating-point multiplication," *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2623–2635, 2018. DOI: 10.1109/TCAD.2018.2857262.

[110] M. Shoba and R. Nakkeeran, "Energy and area efficient hierarchy multiplier architecture based on Vedic mathematics and GDI logic," *Engineering Science and Technology, an International Journal*, vol. 20, no. 1, pp. 321–331, 2017. DOI: 10.1016/j.jestch.2016.06.007.

[111] S. Ravi, A. Patel, M. Shabaz, P. M. Chaniyara, and H. M. Kittur, "Design of low-power multiplier using UCSLA technique," in *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, New Delhi, 2015, pp. 119–126. DOI: 10.1007/978-81-322-2135-7_14.

[112] J. Johnson, *Rethinking floating point for deep learning*, 2018. Preprint at: https://arxiv.org/abs/1811.01721.

[113] G. A. Keeler, D. Agarwal, C. Debaes, B. E. Nelson, N. C. Helman, H. Thienpont, and D. A. Miller, "Optical pump-probe measurements of the latency of silicon CMOS optical interconnects," *IEEE Photonics Technology Letters*, vol. 14, no. 8, pp. 1214–1216, 2002. DOI: 10.1109/LPT.2002.1022022.

[114] L. Tang, S. E. Kocabas, S. Latif, A. K. Okyay, D.-S. Ly-Gagnon, K. C. Saraswat, and D. A. B. Miller, "Nanometre-scale germanium photodetector enhanced by a near-infrared dipole antenna," *Nature Photonics*, vol. 2, no. 4, pp. 226–229, 2008. DOI: 10.1038/nphoton.2008.30.

[115] Thorlabs, *High-speed fiber-coupled detectors*, Accessed: 2023-10-25, 2020. [Online]. Available: https://thorlabs.com/newgrouppage9.cfm?objectgroup_id=1297&pn=DET02AFC.

[116] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021. DOI: 10.1109/TNNLS.2020.2978386.

[117] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2022. DOI: 10.1109/TKDE.2020.2981333.

[118] L. Bernstein, A. Sludds, C. Panuski, S. Trajtenberg-Mills, R. Hamerly, and D. Englund, "Single-shot optical neural network," *Science Advances*, vol. 9, no. 25, eadg7904, 2023. DOI: 10.1126/sciadv.adg7904.

[119] E. A. Huerta, G. Allen, I. Andreoni, *et al.*, "Enabling real-time multi-messenger astrophysics discoveries with deep learning," *Nature Reviews Physics*, vol. 1, no. 10, pp. 600–608, 2019. DOI: 10.1038/s42254-019-0097-4.

[120] J. Degrave, F. Felici, J. Buchli, *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022. DOI: 10.1038/s41586-021-04301-9.

[121] B. Saleh and M. Teich, *Fundamentals of Photonics*, Second Edition. Hoboken, New Jersey: John Wiley & Sons, 2007.

[122] G. H. Li, R. Sekine, R. Nehra, R. M. Gray, L. Ledezma, Q. Guo, and A. Marandi, "All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning," *Nanophotonics*, vol. 12, no. 5, pp. 847–855, 2022. DOI: 10.1515/nanoph-2022-0137.

[123] M. Burla, C. Hoessbacher, W. Heni, *et al.*, "500 GHz plasmonic Mach-Zehnder modulator enabling sub-THz microwave photonics," *APL Photonics*, vol. 4, no. 5, p. 056 106, 2019. DOI: 10.1063/1.5086868.

[124] C. Han, Z. Zheng, H. Shu, *et al.*, *Slow light silicon modulator beyond 110 GHz bandwidth*, 2023. Preprint at: https://arxiv.org/abs/2302.03652.

[125] P. Kharel, C. Reimer, K. Luke, L. He, and M. Zhang, "Breaking voltage-bandwidth limits in integrated lithium niobate modulators using micro-structured electrodes," *Optica*, vol. 8, no. 3, pp. 357–363, 2021. DOI: 10.1364/OPTICA.416155.

[126] M. Kuramoto, S. Kobayashi, T. Akagi, K. Tazawa, K. Tanaka, T. Saito, and T. Takeuchi, "High-output-power and high-temperature operation of blue GaN-based vertical-cavity surface-emitting laser," *Applied Physics Express*, vol. 11, no. 11, p. 112 101, 2018. DOI: 10.7567/APEX.11.112101.

[127] B. E. Jonsson, "An empirical approach to finding energy efficient ADC architectures," in *Proc. of 2011 IMEKO IWADC & IEEE ADC Forum*, Orvieto, Italy, 2011, pp. 132–137.

[128] V. Tripathi and B. Murmann, "An 8-bit 450-MS/s single-bit/cycle SAR ADC in 65-nm CMOS," in *Proceedings of the ESSCIRC*, IEEE, Bucharest, Romania, 2013, pp. 117–120. DOI: 10.1109/ESSCIRC.2013.6649086.

[129] O. Morales Chacón, J. J. Wikner, C. Svensson, L. Siek, and A. Alvandpour, "Analysis of energy consumption bounds in CMOS current-steering digital-to-analog converters," *Analog Integrated Circuits and Signal Processing*, vol. 111, pp. 339–351, 2022. DOI: 10.1007/s10470-022-02013-2.

[130] R. Di Leonardo, F. Ianni, and G. Ruocco, "Computer generation of optimal holograms for optical trap arrays," *Optics Express*, vol. 15, no. 4, pp. 1913–1922, 2007. DOI: 10.1364/OE.15.001913.

[131] HOLOEYE Photonics AG, *GAEA-2 10 megapixel phase only LCOS-SLM*, Accessed: 2024-01-09. [Online]. Available: https://holoeye.com/products/spatial-light-modulators/gaea-2-phase-only/.

[132] L. Kull, D. Luu, C. Menolfi, *et al.*, "A 10b 1.5 GS/s pipelined-SAR ADC with background second-stage common-mode regulation and offset calibration in 14nm CMOS FinFET," in *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2017, pp. 474–475. DOI: 10.1109/ISSCC.2017.7870467.

[133] N. V. Vijaya Krishna Boppana and S. Ren, "A low-power and area-efficient 64-bit digital comparator," *Journal of Circuits, Systems and Computers*, vol. 25, no. 12, p. 1 650 148, 2016. DOI: 10.1142/S0218126616501486.

[134] M. Kuramoto, S. Kobayashi, T. Akagi, K. Tazawa, K. Tanaka, K. Nakata, and T. Saito, "Watt-class blue vertical-cavity surface-emitting laser arrays," *Applied Physics Express*, vol. 12, no. 9, p. 091 004, 2019. DOI: 10.7567/1882-0786/ab3aa6.

[135] W. Dong, H. Liu, J. K. Behera, L. Lu, R. J. H. Ng, K. V. Sreekanth, X. Zhou, J. K. W. Yang, and R. E. Simpson, "Wide bandgap phase change material tuned visible photonics," *Advanced Functional Materials*, vol. 29, no. 6, p. 1 806 181, 2019. DOI: 10.1002/adfm.201806181.

[136] Y. Jung, H. Han, A. Sharma, J. Jeong, S. S. Parkin, and J. K. Poon, "Integrated hybrid VO_2–silicon optical memory," *ACS Photonics*, vol. 9, no. 1, pp. 217–223, 2022. DOI: 10.1021/acsphotonics.1c01410.

[137] Y. Zhang, C. Fowler, J. Liang, *et al.*, "Electrically reconfigurable non-volatile metasurface using low-loss optical phase-change material," *Nature Nanotechnology*, vol. 16, no. 6, pp. 661–666, 2021. DOI: 10.1038/s41565-021-00881-9.

[138] N. Quack, H. Sattari, A. Y. Takabayashi, Y. Zhang, P. Verheyen, W. Bogaerts, P. Edinger, C. Errando-Herranz, and K. B. Gylfason, "MEMS-enabled silicon photonic integrated devices and circuits," *IEEE Journal of Quantum Electronics*, vol. 56, no. 1, pp. 1–10, 2020. DOI: 10.1109/JQE.2019.2946841.

[139] G. Yeap, S. S. Lin, Y. M. Chen, *et al.*, "5 nm CMOS production technology platform featuring full-fledged EUV, and high mobility channel FinFETs with densest 0.021 μm^2 SRAM cells for mobile SoC and high performance computing applications," in *International Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA, USA, 2019, pp. 36.7.1–36.7.4. DOI: 10.1109/IEDM19573.2019.8993577.

[140] A. Sludds, S. Bandyopadhyay, Z. Chen, *et al.*, "Delocalized photonic deep learning on the internet's edge," *Science*, vol. 378, no. 6617, pp. 270–276, 2022. DOI: 10.1126/science.abq8271.