PHYSICAL NEURAL NETWORKS USING ACOUSTICS AND PHOTONICS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Martin Stein

August 2024

PHYSICAL NEURAL NETWORKS
USING ACOUSTICS AND PHOTONICS

Martin Stein, Ph.D.
Cornell University 2024

This thesis explores alternatives to the currently dominant approach of simulating artificial deep neural networks on digital electronic hardware. There is no serious alternative to computing with digital electronics at the moment, but no physical law singles it out as the superior approach. Strong arguments can be made for approaches using other physical processes such as optics or electro-chemistry.

We introduce *physical neural networks* as an alternative to digital electronic simulation of neural networks. In analogy to deep neural networks using layers of fine-tuned mathematical transformations, physical neural networks harness the controllable transformations of layers of physical systems for computation. As a proof-of-concept, we trained a neural network that uses acoustic signals transformed by a vibrating metal plate mounted on an audio speaker to perform image classification. We also present the results from an optical and analog electronic system that experimentally perform audio and image classification. Physical neural networks may ultimately perform machine learning orders-of-magnitude faster and more energy-efficiently than digital electronic processors.

We then turn to a photonic implementation of a physical neural network. On-chip photonic neural-network processors have potential benefits in both speed and energy efficiency but current approaches cannot reach the scale at which they can compete with digital electronic processors. Physics-aware training enabled us to pursue a different approach for on-chip photonic-neural-network processors in which the computation is performed by freely propagating waves in two dimensions. We propose and demonstrate a device whose refractive index as a function of space, $n(x, z)$, can be rapidly reprogrammed, allowing arbitrary control over the wave propagation in the device. We used a prototype device with a functional area

of 12 mm^2 to perform neural-network inference with up to 49-dimensional input vectors in a single pass, achieving 96% accuracy on vowel classification and 86% accuracy on MNIST handwritten-digit classification, with no trained digital-electronic pre- or post-processing. This is a scale beyond that of previous photonic chips relying on discrete components, illustrating the benefit of the continuous-waves paradigm.

In principle, with large enough chip area, the reprogrammability of the device's refractive index distribution enables the reconfigurable realization of any passive, linear photonic circuit or device. This promises the development of more compact and versatile photonic systems for a wide range of applications, including optical processing, smart sensing, spectroscopy, and optical communications.

BIOGRAPHICAL SKETCH

Martin Stein was born in Germany in 1994. He received his Bachelor of Science in Physics from Heidelberg University, Germany, in 2017 under direction of Stefan Flörchinger. After a stint at the Max-Plack Institute for Plasma Physics in Greifswald, Germany, he moved to Cornell University as an exchange student in 2017. He remained at Cornell University and received his Master of Science in Physics in 2019 under the direction of Natasha Holmes and continued with his Ph.D. research at the intersection of photonics and machine learning under the direction of Peter McMahon.

Dedicated to my parents.

ACKNOWLEDGEMENTS

I am indebted to Peter McMahon, Tatsuhiro Onodera, Logan Wright, and Tianyu Wang for their invaluable mentorship over the course of my Ph.D. I also want to thank Mandar Sohoni, Ryotatsu Yanagimoto, Federico Presutti, Shiyuan Ma, Alen Senanian, Maxwell Anderson, and Benjamin Ash for their countless contributions to this thesis.

I want to thank Natasha Holmes and Emily Smith for their formative guidance in the first two years of my Ph.D. During those years, I also worked closely with Cole Walsh, Ryan Tapping, and Yasemin Kalender. I am deeply grateful for their contributions to my Ph.D work and my development as a scholar.

I further want to thank Mark Lory-Moran, Jenny Wurster, Peter Lepage, Rene Kizilcec, Ben Zwickl, Gina Passante, Erich Mueller, all the members of the Cornell Physics Education Research Laboratory and the McMahon lab for their support during my Ph.D.

Lastly, I want to thank my parents, my brother, my partner Cindy, her parents, and all the other friends and family members who have supported me in myriad ways over the past seven years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Over the past 50 years the world has experienced unprecedented change brought on by the development and availability of digital electronic computers. Never before has a technology sustained such fast exponential growth for so long. Exponential growth is a process brought on by self-reinforcing feedback cycles: More of something begets even more growth of the same thing. But exponential growth is eventually always stopped by limited resources. For electronic computers, this favorable feedback cycle of technological and economic forces has been sustained for many orders of magnitudes, for example, in increasing the density of transistors from approximately $1/\text{cm}^2$ to more than $10^{10}/\text{cm}^2$. Soon, the transistor density will stop growing exponentially, limited by the size of atoms. This will end what is called *Moore's law*. Other crucial growth processes in electronics have already stopped, such as the increase of processor clock rate or Dennard scaling [1].

This alone is a compelling reason to research the viability of computing platforms other than conventional digital electronics. Another trend adds legitimacy to research on unconventional computing: The changing nature of the algorithms executed on computers favors a change of the underlying hardware (and vice versa). Computing has long been dominated by the serial execution of logic operations, championed by central processing units (CPUs). Large-scale changes to the demand on computations is bringing about the rise of alternative computing hardware. Over the past two decades, the "connectionism" approach to modeling neural networks has become extremely popular, owing to its surprising success in fields like image- or natural-language-processing, game-playing, or scientific computing. For such models, computers imitate cognitive functions by simulating the low-level structure of the brain: Neurons, densely connected to each other via synapses. Such simulation heavily relies on linear algebra to model the synaptic connections, for which other types of hardware is better suited than CPUs. The connectionism approach has been so succesful that the size of the models simulated has grown at a pace much faster than even the development of digital electronic computing hardware. Over the past ten years, the number of synapses

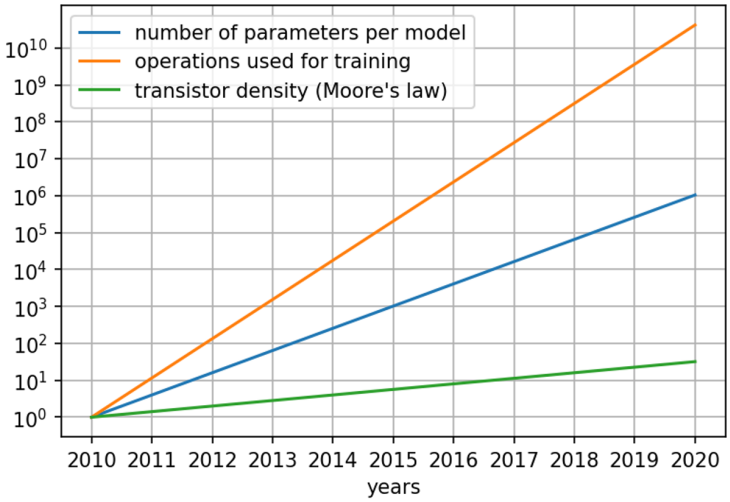

Figure 1.1: **Trends in computer hardware and machine learning.** Shown are the trendlines of three important metrics, normalized to 1 in 2010. Green: The transistor density on state-of-the-art electronic microprocessors (trendline from [1, 2]. Blue: The number of trainable parameters of state-of-the-art machine learning models (trendline from [3]. Orange: The number of floating point operations used to train the largest state-of-the-art models (trendline from [4, 5]). Between 2010, the transistor density has increased approximately 30-fold, the number of parameters per model has increased million-fold, and the number of operations used during training has increased more than ten-billion times.

(parameters) simulated in state-of-the art neural network models has grown more than a million times [4]. This fast-paced growth has made further developments of artificial neural networks enormously expensive and a burden on already strained CO_2 budgets [6].

About 10 years ago, Krizhevsky et al [7] demonstrated to large appeal that it is possible to simulate artificial neural networks efficiently on graphics processing units (GPUs) instead. GPUs were initially developed as specialized electronic circuits to efficiently simulate the name-giving computer graphics, such as shading of 3D objects, which also happen to involve expensive linear algebra operations. The realization that GPUs were useful in applications beyond computer graphics led to much investment in the development of GPUs, and also the applications that GPUs helped simulate. More broadly, the available hardware has shaped the adaption of algorithms [7–9] and the use of algorithms has spurred further development of hardware. Yet, GPUs are likely not the end of all for hardware imitating neural networks. Take for example the development of tensor-processing unit (TPUs) [10], analog electronic

and optical accelerators [11, 12], or the observation that the human brain can perform many tasks while consuming much less energy than state-of-the-art artificial neural networks [13].

In this light, research on unconventional computing platforms can lead to increased speed and energy-efficiency of neuromorphic computations, or even pave the way for new algorithms to take off, as GPUs did for the current linear algebra-dominated artificial neural networks. The work of my thesis is largely motivated by these considerations.

In **Chapter 2**, I review the basic structure of modern artificial neural networks, as commonly realized on digital electronic computers. I then introduce *physical neural networks*. These are analog computers which mimick the structure of of artificial neural networks. Instead of running on digital electronic computers, the networks "run on" virtually arbitrary, interconnected, controllable physical systems. This unusual choice of hardware is motivated by similarities between the transformations performed on inputs to many physical systems and inputs to artificial neural networks. Hopefully such computers can one day benefit from the astonishing energy and speed benefit one can gain from directly observing the dynamics of physical systems rather than simulating them on a digital electronic computer. I also introduce the algorithm, *physics-aware training*, that makes it possible to apply the back-propagation algorithm to physical systems, thereby enabling efficient and accurate training. I show the application of these ideas to a physical neural network made from an audio speaker and microphone, as well as an electrical and optical system.

In **Chapter 3**, I will highlight the advantages that analog optical computing, particularly on-chip, can bestow on neural network accelerators. I review the state-of-the-art of on-chip optical neural network processors and re-derive foundational equations describing the propagation and manipulation of light on on-chip photonic devices.

In **Chapter 4**, I report on the centerpiece of my Ph.D. work, the realization of an on-chip photonic processor capable of creating arbitrary two-dimensional refractive index profiles. Such a device has a unique advantage in space-efficiency over other on-chip optical neural networks, by controlling multimode wave propagation. We tune the refractive index profile

of the device in situ using the physics-aware training algorithm to perform machine learning tasks, specifically audio and image classification with up to 50-dimensional inputs.

Chapter 5 contains important additional material on the device presented in the previous chapter, such as details on design choices, its fabrication process, and training algorithm.

In **Chapter 6**, I discuss future research directions on physical neural networks, and their photonic implementation.

I also include two chapters related to research I conducted in the first two years of my Ph.D. on physics education. **Chapter 7** presents a study observing a curious effect in introductory physics labs. Many institutions are changing the focus of their introductory physics labs from verifying physics content towards teaching students about the skills and nature of science. As instruction shifts, so too should the ways students approach and behave in the labs. In this study, we evaluated students' lab notes from an early activity in an experimentation-focused lab course. We found that about 30% of student groups (out of 107 groups at three institutions) recorded questionable research practices in their lab notes, such as subjective interpretations of results or manipulating equipment and data. The large majority of these practices were associated with confirmatory goals, which we suspect stem from students' prior exposure to verification labs.

Chapter 8 is a reflection on the disparities I have observed between what students experience in introductory physics labs and what I experienced as an applied physicist.

CHAPTER 2

DEEP PHYSICAL NEURAL NETWORKS TRAINED WITH BACKPROPAGATION

Much of the work presented in this chapter was published in Wright, L. G. *et al.* Deep physical neural networks trained with backpropagation. *Nature* **601** (2022), of which I was a co-author. I have reprinted the figures of the publication, their captions, and the parts of the manuscript that I mainly wrote myself. I also wrote a popular science article, Stein, M. Making nature compute for us. *TheScienceBreaker* **9** (2023), which I have reprinted as a popular introduction to this chapter (Sec. 2.1). The remaining text is original and presents the findings of our paper [14] in my words.

2.1 Popular summary

Models of neural networks are used ubiquitously in science and technology these days. Such artificial neural networks are computer algorithms inspired by the way the human brain works. While they enable revolutionary technology, their execution on digital computers eats up increasingly large amounts of energy and time, hampering progress on understanding and improving these algorithms. Yet before digital computers were cheap and widely available, engineers and scientists mostly used a different kind of computer, which scientists are revisiting in light of ever-more energy-hungry artificial neural networks.

To simulate complex phenomena like the erosion of riverbeds or the flight of an airplane, small models of these systems were built in the lab, like hydraulic flumes or wind-tunnels. The advantage is that once such a system is set up, it is incredibly efficient at simulating what it was built for: E.g., in a hydraulic flume, water flows down the artificial riverbed and erosion just happens. Such "computers" are sometimes called analog computers, not just because they are not digital, but because they rely on a mathematical analogy between the simulated and the simulating system. In contrast, simulations on digital supercomputers

cost plenty of energy and can take anywhere from a few minutes to several months. Yet, the flexibility that digital computers offer over analog computers almost always tips the scale in their favor. To get reliable answers from an analog computer, much effort is put into engineering the analogy to high accuracy. On a digital computer, programmers can do that on the fly.

Given their ubiquitous use, a fast and energy-efficient analog model of a neural network could be widely applicable, though. In our work, we set out to build the equivalent of a wind tunnel for an artificial neural network and demonstrate a general way for scientists to construct such models.

A surprising finding that the field of artificial intelligence is thriving on is that even complex tasks can be broken down into many automatically trained simple operations. For the brain to tell whether it is looking at a cat or a dog, attention is paid to characteristic features, while irrelevant information like a sofa in the background is blinded out. Artificial neural networks realize this process by propagating information through various layers of abstraction. Each layer only does very simple operations, e.g. recognizing an oval as a head, line-like objects as limbs, etc. However, a sequence of many such simple operations can solve complex tasks.

We found that almost any physical system can perform these simple operations if there is a way to send data in and out of the system. For example, we tried this on a loudspeaker: Data can be injected in the form of electrical signals, just like when playing music. The loudspeaker provides complex oscillations of the membrane—essentially amplifying some sound frequencies and attenuating others. Data can be read out of the system by recording the sound with a microphone. The attenuation and amplification of features is an operation that can be used for neural network computations. We demonstrated that a loudspeaker, a hobbyist electrical breadboard, and an optical laser system can each help perform neural computations in this way. We don't need to precisely engineer an analogy; many physical systems are a good analog model for a neural network! Of course, many other physical systems are not—an important research question is to determine which systems are the best

choice.

The technical breakthrough that allowed us to turn these systems into physical neural networks was the development of an algorithm that automatically trains controllable parameters of a physical system, such as its shape, the forces applied to it, or a control signal. The algorithm can simultaneously train many physical parameters in a network of connected physical systems. This allows various physical systems to cooperate and break down a complicated task into many successive simple operations, just as in modern artificial neural networks.

Such physical neural networks offer some intriguing possibilities for scientists in the near term. So far, scientists have been restricted to a very narrow class of physical systems with which to build computers for neural networks. Our work hugely broadens that class to systems that were never thought of as computers, including some that could be much faster and more energy-efficient than current computers. Physical neural networks can also directly interact with the physical world to make, for instance, cameras that work more like our eyes, microphones that work more like our ears, etc.

Different computing paradigms besides electronic digital computing are poised to undergo a renaissance with the rise of neuromorphic computing. Our brains are living proof that highly efficient analog computers operating beyond the capabilities of modern digital computers must be possible. We hope our work is an inspiration for scientists working in this direction.

2.2 Introduction and Background on Physical Neural Networks (PNNs)

Deep learning has become a ubiquitous tool in science and technology. The explosive growth of deep learning models has far outpaced the development of hardware, which is now limiting the pace of further developments [6]. This trend has created a flourishing scientific field exploring the possibilities of unconventional computing substrates [16], such as optics [12,

17, 18] or memristor crossbar arrays [19]. A common approach is to engineer physical systems to act as neural network accelerators by performing parts of the computations of a neural network faster and/or more energy-efficient than a digital electronic computer [11, 16]. The challenge usually lies in carefully crafting a mathematical isomorphism [20] between the intended computation and the transformation performed by the physical system in question [21].

The archetypal model for modern artificial neural networks is the multilayer perceptron, because it exhibits desirable universal approximation properties [22]. Suppose a training dataset with samples of input-output pairs is given by $S = \{(\mathbf{x}_i, \mathbf{t}_i) \,|\, i = 1...N\}$. A multilayer perceptrons consist of 'layers' of information processing that are successively applied to input data. The layers are trained such that the outputs of the perceptron approximate the target outputs from the training dataset: $\mathbf{y}_i \approx \mathbf{t}_i$. An L-layer perceptron is the composition of L such layers. Its mathematical form can be written as:

$$\mathbf{y} = f(f(...f(f(\mathbf{x}, W^{[1]}), W^{[2]}), ...), W^{[L-1]}), W^{[L]}), \tag{2.1}$$

where f is a nonlinear activation function, $W^{[l]}$ are the trainable parameters of layer l. The transformation performed by a single layer is commonly given by:

$$f(\mathbf{x}^{[l]}, W^{[l]}) = \mathrm{ReLU}(W^{[l]} \cdot \mathbf{x}^{[l]}), \tag{2.2}$$

where $x^{[l]}$ is the input to layer l, $W^{[l]}$ is the weight matrix of controllable parameters of layer l, and f is a nonlinear activation function, here, a rectified linear unit function. The aforementioned neural network accelerators often attempt to speed up the execution of the matrix-vector multiplication $W^{[l]} \cdot \mathbf{x}^{[l]}$. The matrix-vector multiplication which is the most expensive part of the computation due to the $\mathcal{O}(n^2)$ operations necessary to multiply a vector of dimension n. The success of neural network accelerators then hinges on engineering a physical transformation that is analogous to this equation, while suppressing effects such as crosstalk, noise, or fabrication imperfections.

Fortunately, physical transformations with less stringent requirements might be equally useful for deep learning. The transformations performed by many physical systems reflect

common operations used in deep learning. Virtually all physical systems are described by differential equations and there are strong similarities between the flow of information through neural networks and differential equations [23]. Even more fundamental features of physical systems such as symmetry, locality, compositionality, and hierarchy are also found in many neural networks [24]. The field of physical reservoir computing has long tapped into the transformations performed by physical systems as a resource for cheap (and shallow) learning [25, 26].

Take the input-output relationship of any controllable physical system:

$$\mathbf{y}_\mathrm{p} = f_\mathrm{p}(\mathbf{x}, \theta). \tag{2.3}$$

Here f_p is the mathematical transformation a physical system performs on its input $\mathbf{x}$, given a set of parameters θ in the absence of noise or any stochasticity. In physical reservoir computing, a single-layer perceptron operates on the outputs of the physical system:

$$\mathbf{y} = W \cdot \mathbf{y}_\mathrm{p} = W \cdot f_\mathrm{p}(\mathbf{x}, \theta). \tag{2.4}$$

Here, the data $\mathbf{x}$ is inserted to a physical system with the (untrained) parameters θ. The outputs $\mathbf{y}_\mathrm{p}$ of the physical system are measured and multiplied by matrix W, usually using a digital computer, but sometimes directly in the physical domain [27]. Such projections f_p, like the one performed by the physical system here makes data often better linearly separable [28, 29]. While this shallow learning approach has shown initial success, it is unclear how to generalize this to more difficult tasks [24].

Recent proposals have suggested to train the parameters of the physical systems themselves [30–38]. This approach can overcome the inherent shallowness of physical reservoir computing and create similarly deep structures as in multilayer perceptrons. Only few experimental studies exist that train physical transformations directly, and these often rely on gradient-free training algorithms that scale poorly to large-scale networks [34, 36, 39, 40]. This is particularly relevant as large-scales are absolutely necessary to reap the benefits of deep learning [41]. To train large-scale artificial neural networks, the backpropagation algorithm is indispensable. Proposals to apply the backpropagation algorithm to physical systems exist

[30–33, 37, 42–44],but often rely on stringent assumptions such as linearity or dissipation-free evolution.

Here, we propose physical neural networks, deep hierarchical structures concatenating the transformations of a network of physical systems to perform deep learning tasks. In analogy to the multi-layer perceptron, the input-output relation for an L-layer physical neural network is:

$$\mathbf{y} = f_{\mathrm{p}}(f_{\mathrm{p}}(...f_{\mathrm{p}}(f_{\mathrm{p}}(\mathbf{x}, \theta^{[1]}), \theta^{[2]}), ...), \theta^{[L-1]}), \theta^{[L]}), \tag{2.5}$$

where $\theta^{[l]}$ are the controllable and trainable parameters of the l-th physical system in the network. We propose an algorithm that trains the parameters $\theta^{[l]}$ of all systems in the network simultaneously with the efficiency of backpropagation. The algorithm is a hybrid in situ-in silico algorithm, benefiting from in situ feedback from the physical systems and relying on an in silico differentiable digital model of the physical systems.

We demonstrated this approach with three distint physical systems: A mechanically oscillating plate at audio frequncies, an analog electronic nonlinear oscillator, and a nonlinear optical second-harmonic generation process. We trained all systems to be accurate classifiers for benchmark machine learning tasks, even in the presence of noise and experimental imperfections. Physical neural networks can easily be integrated into machine learning frameworks. We integrated physical neural networks with PyTorch [45] where they act as drop-in replacements for digital networks.

In Figure 2.2, we show the details of an example PNN using the propagation of a femtosecond optical pulse through a quadratic nonlinear optical medium. We encoded information about data and parameters into the frequency amplitudes of the pulse and inferred the result of the computation from the spectral components of the pulse after propagation through the medium. While passing the medium, the near-infrared pulse centered around 800nm is transformed into blue wavelengths at around 400nm in a second-harmonic generation (SHG) process. Mathematically speaking, the information encoded in the pulse is transformed in a way roughly equivalent to a nonlinear convolution.

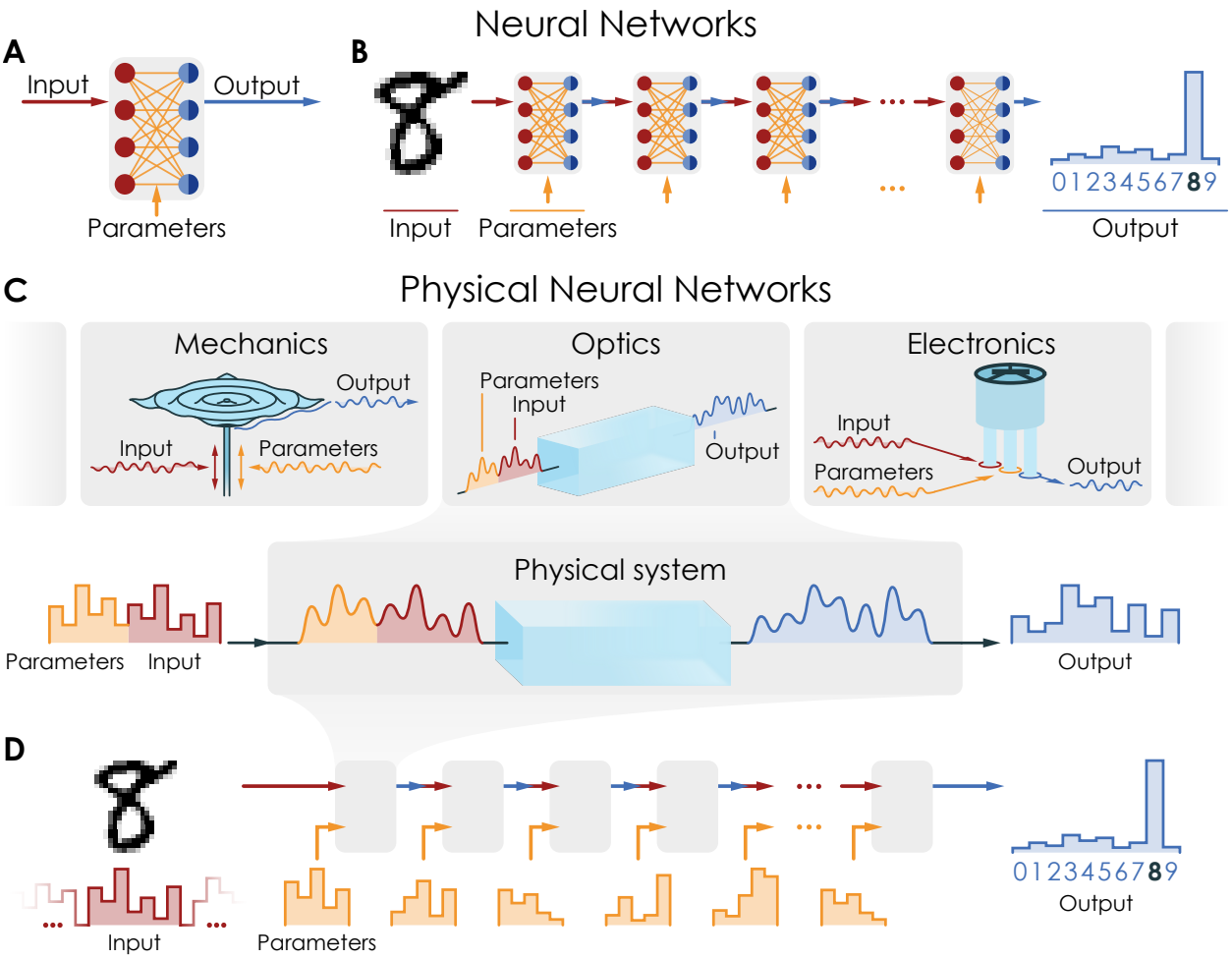

Figure 2.1: **Introduction to physical neural networks. A.** Artificial neural networks (NNs) contain operational units (layers): trainable matrix-vector multiplications followed by element-wise nonlinear activation functions. **B.** Deep neural networks (DNNs) cascade layers, and can be trained to implement multi-step (hierarchical) computations. **C.** When physical systems evolve, they perform computations. We partition their controllable properties into input data, and control parameters. By changing parameters, we alter the transformations performed on input data. We consider three examples. In a mechanical (electronic) system, input data and parameters are encoded into time-dependent forces (voltages) applied to a metal plate (nonlinear analog circuit). The physical computations result in complex multimode oscillation patterns (transient voltages), which are read out by a microphone (oscilloscope). In a nonlinear optical system, pulses pass through a nonlinear $\chi^{(2)}$ crystal, producing frequency-doubled outputs. Input data and parameters are encoded in the pulses' spectra, and outputs are obtained from the frequency-doubled pulses' spectra. **D.** Like DNNs constructed from cascading trainable nonlinear functions, we construct deep physical neural networks (PNNs) by cascading trainable physical transformations. In PNNs, each physical layer implements a controllable function, which can be of a more general form than a conventional DNN layer.

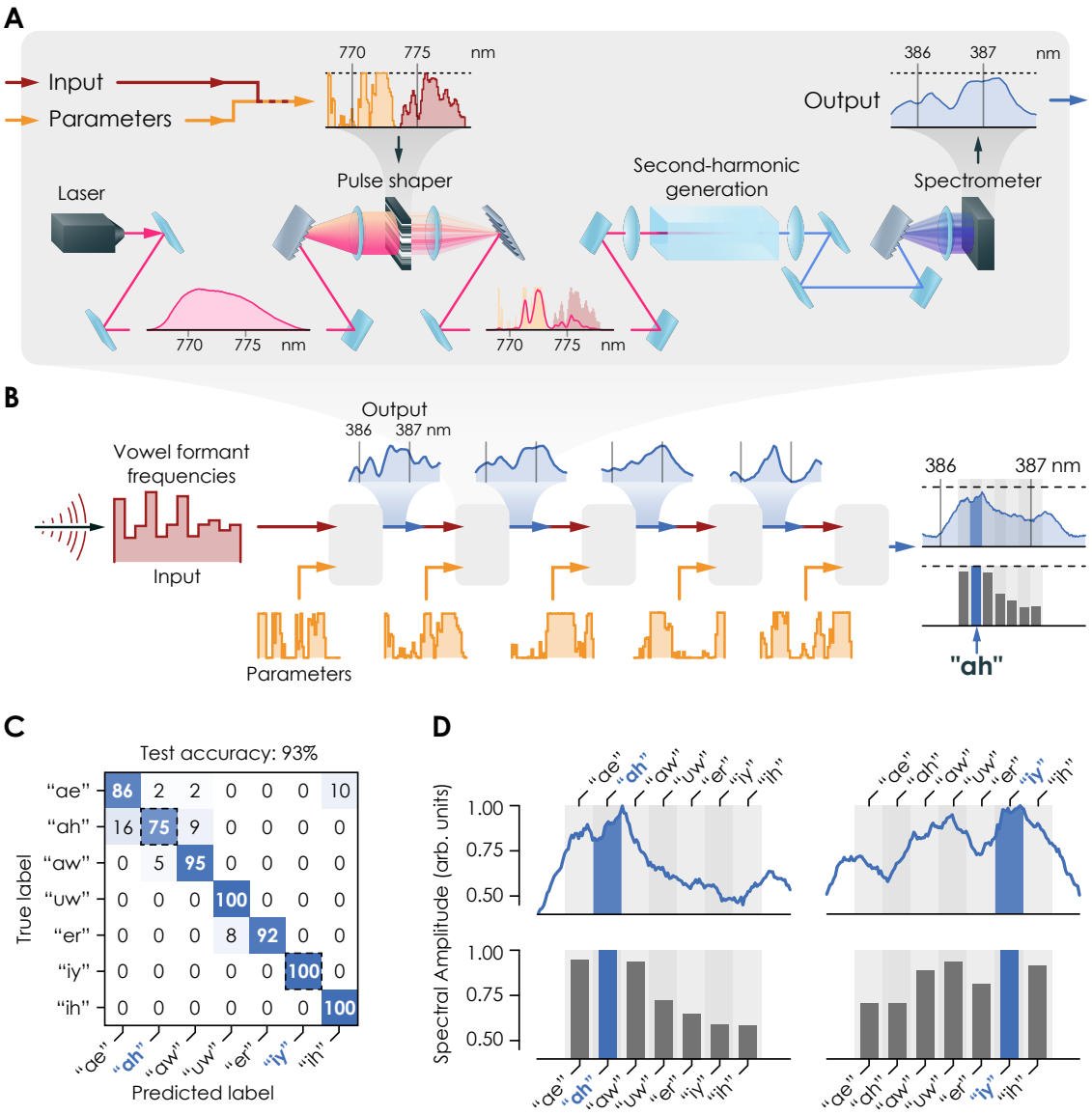

Figure 2.2: **An example physical neural network, implemented experimentally using broadband optical second-harmonic generation (SHG). a.** Input data is encoded into the spectrum of a laser pulse using a pulse shaper. To control physical transformations implemented by the broadband SHG process, a portion of the pulse's spectrum is used as trainable parameters (orange). The physical computation result is obtained from the spectrum of a frequency-doubled (blue, $\sim$ 390 nm) pulse generated within a $\chi^{(2)}$ medium. **b.** To construct a deep PNN, the output of the SHG process is used as the input to a second SHG process with independent trainable parameters. This is repeated three more times for a PNN with five trainable physical layers. The final layer's spectrum is summed into seven equal-sized spectral bins and the predicted vowel corresponds to the bin with the most energy. **c-d.** After training the SHG-PNN with PAT (see main text, Fig. 3) on the training vowels, it classifies test vowels with 93% accuracy. **c.** The confusion matrix for the PNN on the test set, showing the labels predicted by the SHG-PNN versus the correct labels. **d.** Representative examples of final-layer output for vowels correctly classified as "ah" and "iy".

We demonstrated vowel classification [46] with this system by creating a multilayer PNN in which the spectral information measured at the end of one SHG process is encoded in the input spectrum of the next process. In the first layer, the 12-dimensional vector containing the formant frequencies of a vowel recording is encoded in the infrared spectrum along with trained parameters that control the nonlinear physical convolution. The blue output spectrum is measured, digitally rescaled, and imprinted into the infrared spectrum of the next layer along with another set of trained parameters. This process is repeated five times. The output spectrum from the final SHG process is integrated into seven spectral bins, each corresponding to one of the seven vowels represented in the dataset. The result of the classification is the vowel corresponding to the spectral bin that received the most energy. This classification works correctly for 92% of the vowels in the training dataset. Note that the classification is executed almost entirely by physical computation, with only minimal digital processing. Fore more details about this particular PNN, refer to [14].

So far, we have only briefly explained that we used a novel hybrid in situ-in silico algorithm to arrive at the trained parameters that made the classification work for 92% of vowels. The algorithm to arrive at those parameters arguably represents the main technical advancement of this project and will be discussed in the next section.

2.3 Introduction to Physics-Aware Training (PAT)

In digital electronic neural networks, the backpropagation algorithm is used to train deep learning models to approximate a mapping defined by a training dataset. The mismatch between the model's output y and the target is quantified by a loss function

$$L = \frac{1}{N} \sum_i l(\mathbf{y}_i, \mathbf{t}_i), \tag{2.6}$$

where the sum is over N training samples. The function l is zero if $\mathbf{y}_i = \mathbf{t}_i$ and positive otherwise, usually quantifying a distance between output and target. To make the function of the deep learning model approximate the mapping of the dataset, the parameters of the

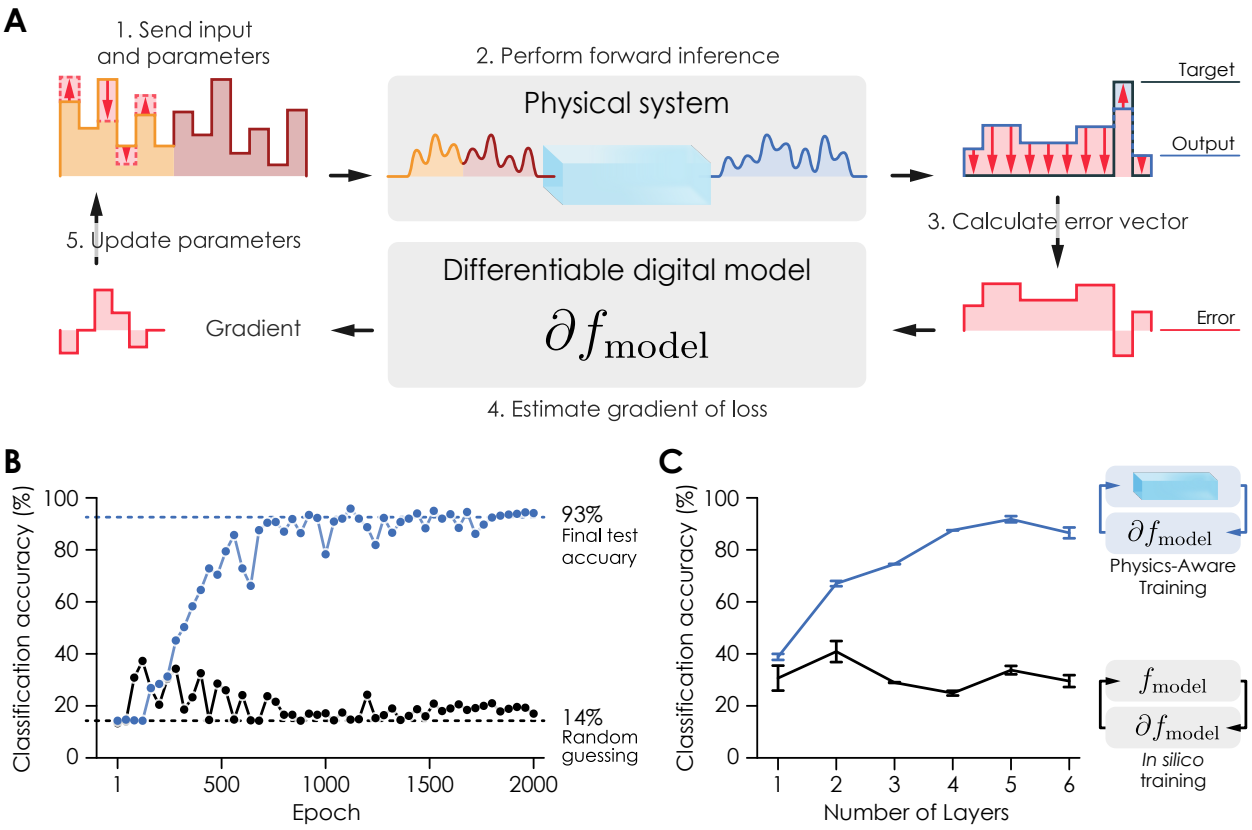

Figure 2.3: **Physics-Aware Training.** **A.** Physics-aware training (PAT) is a hybrid physical-digital algorithm to apply backpropagation to train the controllable parameters of sequences of dynamical physical evolutions, *in situ*. PAT's goal is to train physical systems to perform machine-learning tasks accurately, even though digital models are always imperfect, and systems have noise and other imperfections. The key is that, instead of performing the training solely within a digital model (i.e., *in silico*), PAT uses the physical systems to compute forward passes. Although only one is depicted in **A**, the algorithm generalizes naturally to multiple layers and other loss functions. **B.** Comparison of the validation accuracy versus training epoch with PAT and *in silico* training, for the experimental SHG-PNN depicted in Fig. 2B (5 physical layers). **C.** Final experimental test accuracy for PAT and *in silico* training for SHG-PNNs with increasing numbers of physical layers. Due to accumulating simulation-reality mismatch error through training, *in silico* training results in experimental accuracy indistinguishable from random guessing. PAT's physical forward pass suppresses this error accumulation, and permits training a 5-layer PNN that achieves 93% test accuracy.

model are updated according to the gradient:

$$\boldsymbol{W}^{[l]} \rightarrow \boldsymbol{W}^{[l]} - \eta \frac{\partial L}{\partial \boldsymbol{W}^{[l]}}, \tag{2.7}$$

where η is the learning rate. The backpropagation algorithm is used to determine the gradient of the loss function with respect to trainable parameters $\frac{\partial L}{\partial \boldsymbol{W}^{[l]}}$ and inputs $\frac{\partial L}{\partial \boldsymbol{x}}$ efficiently:

$$\frac{\partial L}{\partial \boldsymbol{x}} = \left(\frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{W}) \right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{y}}, \tag{2.8}$$

$$\frac{\partial L}{\partial \boldsymbol{W}} = \left(\frac{\partial f}{\partial \boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{W}) \right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{y}}, \tag{2.9}$$

and is used to efficiently evaluate the Jacobian matrices of the function fw.r.t. $\boldsymbol{x}$ evaluated at $(\boldsymbol{x}, \boldsymbol{W})$. where $\frac{\partial L}{\partial \boldsymbol{y}}, \frac{\partial L}{\partial \boldsymbol{x}}, \frac{\partial L}{\partial \boldsymbol{W}}$ are the gradients of the loss w.r.t. the output, input and parameters respectively. The backpropagation algorithm is N-times more efficient in computing gradients than conventional finite-difference methods for gradient estimation, as measured by the number of forward-passes. The algorithm relies on the existence of a computational graph and decomposes the overall computation into elementary operations whose derivative is analytically known. The backpropagation in its fundamental form is therefore not directly usable to train physical neural networks.

The backpropagation algorithm can still be used to design physical systems by using differentiable digital models. A mathematical model that approximates the input-output relationship of a physical system needs to be found and created from elementary operations on a digital computer. As long as the derivatives of the constituents of the model are analytically known, the outputs of the model can be differentiated with respect to its parameters. This approach is often used to design physical systems on a digital computer (in silico) and then fabricate them [30, 31, 33]. However, such a procedure is prone to simulation-reality mismatches between the digital model and the fabricated physical systems.

The key advancement of PAT is to use the physical system in the forward-pass and use a digital model in the backward-pass to estimate gradients. This approach was inspired by quantization-aware training [47], but mismatched forward and backward passes in general are widely used in neuromorphic computing [48–50]. Here, the advantage of the mismatched

forward- and backward-pass is that physics-aware training has access to the real outputs from the physical system and uses that information in the backward-pass.

PAT updates the parameters of the physical systems according to:

$$\boldsymbol{\theta}^{[l]} \rightarrow \boldsymbol{\theta}^{[l]} - \eta g_{\boldsymbol{\theta}^{[l]}}, \tag{2.10}$$

where $g_{\boldsymbol{\theta}^{[l]}}$ is an estimate of $\frac{\partial L}{\partial \boldsymbol{\theta}^{[l]}}$ and is computed according to:

$$g_{\boldsymbol{x}} = \left(\frac{\partial f_{\mathrm{m}}}{\partial \boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\theta}) \right)^{\mathrm{T}} g_{\boldsymbol{y}}, \tag{2.11}$$

$$g_{\boldsymbol{\theta}} = \left(\frac{\partial f_{\mathrm{m}}}{\partial \boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{\theta}) \right)^{\mathrm{T}} g_{\boldsymbol{y}}, \tag{2.12}$$

where $g_{\boldsymbol{y}}$, $g_{\boldsymbol{x}}$, and $g_{\boldsymbol{\theta}}$ are estimators of the gradients $\frac{\partial L}{\partial \boldsymbol{y}}$, $\frac{\partial L}{\partial \boldsymbol{x}}$, and $\frac{\partial L}{\partial \boldsymbol{\theta}}$ respectively.

The algorithm is pictorially represented in Fig. 2.3. First, a physical signal encoding data and trainable parameters is sent to the physical system. Second, the physical system processes this input and thereby effectively performs the forward pass. Third, the output from the physical system is measured, compared to the intended output, and an error signal is calculated. Fourth, this error signal is backpropagated through the computation graph of a differentiable digital model of the physical system. Lastly, the estimated gradient produced by the digital model is used to update the trainable parameters for the next forward-pass. This process generalizes to arbitrary multilayer networks and any programmable physical device for which a digital model can be created. Take for example the SHG-PNN discussed in the previous section: Despite creating an accurate physical model for the SHG process, PAT far outperformed an in silico training procedure that does not have access to feedback from the physical system. In Fig. 2.3b, in silico training reaches a maximum vowel-classification accuracy of 40%, while PAT reaches 93%.

2.4 An oscillating metal plate as a PNN

My main contribution to the experiments in this project was building a PNN out of an oscillating plate mounted on an audio speaker. Our goal was to create a pedagogical PNN that operates on "human" timescales, such that the computation would be audible or visible and, ideally, intuitively understandable. We experimented with multiple systems including billiard balls, driven double pendulums, and Chladni-plates, but eventually settled on an oscillating plate in which data and parameters are encoded in a time-dependent force driving the plate. This system presented a good compromise between an intuitively understandable system, a useful physical transformation (a convolution in time), and an audible computation. When classifying MNIST samples, the system can be heard chirping. For example, one can audibly distinguish different layers of the PNN since the frequencies at which the plate is driven is different in each layer. Unfortunately, we had to give up on a humanly-perceptible classification, as we needed the PNN to operate fast enough to permit training with a large dataset. The classification is performed on an audio signal approximately 0.25ms long and occurs too quickly to be audible.

Experimental setup

The setup of the oscillating plate PNN consists of an audio amplifier (Kinter K2020A+), a commercially available full-range speaker, a microphone (Audio-Technica ATR2100x-USB Cardioid Dynamic Microphone), and a computer controlling the setup.

We use the speaker to drive mechanical oscillations of a 3.2 cm $\times$ 3.2 cm $\times$ 1 mm titanium plate that is mounted on the speaker's voicecoil. The diaphragm has been removed from the speaker such that most of the sound produced stems from the oscillations of the titanium plate. In Figure 2.5, the steps taken to construct the oscillator with the mounted plate are shown.

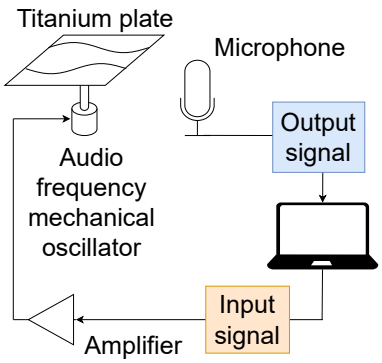

Figure 2.4: **A schematic of the information flow in the oscillating plate experimental setup.** An input and control signal from the computer is amplified and applied to the mechanical oscillator (realized by the voice coil of an acoustic speaker). A microphone records the sound produced by the oscillating plate and returns it to the computer.

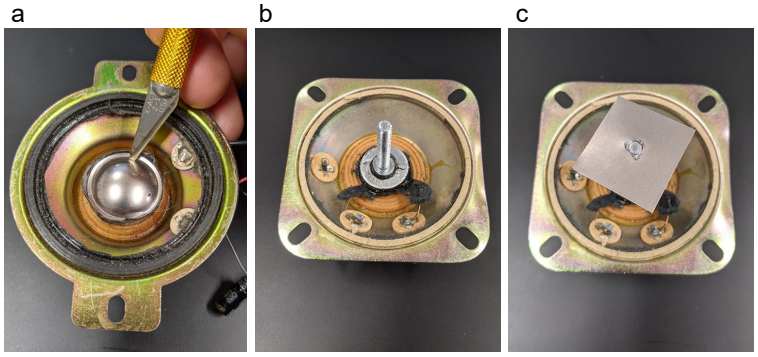

Figure 2.5: **Photographs of steps taken to construct an audio-frequency mechanical oscillator from a commercially available speaker. a.** First, we remove the diaphragm and dust cap from a speaker with a precision knife to expose the voice coil. **b.** Next, we glue a screw and washer to the voice coil with commercially available two component glue. **c.** After letting the glue dry for 24 hours, we attach the titanium plate on the screw. Finally, we securely mount the speaker on a stable surface to suppress vibrations of the whole device. Different speakers were used over the course of the experiment and not all speakers shown above are the same model.

Input and output encoding

We encode the physical inputs in the time domain. Figure 2.6 shows the different steps of encoding and decoding signals in the oscillating plate PNN. First, inputs are encoded in a time-series of rectangular pulses that are transformed into an analog signal at 192 kS/s by the laptop's soundcard DAC. The signal is amplified by an audio amplifier and drives the voicecoil of a speaker that in turn drives the oscillating titanium plate. We ensured

that both the soundcard DAC and the pre-amplifier are operated in a regime where their input-output relation is completely linear. The signal arriving at the oscillating plate was therefore a linearly amplified and slightly low-pass filtered version of the input and control signal created by the computer. The oscillations of the plate produce soundwaves which are recorded by a microphone and converted into digital signals at 192 kS/s. The signal is further downsampled by partitioning it into a number of equally long subdivisions and averaging the signal's amplitudes over this window. The length of the window is determined by the desired output dimension of the PNN layer.

Inputs and outputs are synchronized by a repeating trigger signal which precedes every sample that is played on the speaker. By overlapping the trigger signals we can synchronize samples to about 5 µs (1/sampling frequency).

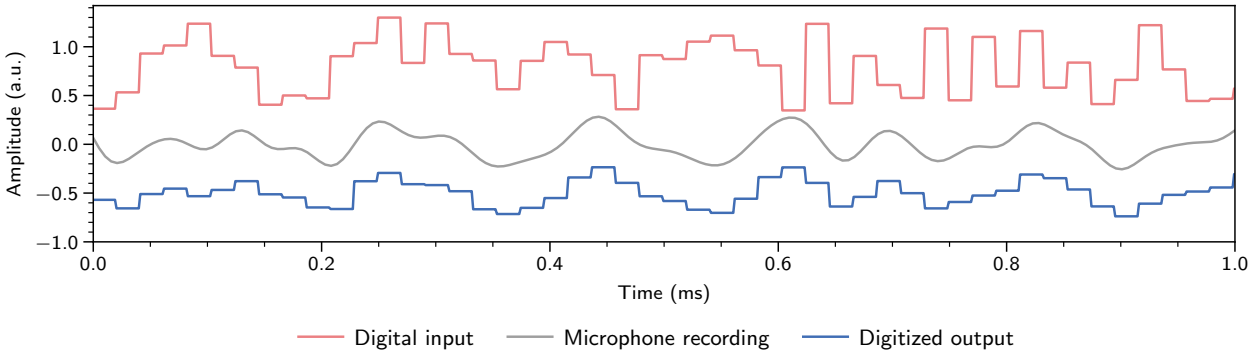

Figure 2.6: **An example of a 48-dimensional digital input to the oscillating plate setup** (red). After driving the oscillating plate, the signal is recorded by the microphone (grey) and digitized in time to the desired output dimension, here 24 (blue). Y-axis units are arbitrary normalized amplitudes, and curves are offset vertically for ease of viewing.

Characterization of input-output transformation

While we initially aimed to create high-amplitude, nonlinear oscillations on the titanium plate, we realized that we could not reach such amplitudes with our setup. In Fig. 2.7c, we observe that in the time-domain the output voltage's response is overwhelmingly linear with respect to the input voltage.

Neglecting noise and assuming the oscillations of the plate are completely linear, each input to the plate can be regarded as exciting a band of modes between the frequency of the input and the Nyquist frequency of the digital sampler (192 kHz). Following their excitation, the modes decay in a transient oscillation depending on the damping of the material, thereby affecting the following outputs in a way that can be expressed as a convolution: $y_k = \sum_{0<j<k}^{N} c_{k-j} x_{j+1}$. Here y_k is the kth output, x_j is the jth input, and the c_j coefficients are constants determined by the medium of the oscillation and the excited modes.

As a matrix operation, the map from inputs to outputs can be approximated by:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_d \end{pmatrix} = \begin{pmatrix} c_1 & 0 & 0 & \dots & 0 \\ c_2 & c_1 & 0 & \dots & 0 \\ c_3 & c_2 & c_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_d & c_{d-1} & c_{d-2} & \dots & c_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} \tag{2.13}$$

For a typical input signal to the oscillating plate PNN, the resulting output is shown in Fig. 2.7a and b. In Fig. 2.7a, eleven input time-series that are identical except for 20 inputs at around 0.3ms are plotted. Inputs change with a frequency of 192 kHz. The transient oscillation set in motion by the inputs at 0.3ms persists for about 2ms (not completely shown in the figure). Hence, at this input frequency, an output of the oscillating-plate PNN is a convolution of approximately $N = 384$ previous inputs. If we encoded inputs at a slower frequency, for example 5 kHz, we would only convolve the previous $N = 10$ inputs. Therefore, by controlling the frequency of inputs, the oscillating plate emulates the operation of a convolution with variable kernel size and fixed kernel coefficients. This feature was utilized by designing our PNN for MNIST handwritten digit classification.

For the oscillating plate, we initially tried to find a deep neural network architecture. However, once we realized the system dynamics were linear, we instead adopted a fully linear model, which was implemented in our deep neural network training framework by creating

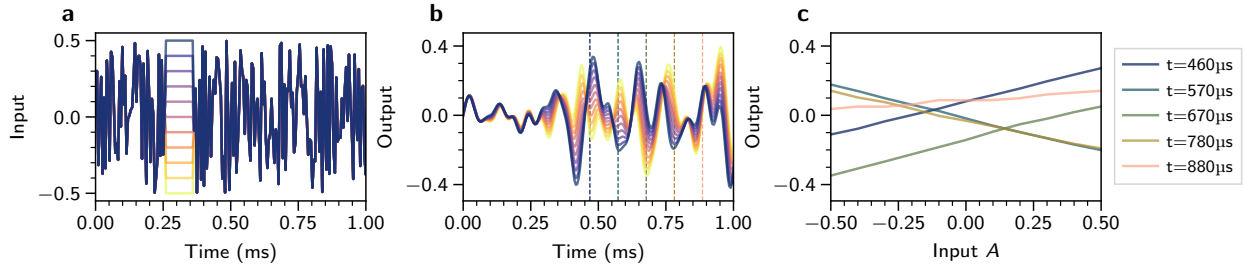

Figure 2.7: **Response of oscillating plate to a complex waveform with one varied constant section. a.** Input waveforms. **b.** Output waveforms. Due to causality, all outputs before 0.3 ms are the same (up to noise). Afterwards, different input voltages produce different transient oscillations. **c.** Slices through output waveforms as a function of the input voltage (in arbitrary units) show the overwhelmingly linear response of the output to the input voltage.

a network with no hidden layers.

Oscillating plate MNIST handwritten digit image classification PNN

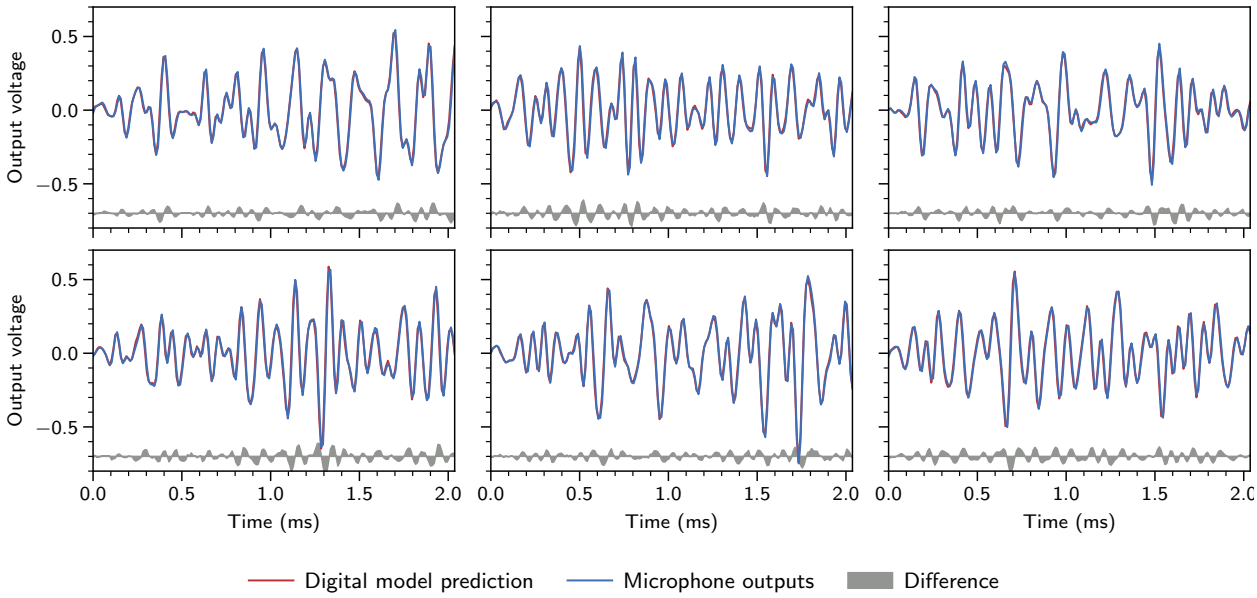

Figure 2.8: **Agreement between experimental outputs and digital model predictions for the oscillating plate setup.** Multiple microphone recordings for 196 uniformly distributed input voltages at an input frequency of 192kHz and the digital model's predictions. The linearity of the input-output transformation allowed excellent agreement between digital model and experiment as evident from the almost indistinguishable traces.

The linearity of the oscillating plate's input-output transformation allowed us to accurately

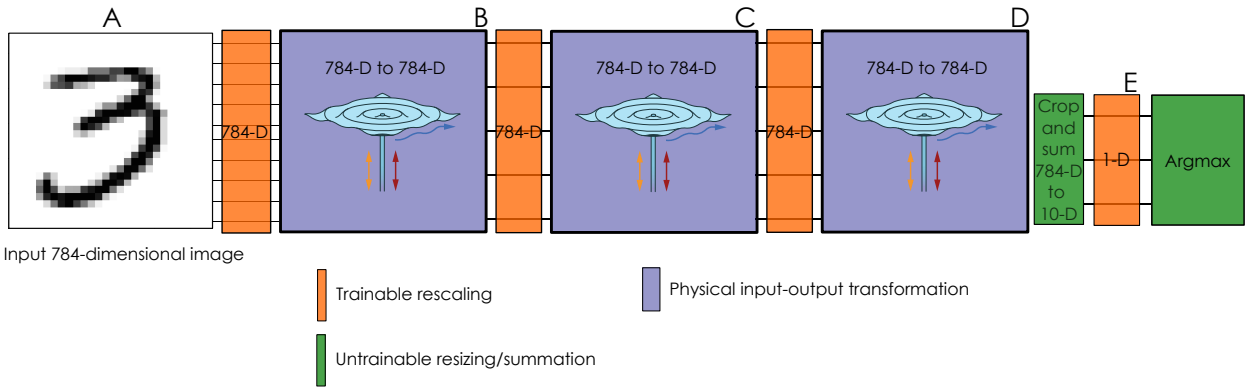

Figure 2.9: **The full PNN architecture used for the oscillating plate MNIST digit classifier.** Annotation letters (a,b,c,d,e) refer to the plots in Fig 2.10 and the description in the main text. The architecture has a total of $784 \times 2 \times 3 + 1 = 4705$ trainable parameters.

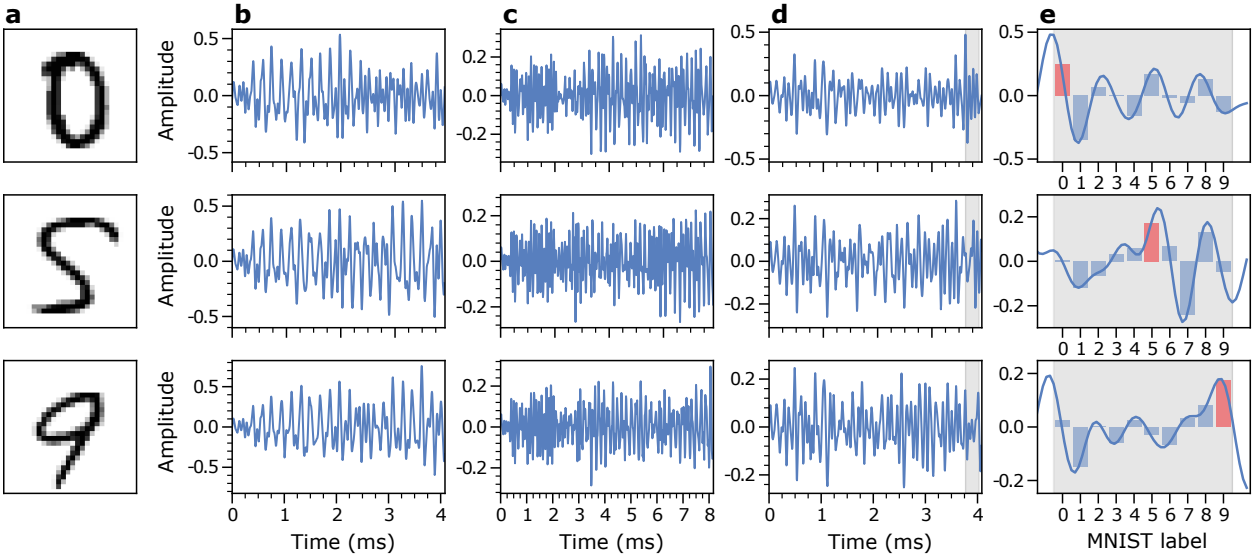

Figure 2.10: **Back-to-back classification of three sample MNIST digits with the oscillating plate PNN.** For details on each stage (a-e), see main text.

learn even very high-dimensional input-output transformations. There was no need to downsample the original 784-D MNIST images as we could learn a 784-D to 784-D dimensional input-output map. Instead, we could unroll and encode a whole image into a time-multiplexed voltage signal. Outputs of the same dimension were read out from the voltage over time signal of the microphone.

The strong performance of a linear single-layer perceptron on the MNIST dataset and the realization that the oscillating plate PNN performs a matrix multiplication as in Eq. 2.13

inspired us to use the linear oscillating plate in a similar fashion. By cascading multiple layers of element-wise-rescaling and passing the signal through the speaker we roughly emulate the operation of a single-layer perceptron. Before passing data to the speaker, we perform an element-wise rescaling, i.e. $x_i \to x_i a_i + b_i$, where a_i and b_i are trainable parameters. Then we perform the physical transformation (Eq. 2.13), producing an output 784-D vector $\vec{y}$. The combination of these two operations effectively results in the following matrix applied to an input signal:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_d \end{pmatrix} = \begin{pmatrix} c_1 a_1 & 0 & 0 & \dots & 0 \\ c_2 a_1 & c_1 a_2 & 0 & \dots & 0 \\ c_3 a_1 & c_2 a_2 & c_1 a_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_d a_1 & c_{d-1} a_2 & c_{d-2} a_3 & \dots & c_1 a_d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} \sum_{j=1}^{1} c_j b_{d+1-j} \\ \sum_{j=1}^{2} c_j b_{d+1-j} \\ \sum_{j=1}^{3} c_j b_{d+1-j} \\ \vdots \\ \sum_{j=1}^{d} c_j b_{d+1-j} \end{pmatrix} \tag{2.14}$$

Note that the physical transformation itself is not trainable, i.e. all c_i are fixed throughout training. Only the parameters a_i (and the offsets b_i) are trainable. As can be seen from this equation, a single elementwise-rescaling combined with a single pass through the speaker does not result in a universal linear operation (there are only 1568 trainable parameters instead of the 8624 parameters in a universal linear operation). Repeated application of this matrix with varying sets of rescaling-parameters a_i will result in a better approximation to a general lower triangular matrix. We demonstrate this at the example of two 3×3 lower triangular matrices of the form of Eq. 2.14. For simplicity we set $b_i = 0$ and $c_3 = 0$. The superscript i of $a_j^{(i)}$ indicates that they are independent parameters of the i-th rescaling operation. Multiplication of the matrices yields:

$$\begin{pmatrix} c_1 a_1^{(2)} & 0 & 0 \\ c_2 a_1^{(2)} & c_1 a_2^{(2)} & 0 \\ 0 & c_2 a_2^{(2)} & c_1 a_3^{(2)} \end{pmatrix} \begin{pmatrix} c_1 a_1^{(1)} & 0 & 0 \\ c_2 a_1^{(1)} & c_1 a_2^{(1)} & 0 \\ 0 & c_2 a_2^{(1)} & c_1 a_3^{(1)} \end{pmatrix} \tag{2.15}$$

$$= \begin{pmatrix} a_1^{(2)} a_1^{(1)} c_1^2 & 0 & 0 \\ a_1^{(2)} a_1^{(1)} c_1 c_2 + a_2^{(2)} a_1^{(1)} c_1 c_2 & a_2^{(2)} a_2^{(1)} c_1^2 & 0 \\ a_2^{(2)} a_1^{(1)} c_2^2 & a_2^{(2)} a_2^{(1)} c_1 c_2 + a_3^{(2)} a_2^{(1)} c_1 c_2 & a_3^{(2)} a_3^{(1)} c_1^2 \end{pmatrix}. \tag{2.16}$$

We note that the result of the matrix multiplication yields a parametrization of a fully general 3×3 lower triangular matrix. This explains how for this form of a physical neural network, repeated application of underparametrized linear layers can increase performance. This is in contrast to how repeated applications of fully-parametrized fully-connected linear layers does not increase performance as they can always be collapsed to a single linear layer (in the absence of nonlinear activation functions).

The final layer of the oscillating plate consists of reading out a short time window from the microphone output voltage. The exact position and length of the time window was a trainable hyperparameter of our network architecture, however, its position was generally closer to the end of the output as information from the input can only interact with later inputs due to causality. The window is divided into 10 bins. MNIST digits were classified according to which bin had the strongest average signal.

Each inference of the full architecture involves the following steps:

1. 784-D MNIST images are element-wise rescaled, i.e. $x_i \to x_i a_i + b_i$, where a_i and b_i are trainable parameters.
2. Run the physical transformation (Eq. 2.13) once, producing an output 784-D vector $\vec{y}$ (Figures 2.9b and 2.10b).
3. Apply trainable, element-wise rescaling to produce the input for the second 784-D physical transformation, i.e. $x_i \to x_i a_i + b_i$.

4. Run the physical transformation (Eq. 2.13) once, producing an 784-D vector $\vec{y}$ (Figures 2.9c and 2.10c).
5. Apply trainable, element-wise rescaling to produce the input for the third 784-D physical transformation, i.e. $x_i \to x_i a_i + b_i$.
6. Run the physical transformation (Eq. 2.13) once, producing an 784-D vector $\vec{y}$ (Figures 2.9d and 2.10d).
7. Take outputs 724 to 774 and average 5 consecutive points to produce a 10-D vector $\vec{y}_{\text{out}}$ (Figures 2.9e and 2.10e).

Due to the simple PNN architecture and the very accurate linear digital twins, we could achieve high accuracy on MNIST by only using *in silico* training and transferring the trained parameters onto the experiment. Using PAT still resulted in performance gains, although not as dramatic as for the nonlinear physical systems. Usually, *in silico* performance in experiment only fell a few percentage points short of PAT, and after just a few epochs (3) of transfer learning with PAT, the full performance could be retrieved. We expect therefore that, at least for very shallow networks, PAT is less crucial for linear physical systems, and we expect that *in silico* training can be generally applied to greater effect in such systems. While this may be useful in some contexts, we note that sequences of linear systems cannot realize the computations performed by deep neural networks, since they ultimately amount to a single matrix-vector multiplication.

2.5 Overview over additional PNN demonstrations

PNNs can be realized with virtually any physical system for which an accurate digital model can be created. As shown in Fig. 2.11, we demonstrated the MNIST handwritten digit classification task with three different physical systems to illustrate the wide range of possible PNNs.

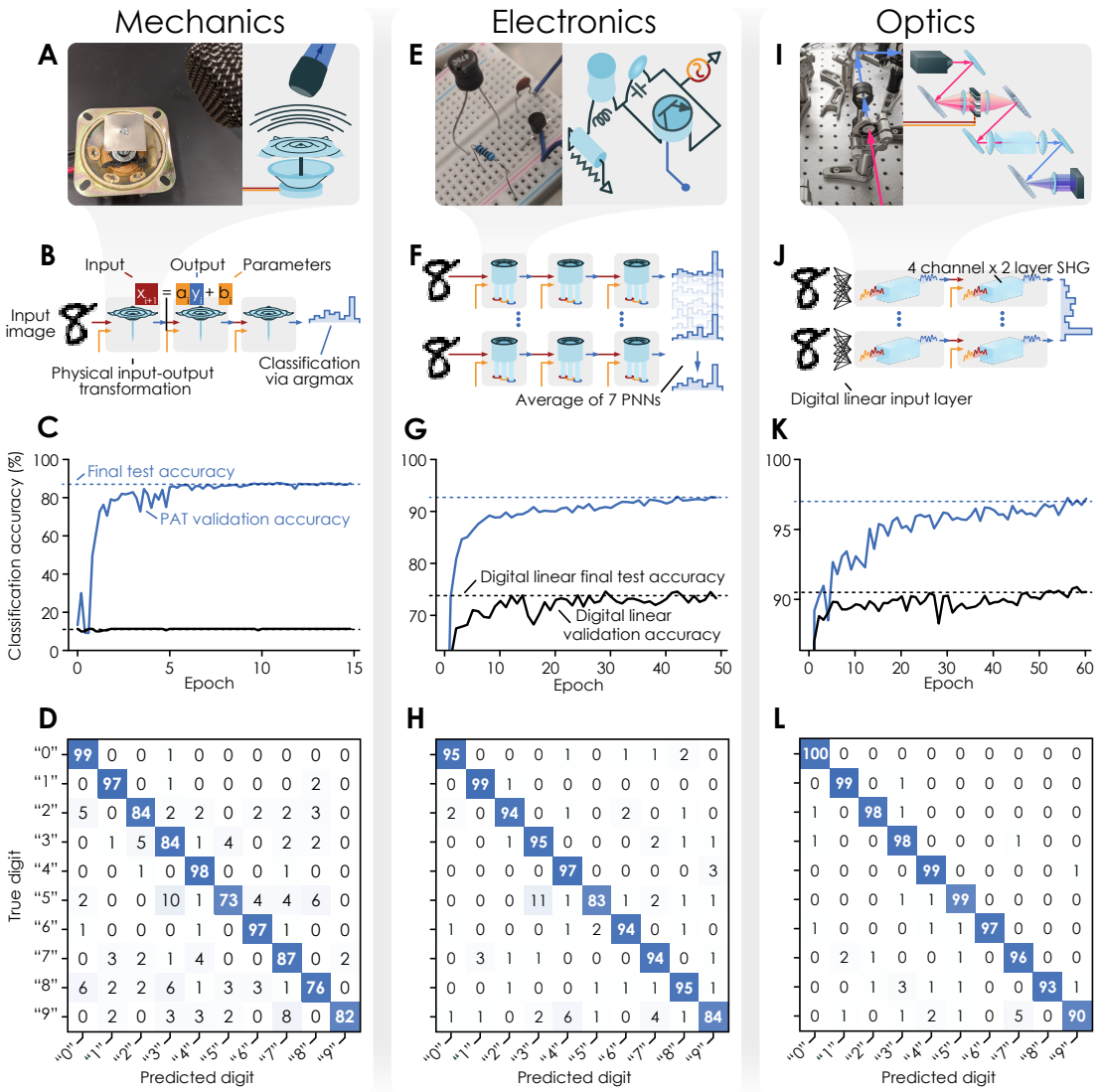

Figure 2.11: **Image classification with diverse physical systems.** We trained PNNs based on three distinct physical systems (mechanics, electronics, and optics) to classify images of handwritten digits. **A.** A photo and sketch of the mechanical PNN. The multimode oscillations of a metal plate are driven by time-dependent forces that encode the input image data and parameters. **B.** A depiction of the mechanical PNN multi-layer architecture. To encode parameters and input data, we apply digital element-wise rescaling at each time step. **C.** The validation classification accuracy versus training epoch for the mechanical PNN trained using PAT. The same curves are shown also for a reference model where the physical transformations implemented by the speaker are replaced by identity operations. The PNN reaches nearly 90% accuracy, whereas the digital-baseline does not exceed random-guessing (10%). **D.** Confusion matrix showing the classified digit label predicted by the mechanical PNN versus the correct result. **E-H.** The same as A-D, but for a nonlinear analog-electronic PNN. **I-L.** The same as A-D, for a hybrid physical-digital PNN combining digital linear input layers (trainable matrix-vector multiplications) followed by trainable physical transformations using broadband optical second-harmonic generation. Final test accuracy is 87%, 93% and 97% for mechanical, electronic, and optics-based PNNs respectively.

In the mechanical PNN (Fig. 2.11a–d), a metal plate is driven by time-varying forces, which encode both input data and trainable parameters. The plate's multimode oscillations enact controllable convolutions on the input data. Using the plate's trainable transformation sequentially three times, we classify 28-by-28 (784 pixel) images that are input as an unrolled time series. To control the transformations of each physical layer, we train element-wise rescaling of the forces applied to the plate (Fig. 2.11b). PAT trains the three-layer mechanical PNN to 87% accuracy, close to a digital linear classifier. When the mechanical computations are replaced by identity operations, and only the digital rescaling operations are trained, the performance of the model is equivalent to random guessing (10%). This shows that most of the PNN's functionality comes from the controlled physical transformations.

In the center column, an analog electronic PNN is shown that consists of a circuit with a transistor and other linear elements. Like the mechanical PNN, data and parameters are encoded in time-series signals driving the system, here, resulting in noisy, nonlinear transients. Inputs and parameters are combined by an element-wise rescaling operation. We used a committee of seven, three-layer deep networks, whose outputs are averaged to produce the final classification. Due to the increased number of digital parameters, a purely digital architecture of this kind would achieve 74% accuracy on this task, though the inclusion of the analog-electronic layers boost the performance to 93%.

Using the same SHG system that was used to perform vowel-classification in Fig. 2.2, we demonstrated a hybrid physical-digital PNN (Fig. 2.11i-l). Before encoding a down-sampled 196-dimensional MNIST image into the infrared spectrum of the optical pulse, we process the input with digital layers. After propagating through two physical layers, we combine the outputs from 4 channels to produce the final output. A purely digital architecture of this kind would achieve 91% accuracy and including the physical layers, we achieve 97% accuracy. Increasing the accuracy on the MNIST classification task beyond approximately 90% becomes incrementally harder: Typically going from 90% to 97% accuracy requires an increase in networks size of about 10x.

A note on the mathematical (dis-)similarity of interleaving and rescaling parameter encoding for linear and nonlinear PNNs

After publication of Wright et al [14], we were asked why we chose to encode parameters in an element-wise rescaling operation for some PNNs, while we chose a "concatenation" or "interleaving" operation for the PNN presented in Fig. 2.2. We recall that these operations are of the form:

$$\text{element-wise-rescaling}(\mathbf{x}, \mathbf{a}, \mathbf{b}) = \begin{pmatrix} a_0 x_0 + b_0 \\ a_1 x_1 + b_1 \\ \vdots \\ a_n x_n + b_n \end{pmatrix} \tag{2.17}$$

and

$$\text{interleaving}(\mathbf{x}, \mathbf{a}) = \begin{pmatrix} a_0 \\ x_0 \\ a_1 \\ x_1 \\ \vdots \\ a_n \\ x_n \end{pmatrix}. \tag{2.18}$$

A concatenation of parameters is fundamentally very similar to the interleaving operation and only differs in the order in which parameters and inputs are interleaved.

No linear PNN using the interleaving operation can produce outputs similar to the 'rescale' operation. This is because a linear PNN using interleaving will always have outputs of the form

$$y_n \propto \sum_i c_i x_i + \sum_j c_j a_j, \tag{2.19}$$

and never outputs multiplicatively mixing inputs and parameters, i.e. terms of the form $a_i x_i$, as created by the element-wise rescaling operation.

On the other hand, a nonlinear PNN with a non-instantaneous nonlinearity can produce multiplicative mixing terms between inputs and parameters, even with an interleaving input

scheme. We imagine a physical system that creates time series similar to Eq. (6) from Wright et al [14]:

$$y_n = x_n x_n + \gamma x_n x_{n-1} + \gamma^2 x_n x_{n-2} + ... \tag{2.20}$$

Plugging a vector of interleaved parameters and inputs into such a nonlinear PNN reproduces mixing terms between inputs and parameters similar to the ones produced by the same nonlinear PNN with the element-wise rescaling scheme. This is why we used an interleaving/concatenation scheme for the nonlinear SHG-PNN from Fig. 2.2, but an element-wise rescaling scheme for the linear speaker PNN.

2.6 Discussion and Outlook

We have shown that a variety of physical systems can be used to perform computations similar to those performed in artificial deep neural networks. The demonstrations here are limited to a simple classification task but in principle, PNNs can be deployed on much more complex tasks. This begs the question which physical systems can create PNNs that can compete or outperform digital electronic neural networks in terms of speed or energy-efficiency. One approach to quantify the potency of a physical system, inspired by recent work on quantum computational advantages [51], is to look at the "self-simulation advantage", i.e. a comparison of the energy necessary to instantiate a physical process vs. the energy necessary to simulate it on a digital electronic computer.

A physical system "simulating itself" can routinely be millions or billions of times faster and more energy-efficient than a simulation thereof on a digital electronic computer [14]. Although this is an absolute upper bound on the possible performance gain, it can give valuable insight. The challenge is usually to find a physical system that 1) can be controlled at a sufficiently fine scale and 2) whose symmetries and other constraints are compatible with the task at hand. In Chapter 4, we set out to create a photonic device that offers extremely versatile programmability and in principle, a large self-simulation advantage.

A second constraint is that PNNs rely on differentiable digital models during the training stage. This means the energy-use of a very efficient PNN might be dominated by the energy used to train it with a digital model. In general, about 90% of energy is used in the inference-phase of neural networks [6]. Improvements in the training of PNNs could circumvent this constraint. For example, a PNN could learn to predict the gradient of another PNN and make digital models obsolete. This could result in a bootstrapping-scenario in which PNNs beget ever more complex PNNs. A less speculative improvement could come from algorithms that promise backpropagation-like efficiency but do not require a backward-pass for gradient estimation [43, 52–60].

CHAPTER 3

BACKGROUND ON ON-CHIP PHOTONIC PROCESSORS

3.1 Survey of on-chip optical neural networks

In this section, we review the state-of-the-art of on-chip optical neural networks with a focus on spatial domain implementations. This field of research was incepted decades ago. To our knowledge, the first published research of an on-chip optical neural network is from 1991 [61]. A thin layer of photorefractive crystal effectively performed a matrix-vector multiplication by coupling multiple beams of light propagating in the crystal via gratings. While this small experimental demonstration with three optical modes–or any later demonstrations–did not result in the millions of weighted interconnections the authors speculated about, the idea proved to be fruitful and almost three decades later gave rise to a flourishing field of research. To this date, multiple reviews and books have been written on the topic: The review of Shastri et al [62] and the book by Prucnal & Shastri [63] give a broad overview over the field and are particularly recommended for anyone interested in photonic neurons and spiking neural networks. Al-Qadasi et al [64] focus on silicon photonic implementations of neuromorphic photonics with detailed data on energy-efficiency and scaling of components on this platform. Capmany and Pérez also cover waveguide meshes and neuromorphic photonics in their book on programmable integrated photonics [65]. The reviews of Peserico et al [66] and Farmakidis et al [67] are the most relevant to this thesis due to their extensive and up-to-date coverage of spatially multiplexed photonic neural networks. Readers are encouraged to consult these reviews in addition to what follows.

Before discussing the particulars of spatial domain *on-chip* optical computing, we start by highlighting the general advantages of optical computing.

Physics of optical computing

Optics exhibits a variety of fortuitous features that, fundamentally, could allow optical computers to outperform digital electronic computers on specific tasks. We discuss the three most important features according to Ref. [68] here. It should be noted that, in practice, optical computing suffers from a variety problems, which we will also discuss at the end of this section.

1), The enormous **bandwidth** of optical signals might allow orders of magnitudes faster processing of optical signals compared to electronic signals. While the clock rate of digital electronic processors has reached a ceiling at a few GHz [1], optical signals can in principle access bandwidths on the order of hundreds of THz (though so far optical modulators have only reached about 100 GHz [69, 70]). Even if only a fraction of this bandwidth can be accessed, it provides ample space to process upconverted microwave and radio frequency signals without the need to carefully balance processing elements [71].

2), Unlike electronic signals, optical signals propagate virtually **dissipationless**. This is of course the reason that most long-haul communication networks rely on optical signals propagating in fibre. But even on a much smaller scale, in electronic integrated circuits containing digital processors, energy dissipation is dominated by charging electrical communication lines due to parasitic capacitance and resistance. The energy necessary to communicate between logic gates far outweights the energy dissipated in the logic gates itself nowadays [72]. By comparison, optical signals (almost) only require energy to create photons. Once created, photons propagate virtually dissipation-free over distances associated with integrated circuits (centimeter-scale distances with losses of a few dB/m [73]). This can in principle drastically reduce the energy dissipation in computers. However, as long as signals are processed electronically and electronic components remain orders of magnitudes smaller, there is little incentive to communicate between logic units with optical signals.

3), Optical processors are well-suited to benefit from **spatial parallelism**. Spatial light modulators are a mature technology, being used in projectors and screens, and routinely

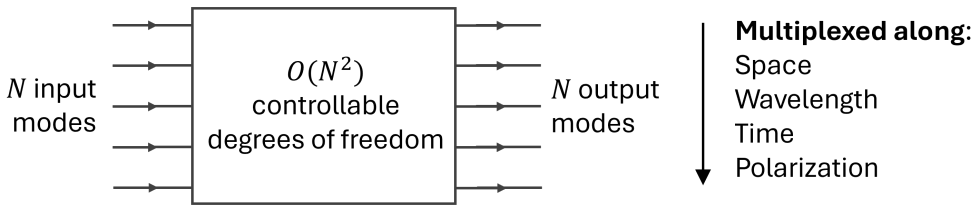

Figure 3.1: **Scheme of a generic on-chip optical neural network accelerator.** N optical input modes enter the device and are mixed and processed by $\mathcal{O}(N^2)$ controllable degrees of freedom (optical modulators). The optical modes can be multiplexed along on or more of the following dimensions: Space, wavelength, time, and polarization.

modulate more than one million optical modes (pixels). The number of components on integrated photonic circuits is also exponentially growing and has reached over 100,000 components per chip [74]. While optical processors lag far behind their electronic counterparts in the spatial density of elements (and likely always will due to the large wavelength of optical waves), they may gain a unique advantage over electronics from three-dimensional integration (as for example rudimentarily realized in a bulk photorefractive crystal [75] or 3D-printed phase-plates [76]).

On-chip optical neural networks aim to exploit these advantages. It should be noted that low-energy optical nonlinearities are difficult to implement. Therefore, almost all optical neural network accelerators focus on accelerating the linear matrix-vector multiplication of neural networks. Fig. 3.1 shows a generic schematic of such an optical neural networks accelerator, in which N input modes are processed by $\mathcal{O}(N^2)$ controllable degrees of freedom. The input modes and controllable degrees of freedom can be multiplexed along dimensions of space, wavelength, time, or polarization. Spatial domain optical neural networks are currently the dominant approach, and the core of this thesis. In this approach, the N input modes are spatial modes, such as the modes of N parallely running single-mode waveguides. The $\mathcal{O}(N^2)$ degrees of freedom are spatial degrees of freedom, e.g. many modulators that are fabricated on a chip and individually addressed. We proceed by reviewing the most important approaches of this kind.

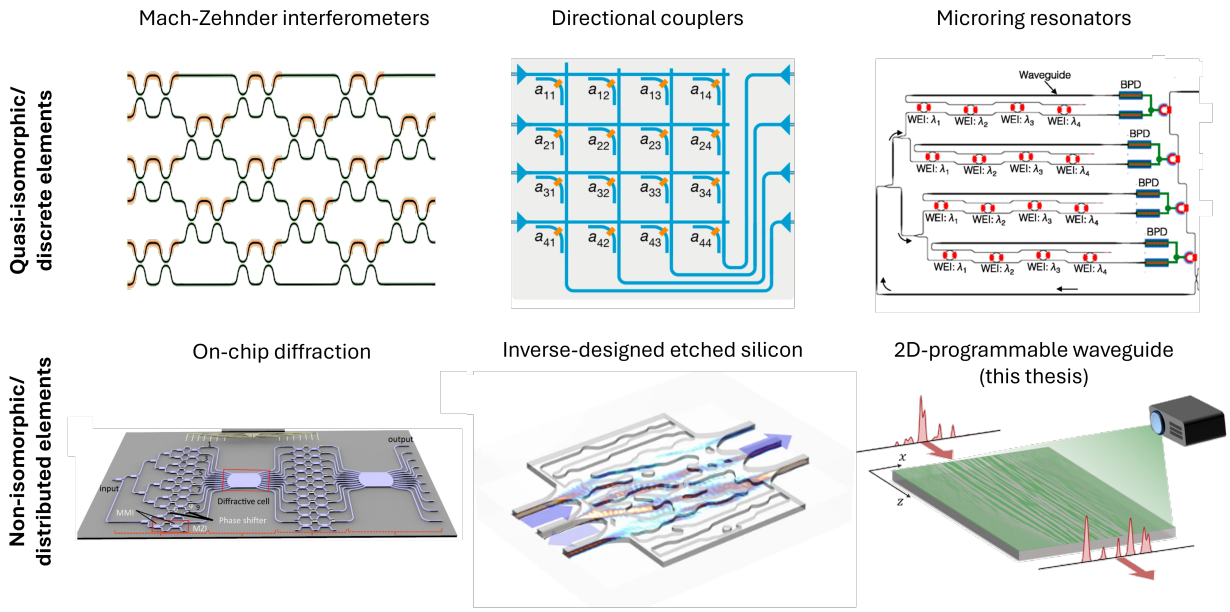

Figure 3.2: **Examples of select spatially multiplexed on-chip optical neural network schemes.** Mach Zehnder interferometer schematic reprinted from [77]. Directional coupler schematic reprinted from [78]. Microring resonator schematic reprinted from [79]. On-chip diffraction schematic reprinted from [80]. Inverse-designed etched silicon reprinted from [81]. The 2D-programmable waveguide schematic is our work.

Spatial domain on-chip optical neural networks

There are many different implementations of spatially-multiplexed optical neural networks. Here, we focus on a qualitative comparison of different implementations. In section 5.6, we present a quantitative comparison of the neuromorphic computations performed with different approaches. Shown in Fig. 3.2 are some of the most important approaches to spatial domain ONNs. Since it is of relevance for this thesis, we have sorted them into two categories according to the way in which controllable parameters map to matrix elements of the transformation they perform.

In the first row, we show "quasi-isomorphic" optical neural networks. In one way or another, these approaches rely on a decomposition of a matrix into many beamsplitter operations performed by discrete optical elements. We call these approaches quasi-isomorphic because there exists a bijective mapping between the desired matrix elements and the modulation

applied at discrete element.

In the second row, we show non-isomorphic optical neural networks. These approaches (partially) rely on multimode wave propagation and there is no clear mapping between the matrix elements and the distributed elements of the medium. Instead, the controllable parameters are usually iteratively computationally designed.

Quasi-isomorphic matrix vector multipliers rely on a variety of optical elements that effectively perform a beamsplitter operation. Popular approaches are Mach-Zehnder interferometers (MZIs) [17, 77, 82, 83], directional couplers [78], or microring resonators (MRRs) [79, 84]. Of those, Mach-Zehnder interferometer meshes is the dominant approach. This technology largely grew out of the desire to implement arbitrary unitary operations for application in quantum computing in integrated photonic circuits [85, 86]. The controllable MZIs implemented in an integrated photonic circuit are effectively a controllable beamsplitter operating on the two spatial modes of two single-mode waveguides. They are implemented via a 50:50 directional coupler, followed by a phase-shifter on one arm, followed by another 50:50 directional coupler, followed by another phase-shifter on one arm. The operation performed by this controllable MZI on the amplitudes of the modes a and b is

$$\hat{U}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} e^{\varphi}\sin(\theta/2) & e^{\varphi}\cos(\theta/2) \\ \cos(\theta/2) & -\sin(\theta/2) \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}. \tag{3.1}$$

The phase shifts θ and ϕ are often controlled by thermo-optic or electro-optic phase-shifters. An important insight is that an arbitrarily large unitary matrix can be decomposed into successive applications of many such beamsplitter operations [87]. Applying this insight to implement a neural network accelerator resulted in a seminal paper [88] performing controllable matrix multiplications on a photonic chip with a mesh of MZIs. An arbitrary $N \times M$ matrix was decomposed via singular value decomposition into the form $U\Sigma V^{\dagger}$, where U and V are unitary matrices and Σ is a diagonal matrix. Each unitary matrix was realized with an MZI mesh and the diagonal matrix performed with N MZIs in which the output from one arm was dropped. Many follow-up papers to this work were published. We list some of note here. Ref. [83] performed coherent detection on the output ports to realize a complex-valued

matrix-multiplication. Ref. [77] created an all-analog multi-layer ONN from MZI meshes including nonlinear activation functions. The activation functions are realized by photodiodes that drive a microring resonator on and off resonance. Ref. [53] implemented in situ measurements of gradients for the training of MZI parameters via a scheme proposed in Ref. [43].

Other seminal contributions to the field of spatial-domain on-chip photonic processors address some of the glaring issues with meshes of MZIs. One of the main concerns is the large spatial footprint of MZIs. Each MZI usually has a footprint of 10,000s of μm^2. In seminal work preceeding the work of Shen & Harris et al [17], Tait et al [84] use arrays of thermo-optically modulated microring resonators (MRRs) with balanced photodiodes for matrix-vector multiplications. Such MRRs are much more space-efficient with a footprint of 100s of μm^2 and in principle even smaller [89]. While the MRRs are spatially-multiplexed in this approach, the optical modes are frequency modes, which are incoherently added up with balanced photodetectors. Drawbacks of this scheme are a lower bandwidth due to the intermittent conversion to an electronic signal and the slower modulation speed of MRRs, as well as thermal stability problems associated with the delicate resonances of MRRs. These stability issues can be addressed by sharpness-aware training [90], for example.

Whether modulating a MZI or a MRR, the thermo-optic modulators used draw a non-negligible amount of power. Feldmann et al address this problem by modulating beams propagating through a network of directional couplers with phase-change materials that, once set, do not consume any power [78]. Their scheme implements a matrix-vector multiplication with 4 inputs and 4 outputs with frequency multiplexing, performing the same MVM on different inputs at four different wavelengths. Ref. [91] realizes an all-analog multi-layer ONN with an activation function similar to Ref. [77], but with photo-absorption modulators instead of MZIs.

Taking a step towards non-isomorphic on-chip ONNs are so-called on-chip diffractive neural networks [80, 92], in which the single-mode dynamics are interspersed with multimode

waveguide propagation, effectively mixing multiple spatial modes. A series of theoretical proposals and small experimental demonstrations take this approach to the extreme. Rather than using discrete optical components that each correspond to one matrix element, [30, 93, 94] propse to use wave propagation through an inverse-designed microphotonic medium for neuromorphic computations. Small experimental demonstrations of this approach exist in photorefractive crystals [61] or silicon photonics [81]. The latter one is exceptionally compact and compatible with volume-manufacturing but, since it is using permanently etched features, does not offer post-fabrication tunability. A twist to this approach is the work by Wu et al [95], who instead of programming the real part of the refractive index program the imaginary part, creating a gain/loss landscape. The authors encode information in mutually incoherent light, effectively demonstrating a matrix-vector multiplication with positive weights.

All schemes presented so far perform the computation in the time-of-flight of photons passing through the device. A different approach swaps some of the spatial multiplexing for temporal multiplexing [96–98]. A matrix-vector multiplication can be performed with just N beamsplitters and N photodetectors integrating a signal over N time steps [96]. Rahimi et al [97] implement a version of this proposal in which N^2 beam splitters are used to perform a matrix-matrix multiplication with N^3 operations in N timesteps.

3.2 Physics of integrated photonic devices

We now derive important equations that are necessary to understand the work in Chapter 4 in detail. We start by deriving the Helmholtz equation since it marks the starting point for two separate derivations relevant for Chapter 4. The equation describes the spatial distribution of electric and magnetic fields under the assumption of a continuous wave signal, that is, a field created by sources continuously oscillating at one frequency ω. We start our derivation from the macroscopic Maxwell equations:

$$\nabla \cdot \mathbf{D} = \rho_f$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t}$$

where $\nabla\cdot$ is the divergence, $\nabla\times$ is the curl, $\mathbf{D}$ is the electric displacement field ($\mathrm{C/m^2}$), $\mathbf{B}$ is the magnetic flux density (T), $\mathbf{E}$ is the electric field (V/m), $\mathbf{H}$ is the magnetic field (A/m), $\mathbf{J}_f$ is the free current density ($\mathrm{A/m^2}$), ρ_f is the free charge density ($\mathrm{C/m^3}$), t is the time (s), and the constitutive relations are

$$\mathbf{D} = \epsilon \mathbf{E} \tag{3.2}$$
$$\mathbf{H} = \frac{1}{\mu}\mathbf{B} \tag{3.3}$$

In photonics we almost always deal with non-magnetic materials for which $\mu = \mu_0$, but the electric permittivity can in the general case be quite complicated. In the case of anisotropic materials, the permittivity is of tensor form $\epsilon = \epsilon_{ij}$ and, for "nonlinear materials", it is dependent on the electric field $\epsilon = \epsilon(\mathbf{E})$.

For now, we consider the case of an isotropic, linear optical material with a scalar and constant permittivity ϵ. In this case, the following wave equation follows from Maxwell's equations:

$$\nabla^2 U - \frac{n^2}{c_0^2}\frac{\partial^2 U}{\partial t^2} = 0, \tag{3.4}$$

where $c = 1/\sqrt{\epsilon_0 \mu_0}$ and $n = \sqrt{\epsilon_r}$ the refractive index. This equation holds for all components of the electric field E, magnetic flux density B, and–assuming a non-magnetic medium, which is common in photonics–also for the magnetic field H.

Plugging a continuous wave field of constant frequency of the form $U(\mathbf{r}, t) = U(\mathbf{r})e^{-i\omega t}$ into the wave equation yields an equation for the spatial distribution of the wave:

$$\nabla^2 U + n^2 k_0^2 U = 0, \tag{3.5}$$

where $k_0 = \frac{\omega}{c_0}$. This equation is known as the Helmholtz equation and it represents the starting point for many important derivations in the field of photonics.

Guides modes of slab waveguides

We investigate solutions to the Helmholtz equation with a refractive index that has a "step" profile in one dimension,

$$n(y) = \begin{cases} n_{\text{cl}} & \text{for } y \geq 0 \\ n_{\text{co}} & \text{for } 0 \geq y \geq -h \\ n_{\text{cl}} & \text{for } -h \geq y \end{cases} \tag{3.6}$$

and is translationally invariant in the dimensions x and z. This corresponds to a three-layer physical device in which a core layer with index n_{co} is clad by two layers with index $n_{\text{cl}} < n_{\text{co}}$, commonly called a *slab waveguide*, because spatially confined/guided solutions to the Helmholtz equation arise in this scenario.

We investigate a solution in which light is primarily propagating in the z-direction and the electric field points in the x-direction

$$\mathbf{E}(\mathbf{r}) = E_x(y, z)\mathbf{e}_x. \tag{3.7}$$

Due to the translational invariance in x and z, we expect the field to only pick up a phase in z and be invariant in x:

$$E_x(y, z) = E_x(y)e^{i\beta z}, \tag{3.8}$$

where β is a yet unknown propagation constant.

Plugging this ansatz into the Helmholtz equation 3.5, we find solutions of the form

$$E_x(y) = \begin{cases} Ae^{-\gamma y} & \text{for } y \geq 0 \\ B\cos\kappa y + C\cos\kappa y & \text{for } 0 \geq y \geq -h\,, \\ De^{\gamma(y+h)} & \text{for } -h \geq y \end{cases} \tag{3.9}$$

where $\gamma = \sqrt{\beta^2 - n_{\text{cl}}^2 k_0^2}$ and $\kappa = \sqrt{n_{\text{co}}^2 k_0^2 - \beta^2}$.

The associated magnetic field is found via Faraday's law to be $H_y(y) = -\frac{\beta}{\mu_0 \omega} E_x(y)$ and $H_z(y) = -\frac{i}{\mu_0 \omega} \frac{\partial E_x(y)}{\partial y}$. The tangential components of the electric and magnetic field are continuous across interfaces of varying index in the absence of electric currents, which we assume here. Therefore $E_x(y)$ and $H_y(y)$ will be continuous at $y = 0$ and $y = -h$, providing four equations that determine the constants B, C, and D, as well as the relation

$$\tan h\kappa = \frac{2\gamma}{\kappa(1 - \frac{\gamma^2}{\kappa^2})}, \tag{3.10}$$

from which the propagation constant β can be determined. These solutions to the Helmholtz equation are termed transverse-electric (TE) modes, as the electric field (and only the electric field) is transverse to the direction of propagation. Regardless of the values of h, n_{co} or n_{cl}, one solution to this equation always exists. The number of solutions grows approximately linearly with the thickness of the core layer and can be estimated by $N_{\text{modes}} \approx \frac{k_0 h}{\pi} \sqrt{n_{\text{co}}^2 - n_{\text{cl}}^2}$.

An analogous calculation can be performed with the ansatz

$$\mathbf{H}(\mathbf{r}) = H_x(y) e^{i\beta z} \mathbf{e}_x, \tag{3.11}$$

in which case the functional form of Eq. 3.9 holds for $H_x(y)$, but different boundary conditions lead to different values for the constants B, C, and D. Additionally, the condition for β becomes

$$\tan h\kappa = \frac{2\gamma\kappa \frac{n_{\text{co}}^2}{n_{\text{cl}}^2}}{\kappa^2 - \frac{n_{\text{co}}^4}{n_{\text{cl}}^4}\gamma^2}. \tag{3.12}$$

In contrast to TE modes, these solutions are termed transverse-magnetic (TM) modes, as the magnetic field is transverse to the direction of propagation. Similarly to TE modes, one solution to this equation always exists. In other words, there is always one guided TM mode in a symmetric slab waveguide.

A similar analysis with a step-index profile in both x and y yields guided modes confining light in both dimensions. Such waveguides (buried waveguides, strip-loaded waveguides,

ridge waveguides, etc.) form the foundation of most integrated photonic devices. More pertinent to the work in this thesis though is a waveguide in which light is tightly confined in one dimension and only lightly perturbed in the other dimension. We analyze this waveguide in the following section.

3.2.1 Two-dimensional wave equation for waves propagating in a weakly perturbed slab waveguide

We analyze a weakly perturbed slab waveguide with

$$n(x,y,z) = \begin{cases} n_{\text{cl}} & \text{for } y \geq 0 \\ n_{\text{co}} + \Delta n(x,z) & \text{for } 0 \geq y \geq -h \\ n_{\text{cl}} & \text{for } -h \geq y \end{cases}, \tag{3.13}$$

where $\Delta n(x,z) \ll n_{\text{co}} - n_{\text{cl}}$. In this weakly perturbed scenario, the Helmholtz equation is approximately separable (as long as $\frac{\partial^2 U}{\partial y^2} \gg \Delta n(x,z)$), such that we may use an ansatz of the form

$$\mathbf{E}(\mathbf{r}) = E_x(y)A(x,z)e^{i\beta z}\mathbf{e}_x, \tag{3.14}$$

and $E_x(y)$ and $A(x,z)e^{i\beta z}$ both separately fulfill the Helmholtz equation. The solutions to $E_x(y)$ are the ones presented in Section 3.2. Plugging the ansatz for the x- and z-dependent parts of $\mathbf{E}(\mathbf{r})$ into the Helmholtz equation and approximating $n^2 = (n_{\text{co}} + \Delta n(x,z))^2 \approx n_{\text{co}}^2 + 2\Delta n(x,z)$, we find

$$\frac{\partial A}{\partial z} = \frac{i}{2k}\frac{\partial^2 A}{\partial x^2} + i\Delta n k_0 A, \tag{3.15}$$

where $k = nk_0$. We use this equation to describe the propagation of fields in our two-dimensionally programmable waveguide. Next, we discuss how the index perturbations $\Delta n(x,z)$ are created.

3.2.2 Creating index perturbations with electro-optic materials

Nonlinear optical media are characterized by an electric polarization $\mathbf{P}$ that is nonlinear with regards to the applied electric field $\mathbf{E}$:

$$\mathbf{P} = \epsilon_0 \epsilon_r(\mathbf{E})\mathbf{E}. \tag{3.16}$$

In response to strong electric fields inside the medium, be they from high intensity lasers or from externally applied fields, charge carriers inside the medium polarize with varying strengths. Depending on the structure of the medium and the frequency of the electric field various effects take place: Electron clouds become displaced with respect to the nucleus (up to attoseconds), ions move within the crystal structure (up to femtoseconds), polarized molecules align along the direction of the electric field (up to picoseconds), or phase transitions between disordered and ordered phases are induced (microseconds in liquid crystals), etc. None of these effects necessarily must have a linear effect on the polarization. However, in practice, virtually all nonlinear effects on the polarization are weak, so (assuming an instantaneous polarization response) a complete description of the nonlinear polarization is possible via a Taylor series:

$$\mathbf{P} = \epsilon_0(\chi^{(1)}\mathbf{E} + \chi^{(2)}\mathbf{EE} + \chi^{(3)}\mathbf{EEE} + ...). \tag{3.17}$$

The coefficients of this Taylor Series are tensors of increasing order. The coefficient of the linear term is a tensor of second-order, $\chi^{(1)}_{ij}$, the coefficient of the quadratic term is a tensor of third-order, $\chi^{(2)}_{ijk}$, etc. This description of the nonlinear polarization gives rise to a description of the linear (Pockels) and nonlinear (Kerr) electro-optic effect, which are of crucial importance for integrated photonic devices. While the description of the nonlinear polarization gives the most physical intuition about the origin of electro-optic effects, measurements of the nonlinear polarization coefficients are most often reported in another form in the literature.

It is customary to describe the refractive index of a material by an "index ellipsoid", that is a geometric representation of the quadratic form which appears in the anisotropic wave

equation:

$$n^2 = \epsilon_{ij} k_i k_j, \tag{3.18}$$

where ϵ_{ij} is the permittivity tensor. Often a slightly different description is chosen, by using the inverse of the permittivity tensor, the impermeability tensor:

$$\eta_{ij} = \epsilon_0 \epsilon_{ij}^{-1}. \tag{3.19}$$

Due to the nonlinear polarization described above, the impermeability tensor is a function of the applied electric field:

$$\eta_{ij}(\mathbf{E}) = \eta_{ij} + r_{ijk} E_k + s_{ijkl} E_k E_l + ... \tag{3.20}$$

It is the elements of the r-tensor (Pockels) and s-tensor (Kerr) that are most often reported in the literature for electro-optic effects. There are certain symmetry constraints that constrain the degrees of freedom in Pockels- and Kerr-tensor, and usually a different indexing of tensor elements using only two instead of three indices for the Pockels tensor is used. The most important tensor elements for this work are the $r_{33} = r_{333} \approx 30$ pm/V and $r_{13} = r_{113} \approx 10$pm/V elements of lithium niobate. The change of the refractive index is then determined by

$$\Delta(\frac{1}{n^2})_i = r_{ij} E_j, \tag{3.21}$$

or, using a Taylor expansion to find the change of Δn directly:

$$\Delta n_i \approx -\frac{n^3}{2} r_{ij} E_j. \tag{3.22}$$

As will be explained in the next chapter, the combination of a spatially-configurable electric field gives rise to index perturbations according to Eq. 3.22, which in turn give rise to programmable wave propagation determined by Eq. 3.15. Such programmable wave propagation can be exploited for optical computing and offers a promising alternative to the approaches reviewd in the next section.

CHAPTER 4

A PHOTONIC PROCESSOR BASED ON ARBITRARILY PROGRAMMABLE WAVE PROPAGATION

This chapter is a reprint of Onodera, T. *et al.* Scaling on-chip photonic neural processors using arbitrarily programmable wave propagation. *arXiv* (2024), of which I was a co-first author. The text has been slightly upgraded since submission to the arXiv.

4.1 Introduction

Deep neural networks (DNNs) have gained widespread adoption across many domains ranging from computer vision to natural language processing [99]. The size of DNN models has been increasing exponentially over the past decade, leading to exponentially increasing energy costs for running them. Limits to energy costs now impose a practical constraint on how large models can be [6], strongly motivating the exploration of alternative, energy-efficient computing approaches for executing DNNs, whose computational cost is typically dominated by that of matrix-vector multiplications (MVMs). Optical neural networks (ONNs) that specialize in performing MVMs with optics instead of electronics are one promising candidate approach [62, 84, 88, 100].

Integrated photonics is a leading platform for optical neural networks due to its compact form factor, excellent phase stability, availability of high-bandwidth modulators and detectors, manufacturability, and ease of integration with electronics [77, 78, 84, 88, 91, 101, 102]. The dominant paradigm for designing integrated photonic neural networks is to construct networks of discrete, programmable photonic components—such as Mach–Zehnder interferometers, microring resonators, or phase-change-memory cells—connected by single-mode waveguides [62]. These networks execute a linear-optical operation: an MVM between the vector encoded in the optical input to the chip and the matrix programmed in the discrete components. However, the maximum vector size, N, supported by chips using this approach

has so far been restricted (Supplementary Table 1) to sizes far below what is necessary for optics to deliver an energy-efficiency advantage ($N \gtrsim 1000$) [68, 96, 103, 104]. The scale of such chips has been limited by at least two factors: (1) the large spatial footprint of individual components and the inefficiency of dedicating a substantial portion of the chip's area to non-programmable interconnection regions comprising well-isolated waveguides that connect relatively sparsely arranged programmable elements, and (2) the systems-integration complexity of controlling each discrete component with electronic wires carrying the trainable parameters of the neural network.

We could achieve far greater spatial efficiency [94, 105] if, instead of building the integrated photonic neural network from discrete components, we treated the entire chip as a blank slate that we could arbitrarily and reprogrammably sculpt. Here lies the central challenge that our work tackles: for such a chip to perform an MVM with a programmable matrix, we need to be able to continuously program the chip's refractive-index distribution, $n(x, z)$ [81, 93–95, 105, 106]. How can we make a photonic chip whose refractive-index distribution is programmable, ideally in a way that avoids the integration complexity of introducing electronic wiring? In conventional nanophotonic chips, $n(x, z)$ is controlled by etching away material in lithographically defined regions—and is fixed at fabrication time. While inverse-designed chips [107] realizing MVMs with fixed matrices can be made [81], we generally would like to be able to program the matrix. Photorefractive crystals were explored several decades ago as a means to implement programmable linear operations with slab waveguides [61, 108], but fell out of favor because the small achievable refractive-index modulation (10^{-4}) meant that even centimeter-scale waveguides were unable to perform large-scale operations. Additionally, phase-change materials have recently been demonstrated to realize arbitrary refractive-index distributions [109, 110] but suffer from a limited number of rewrite cycles (4000 in ref. [111]) and high loss (greater than 2.8 dB/mm in ref. [110, 111]). The scale is also currently limited: ref. [110] reported programming a device with a 3-dimensional input and a 3-dimensional output.

In this work, we introduce a photonic chip with a waveguide that is fully programmable in two

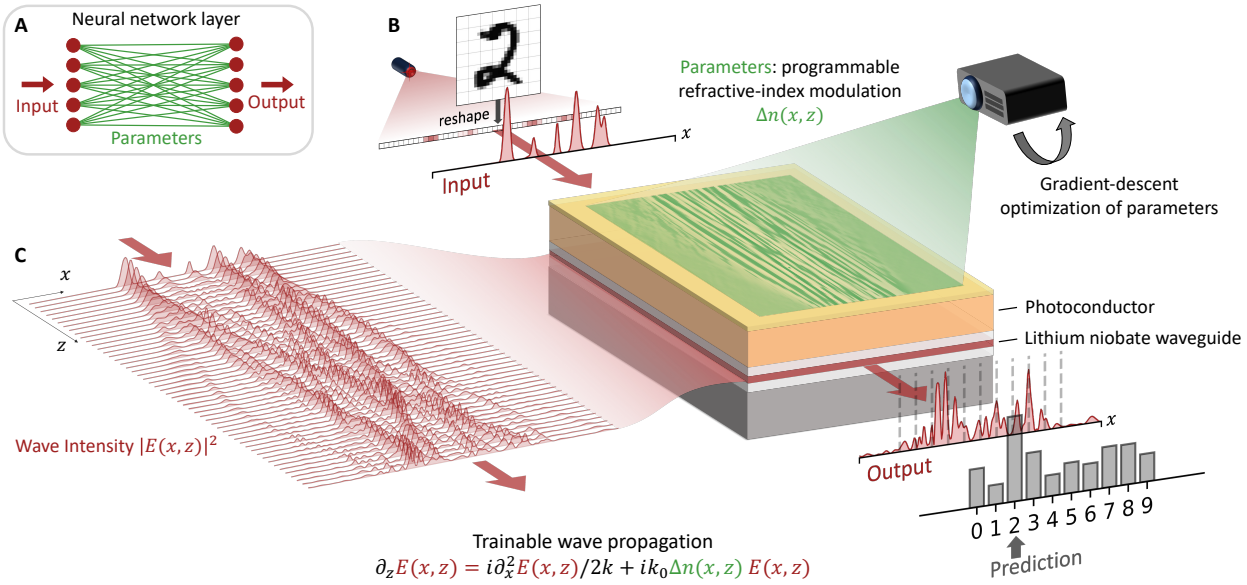

Figure 4.1: **Machine learning with multimode wave propagation in the 2D-programmable waveguide.** (**A**) The fundamental unit of an artificial neural network—a *layer*—transforms an input vector into an output vector via a trainable matrix multiplication. (**B**) Analogous to a neural-network layer, the 2D-programmable waveguide linearly transforms an input optical field into an output optical field, via wave propagation through a lithium niobate slab waveguide whose two-dimensional refractive-index modulation $\Delta n(x, z)$ can be continuously and arbitrarily programmed (up to practical limits on resolution and the maximum modulation; see Supplementary Sections 1C and 1D). This refractive-index modulation, which is directly set by the illumination pattern that is projected onto the device (shown in green), is trained so that wave propagation through the waveguide performs machine-learning tasks (handwritten-digit classification shown here as an example). To determine the result of the classification, the output beam's intensity is measured across equally-sized bins; the bin with the highest total power corresponds to the predicted label. (**C**) Simulated wave intensity in the slab waveguide, which shows that the neural-network computation is performed with complex multimode wave propagation.

dimensions: a *2D-programmable waveguide*. The chip uses massively parallel electro-optic modulation to program $n(x, z)$ across ∼10,000 individual regions of a lithium niobate slab waveguide, and we train multimode photonic structures within the chip that perform neural-network inference (Fig. 4.1). The structures realized by our 2D-programmable waveguide are similar to inverse-designed nanophotonic devices [81, 107]: they are computer-optimized, two-dimensional metastructures that control multimode wave propagation. A distinguishing feature of our device is its programmability, setting it apart from typical inverse-designed photonic devices, which are fixed after manufacturing. We achieve programmability optically,

decoupling the electronic wiring for programming from the photonic chip: a pattern of light shone on top of our device creates a spatially-varying refractive-index modulation $\Delta n(x, z)$ in the slab waveguide. This is achieved by using the principle of photoconductive gain [112, 113] to induce a refractive-index modulation via the strong electro-optic effect in lithium niobate. In our work we show how we can train the refractive-index distribution so that the complex wave propagation through the device performs a desired neural-network inference (Fig. 4.1C).

4.2 Operating principle of the device

In Fig. 1, we show a conceptual schematic for how to perform machine learning with the 2D-programmable waveguide. We amplitude-encode the machine-learning input data in the 1D input optical field distribution, $E(x, z = 0)$, which serves as the initial condition for programmable wave propagation described by the partial differential equation

$$\frac{\partial E(x,z)}{\partial z} = \frac{\mathrm{i}}{2k}\frac{\partial^2 E(x,z)}{\partial x^2} + \mathrm{i}k_0 \Delta n(x,z) E(x,z). \tag{4.1}$$

Here, z denotes the propagation direction and x the transverse direction, while k_0 and k are the wavenumbers in free space and the slab waveguide, respectively. In the slab waveguide, there is a refractive-index distribution $n(x, z) = n_0 + \Delta n(x, z)$ that has two contributions: a spatially uniform part, n_0, that is the refractive index of the waveguide when no programming light is impinging on it, and a programmable part that is induced by electro-optic modulation, $\Delta n(x, z)$. After the optical field propagated through the device, we measured its intensity, $I_{\mathrm{out}}(x) \propto |E(x, z = L)|^2$, at the output facet and binned it to produce the machine-learning task's output vector.

In Fig. 2, we show how patterns of light program the refractive-index modulation $\Delta n(x, z)$. Our device, which is inspired by optoelectronic tweezers enabled by photoconductive gain [112, 113], is composed of a lithium niobate slab waveguide and a photoconductive film, which are sandwiched between a pair of electrodes that have an oscillating bias voltage placed across

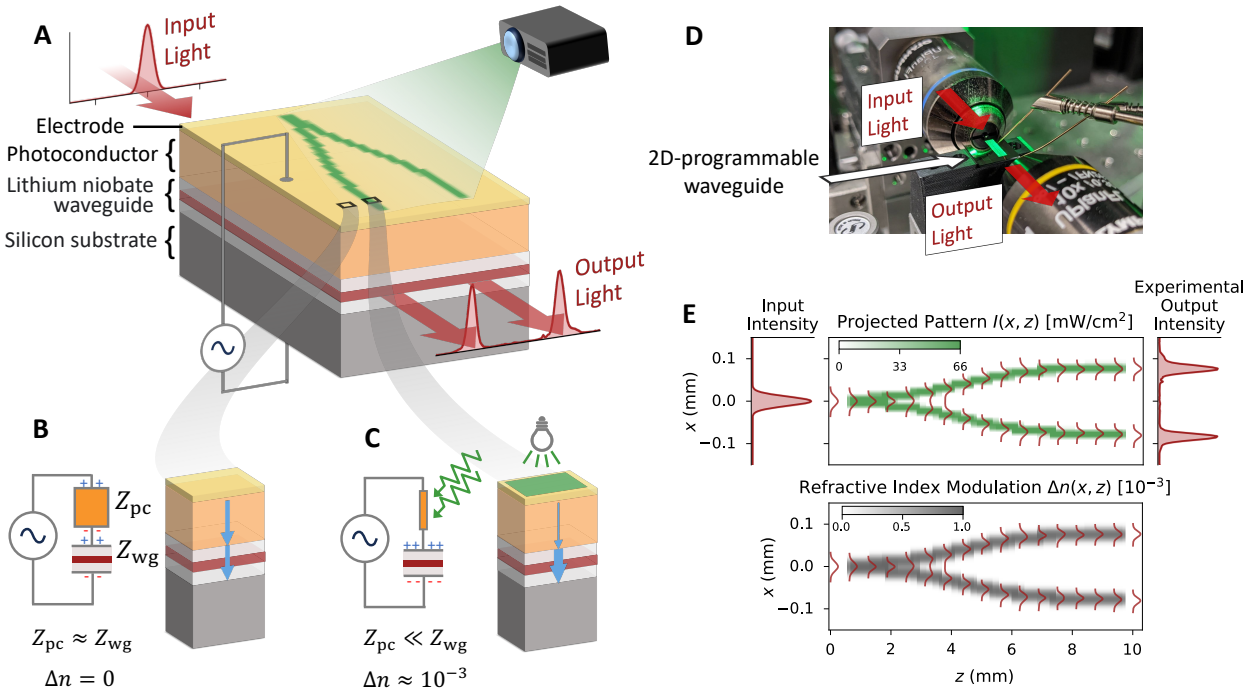

Figure 4.2: **Operating principle of the 2D-programmable waveguide.** (**A**) The 2D-programmable waveguide consists of a nanophotonic stack of four layers: (1) A conductive silicon substrate that doubles as the ground electrode, (2) a Z-cut lithium niobate (in red) slab waveguide with silicon dioxide cladding (in white), (3) a photoconductive layer for optical control of the refractive index, and (4) a gold electrode. (**B-C**) Electrical-circuit models of the 2D-programmable waveguide in regions with and without illumination. There is a voltage division between the photoconductor and the slab waveguide, with impedances Z_{pc} and Z_{wg}, respectively. (**C**) Upon illumination, the resistance of the photoconductor decreases, leading to an increase in the electric field (blue arrows) inside the waveguide, which induces a refractive-index modulation via the electro-optic effect. (**D**) A photograph of our prototype 2D-programmable waveguide in our experimental setup. (**E**) Experimental realization of a Y-branch splitter on the 2D-programmable waveguide, which splits the input light into two equal output beams. The projected pattern (in green) directly corresponds to the induced refractive-index modulation (in gray). A simulation of the wave propagating through the device is overlaid with the patterns (in red).

them (Fig. 2A). Projecting a light pattern onto the chip causes the electric field (from the bias voltage) in the slab waveguide to vary spatially according to the projected pattern. As shown in Fig. 2B and 2C, this results from the voltage division between the photoconductor and the slab waveguide. For regions of the chip that are illuminated at intensities of tens of mW/cm^2, the photoconductor's impedance drops substantially, increasing the bias electric field within the slab waveguide. Combined with lithium niobate's strong electro-optic effect,

this spatially varying bias electric field induces a spatially varying refractive-index modulation. In our device, the largest refractive-index modulation is approximately 10^{-3}, limited by the geometry of the material stack and a safety margin to prevent dielectric breakdown. In principle, it can be improved to 10^{-2}, a limit set by lithium niobate's breakdown field [114]. We note that unlike conventional etched nanophotonic structures, the refractive-index modulation in our device can take on continuous values by continuously varying the intensity of the projected pattern. For more details on the device fabrication and design, see Methods and Supplementary Section 1.

We projected illumination patterns across a 9 mm × 1 mm area, achieving a pixel resolution of 9 µm × 9 µm. This configuration enabled us to control the refractive-index distribution $n(x, z)$ with 10,000 degrees of freedom (Supplementary Section 3A) and update the entire distribution at a rate of 3 Hz. To maximize the refractive-index modulation, we set the amplitude of the oscillating bias voltage across the electrodes to be up to 1000 V. Given that CMOS electrode backplanes can only support spatially programmable voltages of around 10 V, our approach of using photoconductive gain is crucial to achieving large electro-optic modulation: it allows us to apply a large voltage to a single unpatterned electrode, and realize controllable high voltages at *virtual electrodes* via the patterned illumination [112, 113].

To illustrate the operating principle of our device, we projected a pattern in the shape of a Y-branch splitter onto the 2D-programmable waveguide (Fig. 2A). The refractive-index modulation was approximately proportional to the projected pattern $\Delta n(x, z) \propto I(x, z)$ (Fig. 2E), up to spatial smoothing and a weak nonlinearity due to saturation of the photoconductor. Therefore, the projected pattern instantiated a refractive-index distribution of a Y-branch splitter. We coupled a single input Gaussian beam into the device using a beamshaper and measured the intensity of the output light with a camera. For further details on the experimental setup, see Methods. The experimental result in Fig. 2E shows that the input beam was split into two equal output beams, in agreement with the simulated wave propagation.

4.3 Machine-learning demonstrations with the 2D-programmable waveguide

We next applied the 2D-programmable waveguide to performing machine-learning tasks: vowel classification [46] and MNIST handwritten-digit classification [115]. Both tasks are used as benchmarks in studies of similar on-chip optical neural networks, providing useful points of comparison [78, 88, 95].

The vowel-classification dataset [46] comprises formant frequencies extracted from audio recordings of spoken vowels by various speakers. The task is to predict which of the 7 vowels is spoken, given a 12-dimensional input vector of formant frequencies. We divided the dataset into a training set and a test set, comprising 196 (75% of the dataset) and 63 (25%) samples, respectively.

In Fig. 3, we present our experimental results on performing vowel classification with the 2D-programmable waveguide. As shown in Fig. 3A and 3B, we encoded the 12-dimensional input vectors in the amplitudes of 12 spatial Gaussian modes (whose center positions were equally spaced) at the input facet of the device. This input optical field distribution, which was produced by the beamshaper, underwent complex wave propagation in the 2D-programmable waveguide. For readout, we measured the intensity at the output facet with a camera and binned the camera pixels into 7 different regions, with each region corresponding to a specific vowel. The predicted vowel for an input is given by the region that receives the most optical power. Thus, the refractive-index distribution was trained so that the wave propagation directed the most power toward the region corresponding to the correct vowel. For more details on the output decoding and the overall computational model of our ONN demonstrations, see Supplementary Section 5A. As shown in the simulated intensity distribution $|E(x, z)|^2$ in Fig. 3B, the device learned to use complex multimode wave propagation to perform the neural-network inference.

The refractive-index distribution to implement vowel classification was learned using *physics-*

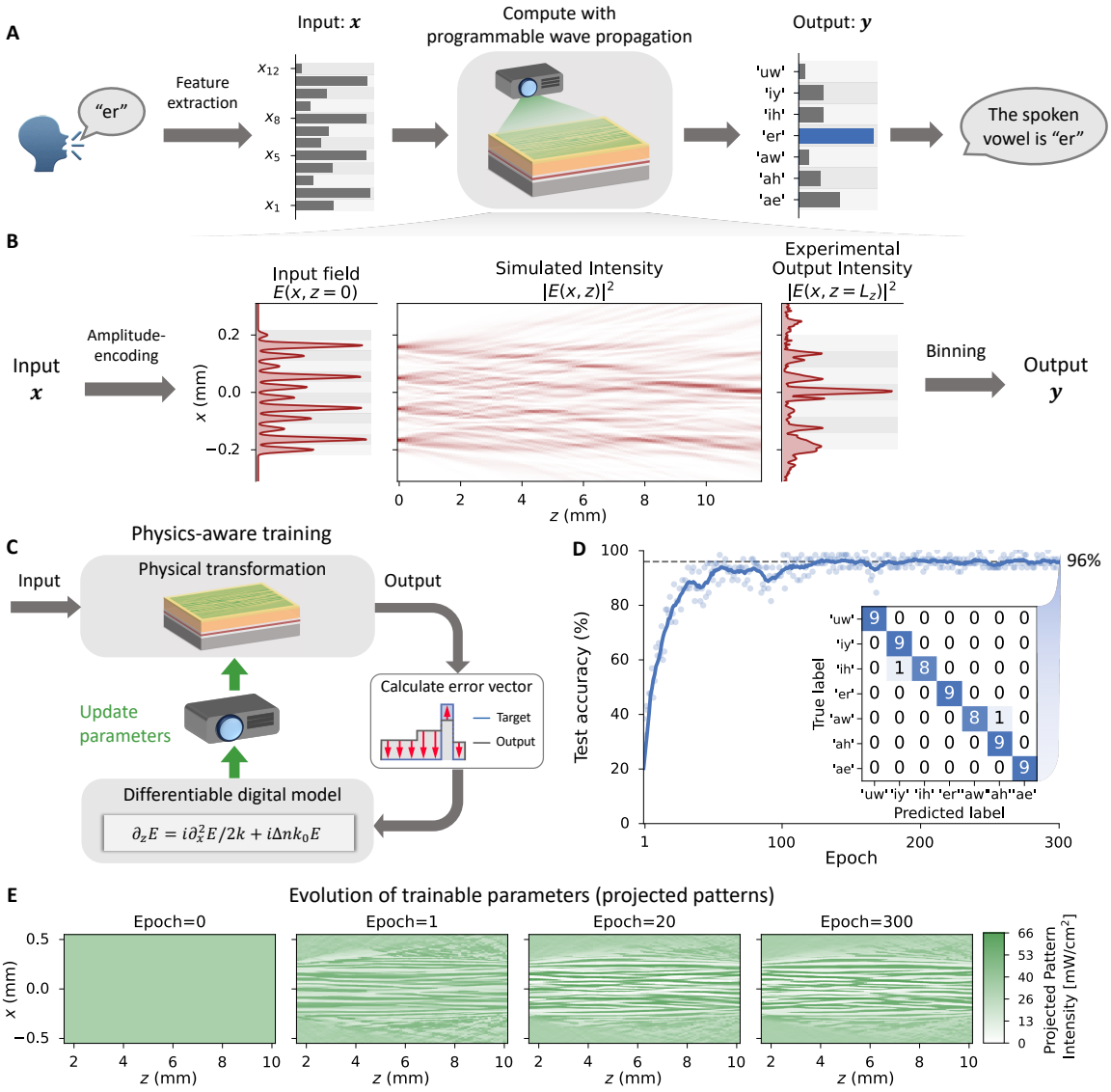

Figure 4.3: **Vowel classification with the 2D-programmable waveguide.** (**A**) Overview of approach: The task involves predicting a spoken vowel, here "er", from a 12-dimensional input vector representing formant frequencies extracted from audio recordings. The 2D-programmable waveguide was trained to take in this input vector and output a 7-dimensional vector with a one-hot encoding format that indicates the predicted vowel. (**B**) Left: The input vector was amplitude-encoded into 12 Gaussian spatial modes to produce the initial optical field distribution. Center: Simulated wave propagation in the chip after training of the projected pattern. Right: The experimentally measured output intensity. It was binned, i.e., the total power within equally-spaced spatial bins was calculated to produce the 7-dimensional output vector. (**C**) Illustration of physics-aware training, a hybrid in-situ–in-silico backpropagation algorithm, which we used to train the parameters of the 2D-programmable waveguide. (**D**) Test accuracy as a function of epoch. The inset shows the confusion matrix on the test dataset after training. (**E**) Evolution of the trainable parameters, the projected patterns, at different stages of training.

aware training [116], a modified backpropagation algorithm (Fig. 3C). In this algorithm, the forward pass is performed by the experimental setup, while the backward pass is computed with a digital model of the experiment. The hybrid in-situ–in-silico nature of the algorithm allows for efficient training even in the presence of both imperfect models and experimental noise (Supplementary Section 3C). The digital model was challenging to construct due to the large number of parameters and the complexity of the wave propagation. Initially, a purely physics-based model (using Eq. 4.1 governing the wave evolution) provided qualitative but not quantitative agreement with the experimental results. This discrepancy led us to integrate data-driven refinements to the physics-based model, which then achieved quantitative agreement with the experiment (Supplementary Section 4).

Using physics-aware training, we trained the 2D-programmable waveguide for a total of 300 epochs, which took approximately one hour on the experimental setup (see Fig. 3D). As shown in Fig. 3E, the projected pattern, which was initialized as a uniform illumination, evolved into a complex pattern that is challenging to interpret (see Supplementary Fig. 15 for the corresponding evolution of the wave propagation). These patterns resemble the refractive-index distributions found in inverse-designed photonic devices; this is expected as we share the same principle of using computer optimization to design/train the device to perform a desired function. Fig. 3D shows that despite the complexity of the projected pattern, the 2D-programmable waveguide successfully performed the vowel-classification task, and achieved a test accuracy of 96% after training.

In Fig. 4, we present our experimental results on MNIST handwritten-digit classification. The task consists of classifying 14-by-14-pixel images of handwritten digits from 0 to 9. We divided the MNIST dataset in the standard manner into 60,000 training images and 10,000 test images. We down-sampled each MNIST image to 7-by-7 pixels, then reshaped them to 49-dimensional input vectors.

To train a refractive-index distribution to perform MNIST classification, we followed the same procedure used for the vowel-classification task (Fig. 3): the 2D-programmable waveg-

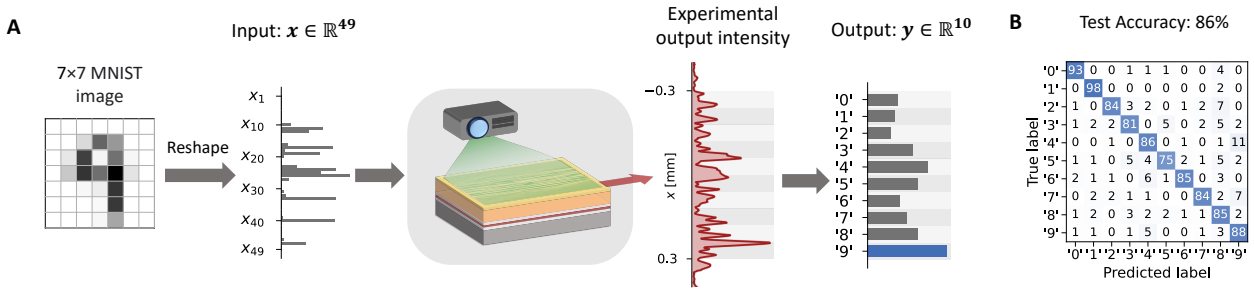

Figure 4.4: **MNIST handwritten-digit classification with the 2D-programmable waveguide: neural-network inference with high-dimensional input vectors.** (**A**) We performed MNIST handwritten-digit classification with the 2D-programmable waveguide. Each image from the MNIST dataset was electronically downsampled and reshaped to a 49-dimensional vector. We trained the device to perform machine learning on this high-dimensional input vector with the same procedure as the vowel-classification task (see Fig. 3). (**B**) The confusion matrix, derived from evaluating on the test dataset of 10,000 images. After 10 epochs of training, the system achieved 86% accuracy on the test dataset. As a baseline, a single-layer digital neural network with a 49×10 matrix achieves 90% accuracy on this same task.

uide processed the 49-dimensional input vector to produce a 10-dimensional output vector that corresponds to the 10 possible digits (see Supplementary Section 5C for more details on the MNIST experiments). As shown in Fig. 4B, the system achieved 86% accuracy on the test dataset after 10 epochs of training, which took about 10 hours on the experimental setup. This falls 4 percentage points short of the 90% accuracy that a one-layer digital neural network achieves on this downsampled MNIST classification task, likely due to imperfect modeling and experimental drifts. Nevertheless, this result demonstrates that complex wave propagation in our device can be harnessed to perform computations comparable to that of a single-layer neural network with a 49×10 matrix of trainable parameters.

4.4 Discussion and outlook

We have introduced and demonstrated a 2D-programmable photonic processor comprising a lithium niobate slab waveguide whose refractive-index distribution, $n(x, z)$, can be continuously programmed. The device design enables programming by massively parallel electro-optic modulation with approximately 10,000 degrees of freedom. We used our chip

to perform neural-network inference by training the refractive-index distribution and consequently the multimode wave propagation through the chip. To train the device, we developed a physics-based model of the chip's behavior, along with a data-driven refinement allowing the model to be sufficiently accurate that it supports backpropagation-based training [116].

The predominant approach to building integrated photonic neural networks is to fabricate large arrays of discrete components connected by single-mode waveguides [62]. In contrast, we adopted the conceptual approach of using wave propagation in distributed spatial modes [61, 81, 93, 94, 105, 106, 117], and experimentally validated the theoretical predictions[94, 105, 117] that this approach will be more space-efficient. Our prototype chip was able to perform neural-network inference with input vectors of dimension up to 49, which is larger than the capability of the neural-network photonic chips reported in refs. [53, 77, 78, 83, 88, 91, 95, 118], and more space efficient than any of these chips based on networks of discrete components (see Supplementary Section 6 for a detailed comparison). This large input dimension enabled us to use our chip to perform MNIST handwritten-digit classification with a single pass through the chip, and without using any digital-electronic parameters.

Looking to the future, relatively modest device improvements would allow us to scale to even larger (vector dimension $N \geq 100$) photonic computations (Supplementary Section 7E). One could also periodically pole the waveguide lithium niobate, enabling strong nonlinear-optical interactions between the propagating waves. The resulting nonlinear wave propagation would enable the on-chip realization of a variety of proposed wave-based deep and recurrent neural networks[93, 106, 119, 120] (Supplementary Section 7A). An even more sophisticated modification of the device would be to not just introduce nonlinearity to the wave propagation, but to make the nonlinearity arbitrarily programmable, by allowing the second-order nonlinear optical susceptibility distribution $\chi^{(2)}(x, z)$ to be modified in real time (Supplementary Section 7C).

To conclude, we believe that our device concept, with its ability to programmably control multimode wave propagation, may create new opportunities in the broader fields of optical

computing and optical information processing [62, 68, 100]. Although our work in this paper has focused on machine learning, our device could also be used to solve integral equations [121] and combinatorial-optimization problems [122]. More broadly, our chip is essentially an arbitrary (passive) photonic device that can be reconfigured on demand: any photonic device that can be specified as an inhomogeneous refractive-index distribution can be realized. Such devices can even be learned directly—effectively by performing inverse design [107], but in situ in real time. The class of reprogrammable photonic devices that our concept enables is well-suited to applications where a single device must adapt to different conditions [123], such as in smart sensing [91], radio-frequency communications [124, 125], and dynamic routing in optical interconnects [126]. It may ultimately even be possible to make a device that combines programmable linear wave propagation (this work), programmable nonlinear wave propagation (a natural extension of this work to having programmable $\chi^{(2)}(x, z)$), and programmable gain/loss (demonstrated in ref. [95]), giving rise to a reconfigurable on-chip platform capable of realizing almost every functionality we have in free-space optics.

4.5 Methods

4.5.1 Device fabrication and design

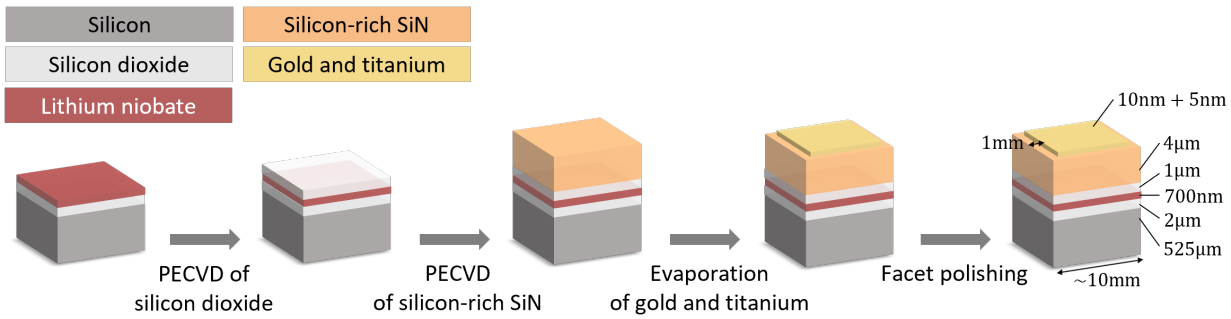

Figure 4.5: **Fabrication process and device geometry of the 2D-programmable waveguide.**

We started our fabrication processes from a thin-film lithium niobate wafer purchased from

NanoLN. It was a p-type silicon wafer with a substrate conductivity of 0.01-0.05 Ω cm, 2 µm of silicon dioxide deposited via plasma-enhanced chemical vapor deposition (PECVD), and 700 nm of Z-cut MgO-doped lithium niobate that is wafer-bonded with the ion-cut technique. We diced small pieces from the wafer using a Disco Dicing Saw for further processing. We deposited an additional 1 µm of silicon dioxide via PECVD as a cladding, followed by another deposition of 4 µm of silicon-rich silicon nitride (SRN), which is the photoconductive layer, via PECVD. The SRN layer was deposited in an Oxford Plasmalab 100 by flowing 40 sccm of SiH_4, 10 sccm of N_2O, and 1425 sccm of N_2 into the deposition chamber at a temperature of 350 °C and a pressure of 1900 mTorr. We alternated pulses of high and low frequency power during deposition to minimize film stress, with 160 W, 12 s low frequency pulses and 200 W, 8 s high frequency pulses.

Next, we evaporated electrodes onto the chip using a CVC SC4500 E-gun Evaporation System. We first evaporated 10 nm of titanium as an adhesion layer, then 5 nm of gold. To prevent dielectric breakdown between the top electrode and the conductive substrate through air at the edges of the chip, we covered the perimeter of the chip with tape before evaporation. The tape acted as a mask, preventing deposition closer than around 1 mm to the edges, thereby increasing the path length between the top electrode and substrate through air. To minimize coupling losses into the waveguide, we used an Allied Multiprep Polisher to polish the waveguide facets. We polished using silicon carbide paper of successively finer grain size, starting at 3 µm, then moving to 1 µm, and 0.5 µm roughness.

Table 4.1: **Processing steps for 2D-programmable waveguide fabrication** with instructions specific to tools in the Cornell Nanoscale Science & Technology Facility (CNF). These processing steps worked in December 2022, but might not continue to work indefinitely due to tool changes. Diagnostic steps to check the success completion of every step are included where they are deemed necessary.

Processing step	**Tool**	**Steps**
Preparation	Fume hood	• Clean wafer as necessary with Acetone, Methanol, & IPA.
Upper cladding deposition	Oxford Plasmalab 100	• Season the chamber via 2min of "SiO2 (high rate)" recipe with default parameters in an empty chamber. • Deposit 1um of SiO2 cladding via 4min12s of "SiO2 (high rate)" with default parameters. • Clean the chamber via 17 minutes of "high rate clean" recipe with default parameters in an empty chamber. • (Optional Diagnostic) Measure the index and thickness of the silicon dioxide layer with the Woollam RC2 Spectroscopic Ellipsometer, using "SiO2" layer.

Processing step	**Tool**	**Steps**
Photoconductor deposition	Oxford Plasmalab 100	• Season the chamber via 2min of "SiO2 (high rate)" recipe with default parameters in an empty chamber. • Deposit 2um photoconductor layer via 28min of "SiN_x (low stress)" recipe with SiH_4 flow of 40sccm and NH_3 flow of 10sccm. All other parameters remain at default. • Clean the chamber via 40 minutes of "high rate clean" recipe with default parameters in an empty chamber. • Season the chamber via 2min of "SiO2 (high rate)" recipe with default parameters in an empty chamber. • Deposit 2um photoconductor layer via 28min of "SiN_x (low stress)" recipe with SiH_4 flow of 40sccm and NH_3 flow of 10sccm. All other parameters remain at default. • Clean the chamber via 40 minutes of "high rate clean" recipe with default parameters in an empty chamber. • (Optional Diagnostic) Measure the index and thickness of the silicon dioxide layer with the Woollam RC2 Spectroscopic Ellipsometer. We have obtained good fits using "Cauchy" layer with parameters $A = 2.252$, $B = 0.04230$, and $C = 0.00759$.

Processing step	**Tool**	**Steps**
Electrode deposition	CVC SC4500 E-gun Evaporation System	• Create a mask with appropriately sized holes for electrodes with Kapton polymide tape. To prevent dielectric breakdown through air along the edges of the device, space the holes such that after dicing the wafer, the electrodes will be at least 1mm away from any device edge. • Deposition of 10nm of titanium as an adhesion layer. • Deposition of 5nm of gold. • Carefully remove tape and any residues after deposition. • (Optional Diagnostic) Test that the resistance between any two points on the electrode is no more than 10s of Ohms.
Cutting wafer into individual devices	Disco Dicing Saw	• Dice wafer into individual pieces ensuring the electrode cut line is at least 1mm away from any device edge (we used 1.5mm to account for material removal during polishing). • A silicon-only blade is recommended to make polishing easier, but not necessary.

Processing step	**Tool**	**Steps**
Polishing of front- and back-facets	Allied Multi-prep Polisher	• If all-purpose blade was used in Dicing Saw (as opposed to a silicon-only blade), polish both facets using silicon carbide paper of successively finer grain size starting with a 30um grain size, then 15um, then 6um. • Finish polish using silicon carbide paper of successively finer grain size, starting at 3 um, then moving to 1 um, and 0.5 um roughness. • For all polishing steps, remove at least three times the grain size in material, i.e. for a 3um grain size paper remove at least 9um of material over a span of about one minute. • While polishing, keep polishing paper lubricated by spraying green lube about every 20 seconds (or when necessary). • (Optional Diagnostic) In between polishing steps, inspect facets with microscope for chipped off material, especially the photoconductor layer. If pieces of the photoconductor layer have chipped off, repeat the same polishing step with lower pressure until a smooth surface is created again. • After finishing all polishing steps on one facet, turn device around and polish other facet. • Clean wafer as necessary with Acetone, Methanol, & IPA. • Finish by dipping a cotton swab into rinse-aid and lightly wiping the polished facets with it. Then rinse with DI water and blow-dry.

Device design and characterization

We used Z-cut lithium niobate for the slab waveguide, so that the crystal axis of the lithium niobate is parallel to the strong electric field that is induced by the bias voltage, which points out of the waveguide plane (in the y direction). As shown in Supplementary Fig. 1, we used the transverse magnetic (TM) mode of the slab waveguide, whose optical electric field is also oriented in the y direction. Since the r_{33} electro-optic coefficient is largest in lithium niobate, this configuration maximized the strength of the electro-optic modulation.

The thickness of the lithium niobate layer is chosen for single-mode operation (see Supplementary Section 1A). To maximize the refractive-index modulation, it is beneficial to have a thicker photoconductor and a thinner silicon dioxide cladding (see Supplementary Section 1B). The silicon dioxide cladding is chosen to be sufficiently thick to ensure low propagation loss. Thus, we balanced these tradeoffs to arrive at the device geometry shown in Extended Data Fig. 1. The device has a propagation loss of less than 1 dB/cm at a wavelength of 1550 nm (see Supplementary Section 1E).

In order to maximize the refractive index contrast of the slab waveguide between the bright (illuminated) and dark regions, it is important to design the photoconductor to have high dark resistance and low bright resistance. Thus, to optimize the material properties of the photoconductor, we swept the silicon to nitrogen ratio in the silicon-rich silicon nitride photoconductor (by varying the amount of SiH_4 gas we flowed into the PECVD) and chose the material with the largest photoconductive contrast. We characterized the refractive-index modulation as a function of the intensity of the projected pattern with an off-axis holography setup (see Supplementary Section 1C). The maximum refractive index modulation that we achieved in this work is approximately 10^{-3}. We show in Supplementary Fig. 3 that this can be increased to beyond 4×10^{-3} by using a photoconductor layer that is twice as thick (8 µm) and by further optimizing the photoconductive properties. In Supplementary Section 7B, we also discuss how the refractive-index modulation can be further increased by switching to a different material for the waveguide core.

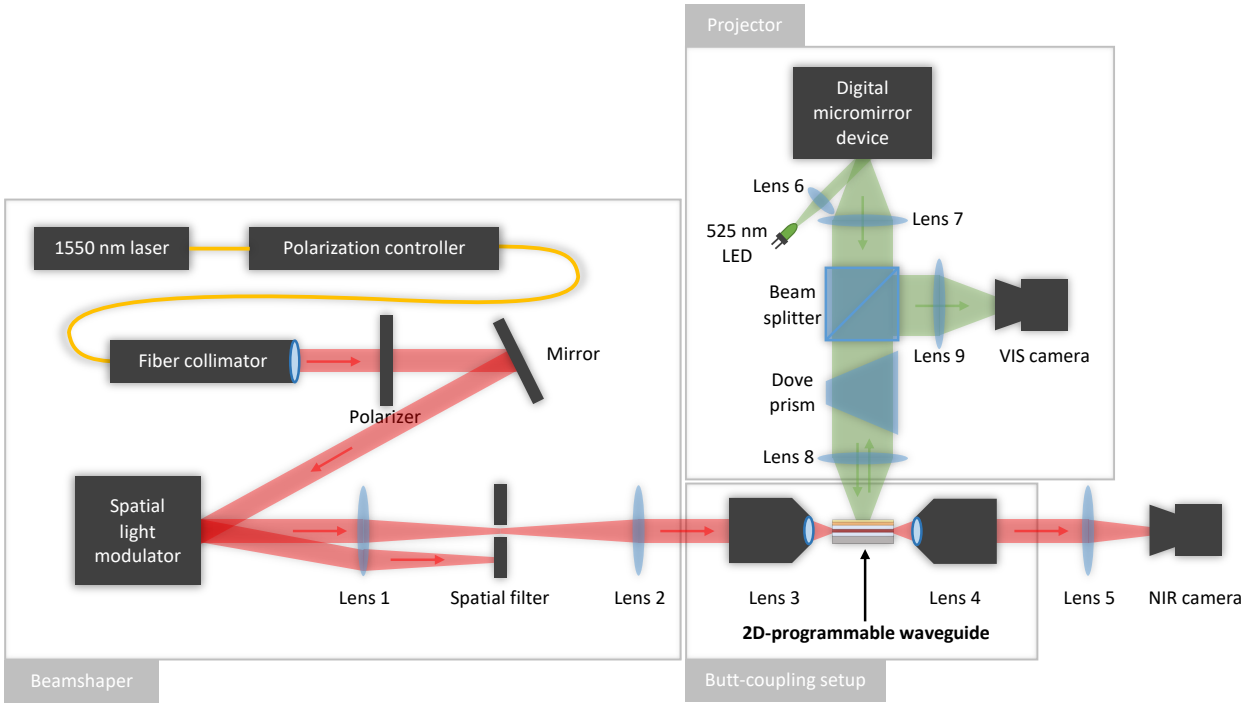

Figure 4.6: **Schematic of the experimental setup.**

4.5.2 Experimental setup

The experimental setup can be roughly divided into five units: 1) An optical beamshaper to create spatially-varying one-dimensional electric field inputs for the 2D-programmable waveguide, 2) a projector to create a programmable illumination pattern that controls the refractive-index distribution inside the waveguide, 3) a butt-coupling setup to couple light in and out of the 2D-programmable waveguide, 4) a high-voltage source to apply an oscillating bias voltage across the electrodes of the 2D-programmable waveguide, and 5) a camera to measure the intensity of the output beam. We note that the experimental setup relies on more free-space optical components than usual for an integrated photonics experiment. This is a direct consequence of our decision to keep the fabrication of the device simple, without lithographically defined structures for this initial proof-of-concept demonstration. We envision that a more compact, fully integrated version of the 2D-programmable waveguide could be built by integrating on-chip lithium niobate modulators and detectors, and a micro-LED display (see Supplementary Section 7D).

In this section, we provide an overview of the key components and functionalities of the experimental setup. For a more detailed description, including photographs of the optical setup and specifics on the components such as part numbers and manufacturers, see Supplementary Section 2.

The free-space beamshaper allows for the realization of arbitrary input optical fields $E(x, z = 0)$, up to a spatial resolution of 2 µm and over a distance of 600 µm. In this experiment, we used the beamshaper to create both simple input fields, such as a single Gaussian beam for the Y-branch splitter demonstration, and more complex input fields for the machine-learning demonstrations. This flexibility also enabled us to freely vary the encoding of input vectors into the optical field. For instance, we varied the width of the input modes and adjusted their spacing, which is tailored to each machine learning task. Finally, because the beamshaper is capable of shaping both the amplitude and phase of the input field, it was also used to calibrate the 2D-programmable waveguide (see Supplementary Section 4B).

The design of the beamshaper we built closely follows ref. [127], which also programmably shapes the input light that is coupled into slab waveguides. The core working principle of the beamshaper is to create spatially varying phase-gratings on a 2D-phase spatial light modulator [128] (SLM, Meadowlark Optics UHSP1K-850-1650-PC8). We varied the amplitude and relative positions of these phase-gratings to control the input optical field $E(x, z = 0)$. A lens after the SLM performed a Fourier transform that separates the diffraction maximums of the phase-gratings, and a spatial filter selected the first-order diffraction maximum. Finally, a 4f relay system (comprising lens 2 and lens 3 as shown in Extended Data Fig. 2) demagnified the optical field at the focal plane of the first lens and coupled the light into the 2D-programmable waveguide. Due to the response time of the liquid crystal in the SLM, the beamshaper's fastest update speed is approximately 50Hz.

The projector setup was designed to create a high-resolution programmable illumination pattern over a large field of view. We used a digital micromirror device (DMD, Vialux V-7000) with a resolution of 1024×768 pixels and a pixel pitch of 13.7 µm. The DMD was

illuminated with green light (525 nm) from an LED. We imaged the surface of the DMD onto the surface of the 2D-programmable waveguide via a 4f setup consisting of two tube lenses. The focal length of the tube lenses was chosen to demagnify the image of the DMD by a factor of 1.5, so the projected pattern on the surface of the 2D-programmable waveguide has dimensions of 9.1 mm × 6.8 mm, with each individual pixel of the projected pattern measuring 9 µm × 9 µm. Because the complex wave propagation spans a distance of 1 mm in the x direction, in practice, we only use a 9.1 mm × 1 mm region of the projected pattern to control the wave propagation in the 2D-programmable waveguide. Finally, although the DMD provides only binary modulation, we achieve continuous refractive-index modulation by pulse-width modulating the illumination pattern. This is feasible because the DMD can be switched on and off at a rate of 20 KHz, much faster than the RC time constant of the device, which is about 10 Hz.

In order to maximize the electro-optic effect in lithium niobate, we used high voltages of about 1 kV. We created sinusoidal voltages with an arbitrary function generator and amplified the voltage with a Trek 2220 high voltage amplifier, which has a voltage gain of 200× and is capable of outputting voltages of up to 2 kV. We electrically contacted the device using high-voltage-rated probe arms with BeCu probe tips: one probe tip was put in contact with the gold electrode on top of the device, while a grounded probe tip touched the silicon substrate (see Supplementary Figure 9).

We used an AC frequency of 10 Hz for the experiments shown in Fig. 2, and 26 Hz for the experiments shown in Fig. 3 and 4. Because AC voltage is applied, the desired refractive-index distribution is realized at the peak of the sinusoidal modulation. Thus, we trigger the camera to this peak, which explains the use of a higher frequency for the machine learning demonstrations—to maximize the update rate of the experiment. Finally, we opted for AC over DC operation because the resistivity of the silicon dioxide cladding exceeds the dark resistance of the photoconductor; under DC operation, the photoconductor would not modulate the circuit's overall impedance. Future modifications can enable DC operation, such as using an alternative cladding material that is more conductive or by increasing the

dark resistivity of the photoconductor (see Supplementary Section 1B).

To measure the output of the computation performed by our device, we imaged the output facet of the device with an infrared camera (Allied Vision Goldeye CL-033). We built a 4f relay with a magnification factor of 5.3, allowing us to image the intensity distribution at the output facet, $I_{\text{camera}}(x, y)$, with a resolution of 2.8 µm per pixel, and a field of view of 1.7 mm in the x direction. We defined a small range of y to be the region of interest and integrated the intensity over this range to obtain the 1-D intensity output of the 2D-programmable waveguide: $I_{\text{out}}(x) = |E(x, z = L)|^2 = \int_{y_{\min}}^{y_{\max}} I_{\text{camera}}(x, y)\mathrm{d}y$. We used an exposure time on the order of 500 µs, chosen to be much shorter than the AC voltage's period, which is approximately 40 ms.

4.6 Data availability

All data generated during this work are available at `https://doi.org/10.5281/zenodo.10775722`.

4.7 Code availability

All the code used for this work is available at `https://doi.org/10.5281/zenodo.10775722`.

4.8 Acknowledgements

We gratefully acknowledge the Air Force Office of Scientific Research for funding under Award Number FA9550-22-1-0378, and the National Science Foundation for funding under Award Number CCF-1918549. We thank NTT Research for their financial and technical

support. This work was performed in part at the Cornell NanoScale Facility, a member of the National Nanotechnology Coordinated Infrastructure (NNCI), which is supported by the National Science Foundation (Grant NNCI-2025233). P.L.M. acknowledges financial support from a David and Lucile Packard Foundation Fellowship. We acknowledge helpful discussions with Chris Alpha, Nicholas Bender, Jeremy Clark, Anthony D'Addario, Noah Flemens, John Grazul, Ryan Hamerly, David Heydari, Phil Infante, Mario Krenn, Kangmei Li, George McMurdy, Roberto Panepucci, Carl Poitras, Sridhar Prabhu, Aaron Windsor, and Yiqi Zhao.

4.9 Author contributions

T.O., L.G.W. and P.L.M. conceived the project. M.M.Stein, T.O., L.G.W. and P.L.M. designed the devices and experiments. M.M.Stein, T.O., B.A.A., and R.Y. performed the device fabrication with aid and recipe development from M.R.S., M.B., M.J., and T.P.M.. G.S. supervised M.R.S. and M.B.. M.M.Stein, T.O., M.M.Sohoni and T.W. designed and built the imaging setup to program the refractive-index patterns. M.M.Stein, T.O. built the high-voltage and beamshaper setups, performed the experiments, and analyzed the results. T.O., M.M.Stein, L.G.W. and P.L.M. wrote the manuscript with input from all authors. P.L.M. supervised the project.

CHAPTER 5

SUPPLEMENTARY MATERIAL–A PHOTONIC PROCESSOR BASED ON ARBITRARILY PROGRAMMABLE WAVE PROPAGATION

This chapter is a reprint of the supplementary material from Onodera, T. *et al.* Scaling on-chip photonic neural processors using arbitrarily programmable wave propagation. *arXiv* (2024), of which I was a co-first author. I reprinted the supplementary material as a separate chapter here as it contains much technnical information that is essential to the project. The text has been slightly upgraded since submission to the arXiv.

5.1 Device design and characterization

5.1.1 Waveguide design

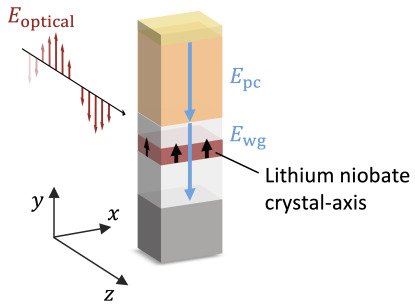

Figure 5.1: **Orientation of fields and materials with respect to the coordinate system.** Blue arrows: Applied electric fields in the waveguide (E_{wg}) and in the photoconductor (E_{pc}). Red arrows: Optical electric field E_{optical} of the transverse-magnetic mode. Black arrows: Lithium niobate crystal axis.

Choice of core material: We selected lithium niobate as the material for the waveguide core, as it has a large electro-optic coefficient, low optical loss, and thin-film wafers are commercially available. The change in the refractive index of lithium niobate under an external applied field is given by

$$\Delta n_i = -n_i^3 \sum_j r_{ij} E_j / 2, \tag{5.1}$$

where r_{ij} is the electro-optic tensor, n_i are principal semiaxes of the index ellipsoid, and E_i are the components of the electric field that are induced by the applied voltage. The strongest component of the electro-optic tensor of lithium niobate is the r_{33} component. To select the term in the above equation that includes r_{33}, we need to ensure that the electric field of the optical mode in the waveguide points in the same direction as the externally applied electric field. Since the planar electrode and conductive substrate of our device constrain the applied electric field to predominantly point in the (vertical) y-direction, we chose to use a transverse-magnetic (TM) mode such that the electric field of the optical mode also predominantly points in the y-direction. Finally, to capitalize on the r_{33} component, the crystal z-axis also needs to be oriented to point in the y-direction in this coordinate system, which is the reason we chose a Z-cut lithium niobate film.

Choice of cladding material: For the claddings of the waveguide, we chose silicon dioxide as it is readily available as buried oxide beneath lithium niobate, and simple to deposit via PECVD as top cladding. It also has desirable material properties for our device, including a high dielectric-breakdown field and low optical loss.

Choice of waveguide dimensions: We designed the thickness of each layer to ensure single-mode waveguiding in the y-direction, while maximizing the TM_0 mode overlap with the lithium niobate core. At the same time, to optimize the refractive-index modulation in the lithium niobate core, we minimized the overall thickness of the waveguide (see Sec. 5.1.2) while ensuring minimal propagation loss. These needs were balanced with the availability of thin-film lithium niobate wafers from NanoLN. We settled on a 700 nm thick film of Z-cut lithium niobate with 2 µm of silicon dioxide as the bottom cladding and 1 µm of silicon dioxide as the top cladding. This slab waveguide technically has two modes, but the higher order TM_1 mode has a significant substrate loss ($> 100\,\mathrm{dB/cm}$), and thus the waveguide is effectively single mode. The cladding thicknesses are chosen to minimize the loss of the TM_0 mode of the waveguide, which we characterize both numerically and experimentally to be less than $1\,\mathrm{dB/cm}$ (see Sec. 5.1.5 for more detail).

5.1.2 Photoconductor design

Choice of photoconductor material: We chose silicon-rich silicon nitride (SRN) as the photoconductor material due to its high dielectric-breakdown field and strong photoconductive response. In addition, the material properties of SRN are easy to tune by varying its silicon-nitrogen ratio [130].

Equivalent circuit model: Here, we explain the electrical modeling of the device. We consider a simplified model, where each small region of the 2D-programmable waveguide is modeled as multiple layers of homogeneous materials with different electrical resistivity ρ for each layer. This simplified model does not address the fringing fields present when a complex pattern of illumination is applied to the photoconductor (for this, a more complex model is necessary, and is discussed in Sec. 5.1.4), but it is useful for understanding the basic principles of the device. We model each layer as a leaky capacitor, i.e. each layer as a capacitor in parallel with a resistor, and multiple layers are connected in series with each other. The electric field $E_y(y)$ is then a piecewise constant function where the electric field in each material layer is constant and can be found from the corresponding lumped element impedance (Z_i) via $E_{y,i} = V_i/d_i$, where $V_i = V_{\text{applied}} Z_i/Z_{\text{total}}$ and d_i is the layer thickness for layer i. The gold electrode and strongly-doped silicon substrate are sufficiently conductive that for the purpose of this analysis they can be modeled as perfect conductors and are ignored in the lumped element model. On the other hand, the lithium niobate core (with $\rho \sim$ GΩcm) and both silicon dioxide claddings are sufficiently electrically insulating that we can model the resistors as open circuits. Therefore, we simplify the circuit for the waveguide, consisting of top cladding, lithium niobate core, and bottom cladding, as a single capacitor with capacitance C_{wg}. Hence, a simplified equivalent circuit to model the device is a single capacitor for the lithium niobate waveguide in series with a leaky capacitor for the photoconductor, where the resistance associated with the leaky capacitor can be reduced via external illumination.

Direct current (DC) vs alternating current (AC): Widely used waveguide claddings

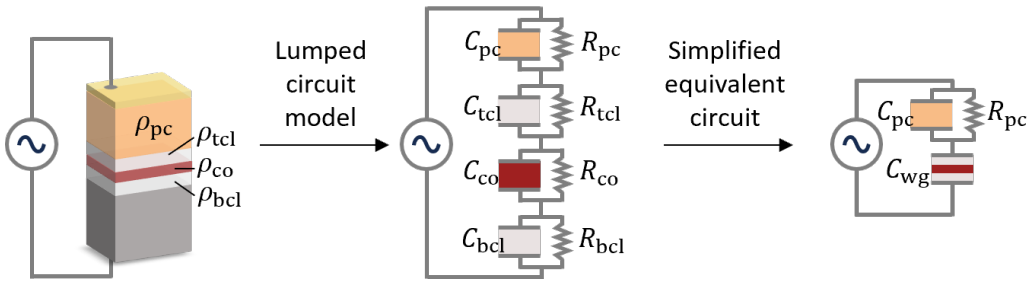

Figure 5.2: **Different abstractions of the electrical model of the 2D-programmable waveguide.** Left: The device is modeled as a stack of homogeneous layers with different electrical resistivities. Center: A lumped element abstraction models each layer as a leaky capacitor with different resistance and capacitance. Right: A simplified model of the material stack consisting of only one capacitor in series with a leaky capacitor. Abbreviations used in subscripts: **pc**: photoconductor, **tcl**: top cladding, **co**: lithium niobate core, **bcl**: bottom cladding, **wg**: waveguide

.

are extremely good electrical insulators. For example, thermally grown silicon dioxide's resistivity at room temperature is reported to be more than 10^{14}. This resistivity is far higher than the resistivity of silicon-rich silicon nitride and therefore, at DC voltages, the impedance of the device is dominated by the resistance of silicon dioxide. This implies that one cannot change the refractive index of the waveguide with a constant applied voltage, as the steady-state electric field would be zero everywhere except across the silicon dioxide layers and the electro-optic effect in lithium niobate would be negligible. We evade this problem by applying AC voltages to our device with frequencies that roughly satisfy $R_{pc} \sim 1/\omega C_{pc}$. As will be explained in the following section, this will allow us to effectively switch the electric field in the lithium niobate core on and off.

DC operation becomes feasible when waveguide claddings are made of materials with lower resistivity, such as doped silicon oxynitride [131, 132]. Such structures are also expected to yield a higher refractive-index modulation, due to the reduced impedance of the waveguide. Additionally, they would require a lower operational voltage to achieve the same refractive-index modulation (Δn). Another approach towards DC operation is to develop photoconductors with higher dark resistivity than the waveguide core and cladding.

Choice of photoconductor thickness: In order to design the material stack to effectively

switch the electric field inside the lithium niobate core on and off, we optimize the parameters of the circuit and the frequency of the applied voltage. The goal of the design was to find a photoconductor for which the electric field inside the lithium niobate core is as large as possible when the photoconductor is illuminated, and as low as possible when the photoconductor is not illuminated. The photoconductor's resistance R_{pc} varies depending on the illumination. In the ideal "bright" state (that is, with illumination on the photoconductor), the photoconductor's resistance is so low that all voltage drops across the waveguide and the electro-optic effect is strongest. In the ideal "dark" state (that is, with no illumination on the photoconductor), the photoconductor's resistance is so high that the voltage drop across the waveguide is as low as possible and the electro-optic effect is weakest. As mentioned above, it is very challenging to design a photoconductor whose dark resistance is higher than that of silicon dioxide. Therefore, we chose to use AC voltage instead and analyze the complex impedances of the circuit. The voltage drop across the waveguide is given by:

$$V_{\mathrm{wg}} = V_{\mathrm{applied}} \frac{Z_{\mathrm{wg}}}{Z_{\mathrm{wg}} + Z_{\mathrm{pc}}} = \frac{V_{\mathrm{applied}}}{1 + \frac{-i\omega C_{\mathrm{wg}}}{R_{\mathrm{pc}}^{-1} - i\omega C_{\mathrm{pc}}}}. \tag{5.2}$$

Assuming fixed capacitances, this expression is maximal in the limit $R_{\mathrm{pc}} \to R^{\mathrm{bright}} \ll 1/\omega C_{\mathrm{pc}}$ ("bright" state of photoconductor) and evaluates to $V_{\mathrm{wg}} = V_{\mathrm{applied}}$, i.e. all voltage drops across the waveguide. The expression is minimal in the limit $R_{\mathrm{pc}} \to R^{\mathrm{dark}} \gg 1/\omega C_{\mathrm{pc}}$ ("dark" state of photoconductor), and evaluates to $V_{\mathrm{wg}} = V_{\mathrm{applied}}/(1 + C_{\mathrm{wg}}/C_{\mathrm{pc}})$. First, this implies that, to maximize the difference of V_{wg} in the bright and dark state, the device needs to be operated with an AC applied voltage whose frequency ω satisfies $R^{\mathrm{bright}} \ll 1/\omega C_{\mathrm{pc}} \ll R^{\mathrm{dark}}$. Second, the difference between V_{wg} in the bright and dark state will be larger when the ratio $C_{\mathrm{wg}}/C_{\mathrm{pc}} = \epsilon_{\mathrm{wg}} d_{\mathrm{pc}}/\epsilon_{\mathrm{pc}} d_{\mathrm{wg}}$ is larger. Since the relative permittivities are determined by the choice of materials and the thickness of the waveguide d_{wg} is determined by the considerations presented in Sec. 5.1.1, the only freely tunable parameter is the thickness of the photoconductor d_{pc}. Therefore, a thicker photoconductor layer minimizes the electric field inside the lithium niobate core in the dark state and maximizes the programmable

refractive-index modulation.

Fig. 5.3 shows the maximal programmable refractive-index modulation Δn as a function of the photoconductor thickness, otherwise assuming the waveguide parameters shown in Extended Data Fig. 1 and a maximal field of 50 V/µm in the lithium niobate core. The programmable refractive-index modulation is defined as the difference between the refractive-index modulation in the bright and dark photoconductor state:

$$\Delta n_{\text{programmable}} = \Delta n_{\text{bright}} - \Delta n_{\text{dark}}, \tag{5.3}$$

where the refractive-index modulation in the dark and bright photoconductor state is calculated using Eq. 5.1 with the electric field determined from Eq. 5.2. For brevity we refer to the programmable refractive-index modulation $\Delta n_{\text{programmable}}$ simply as Δn throughout the manuscript. In Fig. 5.3, the blue line shows the maximal programmable refractive-index modulation assuming an optimal photoconductor with $R^{\text{bright}} \ll 1/\omega C_{\text{pc}}$ and $R^{\text{dark}} \gg 1/\omega C_{\text{pc}}$. We selected as thick of photoconductor layer thickness as reasonable ($d_{\text{pc}} = 4$ µm), balancing the benefit from a thicker layer with fabrication constraints (layer stress, deposition time, etc.) and the spatial resolution trade-off discussed in Sec. 5.1.4. The orange cross shows the programmable refractive-index modulation that we achieve in experiment, which is lower than the maximal value due to the photoconductor not being perfectly insulating in the dark state and not perfectly conductive in the bright state. More details about this measurement are presented below in Sec. 5.1.3.

5.1.3 Magnitude of refractive-index modulation

In this subsection, we present measurements of the programmable refractive-index modulation as a function of the applied voltage and illumination intensity on the photoconductor. We performed this measurement with an off-axis holography setup, by interfering the light that propagated through the waveguide with an off-axis reference beam (see Fig. 5.4). All

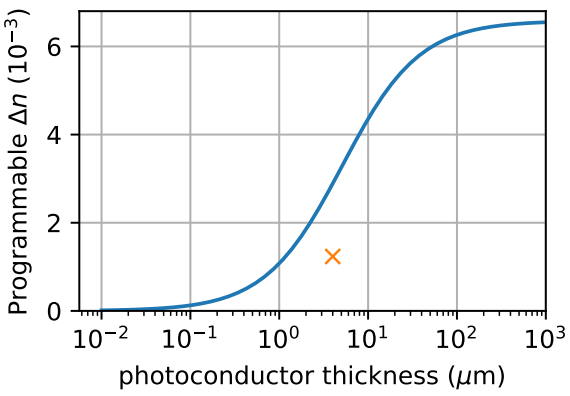

Figure 5.3: **Maximal programmable refractive-index modulation as a function of photoconductor thickness.** Blue: Maximal programmable Δn as a function of photoconductor thickness at an applied voltage of 1100 V and $f = 10$ Hz, assuming a photoconductor that switches from a perfectly "dark" state to a perfectly "bright state" as described in Sec. 5.1.2. Orange marker: Photoconductor thickness chosen and programmable Δn achieved in this work.

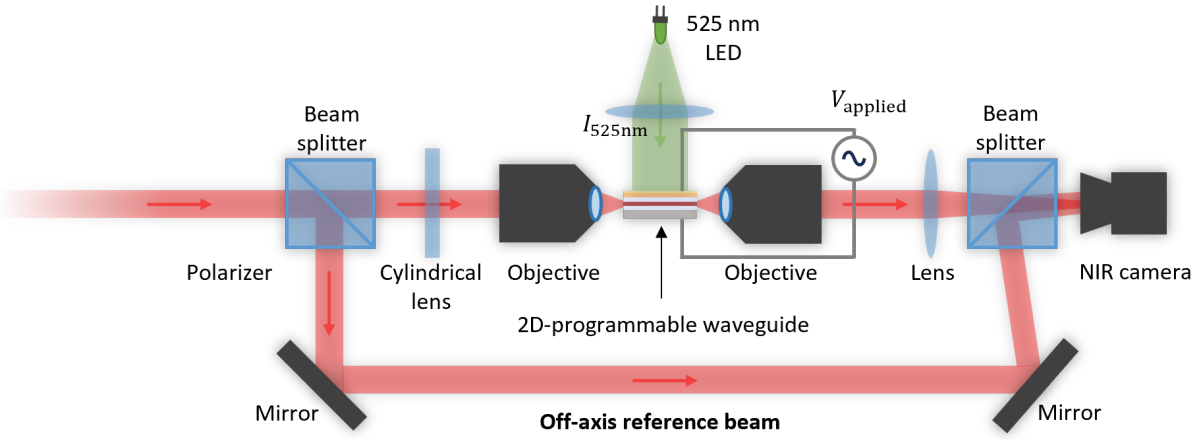

Figure 5.4: **Schematic of the off-axis holography setup.** We used this setup to measure the refractive-index modulation under different illumination conditions and applied voltages. We interfered light propagating through the 2D-programmable waveguide with light from the off-axis reference beam. We determined the refractive-index modulation at different applied voltages and illumination intensities from changes of the interference pattern.

measurements presented here are measurements of the effective index of the TM_0 mode, and are given in reference to the refractive index of this mode with no external voltage applied. We measured the refractive index at different applied voltage amplitudes (0 to 1100V) and different illumination strengths (0 - 57 mW/cm^2), at a fixed AC voltage frequency of 10 Hz. As shown in Fig. 5.5A, without any illumination on the photoconductor ("dark"), the refractive index linearly increases with the applied voltage to about $4 \cdot 10^{-3}$ at 1100 V. With the photoconductor being illuminated at about 50 mW/cm^2, the refractive-index modulation increases to about $5.2 \cdot 10^{-3}$ at the highest voltage of 1100V. We measured the difference between the dark- and bright-state refractive-index modulation, i.e. the programmable refractive-index modulation, to be around $1.2 \cdot 10^{-3}$ at the highest voltage. We also present a measurement of the refractive-index modulation as a function of the illumination intensity, interpolating between the "dark" and "bright" state.

Finally, we note that in Fig. 2 of the main manuscript, the largest achievable Δn is 1×10^{-3}, as opposed to $1.2 \cdot 10^{-3}$ presented here. This discrepancy arises because we conducted the experiments for Fig. 2 at 1000 V, as opposed to the max voltage applied here of 1100 V. Moreover, we note that the off-axis holography measurements were performed on a different device than the one used for the main manuscript's measurements. Variations in the film thickness of the electrode during deposition led to a higher transmission of the illumination through the electrode in the device used for off-axis holography measurements. Thus, more optical power was delivered to the photoconductor, leading to a higher refractive-index modulation.

5.1.4 Spatial resolution of refractive-index modulation

In this subsection, we discuss an extension to the 1D-model of the electric field inside the waveguide that was introduced in Sec. 5.1.1. The model presented here takes into account the 3D-distribution of electric fields inside the dielectric stack when parts of the photoconductor are illuminated to create a spatially varying refractive-index distribution. The considerations

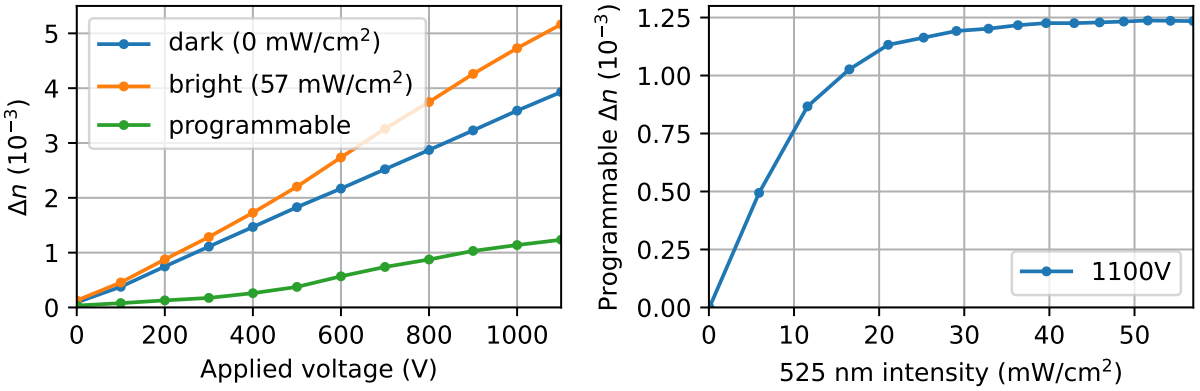

Figure 5.5: **Measured refractive-index modulation.** Left: Refractive-index modulation of the TM_0 mode as a function of applied voltage for the photoconductor in a bright and dark state, and their difference, which is the programmable refractive-index modulation. Right: Programmable refractive-index modulation of the TM_0 mode as a function of illumination intensity.

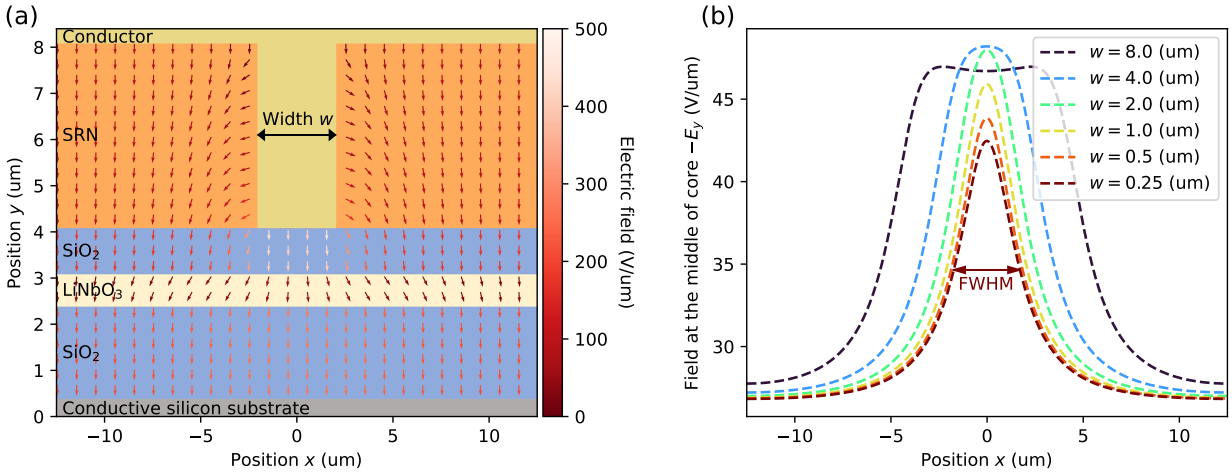

Figure 5.6: **Electric field distribution in the 2D-programmable waveguide with spatially varying projected illumination.** (a) A simplified model to capture the electric-field spreading effects in light-based programming of the refractive-index distribution. We assume dielectric permittivity of $\epsilon = 3.9$ for SiO_2 and $\epsilon = 7.2$ for SRN. Lithium niobate exhibits anisotropy and the permittivities are 85.2 and 27.8 for the x- and y-directions, respectively. The directions and the colors of the arrows represent the direction and strength of electric field, respectively, for $V_{cc} = 1000\,V$. (b) Distribution of the vertical electric field in the middle of the $LiNbO_3$ core layer for various conductive strip width w. For all the simulations, we assume periodic boundary condition in the x-direction.

in this paragraph show how the spread of the electric field affects the minimal features size achievable in the programmable refractive-index distribution.

Inside a dielectric medium with no free charge, the displacement field follows the macroscopic Maxwell equation

$$\nabla \cdot \mathbf{D}(\mathbf{r}) = 0, \tag{5.4}$$

where $\mathbf{D}(\mathbf{r})$ is the displacement field at position $\mathbf{r} = (x, y, z)^\intercal$. Using the constitutive relation $\mathbf{D}(\mathbf{r}) = \epsilon_0 \epsilon(\mathbf{r}) : \mathbf{E}(\mathbf{r})$, and $\mathbf{E}(\mathbf{r}) = -\nabla\phi(\mathbf{r})$ with the scalar potential ϕ, we obtain

$$\nabla \cdot (\epsilon(\mathbf{r}) : \nabla\phi(\mathbf{r})) = 0. \tag{5.5}$$

Here, ϵ_0 and $\epsilon(\mathbf{r})$ are the vacuum and relative permittivities of the medium, respectively. Solving (5.5) under appropriate boundary conditions, we can calculate how the electric field spreads inside dielectric materials.

As shown in Fig. 5.6(a), as a model for the programmable slab waveguide, we consider a stack of layers composed of conductive Si substrate, 2.0 µm of SiO_2 bottom cladding, 0.7 µm of $LiNbO_3$ core, and 1.0 µm of SiO_2 top cladding. The layers are stacked in the y-direction and are modeled to be infinitely extending in the x- and z-directions. On top of the top cladding is a 4.0 µm photoconductive SRN layer, with a T-shaped conductive region of width w in the x-direction. The substrate is grounded, and the top conductive region is connected to a voltage source with voltage V_{cc}. We use such a structure to model the situation in which a limited region on the photoconductive SRN is illuminated by a focused light with spot size w, making the material locally conductive. It is our goal to use this simplified model to unravel how the spot size w is related to the distribution of the electric field inside the core material, studying the impact of electric-field spreading on the spatial resolution we can achieve in light-based programming of the refractive-index distribution.

For the structures mentioned above, we solve (5.5) using the biconjugate gradient stabilized method. In Fig. 5.6(a), we show a typical electric field distribution inside the medium for $w = 4\,\mu\mathrm{m}$. To be more quantitative, we show in Fig. 5.6(b) the vertical-electric-field distribution E_y at the middle of the lithium niobate core for various conductive strip width w. For a large enough w, the width of the distribution of E_y is approximately proportional to w. On the other hand, as w approaches a value comparable to the thickness of the

dielectric stacks, the spreading of electric fields sets a finite lower bound to the width of the distribution of E_y. Using the results from $w = 0.25\,\mu\text{m}$, we find the full-width at half maximum (FWHM) of the field distribution to be $3.5\,\mu\text{m}$. This sets a limit to the minimum feature size we can achieve, even in the absence of any diffusion of the carriers inside SRN or finite resolution of the illumination profile impinging onto the SRN.

5.1.5 Propagation loss

In this section, we discuss the various sources of optical propagation loss in the 2D-programmable waveguide, along with our numerical and experimental characterization of the loss. There are two primary sources of loss in our device: the first is associated with the lithium niobate slab waveguide, and the second is radiative (substrate) loss into the photoconductor and the conductive silicon carrier wafer.

One source of optical loss in the lithium niobate slab waveguide is due to the ion-sliced thin-film lithium niobate wafer from NanoLN, as well as the silicon dioxide deposited via PECVD. According to Ref. [134], the optical loss in the lithium niobate slab waveguide is typically less than $1.5\,\text{dB/m}$ for samples that have not been thermally annealed, as is the case with our device. A key factor contributing to the low loss is the absence of lithographic etching in our process, thereby avoiding side-wall induced loss, which is often the predominant source of loss.

The second source of loss in the 2D-programmable waveguide is associated with the photoconductor and the conductive silicon carrier wafer. We measured the material loss of the photoconductor, silicon-rich silicon nitride, to be $5\,\text{dB/cm}$ using a Metricon prism coupler. Thus, the material has a refractive index of $2.03 + 1.4 \times 10^{-5} i$. The conductive carrier wafer, a p-type silicon wafer, has a refractive index of $3.48 + 0.027i$. Thus, the loss is primarily due to the optical power of the lithium niobate slab waveguide radiating into the photoconductor and the carrier wafer, rather than being absorbed by the materials. This phenomenon,

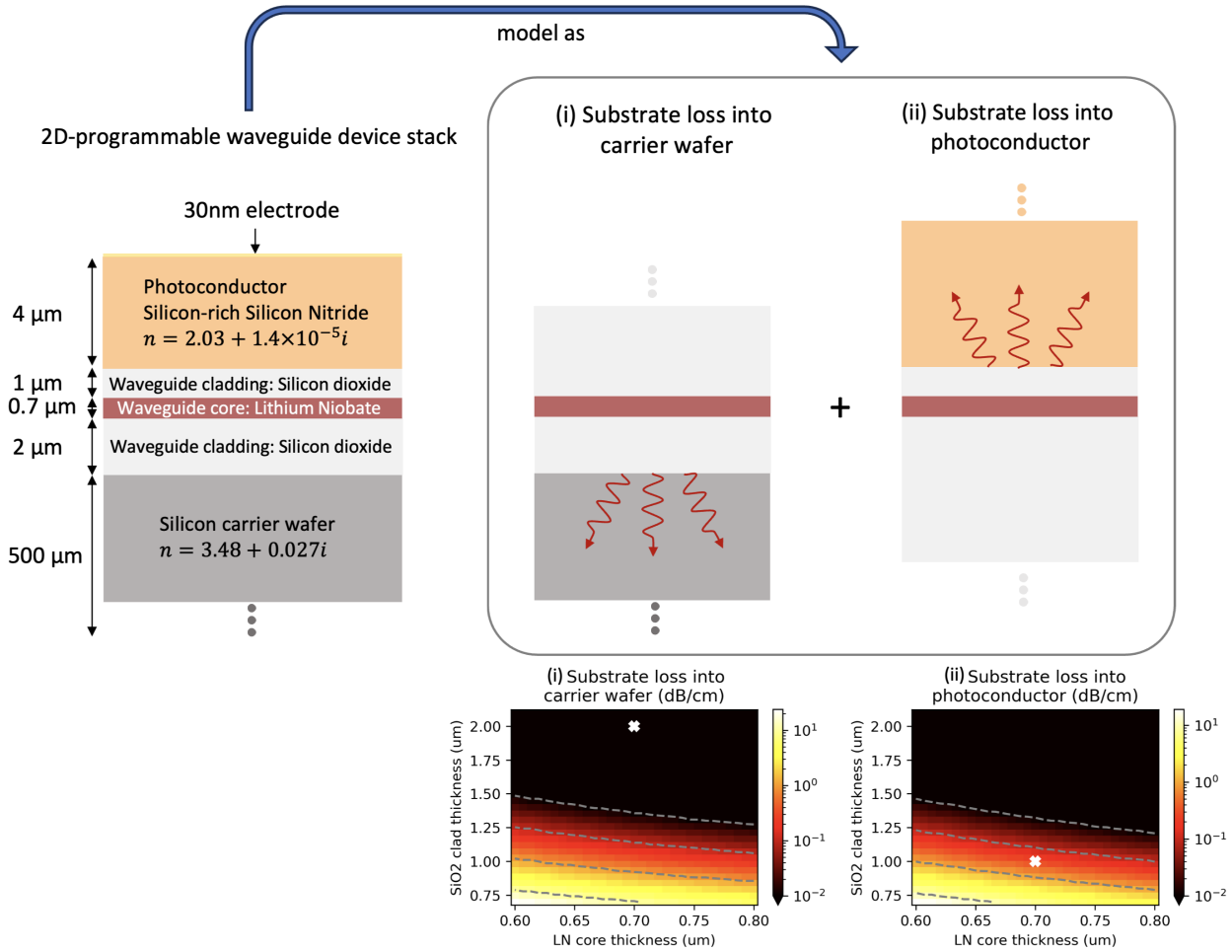

Figure 5.7: **Simulation of propagation loss in the 2D-programmable waveguide.** The dominant source of propagation loss is due to radiative substrate loss into both the silicon carrier wafer and the photoconductor. We model this loss as the sum of two independent substrate loss contributions from the carrier wafer and photoconductor. This loss is simulated for different waveguide core and cladding thicknesses with the transfer matrix method [133]. The device parameters in this work are represented by the white "×" marker in the two subfigures.

traditionally referred to as substrate loss, typically pertains only to the carrier wafer. However, given the photoconductor's substantial thickness (approximately 4 µm), it can also be approximately modeled as radiative substrate loss. Therefore, we model this loss in the device as the sum of two independent substrate loss contributions from the silicon carrier wafer and photoconductor (see Fig. 5.7). The substrate loss is numerically computed using the transfer matrix method [133] for different core and cladding thicknesses. For the carrier wafer, the loss is negligible, below 1 dB/m, owing to a thick bottom cladding of 2 µm. For

the photoconductor, the loss is estimated at 0.3 dB/cm, attributed to a thinner top cladding of 1 µm. As loss varies sharply with device parameters, we conservatively estimate that the overall propagation loss in the device loss is below 1 dB/cm.

Due to the absence of lithographic etching in our device, precise experimental measurement of this numerically estimated low loss is challenging. We fabricated chips of varying lengths (5mm, 1cm, 2cm) and qualitatively observed that transmission through these chips does not vary significantly with length, consistent with the simulations.

Finally, we emphasize that for applications like quantum photonics, the loss can be reduced to be less than 1 dB/m by further increasing the top cladding thickness. We also note that it is possible to further enhance device performance, specifically the refractive-index modulation (Δn), without a substantial increase in optical loss, by reducing the thickness of the bottom cladding.

5.2 Experimental setup

In this section, we further detail the experimental setup described in the Methods, focusing on the beamshaper and the projector. In the descriptions that follow, we reference Extended Data Fig. 2 and Fig. 5.8 for clarity. To maintain consistency, naming conventions for components within the experimental setup (e.g., "Lens 3") are aligned with those used in the figures.

1D spatial beamshaper to create optical inputs: The core working principle of the beamshaper is to create spatially varying phase-gratings on a 2D-phase spatial light modulator [127, 128]. More specifically, at every position on the lateral dimension (x direction), a vertical grating with variable amplitude is displayed on the SLM. These gratings are also shifted by a variable amount in the vertical y dimension. By controlling the amplitude and this shift, we controlled the 1D amplitude $A(x)$ and phase $\phi(x)$ distribution of the first-

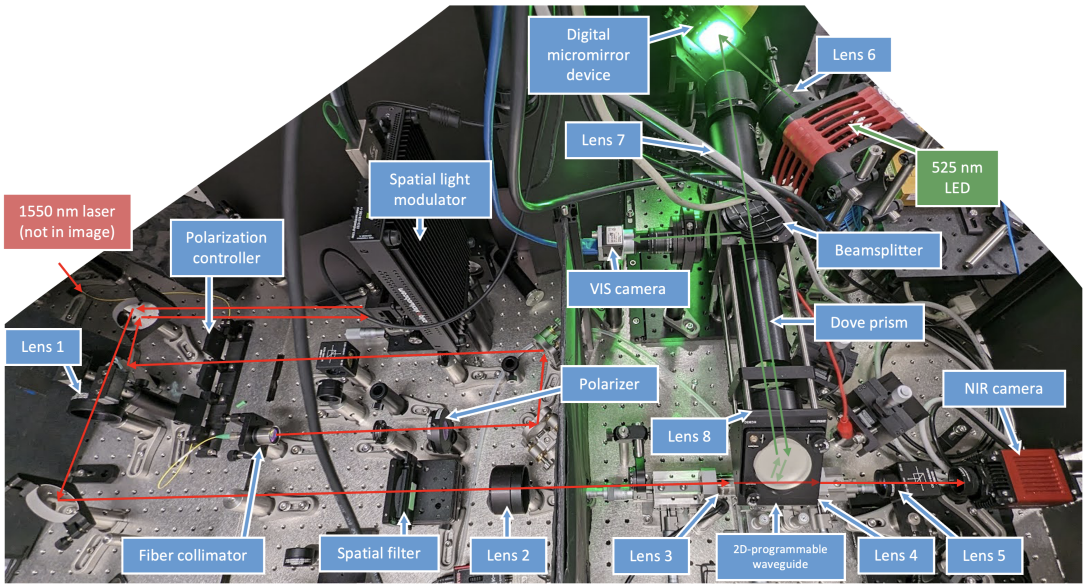

Figure 5.8: **Photograph of the experimental setup.** The photograph is overlaid with the optical beam path (red) and the photoconductor illumination path (orange). The laser beam propagating through the 2D waveguide originates on the left and propagates to the camera in the bottom right. The 525nm control light originates in the top right and is projected onto the waveguide in the bottom right.

order diffracted light. We shone a collimated beam of vertically polarized, continuous-wave 1550 nm light from a JDS VIAVI MAP MTLG-B1C10 Tunable DBR Laser onto a Meadowlark Optics UHSP1K-850-1650-PC8 spatial light modulator (SLM). More specifically, the beam from the laser was collimated to a 7 mm waist before it hits the two-dimensional phase-SLM, on which spatially varying gratings with a period of 16 pixels are displayed with a pixel pitch of 17 µm. Next, the light propagated through a lens with focal length 500 mm (Lens 1, Thorlabs A1380-C-ML), which separated the diffraction maximums of the phase-gratings displayed on the SLM by approximately 3 mm. A spatial filter (a slit) only let the first-order diffraction maximum pass through to another lens (Lens 2, Thorlabs TTL200-A, $f = 200\,\text{mm}$). From there on, the light propagated to an objective (Lens 3, Olympus UPLFLN 40X Objective, NA = 0.75, $f = 4.5\,\text{mm}$) that coupled the light into the 2D-programmable waveguide. The second and third lens are essentially a 4-f relay system to demagnify the beam at the focal plane of the first lens. Thus, the distribution of light at the input facet of the 2D-programmable waveguide is approximately given by the spatial Fourier

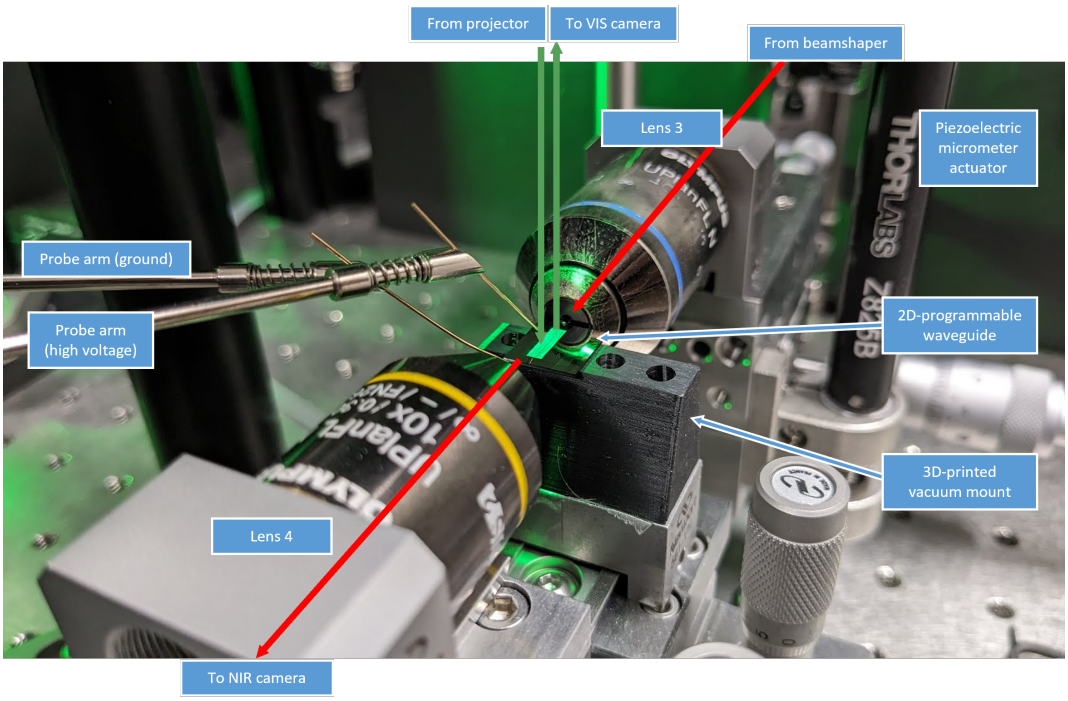

Figure 5.9: **Photograph of the butt-coupling setup.** The photograph is overlaid with the optical beam path (red) going from the top right to bottom left and the photoconductor illumination path (orange) coming from the top.

transform of the amplitude- and phase-profile corresponding to the gratings displayed on the phase spatial light modulator. We can create arbitrary input electric fields with features as small as $w_0 = 1.6\,\mu\text{m}$ on the input facet, over a distance of $600\,\mu\text{m}$.

Projector to create a programmable illumination pattern: To create programmable illumination patterns, we used a digital micromirror device (DMD, Vialux V-7000) with a resolution of 1024x768 pixels and a pixel pitch of $13.7\,\mu\text{m}$. We illuminated the DMD with green light ($525\,\text{nm}$) from a SOLIS-525C high-power LED, collimated by a condenser lens setup (Lens 6). We imaged the surface of the DMD onto the surface of the 2D-programmable waveguide via a 4-f setup consisting of two tube lenses (Lens 7, Thorlabs TL300-A, and Lens 8, Thorlabs TTL200-A). To account for the 45-degree rotation of the micromirror array of our DMD, we inserted a Dove prism (Thorlabs PS993M) between the tube lenses, which, appropriately positioned, rotates the image by 45 degrees. Using the same optical path as for the projection, we imaged the surface of the 2D-programmable waveguide onto a camera by

inserting an 8:92 (R:T) Pellicle beamsplitter (Thorlabs BP245B1) between the tube lenses and imaged the waveguide surface via an additional tube lens (Lens 9, TTL100-A) onto a Basler ace acA5472-17um camera. The projection illumination pattern on the surface of the 2D-programmable waveguide has dimensions of 9.1 mm × 6.8 mm, and each individual pixel of the projected pattern is 9 µm × 9 µm in size.

5.3 Training the 2D-programmable waveguide to perform machine learning

In this section, we explain how the parameters of the 2D-programmable waveguide is trained to perform machine learning. This includes describing the different training algorithms that we considered, and explaining our decision to use physics-aware training for this work. We begin the section with a discussion on how many parameters are present in our device, as it is a dominant factor that determined the choice of training algorithm.

5.3.1 Parameter count of 2D-programmable waveguide

The parameters of the 2D-programmable waveguide is the refractive-index modulation $\Delta n(x, y)$ of the slab waveguide. While it is conceptually advantageous to reason about the refractive-index modulation as a continuous function over space, it is in practice a discretized quantity as we use a DMD to project different patterns of illumination on the device, which has a discrete number of pixels. Furthermore, as a single pixel does not exert much phase-shift on the propagation of light, we group pixels on the DMD into macro-pixels for practical implementation. In other words, the number of parameters that we train for the device depends on how finely we perform this discretization, which we outline below.

In the x dimension, which is the direction that is perpendicular to the direction of propaga-

tion, we chose to use the finest discretization, which is limited by the imaging setup. This is appropriate as finer features in x directly lead to more control over the wave dynamics. Thus, the number of independent parameters in the x dimension is given by dividing the width of the total area that the wave propagates through (1 mm) by the smallest pixel size of the projected pattern on the chip (9 µm).

For the z direction, which aligns with the direction of propagation, the situation differs. To understand this more formally, consider the distance over which a maximal refractive-index modulation can induce a phase shift of π. This quantity $L_\pi = \lambda/(2\Delta n_{\max})$ is approximately 1 mm in our device. Since the length of a single pixel (9 µm) is much smaller than L_π, each pixel only exerts a negligible amount of phase-shift on the light that is propagating in the z direction. Therefore we do not consider each individual pixel to be a freely tunable parameter, but instead only count groups (macropixels) of 11 individual pixels to be one tunable parameter. We chose this number so that the length of macropixels is $L_\pi/10$, to ensure that each programmed macropixel can induce a non-negligible phase shift of $\pi/10$. Thus, the number of independent parameters in the z dimension is given by dividing the length of the chip by the length of the macropixel in z.

Consequently, the total number of parameters for the 2D-programmable waveguide is given by the product of the number of parameters in each direction, which is approximately 1mm/9µm in the x direction and 9 mm/100 µm in the z direction, yielding 10,000 parameters.

5.3.2 Choice of training algorithm

In this subsection, we describe the algorithms we considered for training the 2D-programmable waveguide. There are two critical factors that determined our choice to use physics-aware training: the number of parameters (10,000) for our device, and update speed of our experiment (20 Hz for inputs and 3 Hz for parameters).

Model-free training algorithms: These algorithms do not require a digital model to train

the physical system. One approach involves individually perturbing each parameter and computing the gradient of the loss function with respect to each parameter [88]. Alternatively, random gradient descent can be used, where the parameters are perturbed in a random direction [77]. Generally, both approaches slow down the rate of training, as a large number of passes through the setup is required to accurately sample the gradient. Given the large number of parameters and the relatively slow update speed of our experiment, this method would have resulted in impractically long training times for the MNIST machine learning task. We note that these algorithms could be effective for training the 2D-programmable waveguide if the number of parameters is reduced, for instance, by a low-dimensional parametrization of the projected patterns.

In-silico training: An alternative approach is to use a purely model-based method, termed in-silico training. In this approach, training is performed entirely on a digital computer with the digital model. For differentiable digital models, the physical system can be trained with the backpropagation algorithm, for faster training. After completing the training, the parameters are transferred from the digital computer to the physical system. This requires an excellent digital model of the system. We attempted this approach and found that despite having good digital models (see Sec. 5.4), poor performance was attained on the experiment. For instance, the test accuracy on MNIST handwritten-digit classification is only 60% with in-silico training.

Physics-aware training: In this work, we use physics-aware training, a hybrid in-silico in-situ training algorithm [116]. This method combines the advantages of both model-free and model-based training. Our reasons for choosing this algorithm are as follows. Firstly, it is a backpropagation algorithm. Thus, approximate gradients are computed in a single backward pass, resulting in faster training. Additionally, since the physical system is used to compute the forward pass, the training is able to mitigate the mismatch between the experiment and digital model as well as the effects of experimental noise. A requirement for this approach is a differentiable digital model of the physical system, which we detail in Sec. 5.4. We will explain physics-aware training in more detail in the following subsection.

In-situ backpropagation algorithms: These algorithms use the physical system to obtain the gradient of the loss function [43, 53, 135]. Thus, a digital model of the system is not required, and the training is fast as the algorithm is gradient-based. However, the algorithm requires having bidirectional input of light [53, 135] and measuring the intensity distribution of light within the chip [127, 136]. For this reason, the method was not implemented in our current experiment. We note that as we scale up the device to a larger number of parameters, this method may become essential for training the 2D-programmable waveguide.

5.3.3 Physics-Aware Training

This section will explain the physics-aware training algorithm, which is a hybrid in-silico in-situ training algorithm. It is a summary of the algorithm that is specialized for the 2D-programmable waveguide—a more detailed and general explanation of the algorithm can be found in the original paper [116]. The schematic of the algorithm is shown in Fig. 5.10, which outlines the four key steps of the algorithm. The figure also explains the algorithm in the specific context of vowel classification, where the input vector is 12 dimensional and output vector is 7 dimensional. Finally, we note for brevity that the following equations will assume a batch size of one.

1. Forward-pass through physical system: The first step is to perform a forward pass through the physical system, which in our case is the 2D-programmable waveguide. Mathematically, this is given by

$$\mathbf{y} = f_{\mathrm{p}}(\mathbf{x}, \boldsymbol{\theta}). \tag{5.6}$$

Note that the physical transformation f_{p} includes both the amplitude encoding of the input machine learning data into an input optical field, and the binning operation that converts the output intensity to the machine learning output. Thus, this is a function that takes in an input vector $\mathbf{x} \in \mathbb{R}^{12}, \boldsymbol{\theta} \in \mathbb{R}^{\sim 10{,}000}$ and returns an output $\mathbf{y} \in \mathbb{R}^{7}$.

2. Compute the error vector: Using the output of the physical system $\mathbf{y}$ and the target

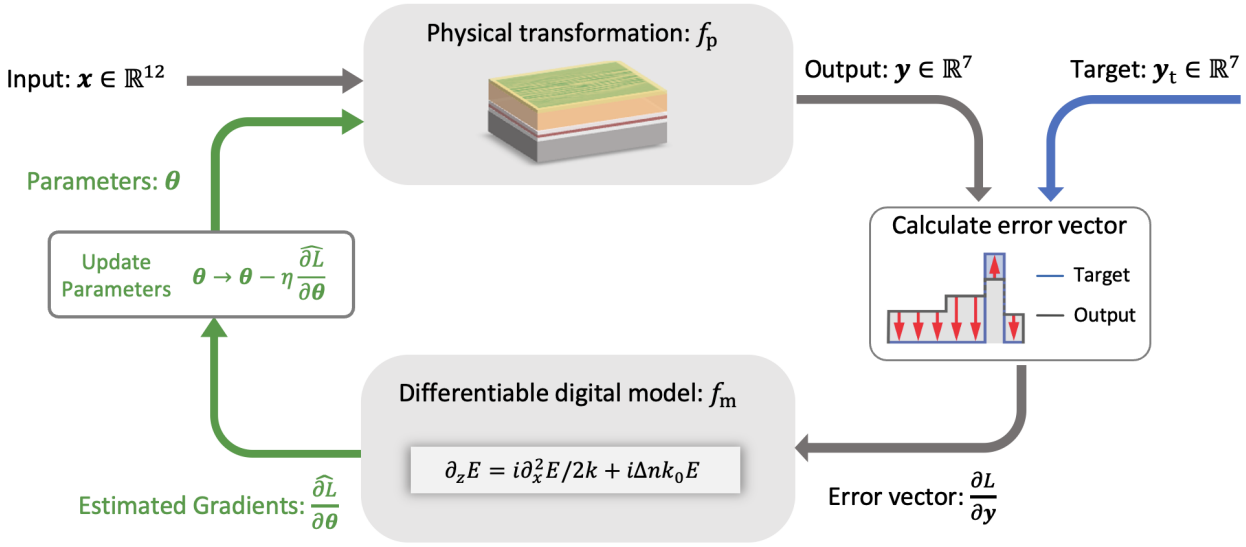

Figure 5.10: **Schematic of physics-aware training.** The physical system is used in the forward pass, and the digital model is used in the backward pass, to obtain an approximate gradient of the loss function with respect to the physical parameters.

output $\mathbf{y}_t$, the second step is to compute the error vector with a digital computer. The error vector indicates how the outputs of the physical system should be adjusted to minimize the loss function, and plays a key role in a backpropagation algorithm. Mathematically, this is given by

$$\frac{\partial L}{\partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{y}} H(\mathbf{y}_t, \text{softmax}(\mathbf{y})), \tag{5.7}$$

where the softmax function is used to turn the outputs from the physical system into a probability, and H is the cross-entropy function that is usually used as a loss function for classification tasks. One key intuition for why physics-aware training works well is that the error vector is accurately computed, as the algorithm has access to the real output of the physical system.

3. Compute the estimated gradients: The third step is to compute the estimated gradients, which is the approximate gradient of the loss function with respect to the physical parameters. This is done by performing a backward pass through the differential digital model of the physical system (f_m). Mathematically, this is given by

$$\frac{\widehat{\partial L}}{\partial \boldsymbol{\theta}} = \left(\frac{\partial f_m(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \frac{\partial L}{\partial \mathbf{y}}, \tag{5.8}$$

where $\frac{\partial f_m(\mathbf{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the Jacobian matrix of the digital model of the physical system. The transposed Jacobian matrix performs a matrix-vector multiplication on the error vector to arrive at the estimated gradients. It should be noted that the estimated gradients $\widehat{\frac{\partial L}{\partial \boldsymbol{\theta}}}$ computed here are approximate rather than exact, as the digital model is only an approximation of the physical system. A key reason for the effectiveness of physics-aware training is that as long as the estimated gradient predicts the direction of the true gradient within a 90-degree cone ($\angle\left(\widehat{\frac{\partial L}{\partial \boldsymbol{\theta}}}, \frac{\partial L}{\partial \boldsymbol{\theta}}\right) < 90°$), a small parameter update will result in a reduction of the loss.

4. Update the parameters: With the estimated gradients, the final step is to update the parameters of the physical system. This is done by performing a gradient descent step on the parameters. Mathematically, this is given by

$$\boldsymbol{\theta} \rightarrow \boldsymbol{\theta} - \eta \widehat{\frac{\partial L}{\partial \boldsymbol{\theta}}}, \tag{5.9}$$

where η is the learning rate. After this, step 1 is repeated, and the algorithm continues to iterate until the loss converges.

Though the algorithm has been specified mathematically in (5.6)-(5.9), in practice the algorithm is implemented as a custom autograd function in PyTorch. This custom function can be generated via the package `Physics-Aware-Training` available at `https://github.com/mcmahon-lab/Physics-Aware-Training`. With the package, the user can run the following line of code to define a custom "physics-aware function":
`f_pat = make_pat_func(f_physical, f_model)`. The rest of the code can be written in regular PyTorch to train the system. Thus, the user can access regular PyTorch functionality such as optimizers and schedulers for training.

5.4 Digital model of 2D-programmable waveguide

In this section, we describe the digital model used for the 2D-programmable waveguide. The model is largely physics-based, and is modeled by a partial differential equation. As

the calibration and alignment of the chip was complex, we also describe the procedure in this section. Finally, we found that the purely physics-based model only predicts the experimental outputs qualitatively. Thus, to improve the agreement further, we augmented the physics-based model with additional parameters, which were trained on experimental data. We outline this data-driven approach to fine-tune the physics-based model in the final subsection.

5.4.1 Physics-based model

Here, we derive the theoretical model that we use to emulate the 2D-programmable waveguide on a digital computer. To summarize the approach, we begin from the time-independent wave equation (Helmholtz equation) in 2D, and apply the standard beam-propagation method to arrive at the final equation.

The time-independent wave equation in 2D is given by,

$$\frac{\partial^2 \tilde{E}}{\partial z^2} + \frac{\partial^2 \tilde{E}}{\partial x^2} + n^2(x, y) k_0^2 \tilde{E} = 0, \tag{5.10}$$

where $\tilde{E}$ refers to the electric field in the y dimension as we work with the transverse-magnetic (TM) mode. $n(x, y)$ is the effective refractive index of the slab. We note that the 2D wave equation works well for our setting, as the change in refractive index is generally small [81]. We use the ansatz $\tilde{E}(x, z) = E(x, z) \exp(\mathrm{i}kz)$ and apply the slowly-varying amplitude approximation, to arrive at,

$$\frac{\partial E}{\partial z} = \frac{\mathrm{i}}{2k} \frac{\partial^2 E}{\partial x^2} + \mathrm{i}\Delta n(x, z) k_0 E, \tag{5.11}$$

where $\Delta n(x, z) = n(x, z) - n_0$ is the change in refractive index, which is assumed to be small $\Delta n \ll n_0$. k_0 is the wavevector in vacuum, $k = n_0 k_0$ is the wavevector in the medium. We note that this equation assumes that the wave is only traveling in the forward direction. This is a good approximation as we did not explicitly train for any resonant structures that could induce significant back reflections. To numerically solve this partial differential equation, we

use the split-step Fourier method, which is a numerical method that is commonly used to solve PDEs of this form [137].

Input-output relation: The digital model of the 2D-programmable waveguide is as follows. Recall that the input-output relation is $\mathbf{y} = f_{\mathrm{m}}(\mathbf{x}, \boldsymbol{\theta})$, where $\mathbf{x}$ and $\mathbf{y}$ have dimensionalities associated with the machine learning input data and output data. Thus, the initial condition of the field propagation is given by,

$$E(x, z = 0) = \sum_i x_i \times \underbrace{e^{-(x-\mu_i)^2/w_0^2}}_{E_{\mathrm{mode},i}(x)}, \tag{5.12}$$

where $E_{\mathrm{mode},i}(x)$ are the input modes, which we pick to be Gaussian modes with width w_0, and whose mean position, μ_i, are translated linearly with respect to the index i. By numerically solving (5.11), we can obtain the output field at the output facet of the waveguide $E(x, z = L_z)$. The intensity at the output facet is then binned to arrive at the the machine learning output data $\mathbf{y}$, which is given mathematically by,

$$y_i = \int_{x_{\mathrm{bin},i}-w/2}^{x_{\mathrm{bin},i}+w/2} |E(x, z = L_z)|^2 \mathrm{d}x, \tag{5.13}$$

where $x_{\mathrm{bin},i}$ is the location of the i-th bin, and w is the width of the bin.

Conversion between projected image and refractive-index modulation: As shown in Fig. 5.5, the relation between the intensity of the projected light on the chip $I(x, z)$ and the refractive-index modulation $\Delta n(x, z)$ is nonlinear as saturation effects set in when the intensity is sufficiently high. However, for machine learning, we have found that operating in the linear regime is beneficial to obtain a better digital model. In addition, the spatial resolution of the refractive-index distribution is limited due to the spreading of electric field in the stack, as discussed in Sec. 5.1.4. Taking both of these facts into consideration, the model we use to convert from the parameters, which is the projected image ($\boldsymbol{\theta} = I(x, z)$) to the refractive-index modulation $\Delta n(x, z)$ is given by,

$$\Delta n(x, z) = \frac{\Delta n_{\max}}{I_{\max}} \left(g(x, z) * I(x, z) \right), \tag{5.14}$$

where $*$ is the 2D convolution operation, g is a 2D Gaussian kernel with standard deviation

d_{kernel}, $\Delta n_{\max}$ is the maximum refractive-index modulation, and $I_{\max}$ is the maximum light intensity of the projected pattern that is illuminated on the chip.

5.4.2 Initial calibration of physics-based model

Calibration of beamshaper: In this work, we built a beamshaper that enables us to send arbitrarily programmable input field distributions into the 2D-programmable waveguide. As detailed in Sec. 5.2, the beamshaper is able to set both the amplitude as well as the phase of the input field. Thus, we leveraged this capability to calibrate and account for misalignments in the beamshaper. It should be noted that in the future, when on-chip modulators are used for faster input modulation, this calibration will no longer be necessary.

Mathematically, the miscalibrations of the beamshaper can be characterized as follows:

$$E_{\text{in}}(x) = A_{\text{mis}}(x) \exp\left(\mathrm{i}k_{x,\text{mis}}x + \mathrm{i}C_{\text{mis}}x^2/2\right) E_{\text{in, set}}(x), \tag{5.15}$$

where $E_{\text{in, set}}(x)$ represents the input field that is set to the beamshaper. $A_{\text{mis}}(x)$, $k_{x,\text{mis}}$ and C_{mis} all represents different kinds of misalignments that will be detailed later. Once these misalignments are characterized, (5.15) can be inverted to accurately send in a desired input field. We use the following procedure to calibrate the beamshaper:

- The coupling efficiency $A_{\text{mis}}(x)$ can be characterized by sending in focused Gaussian beam with waist $w_0 \sim 5\,\mu\text{m}$ and different mean input location x to the input facet of the waveguide, and measuring the overall power at the output facet. We found that the transparency window of coupling efficiency (i.e. $A_{\text{mis}}(x)$ drops when $|x|$ is larger than some value) varies as the slab waveguide is rotated relative to the 1D-axis of the input beam. This misalignment can be corrected by rotating the SLM used in the beamshaper. Once corrected, $A_{\text{mis}}(x)$ is mostly flat across the entire field-of-view of the beamshaper, which is about $600\,\mu\text{m}$ wide for our experiment.

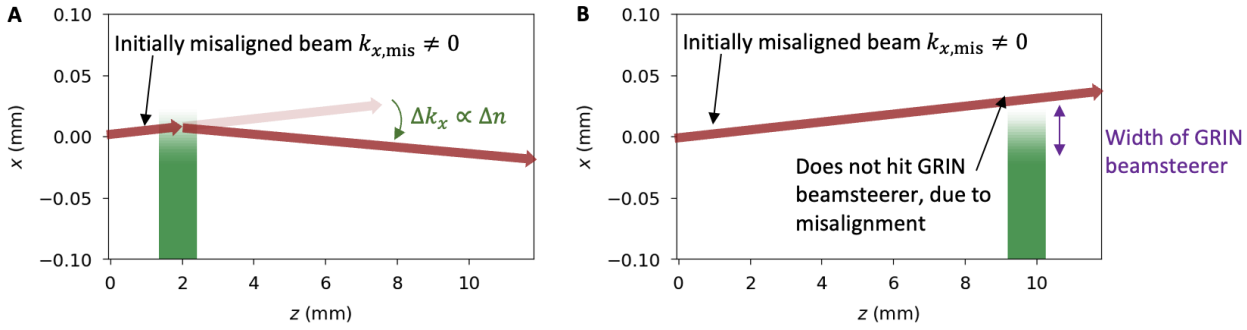

Figure 5.11: **Calibration of the programmable 2D-waveguide with a graded index (GRIN) beamsteerer.** As shown schematically in the figure, it can be used to calibrate for misalignments in the beamshaper, as well as to measure system parameters, including the maximal refractive-index modulation Δn_{max} and the spatial resolution of the chip d_{kernel}.

- As the beamshaper involves multiple lenses, there are many distances between lenses that could be misaligned. In practice, we observed that the most relevant misalignment is the distance between lens 2 and 3 not being equal to f_1+f_2, as this distance changes whenever a new chip is installed into the setup. When this occurs, the input field picks up a quadratic phase front ($C_{\mathrm{mis}} \neq 0$), which can be understood as a virtual lens being inserted at the input facet. Thus, this lens imparts a wavevector kick that is a linear function of the input location $\Delta k_x = C_{\mathrm{mis}} x_{\mathrm{in}}$. To calibrate for this, we send in collimated Gaussian beams ($w_0 \sim 50\mu$m) with different mean input location x_{in} and measure how much the mean output location x_{out} deviated from the input location. With the formula $x_{\mathrm{out}} = x_{\mathrm{in}} + C_{\mathrm{mis}} x_{\mathrm{in}} L_z \lambda_0/(2\pi n_0)$, C_{mis} can be measured and corrected for.
- To calibrate for $k_{x,\mathrm{mis}}$, which represents the input beams coming in at an angle instead of being straight, we send in a collimated Gaussian beam ($w_0 \sim 50\mu$m) and project two different graded index (GRIN) beamsteerer patterns onto the 2D-programmable waveguide. As shown in Fig. 5.11, when there is a misalignment, the beam will only be steered by the GRIN beamsteerer in the front, and not the one in the back. By comparing with theory, we can measure $k_{x,\mathrm{mis}}$ and correct for it.

Calibration of programmable 2D-waveguide parameters: Once the beamshaper is

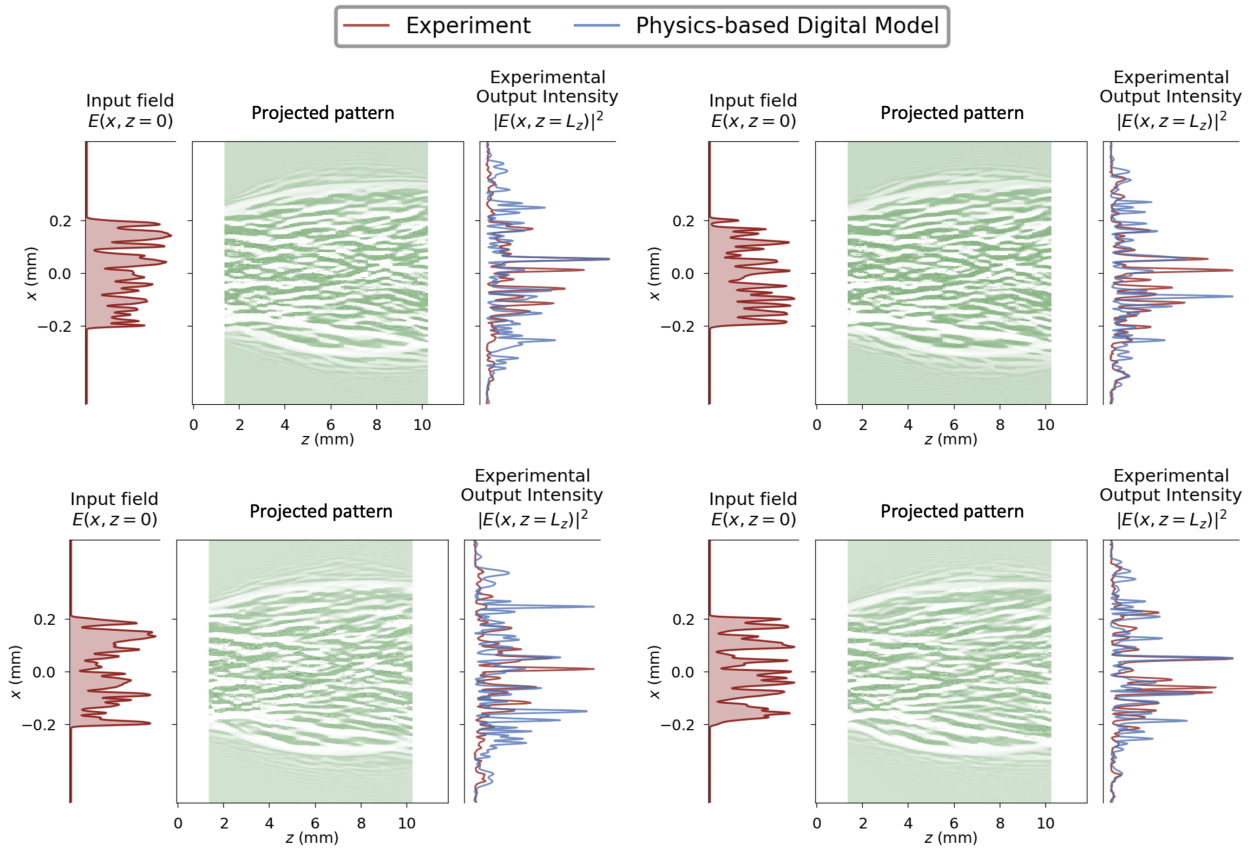

Figure 5.12: **Agreement between the output intensity for the experiment and the purely physics-based model.** The agreement is plotted for four different input field distributions and projected patterns. We observe that there is qualitative, but not quantitative agreement between the outputs of the model and experiment.

calibrated, what remains is to characterize the system parameters of the 2D-programmable waveguide. This is also performed by projecting the GRIN beamsteerers onto the device.

- The maximal refractive-index modulation $\Delta n_{\max}$ can be quantified by measuring the amount of beamsteering exerted by the GRIN beamsteerer. The change in the x-component of the wavevector Δk_x is linearly proportional to the change in refractive index, and thus $\Delta n_{\max}$ can be measured by comparing with theory. We characterized that $\Delta n_{\max} = 10^{-3}$ for the experiment in Fig. 2 of the main manuscript, $\Delta n_{\max} = 0.6 \times 10^{-3}$ for vowel classification, and $\Delta n_{\max} = 0.8 \times 10^{-3}$ for MNIST classification.
- The spatial resolution of the chip d_{kernel} can be measured by varying the width of the GRIN beamsteerer, measuring the amount of beamsteering, and comparing it to

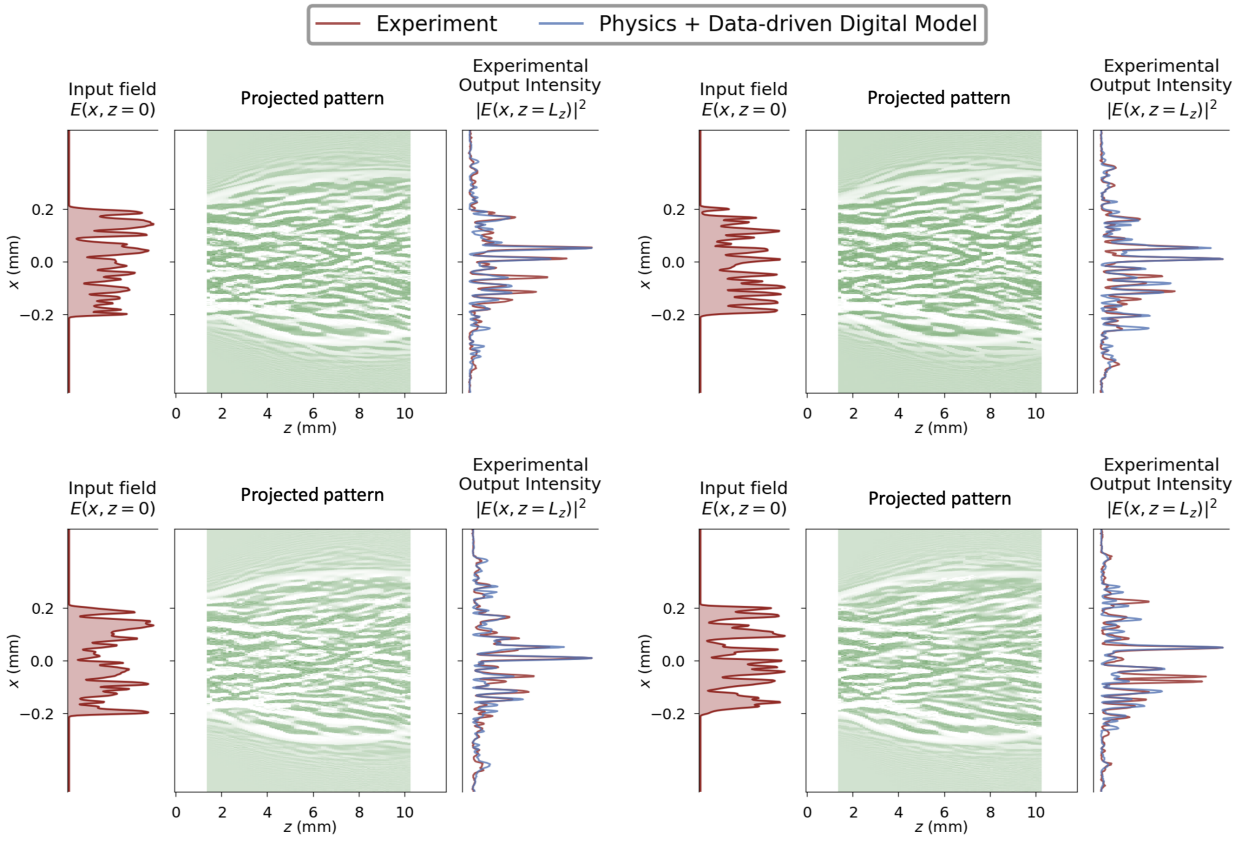

Figure 5.13: **Agreement between the output intensity of the experiment and a model that uses data-driven approaches to fine-tune the physics-based model.** The same input field distributions and projected patterns as in Fig. 5.12 is shown. We observe that the data-driven refinement improves the agreement between the model and experiment significantly.

theory. Using this procedure, we found that $d_{\text{kernel}} = 5\mu$m, which is consistent with the theoretical analysis in Sec. 5.1.4.

In Fig. 5.12, we show the agreement between the physics-based model and the experimental data for different input fields and different projected patterns on the 2D-programmable waveguide. While the experimental output and predicted output agrees qualitatively, we observe quantitative differences, especially towards the edges of the output.

5.4.3 Data-driven approach to fine-tune the digital model

To further improve the agreement between the model and the experiment, we used a data-driven approach to fine-tune the physics-based model. To do so, we added additional trainable parameters into the physics-based model and trained them with input-output pairs that are generated from the experimental data.

The model still solves the beam propagation equation (5.11), but now with the refractive-index modulation being

$$\Delta n(x,z) = \frac{\Delta n_{\max}}{I_{\max}} \left(g(x,z) * \left(I(x,z) \times r(x,z)\right)\right) + \Delta n_{\text{noise}}(x,z). \tag{5.16}$$

Here, $*$ represents the 2D convolution operation, Δn_{noise} denotes a background noise refractive-index distribution, and $r(x,z)$ is another trainable parameter that quantifies the spatial dependence of Δn induced by a given amount of illumination. The other variables in (5.16) are defined in the text following (5.11). We hypothesize that Δn_{noise} may be attributed to charge noise in the lithium niobate slab waveguide, previously observed in experiments involving light propagation through lithium niobate, which resulted in effects such as wavefront damage [138]. The parameter $r(x,z)$ accounts for non-uniform illumination in the imaging setup, specifically referring to the spatial variance in power absorbed by the photoconductor across the field-of-view of the 2D-programmable waveguide, even when $I(x,z)$ is intended to be uniform. Additionally, $r(x,z)$ accounts for film thickness variations in the device stack and the potential presence of small domains within lithium niobate where the crystal axis may be flipped [139]. In summary, both Δn_{noise} and $r(x,z)$ allow the model to address nonidealities associated with lithium niobate, the experimental setup, and the device fabrication.

Finally, we added a trainable output coupling efficiency term $B_{\text{mis}}(x)$ to model for imperfections in our measurement of the output intensity ($I_{\text{meas, out}}(x) = B_{\text{mis}}(x)|E(x, z = L_z)|^2$). We also made the previously introduced input coupling efficiency term $A_{\text{mis}}(x)$ in (5.15) a trainable parameter.

To collect the training data for the data-driven model, an important detail is to obtain the right distribution of projected patterns that are encountered during machine learning tasks. To do so, we in-silico trained the 2D-programmable waveguide on multiple different variations of the MNIST classification task, with permuted inputs and outputs. This way, we obtained a large number of distinct projected patterns, which we then used to train the data-driven model. For the input fields, we sent in input field distributions with random amplitude and phase in a pixel mode basis. In other words, they were generated with (5.12) with x_i being random in both amplitude and phase. Using many experimentally measured input-output pairs, we trained all of these additional parameters ($\Delta n_{\text{noise}}(x, z)$, $r(x, z)$, $B_{\text{mis}}(x)$, $A_{\text{mis}}(x)$) on a digital computer with the backpropagation algorithm.

In Fig. 5.13, we show the agreement between the experimental data and the physics-based model that is refined with a data-driven approach. We see that the agreement has improved significantly compared to the purely physics-based model, especially at the edges of the output (at large $|x|$).

5.5 Machine learning

In this section, we present additional information regarding the machine learning tasks that we performed on the 2D-programmable waveguide.

5.5.1 Computational model of optical-neural-network demonstrations

In this section, we explain the computational model of the optical neural network demonstrations performed by the 2D-programmable waveguide. For concreteness, we focus on the vowel classification task, which has a 12-dimensional input and 7-dimensional output, and

the MNIST classification task, which has a 49-dimensional input and 10-dimensional output. At the end of the section, we argue that the computation performed by the 2D-programmable waveguide is similar to that of a 1-layer neural network, and we use this to characterize the computational complexity of the device.

The input vector $\mathbf{x}$ is encoded in the amplitudes of 12 (49 for the MNIST task) spatial Gaussian modes via (5.12). Since each Gaussian mode is normalized, this can be abstracted

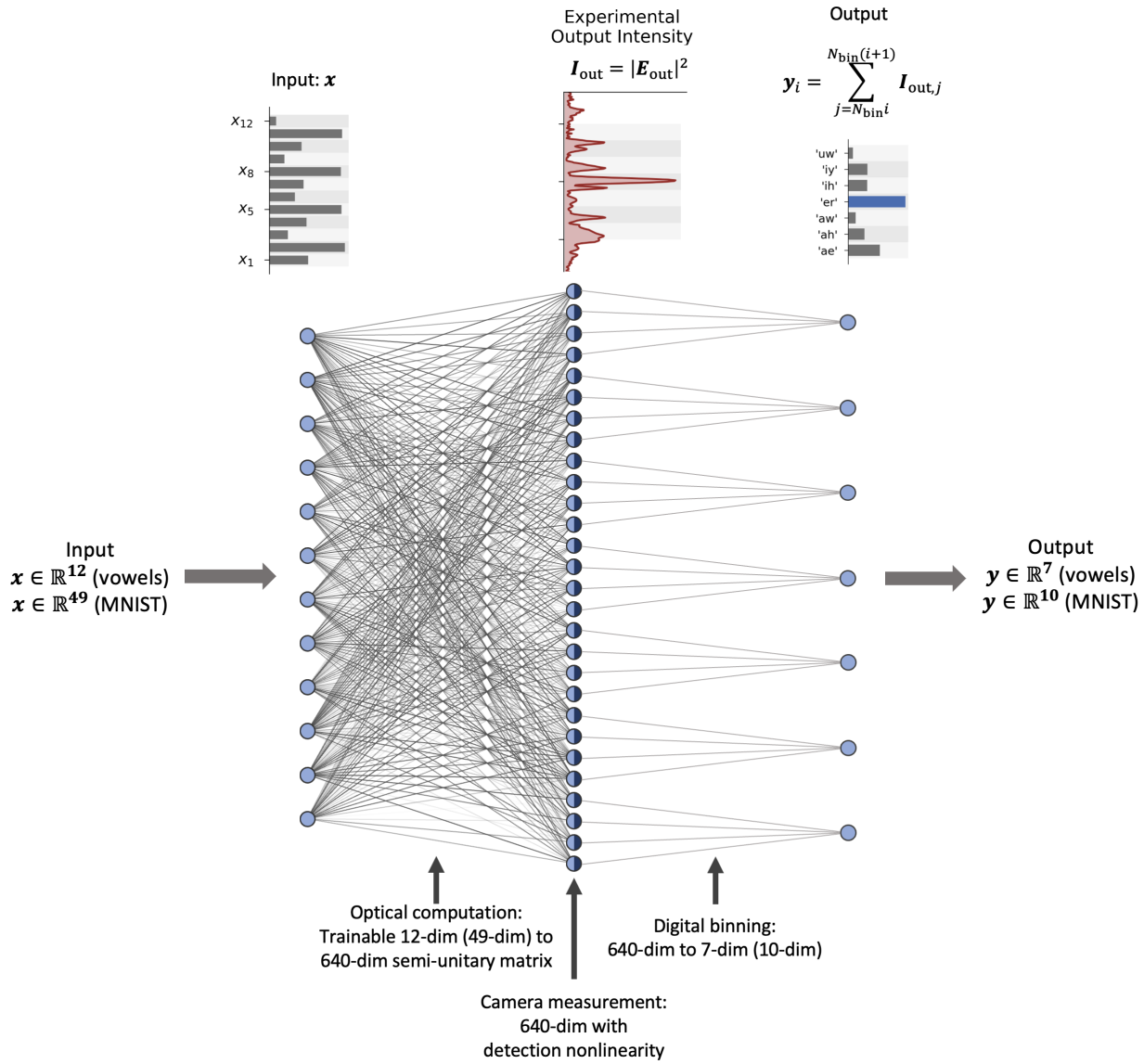

Figure 5.14: **Schematic for the computational model of the ONN demonstrations in this work.**

as a unitary relation: $\mathbf{E}_{\text{in}} = \mathbf{U}_{\text{in}}\mathbf{x}$, where $\mathbf{E}_{\text{in}}$ represents the input field in a pixel basis. Due to the low propagation loss, the field propagation through the device is also a unitary operation, that takes the input field $\mathbf{E}_{\text{in}}$ and returns the output field $\mathbf{E}_{\text{out}}$. Thus, the output field is given by $\mathbf{E}_{\text{out}} = \mathbf{U}_{\text{prop}}\mathbf{E}_{\text{in}}$. In this notation, we see that by training the refractive-index distribution, we are able to change the unitary matrix $\mathbf{U}_{\text{prop}}$ applied by the device $\mathbf{U}_{\text{prop}}(\Delta n(x, y))$. Moreover, since the first two operations can be collapsed into a single unitary matrix, the device's overall operation can be expressed as $\mathbf{E}_{\text{out}} = \mathbf{U}\mathbf{x}$, where $\mathbf{U} = \mathbf{U}_{\text{prop}}\mathbf{U}_{\text{in}}$. Here, $\mathbf{U} \in \mathbb{C}^{640\times 12}$ ($\mathbf{U} \in \mathbb{C}^{640\times 49}$ for MNIST), as we discretize the output field into a 640-dimensional vector defined by the number of pixels on the camera used to measure the output.

We model the measurement of the intensity at the output facet by a quadratic nonlinearity applied to the optical field: $\mathbf{I}_{\text{out}} \propto |\mathbf{U}\mathbf{x}|^2$. After the camera measurement, we digitally bin groups of camera pixels—average pooling in the language of machine learning—to create a lower-dimensional output of 7 (10 for MNIST). In summary, the computation performed can be expressed as

$$\mathbf{y} = \text{AvgPool}(|\mathbf{U}\mathbf{x}|^2) \tag{5.17}$$

where the matrix vector multiplication $\mathbf{U}\mathbf{x}$ is performed optically, the nonlinear activation function is performed by the analog electronics of the camera, and the fixed average pooling is performed by a digital computer. This relation is also shown schematically in Fig. 5.14. We note that the entire computation could be performed without a digital computer, for example, by using a lens to optically perform the fan-in operation, and using a photodetector for each class of the classification task.

We emphasize that although we can train the unitary matrix $\mathbf{U}$ by programming the refractive-index distribution, it is not possible to realize an arbitrary $\mathbf{U}$. This limitation is physically intuitive as the number of parameters in $\mathbf{U}$ (125440 for the MNIST task) exceeds the number of parameters available in the 2D-programmable waveguide (10,000).

While the input-output relation defined in (5.17) is mathematically equivalent to a 2-layer

neural network, the computation it performs is similar to that of a 1-layer neural network due to the second layer being sparse and not trainable, and the square nonlinearity not being significantly nonlinear (compared to activation functions such as ReLu and Sigmoid). An alternative argument is that if the square nonlinearity were replaced with an identity operation, (5.17) would indeed reduce to a 1-layer neural network. Therefore, we characterize the complexity of the computational operation performed by the 2D-programmable waveguide by the number of operations performed by the effective 1-layer neural network, rather than the number of operations in $\mathbf{U}$.

The most complex machine learning task we have undertaken with the 2D-programmable waveguide is the MNIST task, where it achieves an 86% accuracy, moderately close to the 90% accuracy achievable by a 1-layer digital neural network (with 49 inputs and 10 outputs). Hence, we characterize the device as capable of performing 490 multiply-accumulate operations in a single pass through the chip.

5.5.2 Vowel classification

Experiment Settings: The frequency of the AC voltage drive is set to 26 Hz, to increase the speed of data collection. The amplitude of the AC voltage was set to 800 V instead of 1000 V (as in Fig. 2 of the main manuscript), because this experiment was performed at an earlier stage of the project, when we used a more conservative voltage buffer for the device. These two choices led to a decrease in the maximal refractive-index modulation $\Delta n_{\max}$ to 0.6×10^{-3}, compared to 1×10^{-3} shown in Fig. 2 of the main manuscript.

Additional Results: In Fig. 5.15, we show how the wave propagation evolved during the training process. The 4 projected patterns shown on the left corresponds to the same projected patterns shown in Fig. 3 of the main manuscript. The same input vector is also used for consistency, whose correct label is the vowel 'er', which corresponds to the fourth bin in the output. We observe that at initialization, the projected pattern was uniform, and

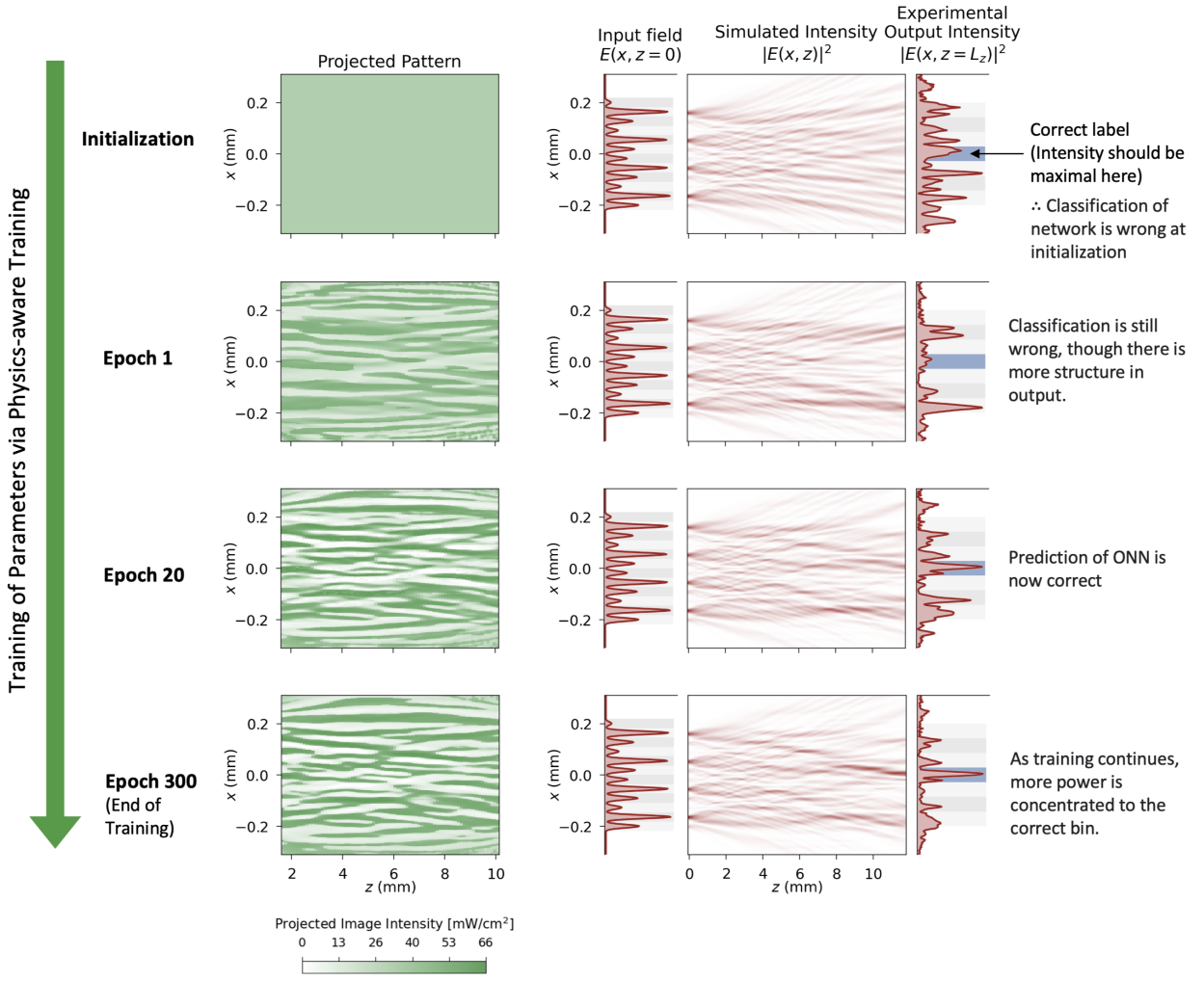

Figure 5.15: **Evolution of wave propagation during physics-aware training.**

the output intensity was unstructured uniformly spread across all of the 7 bins. After just 1 epoch of training, the wave propagation was already significantly altered, where there is more structure in the output intensity.

We emphasize that small changes (to the visual eye) in the projected pattern can lead to large changes in the wave propagation. This is particularly evident in the case of the last two projected patterns, at Epoch 20 and Epoch 300. Although these patterns appear similar, the wave propagation differs significantly. More power is concentrated in the bin that corresponds to the correct vowel by Epoch 300. This highlights the importance of an *in-situ* programmable device for controlling complex wave propagation.

5.5.3 MNIST classification

Experiment Settings: The frequency of the AC voltage drive is set to 26 Hz, to increase the speed of data collection. The amplitude of the AC voltage was set to 1000 V, consistent with Fig. 2 of the main manuscript. Therefore, there is a slight decrease in the maximal refractive-index modulation $\Delta n_{\max}$ to 0.8×10^{-3}, compared to 1×10^{-3} shown in Fig. 2 of the main manuscript.

Additional Results: In Fig. 5.16, we show additional results regarding the MNIST classification task. Fig. 5.16A shows the parameters of 2D-programmable waveguide, the projected pattern, after training. When compared to the projected pattern for the vowel classification, we observe that more the of the pattern takes on more extreme values, where many pixels are at maximum or minimal intensity. This suggest that the MNIST task is more difficult than the vowel classification task, pushing the capabilities of the device closer to its limit.

In Fig. 5.16B, we show the wave propagation for the particular input MNIST image shown in Fig.4 of the main manuscript. The wave propagation is more complex than that in the vowel task. We note that we chose the width of the Gaussian input modes (w_0 in (5.12)) to be 6 µm, to limit the diffraction of the modes, while the spacing between the modes is 8.2 µm. Thus, the input modes used in (5.12) overlap, resulting in the input field shown in Fig. 5.16B. This is not a fundamental limitation and will be addressed in future work.

Finally, in Fig. 5.16C, we present the training curve for the MNIST classification task. We observe fluctuations in accuracy; for instance, at epoch 8, it reached 87.2%. The accuracy is 86.2% after 10 epochs of training, compared to the 90% accuracy achieved by a 1-layer digital neural network on the same 49-dimensional MNIST task.

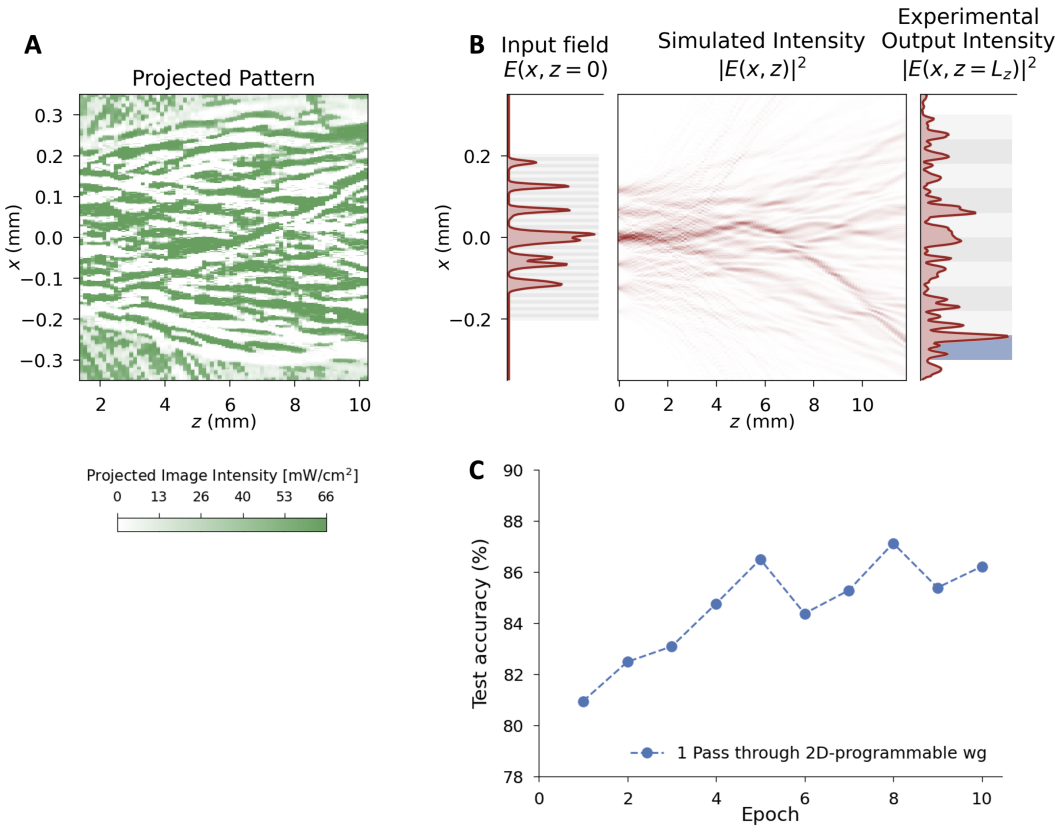

Figure 5.16: **Detailed MNIST classification results.** **A**: Projected pattern after 10 epochs of training. **B**: Illustration of wave propagation in 2D-programmable waveguide for the particular input MNIST image shown in Fig.4 of main manuscript. **C**: The accuracy of the MNIST classification task on the test set as a function of the number of training epochs.

5.6 Comparison with other on-chip optical neural network demonstrations

In this section, we compare our work with other on-chip optical neural network demonstrations*, focusing particularly on demonstrations that perform matrix-vector multiplication (MVM) with a weight-stationary approach, where the matrix elements are encoded in programmable elements on the chip.

We acknowledge that various metrics can be used to compare the performance of different

*We have not included references, such as Ref. [140], if they don't report machine-learning-inference accuracy.

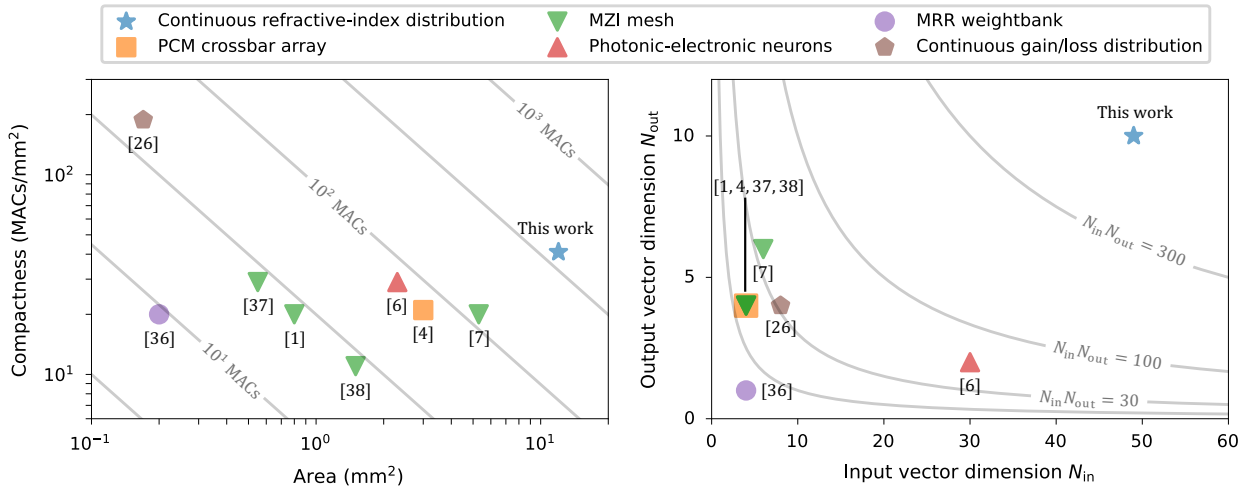

Figure 5.17: **Comparison between different on-chip optical neural network demonstrations.** Data from Table 5.1. Left: Area and compactness of various on-chip ONNs. Grey lines are contours of total MAC operations, which represents the number of multiply-accumulate operations executed in a single pass through the device. Right: Input and output vector dimension of various on-chip ONNs. Grey lines are contours of constant $N_{\text{in}} \times N_{\text{out}}$, which corresponds to the number of MAC operations for single-layer ONNs.

on-chip optical neural network demonstrations. Here, we focus on the number of multiply-accumulate operations (MACs) executed in a single pass through the device. For single-layer neural networks, the number of MAC operations is determined by the product of the input and output dimensions of the MVM operation. In multi-layer neural networks, it is the sum of each layer's MAC operations. For ONNs utilizing frequency multiplexing, the number of MAC operations is multiplied by the number of channels, N_{ch}, used for frequency division multiplexing. Additionally, we also compare the devices' compactness, measured by the number of MAC operations per unit area. Here, "area" specifically refers to the portion of the chip where light propagates to perform the MVM and excludes regions dedicated to electrical wiring and bond pads. In Table. 5.1, we summarize the comparison between our work and other on-chip optical neural network demonstrations, and plot some of the key metrics in Fig. 5.17.

*Multilayer ONNs have been demonstrated in a related work [102] on this platform.

†The first layer of this multilayer ONN performs a convolution operation on the input image. Since it performs dot products on 4 sub-images, each with 12 pixels, we have characterized this first layer as executing 48 MAC operations.

Ref.	MVM Scheme	N_{layer}	N_{ch}	N_{in}	N_{out}	MACs	Area [mm^2]	Compactness [MACs/mm^2]
This work	Continuous refractive-index distribution	1	1	49	10	490	12	41
[88]	MZI mesh	1	1	4	4	16	0.8	20
[83]	MZI mesh	1	1	4	4	16	0.55	29
[77]	MZI mesh	3	1	6	6	108	5.3	20
[53]	MZI mesh	1	1	4	4	16	1.5	11
[91]	Photonic-electronic neurons	3	1	30	2	66†	2.3	29
[78]	PCM crossbar array	1	4	4	4	64	3	21
[118]	Microring resonator weightbank	1*	1	4	1	4	0.2	20
[95]	Continuous gain/loss distribution	1	1	8	4	32	0.17	188

Table 5.1: **Comparison between different on-chip optical neural network demonstrations.** We focus on neural-network-inference demonstrations that perform matrix-vector multiplication (MVM) on-chip with a weight-stationary approach, where the matrix elements are encoded in programmable elements on the chip. N_{layer} is the number of layers (depth) in the ONN. N_{ch} is the number of channels used for frequency division multiplexing. N_{in} and N_{out} are the input and output dimensions of the MVM operation for single-layer neural networks. For multi-layer neural networks with a more complex architecture (such as [91]), N_{in} and N_{out} represent the dimensions of the input and output vectors of the overall ONN. MACs denote the number of multiply-accumulate operations executed in a single pass through the device. MZI and PCM stands for Mach–Zehnder interferometer and phase-change memory, respectively.

Table 5.1 shows that the 2D-programmable waveguide has demonstrated the largest input and output vector dimensions, as well as the highest number of MAC operations among the on-chip optical neural network demonstrations reviewed. In terms of compactness, the 2D-programmable waveguide is also highly competitive, ranking second only to Ref. [95]. This work shares many parallels with ours, notably in its avoidance of discrete programmable photonic elements and in leveraging a continuous, spatially programmable slab waveguide (in their case a spatial gain/loss distribution) to perform neural-network computations. A notable difference between our work and that in Ref. [95] is in the type of light used for computation. We use coherent light, enabling complex-valued matrix-vector multiplication. In contrast, their approach uses incoherent light, which is more robust to phase noise, but limits the applied matrix to real and non-negative elements.

5.7 Future device improvements to the 2D-programmable waveguide

In this section, we discuss potential device improvements to the 2D-programmable waveguide and their implications for neural-network computation. We discuss different directions to improve the device, such as incorporating all-optical nonlinearity, increasing the refractive-index modulation depth, and potential routes to a fully integrated 2D-programmable waveguide. We also discuss how the device can be modified to realize a programmable nonlinear medium. Finally, we present simulation results of performing unitary matrix operations with a prospective, scaled up 2D-programmable waveguide.

5.7.1 Increasing the depth of the neural network computation

In our current work, our chip performs a single-layer neural-network computation (i.e., having a depth of 1). Increasing the depth of the neural network run in a single pass through the chip would require performing nonlinear activation functions on-chip [77, 91, 102]. Given that our chip's slab waveguide already uses a nonlinear material, lithium niobate, one could periodically pole the lithium niobate in certain regions of the chip, enabling nonlinear interaction between optical waves in those regions, which can be used to engineer all-optical nonlinear activation functions [141]. Alternatively, multilayer computation can be performed by propagation of waves in a continuous nonlinear medium [93, 94, 106], since programmable nonlinear wave propagation can be mathematically mapped to a deep multilayer neural network [93, 106, 120]. A power-efficient realization of this approach in our platform could involve using an appropriately poled lithium niobate slab waveguide with a cascaded second-order nonlinearity [142].

5.7.2 Increasing the refractive-index modulation

A weakness of our current demonstration device is the relatively small refractive-index modulation ($\Delta n \sim 10^{-3}$) compared to conventional lithographically defined nanophotonics ($\Delta n \sim 1$): the refractive-index modulation one can achieve in any device with our design is ultimately constrained by the material properties of the waveguide core: its electro-optic coefficient and the maximum electric field it can sustain. With lithium niobate, the modulation we measured, $\Delta n \sim 10^{-3}$, could be improved to $\Delta n = 10^{-2}$ by increasing the thickness and improving the material properties of the photoconductor (see Section 5.1.2). However, using an alternative electro-optic material for the waveguide, such as barium titanate [143], highly nonlinear polymers [144], or liquid crystals [145], could allow future realizations to achieve a refractive-index modulation of $\Delta n \sim 5 \times 10^{-2}$ or larger.

5.7.3 Realizing a programmable nonlinear medium

We could also modify our concept of a 2D-programmable waveguide to realize an arbitrarily programmable *nonlinear* medium, where the second-order nonlinear optical susceptibility $\chi^{(2)}(x, z)$ could be programmed in real time. This may be achieved by using a waveguide core material with a large third-order nonlinearity $\chi^{(3)}$ (such as silicon [146], silicon nitride [147], or tantalum pentoxide [148]). Using the principle of photoconductive gain, we can spatially program a strong (quasi-DC) electric field $E_{\mathrm{DC}}(x, z)$ in the waveguide core that can induce an effective second-order nonlinearity $\chi^{(2)}_{\mathrm{eff}}(x, z)$ [149]. Such induced second-order nonlinearities can be quite strong: Ref. [149] demonstrated that the nonlinearity induced in silicon ($\chi^{(2)}_{\mathrm{eff}} = 41\,\mathrm{pm/V}$) is comparable to that of lithium niobate ($\chi^{(2)} = 50\,\mathrm{pm/V}$). This field-induced second order nonlinearity can be thought of as the higher-order equivalent of the linear refractive-index modulation (5.1), and is mathematically described by

$$\chi^{(2)}_{\mathrm{eff}}(x, z) = 3\chi^{(3)} E_{\mathrm{DC}}(x, z) \tag{5.18}$$

By programming $\chi^{(2)}_{\text{eff}}(x,z)$, one could reconfigurably phase-match different nonlinear optical processes in real time. This enables the development of programmable nonlinear optical devices, including highly tunable classical and quantum light sources, and more generally, enables programmable nonlinear processing of optical signals.

5.7.4 Potential routes to a fully integrated 2D-programmable waveguide

In this section, we discuss potential routes to a fully integrated 2D-programmable waveguide that would be more compact. As discussed in Section 5.2, the components occupying the most space in the current experiment are the beamshaper, the detection setup, and the projector. Thus, we address how to integrate these components on-chip in this section.

Fig. 5.18 shows a conceptual schematic of a fully integrated 2D-programmable waveguide that could be built in the future. Input fields can be created with an array of on-chip lithium niobate modulators. Due to the strong electro-optic effect in lithium niobate, these modulators have been shown to have high bandwidth at low operating voltages [69]. These inputs can be directed into a region of the chip featuring the programmable refractive-index distribution. It is important to fan out the waveguide mode to a larger width before entering this region, as the programmable 2D-waveguide region can only control beams with a finite spatial bandwidth Δk_x, due to the low numerical aperture associated with the maximum refractive-index modulation in the 2D-programmable waveguide. The detection setup can be replaced with an array of on-chip photodetectors. Although integrating photodetectors on lithium niobate is generally challenging, there has been progress, including heterogeneously integrated III-V materials [150], as well as the integration of 2D-material [151], and tellurium thin films [151, 152]. High bandwidths of over 40GHz have been demonstrated with these approaches [152]. For both the modulator and the detector, it is crucial to add the capability to lithographically etch the lithium niobate layer in the 2D-programmable waveguide. This

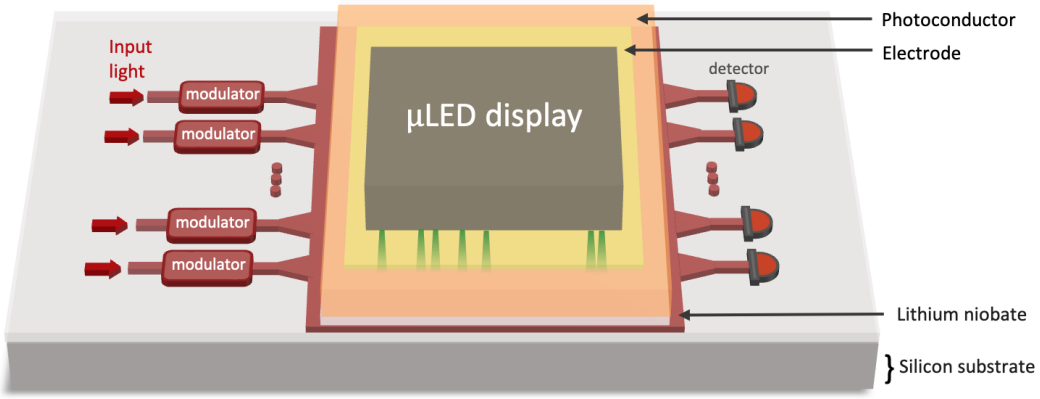

Figure 5.18: **Conceptual schematic of a future vision for a fully integrated 2D-programmable waveguide.** The device, which can be built in the future to miniaturize many of the components in the current setup, consists of an array of on-chip lithium niobate modulators to create the shaped input light, a region of the chip with the programmable refractive-index distribution, and an array of on-chip photodetectors to measure the output of the computation. The photoconductor is programmably illuminated by a micro-LED display, which is bonded onto the electrode of the 2D-programmable waveguide.

should be feasible by depositing the photoconductive layers (via PECVD) onto a designated region, separate from the region where the lithium niobate is etched for modulators and detectors.

In our current experiment, we use a DMD, illuminated by a green LED to project a programmable illumination pattern onto the 2D-programmable waveguide. As shown in Fig. 5.18, this programmable illumination can be projected in a more compact manner by bonding a micro-LED display (µ-LED) directly onto the electrode of the 2D-programmable waveguide. Among the different display technologies, which includes LCDs, organic LEDs (OLEDs), and micro-LEDs, micro-LEDs are the most promising, because they offer high brightness and small pixel pitches. In fact, displays with 30,000 PPI and a brightness of 100,000 nits have been demonstrated [153]. This corresponds to a pixel pitch of 0.87 µm, and assuming that 50% of the light emitted is absorbed by the photoconductor, this corresponds to an optical intensity of 45mW/cm^2. Thus, such micro-LEDs can deliver sufficient optical power to the photoconductor, for the operation of the 2D-programmable waveguide (see Fig. 5.5). The key technical challenge will be in getting the LEDs close enough to the

waveguide to ensure that the emitted light does not diffract significantly before it is absorbed by the photoconductor. In order to do so, the protective layer (encapsulation) over the LEDs will first need to be either removed or partially etched down to a thin layer. Alternatively, micro-LEDs with monolithically integrated micro-lenses can be used [154].

5.7.5 Simulations of unitary matrix operations with a prospective, scaled-up 2D-programmable waveguide

In this section, we present simulations of performing large-scale unitary matrix operations with a 2D-programmable waveguide that has a length of 6 cm and a maximal refractive-index modulation of 5×10^{-3}. Both of these parameters are five times larger than those of our current experiment ($L_z = 1.2\,\text{cm}$ and $\Delta n_{\max} = 1 \times 10^{-3}$).

In order to perform unitary operations, we decompose the output field in discrete output modes. Thus, we do not use the output decoding used in the experiments, where we measure and bin the output intensity distribution, but instead assume that there are output waveguides that filter the output field into discrete modes (this is also the output encoding shown in Fig. 5.18). We perform simulations for input and output vectors of dimensions $N = 100$ and 150, for random unitaries sampled from the Haar measure. The input and output modes are assumed to be the same. For $N = 100$, the width of the Gaussian modes is $w_0 = 5\,\mu\text{m}$, while the spacing is 16 µm (thus the inputs span a distance of 1.6 mm). For $N = 150$, the width of the Gaussian modes is $w_0 = 4\,\mu\text{m}$, while the spacing is 13 µm (thus the inputs span a distance of 2 mm). We note that the matrix operations are performed up to a rescaling factor, which represents loss. The $N = 100$ was performed with a transmission of 16% (loss of 84%), while the $N = 150$ was performed with a transmission of 4% (loss of 96%). Adding this rescaling factor to the training eases the burden on the programmable wave propagation, as it no longer has to funnel all of the power into the discrete and incomplete basis of output modes.

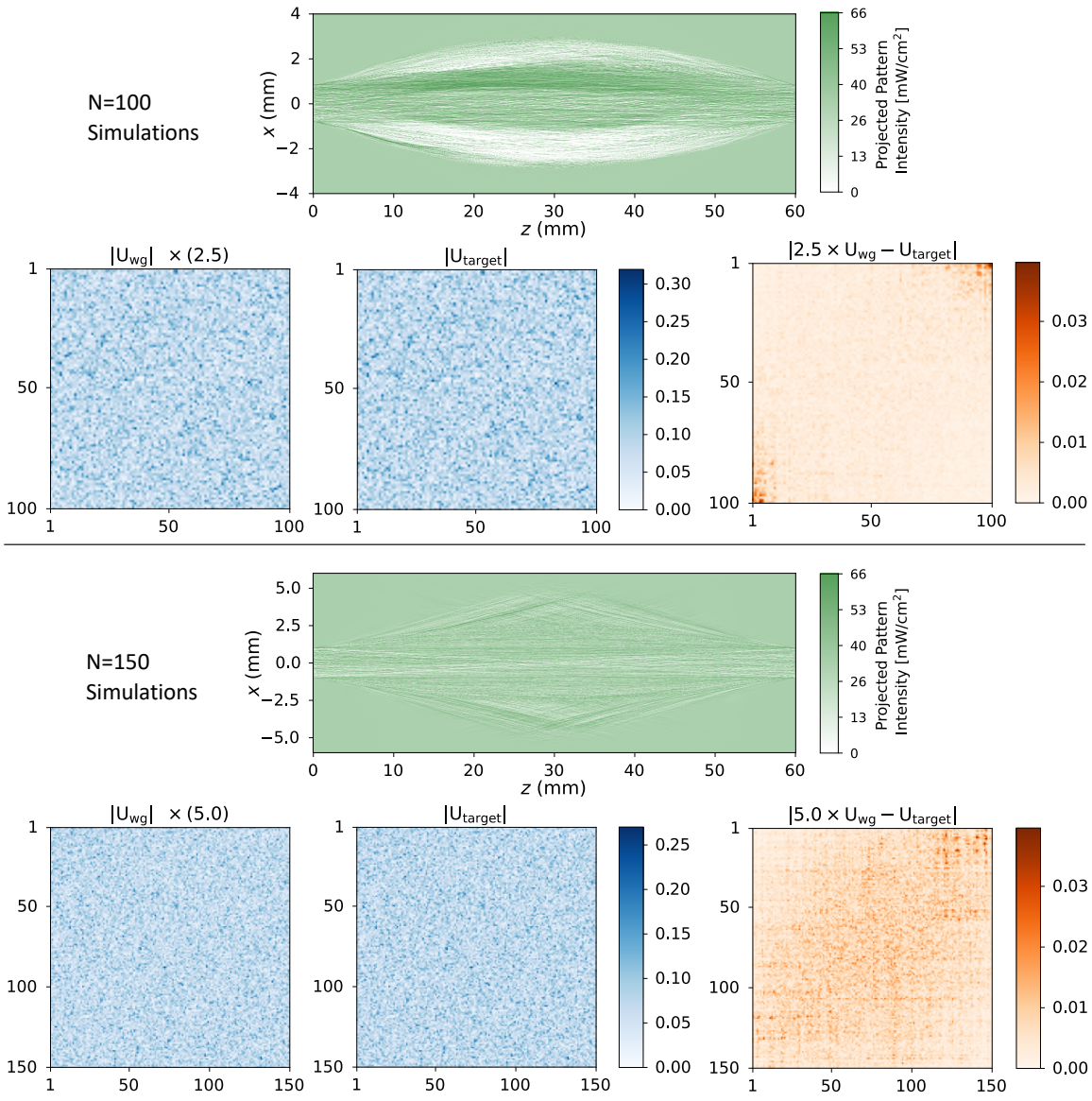

Figure 5.19: **Simulations of large-scale unitary matrix operations with a prospective, scaled-up 2D-programmable waveguide.** We train the waveguide to perform unitary matrix operations on input and output vectors of dimensions $N = 100$ and 150. The target unitary matrices are random unitaries sampled from the Haar measure. U_{wg} is the matrix operation performed by the 2D-programmable waveguide after training. We note that the multiplicative factor (2.5 for $N = 100$, and 5 for $N = 150$) is a rescaling factor, indicating that the matrix operation is performed with optical loss (loss of 84% for $N = 100$ and 96% for $N = 150$). In these simulations, we assume a $\Delta n_{\text{max}} = 5 \times 10^{-3}$, five times the experiment's current value, and a chip length, $L_z = 6\,\text{cm}$, also five times longer than our current experiment.

The results are shown in Fig. 5.19. We observe that the 2D-programmable waveguide is able to execute the unitary matrix operations with high fidelity for the $N = 100$ case. For

$N = 150$, the fidelity is lower, but the matrix elements of the target unitary matrix and the matrix performed by the 2D-programmable waveguide still agree to within 15%.

CHAPTER 6

OUTLOOK

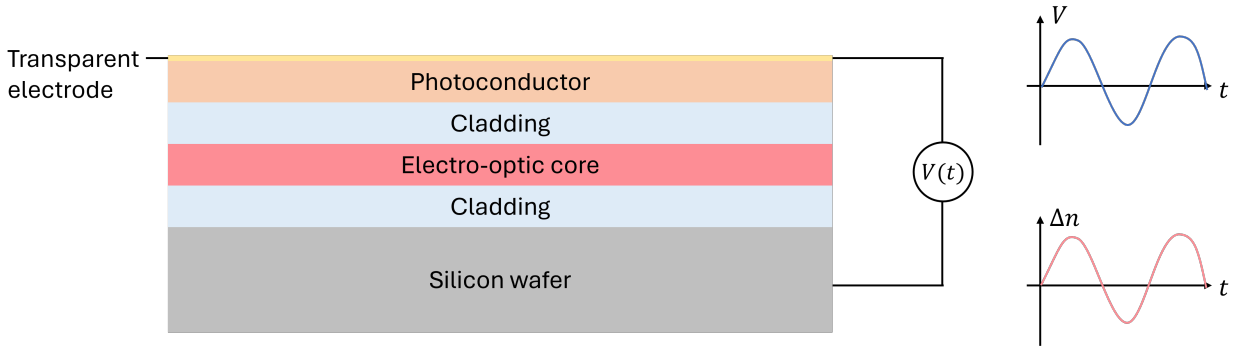

Figure 6.1: **Schematic of current 2D-programmable waveguide.** This is the programmable waveguide presented in chapter 4. As indicated on the right by the Δn vs time graph, a 2D-programmable waveguide like this only periodically achieves the desired maximum refractive index change (Δn).

The purpose of the 2D-programmable waveguide presented in Chapter 4 is to make on-chip optical neural networks more space-efficient, such that large matrix-vector multiplications beyond the break-even point (as explained in section 3.1) may be realized. One obvious point to improve is the index contrast, which is currently at a meager $10^{-}3$. The dimension of the matrix multiplication that may be realized scales at least linearly, and potentially super-linearly, with the maximal index contrast [105]. Increases in the refractive index may be realized by improved photoconductors (see. Fig. 5.3) or better electro-optic materials.

Another point to improve is the low "clock-rate" at which the device is working, resulting in an effectively very low duty-cycle. One promising feature of optics is its high bandwidth [68], which is not compatible with the current implementation of the device (Sec. 5.1.2). An improvement might come from the development of low-loss and conductive claddings, which could realize a device operating with 100% and hence process inputs of practically arbitrary bandwidth. Here, I present two promising continuations of my research that could realize these improvements.

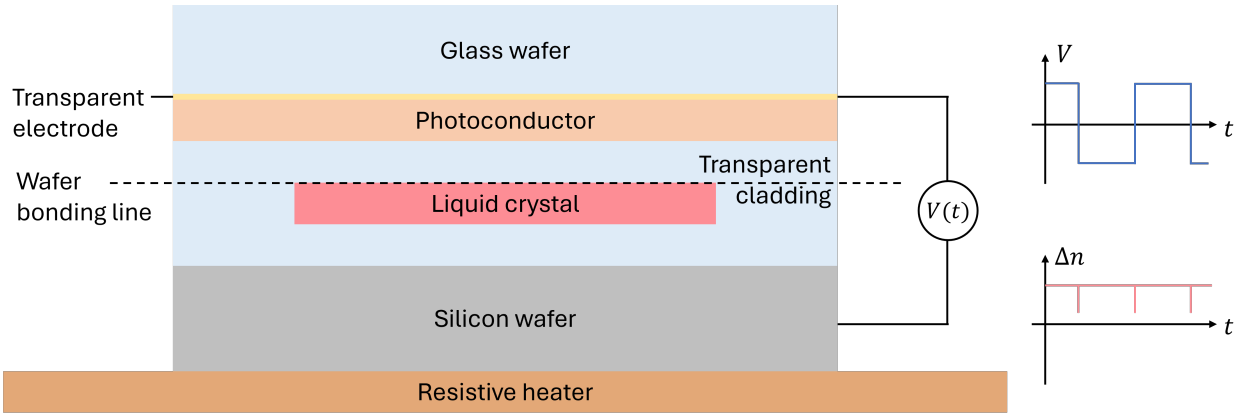

Figure 6.2: **Schematic of liquid crystal-based 2D-programmable waveguide.** Compared to the programmable waveguide presented in chapter 4, the guiding layer is replaced by a liquid crystal, which is heated to just above the nematic/para-nematic phase transition by a resistive heater, and the applied voltage is a square-wave. The liquid crystal is filled into an etched cavity which is sealed via wafer bonding. As indicated on the right, by the Δn vs time graph, a 2D-programmable waveguide like this could achieve a high duty cycle of around 99.9%.

6.1 A liquid-crystal based 2D-programmable waveguide

Physics of programmable liquid crystal waveguides

The device presented in chapter 4 is the first device of its kind that operates at such a scale, but would tremendously benefit from a larger programmable refractive index contrast. For example Nikkah et al [81] realized fixed devices with an index contrast of 0.3, still permitting fast 2D simulations, but enabling much more space-efficient devices. Liquid crystals might be capable of creating such a large index change programmably in experiment. Another important improvement would be the that a liquid crystal based 2D-programmable waveguide could operate with a duty cycle of around 99.9%, i.e. programmably create waveguides that exist almost permanently, instead of periodically. For liquid crystals, this is possible due to a fortunate combination of material properties.

Liquid crystal's extraordinary electro-optic coefficients

The reason we are interested in liquid crystal waveguides is their extraordinarily strong electro-optic effect. At room temperature, at which the liquid crystal is in the "nematic phase", the birefringence, i.e. the change in refractive index between different orientations of the crystal, is about 0.1 to 0.3. By applying electric fields as weak as < 1 V/um, the orientation of the liquid crystal can be switched at <kHz speed. This effect is used in most spatial light modulators, for example. At higher temperatures, in the paranematic or isotropic phase, the birefringence of the crystal vanishes, but at temperatures just above the phase-transition, a strong Kerr effect emerges. By applying electric fields as strong as 10 V/µm, the refractive index of the liquid crystal can change by up 0.01, and can be switched at MHz speeds. There is a tradeoff between speed and index change: Over a temperature change of a few degrees Celsius, the Kerr coefficient drops by a factor of 10, while the switching speed increases by a factor of 10. The closer the temperature is held to the phase transition, the stronger and slower the refractive index change will be [155].

Compatibility with nanofabrication processes

Liquid crystals are organic compounds and deteriorate at high temperatures above $200 - 300$C. They appear therefore incompatible with many nanofabrication processes. There is a widely used and simple workaround, which is that cavities/enclosures are etched into nanofabricated chips and are filled with the liquid crystal through a little filling hole post-fabrication. The liquid crystals suck themselves into the cavities via capillary forces. The process used by Blasl [155] is to create a-Si filled cavities, etch them with XeF_2 after fabrication and then fill the cavities with the liquid crystal. More rudimentary processes are possible: A thin plastic shim (≈ 10µm) can be cut into shape and placed between between two wafers to act as a spacer. Bonding the wafers together with epoxy creates a cavity and a liquid crystal slab waveguide. Additional layers can be added to the wafers as desired.

High duty-cycle liquid crystal waveguides

It is possible to create almost stationary electro-optically induced waveguides in liquid crystals despite the liquid crystals being sandwiched between highly insulating silicon-dioxide claddings. This is possible due to peculiarities of liquid crystals and not transferable to 2D-programmable waveguides with lithium niobate. The idea is as follows: Liquid crystals are more or less perfectly insulating for a short time (around $1 - 100$ms, depending on the type and purity of the liquid crystal). By using a rectangular alternating voltage that switches faster than $1 - 100$ms, the electric field strength across the liquid crystal waveguide can be held almost perfectly constant, although the direction of the electric field will flip at the frequency of the alternating voltage. When using an isotropic liquid crystal that exhibits a Kerr effect, the direction of the electric field will not affect the index change since $\Delta n \propto E^2$. Therefore, a waveguide can persist for more than 99.9% of time (except for about 1µs needed to reorient the liquid crystals every $1 - 100$ms, as the alternating rectangular voltage flips).

Foreseeable problems of a liquid-crystal based 2D waveguide

A number of technical problems will need to be solved before such a device can be created:

- Low index contrast between liquid crystal waveguide and silicon dioxide claddings:
  - The base refractive index of the liquid crystal 5CB is 1.58, while the index of silicon dioxide is 1.44 at 1550 nm. This makes the mode much larger than in a lithium niobate waveguide, reducing the mode overlap with the liquid crystal, or requiring a very thick liquid crystal, or a thick cladding. Either solution would require a thicker photoconductor, as the ratio of thickness between the two layers determines how effectively we can modulate the applied electric field. This seems solvable, especially if conductive claddings can be developed that have a conductivity about as low as the photoconductor. This would eliminate the necessity to use a photoconductor that is thicker than the waveguide.

- Development of a photoconductor for lower field strengths
  - Liquid crystal require much weaker applied electric fields to change their refractive index than the lithium niobate used in Chapter 4 ($<$ 1 V/um in the nematic phase, or $<$ 10 V/um in the isotropic phase, compared to $\approx$ 50 V/um in lithium niobate). This is in principle highly desirable, but necessitates the development of a photoconductor that works at such field strengths. The photoconductor used in Chapter 4 has a highly field-dependent conductivity (Poole-Frenkel effect) and is unlikely to work at the smaller fields required for liquid crystals.
  - The relative permittivity of 5CB is 11 and its conductivity $10^{-}9$ S/cm. This conductivity is slightly too high for our current photoconductors, ideally we would want a photoconductor that is more conductive that that in the bright and dark state. Currently, our photoconductors are less conductive in the bright state.
  - To operate waveguides with rectangular voltages as proposed in [155], we would a photoconductor that is very conductive in the bright state ($\ll 10^7$ Ohm.cm in the bright state; currently, our photoconductors have a resistivity of around 10^8 Ohm.cm in the bright state)
- Developing temperature-control: The liquid crystals needs to be held precisely at a temperature of around 37C, just above the nematic-isotropic phase transition, constant to about 0.1C to achieve a consistent Kerr-effect. This is in principal not very challenging but will involve careful calibration and characterization of materials.

6.2 A 2D-programmable waveguide with conductive claddings

In section 5.1.2, the programmable refractive index contrast was derived under the assumption of the claddings being perfect dielectrics. The important quantities appearing in Eq. 5.2 change markedly when assuming perfectly conductive claddings and a constant applied

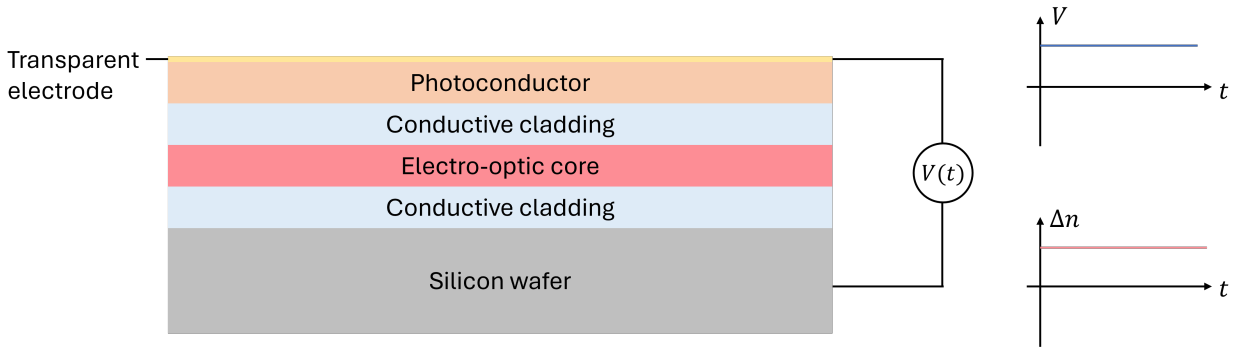

Figure 6.3: **Schematic of a conductive cladding 2D-programmable waveguide.** Compared to the programmable waveguide presented in chapter 4, the claddings are conductive and the applied voltage is constant. As indicated on the right, by the Δn vs time graph, a 2D-programmable waveguide like this could achieve a duty cycle of 100%.

voltage, instead of an alternating voltage.

$$V_{\mathrm{co}} = V_{\mathrm{applied}} \frac{R_{\mathrm{co}}}{R_{\mathrm{co}} + R_{\mathrm{pc}} + R_{\mathrm{cl}}} = V_{\mathrm{applied}} \frac{\rho_{\mathrm{co}} d_{\mathrm{co}}}{\rho_{\mathrm{co}} d_{\mathrm{co}} + \rho_{\mathrm{cl}} d_{\mathrm{cl}} + \rho_{\mathrm{pc}} d_{\mathrm{pc}}}, \tag{6.1}$$

where co, cl, and pc stand for core, cladding, and photoconductor, respectively, R denotes resistance, ρ is resistivity, and d the thickness of the respective layer. The most severe limitation on the maximal refractive index in Chapter 4 is the inability to properly switch off the electric field in the device's core. In Eq. 6.1, this is more easily achieved by a photoconductor whose dark resistance $\rho_{\mathrm{pc}}^{\mathrm{dark}} \gg \rho_{\mathrm{co}} \frac{d_{\mathrm{co}}}{d_{\mathrm{pc}}}$. The development of suitably conductive claddings and a photoconductor could increase the maximally achievable refractive index contrast to the limit of the electro-optical material. In case of lithium niobate, this limit is 10^{-2} [114], ten times higher than what we achieved in Chapter 4. Conductive claddings could be made from phosphor silicon glasses, zinc oxide, fluorinated tin oxide, indium tin oxide, tungsten oxide, or silicon oxynitrides [156].

This idea will not come without tradeoffs: Too conductive claddings would be disadvantageous for two reasons. 1), with increasing cladding conductivity, the optical losses will increase, as for example described by the Drude model. In practice, there exists a wide regime in which the cladding conductivity should be high enough to be treated perfectly conductive in the circuit model while not being a significant source of loss. For example, intrinsic silicon is conductive enough and exhibits very low optical loss (but cannot be used

as a cladding because of its high refractive index). More importantly, 2), a perfect conductor will be at the same electrical potential at any point in the conductor, therefore erasing any spatially patterned electric field. One will need to ensure that the skin depth of the conductive cladding is much larger than its thickness. An analysis similar to the one performed in section 5.1.4 will be needed to ascertain whether a good compromise between cladding conductivity and spatial resolution can be found.

Lastly, the RC constant associated with this circuit is potentially very large as the dark resistivity of the photoconductor is required to be large. This might prove challenging for training a refractive index contrast iteratively, e.g. with a gradient-descent algorithm.

6.3 Programmable photonic devices for communication networks

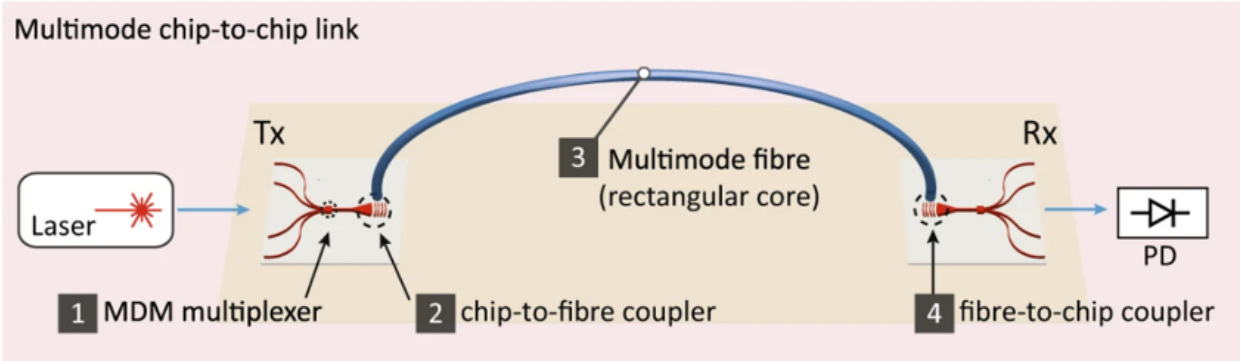

Figure 6.4: **Schematic of a multimode chip-to-chip link.** Multimode chip-to-chip links promise to increase the bandwidth of communication links up to N-fold by transmitting signals encoded in N spatial modes simultaneously. The mode division multiplexer (MDM) (1) converts the modes of N single-mode waveguides, each modulated by a transmitter, into the N modes of a multimode waveguide. A chip-to-fibre coupler couples those modes to a multimode fiber, which sends the signal to a receiver where the process is repeated in reverse. Reprinted from Ref. [157].

Communication networks use optical signals because of the possibility to communicate over long distances with little energy and high data rates. In long-range communication, high data rates are achieved via wavelength-division multiplexing. On the transmitter end, multiple signals are encoded by the modulation of multiple lasers operating at slightly different

wavelengths. The modulated signals are combined via a "multiplexer" into a single optical fiber which transmits all wavelengths at once. On the receiver end, a "demultiplexer" separates the wavelengths and each signal is individually demodulated. For a fixed modulation speed, this procedure speeds up the data communication rate between transmitter and receiver by a factor equal to the number of different wavelength signals. A typical dense wavelength-division multiplexing scheme (DWDM) uses about 80 channels separated by 50 GHz in the C-band (1530 - 1565 nm).

The number of available wavelength channels is, among other things, limited by the availability of amplifiers, dispersion, and the eventual onset of nonlinear effects once the energy density inside the fiber becomes too high.

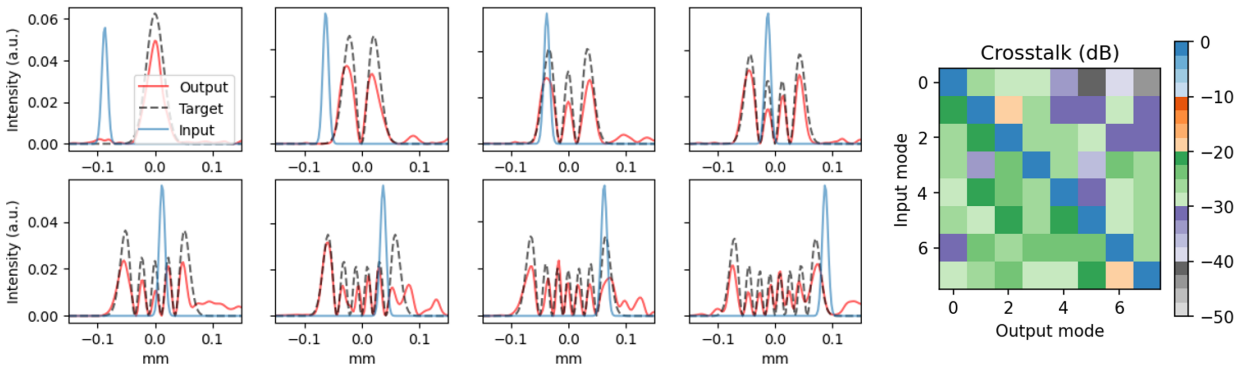

Figure 6.5: **Visualization of simulated mode conversion on 2D-programmable waveguide.** Left: Input, output, and target modes for eight channels. Right: Crosstalk between modes calculated via overlap integrals of the output and target modes. Crosstalk is < 20 dB for almost all modes.

To further extend the number of independent channels in a communication link and suppress optical nonlinearities, modal-division multiplexing (MDM) is a technique that is proposed in addition to WDM. For MDM, larger-diameter fibers that enable the transmission of multiple orthogonal spatial modes are used (similar to multiple spatial modes supported in slab waveguides as discussed in section 3.2. An overview over a spatially multimode communication link is shown in Fig. 6.4. The link consists of a laser that is split into multiple single-mode waveguides and modulators that modulate each waveguide individually. The single-mode waveguides are combined via a MDM multiplexer into one larger waveguide.

The MDM multiplexer is a mode converter that converts the mode of single-mode waveguide 1 to the first mode of the multimode waveguide, the mode of single-mode waveguide 2 to the second mode of the multimode waveguide, and so on (see a simulation of such mode conversion in Fig. 6.5). The multimode light then propagates to the receiver end, where the process is repeated in reverse and the signals are demodulated.

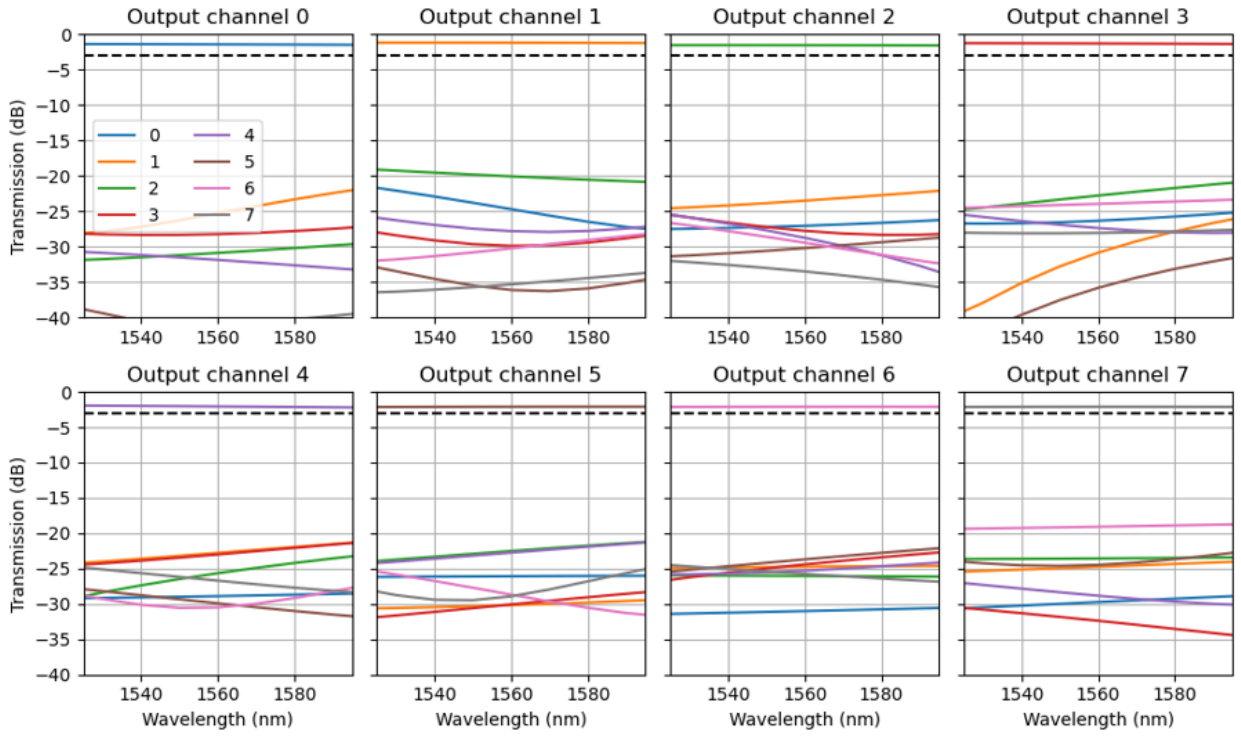

Figure 6.6: **Wavelength-dependence of transmission and crosstalk of mode converter.** Transmission and crosstalk of for each input channel for different wavelengths of the C-band. The crosstalk is almost everywhere $<$ 20dB, while the transmission is higher than 3dB for all wavelengths.

In Chapter 4, we have used controllable linear wave propagation in the 2D-programmable waveguide to perform computation. But all linear optical devices are also mode converters [158]. We can use the 2D-programmable waveguide as a programmable mode converter that can learn the functionality of a MDM multiplexer or demultiplexer.

This idea works well in simulation. In Fig. 6.5, we show a simulation of the device from Chapter 4 trained to perform the conversion of eight single-mode waveguide modes to the eight modes of a multimode waveguide. Albeit not perfect, the crosstalk between modes is

less than −20 dB for almost all modes, and the insertion loss is less than 3 dB for all channels. Moreover, as shown in Fig. 6.6, the transformation is exceptionally broadband, easily spanning the whole C-band and therefore compatible with wavelength-division multiplexing.

A particular advantage of a programmable MDM multiplexer would be the ability to respond to undesired and potentially changing crosstalk between modes in the multimode fiber. Such crosstalk is common in multimode fiber. Communication links can be exposed to mechanical or thermal deformations from the environment. Fibers are sensitive, for example Yang et al [157] observed signal degradation due to modal crosstalk in their lab demo of a MDM communication link. A MDM (de-)multiplexer could be periodically retrained to compensate for unwanted crosstalk. A programmable waveguide could compensate for other unwanted effects as well, such as dispersion or nonlinearity compensation, potentially making it a very versatile part of transmitters or receivers.

BIBLIOGRAPHY

1. Rupp, K. *Microprocessor Trend Data* `https://github.com/karlrupp/microprocessor-trend-data`. 2022.
2. Waldrop, M. M. The chips are down for Moore's law. *Nature* **530,** 144–147. ISSN: 1476-4687. `http://dx.doi.org/10.1038/530144a` (Feb. 2016).
3. Bernstein, L. *et al.* Freely scalable and reconfigurable optical hardware for deep learning. *Scientific Reports* **11.** ISSN: 2045-2322. `http://dx.doi.org/10.1038/s41598-021-82543-3` (Feb. 2021).
4. Epoch AI. *Key Trends and Figures in Machine Learning* Accessed: 2024-07-16. 2023. `https://epochai.org/trends`.
5. Amodei, D. & Hernandez, D. *AI and Compute* `https://openai.com/blog/ai-and-compute/` (2021).
6. Patterson, D. *et al.* Carbon Emissions and Large Neural Network Training. *arXiv:2104.10350* (2021).
7. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* **25,** 1097–1105 (2012).
8. Hooker, S. The hardware lottery. *Communications of the ACM* **64,** 58–65 (2021).
9. Laydevant, J., Wright, L. G., Wang, T. & McMahon, P. L. The hardware is the software. *Neuron* **112,** 180–183. ISSN: 0896-6273. `http://dx.doi.org/10.1016/j.neuron.2023.11.004` (Jan. 2024).
10. Jouppi, N. P. *et al. In-datacenter performance analysis of a tensor processing unit* in *Proceedings of the 44th annual international symposium on computer architecture* (2017), 1–12.
11. Reuther, A. *et al. Survey of Machine Learning Accelerators* in *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (2020), 1–12.

12. Wetzstein, G. *et al.* Inference in artificial intelligence with deep optics and photonics. *Nature* **588,** 39–47 (2020).

13. Balasubramanian, V. Brain power. *Proceedings of the National Academy of Sciences* **118.** ISSN: 1091-6490. http://dx.doi.org/10.1073/pnas.2107022118 (Aug. 2021).

14. Wright, L. G. *et al.* Deep physical neural networks trained with backpropagation. *Nature* **601,** 549–555. ISSN: 1476-4687. http://dx.doi.org/10.1038/s41586-021-04223-6 (Jan. 2022).

15. Stein, M. Making nature compute for us. *TheScienceBreaker* **9.** ISSN: 2571-9262. http://dx.doi.org/10.25250/thescbr.brk667 (Jan. 2023).

16. Marković, D., Mizrahi, A., Querlioz, D. & Grollier, J. Physics for neuromorphic computing. *Nature Reviews Physics* **2,** 499–510 (2020).

17. Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11,** 441 (2017).

18. Onodera, T. *et al.* Scaling on-chip photonic neural processors using arbitrarily programmable wave propagation. *arXiv.* https://arxiv.org/abs/2402.17750 (2024).

19. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521,** 61–64 (2015).

20. Momeni, A. *et al.* Training of Physical Neural Networks. *arXiv preprint arXiv:2406.03372* (2024).

21. Bandyopadhyay, S., Hamerly, R. & Englund, D. Hardware error correction for programmable photonics. *Optica* **8,** 1247. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.424052 (Sept. 2021).

22. Haykin, S. *Neural networks: a comprehensive foundation* (Prentice Hall PTR, 1998).

23. Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. *Neural ordinary differential equations* in *Advances in Neural Information Processing Systems* (2018), 6571–6583.

24. Lin, H. W., Tegmark, M. & Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics* **168,** 1223–1247 (2017).

25. Tanaka, G. *et al.* Recent advances in physical reservoir computing: a review. *Neural Networks* **115,** 100–123 (2019).

26. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nature communications* **2,** 1–6 (2011).

27. Liang, X. *et al.* Physical reservoir computing with emerging electronics. *Nature Electronics* **7,** 193–206. ISSN: 2520-1131. `http://dx.doi.org/10.1038/s41928-024-01133-z` (Mar. 2024).

28. Rahimi, A. & Recht, B. *Random Features for Large-Scale Kernel Machines* in *Advances in Neural Information Processing Systems* (eds Platt, J., Koller, D., Singer, Y. & Roweis, S.) **20** (Curran Associates, Inc., 2007). `https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.

29. Cover, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers,* 326–334 (1965).

30. Hughes, T. W., Williamson, I. A., Minkov, M. & Fan, S. Wave physics as an analog recurrent neural network. *Science Advances* **5,** eaay6946 (2019).

31. Wu, Z., Zhou, M., Khoram, E., Liu, B. & Yu, Z. Neuromorphic metasurface. *Photonics Research* **8,** 46 (2020).

32. Furuhata, G., Niiyama, T. & Sunada, S. Physical deep learning based on optimal control of dynamical systems. *Physical Review Applied* **15,** 034092 (2021).

33. Lin, X. *et al.* All-optical machine learning using diffractive deep neural networks. *Science* **361,** 1004–1008 (2018).

34. Romera, M. *et al.* Vowel recognition with four coupled spin-torque nano-oscillators. *Nature* **563,** 230–234. ISSN: 0028-0836 (Nov. 2018).

35. Grollier, J. *et al.* Neuromorphic spintronics. *Nature Electronics* **3,** 360–370 (2020).

36. Chen, T. *et al.* Classification with a disordered dopant-atom network in silicon. *Nature* **577,** 341–345 (2020).

37. Euler, H.-C. R. *et al.* A deep-learning approach to realizing functionality in nanoelectronic devices. *Nature Nanotechnology* **15,** 992–998 (2020).

38. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Physical Review A* **98,** 032309 (2018).

39. Miller, J. F., Harding, S. L. & Tufte, G. Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence* **7,** 49–67 (2014).

40. Bueno, J. *et al.* Reinforcement learning in a large-scale photonic recurrent neural network. *Optica* **5,** 756–760 (2018).

41. Kaplan, J. *et al.* Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

42. Hermans, M., Burm, M., Van Vaerenbergh, T., Dambre, J. & Bienstman, P. Trainable hardware for dynamical computing using error backpropagation through physical media. *Nature Communications* **6,** 1–8 (2015).

43. Hughes, T. W., Minkov, M., Shi, Y. & Fan, S. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* **5,** 864–871. `https://opg.optica.org/optica/abstract.cfm?URI=optica-5-7-864` (July 2018).

44. Lopez-Pastor, V. & Marquardt, F. Self-learning Machines based on Hamiltonian Echo Backpropagation. *arXiv:2103.04992* (2021).

45. Paszke, A. *et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library* in *Advances in Neural Information Processing Systems 32* (2019), 8024–8035.

46. Hillenbrand, J., Getty, L. A., Clark, M. J. & Wheeler, K. Acoustic characteristics of American English vowels. *The Journal of the Acoustical society of America* **97,** 3099–3111 (1995).

47. Jacob, B. *et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference* in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018).

48. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research* **18,** 6869–6898 (2017).

49. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications* **7,** 1–10 (2016).

50. Nøkland, A. *Direct feedback alignment provides learning in deep neural networks* in *Neural Information Processing Systems* (2016).

51. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574,** 505–510 (2019).

52. Momeni, A., Rahmani, B., Malléjac, M., del Hougne, P. & Fleury, R. Backpropagation-free training of deep physical neural networks. *Science* **382,** 1297–1303. `https://www.science.org/doi/abs/10.1126/science.adi8474` (2023).

53. Pai, S. *et al.* Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science* **380,** 398–404. `https://www.science.org/doi/abs/10.1126/science.ade8450` (2023).

54. Laborieux, A. *et al.* Scaling Equilibrium Propagation to Deep ConvNets by Drastically Reducing Its Gradient Estimator Bias. *Frontiers in Neuroscience* **15.** ISSN: 1662-453X. `http://dx.doi.org/10.3389/fnins.2021.633674` (Feb. 2021).

55. Scellier, B. & Bengio, Y. Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. *Frontiers in Computational Neuroscience* **11.** ISSN: 1662-5188. `http://dx.doi.org/10.3389/fncom.2017.00024` (May 2017).

56. López-Pastor, V. & Marquardt, F. Self-Learning Machines Based on Hamiltonian Echo Backpropagation. *Physical Review X* **13.** ISSN: 2160-3308. http://dx.doi.org/10.1103/PhysRevX.13.031020 (Aug. 2023).

57. Hermans, M., Burm, M., Van Vaerenbergh, T., Dambre, J. & Bienstman, P. Trainable hardware for dynamical computing using error backpropagation through physical media. *Nature Communications* **6.** ISSN: 2041-1723. http://dx.doi.org/10.1038/ncomms7729 (Mar. 2015).

58. Dillavou, S., Stern, M., Liu, A. J. & Durian, D. J. Demonstration of Decentralized Physics-Driven Learning. *Physical Review Applied* **18.** ISSN: 2331-7019. http://dx.doi.org/10.1103/PhysRevApplied.18.014040 (July 2022).

59. Martin, E. *et al.* EqSpike: Spike-driven equilibrium propagation for neuromorphic implementations. *iScience* **24,** 102222. ISSN: 2589-0042. http://dx.doi.org/10.1016/j.isci.2021.102222 (Mar. 2021).

60. Ernoult, M., Grollier, J., Querlioz, D., Bengio, Y. & Scellier, B. Equilibrium propagation with continual weight updates. *arXiv preprint arXiv:2005.04168* (2020).

61. Brady, D. J. & Psaltis, D. Holographic interconnections in photorefractive waveguides. *Applied Optics* **30,** 2324. ISSN: 1539-4522. http://dx.doi.org/10.1364/AO.30.002324 (June 1991).

62. Shastri, B. J. *et al.* Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics* **15,** 102–114 (2021).

63. Prucnal, P. R., Shastri, B. J. & Teich, M. C. *Neuromorphic Photonics* (eds Prucnal, P. R. & Shastri, B. J.) ISBN: 9781315370590. http://dx.doi.org/10.1201/9781315370590 (CRC Press, May 2017).

64. Al-Qadasi, M. A., Chrostowski, L., Shastri, B. J. & Shekhar, S. Scaling up silicon photonic-based accelerators: Challenges and opportunities. *APL Photonics* **7.** ISSN: 2378-0967. http://dx.doi.org/10.1063/5.0070992 (Feb. 2022).

65. Capmany, J. & Pérez, D. in *Programmable Integrated Photonics* 1–37 (Oxford University Press, Mar. 2020).

66. Peserico, N., Shastri, B. J. & Sorger, V. J. Integrated Photonic Tensor Processing Unit for a Matrix Multiply: A Review. *Journal of Lightwave Technology* **41,** 3704–3716. ISSN: 1558-2213. http://dx.doi.org/10.1109/JLT.2023.3269957 (June 2023).

67. Farmakidis, N., Dong, B. & Bhaskaran, H. Integrated photonic neuromorphic computing: opportunities and challenges. *Nature Reviews Electrical Engineering* **1,** 358–373. ISSN: 2948-1201. http://dx.doi.org/10.1038/s44287-024-00050-9 (June 2024).

68. McMahon, P. L. The physics of optical computing. *Nature Reviews Physics* **5,** 717–734. ISSN: 2522-5820. http://dx.doi.org/10.1038/s42254-023-00645-5 (Oct. 2023).

69. Wang, C. *et al.* Integrated lithium niobate electro-optic modulators operating at CMOS-compatible voltages. *Nature* **562,** 101–104. ISSN: 1476-4687. http://dx.doi.org/10.1038/s41586-018-0551-y (Sept. 2018).

70. Kharel, P., Reimer, C., Luke, K., He, L. & Zhang, M. Breaking voltage–bandwidth limits in integrated lithium niobate modulators using micro-structured electrodes. *Optica* **8,** 357. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.416155 (Mar. 2021).

71. Marpaung, D., Yao, J. & Capmany, J. Integrated microwave photonics. *Nature Photonics* **13,** 80–90. ISSN: 1749-4893. http://dx.doi.org/10.1038/s41566-018-0310-5 (Jan. 2019).

72. Miller, D. A. B. Attojoule Optoelectronics for Low-Energy Information Processing and Communications. *Journal of Lightwave Technology* **35,** 346–396. ISSN: 1558-2213. http://dx.doi.org/10.1109/JLT.2017.2647779 (Feb. 2017).

73. Desiatov, B., Shams-Ansari, A., Zhang, M., Wang, C. & Lončar, M. Ultra-low-loss integrated visible photonics using thin-film lithium niobate. *Optica* **6,** 380. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.6.000380 (Mar. 2019).

74. Shekhar, S. *et al.* Roadmapping the next generation of silicon photonics. *Nature Communications* **15.** ISSN: 2041-1723. http://dx.doi.org/10.1038/s41467-024-44750-0 (Jan. 2024).

75. Li, H.-Y. S., Qiao, Y. & Psaltis, D. Optical network for real-time face recognition. *Applied Optics* **32,** 5026. ISSN: 1539-4522. http://dx.doi.org/10.1364/AO.32.005026 (Sept. 1993).

76. Lin, X. *et al.* All-optical machine learning using diffractive deep neural networks. *Science.* ISSN: 10959203 (2018).

77. Bandyopadhyay, S. *et al.* Single chip photonic deep neural network with accelerated training. *arXiv:2208.01623* (2022).

78. Feldmann, J. *et al.* Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589,** 52–58 (2021).

79. Tait, A. N. *et al.* Silicon Photonic Modulator Neuron. *Physical Review Applied* **11.** ISSN: 2331-7019. http://dx.doi.org/10.1103/PhysRevApplied.11.064043 (June 2019).

80. Zhu, H. *et al.* Space-efficient optical computing with an integrated chip diffractive neural network. *Nature communications* **13,** 1044 (2022).

81. Nikkhah, V. *et al.* Inverse-designed low-index-contrast structures on a silicon photonics platform for vector–matrix multiplication. *Nature Photonics.* https://doi.org/10.1038/s41566-024-01394-2 (2024).

82. Fyrillas, A., Faure, O., Maring, N., Senellart, J. & Belabas, N. Scalable machine learning-assisted clear-box characterization for optimally controlled photonic circuits. *arXiv:2310.15349* (2023).

83. Zhang, H. *et al.* An optical neural chip for implementing complex-valued neural network. *Nature Communications* **12,** 457. https://doi.org/10.1038/s41467-020-20719-7 (2021).

84. Tait, A. N. *et al.* Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports* **7,** 7430. ISSN: 2045-2322. http://dx.doi.org/10.1038/s41598-017-07754-z (Aug. 2017).

85. Carolan, J. *et al.* Universal linear optics. *Science* **349,** 711–716. ISSN: 1095-9203. http://dx.doi.org/10.1126/science.aab3642 (Aug. 2015).

86. Harris, N. C. *et al.* Quantum transport simulations in a programmable nanophotonic processor. *Nature Photonics* **11,** 447–452. ISSN: 1749-4893. http://dx.doi.org/10.1038/nphoton.2017.95 (June 2017).

87. Reck, M., Zeilinger, A., Bernstein, H. J. & Bertani, P. Experimental realization of any discrete unitary operator. *Physical Review Letters* **73,** 58–61. ISSN: 0031-9007. http://dx.doi.org/10.1103/PhysRevLett.73.58 (July 1994).

88. Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11,** 441–446 (2017).

89. Zhou, L., Sun, X., Li, X. & Chen, J. Miniature Microring Resonator Sensor Based on a Hybrid Plasmonic Waveguide. *Sensors* **11,** 6856–6867. ISSN: 1424-8220. http://dx.doi.org/10.3390/s110706856 (July 2011).

90. Xu, T. *et al.* Control-free and efficient silicon photonic neural networks via hardware-aware training and pruning. *arXiv preprint arXiv:2401.08180* (2024).

91. Ashtiani, F., Geers, A. J. & Aflatouni, F. An on-chip photonic deep neural network for image classification. *Nature* **606,** 501–506 (2022).

92. Fu, T. *et al.* Photonic machine learning with on-chip diffractive optics. *Nature Communications* **14.** ISSN: 2041-1723. http://dx.doi.org/10.1038/s41467-022-35772-7 (Jan. 2023).

93. Nakajima, M., Tanaka, K. & Hashimoto, T. Neural Schrödinger Equation: Physical Law as Deep Neural Network. *IEEE Transactions on Neural Networks and Learning Systems* **33,** 2686–2700. ISSN: 2162-2388. http://dx.doi.org/10.1109/TNNLS.2021.3120472 (June 2022).

94. Khoram, E. *et al.* Nanophotonic media for artificial neural inference. *Photonics Research* **7,** 823–827 (2019).

95. Wu, T., Menarini, M., Gao, Z. & Feng, L. Lithography-free reconfigurable integrated photonic processor. *Nature Photonics* **17,** 710–716. ISSN: 1749-4893. http://dx.doi.org/10.1038/s41566-023-01205-0 (Apr. 2023).

96. Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M. & Englund, D. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X* **9,** 021032 (2019).

97. Rahimi Kari, S., Nobile, N. A., Pantin, D., Shah, V. & Youngblood, N. Realization of an integrated coherent photonic platform for scalable matrix operations. *Optica* **11,** 542. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.507525 (Apr. 2024).

98. Dong, B. *et al.* Higher-dimensional processing using a photonic tensor core with continuous-time data. *Nature Photonics* **17,** 1080–1088. https://doi.org/10.1038/s41566-023-01313-x (2023).

99. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444. https://doi.org/10.1038/nature14539 (2015).

100. Wetzstein, G. *et al.* Inference in artificial intelligence with deep optics and photonics. *Nature* **588,** 39–47 (2020).

101. Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. & Pernice, W. H. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569,** 208–214 (2019).

102. Huang, C. *et al.* A silicon photonic–electronic neural network for fibre nonlinearity compensation. *Nature Electronics* **4,** 837–844 (2021).

103. Nahmias, M. A. *et al.* Photonic multiply-accumulate operations for neural networks. *IEEE Journal of Selected Topics in Quantum Electronics* **26,** 1–18 (2019).

104. Anderson, M. G., Ma, S.-Y., Wang, T., Wright, L. G. & McMahon, P. L. Optical Transformers. *arXiv:2302.10360* (2023).

105. Larocque, H. & Englund, D. Universal linear optics by programmable multimode interference. *Optics Express* **29,** 38257–38267 (2021).

106. Hughes, T. W., Williamson, I. A. D., Minkov, M. & Fan, S. Wave physics as an analog recurrent neural network. *Science Advances* **5,** eaay6946. ISSN: 2375-2548. http://dx.doi.org/10.1126/sciadv.aay6946 (Dec. 2019).

107. Molesky, S. *et al.* Inverse design in nanophotonics. *Nature Photonics* **12,** 659–670. ISSN: 1749-4893. http://dx.doi.org/10.1038/s41566-018-0246-9 (Oct. 2018).

108. Psaltis, D., Brady, D., Gu, X.-G. & Lin, S. Holography in artificial neural networks. *Nature* **343,** 325–330. ISSN: 1476-4687. http://dx.doi.org/10.1038/343325a0 (Jan. 1990).

109. Delaney, M. *et al.* Nonvolatile programmable silicon photonics using an ultralow-loss Sb_2Se_3 phase change material. *Science Advances* **7,** eabg3500. ISSN: 2375-2548. http://dx.doi.org/10.1126/sciadv.abg3500 (June 2021).

110. Wu, C. *et al.* Freeform direct-write and rewritable photonic integrated circuits in phase-change thin films. *Science Advances* **10,** eadk1361. ISSN: 2375-2548. http://dx.doi.org/10.1126/sciadv.adk1361 (Jan. 2024).

111. Delaney, M., Zeimpekis, I., Lawson, D., Hewak, D. W. & Muskens, O. L. A New Family of Ultralow Loss Reversible Phase-Change Materials for Photonic Integrated Circuits: Sb_2S_3 and Sb_2Se_3. *Advanced Functional Materials* **30,** 2002447. ISSN: 1616-3028. http://dx.doi.org/10.1002/adfm.202002447 (July 2020).

112. Chiou, P. Y., Ohta, A. T. & Wu, M. C. Massively parallel manipulation of single cells and microparticles using optical images. *Nature* **436,** 370–372 (2005).

113. Wu, M. C. Optoelectronic tweezers. *Nature Photonics* **5,** 322–324. ISSN: 1749-4893. http://dx.doi.org/10.1038/nphoton.2011.98 (May 2011).

114. Luennemann, M., Hartwig, U., Panotopoulos, G. & Buse, K. Electrooptic properties of lithium niobate crystals for extremely high external electric fields. *Applied Physics B* **76,** 403–406 (2003).

115. LeCun, Y. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/* (1998).

116. Wright, L. G. *et al.* Deep physical neural networks trained with backpropagation. *Nature* **601,** 549–555 (2022).

117. Gu, J. *et al.* M3ICRO: Machine Learning-Enabled Compact Photonic Tensor Core based on PRogrammable Multi-Operand Multimode Interference. *arXiv:2305.19505* (2023).

118. Huang, C. *et al.* Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits. *APL Photonics* **5** (2020).

119. Marcucci, G., Pierangeli, D. & Conti, C. Theory of Neuromorphic Computing by Waves: Machine Learning by Rogue Waves, Dispersive Shocks, and Solitons. *Physical Review Letters* **125.** ISSN: 1079-7114. http://dx.doi.org/10.1103/PhysRevLett.125.093901 (Aug. 2020).

120. Teğin, U., Yıldırım, M., Oğuz, İ., Moser, C. & Psaltis, D. Scalable optical learning operator. *Nature Computational Science* **1,** 542–549. ISSN: 2662-8457. http://dx.doi.org/10.1038/s43588-021-00112-0 (Aug. 2021).

121. Mohammadi Estakhri, N., Edwards, B. & Engheta, N. Inverse-designed metastructures that solve equations. *Science* **363,** 1333–1338 (2019).

122. Roques-Carmes, C. *et al.* Heuristic recurrent algorithms for photonic Ising machines. *Nature Communications* **11,** 249. ISSN: 2041-1723. http://dx.doi.org/10.1038/s41467-019-14096-z (Jan. 2020).

123. Bogaerts, W. *et al.* Programmable photonic circuits. *Nature* **586,** 207–216 (2020).

124. Xie, Y. *et al.* Programmable optical processor chips: toward photonic RF filters with DSP-level flexibility and MHz-band selectivity. *Nanophotonics* **7,** 421–454 (2017).

125. Feng, H. *et al.* Integrated lithium niobate microwave photonic processing engine. *Nature* **627,** 80–87. https://doi.org/10.1038/s41586-024-07078-9 (2024).

126. Cheng, Q., Rumley, S., Bahadori, M. & Bergman, K. Photonic switching in high performance datacenters. *Optics Express* **26,** 16022–16043 (2018).

127. Bender, N. *et al.* Depth-targeted energy delivery deep inside scattering media. *Nature Physics* **18,** 309–315 (2022).

128. Frumker, E. & Silberberg, Y. Phase and amplitude pulse shaping with two-dimensional phase-only spatial light modulators. *JOSA B* **24,** 2940–2947 (2007).

129. Psaltis, D., Brady, D. & Wagner, K. Adaptive optical networks using photorefractive crystals. *Applied Optics* **27,** 1752–1759 (1988).

130. Piccirillo, A. & Gobbi, A. L. Physical-Electrical Properties of Silicon Nitride Deposited by PECVD on III–V Semiconductors. *Journal of The Electrochemical Society* **137,** 3910–3917. ISSN: 1945-7111. http://dx.doi.org/10.1149/1.2086326 (Dec. 1990).

131. Janotta, A. *et al.* Doping and its efficiency in $a - \mathrm{SiO}_x : \mathrm{H}$. *Phys. Rev. B* **69,** 115206. https://link.aps.org/doi/10.1103/PhysRevB.69.115206 (11 Mar. 2004).

132. Piccoli, G., Sanna, M., Borghi, M., Pavesi, L. & Ghulinyan, M. Silicon oxynitride platform for linear and nonlinear photonics at NIR wavelengths. *Optical Materials Express* **12,** 3551. ISSN: 2159-3930. http://dx.doi.org/10.1364/OME.463940 (Aug. 2022).

133. Ghatak, A., Thyagarajan, K. & Shenoy, M. Numerical analysis of planar optical waveguides using matrix approach. *Journal of Lightwave Technology* **5,** 660–667. ISSN: 0733-8724. http://dx.doi.org/10.1109/JLT.1987.1075553 (1987).

134. Zhang, M., Wang, C., Cheng, R., Shams-Ansari, A. & Lončar, M. Monolithic ultra-high-Q lithium niobate microring resonator. *Optica* **4,** 1536. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.4.001536 (Dec. 2017).

135. Spall, J., Guo, X. & Lvovsky, A. I. Training neural networks with end-to-end optical backpropagation. *arxiv:2308.05226.* https://arxiv.org/abs/2308.05226 (2023).

136. Iluz, M. *et al.* Unveiling the evolution of light within photonic integrated circuits. *Optica* **11,** 42–47. https://opg.optica.org/optica/abstract.cfm?URI=optica-11-1-42 (Jan. 2024).

137. Agrawal, G. in (Academic Press, Boston, 2012).

138. Schwesyg, J. R. *et al.* Pyroelectrically induced photorefractive damage in magnesium-doped lithium niobate crystals. *J. Opt. Soc. Am. B* **28,** 1973–1987. https://opg.optica.org/josab/abstract.cfm?URI=josab-28-8-1973 (Aug. 2011).

139. Paturzo, M. *et al.* On the origin of internal field in Lithium Niobate crystals directly observed by digital holography. *Optics Express* **13,** 5416. ISSN: 1094-4087. http://dx.doi.org/10.1364/OPEX.13.005416 (July 2005).

140. Ramey, C. *Silicon Photonics for Artificial Intelligence Acceleration* in *2020 IEEE Hot Chips 32 Symposium (HCS)* (2020), 1–26.

141. Li, G. H. *et al.* All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics* **12,** 847–855. ISSN: 2192-8614. http://dx.doi.org/10.1515/nanoph-2022-0137 (May 2022).

142. Cui, C., Zhang, L. & Fan, L. In situ control of effective Kerr nonlinearity with Pockels integrated photonics. *Nature Physics* **18,** 497–501. https://doi.org/10.1038/s41567-022-01542-x (2022).

143. Tang, P., Meier, A. L., Towner, D. J. & Wessels, B. W. $BaTiO_3$ thin-film waveguide modulator with a low voltage–length product at near-infrared wavelengths of 098 and 155 µm. *Optics Letters* **30,** 254. ISSN: 1539-4794. http://dx.doi.org/10.1364/OL.30.000254 (Feb. 2005).

144. Koeber, S. *et al.* Femtojoule electro-optic modulation using a silicon–organic hybrid device. *Light: Science & Applications* **4,** e255. ISSN: 2047-7538. http://dx.doi.org/10.1038/lsa.2015.28 (Feb. 2015).

145. Davis, S. R., Farca, G., Rommel, S. D., Johnson, S. & Anderson, M. H. *Liquid crystal waveguides: new devices enabled by >1000 waves of optical phase control* in *Emerging Liquid Crystal Technologies V* **7618** (SPIE, 2010), 76180E.

146. Leuthold, J., Koos, C. & Freude, W. Nonlinear silicon photonics. *Nature Photonics* **4,** 535–544. https://doi.org/10.1038/nphoton.2010.185 (2010).

147. Blumenthal, D. J., Heideman, R., Geuzebroek, D., Leinse, A. & Roeloffzen, C. Silicon Nitride in Silicon Photonics. *Proceedings of the IEEE* **106,** 2209–2231 (2018).

148. Jung, H. *et al.* Tantala Kerr nonlinear integrated photonics. *Optica* **8,** 811. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.411968 (May 2021).

149. Timurdogan, E., Poulton, C. V., Byrd, M. J. & Watts, M. R. Electric field-induced second-order nonlinear optical effects in silicon waveguides. *Nature Photonics* **11,** 200–206. ISSN: 1749-4893. http://dx.doi.org/10.1038/nphoton.2017.14 (Feb. 2017).

150. Zhang, X. *et al.* Heterogeneously integrated III–V-on-lithium niobate broadband light sources and photodetectors. *Optics Letters* **47,** 4564. ISSN: 1539-4794. http://dx.doi.org/10.1364/OL.468008 (Aug. 2022).

151. Zhu, S. *et al.* Waveguide-Integrated Two-Dimensional Material Photodetectors in Thin-Film Lithium Niobate. *Advanced Photonics Research* **4,** 2300045. ISSN: 2699-9293. http://dx.doi.org/10.1002/adpr.202300045 (Apr. 2023).

152. Ahn, G. H. *et al.* Platform-agnostic waveguide integration of high-speed photodetectors with evaporated tellurium thin films. *Optica* **10,** 349. ISSN: 2334-2536. http://dx.doi.org/10.1364/OPTICA.475387 (Mar. 2023).

153. Liu, Z. *et al.* Micro-light-emitting diodes with quantum dots in display technology. *Light: Science & Applications* **9,** 83. ISSN: 2047-7538. http://dx.doi.org/10.1038/s41377-020-0268-1 (May 2020).

154. Choi, H. W. *et al.* GaN micro-light-emitting diode arrays with monolithically integrated sapphire microlenses. *Applied Physics Letters* **84,** 2253–2255. ISSN: 1077-3118. http://dx.doi.org/10.1063/1.1690876 (Mar. 2004).

155. Blasl, M. *Elektrooptisch induzierte Wellenleiter in paranematischen Flüssigkristallen* doctoralthesis (BTU Cottbus - Senftenberg, 2017).

156. Brinkmann, N., Sommer, D., Micard, G., Hahn, G. & Terheiden, B. Electrical, optical and structural investigation of plasma-enhanced chemical-vapor-deposited amorphous silicon oxynitride films for solar cell applications. *Solar Energy Materials and Solar Cells* **108,** 180–188. ISSN: 0927-0248. http://dx.doi.org/10.1016/j.solmat.2012.09.025 (Jan. 2013).

157. Yang, K. Y. *et al.* Multi-dimensional data transmission using inverse-designed silicon photonics and microcombs. *Nature Communications* **13,** 7862 (2022).

158. Miller, D. A. B. All linear optical devices are mode converters. *Optics Express* **20,** 23985. ISSN: 1094-4087. http://dx.doi.org/10.1364/OE.20.023985 (Oct. 2012).

CHAPTER 7

OBSERVATION OF QUESTIONABLE RESEARCH PRACTICES IN INTRO PHYSICS LABS

The chapter is a reprint of Stein, M. M. *et al. Confirming what we know: Understanding questionable research practices in intro physics labs* in *2018 Physics Education Research Conference Proceedings* (American Association of Physics Teachers, 2019).

A more complete analysis, triangulating the observations we made in lab notes with video analysis and interviews of students is published in Smith, E. M. *et al.* How expectations of confirmation influence students' experimentation decisions in introductory labs. *Physical Review Physics Education Research* **16** (2020). This work was led and written up by Emily M. Smith and therefore cannot be reprinted in this thesis. I recommend readers to supplement the study of the paper reprinted here with Dr. Smith's paper.

7.1 Introduction

Introductory physics labs are often used to verify the physics content presented in a course. These traditional, often highly structured, labs have been under heavy scrutiny. Studies have found that these labs do not provide measurable added value to learning the physics content [3] and deteriorate students' perceptions of experimental physics [4].

Rather than using labs to verify physics content, there are calls to shift the focus of labs to teach students about the nature of science and to develop students' scientific abilities. The American Association of Physics Teachers endorsed learning goals for labs that focus on a variety of experimentation skills and abilities [5]. Encouragingly, studies investigating lab curricula centered around these goals, broadly referred to here as experimentation-focused labs, have found that students' abilities and engagement can develop over semester-long courses [6, 7]. However, little is known about the struggles that students encounter in

experimentation-focused labs.

As labs shift away from verifying physics content, students may struggle to understand their role in the lab. In experimentation-focused labs, the focus is on the process, not the product, of the investigation. The intent is not for students to achieve a particular outcome; the intent is for students to work scientifically with data and models and to draw conclusions based on their evidence.

These instructional intentions may challenge students' beliefs about labs and the role of experiments in science. For example, many introductory students believe that the purpose of a lab is to supplement their learning of lecture content [8, 9]. Students also tend to believe that the purpose of experiments in physics labs is to confirm previously known results [10] and that experimental results should be evaluated on their agreement with theory or confirmation of previous results [11]. Previous research has found that some students interpret or manipulate data in ways that unjustifiably verify or confirm particular results. For example, students were found to exhibit difficulties coordinating claims and evidence from complex data sets, such as making claims based on prior knowledge rather than data [12]. Students have also been found to inflate the values of experimental uncertainty to hide systematic errors that cause disagreement between data and theory [13].

We hypothesize that students' beliefs that labs are meant to confirm known results may prevent them from authentically engaging in the scientific process in experimentation-focused labs. An elucidating example supporting this hypothesis is presented in a companion paper [14]. In this paper, we study how students' goals to confirm known results lead them to engage in inauthentic and questionable research practices. Questionable research practices are defined as "actions that violate traditional values of the research enterprise and that may be detrimental to the research process" without being outright misconduct of research [15]. These practices are situated in an ethical grey area and may be acceptable practices within specific fields or under certain circumstances. However, in the context of experimentation-focused labs, some questionable research practices may be detrimental to students' authentic

Table 7.1: Coding scheme used to classify different types of questionable research practices.

Category	Questionable Research practices	Description
Subjective Interpretation	*Concerning results*	The distinguishability of the datasets is described as a concern or issue.
	Emotional response to data	Statements refer to students liking or disliking the results.
	Qualitative judgment of results	The distinguishability of the data or quality of the methods are judged qualitatively (e.g. good, bad, too small, too large, helpful, or an improvement) based on the results.
Unjustified Interpretation	*Claim of accuracy*	The accuracy of data, the method, or an instrument used to take data, are judged based on the test statistic value.
	Claim of systematic error	The distinguishability of the data sets is explained based on the presence or absence of systematic error, without describing the source.
	Doubting statistics	The validity of statistical tools, like the test statistic or standard deviation/error is questioned without justification.
Purpose	*(Dis-) prove model*	The purpose of the lab or intent of the group is explicitly or implicitly stated as to show that the model holds or breaks down.
Data manipulation	*Inflating uncertainty*	Statements demonstrate that students attempted to inflate their uncertainty, either through experimental decisions or manipulation of data.

engagement in experimentation processes and run counter to instructional intention. This study is part of a larger study that aims to evaluate how students transition to labs with no instructional intent to verify equations and where students have control over the outcome of their experiment.

7.2 Methods

Our participants were students enrolled in the first-semester course of a calculus-based physics sequence at three different institutions. The institutions included two research universities and one community college. Data were collected from one semester at institutions A and B, but two separate semesters at institution C.

All four implementations used the same activity during the first lab. This activity uses the Structured Quantitative Inquiry Labs (SQILabs) format [7]. In this activity, students are explicitly instructed to make comparisons between data (or data and a model), interpret those comparisons, and make decisions about how to follow-up on their investigation, with much emphasis on iterating to improve measurements.

Students typically worked in groups of two to four to conduct the investigations. Our data were the groups' written lab notes. They were instructed to record their process in a lab notebook and to treat the notebook like a journal, with extensive notes about what they were doing and why they were doing it. Students submitted the lab notebook by the end of the lab period for grading. At institution A and B, students had three hours to complete the lab. At institution C students had two hours to complete the lab. At institutions A and C students submitted one lab notebook per group, while students at Institution B submitted lab notes individually but only one student's notes were graded. Therefore, at all institutions, students received one grade for the whole group. At institutions A and B, students' notebooks were on paper, while at institution C, students used electronic lab notebooks.

Our data all stem from the first experiment of the lab course that asks students to test whether the period of a pendulum is dependent on its initial amplitude. Students were given the simplified formula for the period of a pendulum, $T = 2\pi\sqrt{\ell/g}$, which predicts that the period is independent of the amplitude—valid under a small angle approximation. The approximation was not explicitly taught in the lab, though some students were aware of the equation's approximation.

During the lab, students were also introduced to a statistical test to distinguish the mean of two datasets within their uncertainties. They were given an interpretation of the test statistic with three levels: a value below 1 indicates statistically indistinguishable datasets, above 3 means distinguishable, and in between, the result is unclear [16].

Regardless of the result of the test statistic, students were encouraged to improve the quality of their measurements by iterating and extending the experiment. During the assigned lab time, many groups found statistically distinguishable datasets for the period at two different amplitudes (37%, n=40), indicating they reached the experimental precision to measure the breakdown of the model.

7.2.1 Development of the coding scheme

We used emergent coding of the lab notes (summarized in Table 7.1) to identify questionable research practices. Two raters coded 15% of the notebooks. The selection included a greater proportion of questionable research practices than the entire dataset. Prior to discussion, raters agreed on average on 77% of codes in each notebook. After simple wording changes and discussion, raters reached full agreement and a single rater coded the remaining lab notes. Each coded questionable research practice was further divided as to whether or not it was oriented towards confirming the model.

We also analyzed lab notes for groups' final result of the test statistic and the conclusions they drew from it. Our coding here matched the interpretation of the test statistic students were given: A test statistic smaller than 1 was coded as "Test statistic supports model", above 3 was coded as "Test statistic contradicts model" and in between as "unclear". The final conclusions in lab notes were coded as "accept model" if groups wrote the period of the pendulum is independent of the amplitude, "reject model" if they concluded there is a difference between the periods at different amplitudes, and "unclear" if it was explicitly written that the results were inconclusive or no definite conclusion was recorded. We used

these codes to identify what fraction of groups drew conclusions that agreed or disagreed with their data, and whether groups' confirmatory goals affected the conclusions they drew.

We performed an ordinal logistic regression to compare the groups with confirmatory questionable research practices and the groups with no coded questionable research practices. Deviations from the given interpretation of the test statistic were used as the ordinal response. Finding a test statistic that contradicts the model but accepting it in the conclusion was assigned the numerical value -2 and finding data that supports the model but rejecting the model in the conclusion was assigned 2. Drawing a conclusion that followed the given interpretation of the test statistic was assigned 0, starting from unclear data or arriving at an unclear conclusion was assigned ± 1, depending on the orientation.

7.3 Results

Questionable research practices were found in 30% ($n = 32$) of the groups' lab notes. However, as shown in Fig. 7.1, the fraction of lab notes indicating these practices varied significantly across implementations. We suspect, due to the significant differences in results between the two implementations at Institution C (Fisher's exact test: $p = 0.013$), that variations in reported questionable practices are due to instructional decisions rather than student populations. Instructors in Implementation 2 were aware that many students express confirmatory goals during the lab and were given strategies to mitigate questionable research practices.

Some coded practices are noted much more frequently than others (Fig. 7.2). Most commonly, groups noted practices that indicated subjective (12% of all groups, n=13) or unjustified (20% of all groups, n=21) interpretations of their data. Most lab notes coded as unjustified interpretations made claims that their measurements were affected by a systematic error based on the test statistic. For example, one group wrote, "*If we could reduce our error [...] our [test statistic] should have been much lower.*" The next most common code in

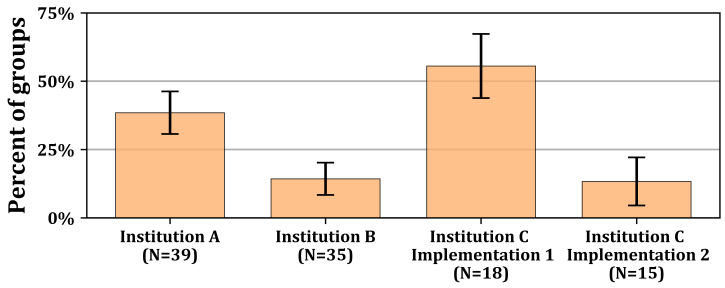

Figure 7.1: Percent of groups that exhibited at least one questionable research practice across institutions. Error bars represent standard errors on the proportions.

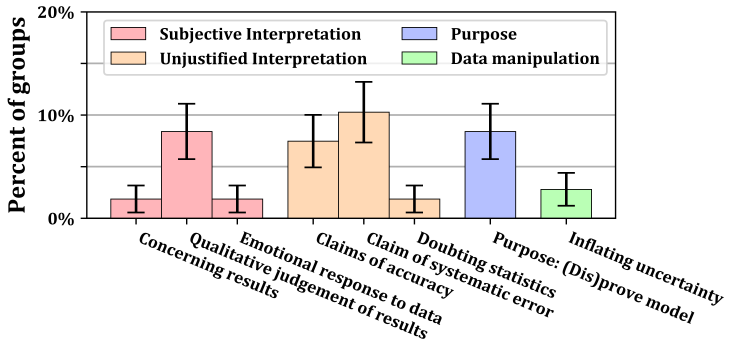

Figure 7.2: Percent of groups that exhibited each questionable research practice. Error bars represent standard errors on the proportions.

this category was from claims that the data or methods were accurate or inaccurate based on the agreement between data and the model. For example, one group wrote, "*This was a very crude experiment which resulted in a high [test statistic] which means our experiment was not very accurate.*" Another group doubted that the statistical tools they used were representative of their data: "*Our [statistical] test is not representative of our data, this is because the standard error value is so large, giving us a lower [test statistic] value.*"

Many groups that recorded subjective interpretations of their results claimed the (dis-)agreement between data and the model was "*good*", "*improved*" or "*needs further help*" (coded as *Qualitative judgment of results*). Others wrote the mismatch between their data and the model was a "*concern*". The most surprising code was that some groups noted an emotional response to data that disagreed with the model. One group indicated they will change the experiment "*if there is any unsatisfaction*" with their results, another group de-

cided to increase their standard errors by an order of magnitude, because they "*liked the low [values of the test statistic]*" and how indistinguishable the periods of the pendulum were with the larger uncertainties.

In fact, a few groups (3%, n=3) actively manipulated their data by inflating experimental uncertainties to obtain results that agreed with the model. This primarily involved groups designing follow-up experiments that deliberately increased their uncertainty to make the two datasets less distinguishable. Finally, about 8% (n=9) of the groups stated that the purpose of the lab was to confirm or disconfirm the model or expressed their intent to do so.

Most (70%) of the groups exhibiting questionable research practices, however, did so with a confirmatory goal. For example, one group stated an aim to make the periods more similar. This was coded with the questionable research practice *(Dis-) prove model* and associated with confirming the model. Another group stated that their large value of the test statistic was a concern, which was coded with the questionable research practice *Concerning results* and associated with confirming the model.

An additional 20% of the groups recording questionable research practices made statements that indicated they were trying to confirm the model, but also made statements that indicated the opposite or were unclear.

This implies that 90% of the groups exhibiting questionable research practices conveyed confirmatory goals at some point. Three groups indicated an aim to disconfirm the model. Because this aim involves a substantially different understanding of models and scientific exploration, these groups were not coded as aiming to confirm.

In Fig. 7.3 we compare groups who exhibited confirmatory questionable research practices with those who did not exhibit questionable research practices. There were too few groups who exhibited non-confirmatory questionable research practices for meaningful comparison. Groups with confirmatory questionable research practices arrived at similar results as groups who did not exhibit questionable research practices, as measured by the values of their test

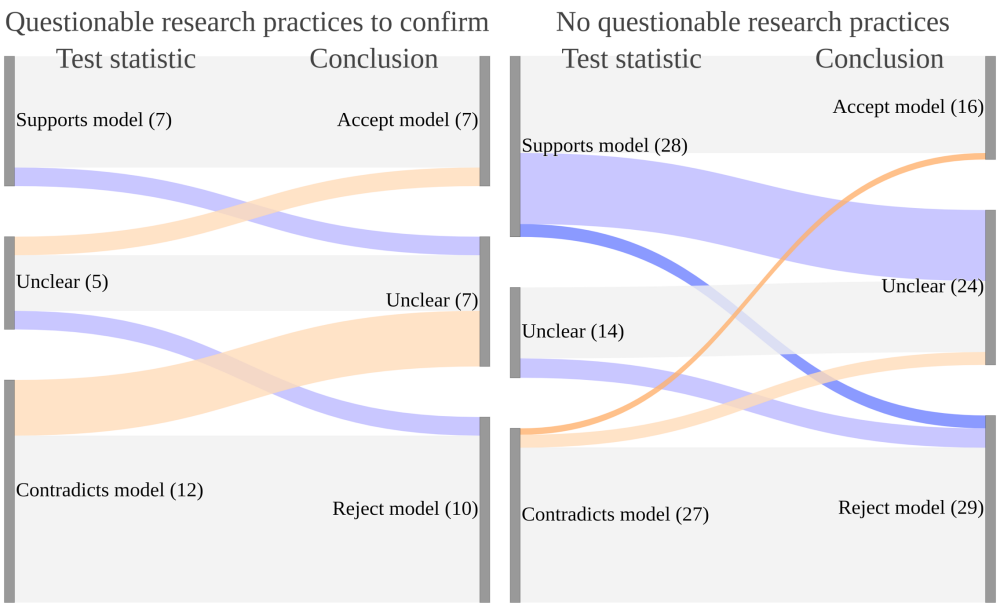

Figure 7.3: Results of test statistics (left nodes) and conclusions (right nodes) drawn by student groups using confirmatory questionable research practices vs. no questionable research practices. Orange lines correspond to conclusions that lean towards confirmation, blue lines towards disconfirmation, gray lines are directly supported by the test statistic obtained.

statistic (Mann Whitney U-Test, $p = 0.394$). The two groups also arrived at similar conclusions at the end of the lab. However, as visualized in Fig. 7.3, groups exhibiting confirmatory questionable research practices less frequently questioned the model if they found data supporting it and less frequently rejected the model if they found data contradicting it. This can be seen in the "upwards" trend in the left diagram and the "downwards" trend in the right diagram. The ordinal logistic regression found these differences statistically significant (Odds ratio: 3.76, 95% C.I.: $[1.21, 11.73]$, $p = 0.022$).

7.4 Discussion

In this study we found that students report questionable research practices in an experimentation-focused lab. Most of these practices were associated with students' interpretation of data and served to confirm the investigated model. Furthermore, students aiming to confirm the model less frequently questioned data that supported the model and

less frequently rejected the model in light of contradicting data.

Although we have only observed a correlation, we suspect that the intent to confirm the model was the cause of most questionable research practices. Research has found that the majority of introductory students believe that the validity of data should be evaluated based on its agreement with theory or the results from others [11] and that the purpose of experiments is to confirm theory [9]. We found that those confirmatory beliefs correlate with students' questionable research practices in the lab as many students motivated these practices with the intent to show agreement between data and the model. In the future, we aim to investigate whether these confirmatory goals come from extensive previous experience with verification labs.

Our data only include those practices that students recorded in their lab notes. It is likely that more students engaged in additional practices but did not report them. The percentages we report, therefore can be understood as lower bounds on their occurrence in the labs. The exhibited practices were similar across different institutions; however, we found their frequency varied significantly across different implementations. We attribute these variations to differences in instruction rather than student population, which could inform possible instructor interventions. Based on the results here, it is possible that clarifying the purpose of the lab to students, asking them to think critically about the limitations of simple models, or constantly strive to reduce uncertainty, could affect student practices. However, it is unclear whether the differences in instruction actually changes the frequency of questionable research practices or just how often they are reported in lab notes. We plan to use video observation to investigate these and other questions in the future. We present an analysis of video observations and interviews for how the presented lab activity can shift students framing of introductory physics labs away from model-verifying frames in [14].

CHAPTER 8

REFLECTION AND OUTLOOK

This chapter is a reflection on how introductory physics labs do and do not provide students authentic science experiences. I spent the first two years of graduate school performing research on measuring what students learn in introductory physics labs. I then spent the remaining five years performing research in applied & engineering physics. I was also a teaching assistant for multiple introductory physics lab classes and mentored many undergraduate students who conducted independent research projects in Peter McMahon's lab. While teaching labs, I had many conversations with students, and I will draw on anecdotal evidence from these conversations. My hope is that I can offer the rare perspective of someone who is deeply familiar with doing research in physics, research in physics education, and was "on the ground" teaching students.* I will attempt to offer an opinion informed by research, but this is inevitably my personal viewpoint at a particular point in time and needs to be taken with a grain of salt. Further, much of this applies to introductory labs as they were taught at Cornell University between Fall 2017 and Spring 2020 and might not be applicable at another time or in another place.

I will list a number of points for which I observed a particularly strong disparity between what students experience in introductory physics labs and the experience of an applied physicist.

Disparity 1: Students and researchers have a different relation to experimental equipment

While working as an applied physicist, I experienced that almost all lab equipment will undergo careful tests and calibrations before use, that lead to a high level of trust of physicists in their measurements. In contrast, my impression is that students in introductory physics labs often have little trust into their measurements and equipment.

*Although maybe rare, this background is of course far from unique and I had the pleasure of meeting multiple individuals with all these experiences and more throughout graduate school.

Evidence for this is beyond just anecdotal: In the previous chapter we saw that students often make claims about the (lack of) accuracy or vague systematic errors in their experiments. We attributed these to students‘ confirmatory beliefs, i.e. their expectations of having to confirm the theory presented to them. A closer look at Fig. 7.3 reveals another effect: In general, students tend to more often draw indeterminate conclusions than would be expected from their experimental results (only 20% of student groups measured an “unclear” test statistic, but 33% drew an “unclear” conclusion to their experiment. Two-proportion z-test: $p = 0.047$). Students appear to be hesitant to draw firm conclusions from their experimental observations. Anecdotal evidence suggests that part of this wariness is explained by distrust in experimental equipment and methods. Distrust in experimental equipment can in turn be explained by experiences of working with poor or malfunctioning equipment in the past.

Proposed solution: Encourage, facilitate, and grade students to build trust in equipment

The tests and calibrations physicists perform to build trust in their equipment are usually performed against what is considered to be very solid knowledge. For shared equipment, testing and calibration is often repeated every time someone else has used the equipment. I believe this component is not represented enough in introductory physics labs, even though it is one of the most important component of experimental and applied physics. It is plausible that more time in physics labs is spent on testing equipment than actually using equipment to measure new effects of interest. In *Representing and intervening*, Ian Hacking, a philosopher of science who studied many experimental physicists writes:

> *There is designing an experiment that might work. There is learning how to make the experiment work. But perhaps the real knack is getting to know when the experiment is working. [17]*

I suspect that less students would dismiss unexpected results and attribute them to poor equipment if they would spend more time to familiarize themselves with equipment. I believe

it could be a good idea to create lab units that are solely dedicated to calibrating and testing equipment. This can be a creative exercise in which students design testing procedures themselves.

This procedure could give students intellectual ownership about their equipment, which can be a source of motivation to rely on equipment more confidently when drawing conclusions. I believe a student that calibrated a piece of equipment themselves will be less likely to assign blame to the equipment supposedly malfunctioning. Such procedures are already implemented for statistical tools, e.g. Excel tables that students develop before class to build familiarity with those tools. Students are asked to compare computations from their Excel sheets against known correct answers. This builds confidence in their statistical tools. Why not emphasize it more for lab equipment such as force meters or ultrasonic rangers, etc.? (How accurate is the measured distance? How large does an object need to be to be detected?) The effectiveness of a lab unit devoted to building confidence with equipment could be tested with questions on the E-CLASS survey that relate to trust in lab equipment [18], or a dedicated survey (which does not yet exist to my knowledge).

Ideally, over time students would develop a sixth sense for when it is necessary to test equipment more, and when an experiment is truly measuring what it is supposed to measure. This is an ability that I observed many students do not naturally develop. Hacking describes this ability as follows:

> *Another kind of observation is what counts: the uncanny ability to pick out what is odd, wrong, instructive or distorted in the antics of one's equipment. [17]*

When supervising students in my own research, when they encountered an unexpected result, I often check or make the students check all the measurement equipment and the data analysis pipeline. Most often, the unexpected result is the consequence of malfunctions or mistakes. Yet, occasionally, the unexpected result is indeed due to an unexpected physical effect. In these moments, one needs to approach the observations with an open mind and be ready to

throw one's understanding overboard. But that is much easier when having full trust that the experiment is working.

Disparity 2: Experimental physicists know and use a lot of theory

Labs aiming to teach experimentation skills regularly ask students to compare between different models and make judgement calls as to which model describes a set of observations the best. The use of theory is otherwise (justifiably) not much encouraged, as it draws the attention of students away from the experimentation skills they are encouraged to develop. My experience is that experimental physicists know and use a lot of theory. This versatile use of theory in experiments elevates the quality of their experiments and is an integral part of it. Among other things:

- Theory can inform what experiments are interesting to do,
- An extensively verified theory can be used to ensure an experimental setup works as expected,
- Theory can be used to inform what quantities to measure, e.g. which observable is most sensitive to the parameter in question, etc.,
- Theory can inform experimental design, for example minimizing the effect of confounding variables (for a beautiful example, see Samuel Barnett's method of compensating for stray magnetic fields in the measurement of the gyromagnetic ratio, presented in Fig. 2.12 in Peter Gallison's "How Experiments End" [19]),
- Theory can give meaning to experiments that appear otherwise dull or rote (for example, my colleague Alen Senanian built a controllable laser cavity with the interesting twist that the theoretical description of the light in the cavity is the same as that of excitations in crystal lattices. The light can therefore simulate the behavior of crystals in different geometries and its observation takes on another level of meaning [20]).

In my experience, integrating theory and experiments is an integral part of applied physics that deeply permeates almost all experiments. Introductory physics labs could give students a more authentic science experience if students honed this skill early on.

Proposed solution: Content-*informed* labs

I believe it would be good to strengthen the connections between theoretical physics classes (which I will call "lecture classes" for simplicity) and the lab classes, while preserving the focus of labs on experimentation skills. For example, one could make physics classes a pre-requisite for the lab classes. A student who would want to take an electromagnetism lab would have to take an electromagnetism lecture class at least one semester before taking the lab class. This would give students a chance to digest theory before applying them in labs, similar to how experimental physicists usually have at least an overview of a theory before setting out to do experiments. The goal of the lab class would still be to learn experimentation skills, and not to reinforce the lecture content (although for example retention might get improved due to the repeated and spaced-out exposure to the same content). The benefit would, ideally, be that students can learn the value of informing their experiments by theory.

Discussion

There are other disparities that could be addressed. It is of course illusional that introductory physics labs can provide the same experience that an experimental physicist. There are limited resources (time, attention, funding) and the instruction needs to make reasonable trade-offs between what is desired and what is possible. But I believe there are two more disparities that deserve mention, even though a fix to these might be difficult:

1) Experimental physicists heavily rely on automated measurements. The use of computer code has become so ubiquitous in applied physics (in my experience) that it warrants explicit

instruction in related *experimental* skills. While there are many classes teaching scientific programming, most of these focus to my knowledge on computational skills (solving differential equations, optimization problems) rather than experimental skills. Creating lab units that teach students, say, how to read from a camera or oscilloscope in a programming language, or how to set the voltage of a function generator programmatically with an industry-standard API like VISA, would be very valuable and give students a more realistic experience of a typical data-acquisition process.

2) Being a physicist is more often than not an interesting job. One gets to use cutting-edge technology and often work on problems that no one knows the answer to. While many theoretical physicists do an amazing job conveying the "sexiness" of theoretical physics in introductory classes, introductory physics labs do not convey a similar appeal. This is perhaps unjustified. There are many very exciting and "cool" experimental techniques that are potentially accessible for beginning college students (think for example stroboscopic measurements or a lock-in amplifier). If experimentation labs decorrelate lecture contents from lab contents, then maybe physics labs do not have to start with the rather un-inspring experiments of classical mechanics (I impartially suggest optics instead).

These are the observations of an applied physicists that taught multiple introductory physics labs. From my stint in education research, I know how difficult it is to consolidate so many learning goals. The decision on whether it makes sense to address the mentioned disparities needs to be left to education researchers. What is undeniable is that much time is wasted preparing students individually for lab work. The burden of this preparation in an apprentice-fashion is often carried by senior graduate students or postdocs, who are not adequately prepared to train younger graduate students for lab work. Freeman el al [21] estimated the economic benefit if active learning were widely implemented. It would be interesting to estimate the economic benefit if lab classes prepared undergraduate students more adequately for lab work. This would underline the value of lab classes and the importance of research on them.

BIBLIOGRAPHY

1. Stein, M. M., Smith, E. M. & Holmes, N. G. *Confirming what we know: Understanding questionable research practices in intro physics labs* in *2018 Physics Education Research Conference Proceedings* (American Association of Physics Teachers, Jan. 2019). http://dx.doi.org/10.1119/perc.2018.pr.Stein.

2. Smith, E. M., Stein, M. M. & Holmes, N. How expectations of confirmation influence students' experimentation decisions in introductory labs. *Physical Review Physics Education Research* **16.** ISSN: 2469-9896. http://dx.doi.org/10.1103/PhysRevPhysEducRes.16.010113 (Mar. 2020).

3. Holmes, N. G. *et al.* Value added or misattributed? A multi-institution study on the educational benefit of labs for reinforcing physics content. *Phys. Rev. PER* **13,** 010129 (2017).

4. Wilcox, B. R. & Lewandowski, H. J. Developing skills versus reinforcing concepts in physics labs: Insight from a survey of students' beliefs about experimental physics. *Phys. Rev. PER* **13,** 1–9 (2017).

5. American Association of Physics Teachers. *AAPT Recommendations for the Undergraduate Physics Laboratory Curriculum* tech. rep. (2014), 29. https://www.aapt.org/Resources/upload/LabGuidlinesDocument_EBendorsed_nov10.pdf.

6. Etkina, E. *et al.* Design and reflection help students develop scientific abilities: Learning in introductory physics laboratories. *Journal of the Learning Sciences* **19,** 54–98 (2010).

7. Holmes, N. G., Wieman, C. E. & Bonn, D. A. Teaching critical thinking. *PNAS* **112,** 11199–11204 (2015).

8. Hu, D. & Zwickl, B. M. *Examining students' personal epistemology: the role of physics experiments and relation with theory* in *2017 PERC Proceedings* (2018), 11–14.

9. Hu, D. *et al.* Qualitative investigation of students' views about experimental physics. *Phys. Rev. PER* **13,** 020134 (2017).

10. Wilcox, B. R. & Lewandowski, H. J. Students' views about the nature of experimental physics. *Phys. Rev. Phys. Educ. Res.* **13,** 020110 (2 2017).

11. Hu, D. & Zwickl, B. M. Examining students' views about validity of experiments: From introductory to Ph.D. students. *Phys. Rev. PER* **14,** 010121 (2018).

12. Bogdan, A. M. & Heckler, A. F. Effects of Belief Bias on Student Reasoning from Data Tables. *2013 PERC Proceedings,* 73–76 (2014).

13. Holmes, N. G. & Bonn, D. A. Doing Science or Doing a Lab? Engaging Students with Scientific Reasoning during Physics Lab Experiments. *2013 PERC Proceedings,* 185–188 (2014).

14. Smith, E. M., Stein, M. M. & Holmes, N. G. *Surprise! Shifting students away from model-verifying frames in physics labs* in *2018 Physics Education Research Conference Proceedings* (American Association of Physics Teachers, 2019). `http://dx.doi.org/10.1119/perc.2018.pr.Smith`.

15. The Integrity of the Research Process: Volume I., R. S. E. *National Academies Press (US)* (2002).

16. Holmes, N. G. & Bonn, D. A. Quantitative Comparisons to Promote Inquiry in the Introductory Physics Lab. *Phys. Teach.* **53,** 352–355 (2015).

17. Hacking, I. *Representing and intervening* (Cambridge University Press, Cambridge, England, June 2012).

18. Zwickl, B. M., Finkelstein, N. & Lewandowski, H. J. *Development and validation of the Colorado learning attitudes about science survey for experimental physics* in *AIP Conference Proceedings* (AIP, 2013). `http://dx.doi.org/10.1063/1.4789747`.

19. Galison, P. *How Experiments End* 2nd ed. en (University of Chicago Press, Chicago, IL, Jan. 1988).

20. Senanian, A., Wright, L. G., Wade, P. F., Doyle, H. K. & McMahon, P. L. Programmable large-scale simulation of bosonic transport in optical synthetic frequency lattices. *Nature Physics* **19,** 1333–1339. ISSN: 1745-2481. http://dx.doi.org/10.1038/s41567-023-02075-7 (May 2023).

21. Freeman, S. *et al.* Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the national academy of sciences* **111,** 8410–8415 (2014).