Design Optimization of Photonic Integrated Circuits for Low-Power Nonlinear Optical Computing

*Haitao Zhou[1], Ilker Oguz[1, *], Niyazi Ulas Dinc[1], Mustafa Yildirim[1], Andreas Maeder[2], Rachel Grange[2], Demetri Psaltis[1], Christophe Moser[1]*

[1]EPFL, Laboratory of Applied Photonics Devices, Institute of Electrical and Micro Engineering, 1015 Lausanne, Switzerland

[2]ETH Zurich, Department of Physics, Institute for Quantum Electronics, Optical Nanomaterial Group, 8093 Zurich, Switzerland

**Correspondence: Ilker Oguz (ilker.oguz@epfl.ch)*

Abstract

Computing with light offers major advantages over traditional electronics, with potential for higher energy efficiency and bandwidth. Photonic integrated circuits (PICs) commonly use Mach-Zehnder Interferometer (MZI) meshes as the optical computing platform due to their high tunability and low loss. These meshes can dynamically direct and manipulate light signals to achieve complex optical transformations, including efficient and reconfigurable matrix multiplication, which is crucial for the artificial intelligence (AI) workloads. However, MZI meshes face two main limitations: they can only perform linear transformations, which limits their computational functions, and they have a larger footprint compared to electronic circuits. This study addresses these limitations by introducing a data representation method called structural nonlinearity to PICs for performing nonlinear operations, and differentiable photonic topology optimization (DTO) method which can maintain high accuracy on machine learning tasks while finding the topology with the minimal footprint.

Structural nonlinearity, achieved by repeating input information across consecutive MZI modulators, obtains nonlinear data mapping without increasing hardware complexity and improves machine learning performance. Whereas DTO optimizes MZI mesh structures by relaxing discrete architectures into continuous variables for gradient-based optimization. This reduces mesh size and enhances efficiency while maintaining accuracy.

In designing an MZI mesh for the Fashion MNIST task, structural nonlinearity raises test accuracy from 82% to 90% using the same number of MZIs, while DTO reduces the mesh footprint by 50% without sacrificing performance. Furthermore, incorporating experimentally obtained MZI characteristics into the optimization leads to a four-fold decrease in susceptibility to manufacturing imperfections and experimental noise sources. With its high efficiency and robustness, this design and computing framework can enable PICs to address large-scale AI tasks.

Keywords: Optical computing, photonic integrated circuits, machine learning, artificial intelligence, topology optimization.

Introduction

Electronic computing hardware faces challenges such as high energy consumption and limited computing speed, especially with the immense requirements of large-scale modern artificial intelligence models[1]. On the other hand, unlike electrons, photons do not interact with each other and this gives optics distinct features such as large bandwidth and low loss making it a promising candidate to address the limitations of the conventional electrical computing methods[2,3]. These advantages are used in different optical computing schemes. Matrix vector multiplication in free space can perform linear algebra operations in a massively parallel manner by transmitting a light beam encoded with the input vector through a light modulator[4]. By cascading multiple modulation layers interleaved with free space diffraction, diffractive optical networks are formed and they can be designed for performing computing tasks such as classification[5], encryption[6] and image reconstruction[7].

Optical computing architectures based on integrated platforms have also been investigated for smaller form factors and simpler manufacturing[8]. These platforms are based on different modulation units such as crossbar connections[9], microring modulators[10], attenuation controlled waveguides[11] and Mach-Zehnder interferometers(MZIs)[12]. Being one of the most widespread units thanks to its applications in telecommunication and quantum optics, MZIs consist of two beam splitters and two phase shifters to generate a signal pair with a certain intensity ratio and phase difference. MZIs constitute elements of photonics integrated computing meshes, which can accelerate linear operations required also by artificial intelligence (AI) models[13]. While these photonics accelerators can perform linear operations efficiently, the lack of interactions between photons make it challenging to obtain nonlinear functions, which are crucial for high accuracy AI models, with optical processors. The obvious solution is to convert optical signals to electrical signals, perform nonlinear functions, and then regenerate optical signals. However, this procedure can bottleneck the computing flow and increase energy consumption[14]. Generating nonlinear functions optically necessitate light-matter interactions, either through specifically highly nonlinear materials such as graphene[15] and MoS_2[16], which are generally not compatible with mature CMOS manufacturing, or with ultrashort pulses[17,18], which require mode-locked lasers.

Especially while executing machine learning (ML) tasks on photonic integrated circuits (PICs), the design of the MZI mesh topology is crucial to find optimal performance-to-size balance and noise resistance. Due to the discrete nature of topology choices, this creates a non-differentiable search space, making the optimization more complicated than with continuous variables, which natively support gradient-based methods for efficient optimization. An analog of this problem is already addressed in the architecture search of neural networks. A large body of work employed algorithms such as transferable architecture search[19], regularized evolution[20], hierarchical representations[21], and a search through hypernetworks[22]. However, these traditional architecture search methods mainly rely on going through most of the available architectures requiring a large amount of computing resources[23]. In contrast, gradient-based methods work on a single architecture and by approximating the gradients of the loss function with respect to the changes to the architecture[24]. In addition to the applications of gradient-based methods in finding optimal continuous variable values in the design of integrated photonic devices, including photonic crystals[25], ring resonators, wavelength demultiplexers[26] and beamforming meshes[27], there has also been efforts to find more compact PICs for optical neural networks through pruning[28].

In this study, we demonstrate a design and training framework for integrated, interferometer mesh-based PICs with structural nonlinearity (SN) information transform method and differentiable photonic topology optimization (DTO). While this approach provides significant improvement in machine learning performance with nonlinear data transform capabilities, achieves this with resiliency to experimental nonidealities and minimal chip footprint. SN effect creates a nonlinear mapping between the input data and output field by modulating an optical field repetitively with the same modulation term with the benefit of low optical powers and without specialized materials[29]. This method creates programmable high-order polynomial terms of the data at the output without optical nonlinearities[30,31]. In the context of PICs, the multiple modulations with the same data input over consequent MZI sub-meshes produce SN as shown in Fig. 1a. The required order and dimensionality of the SN depends on the ML problem and calculating the required scale of the system is not straightforward as in linear implementations. For this reason, DTO is especially crucial to learn the optimal topology of the PIC from the data. It finds the optimal architecture by relaxing the originally non-differentiable, discrete problem to a continuous parameter optimization by defining it as a softmax-weighted sum of possible connectivity choices (mesh type and path direction), allowing gradient descent to identify the best configuration.

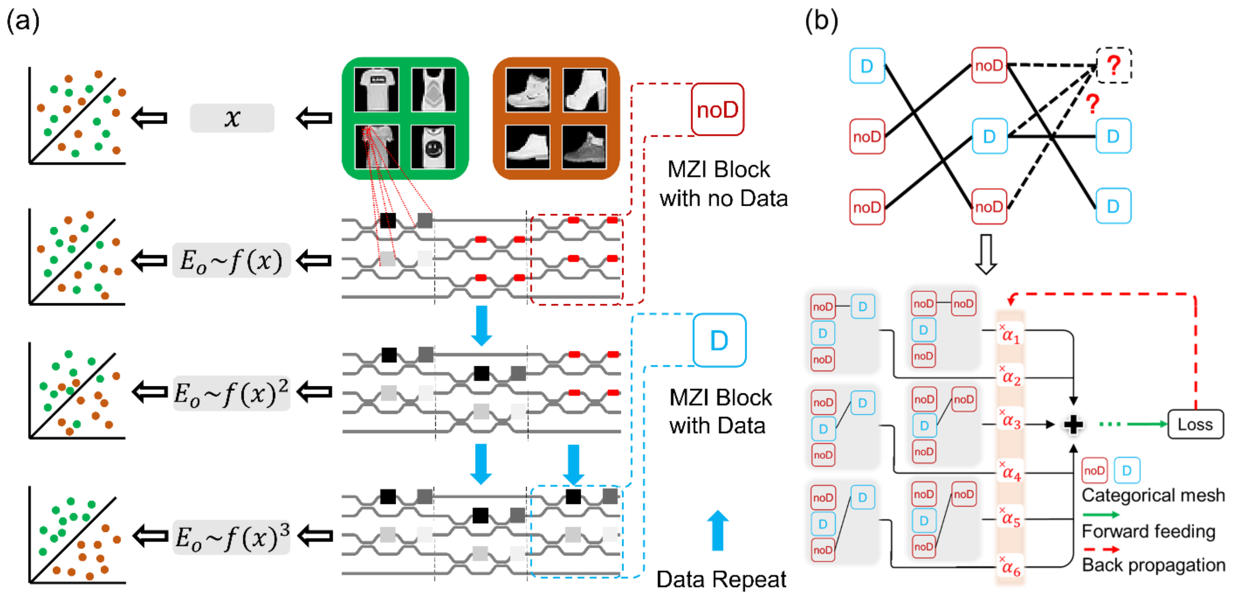


Figure 1 Schematic illustrations of SN and DTO principles. a SN is obtained by introducing input information at different parts of the mesh multiple times. Each repetition of the input on the light path creates a programmable higher order polynomial and improves the performance of optical computing. Grayscale colored modulators include data modulation while red ones do not. **b** DTO assigns every possible topographical choice (sub-mesh type and connectivity) a trainable parameter which weighs the outputs of the corresponding path, then gradient descent training maximizes the parameter of the optimal connection.

The generation of SN in an interferometer mesh is illustrated in Fig. 1a, the repetition of data on the beam path produces a polynomial order nonlinearity.

This effect can be summarized as follows:

$$E_o(x) = c_N f(x)^N + \cdots + c_2 f(x)^2 + c_1 f(x) + c_0 \qquad (1)$$

where N stands for how many times the data is repeated, E_o is the optical electric field at the output, c_n stands for coefficients of different orders of the polynomial terms. $f(x)$ refers to the trainable mapping between the input data, x and transformation matrix elements created by the MZI mesh. The order of nonlinearity (i.e. polynomial terms) increases with the number of times the data is repeated. In this work, we demonstrate that this principle can induce a high-dimensional and complex nonlinear relationship between information inputs to PICs, x, and their outputs, $E_o(x)$. The ideal power of the polynomial and its connectivity is determined using DTO, which optimizes the topology by parameterizing the probability of each connection with a continuous variable, α_i and learning these values through gradient descent. During training, as shown in Fig. 1b, the mesh output is a weighted combination of all possible connections, with the weight of each potential connection calculated from its selection parameter, α_i, after applying softmax nonlinear activation over all options (as detailed in the Methods section). The softmax converts the discrete selection problem into a continuous one, similar to its use in classification tasks, where the total probability of all outcomes is normalized to one. Throughout the training process, the probability of one option approaches one, effectively selecting it. This approach allows DTO to use significantly fewer computing resources than traditional architecture search algorithms. For example, finding the optimal topology in the search space of the problem shown in Fig. 3b takes less than 0.5 GPU-days with DTO, while regularized evolution can require up to 3000 GPU-days.

Results

Effects of Different Data Encoding Approaches and Topology Optimization

In the proposed architecture, the complete MZI mesh consists of different sub-meshes that process information in parallel channels across multiple steps, as shown in Fig. 2, similar to different layers in a neural network. The connectivity within each sub-mesh is arranged in a grid structure, where two output ports of an MZI layer are connected to the input ports of two different consequent MZIs, following the Clements topology[32]. Fig. 1a shows the diagram of a mesh with 3 layers, and the detailed architecture is shown in the blow-ups in Fig. 2.

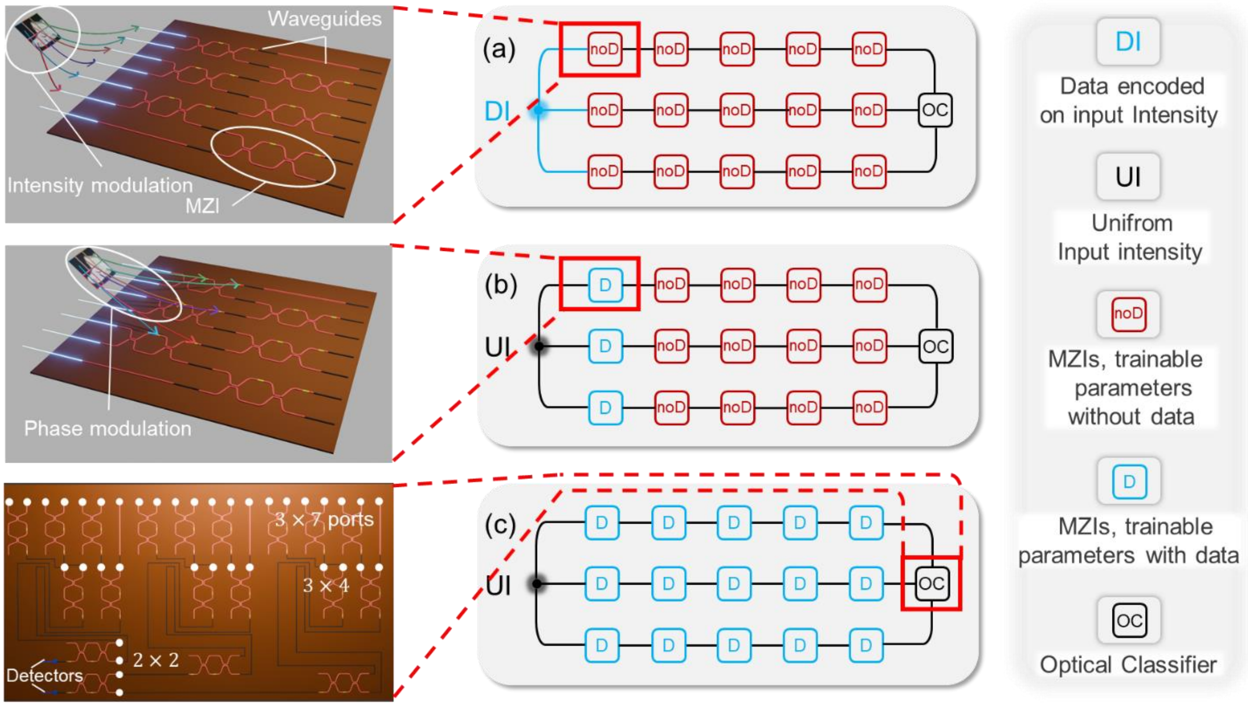


Figure 2 Three hand-designed MZI mesh architectures serving as baselines, all featuring a final optical classifier sub-mesh that directly outputs the prediction as intensities. a The model without nonlinearity (MNNL), input information is introduced as intensity modulation and all MZI sub-meshes have trainable modulation phase delays without the data term. **b** The model with phase nonlinearity (MPNL), input information is introduced as phase modulation in the first layer of MZI sub-meshes and the later layers have trainable modulation phase delays without the data term. **c** The model with 5-order structural nonlinearity (MSNL), input data is introduced to all MZI sub-meshes.

Each MZI within the mesh transforms the electromagnetic field from its input to output ports based on the phase delays between its two arms, controlled on by two modulators, θ and ϕ. This transformation can be expressed as:

$$\boldsymbol{E}_{out} = \bar{\boldsymbol{U}}\boldsymbol{E}_{in} = \begin{bmatrix} e^{i\phi}\cos\theta & -\sin\theta \\ e^{i\phi}\sin\theta & \cos\theta \end{bmatrix} \boldsymbol{E}_{in}. \quad (1)$$

According to this relationship, while $\boldsymbol{E}_{out}$ depends linearly on $\boldsymbol{E}_{in}$, the values of θ and ϕ affect $\boldsymbol{E}_{out}$ nonlinearly. The impact of these linear and nonlinear relationships on machine learning performance is studied through two architectures, where the data modulates either $\boldsymbol{E}_{in}$ (Fig. 2a) or (θ, ϕ) pairs (Fig. 2b), while the rest of the phase delays are treated as trainable parameters ('noD' sub-meshes on Fig. 2). The third architecture shown in Fig.

2c introduces the data on MZI phase pairs again, however not only in the first sub-mesh, but on the consecutive sub-meshes as well ('D' sub-meshes on Fig. 2), enabling SN with polynomial orders up to the 5th. When the data is transferred to the given physical signals, a trainable linear projection is applied as $f(x) = s \cdot x + b$ where s and b are scaling and bias parameters, and x is the raw data channel. These three baseline models are designed to compare the advantages of nonlinearities due to phase encoding and SN against a fully linear approach, in terms of test accuracy and total parameter counts, as later shown in Fig. 4.

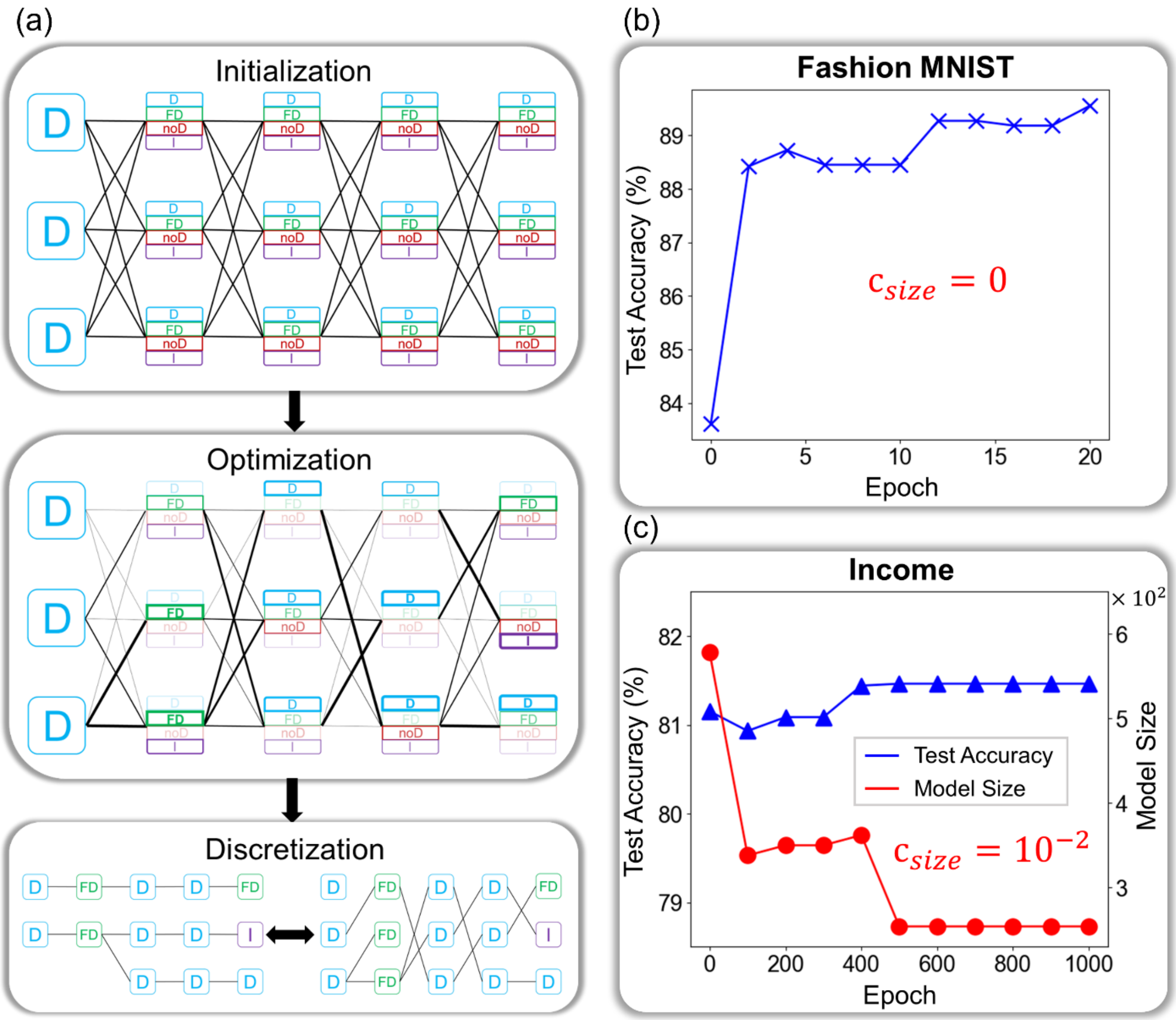


Figure 3 The steps of DTO algorithm and the evolution of mesh performance and size for different tasks. a The pipeline of DTO includes three steps: 'Initialization' constructs the model of DTO by relaxing the categorical choice of a particular submesh

to a softmax activation over all possible choices. 'Optimization' updates the values of architecture parameters using steepest descent. They are the coefficients assigned to preferability of each possible operation ($\alpha_1, \ldots, \alpha_\mathrm{n}$), here visualized with the opacity of the connections and sub-meshes. 'Discretization' builds the final model by fixing the connection with the biggest parameter. **b** The test accuracy on Fashion MNIST dataset[33] during DTO without mesh size loss term ($c_{size} = 0$). **c** The test accuracy on 'Income' dataset and total mesh size during DTO, while penalizing the mesh size ($c_{size} > 0$).

DTO designs the PIC mesh by selecting and connecting compute blocks, or sub-meshes, to maximize performance on a given task. In addition to 'D' and 'noD' sub-meshes used in the baseline architectures, DTO has two additional sub-mesh options; the identity block, 'I', which directly transmits its inputs to outputs unchanged, effectively reducing the mesh size, and the flipped data block, 'FD', which reverses the order of the data vector, $\boldsymbol{x}$, and creates interactions between different locations of the data when it is on the same path with a 'D' sub-mesh. Within the process flow of DTO, as shown in Fig. 3a, all selections of sub-meshes and their connectivities are initialized equally likely with uniform architecture parameters, ($\alpha_1 = \alpha_2 = \cdots = \alpha_\mathrm{n}$). During optimization, gradient descent adjusts both the sub-mesh transforms (via s and b) and the connection probabilities (via α_n). After optimization, the final architecture is determined by selecting the connections with the largest α_n values. Connections with near-zero α_n values, already minimized through softmax, are removed to discretize the model.

The differentiable nature of the proposed topology optimization algorithm allows the incorporation of additional objectives into the workflow. For example, to achieve an optimal trade-off between the task performance and grid size, the loss function of the gradient descent in DTO can be defined as:

$$\mathcal{L} = \mathcal{L}_{performance} + c_{size} \cdot \mathcal{L}_{size} \qquad (3)$$

Here, $\mathcal{L}_{performance}$ represents the cross-entropy loss between classification predictions and true labels, $\mathcal{L}_{size}$ is proportional to the number of active MZI modulators, accounting for the model size and c_{size} determines the relative importance of reducing mesh size compared to improving performance. When $c_{size} = 0$, the optimization focuses solely on maximizing accuracy. As shown in Fig. 3b, DTO improves the test accuracy of the MZI mesh on Fashion MNIST dataset by 6%, reaching 89.5%, which is comparable to the performance of digital convolutional neural networks[34]. The intermediate accuracies on the learning curve are obtained by fully training the phase values of the most probable architecture at each point in the optimization process.

For scenarios where a smaller PIC is preferred (e.g., due to limited chip footprint), setting $c_{size} > 0$ allows DTO to miniaturize the mesh by over 50% without a significant loss of performance, as depicted in Fig. 3c. The Adult Incomes dataset[35] has relatively fewer samples and features than Fashion MNIST[33], therefore the scale of the mesh is smaller but more epochs are required. This highlights the flexibility of the DTO approach across different datasets and constraints.

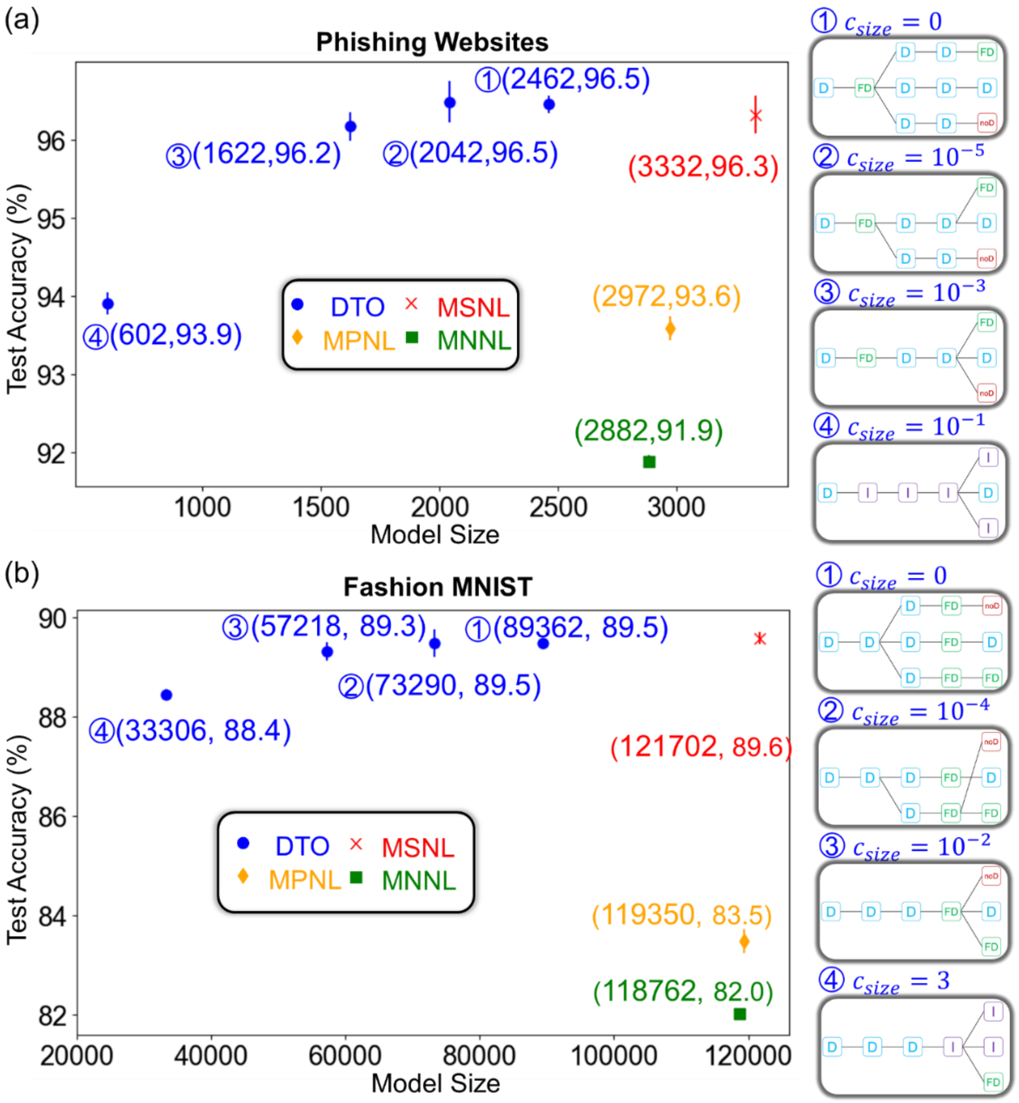


Figure 4 The model size and accuracy comparison between fixed baseline topologies and DTO designed architectures under different c_{size} values. The test results of different architectures on Phishing Websites dataset[36] are shown in **a** and the analysis on Fashion MNIST[33] dataset is in **b**. The first value next to the datapoints gives the total number of parameters in the architecture and the second is the mean test accuracy, while the standard deviation is visualized with the vertical bars, the topologies found with DTO are enumerated and the corresponding meshes and their size penalty factors appear on the right panel.

We analyzed the performance of structural nonlinearity and DTO using four datasets. The Phishing Websites dataset involves classifying websites as phishing or legitimate based on features such as the presence of an IP address or subdomain. The Fashion MNIST dataset contains images of clothing items for classification. Results for these datasets are presented in Figure 4. Additionally, two more datasets are included in the supplementary material: the Mushroom dataset, which classifies mushrooms as edible or poisonous based on physical characteristics, and the Income dataset, which predicts income categories based on demographic information such as age, gender, and education.

For each dataset, we tested three baseline architectures—MSNL, MPNL, and MNNL—along with four architectures designed by DTO under different c_{size} values. For the Phishing Websites dataset, a single sub-mesh ('D') contains 31 ports and 6 layers. To explore model pruning, we incrementally increased c_{size} values from 0 to $10^{-5}, 10^{-3}, 10^{-1}$ Among the baseline architectures, the mesh with structural nonlinearity (MSNL) outperformed MPNL and MNNL while maintaining the same hardware complexity, highlighting the importance of structural nonlinearity. Notably, DTO consistently designed meshes that incorporated structural nonlinearity, demonstrating the algorithm's synergy with this feature. At $\mathrm{c}_{size} = 0$, DTO-optimized meshes achieved the highest test accuracy. As c_{size} increased, the meshes displayed an improved performance-to-size trade-off while still retaining structural nonlinearity.

For the Fashion MNIST dataset, similar trends were observed. One sub-mesh in this dataset comprises 196 ports and 40 layers. We tested c_{size} values of $0, 10^{-4}, 10^{-2}, 3$. The MSNL architecture again outperformed MPNL and MNNL, confirming the benefits of structural nonlinearity. DTO-designed meshes achieved high accuracy at $\mathrm{c}_{size} = 0$ and demonstrated effective size reduction with increasing c_{size}, retaining structural nonlinearity even at $\mathrm{c}_{size} = 3$. While a slight drop in test accuracy was observed at the largest c_{size} , the optimized meshes still outperformed significantly larger baseline meshes (MPNL and MNNL) lacking structural nonlinearity.

Topology Optimization with Experimental Mesh Characterization

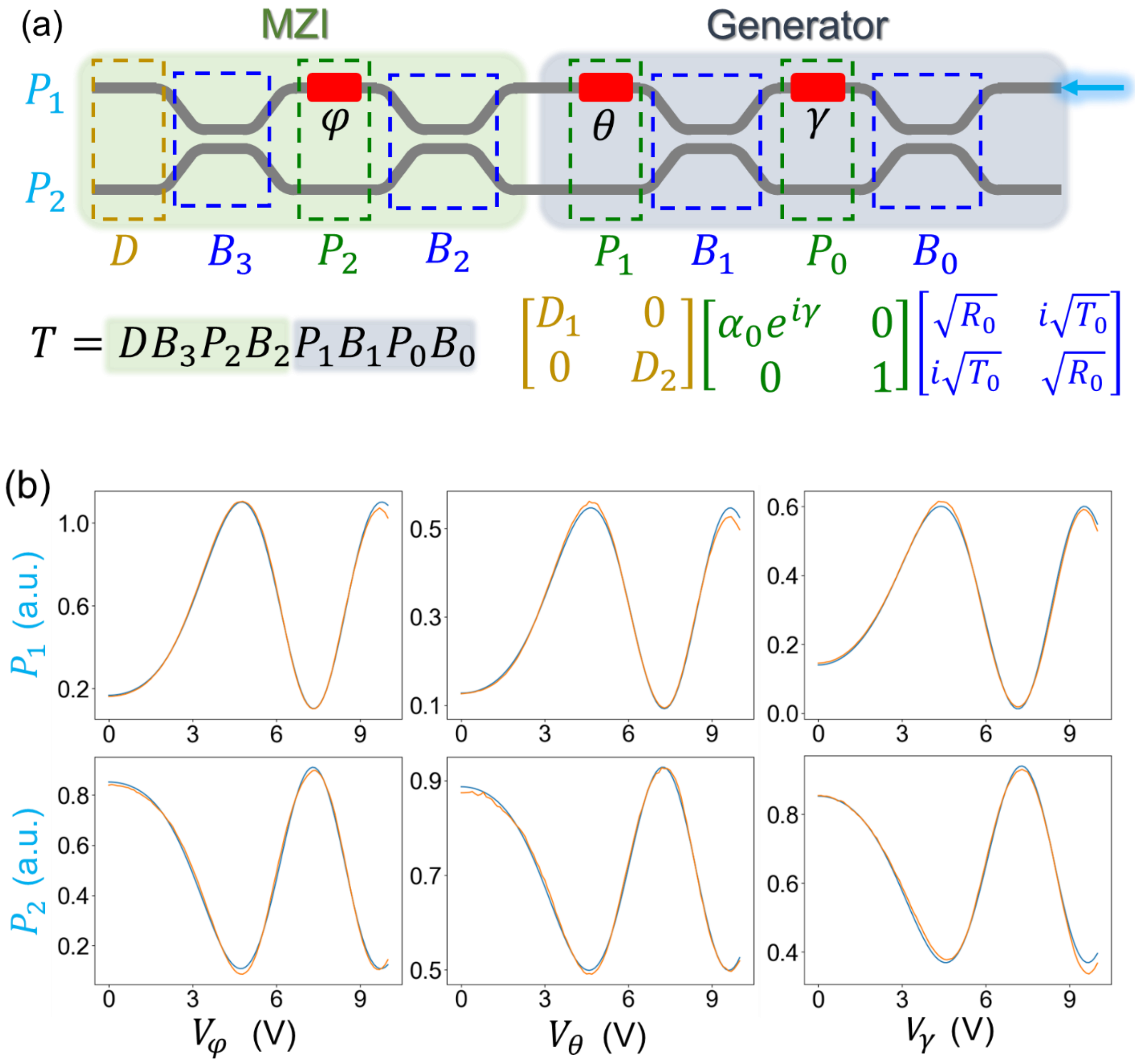

(a)
MZI
Generator
P_1
P_2
φ
θ
γ
D
B_3
P_2
B_2
P_1
B_1
P_0
B_0
$T = DB_3P_2B_2P_1B_1P_0B_0$
$\begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} \alpha_0 e^{i\gamma} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{R_0} & i\sqrt{T_0} \\ i\sqrt{T_0} & \sqrt{R_0} \end{bmatrix}$
(b)
P_1 (a.u.)
P_2 (a.u.)
V_φ (V)
V_θ (V)
V_γ (V)
0
3
6
9
1.0
0.6
0.2
0.5
0.3
0.1
0.6
0.4
0.2
0.0
0.8
0.6
0.4
0.2
0.9
0.7
0.5
0.8
0.6
0.4

Figure 5: (a) shows the schematic of the mesh for experimental validation where constant input power enters from the right hand side and the intermediate results are read out from the left hand side. (b) shows calibration data of the experimental device.

Provide preface The previous study is done with ideal device parameters Bandwidth of MZI modulators are very high- particularly useful to use time multiplexing in addition to scaling chip size wise… For quickly prototypingBant and analyzing the effects of the proposed algorithms, we adopt a time-multiplexing approach. We record the output of one fabricated MZI experimentally and use it to implement a digital twin as the input of the next MZI. Figure 5 (a) shows the diagram of the device for experimental validation. The first MZI in the device is a generator gives signals with wanted intensity ratio and phase difference. The second MZI in the device corresponds to MZIs in the mesh. We only detect intensity in experiment and use simulation phase. Figure 5 (b) shows the output intensity under the sweep of different phase shifters. By calibration with these six curves, we can get imperfet parameters mentioned in Figure 5 (a), e.g., $R_0 = 0.62, R_1 = 0.38, R_2 = 0.38, R_3 = 0.58$. Imperfect R_0 and R_1 gives limited range of intensity ratio of MZI's input. We add a clamp layer in mesh simulation to follow imperfect R_0, R_1 and update transformation matrix of MZIs in mesh to follow imperfect R_2, R_3. We also consider noises besides imperfections. One part is the noise at phase shifters. Another part is the noise at power detectors. The errors caused by noises will accumulate as the number of layers increases. To get anti-noise parameters, we add random noises $\mathcal{N}(0,\ (0.06\pi)^2)$ to MZI phases and $\mathcal{N}(0,\ (0.1\mathrm{I})^2)$ to output intensity of MZIs when we use gradient descent to train the parameters of an MZI mesh. I means the intensity of this signal.

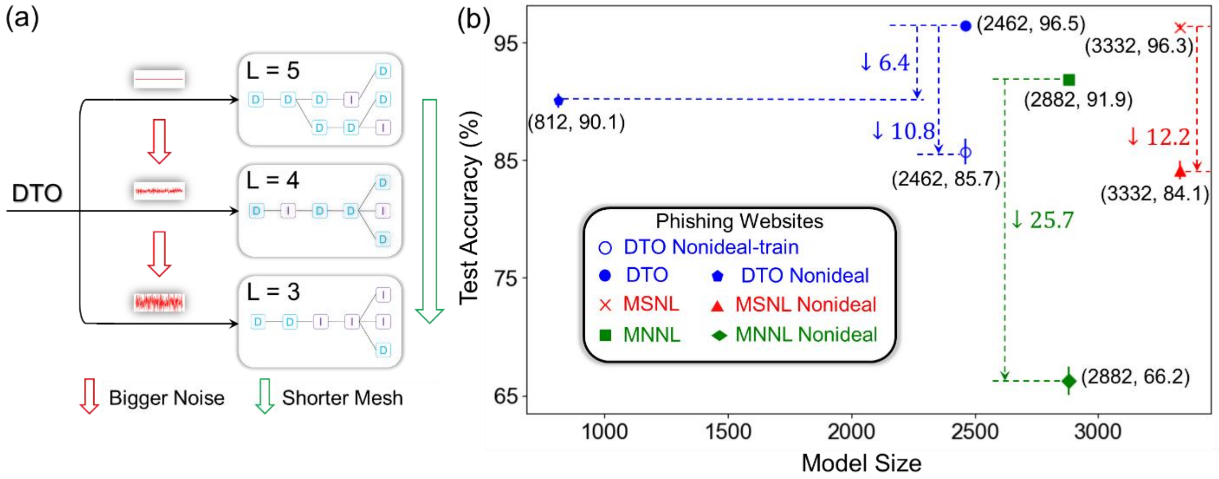


Figure 6: Test results of proposed models on Phishing Websites dataset considering imperfections and noises. (a) shows that DTO can give shorter meshes which contribute to less error accumulation as we increase noises. (b) shows the influence of non-idealities on different architectures. DTO Nonideal-train still uses the architecture search result of DTO under ideal case (① in Figure 4 (a)) while DTO Nonideal also introduce non-idealities in the search procedure of DTO (case 3 of Figure 6 (a)). All meshes have a test accuracy drop. The test accuracy drop of SNL is much smaller that of NNL which proves the anti-noise ability of structural nonlinearity. The test accuracy of the architecture DTO Nonideal corresponds to the smallest test accuracy drop and is noticeably better than other meshes.

We would expect shorter meshes to accumulate less noise and therefore show an increase in accuracy. We want to test if the architectures found with DTO confirm this. We will prove the effects of structural nonlinearity by the compare between NNL and SNL. Hence here we simplify DTO by reducing “selecting from ‘D’, ‘FD’, ‘noD’, ‘I’ to “selecting from ‘D’, ‘I’”. Figure 6 (a) tells how the result architectures and their sizes change as we increase the level of noises. The first architecture is given by DTO consider only imperfections. The second architecture is given by DTO consider both imperfections and noises $\mathcal{N}(0,\ (0.03\pi)^2)$ to MZI phases, $\mathcal{N}(0,\ (0.05\mathrm{I})^2)$ to output intensity. The third architecture is given by DTO consider both imperfections and noises $\mathcal{N}(0,\ (0.06\pi)^2)$ to MZI phases, $\mathcal{N}(0,\ (0.1\mathrm{I})^2)$ to output intensity. It is clear that we can get shorter meshes as we increase noises, because shorter meshes mean smaller accumulation of errors. Figure 6 (b) shows the influence of non-idealities, i.e., imperfections and noises. DTO Nonideal-train means that we still use the architecture search result of DTO under ideal case (① in Figure 4 (a)) and introduce non-idealities in the separate training procedure of discretized architecture. DTO Nonideal means we introduce non-idealities in both the search procedure of DTO (case 3 of Figure 6 (a)) and the separate training procedure of discretized architecture.

Compared to the simulation under ideal case, All meshes have a test accuracy drop. The effects of structural nonlinearity are still obvious. The test accuracy drop of SNL is much smaller that of NNL. It means structural nonlinearity has better anti-noise ability. The test accuracy of the architecture given by DTO under ideal case has smaller test accuracy drop than other two baseline meshes. The test accuracy of the architecture given by DTO considering non-idealities has the smallest test accuracy drop and is noticeably better than other meshes.

Discussion

Our experiment shows that for all four datasets, the baseline mesh with structural nonlinearity (SNL) has noticeable test accuracy improvement compared to the mesh without nonlinearity (NNL) and the mesh with only phase-intensity nonlinearity (PNL).

The architecture optimization and test results show DTO as a powerful tool to guide photonic mesh design. DTO gives architectures that have comparable test accuracy to SNL and also a smaller model size. It navigates through the trade-off between test accuracy and mesh size. When we increase the coefficient c_{size} for the model size loss loss_{size}, the model size has a noticeable drop while the test accuracy stays consistent at a reasonable range. This is especially useful when chip real estate is expensive and circuits need pruning.

The outputs of DTO also reflects the power of structural nonlinearity. . When we set a big value for the model size loss coefficient c_{size}, the model size of the optimized mesh is much smaller than that of NNL or PNL. However, it always keeps structural nonlinearity, and its test accuracy is always higher than the test accuracy of these two baseline meshes. Structural nonlinearity has different performances on different datasets. It has the lowest effects on Income, i.e., for Income, the improvement in the test accuracy is only 0.5% when we use SNL compared to PNL. For the three middle columns among the five columns of the whole mesh, DTO will choose one-order data repetition for Income, while it will choose three-order data repetition for the other three datasets. It means that when structural nonlinearity is very useful for the dataset in investigation, DTO will choose high-order structural nonlinearity automatically. Moreover, when we optimize the meshes for Fashion MNIST, the meshes can have more than 50000 parameters. When we optimize the meshes for the other datasets, the meshes have only hundreds of or thousands of parameters. DTO can work for both cases.

When we introduce imperfect device parameters and experimental noise, the previous conclusions still hold. SNL still has better test accuracy than meshes without structural nonlinearity. The optimized mesh keeps good test accuracy and small pruned mode size, and due to small noise accumulation, the optimized mesh with the smallest model size even has much better test accuracy than other meshes. Overall, these results lay a promising path forward for photonic circuits incorporating nonlinear processing with CW light at low power.

To discuss practical future implementations, we consider two platforms: silicon photonics with thermal modulators, representing a mature technology[12], and thin-film lithium niobate (TFLN), an emerging platform offering high-speed operation with low driving voltages through the electro-optic effect[37,38]. While other platforms, such as graphene-based modulators[39], are under investigation, we focus on these two to provide benchmarks. For the Fashion MNIST dataset, our optimization yields a network with approximately 33,000 parameters. Without employing time multiplexing, realizing this network requires implementing roughly 17,000 MZIs on a chip. Assuming a footprint of 0.004 mm² per MZI in silicon photonics, this results in a total area of about 68 mm². In TFLN, with an assumed footprint of 0.06 mm² per MZI, the total area increases to approximately 1020 mm². The inference rates are estimated to be around 100 kHz for silicon photonics with thermal modulators and up to 100 GHz for TFLN electro-optic modulators, assuming that the bit-depth of modulation and the precision of the model are matched. When we employ time multiplexing with two modulators, one generating intermediate complex light fields and the other performing computations using trained parameters, the footprint can be significantly reduced. However, time multiplexing hampers the inference rate to roughly 6 Hz for silicon photonics and 6 MHz for TFLN. This suggests that TFLN, when combined with time multiplexing, presents a promising path forward due to its high-speed capabilities since the resulting 6 MHz inference rate is still competitive.

Materials and Methods

Here we give the mathematical description of DTO. First, to make the design space continuous, this algorithm relaxes the categorical choice of a particular operation to a softmax activation over all possible operations. The principle of this algorithm can be seen in equation (3) as follows:

$$\bar{o}(\mathrm{x}) = \sum_{o\in O} \frac{\exp(\alpha_o)}{\sum_{o'\in O}\exp(\alpha_{o'})} o(x) \tag{3}$$

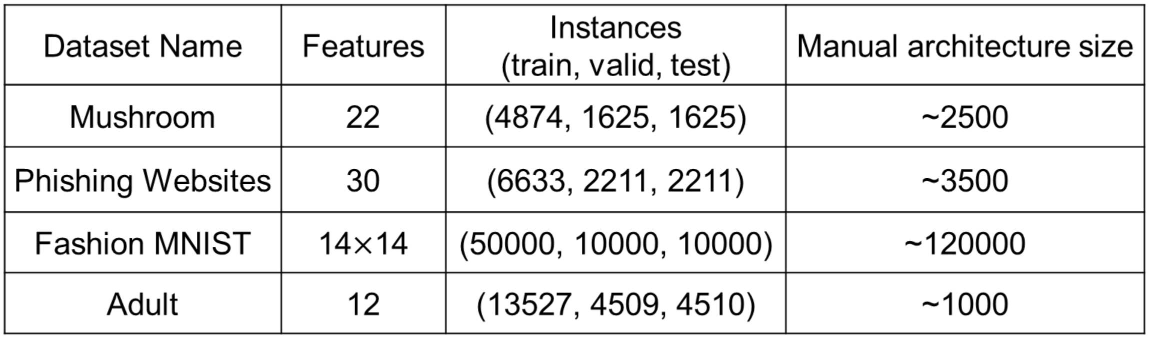
where x is signal, O is a set of candidate operations, i.e., different sub-meshes, o is an operation in set O to be applied to signal x, α is a mixing weight vector connects to operation set O, α_o is an element connects to operation o. The operation used for mesh design is a mix of all possible operations parameterized by a trainable weight vector. The task of mesh design then reduces to learning a set of continuous variables α. Equation (3) corresponds to Initialization of Figure 3 (a). Then, the algorithm jointly learns the architecture α and the weights w within all the mixed operations to solve the bi-level optimization problem described in equation (4) as follows:

$$\begin{aligned}&\underset{\alpha}{\mathrm{Min}}\ \ \mathcal{L}_{val}(w^*(\alpha),\alpha)\\ &\mathrm{s.t.}\ \ w^*(\alpha) = \mathrm{argmin}_{\mathrm{w}}\mathcal{L}_{train}(w,\alpha)\end{aligned} \tag{4}$$

where $\mathcal{L}_{val}$ means valid loss and $\mathcal{L}_{train}$ means train loss. Equation (4) corresponds to Optimization of Figure 3 (a). At the end of the design, a discrete mesh can be obtained by replacing each mixed operation ō with the most likely operation as follows:

$$o\ = \mathrm{argmax}_{\mathrm{o}\in\mathrm{O}}\ \alpha_o \tag{5}$$

Equation (5) corresponds to the discretization of Figure 3 (a).



Dataset Name	Features	Instances (train, valid, test)	Manual architecture size
Mushroom	22	(4874, 1625, 1625)	~2500
Phishing Websites	30	(6633, 2211, 2211)	~3500
Fashion MNIST	14×14	(50000, 10000, 10000)	~120000
Adult	12	(13527, 4509, 4510)	~1000

Table 1: Information about the four datasets used.

Table 1 gives further information about the four datasets we use. It tells features, i.e. the size of one sample. It tells instances, i.e. the size of trainset, validset, testset. We use an RTX 2080Ti GPU to train the deep learning tasks. The packages we directly use or use as reference include Pytorch, Numpy. The mean and variance of test accuracy mentioned in this report are calculated by three tests under different random seeds. When we do DTO, we use SGD optimizer for model weights training and Adam optimizer for architecture variables training which follows the original paper. Adam optimizer performs the training of the baseline or optimized meshes to calculate their final accuracy on the test set. We do preprocessing for the four datasets before using them to do deep learning. When doing dataset preprocessing, we choose to adopt layer normalization as a priority, because we use trainable scale and bias parameters (shared by different samples) to load data which can already be seen as a kind of batch normalization. For Phishing Websites, we don't use normalization to preprocess it, since its feature values only include -1,0,1. For Mushroom, Fashion MNIST, we use layer normalization. For Income, the value range of every feature is big and very different, so we use batch normalization. The original Income dataset has 14 features, we delete two features 'education' and 'occupation' and some samples of this dataset to make it become a balanced dataset.

Codes availability

Codes used in this report are available under reasonable request.

References

1. Luccioni, A. S., Jernite, Y. & Strubell, E. Power Hungry Processing: Watts Driving the Cost of AI Deployment? in *The 2024 ACM Conference on Fairness, Accountability, and Transparency* 85–99 (2024). doi:10.1145/3630106.3658542.
2. McMahon, P. L. The physics of optical computing. *Nat. Rev. Phys.* **5**, 717–734 (2023).
3. Wetzstein, G. *et al.* Inference in artificial intelligence with deep optics and photonics. *Nat. 2020 5887836* **588**, 39–47 (2020).
4. Spall, J., Guo, X., Barrett, T. D. & Lvovsky, A. I. Fully reconfigurable coherent optical vector–matrix multiplication. *Opt. Lett.* **45**, 5752–5755 (2020).
5. Lin, X. *et al.* All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008 (2018).
6. Bai, B. *et al.* Data-Class-Specific All-Optical Transformations and Encryption. *Adv. Mater.* **35**, 2212091 (2023).
7. Zhou, T. *et al.* In situ optical backpropagation training of diffractive optical neural networks. *Photonics Res.* **8**, 940–953 (2020).
8. Bogaerts, W. *et al.* Programmable photonic circuits. *Nature* **586**, 207–216 (2020).
9. Youngblood, N. Coherent Photonic Crossbar Arrays for Large-Scale Matrix-Matrix Multiplication. *IEEE J. Sel. Top. Quantum Electron.* **29**, 1–11 (2023).
10. Huang, C. *et al.* Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits. *APL Photonics* **5**, 040803 (2020).
11. Ashtiani, F., Geers, A. J. & Aflatouni, F. An on-chip photonic deep neural network for image classification. *Nature* **606**, 501–506 (2022).
12. Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nat. Photonics* **11**, 441–446 (2017).

13. Shastri, B. J. *et al.* Photonics for artificial intelligence and neuromorphic computing. *Nat. Photonics 2021 152* **15**, 102–114 (2021).

14. Zhou, T. *et al.* Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nat. Photonics* **15**, 367–373 (2021).

15. You, J. *et al.* Hybrid/Integrated Silicon Photonics Based on 2D Materials in Optical Communication Nanosystems. *Laser Photonics Rev.* **14**, 2000239 (2020).

16. Taghinejad, M. *et al.* Photocarrier-induced active control of second-order optical nonlinearity in monolayer MoS2. *Small* **16**, 1906347 (2020).

17. Teğin, U., Yıldırım, M., Oğuz, İ., Moser, C. & Psaltis, D. Scalable optical learning operator. *Nat. Comput. Sci. 2021 18* **1**, 542–549 (2021).

18. Yildirim, M. *et al.* Nonlinear optical feature generator for machine learning. *APL Photonics* **8**, 106104 (2023).

19. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. Learning transferable architectures for scalable image recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 8697–8710 (2018).

20. Real, E., Aggarwal, A., Huang, Y. & Le, Q. V. Regularized evolution for image classifier architecture search. in *Proceedings of the aaai conference on artificial intelligence* vol. 33 4780–4789 (2019).

21. Liu, H., Simonyan, K., Vinyals, O., Fernando, C. & Kavukcuoglu, K. Hierarchical representations for efficient architecture search. *ArXiv Prepr. ArXiv171100436* (2017).

22. Brock, A., Lim, T., Ritchie, J. M. & Weston, N. Smash: one-shot model architecture search through hypernetworks. *ArXiv Prepr. ArXiv170805344* (2017).

23. Baymurzina, D., Golikov, E. & Burtsev, M. A review of neural architecture search. *Neurocomputing* **474**, 82–93 (2022).

24. Liu, H., Simonyan, K. & Yang, Y. DARTS: Differentiable Architecture Search. in (2018).

25. Minkov, M. *et al.* Inverse Design of Photonic Crystals through Automatic Differentiation. *ACS Photonics* **7**, 1729–1741 (2020).

26. Gao, Z. *et al.* Gradient-Based Power Efficient Functional Synthesis for Programmable Photonic Circuits. *J. Light. Technol.* **42**, 5956–5965 (2024).

27. López, D. P. Programmable Integrated Silicon Photonics Waveguide Meshes: Optimized Designs and Control Algorithms. *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–12 (2020).

28. Gu, J. *et al.* Toward Hardware-Efficient Optical Neural Networks: Beyond FFT Architecture via Joint Learnability. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **40**, 1796–1809 (2021).

29. Eliezer, Y., Rührmair, U., Wisiol, N., Bittner, S. & Cao, H. Tunable nonlinear optical mapping in a multiple-scattering cavity. *Proc. Natl. Acad. Sci.* **120**, e2305027120 (2023).

30. Yildirim, M., Dinc, N. U., Oguz, I., Psaltis, D. & Moser, C. Nonlinear processing with linear optics. *Nat. Photonics* 1–7 (2024).

31. Xia, F. *et al.* Nonlinear optical encoding enabled by recurrent linear scattering. *Nat. Photonics* 1–9 (2024).

32. Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S. & Walmsley, I. A. Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465 (2016).

33. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *ArXiv170807747 Cs Stat* (2017).

34. Bhatnagar, S., Ghosal, D. & Kolekar, M. H. Classification of fashion article images using convolutional neural networks. in *2017 Fourth International Conference on Image Information Processing (ICIIP)* 1–6 (2017). doi:10.1109/ICIIP.2017.8313740.

35. Kohavi, R. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* to appear (1996).

36. Mohammad, R. M., Thabtah, F. & McCluskey, L. An assessment of features related to phishing websites using an automated technique. in *2012 International Conference for Internet Technology and Secured Transactions* 492–497 (2012).

37. Wang, C. *et al.* Integrated lithium niobate electro-optic modulators operating at CMOS-compatible voltages. *Nature* **562**, 101–104 (2018).

38. Zhang, M., Wang, C., Kharel, P., Zhu, D. & Lončar, M. Integrated lithium niobate electro-optic modulators: when performance meets scalability. *Optica* **8**, 652–667 (2021).

39. Sorianello, V. *et al.* Graphene–silicon phase modulators with gigahertz bandwidth. *Nat. Photonics* **12**, 40–44 (2018).