

Programmable Photonic Extreme Learning Machines

José Roberto Rausell-Campo,* Antonio Hurtado, Daniel Pérez-López, and José Capmany Franco

Photonic neural networks offer a promising alternative to traditional electronic systems for machine learning accelerators due to their low latency and energy efficiency. However, the challenge of implementing the backpropagation algorithm during training has limited their development. To address this, alternative machine learning schemes, such as extreme learning machines (ELMs), are proposed. ELMs use a random hidden layer to increase the feature space dimensionality, requiring only the output layer to be trained through linear regression, thus reducing training complexity. Here, a programmable photonic extreme learning machine (PPELM) is experimentally demonstrated using a hexagonal waveguide mesh, and which enables to program directly on chip the input feature vector and the random hidden layer. This system also permits to apply the nonlinearity directly on-chip by using the system's integrated photodetecting elements. Using the PPELM, three different complex classification tasks are solved successfully. Additionally, two techniques are also proposed and demonstrated to increase the accuracy of the models and reduce their variability using an evolutionary algorithm and a wavelength division multiplexing approach, obtaining excellent performance. These results show that programmable photonic processors may become a feasible way to train competitive machine learning models on a versatile and compact platform.

1. Introduction

In recent years, artificial intelligence algorithms have gained increasing attention due to the development of more powerful and

complex architectures.^[1,2] This exponential increase in the size and number of required parameters has led to a higher demand for computational resources. Traditional electronic processors have been the backbone of these advancements, but they present limitations in bandwidth and energy efficiency. Consequently, alternative computing paradigms that address these issues have started to attract increasing research attention.^[3–6]

In this context, photonics has emerged as a viable alternative for machine learning hardware by exploiting the unique features of light-based computing, including complex-valued, multidimensional, and parallel operations.^[7–9] These systems promise to deliver high bandwidth, low latency processing and low power consumption.^[10] There have been several implementations of feed-forward neural networks (FF-NN) using free-space optics with spatial light modulators and diffractive elements.^[11,12] Moreover, the integration of photonic devices on-chip has enabled the development of FF-NN

using wavelength division multiplexing (WDM) with microring resonators (MRRs)^[13,14] and coherent systems with Mach-Zehnder interferometers (MZIs).^[15,16] However, a main challenge for these approaches arises during the training stage, where implementing the widely used backpropagation algorithm, requires very complex architectures, which are very difficult to scale.^[17,18]

Another interesting option is to use unconventional machine learning architectures that require fewer resources for training. Spiking neural networks have shown their potential using photonic systems to increase the efficiency of AI systems.^[19–21] Moreover, photonic reservoir computing has been experimentally demonstrated to solve time-dependent problems using recurrent systems with physical or virtual nodes.^[22–24]

Extreme learning machines (ELMs)^[25,26] are also an appealing alternative to traditional neural networks by simplifying its training process. ELMs are a type of FF-NN in which only the output layer weights are trained, while the connections of the input and hidden layer are randomly assigned and remain fixed. The idea is to randomly increase the dimensionality of an input vector to a new feature space where the data can be linearly separated. This methodology not only accelerates the training phase but also reduces the overall computational cost. ELMs are particularly well-suited for their photonic implementation on

J. R. Rausell-Campo, J. Capmany Franco
Photonics Research Lab, ITEAM
Universitat Politècnica de Valencia
Valencia 46022, Spain
E-mail: joraucam@upv.es

A. Hurtado
Institute of Photonics
Physics Department
University of Strathclyde
Glasgow G4 0NT, UK
D. Pérez-López
iPronics Programmable Photonics S.L.
Valencia 46010, Spain

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/lpor.202400870>

© 2025 The Author(s). Laser & Photonics Reviews published by Wiley-VCH GmbH. This is an open access article under the terms of the [Creative Commons Attribution](#) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/lpor.202400870

devices and systems that can inherently map the input optical signal into a higher dimensional space. Experimental demonstration of photonic ELMs (PELMs) using bulky systems have been already published using free-space propagation,^[27,28] speckle patterns^[29] and frequency multiplexed fibers.^[30] ELMs in photonic integrated circuits have been also demonstrated by processing the scattered light of gratings coupled to microresonators.^[31] This approach reduces the size of the PELM but still requires an external camera to record the hidden outputs. Moreover, the state of the random transformation of the photonic circuit is fixed which reduces the versatility of the PELM.

Here, we report a programmable photonic extreme learning machine (PPELM) using a general-purpose programmable processor.^[32–35] The programmable processor consists of a mesh of photonic tunable couplers in a hexagonal topology. We demonstrate how data can be encoded in both amplitude and phase using the tunable units. The processor can also be configured to adapt the number of inputs and outputs, making it suitable for different complex computational tasks. Numerous random matrices can be implemented on-chip by randomly setting the states of its building blocks. Furthermore, the inclusion of integrated photodetectors in the system allows the implementation of the nonlinear function directly on-chip. The final layer is trained and tested digitally revealing excellent performance in different complex tasks. In particular, the PPELM is experimentally validated using the Smartlight processor of iPrionics on three classification tasks: the recognition of headers in a bit pattern, the categorization of iris flowers specimens, and the authentication of banknotes. Finally, we leverage the programmability of the PPELM to introduce methods to increase the accuracy of the models by optimizing the random hidden layer using an evolutionary algorithm and applying an ensemble technique where different models are trained in parallel thanks to a wavelength division multiplexing approach.

2. Results

2.1. Extreme Learning Machines

Extreme learning machines base their performance in the random projection of the input feature vector \mathbf{X} that belongs to a space \mathbb{R}^{N_f} into a higher-dimensional space \mathbb{R}^{N_H} where the target classes can be linearly separated. N_f refers to the number of input features and N_H is the number of hidden nodes in the network. Mathematically, an input state \mathbf{X} is first multiplied by a random matrix \mathbf{W} and a set of biases \mathbf{b} are added. To achieve a nonlinear random projection, an activation function $f(\cdot)$ is applied:

$$\mathbf{H} = f(\mathbf{W}\mathbf{X} + \mathbf{b}) \quad (1)$$

where \mathbf{H} is the nonlinear random projection of the input state. The projected features are then linearly combined with a trainable weight matrix β to obtain the predicted outputs \mathbf{O} . Layer β can be trained using an analytical formula:

$$\beta = \mathbf{H}^\dagger \mathbf{T} \quad (2)$$

where \mathbf{T} is the ground-truth target matrix of the training dataset.

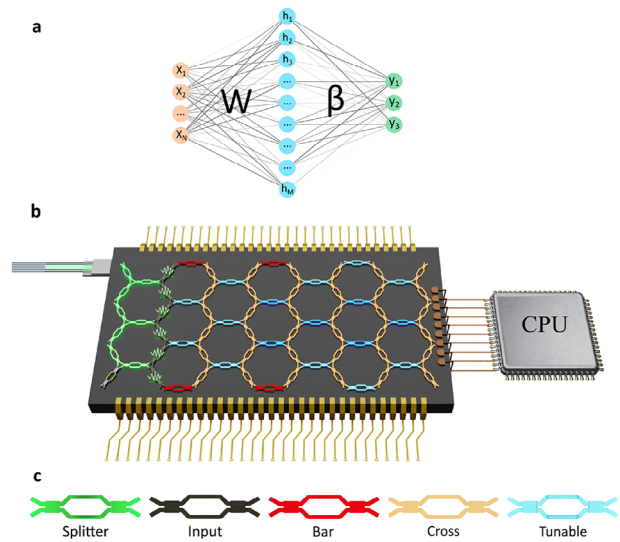


Figure 1. Programmable photonic extreme learning machines: a) Diagram of the architecture of the extreme learning machines, b) Schematic of the experimental setup for the implementation of programmable photonic extreme learning machines. The hexagonal mesh is programmed to multiply the input data by the random matrix and apply the nonlinear function using the square law of the photodetectors. The trainable weight matrix is multiplied on the CPU and, c) different states of the PUCs on the hexagonal waveguide mesh.

2.2. Programmable Photonic Circuit as an Extreme Learning Machine

We proposed the use of the Smartlight processor from iPrionics Programmable Photonics^[36] for the implementation of extreme learning machines in the photonic domain. Smartlight is a commercially available general-purpose programmable photonic integrated circuit with a hexagonal topology that can be tuned via software to create different photonic structures on the same platform. It comprises 72 programmable unit cells (balanced Mach-Zehnder interferometers) with 28 input–output optical ports. On-chip integrated photodetectors enable the conversion of the optical output power into the electronic domain through a readout layer. Further details on the characteristics of the chip and the hardware and software layers can be found in the Experimental Section.

A schematic of the programmable photonic extreme learning machine (PPELM) is shown in **Figure 1**. Light is coupled into the photonic integrated chip using an optical fiber array. The programmable unit cells (PUCs) in green are tuned to create a splitter tree that divides the light into four paths. The black PUCs, with a modulated wave on them, encode the input data into the amplitude of the light by changing their coupling ratio. Data has been previously normalized between -1 and 1 , and the sign is encoded by changing the phase term of the PUCs, which is set to 0 for positive and π for negative numbers. Additional information of the data encoding is presented in Appendix A. In our system, the maximum size of the input vector is 6, where we encode the features. The remainder of the chip is programmed to implement a random transformation of the optical field. PUCs

Table 1. Test accuracy (%) of the PPELM on three classification tasks with different number of hidden nodes.

| Nodes | 4 | 6 | 8 | 10 |
|--------------------|------------|------------|------------|------------|
| Header recognition | 93.6 ± 1.2 | 94.5 ± 2.4 | 96.2 ± 3.0 | 98.6 ± 2.5 |
| Iris | 88.5 ± 4.2 | 93.0 ± 2.8 | 95.0 ± 1.5 | 95.8 ± 1.4 |
| Banknote | 80.4 ± 8.2 | 84.0 ± 6.2 | 87.1 ± 4.7 | 90.3 ± 2.6 |

in red and gold are kept in bar and cross states respectively to avoid resonant structures inside the mesh when using a single-wavelength approach. As it will be explained later, the creation of resonators inside the mesh can be harnessed to increase the capabilities of our system when WDM is used. PUCs in blue are set as tunable, which means that their coupling ratio can be any number between $[0, 1]$ and the phase term is sampled from a uniform distribution between $[0, \pi]$. Once the input data is randomly projected, the nonlinear function is applied using the square law of the integrated photodetectors. Given the programmed structure light can be detected in up to 10 of the available photodetectors. During the experiments, we changed the state of the latest PUCs to use different numbers of outputs to measure the impact of the architecture's complexity. The measured currents were recorded using a CPU, where the addition of the bias term, training and application of the β layer was carried out.

2.3. PPELM on Classification Tasks

We employed the PPELM to solve three different classification tasks: the IRIS flower and the banknote datasets classification tasks as well as header recognition task. For all three problems, 70% of the dataset was used for training and 30% for testing. The training of the models was performed as follows: first, we sampled the coupling ratios and phase terms of the tunable elements from a uniform distribution to create the random matrix. Second, data was normalized, encoded on the corresponding PUCs, and then multiplied by the random linear transformation.

The nonlinear high-dimensional representation of the features was stored in the matrix \mathbf{H}_{train} of size $N \times N_H$, where N is the number of points in the training dataset and N_H is the number of hidden layer nodes. For each task, we repeated the experiments with 4, 6, 8, and 10 nodes. For every node count value, the random architecture was slightly modified to guide the light into the desired photodetectors and lose the minimum amount of light through non-monitored ports. Following Equation (2), the matrix \mathbf{H}_{train} was used to calculate the weights of the β layer. Once the model was trained, we encoded the test set into the PPELM, multiplied it by the same random matrix, and obtained \mathbf{H}_{test} . This matrix was multiplied by the final layer and the predicted outputs \mathbf{O} were obtained. These results were compared with the expected targets \mathbf{T} to measure the performance of the models. All trainings were repeated 40 times using different random projections. The models were also trained using ridge regression from scikit-learn with built-in cross-validation, but no noticeable improvement was observed. A summary of the obtained mean accuracies is presented in Table 1.

2.3.1. Header Recognition Classification Task

The header classification task aims at recognizing a set of 4 bits within a data stream of 6000 points. We generated a random sequence of 1s and 0s, and classified each group of four bits as 1 if they matched the desired sequence, or 0 if they did not. In our case, we chose the sequence $[1, 0, 0, 0]$ as the target. The results for the different number of nodes are shown in Figure 2a which presents a boxplot of the accuracy obtained from the 40 models that we ran. The boxes cover the range from the first to the third quartile, with a line at the median. The whiskers extend to the last data point within 1.5 times the distance between the first and the third quartile. We can see how the accuracy increases with the number of nodes. As this is a simple task, a 100% accuracy can be obtained with 8 and 10 nodes.

2.3.2. Iris Flower Dataset Classification Task

The objective of the Iris Flower dataset task is to classify a set of flower specimens into three different subspecies: setosa, versicolor, and virginica. It comprises 150 feature vectors with 4 distinct characteristics: sepal length, sepal width, petal length, and petal width. Figure 2b reveals that the PPELM was able to successfully perform this complex task with high accuracy. In particular, 2b, shows the accuracy obtained for each number of hidden nodes. This is a more complex task than the header recognition task, and we can see how the variance in the results is reduced as the number of nodes increases, with the mean accuracy converging to a value close to 97% which matches the performance of other PELM proposals with a higher total network node count.^[30,31]

2.3.3. Banknote Dataset Classification Task

The banknote dataset is a collection of 1372 instances with 4 attributes: variance, skewness, kurtosis, and entropy of the banknote images, aiming to distinguish between genuine and forged banknotes. The results for the different models are shown in Figure 2c, where accuracies over 90% are achieved in this complex task. This problem illustrates the importance of the initial random matrix. We can see that with all the hidden node sizes, there are models with high accuracies, but the variance is higher as the number of hidden nodes decreases. In the following section, we will introduce a method to find the optimal initial random matrix, thereby reducing the necessary length of the hidden layer in the PPELM. Moreover, the obtained mean accuracy is slightly lower than that presented by other PELMs. This is a consequence of the reduced number of hidden nodes. However, we will show how we can update the presented PPELM to match the accuracies of more complex photonic systems.

2.4. Evolutionary PPELM

The results obtained from the three classification tasks highlight a key challenge in extreme learning machines: the variance in achieved accuracies due to the randomness in the hidden layer

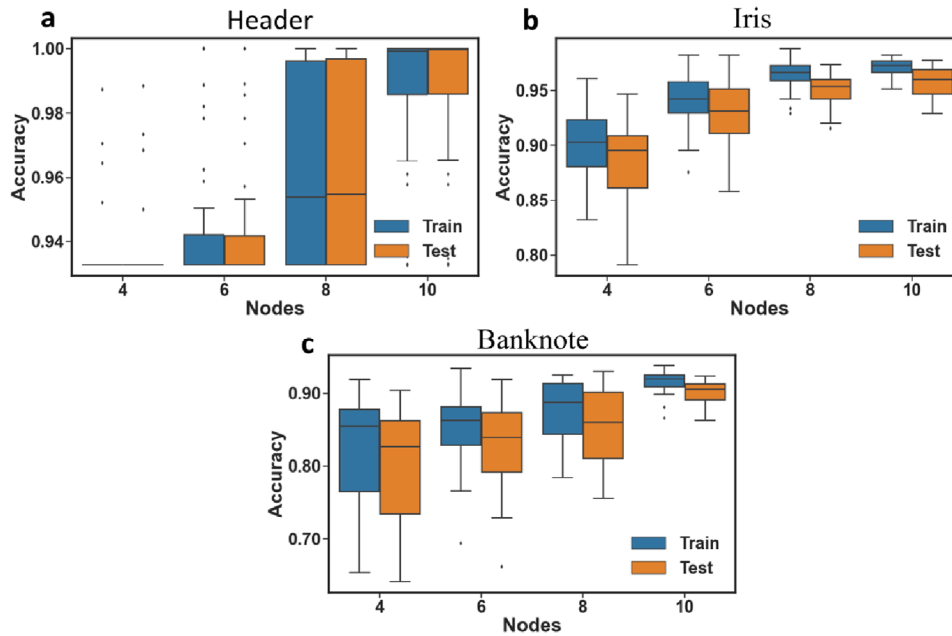


Figure 2. Accuracy of the of the PPELM in the a) Header recognition task, the b) Iris Flower and c) Banknote Authentication Datasets Classification tasks when using 4, 6, 8, and 10 hidden nodes. In blue are the results of the train and in orange of the test set for 40 different random initializations.

weight matrix. While this issue is inherent to ELM models, its consequences are particularly significant in photonic systems. One option to mitigate the variance in model accuracy is to increase the number of hidden neurons, which in turn increases complexity, power consumption, and accumulated losses in photonic ELMs, thereby reducing their scalability. An alternative to increase the size of the system is to use a differential evolution algorithm (DE-PPELM) to find the optimal initialization of the random linear transformation.^[37] The programmability of our hexagonal mesh allows for the direct application of this approach into our system by optimizing the applied phases on the tunable PUCs.

To apply the differential evolution algorithm, at first, a set of N phase vectors is chosen as the initial candidates:

$$\Theta_G = \{\theta_{i,G} | i = 1, \dots, N\} \quad (3)$$

where $\theta_{i,G} = [\theta_0, \theta_1, \dots, \theta_{N_{\text{T PUCs}}}]$ corresponds to the i th array with the phase value for all the tunable PUCs during generation G . The total number of vectors N is referred to as the population size. For each of these vectors, the performance of the PPELM is evaluated using a cost function CF . In our case, the cost function CF is 1 minus the accuracy of the model on a validation set. The validation set is an extra division of the dataset (train - val - test) that allows for monitoring the algorithm's performance during training while leaving the test set untouched. The second step of the algorithm is the mutation, where three of the current candidates are combined using a control factor F as follows:

$$v_{i,G+1} = \theta_{a,G} + F(\theta_{b,G} - \theta_{c,G}) \quad (4)$$

where a , b and c are independent random indices. By selecting elements from the candidate and mutant sets, a new set of trial

vectors $u_{i,G+1}$ is built. These elements are chosen by sampling a random number $b \in [0, 1]$. If b is lower than a crossover factor CR then the vector is picked from the mutation set V_{G+1} , otherwise the vector is chosen from the set of candidates Θ_G . The performance of the model with the trial vectors is evaluated and compared to that achieved by the previous candidates. If a trial vector $u_{j,G+1}$ presents a better CF than the candidate $\theta_{j,G+1}$ then the trial vector replaces the candidate in the new generation. Otherwise, the new generation remains unchanged. If the difference between the two cost functions is less than 0.001, then we compare the norm of the trainable matrix β for both models and choose the solution with the lower norm. This is done because a lower norm in the second layer is related to a better generalization of the model.^[38]

We evaluate the DE-PPELM on the Iris Flower and the Banknote Authentication dataset classification tasks. For this section we avoid the header recognition dataset as it presented very high accuracies for all the for all network hidden node counts investigated. As previously explained, for these experiments we divided the dataset into training, test and validation sets. We used 70 % of the dataset points for training, 15% for validation and 15% for testing. For each dataset, we initialized the random matrix using the differential evolution algorithm for all previous hidden node sizes. We used a population size of 10 for both tasks, a control factor $F = 1$ and a crossover term $CR = 0.5$. The DE-PPELM was trained during 35 iterations for all models. As the dataset sizes are small, especially the Iris Flower dataset, it is highly likely that high accuracies are achieved during the first iteration (which corresponds to 10 evaluations of the PPELM). To show how the algorithm can improve the accuracy of ill-conditioned random layers, we chose suboptimal candidates as a starting point during the first iteration.

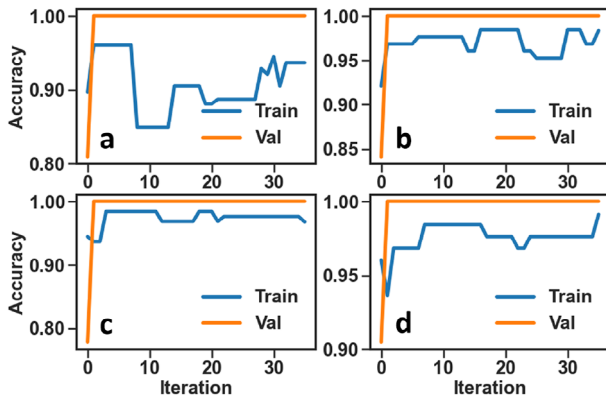


Figure 3. Evolution of the training and validation accuracy during 35 iterations of the DE-PPELM when using a) 4, b) 6, c) 8, and d) 10 hidden nodes in the Iris Flower Dataset classification task.

The evolution of the accuracies during the DE-PPELM training for the Iris flower dataset classification task is depicted in **Figure 3**. We can observe how the validation curve rapidly reaches a 100% accuracy, primarily due to the small size of the validation set. However, the algorithm continues to learn by attempting to reduce the norm of the output layer β . Fluctuations in the training set are influenced by two factors. First, we aim to optimize the validation accuracy rather than the test accuracy, acting as a form of regularization for the training curve. Second, the small dataset size introduces higher divergences between the validation and training curves. For the Banknote dataset classification tasks, the evolution of the accuracies is presented in **Figure 4**. Here, we notice that, except for the 10-node model, the algorithm requires more steps to converge compared to the Iris task as a consequence of the larger dataset size. Additionally, in this case, the validation and training curves are more similar during the training of the algorithm.

The final results after 35 iterations obtained on the test set are presented in **Table 2** for both classification tasks. For the Iris flower dataset classification task, we observe that for all hidden node sizes, the accuracy achieved on the test set is equal to or greater than the maximum accuracy obtained with the best model

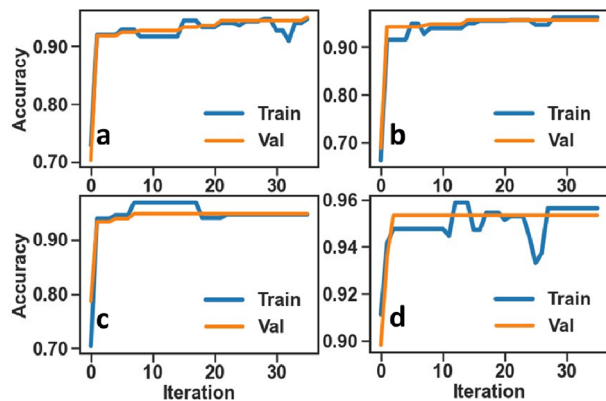


Figure 4. Evolution of the training and validation accuracy during 35 iterations of the DE-PPELM when using a) 4 b) 6 c) 8 and d) 10 hidden nodes in the Banknote authentication dataset classification task.

Table 2. Test accuracy (%) of the DE-PPELM on two classification tasks with different number of hidden nodes.

| Hidden Nodes | 4 | 6 | 8 | 10 |
|--------------|------|------|------|------|
| Iris | 95.5 | 97.0 | 97.0 | 98.5 |
| Banknote | 91.5 | 91.5 | 92.0 | 93.0 |

in the first experiment shown in **Figure 2**. Regarding the Banknote task, we achieved accuracies exceeding 91% regardless of the number of hidden nodes. These results surpass the best results obtained during the first experiments in which a random hidden node weight matrix was set, hence demonstrating the pivotal contribution of the DE-PPELM algorithm for performance optimisation. Finally, it is essential to highlight how we were able to achieve high accuracies even with a reduced number of total network node count. This capability enables training highly complex computational tasks on a more compact photonic hardware than that required if the initial layer was directly sampled from a random distribution.

2.5. WDM Based Ensemble for Increased Accuracy

In the previous section, we introduced an optimization algorithm to mitigate the impact of the random initialization of the internal layer of the ELM. Here, we propose an alternative approach to increase the accuracy of these systems by combining the predictions of a set of low accuracy models leveraging a widely studied technique called ensemble learning.^[39] In classical ELM approaches,^[40,41] ensembling has demonstrated its ability to increase the efficacy in boosting accuracy on test sets while reducing overfitting. In our approach, we suggest taking advantage of the reconfigurability of our programmable photonic processor to train different models simultaneously in the photonic domain. Subsequently, we combine the results of the trained models to generate the final classification prediction. The proposed architecture uses WDM and is shown in **Figure 5**. Using WDM to increase capacity by performing parallel computation has been proposed in the literature for photonic deep learning accelerators.^[42–44]

The architecture works as follows: Initially, different wavelengths are multiplexed into a single-mode fiber (SMF) and connected into the photonic chip using the same procedure as in our previous approaches. Second, both the training and testing sets are encoded on each of the wavelengths. Then, the random matrix architecture is modified by transitioning some of the diagonal PUCs from a cross to a tunable state. This adjustment facilitates the generation of resonant structures within the chip, thereby making the transfer function of the system wavelength-dependent. Following the random linear transformation, each output is demultiplexed, resulting in N hidden output states corresponding to each of the N input wavelengths. Subsequently, N models are trained on a computer by computing the output matrix β . Each model undergoes training using a the same partition of the training and testing sets. Finally, the ensemble step involves combining the predictions of each model into a single array. This combined array is then utilized to compute the final prediction by training a final layer using Equation (2).

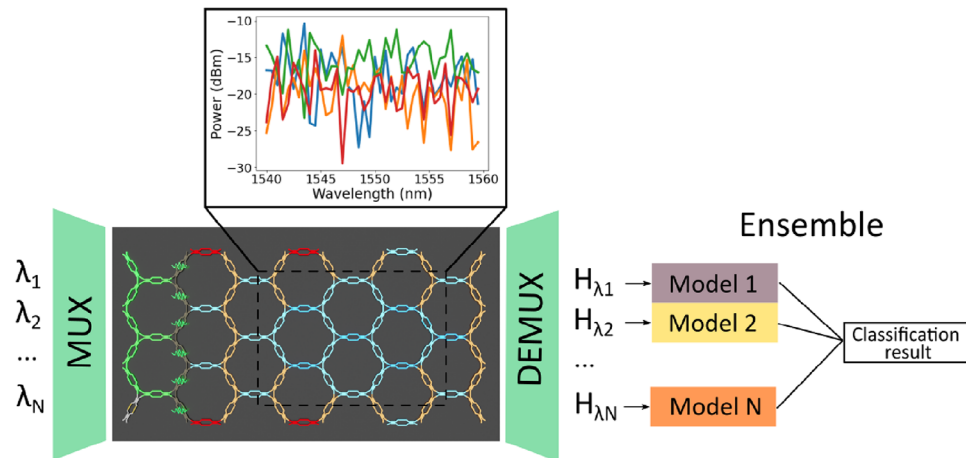


Figure 5. Ensembled extreme learning machines using WDM.

Table 3. Accuracy (%) of the WDM-Ensemble on the Iris Flower classification task for different number of wavelengths.

| # λ | Mean Train | Mean Test | Ensemble Train | Ensemble Test |
|------------------|------------|-----------|----------------|---------------|
| 4 (Parallel WDM) | 95.1 | 93.6 | 98.2 | 97.3 |
| 5 | 95.7 | 93.4 | 96.4 | 96.4 |
| 10 | 96.4 | 95.5 | 99.1 | 97.3 |
| 15 | 95.8 | 94.0 | 1.0 | 98.1 |
| 20 | 95.3 | 93.3 | 1.0 | 98.1 |

Table 4. Accuracy (%) of the WDM-Ensemble on the Banknote classification task for different number of wavelengths.

| # λ | Mean Train | Mean Test | Ensemble Train | Ensemble Test |
|------------------|------------|-----------|----------------|---------------|
| 4 (Parallel WDM) | 91.6 | 90.6 | 94.8 | 93.9 |
| 5 | 91.3 | 89.4 | 93.1 | 92.3 |
| 10 | 88.8 | 87.4 | 95.3 | 93.1 |
| 15 | 89.6 | 88.2 | 97.9 | 97.5 |
| 20 | 89.1 | 87.8 | 99.1 | 99.1 |

As a proof of concept, we initially performed the experiment using a 4-channel CWDM operating in the range of 1531–1591 nm, with a channel spacing of 20 nm. The four wavelengths were multiplexed and fed into the Smartlight processor. At one of the system's outputs, we used a DEMUX to separate the wavelengths and measured their optical power with an external power meter. We recorded the results for both the Iris flower and Banknote classification tasks. Subsequently, we repeated the same process at the remaining outputs. Throughout the experiment, the state of the photonic processor was kept constant for each classification task. The results are summarized in Tables 3 and 4 in the $\lambda = 4$ (parallel) row, showing improved performance compared to the single-wavelength approach.

In order to increase the number of available wavelengths, we substituted the MUX and DEMUX by a sequential process. We used a tunable laser and set one wavelength. We conducted the

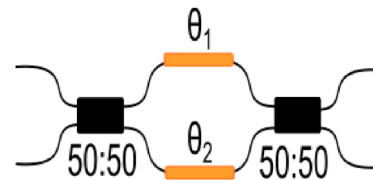


Figure A1. Programmable unit cell.

vector matrix multiplication and stored the results. This process was repeated with 40 wavelengths ranging from 1540 to 1560 nm, with a spacing of 0.5 nm between them. In Figure 5, a graph with the resonant response of the system is illustrated. Each of the four represented colors in the inset in Figure 5 corresponds to a different output of the mesh. The recorded results were used to derive the output matrix of each model, and the predicted outputs were aggregated for the final classification results. The results with varying numbers of used wavelengths are presented in Tables 3 and 4, where we compare the mean accuracy of the independent models with the ensemble results ($\lambda = 5$ –20). Across all cases and classification tasks, the ensemble results consistently outperformed the mean accuracies during both training and testing. As in the evolutionary PPELM, the Iris flower classification task reaches its maximum accuracy with a low number of wavelengths as a consequence of the simplicity of the task. For the banknote classification task, it is important to highlight that we were able to obtain higher accuracies than the maximum level achieved with the random PPELM and with the evolutionary PPELM, with attained accuracies higher than 99% when 25 or more wavelengths are used.

The results are slightly better in the parallel case compared to those obtained using the sequential approach. This is primarily due to the random initialization of the resonant structures, which we have seen that greatly impacts the performance of the PELM. This highlights the potential of combining WDM with the evolutionary PELM to enhance model performance while minimizing the number of nodes required.

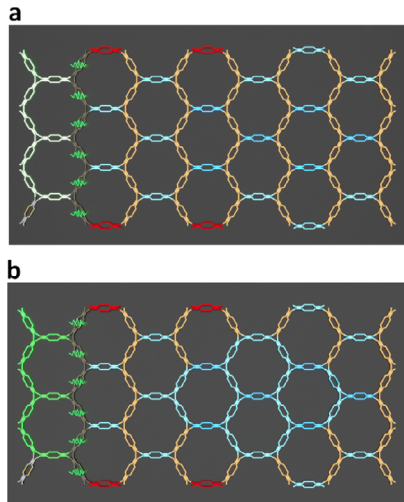


Figure B1. Comparison of matrix configurations, **a** feedforward and **b** resonant.

3. Discussion and Conclusion

We have demonstrated the implementation of photonic extreme learning machines (ELMs) using an integrated photonic programmable circuit. While previous photonic approaches relied on free-space propagation of light or application-specific photonic integrated circuits,^[27,29–31] our solution is based on a general-purpose programmable photonic integrated platform. To implement the programmable photonic ELMs, we first defined a splitter tree to divide the light into the number of dimensions of the input vector of the PPELM. Programming the splitter on-chip allowed us to maintain coherence inside the mesh, enabling us to exploit the complex-valued nature of the photonic devices in the integrated system. The normalized input vector was encoded in the amplitude of the light by adjusting the coupling of an array of PUCs, while the sign was encoded in the phase by simultaneously tuning both actuators of the PUC.

We incorporated the random layer of the ELMs by randomly selecting the coupling coefficient and phase terms of the tunable PUCs. We configured the state of the remaining PUCs to avoid resonant structures and ensure unidirectional light propagation. The nonlinear activation function was implemented using the square law of the integrated photodetectors. Finally, the output layer was calculated and digitally implemented using an analytic formula, similar to previous approaches.

In an initial approach, we experimentally solved three classification tasks using an in-house created header recognition dataset, as well as the Iris flower and Banknote authentication datasets. We repeated all the experiments 20 times using network architectures with 4, 6, 8, and 10 hidden layer nodes. We showed how increasing the number of hidden nodes reduces the variability of the classification results that arises as a consequence of the random nature of the hidden layer. For the case of 10 hidden nodes we obtained mean accuracies of 98.6 ± 2.5 , 95.8 ± 1.4 and 90.3 ± 2.6 , for the Iris Flower and Banknote dataset classification tasks, respectively. Increasing the number of the number of hidden nodes may not always be the optimal solution as the scalability of photonic devices is limited and precision could be

degraded.^[45] As an approximation, the dynamic range at the outputs is:

$$P_{in} - IL_{coupling} - N * IL_{PUC} - NF_{PD} \quad (5)$$

where P_{in} is the input power, $IL_{coupling}$ is the coupling insertion loss, N is the minimum number of PUCs from the input to the photodetectors, IL_{PUC} are the insertion losses of the PUCs and NF_{PD} is the noise floor of the photodetectors. Assuming $IL_{coupling} = 6$ dB, $IL_{PUC} = 0.5$ dB, $NF_{PD} = -35$ dBm, $N = 15$ and $P_{in} = 13$ dBm the dynamic range is 34.5 dB.

As an alternative, we propose two solutions to enhance the classification accuracy of PELMs with a low number of hidden nodes. The first approach leverages the programmability of our system. We apply a differential evolution (DE) algorithm to optimize the hidden layer of the PELM. Our results demonstrate that this technique achieves accuracies equal to or better than the best values obtained for all network node sizes investigated, without the DE algorithm. Additionally, we find that fewer hidden nodes can be utilized with only a slight reduction in model accuracy. The main drawback of this solution is the increased computational resources and time required during the training of the evolutionary algorithm. Specifically, each iteration requires updating the state of the PUCs, which, in the case of thermo-optic phase shifters, is limited to microseconds. However, once the optimal random layer is determined and the model is trained, the weights remain fixed during inference, avoiding any additional overhead compared to the standard PPELM.

The second enhancement involves using an ensemble of models enabled by WDM techniques. By manipulating part of the hexagonal mesh, we enable the creation of resonant structures with a wavelength-dependent transfer function. Multiple vector-matrix multiplications can then be performed simultaneously applying different wavelengths into the system. The results from each wavelength model are then ensembled to derive a final classification prediction. First, we showed a parallel implementation using a MUX/DEMUX system with 4 channels and then we extended the number of available wavelengths using a sequential approach. This method yields promising results, surpassing those obtained in previous experiments. For instance, we achieved a 98.1% accuracy on the Iris classification task using 15 wavelengths and 8 outputs, and a 99.1% accuracy on the Banknote classification task using 20 wavelengths and 8 outputs. When compared to other experimental proposals, a bulky system based on frequency multiplexing,^[30] achieved a 97.7% accuracy on the Iris classification task and 99.5% on the Banknote classification task using 31 hidden nodes. Using an integrated approach based on microring resonators,^[31] accuracies of 97.2% and 97.92% were achieved for the Iris and Banknote classification tasks, respectively, with 18 physical nodes. An additional 126 virtual nodes were utilized to increase the accuracy of the Banknote task to 99%.

In conclusion, we have shown how ELMs can benefit from the advantages of the photonic systems using an integrated programmable processor. Thanks to the latter's programmability we were able to test different random layers and proposed two methods to improve the performance of the models using an evolutionary algorithm and a WDM approach.

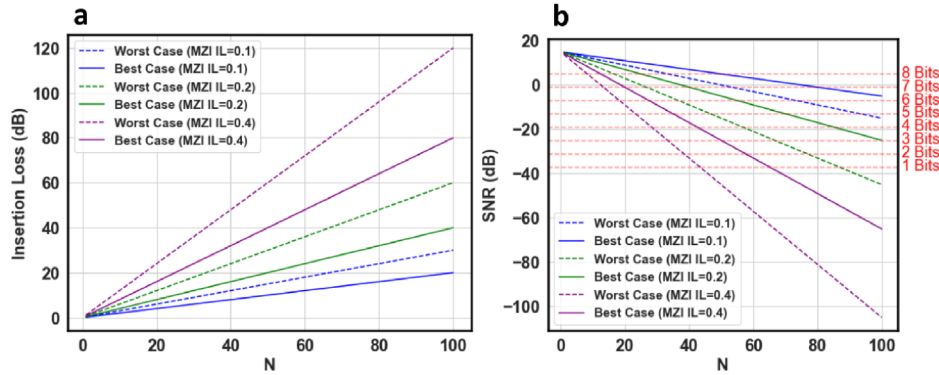


Figure B2. a) Insertion losses and b) bit precision as a function of the matrix size and MZI losses.

4. Experimental Section

The Smartlight processor comprises a photonic chip manufactured in a standard 220-nm Silicon Photonics (SiPh) platform. The system integrates thermo-optical phase shifters with under-etched waveguides. Light input and output are facilitated using fiber arrays with edge coupling featuring a 127 μm pitch. On-chip photodetection utilizes Germanium on silicon photodetectors. On-chip electrical pads are wirebonded to a PCB to enable the control of the phase actuators and the monitoring of the photodetectors. The photonic processor is controlled via software using the Smartlight python package provided by iPronics, allowing for full control of the state of each of the actuators.

For the single-wavelength and evolutionary PELM experiments, we utilized the ORTEL 10481S-FA laser with a fixed optical output power of 9.8 dBm operating at 1557 nm. For the 4 channel WDM experiment, we used a CWDM from Accelink operating in the range of 1531–1591 nm, with a channel spacing of 20 nm. The wavelengths at 1531 and 1551 nm correspond to two IQS-2400 WDM laser sources from EXFO, while the wavelengths at 1571 and 1591 nm correspond to two TUNICS T100S-HP/CLU tunable lasers from Yenista. For the WDM sequential experiments, we employed the tunable laser TUNICS T100S-HP/CLU with a wavelength range from 1500 to 1680 nm and an output optical power of 13 dBm.

Training of the output layer of the PPELM was conducted on an Intel Core i5-10400H 2.6 GHz CPU with 16 GB of RAM.

Appendix A: Data Encoding in the Smartlight Processor

Each feature is encoded on a programmable unit cell by changing the amplitude and phase of the optical signal. An schematic of the building block is presented in Figure A1 and its transfer function is the following:

$$ie^{i\frac{\theta_1+\theta_2}{2}} \begin{pmatrix} \sin\left(\frac{\theta_1-\theta_2}{2}\right) & \cos\left(\frac{\theta_1-\theta_2}{2}\right) \\ \cos\left(\frac{\theta_1-\theta_2}{2}\right) & -\sin\left(\frac{\theta_1-\theta_2}{2}\right) \end{pmatrix} \quad (\text{A1})$$

The first step is to program the amplitude of the light. To do so, we need to know that the information will input the upper arm of the PUC and output the bottom arm, and thus, the output amplitude is:

$$A_{out} = A_{in} \cos\left(\frac{\theta_1 - \theta_2}{2}\right) \quad (\text{A2})$$

Without losing generality, we can assume $A_{in} = 1$ and $A_{out} = |f|$, where $| \cdot |$ is the modulus and f is the feature value normalized between -1 and 1. Then:

$$\Delta\theta = \theta_1 - \theta_2 = 2 * \arccos(|f|) \quad (\text{A3})$$

Once the phase difference required for a certain amplitude is defined, then we need to include the output phase $angle(f)$ using the exponential term in A1:

$$\theta_1 + \theta_2 = 2 * angle(f) \quad (\text{A4})$$

Combining A3 and A4 we get the require phase values to encode a certain amplitude and phase:

$$\theta_1 = angle(f) + \arccos(|f|) \quad (\text{A5})$$

$$\theta_2 = angle(f) - \arccos(|f|) \quad (\text{A6})$$

Appendix B: Scalability of the Programmable PELM

The scalability of the programmable photonic extreme learning machine (PELM) is primarily determined by the insertion losses of the basic tunable elements, implemented as Mach-Zehnder interferometers (MZIs). These losses can be divided into two main components.

First, the splitter tree, where MZIs are tuned to equally distribute light among the input paths. The insertion losses in this stage are proportional to the number of input features (N_{IN}) and correspond to $N_{IN} \cdot IL_{MZI}$, where IL_{MZI} represents the insertion loss per MZI.

The second component involves the random matrix multiplication stage. Assuming a worst-case scenario where the number of inputs equals the number of outputs, all inputs must combine within the mesh. This requires light to traverse $2N_{IN}$ MZIs, leading to insertion losses of $2N_{IN} \cdot IL_{MZI}$.

To mitigate these losses, resonant structures can be introduced, enabling faster interaction between light paths compared to feedforward meshes. This approach reduces the required number of MZIs. As illustrated in Figure 2, the worst-case feedforward configuration (Figure B1a) can be improved using resonant structures (Figure B1b), where light traverses only N_{IN} MZIs in the best-case scenario.

Based on these scenarios, the insertion losses of the system were calculated as a function of the matrix size ($N \times N$) and are presented in Figure B2a. For an MZI insertion loss of 0.1 dB^[46] bit precisions of approximately 5 bits or more can be achieved, as shown in Figure B2b, where we assumed an input light power of 15 dBm. By incorporating additional degrees of freedom, such as time multiplexing used in reservoir computing or other photonic neural networks,^[47] the node count can be scaled to the order of thousands.

Acknowledgements

The authors would like to acknowledge Dr. Joshua Robertson and Dr. Giovanni Donati from the University of Strathclyde for their valuable discussions on photonic extreme learning machines and their contribution

of valuable ideas to this work. This work was supported by the H2020-ICT2019-2 Neoteric 871330 project, the European Research Council (ERC) Advanced Grant programme under grant agreement No. 101097092 (AN-BIT), the ERC Starting Grant programme under grant agreement No. 101076175 (LS-Photonics Project), the EUR2022-134023 grant funded by CIN/AEI/10.13039/501100011033 and the European Union (NextGenerationEU/ PRTR), and the UKRI Turing AI Acceleration Fellowship Programme 'PHOTON-AI' (EP/V025198/1).

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

extreme learning machine, machine learning, programmable photonics, photonic integrated circuit, photonic neural networks

Received: June 11, 2024

Revised: January 9, 2025

Published online:

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, in *Advances in neural information processing systems*, vol. 30, Curran Associates, Inc, Long Beach, CA, USA, **2017**.
- [2] J. Ho, A. Jain, P. Abbeel, in *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*. Curran Associates Inc, Red Hook, NY, USA, ISBN 9781713829546 **2020**.
- [3] P. Stark, F. Horst, R. Dangel, J. Weiss, B. J. Offrein, *Nanophotonics* **2020**, 9, 4221.
- [4] A. N. Tait, T. F. de Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, P. R. Prucnal, *Sci. Rep.* **2017**, 7, 7430.
- [5] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. P. Pernice, H. Bhaskaran, C. D. Wright, P. R. Prucnal, *Nat. Photonics* **2021**, 15, 102.
- [6] K. Liao, T. Dai, Q. Yan, X. Hu, Q. Gong, *ACS Photonics* **2023**, 10, 2001.
- [7] P. L. McMahon, *Nat. Rev. Phys.* **2023**, 5, 717.
- [8] H. Zhang, M. Gu, X. D. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. H. Yung, Y. Z. Shi, F. K. Muhammad, G. Q. Lo, X. S. Luo, B. Dong, D. L. Kwong, L. C. Kwek, A. Q. Liu, *Nat. Commun.* **2021**, 12, 457.
- [9] D. Brunner, A. Marandi, W. Bogaerts, A. Ozcan, *Nanophotonics* **2020**, 9, 4053.
- [10] D. A. B. Miller, *J. Lightwave Technol.* **2017**, 35, 346.
- [11] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, A. Ozcan, *Science (New York, N.Y.)* **2018**, 361, 1004.
- [12] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, Q. Dai, *Nat. Photonics* **2021**, 15, 367.
- [13] W. Zhang, C. Huang, H.-T. Peng, S. Bilodeau, A. Jha, E. Blow, T. F. de Lima, B. J. Shastri, P. Prucnal, *Optica* **2022**, 9, 579.
- [14] S. Ohno, R. Tang, K. Toprasertpong, S. Takagi, M. Takenaka, *ACS Photonics* **2022**, 9, 2614.
- [15] S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, D. Englund **2022** *arXiv:2208.01623*.
- [16] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, M. Soljačić, *Nat. Photonics* **2017**, 11, 441.
- [17] T. W. Hughes, M. Minkov, Y. Shi, S. Fan, *Optica* **2018**, 5, 864.
- [18] S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. D. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti, A. Melloni, S. Fan, O. Solgaard, D. A. B. Miller, *Science* **2023**, 380, 398.
- [19] J. Robertson, P. Kirkland, G. Di Caterina, A. Hurtado, *Neurom. Comput. Eng.* **2024**, 4, 14010.
- [20] T. Inagaki, K. Inaba, T. Leleu, T. Honjo, T. Ikuta, K. Enbutsu, T. Umeki, R. Kasahara, K. Aihara, H. Takesue, *Nat. Commun.* **2021**, 12, 2325.
- [21] M. Hejda, E. A. Doris, S. Bilodeau, J. Robertson, D. Owen-Newns, B. J. Shastri, P. R. Prucnal, A. Hurtado, *Neurom. Comput. Eng.* **2024**.
- [22] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, I. Fischer, *Nat. Commun.* **2011**, 2, 468.
- [23] M. Nakajima, K. Tanaka, T. Hashimoto, *Commun. Phys.* **2021**, 4, 20.
- [24] Y.-W. Shen, R.-Q. Li, G.-T. Liu, J. Yu, X. He, L. Yi, C. Wang, *Optica* **2023**, 10, 1745.
- [25] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, **2004**, pp. 985–990.
- [26] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, *Neurocomputing* **2006**, 70, 489.
- [27] D. Pierangeli, G. Marcucci, C. Conti, *Photon. Res.* **2021**, 9, 1446.
- [28] C. M. Valensise, I. Grecco, D. Pierangeli, C. Conti, *Photon. Res.* **2022**, 10, 2846.
- [29] A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Drémeau, S. Gigan, F. Krzakala, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Press, Shanghai, China, **2016**, pp. 6215–6219.
- [30] A. Lupo, L. Butschek, S. Massar, *Opt. Express* **2021**, 29, 28257.
- [31] S. Biasi, R. Franchi, L. Cerini, L. Pavesi, *APL Photonics* **2023**, 8, 96105.
- [32] D. Pérez, I. Gasulla, J. Capmany, *Opt. Express* **2018**, 26, 27265.
- [33] W. Bogaerts, D. Pérez, J. Capmany, D. A. B. Miller, J. Poon, D. Englund, F. Morichetti, A. Melloni, *Nature* **2020**, 586, 207.
- [34] L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, A. J. Lowery, *Optica* **2015**, 2, 854.
- [35] H. Zhou, Y. Zhao, X. Wang, D. Gao, J. Dong, X. Zhang, *ACS Photonics* **2020**, 7, 792.
- [36] D. Pérez-López, A. Gutierrez, D. Sánchez, A. López-Hernández, M. Gutierrez, E. Sánchez-Gomáriz, J. Fernández, A. Cruz, A. Quirós, Z. Xie, J. Benitez, N. Bekesi, A. Santomé, D. Pérez-Galacho, P. DasMahapatra, A. Macho, J. Capmany, *Nat. Commun.* **2024** 15:1 **2024**, 15, 1.
- [37] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, G.-B. Huang, *Pattern Recognit.* **2005**, 38, 1759.
- [38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *J. Mach. Learn. Res.* **2014**, 15, 1929.
- [39] I. D. Mienye, Y. Sun, *IEEE Access* **2022**, 10, 99129.
- [40] A. O. M. Abuassba, D. Zhang, X. Luo, A. Shaheryar, H. Ali, *Comput. Intell. Neurosci.* **2017**, 2017, 3405463.
- [41] N. Liu, H. Wang, *IEEE Signal Process Lett.* **2010**, 17, 754.
- [42] B. Dong, S. Aggarwal, W. Zhou, U. E. Ali, N. Farmakidis, J. S. Lee, Y. He, X. Li, D.-L. Kwong, C. D. Wright, W. H. P. Pernice, H. Bhaskaran, *Nat. Photonics* **2023**, 17, 1080.

- [43] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, H. Bhaskaran, *Nature* **2021**, 589, 52.
- [44] S. Ou, A. Sludds, R. Hamerly, K. Zhang, H. Feng, E. Zhong, *arXiv preprint arXiv:2401.18050* **2024**, 1–24.
- [45] M. A. Al-Qadasi, L. Chrostowski, B. J. Shastri, S. Shekhar, *APL Photonics* **2022**, 7, 20902.
- [46] N. C. Harris, D. Bunandar, A. Joshi, A. Basumallik, R. Turner, in *2022 IEEE Hot Chips 34 Symposium (HCS)*. **2022**, pp. 1–26.
- [47] Y. Bai, X. Xu, M. Tan, Y. Sun, Y. Li, J. Wu, R. Morandotti, A. Mitchell, K. Xu, D. J. Moss **2023**, 12, 795.