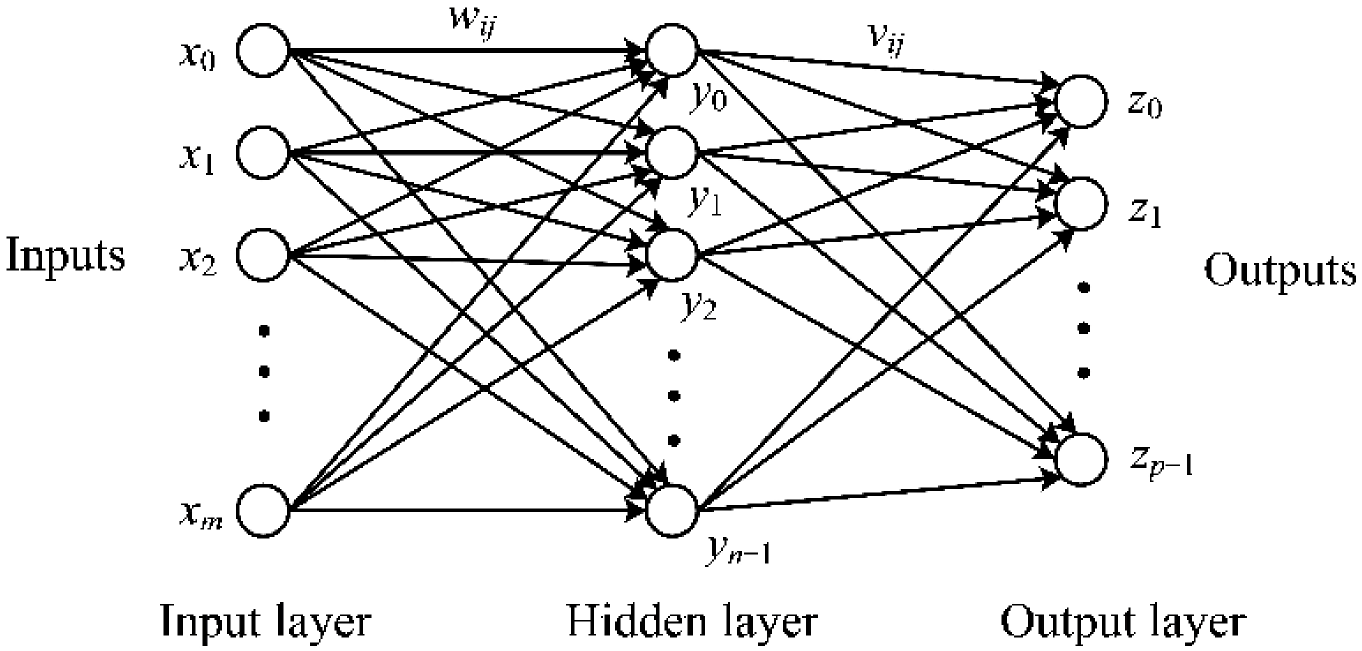
# Photonic Extreme Learning Machines

Ilker Oguz

# Outline

- 1- Multilayer Perceptron vs ELM vs Reservoir Computing
- 2- Photonic ELMs
- 3- Programming Photonic ELMs

# Multilayer Perceptron



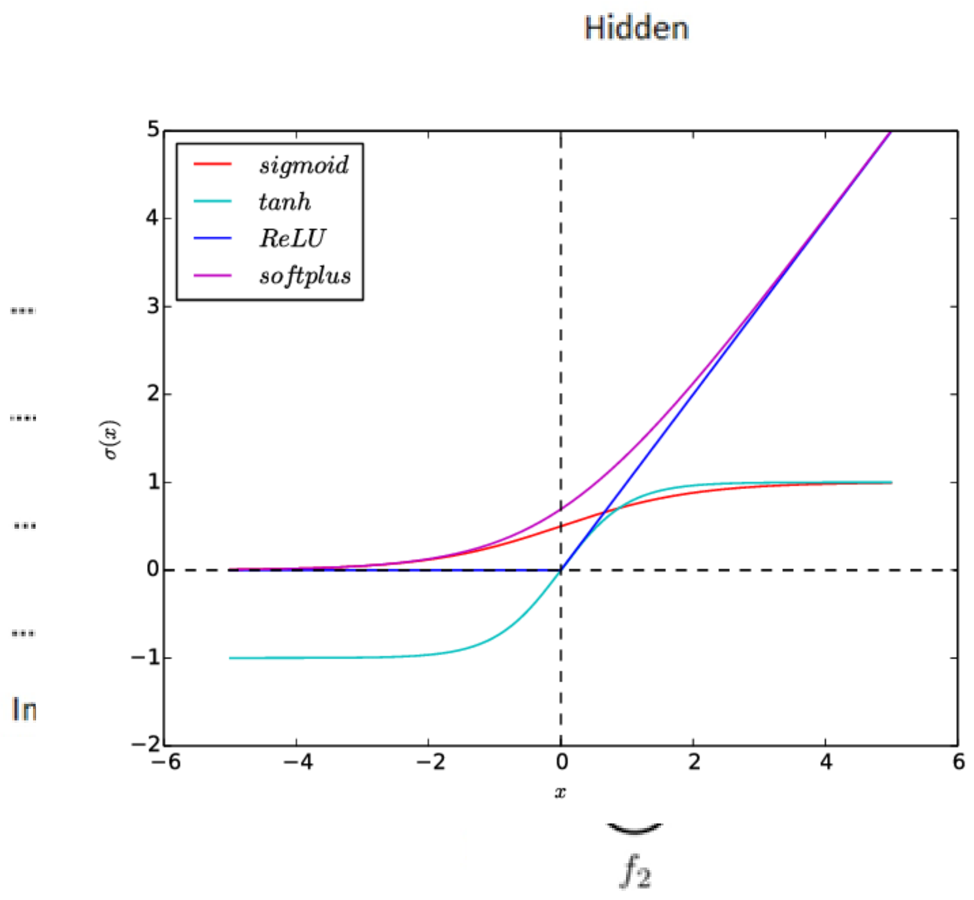

- One of the simplest ways to build neural networks.
- Each neuron applies a nonlinear function to its activation.
- Contains at least a hidden layer.
- All its weights can be trained with the backpropagation algorithm.

# Multilayer Perceptron

- During forward propagation, each neuron value is calculated as $x_j^{(n)} = f^{(n)}(\sum_i^{M_{n-1}} W_{ij}^{(n)} x_i^{(n-1)} + \mathrm{b}_j)$.

- During training, the gradient of the output error is calculated w.r.t. weights, and weights are updated accordingly.

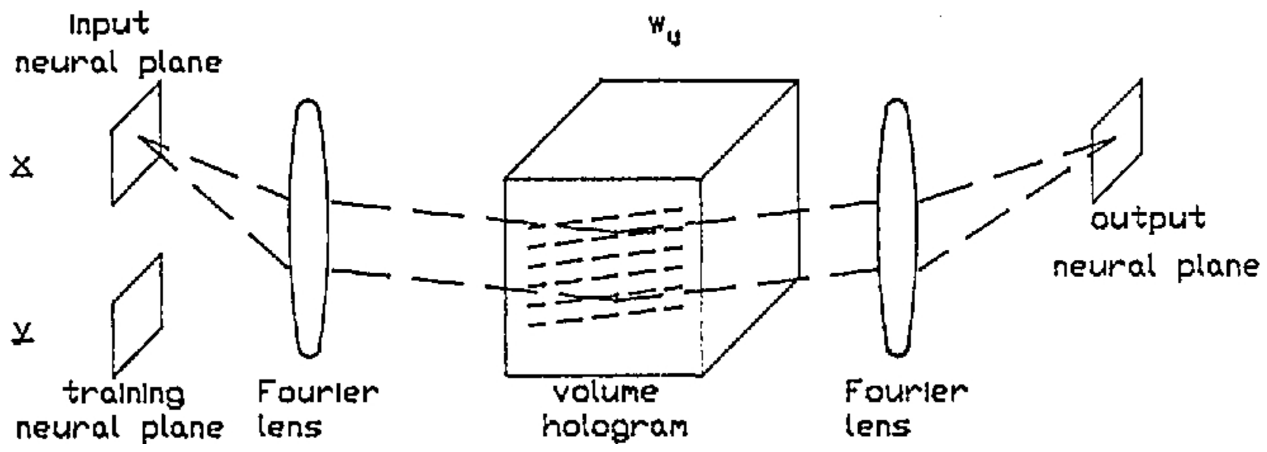
$$w_{t+1} = w_t - \alpha \frac{\partial E(X, w_t)}{\partial w}$$

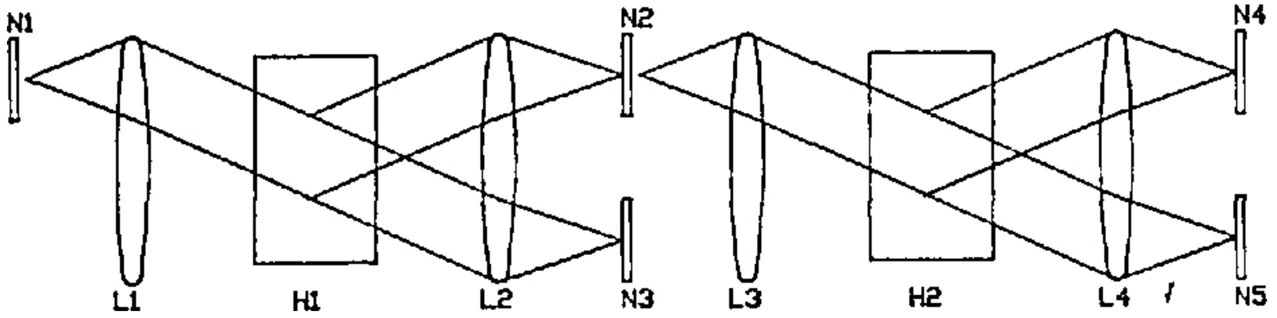


# Multilayer Perceptron

- Volume holograms stored by photorefractive crystals are trained with the dataset optically.
- Nonlinear activation can be obtained by thresholding ( for instance at N2).

$$y_i = f\left(\sum_j w_{ij} x_j\right)$$

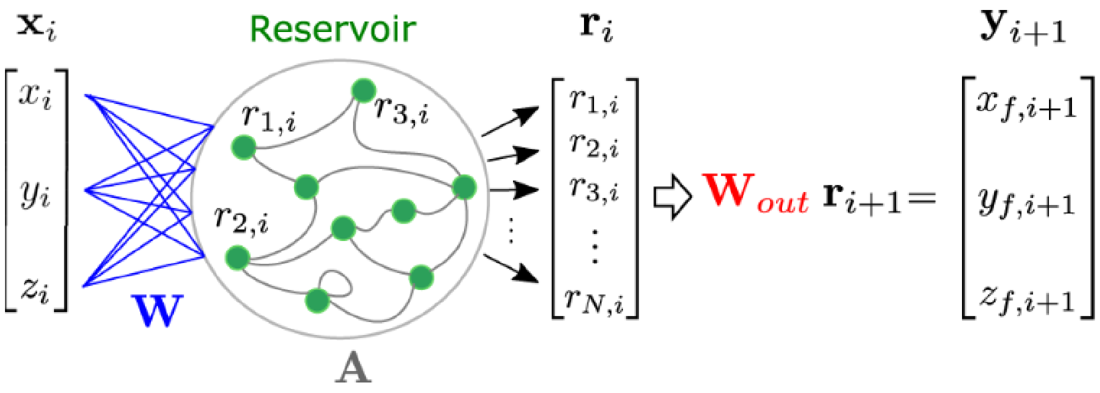



Demetri Psaltis, David Brady, and Kelvin Wagner, "Adaptive optical networks using photorefractive crystals," Appl. Opt. 27, 1752-1759 (1988)

# Reservoir Computing

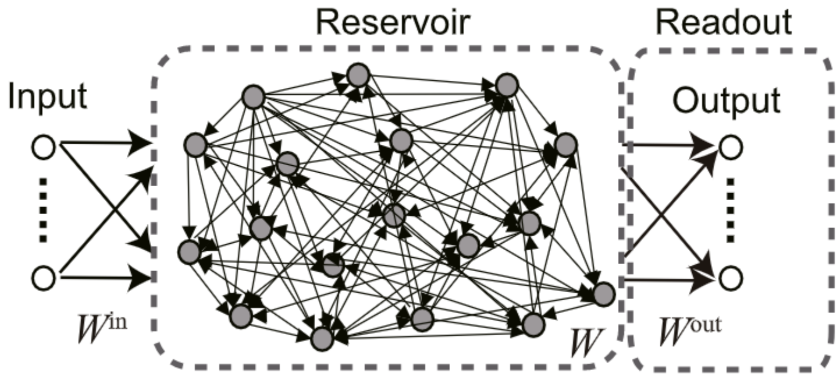
- Consists of high-dimensional, fixed, non-linear connections for transforming data, only output weights are optimized.
- The most common definition of RC requires reservoir to have memory, so that inputs of different time steps interact.




$$\mathbf{r}_{i+1} = (1-\gamma)\mathbf{r}_i + \gamma f\left(\mathbf{A}\mathbf{r}_i + \mathbf{W}\mathbf{X}_i + \mathbf{b}\right)$$

# Reservoir Computing

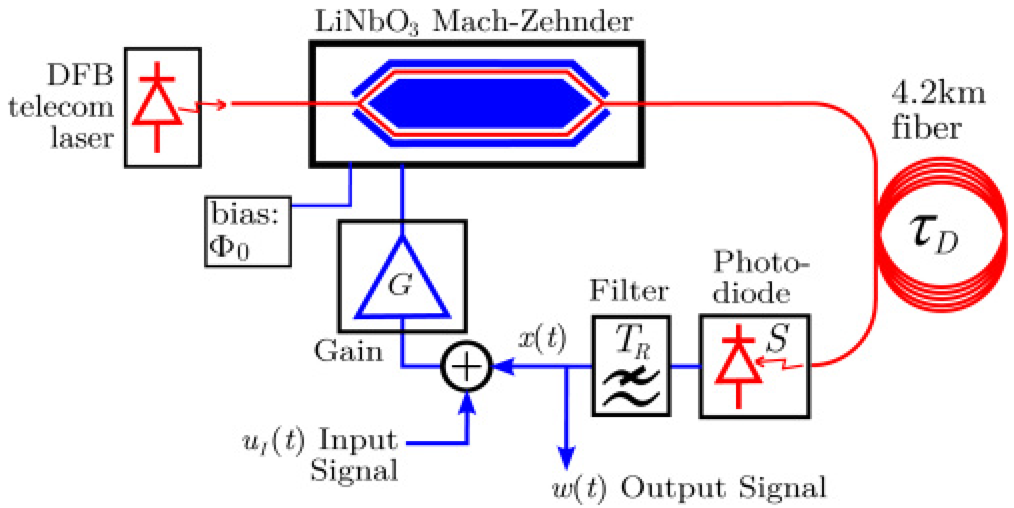
- Especially useful for fast operation and low training cost
- Can be realized by **high dimensional** and **nonlinear** physical systems.

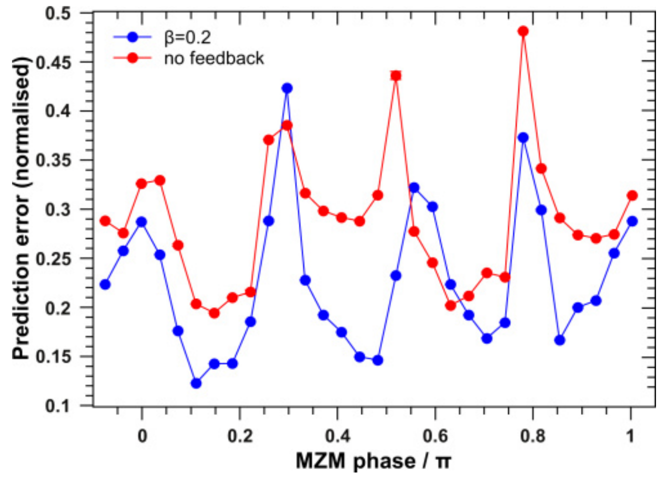



(a) Conventional RC

# Reservoir Computing

- Photonic reservoir computing architectures provide nonlinear feedback with delayed response.
- Output signal depends on all previous inputs nonlinearly.
- Main advantage of using photonic systems is very high speed (>10 GHz) and small energy consumption



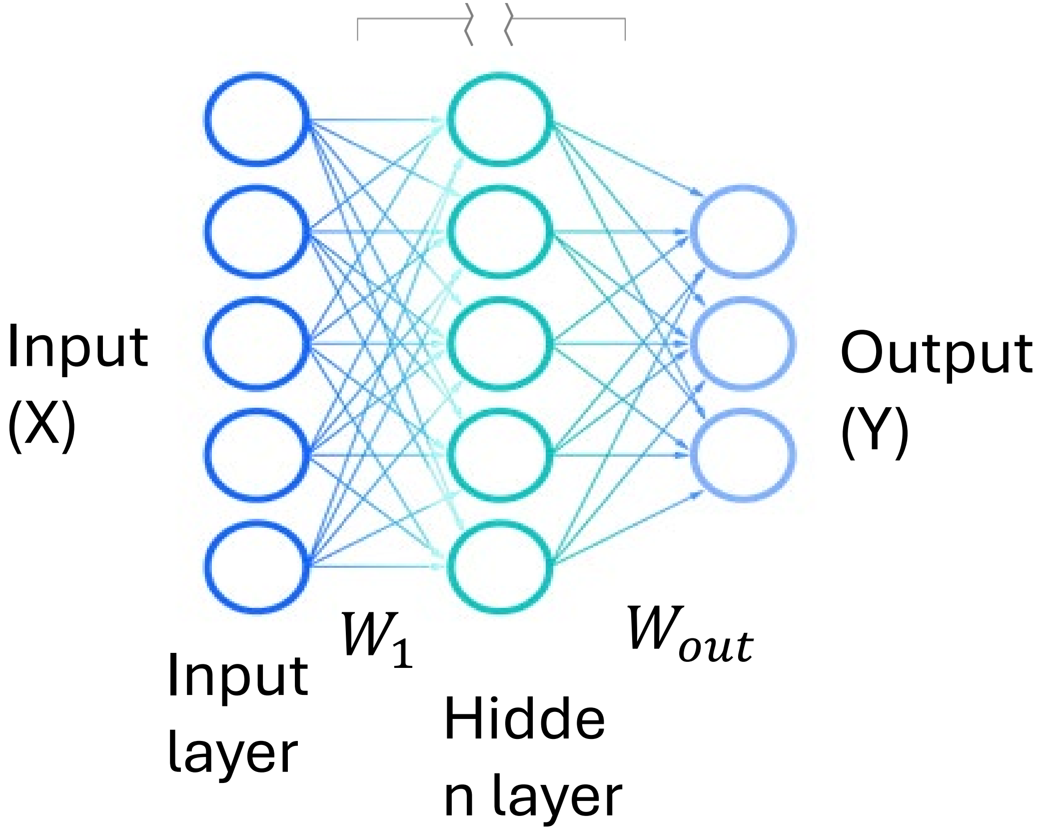




L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing," Opt. Express 20, 3241-3249 (2012)

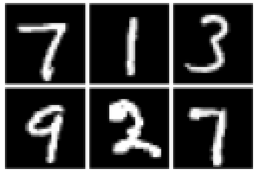
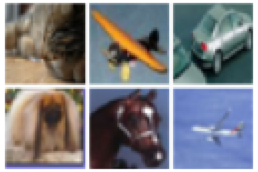
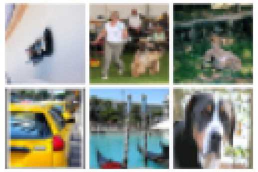
# Extreme Learning Machines

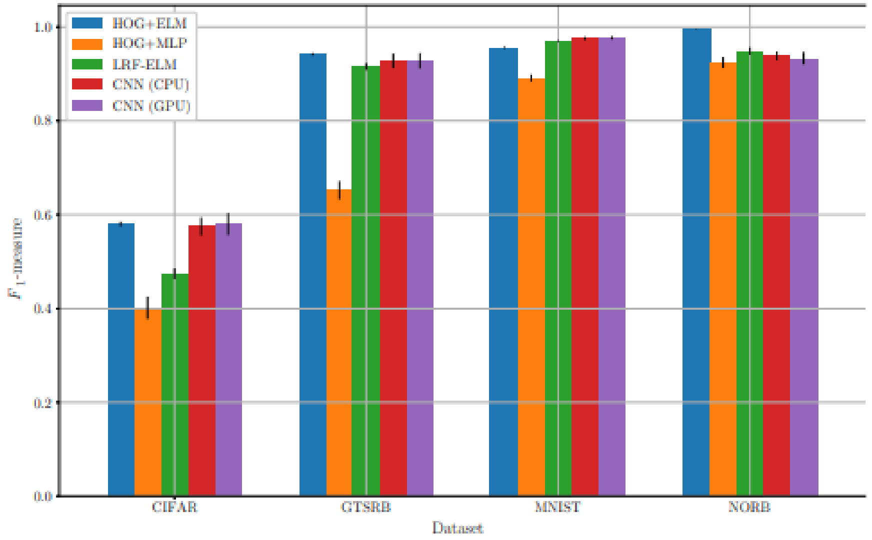
- Can be considered as a hybrid of Reservoir Computing and neural networks.
- Consists of single or multiple neuron layers of **fixed** weights and a single trainable read-out layer.
- Instead of iterative optimization, training is single inverse operation:

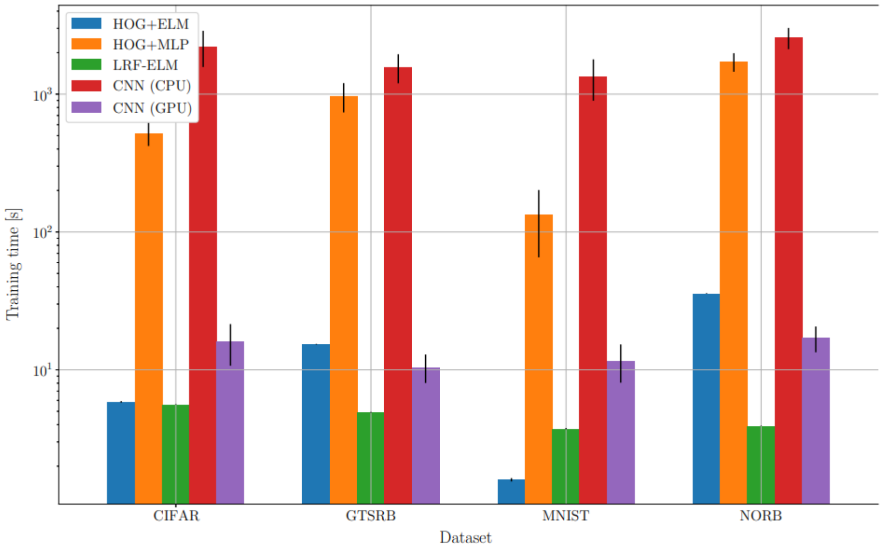
$$W_{out} = \frac{Y}{f(W_1 X + Bias)}$$



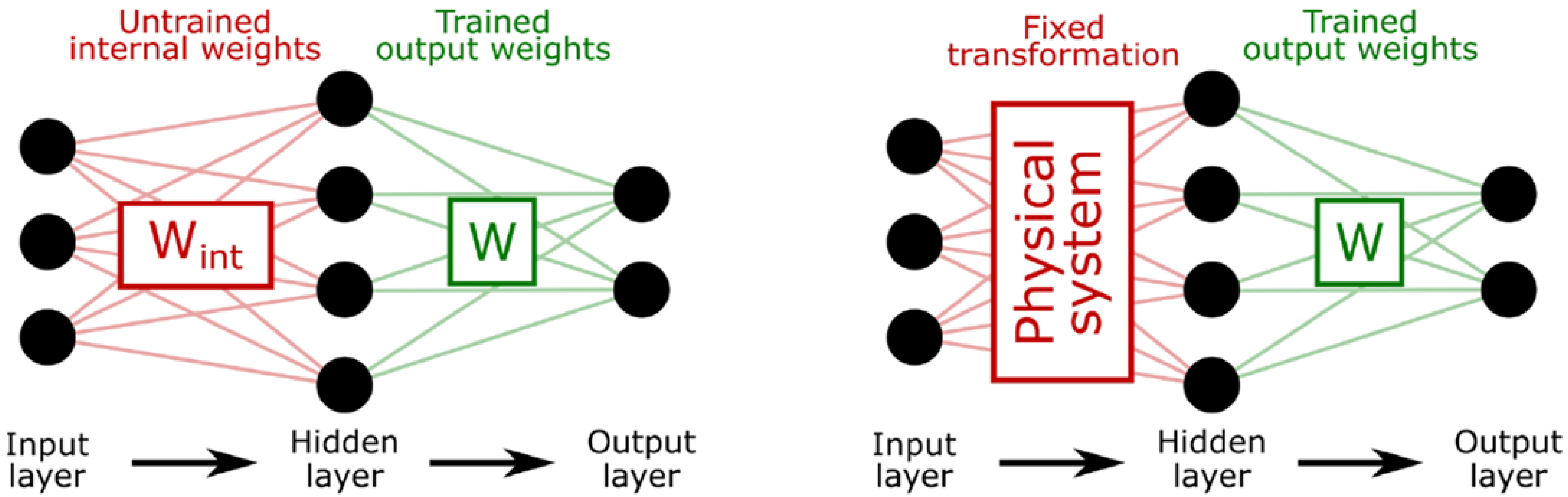
# Digital ELM Benchmarks

| Dataset | Size | Images examples |
|---|---|---|
| MNIST OCR | 70,000 |  |
| CIFAR-10 | 60,000 |  |
| Small NORB | 48,600 |  |
| GTSRB | 51,839 |  |
| ImageNet | 1.2 mln |  |



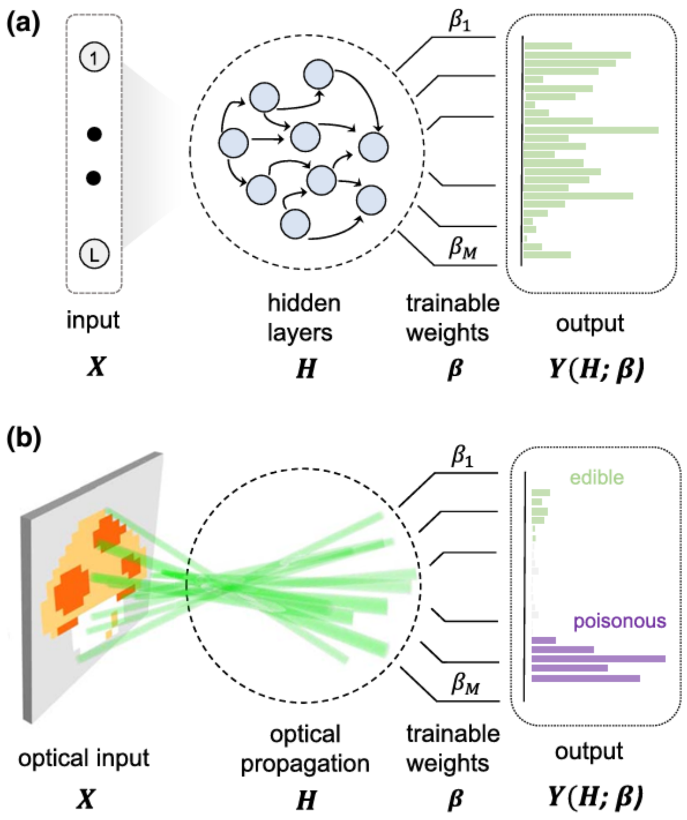
# Physical Extreme Learning Machines




(a) Single hidden layer ELM

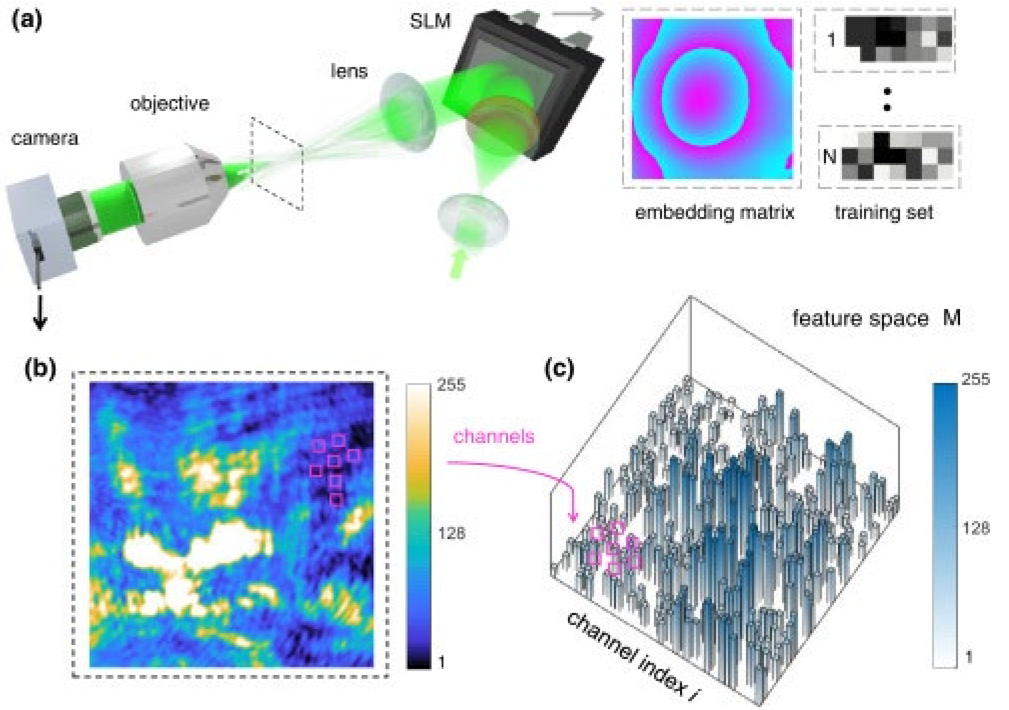
(b) Physical ELM

# Free-Space Photonic ELM

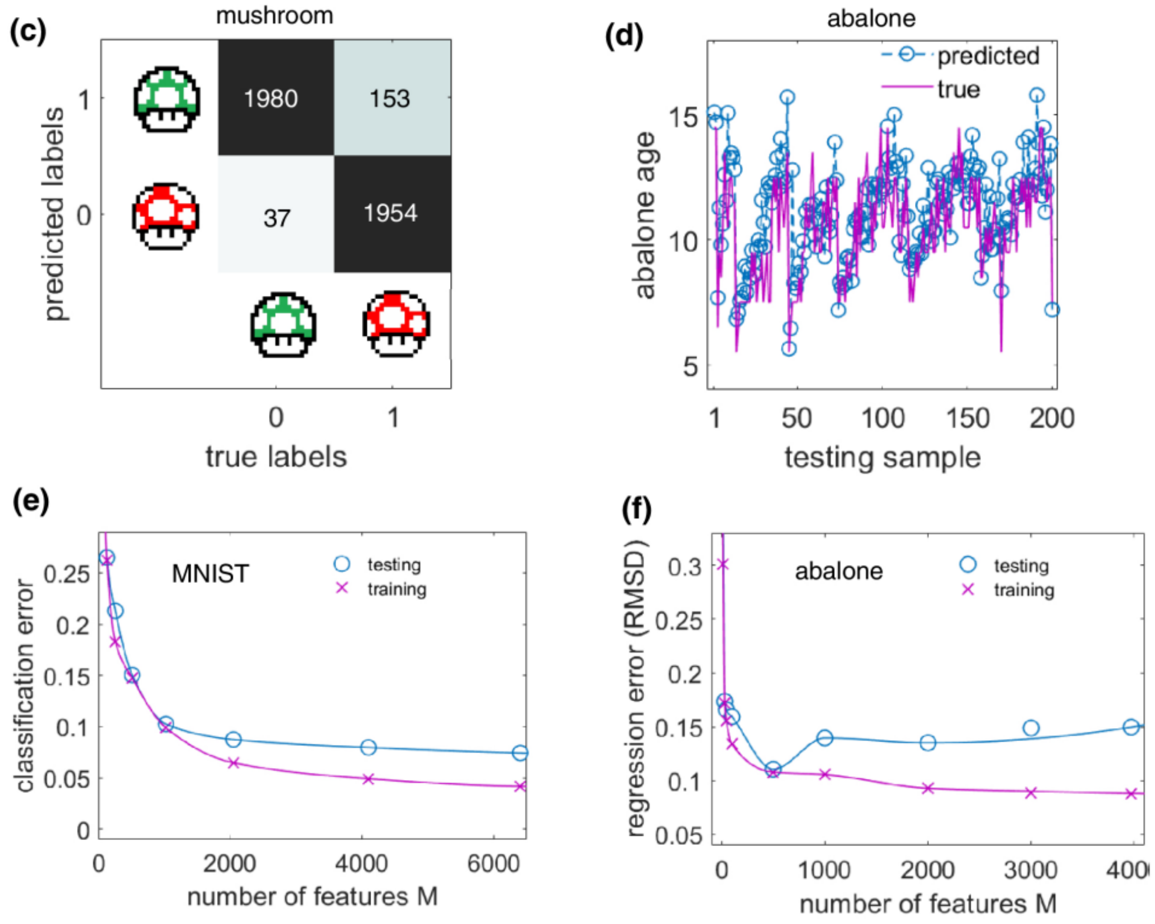



Davide Pierangeli, Giulia Marcucci, and Claudio Conti, "Photonic extreme learning machine by free-space optical propagation," Photon. Res. 9, 1446-1454 (2021)

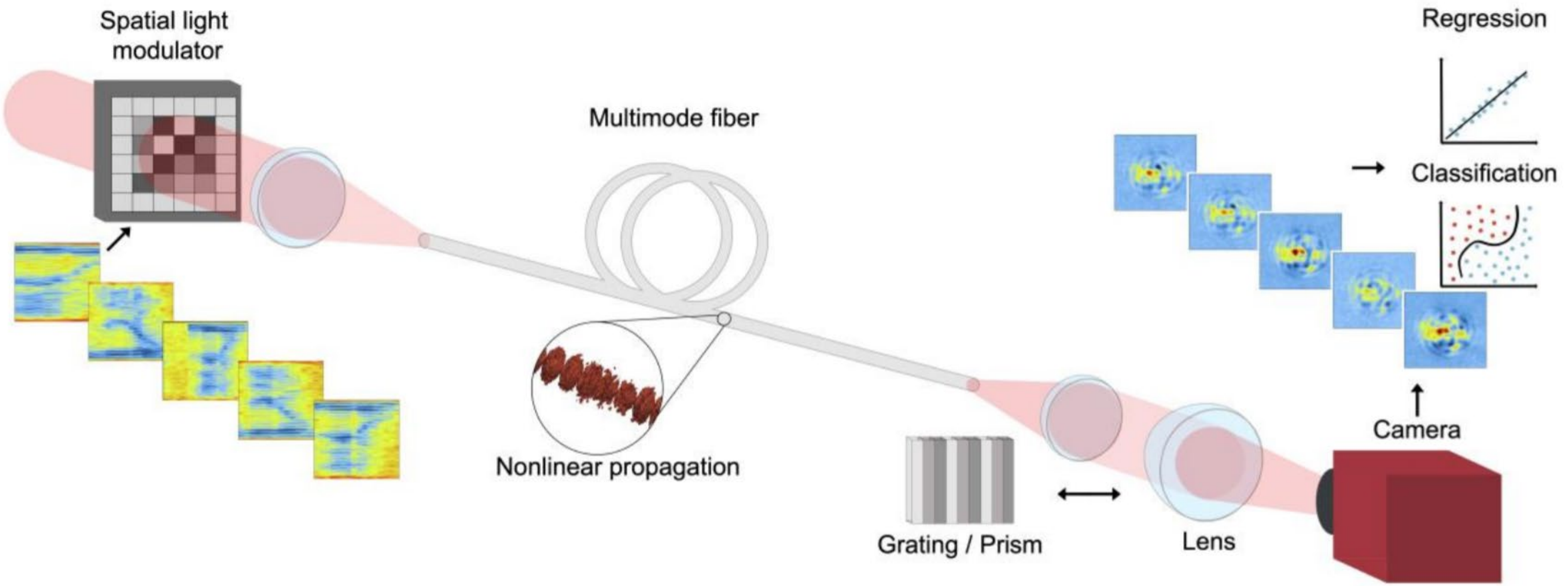
- The simplest ELM with optics is using the weights of the free space propagation.



# Free-Space Photonic ELM



# Computing with Spatiotemporal Nonlinearities of MMFs

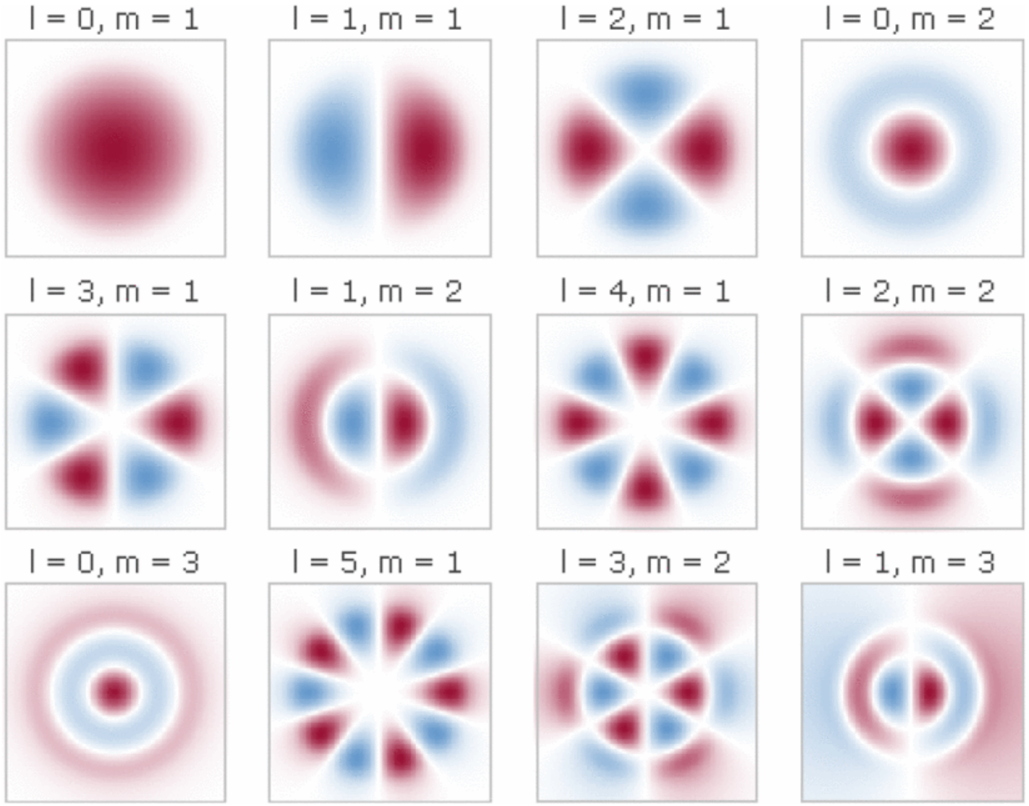



Teğin, U., Yıldırım, M., Oğuz, İ., Moser, C. & Psaltis, D. Scalable optical learning operator. *Nat. Comput. Sci. 2021 18* **1**, 542–549 (2021).

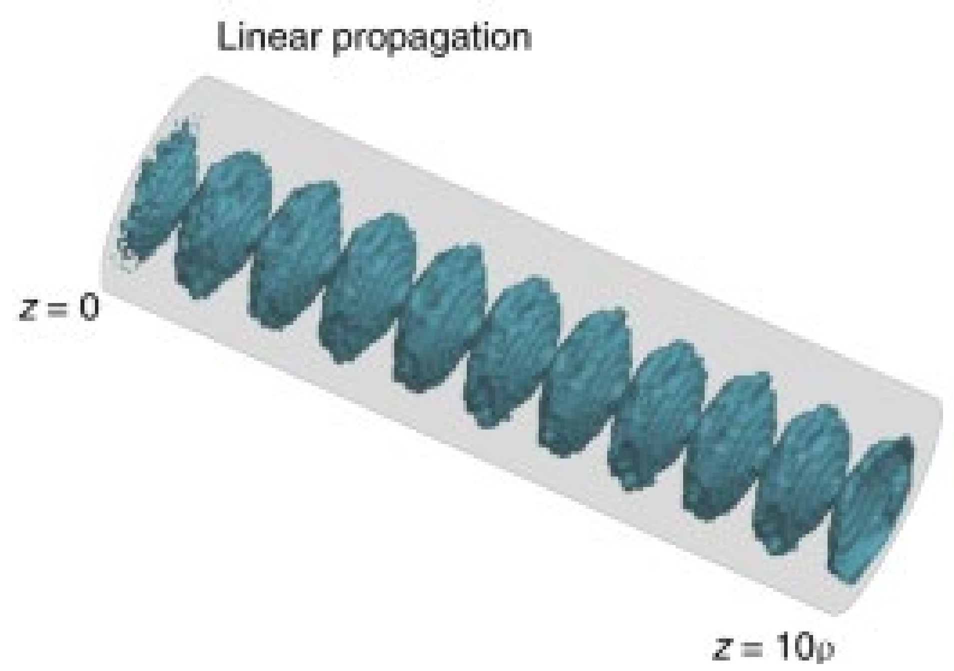
# Computing with Spatiotemporal Nonlinearities of MMFs

- In multimode fibers light propagate in discrete channels, depending on their properties MMFs can support up to millions of channels: $E\,(\mathrm{x}, \mathrm{y}, \omega) = \sum_n^N \mathrm{A}_n F_n(\mathrm{x}, \mathrm{y}, \omega)$

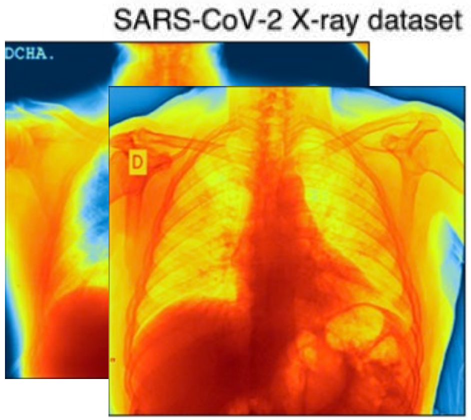
- At high intensities modes start to couple each other due to light-matter interactions.

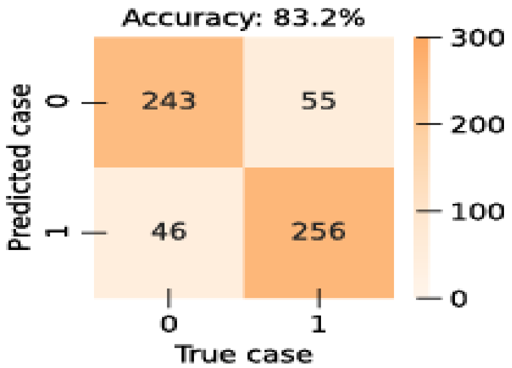


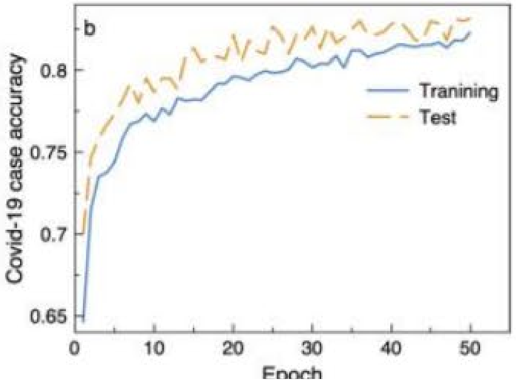
# Computing with Spatiotemporal Nonlinearities of MMFs

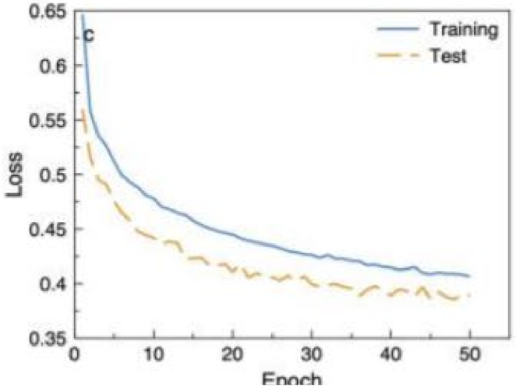


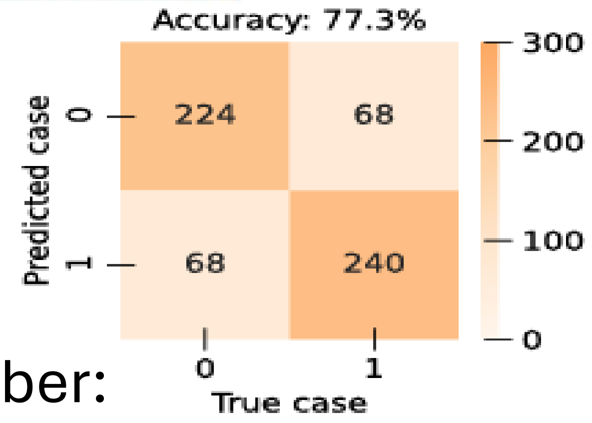
# Computing with Spatiotemporal Nonlinearities of MMFs

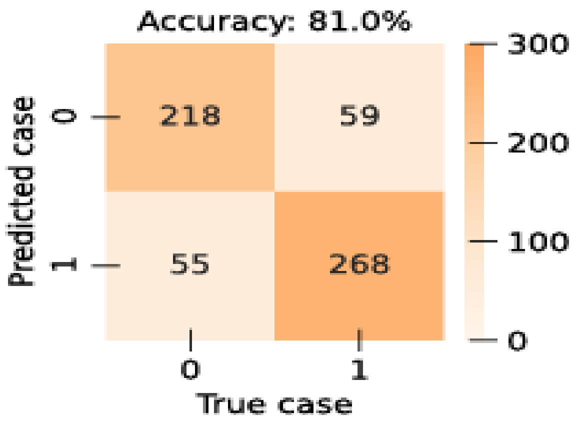


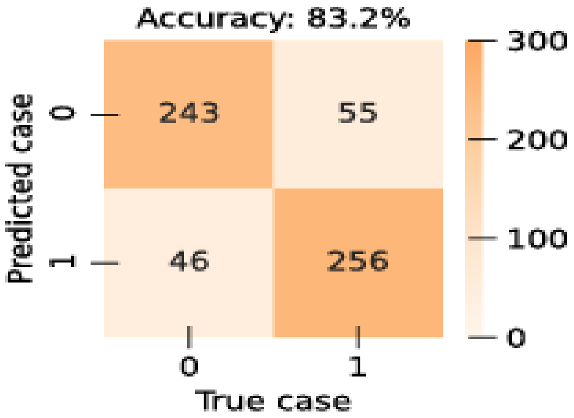



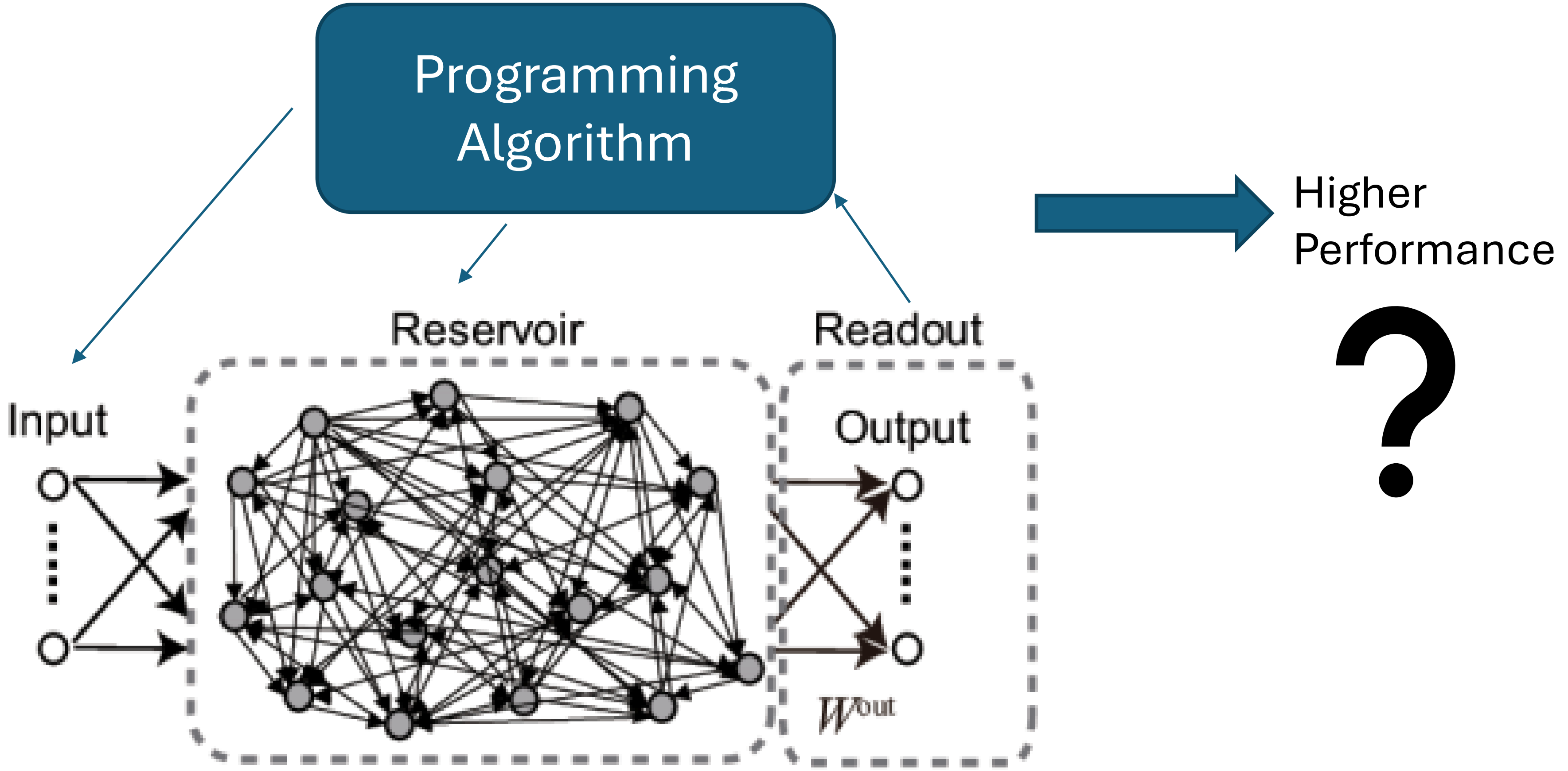




W/out fiber: 77%

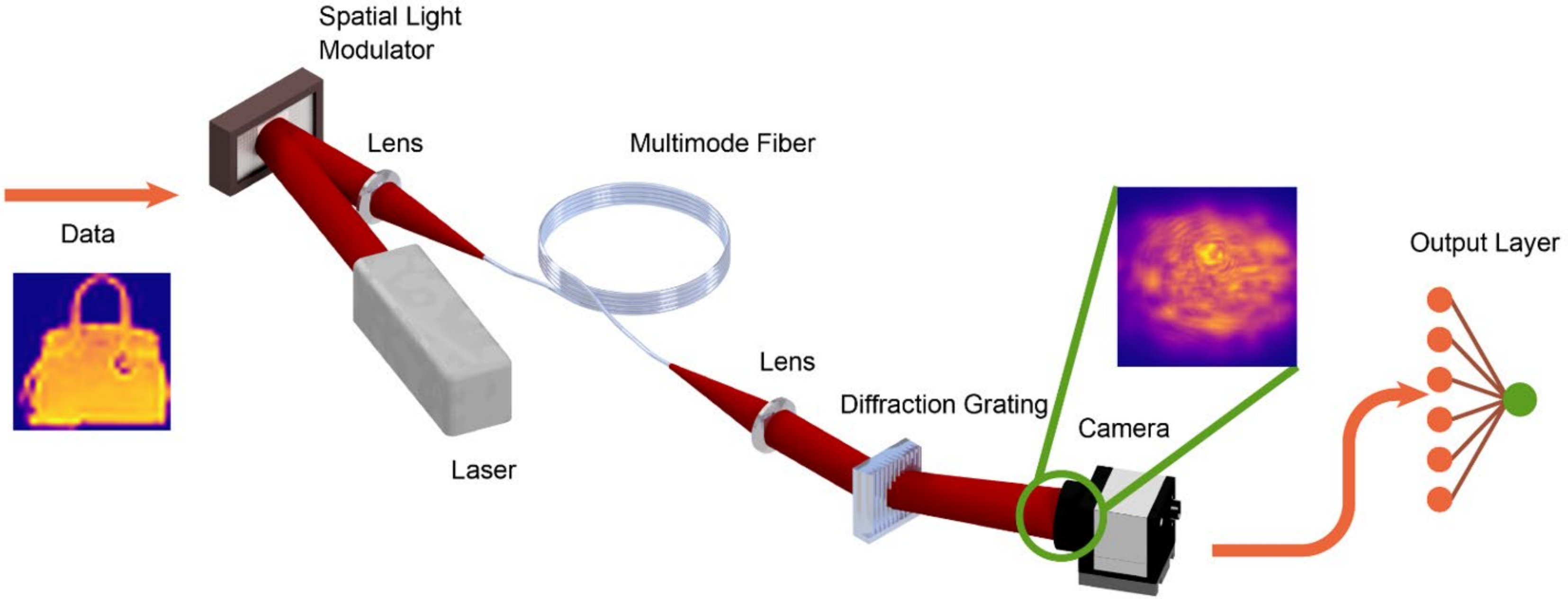
1.8 3.0 3.45

Pulse peak power (kW)

# Programming Propagation inside MMFs



# Programming Propagation inside MMFs

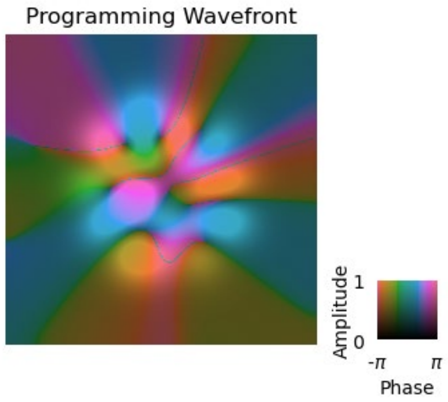


# Programming Propagation inside MMFs

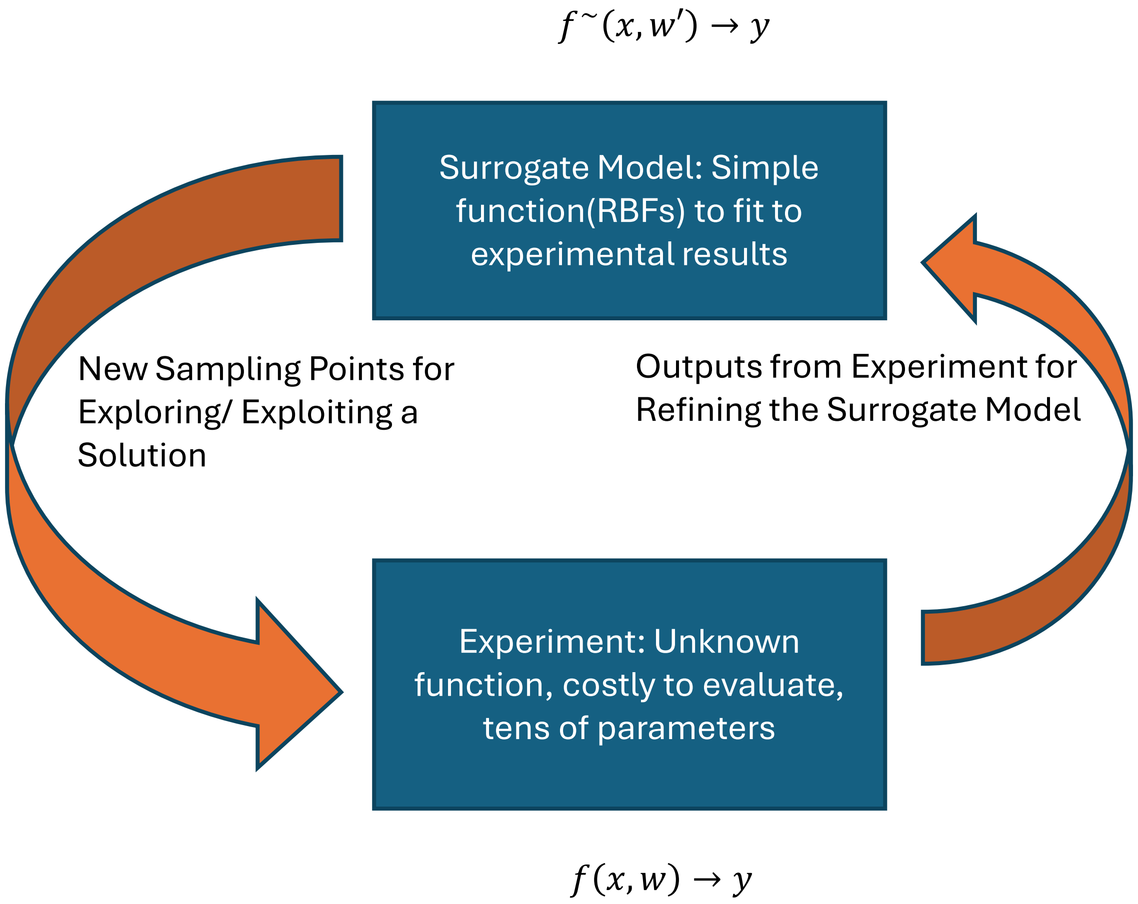
Programming Parameters

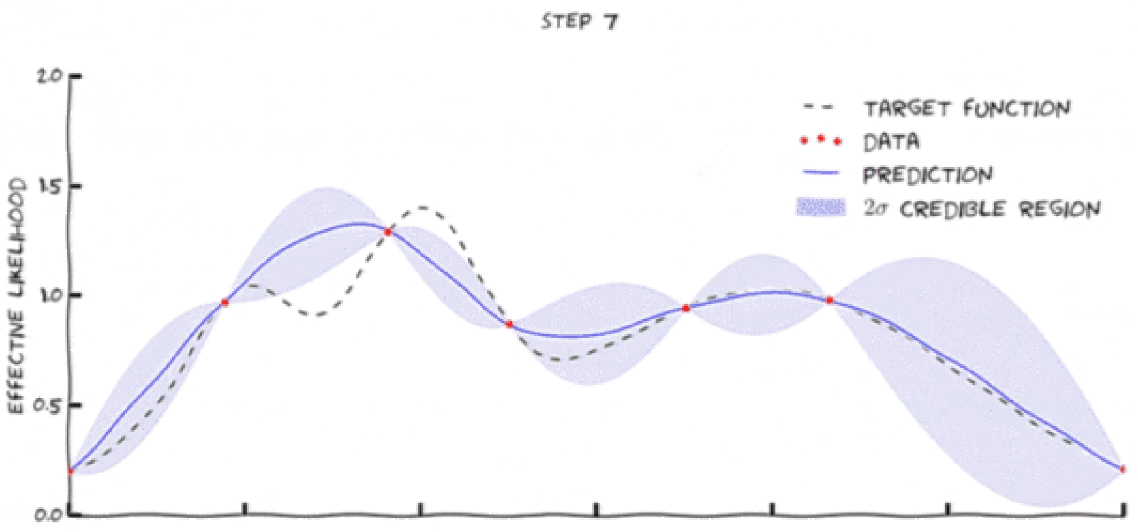
- Mode Shape Coefficients
- Experimental Parameters
  - Intensity
  - Diffraction Angle of Blazed Grating
  - Pattern Displacement
  - Defocus

$$WF(x,y) = \sum_{n=1}^{N} (a_{2n-1} + i * a_{2n}) * F_n(x,y)$$

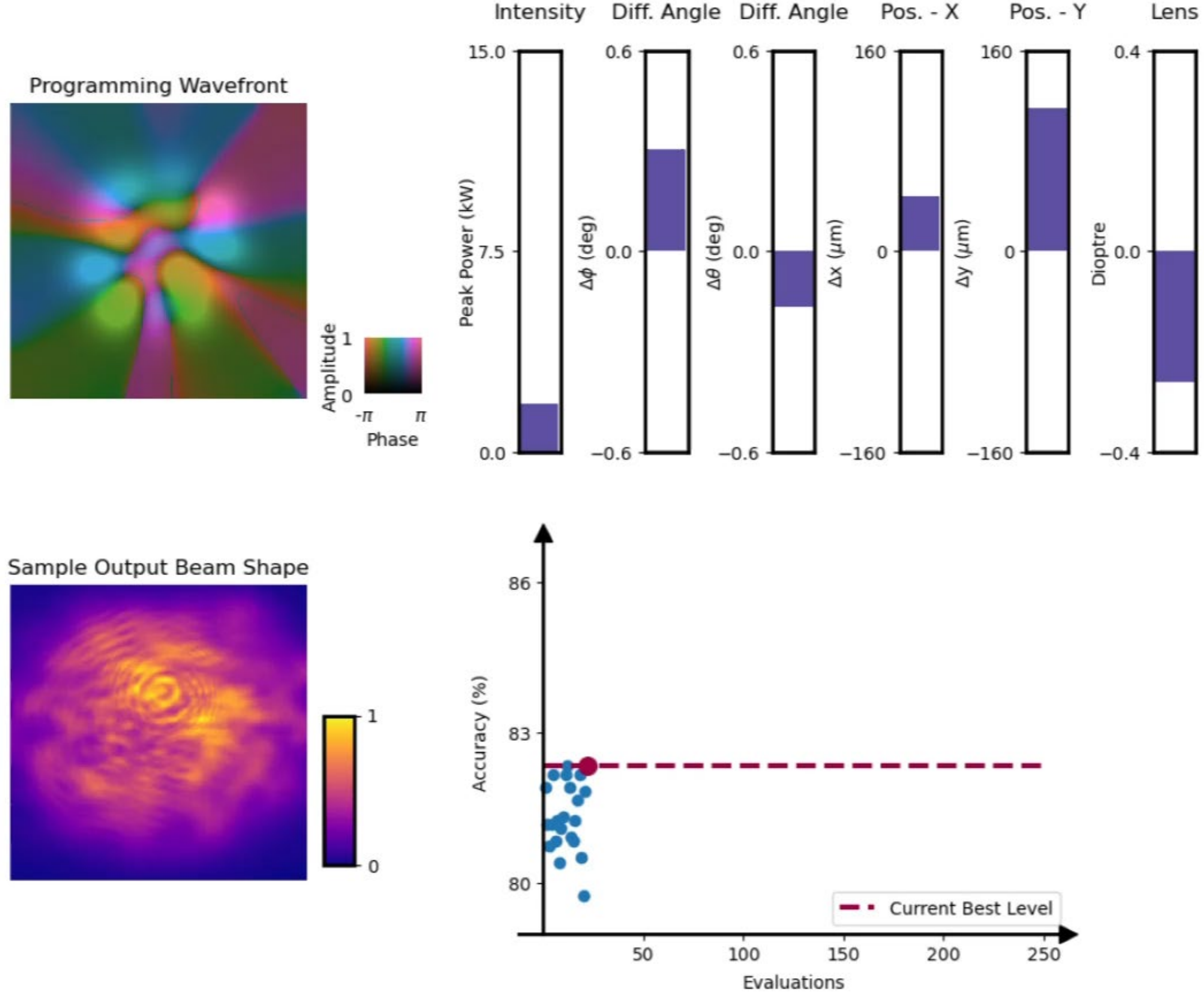


# Optimization of Programming Parameters



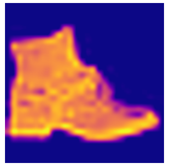
# Programming Propagation for MNIST-Fashion



# Programming Propagation for MNIST-Fashion

1200 training and 300 test samples

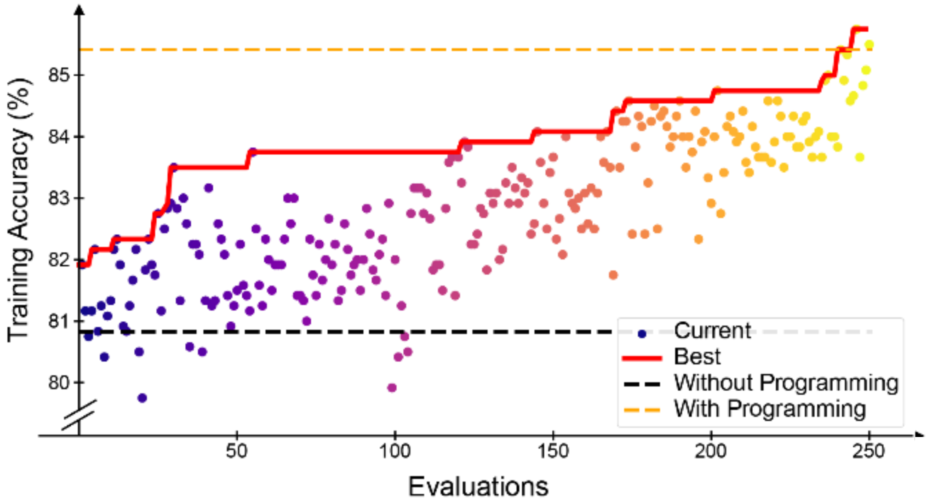
Bag

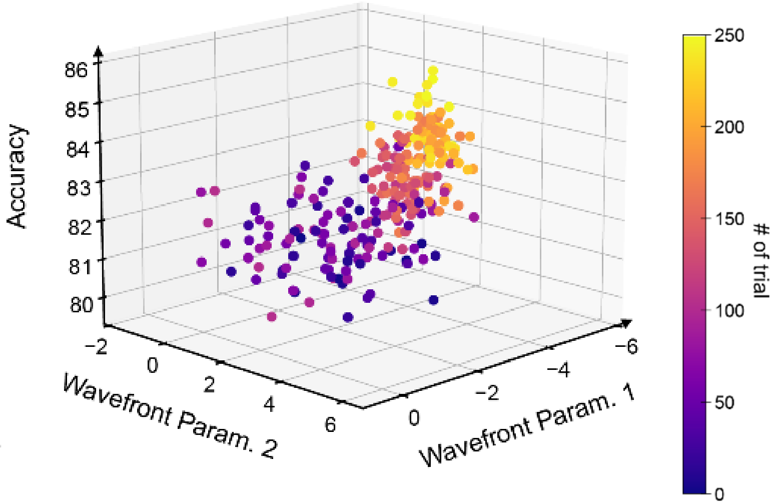


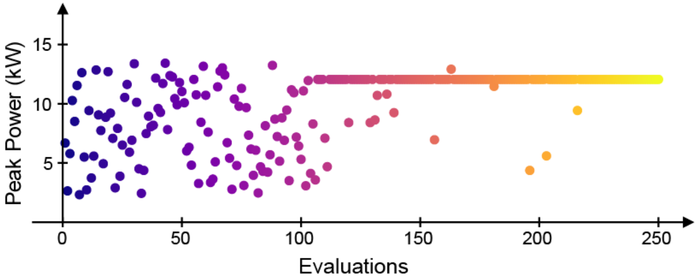
Boot

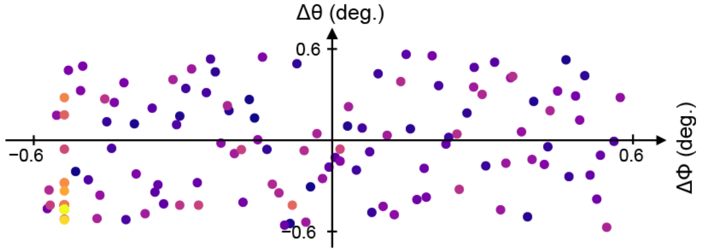
...

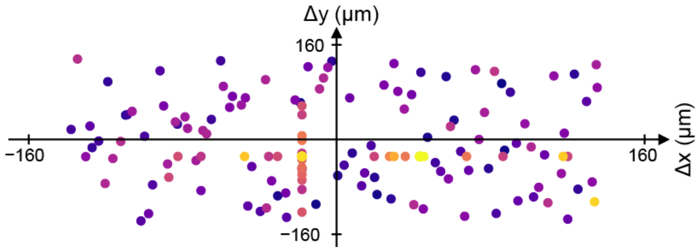
$\Delta Acc = +5.0\%$



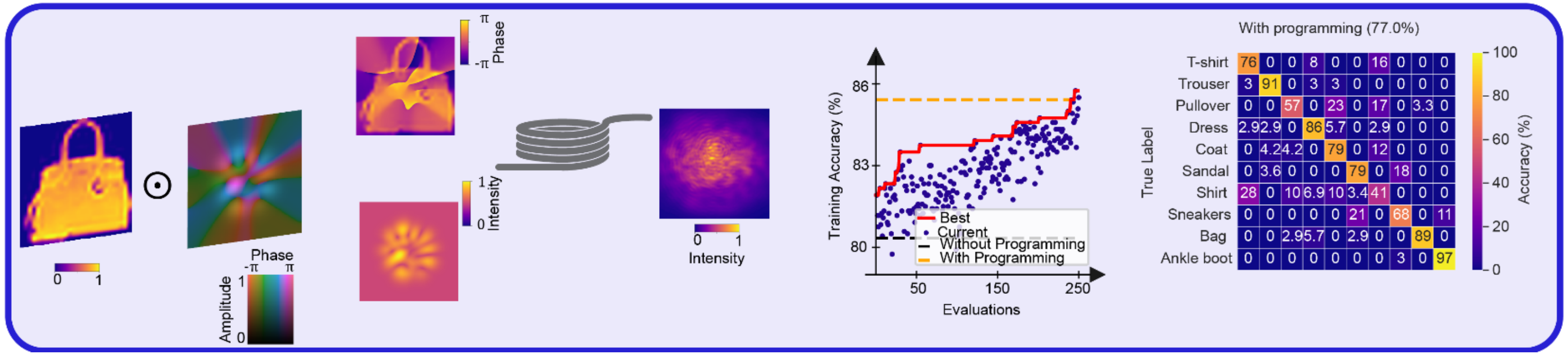



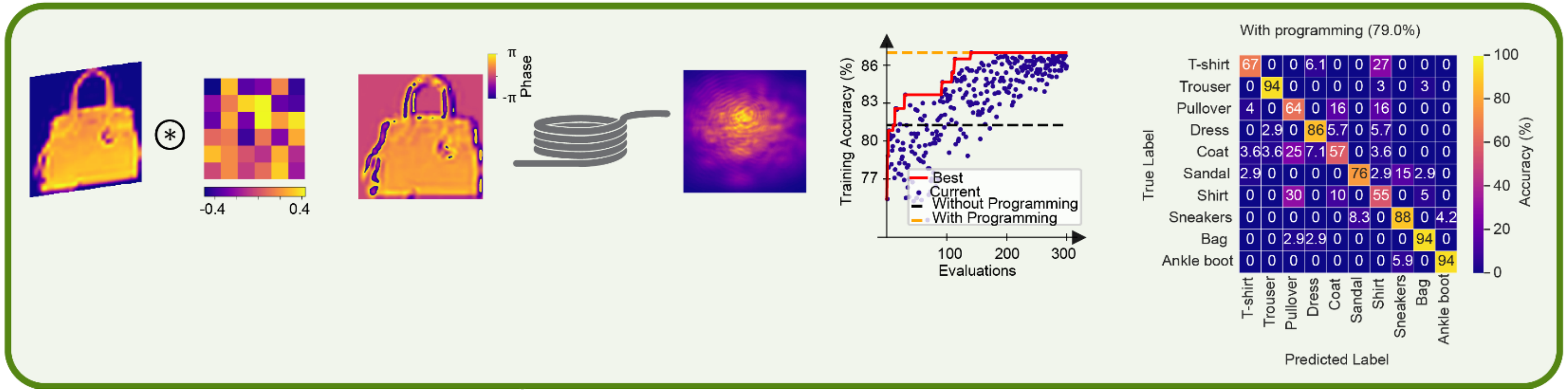



# Programming Propagation for MNIST-Fashion

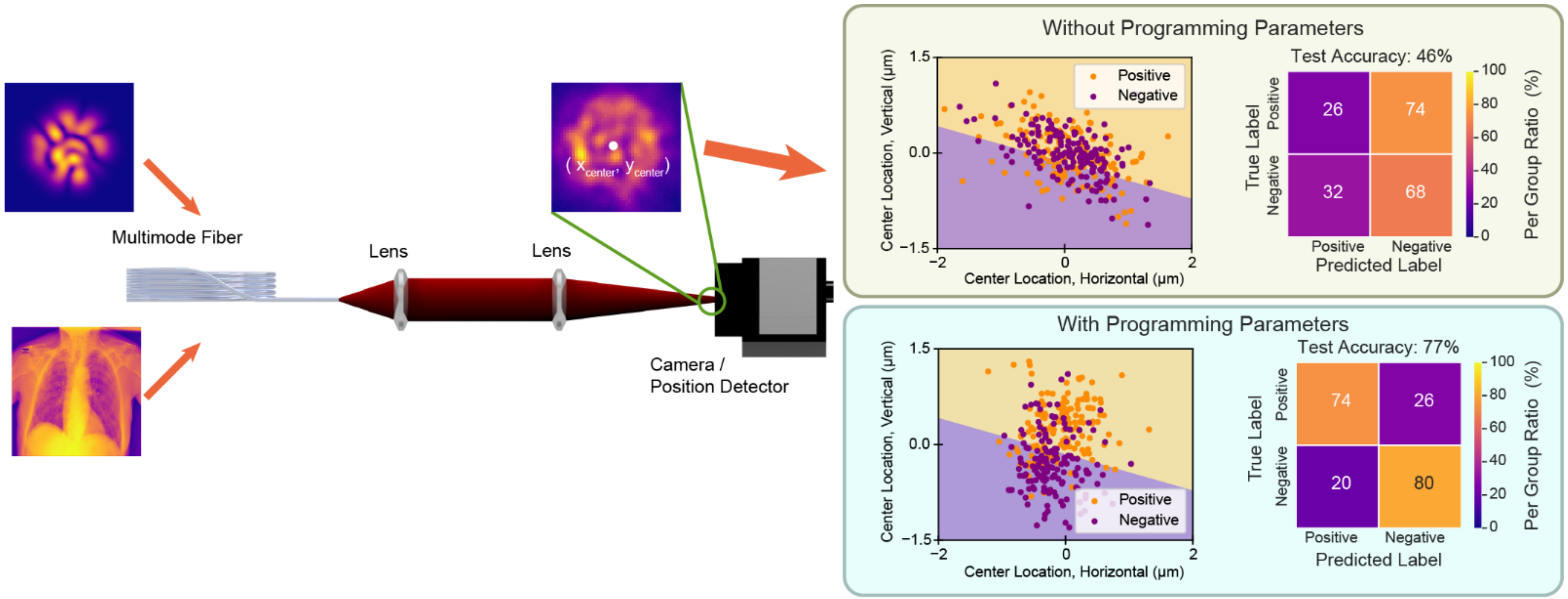
Complex Field Control



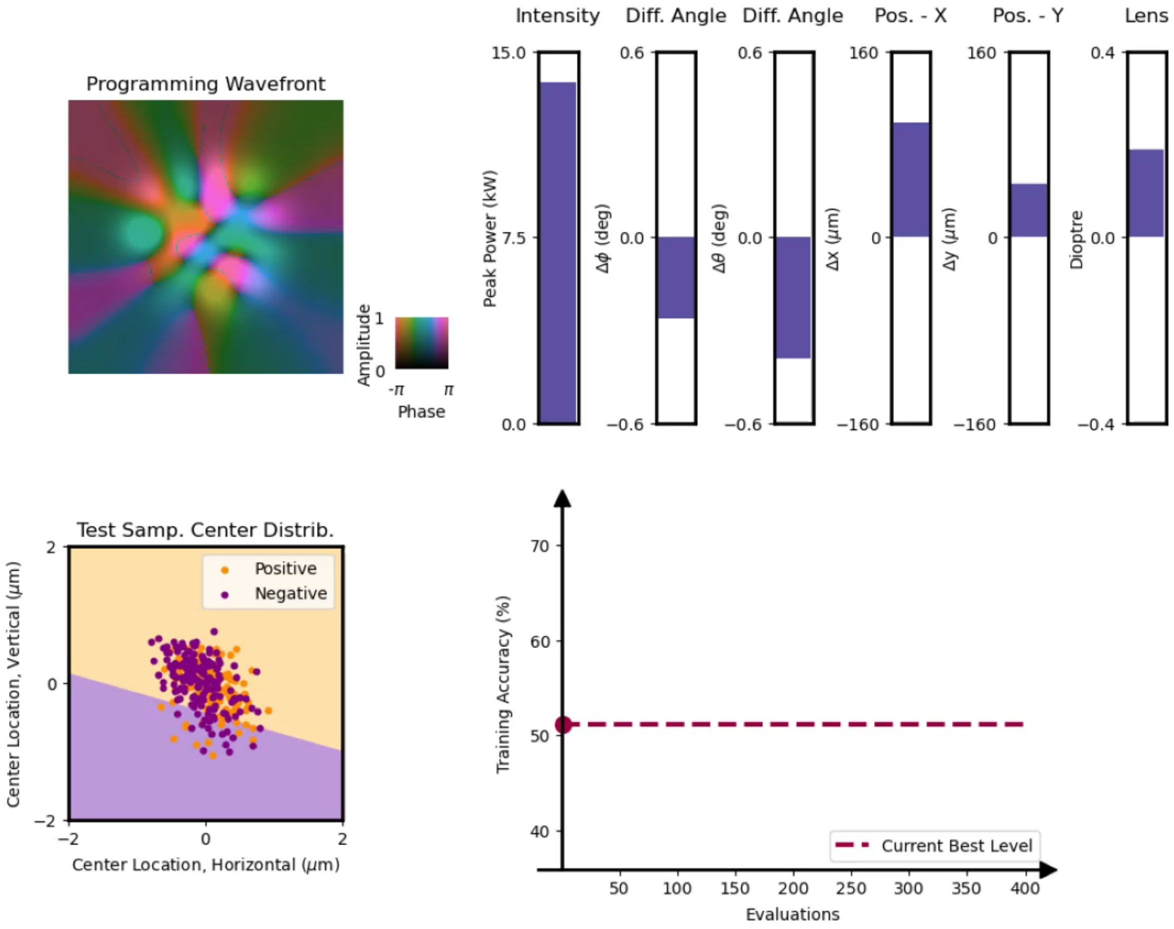

Convolution with Kernel



# Programming Propagation for All-Optical Classification



# Programming Propagation for All-Optical Classification

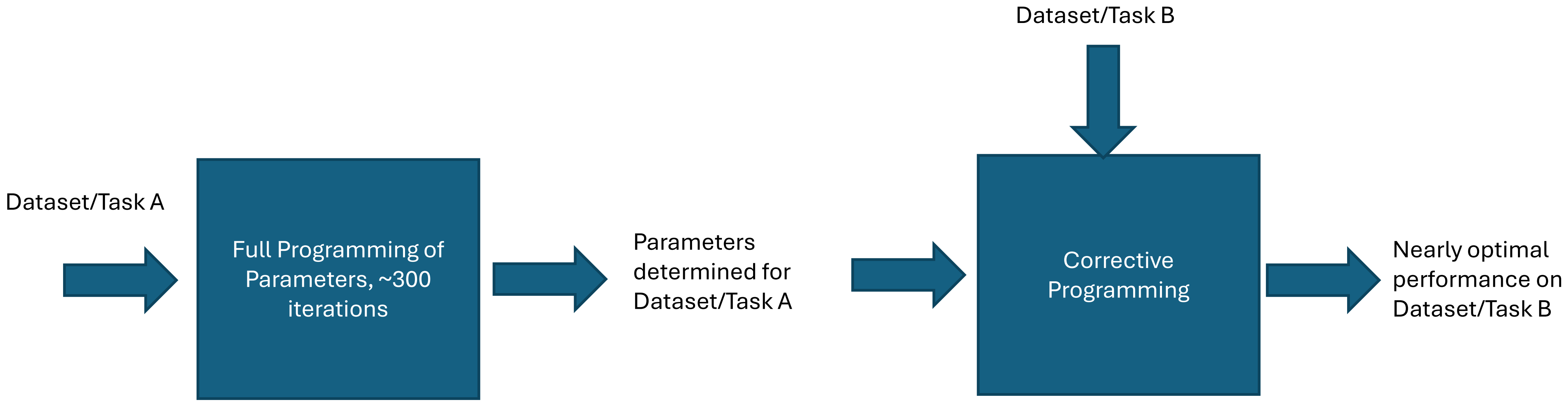


# Programming Propagation for All-Optical Classification

| Network Structure | Total Number of Parameters | Operations per Sample on Digital Computer (FLOP) | Accuracy on Melanoma dataset (%) | Accuracy on COVID- 19 dataset (%) |
|---|---|---|---|---|
| LeNet-5 | 82826 | 1175640 | 64.9 | 74.6 |
| MMF + classification with output location (with programming) | 55 | 2029 | 61.3 | 77.0 |

# Transferring Programming Parameters between Different Tasks

- Current method requires ~300 iterations over the dataset for programming to converge.
- For 1500 samples at 50 images per second, full programming corresponds to ~3 hours of training.



# Transferring Programming Parameters between Different Tasks

CelebA - Gender

CelebA – Young/ Not Young

Full P
Para

$Test\ Acc$


readout
ngle
ass, one
rsion

$Test\ Acc$:
$69.0\% \Rightarrow 76.0\%$

- Training from scratch : 76.3%

# Comparison with GPU-based NNs

Digital

| Network Structure | Total Number of Parameters | Operations per Sample on Digital Computer (FLOP) | Test Accuracy on Age Task | Test Accuracy on Gender Task |
|---|---|---|---|---|
| LeNet-5 | ~82k | ~1.2M | 63.0 | 75.2 |
| 7-layer Convolutional NN | ~410k | ~65M | 65.3 | 80.1 |
| MMF + linear output layer | 2026 | 4050 | 59.0 | 69.0 |
| Programmed MMF for Age Task + linear output layer | 2078 | 6075 | 67.0 | 76.0 |
| Programmed MMF for Gender Task + linear output | 2078 | 6075 | 64.7 | 76.3 |

Optical + Digital