# Multiplane light conversion design with physical neural network

**Zheyuan Zhu[1)], Joe H. Doerr[2)], Guifang Li[1)], Sean Pang[1)]**
[1]*CREOL, The College of Optics and Photonics, University of Central Florida*
[2]*School of Arts and Sciences, Rutgers, The State University of New Jersey*
*Author e-mail address: zyzhu@knights.ucf.edu, joehdoerr@gmail.com, li@ucf.edu, pang@ucf.edu*

**Abstract:** We present a physical neural network (PNN) approach towards multiplane light conversion (MPLC) design. PNN performs a full parameter search with flexible optimization pathways and can tune various design attributes as hyperparameters. © 2022 The Author(s)

## 1. Introduction

Multi-plane light converter (MPLC) designs supporting hundreds of modes are attractive in high-throughput optical communications[1]–[3]. These MPLC devices typically comprise >10 phase plates in free space, with millions of independent parameters to be optimized. Conventional MPLC design with wavefront matching (WFM)[4] updates one phase plate at a time via matching the forward and backward propagated wave fronts. Despite being an intuitive method, WFM lacks the flexibility to search the entire parameter space thoroughly and simultaneously, including the 2D phase profiles on each mask, the inter-mask distances, the number of phase masks, etc. As a result, WFM solution may not be optimal on a global scale, and the results cannot be tailored to specific design preferences.

From the perspective of optimization, MPLC design aims to maximize the coupling efficiency by varying the independent phase and distance parameters. The optical backward propagation and phase update in WFM is an exact solution to this optimization problem, if only one mask is adjustable[2], [5]. Here we present a physical neural network (PNN) that models the MPLC process. PNN training approach toward MPLC design enables a more flexible optimization sequence than WFM, and provides access to the entire parameter space for performance tuning.

## 2. Physical neural network for MPLC

An MPLC that converts $M$ input fields $E_0$ into target output fields $E_t$ using $N$ phase plates $\phi_1$, $\phi_2$,…,$\phi_N$ is equivalent to an $N$-layer PNN model. The distance between adjacent phase plates, $\phi_i$ and $\phi_{i+1}$, is $z_i$. After discretizing and serializing the fields and phase profiles into vectors, the output field, $E_N$, can be calculated from successive free-space propagations and phase modulations as in Eq. (1).

$$E_N = \mathcal{F}^{-1}\mathrm{diag}(\exp(ik_z z_N))\mathcal{F}\left[\prod_{i=1}^{N}(\mathrm{diag}(\exp(i\phi_i))\mathcal{F}^{-1}\mathrm{diag}(\exp(ik_z z_{i-1}))\mathcal{F})\right]E_0 \qquad (1)$$

Here $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the 2D Fourier transform matrix and its inverse, respectively. $\mathrm{diag}(\mathbf{x})$ creates a diagonal matrix from the vector $\mathbf{x}$. $k_z = \left(k_0^2 - k_x^2 - k_y^2\right)^{1/2}$ is the longitudinal wave vector. Based on Eq. (1), we constructed a PNN model, shown in Figure 1, with the weights $W_i = \mathbf{\Phi F}^T\mathbf{ZF}$ , in which parameters $\phi_i$ and $z_i$ are trainable. The loss function is derived from the coupling efficiency $L = -\sum_j \int |E_N^{(j)*} E_t^{(j)}| \mathrm{d}x\mathrm{d}y$ between output and target fields, assuming all fields are normalized. The PNN is built and trained in TensorFlow with ADAM optimizer.

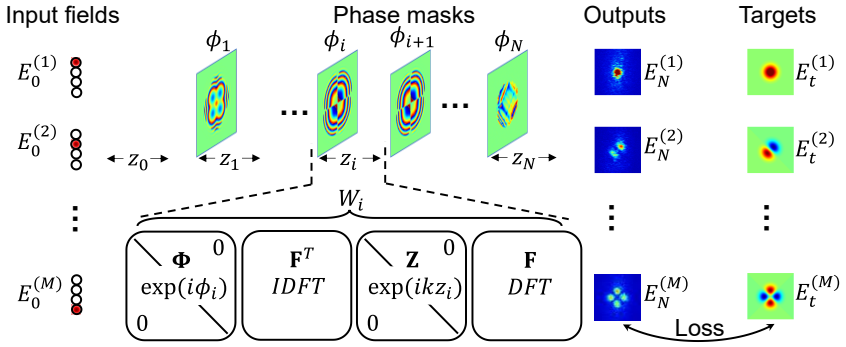


Figure 1. MPLC-equivalent physical neural network (PNN) model. Each phase mask along with the subsequent free-space propagation can be represented by a network layer, $W_i$. MPLC design can then be treated as PNN training.

## 3. Results and discussions

We have compared PNN and WFM results with a 10-mode MPLC using 5 masks, shown in Figure 2(a). In addition to coupling efficiency, we have also examined the sharpness, $\delta L$, defined as the relative change in coupling efficiency under 0.05rad random phase perturbations (equivalent to 2 grayscale levels on an 8-bit SLM). Figure 2(b) illustrates the optimization pathways undertaken by WFM and PNN on the coupling efficiency contour. WFM jumps directly to a series of critical points along one of the mask dimensions with the rest of the masks fixed. In contrast, PNN follows the global gradient towards the nearest local maximum, which converges to a different solution than WFM. Though PNN takes more iterations to converge, the use of GPU in PNN training resulted in slightly shorter execution times (933s) than WFM (961s). The PNN and WFM solutions (Figure 2(c)) have coupling efficiencies of 0.764 and 0.763, and sharpness of 0.8% and 0.65% respectively, suggesting similar performance. We can also engineer a sequential optimization pathway in PNN to mimic the WFM solution, which iteratively sets one mask as trainable and freezes the rests. Figure 2(d) shows that the sequential PNN pathway closely follows that of WFM, and the masks from WFM and PNN in each iteration share a similarity (overlap integral) of >99%. These results show that WFM is a subset of the numerous optimization pathways available in PNN.

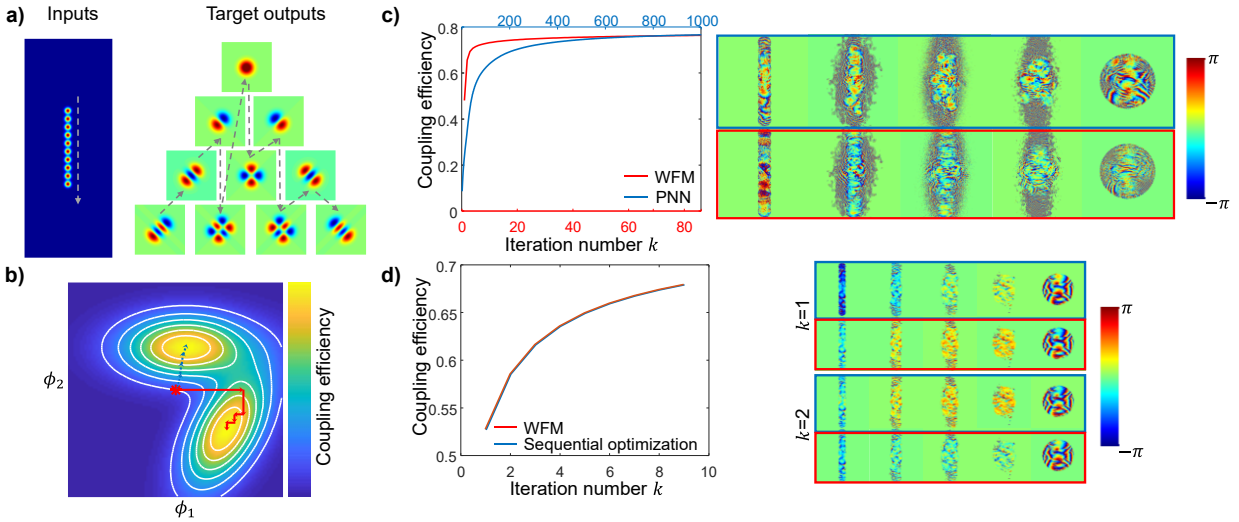


Figure 2. (a) 10-mode MPLC for comparison. (b) Illustration of the optimization pathway for WFM (red) and PNN (blue). (c) Masks from WFM and PNN fall into two different solutions but with similar performance. (d) Customized sequential optimization in PNN produces identical results as WFM.

PNN is not limited to the pathways presented in these examples. We can design better optimization sequences that adaptively select the subset of the parameters and learning rate in each iteration, thus accelerating the PNN convergence. In addition, WFM cannot efficiently optimize $z_i$ alongside the masks unless a brute-force search is performed. PNN considers both masks $\phi_i$ and inter-mask distances $z_i$ as trainable parameters, which can be updated simultaneously in ADAM optimizer. Finally, the hyperparameters in PNN, including the batch size, learning rate, can be tuned to balance the convergence speed and computing resources in large-scale MPLC design. We will also present results on inter-mask distance optimization and large number mode-conversion in our presentation.

### Acknowledgements

This work was supported in part by NSF (1932858), and in part by the ONR (N00014-20-1-2441).

### References

[1] N. K. Fontaine, R. Ryf, H. Chen, D. Neilson, and J. Carpenter, "Design of High Order Mode-Multiplexers using Multiplane Light Conversion," Sep. 2017, doi: 10.1109/ECOC.2017.8346129.

[2] N. K. Fontaine, R. Ryf, H. Chen, D. T. Neilson, K. Kim, and J. Carpenter, "Laguerre-Gaussian mode sorter," *Nat. Commun.*, vol. 10, no. 1, p. 1865, Dec. 2019, doi: 10.1038/s41467-019-09840-4.

[3] N. Barré, B. Denolle, P. Jian, J.-F. Morizur, and G. Labroille, "Broadband, Mode-Selective 15-Mode Multiplexer Based on Multi-Plane Light Conversion," in *Optical Fiber Communication Conference*, 2017, vol. Part F40-O, p. Th2A.7, doi: 10.1364/OFC.2017.Th2A.7.

[4] T. Hashimoto, T. Saida, I. Ogawa, M. Kohtoku, T. Shibata, and H. Takahashi, "Optical circuit design based on a wavefront-matching method," *Opt. Lett.*, vol. 30, no. 19, p. 2620, Oct. 2005, doi: 10.1364/OL.30.002620.

[5] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica*, vol. 5, no. 7, p. 864, Jul. 2018, doi: 10.1364/OPTICA.5.000864.