Physics-Constrained Comprehensive Optical Neural Networks
—Supplemental Material—

A Training Details

The training process diagram, as shown in Fig.1.To provide a comprehensive overview of the training process, the following aspects were considered and meticulously detailed:
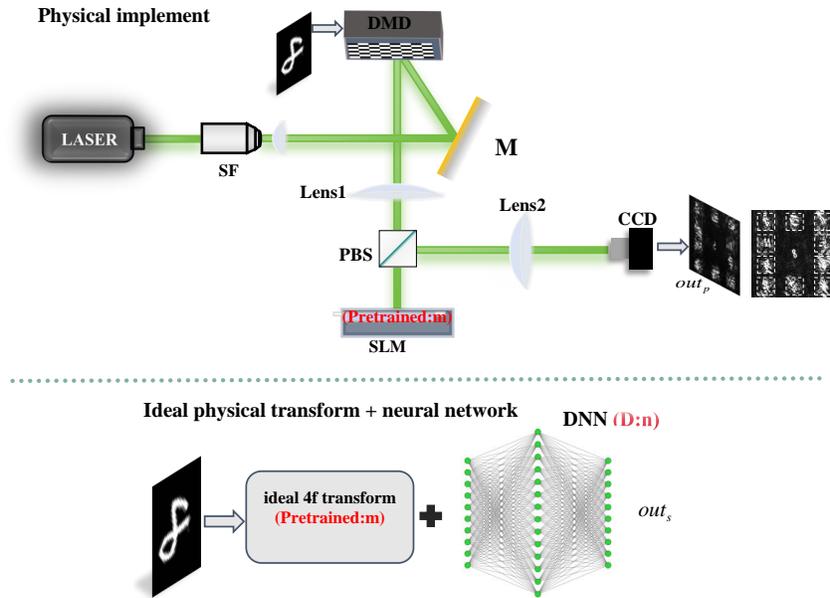
Figure 1: Training process of physics-prior-based error compensation network

Data Preprocessing: Using the MNIST dataset as an example, the entire training set is used to train the parameters of the ideal 4f transform depicted in Fig.1. A random subset of 1000 images is selected from the training set to train the Deep Neural Networks(DNN) shown in Fig.1. Finally, another random subset of 1000 images is selected from the test set for a blind test to evaluate the classification accuracy after deploying the trained parameters in the experiment.

Model Architecture: The model comprises an ideal 4f transformation and a DNN. The input and output of the ideal 4f transformation adhere to the transformation relationship described by Eq 1, where $\mathcal{L}$ and $\mathcal{L}^{-1}$ denote the Fourier transform and its inverse, respectively. The process involves performing a Fourier transform on the input image, modulating its phase in the frequency domain using a phase matrix m, and then applying an inverse Fourier transform to return to the time domain,

Preprint. Under review.

thereby completing the convolution operation on the input image. In Eq 1, *m* represents the phase matrix to be deployed on the spatial light modulator(SLM), and it is the primary optimization target of our focus.

$$Output = \mathcal{L}^{-1}\left[\mathcal{L}\left(Input\right) \times m\right] \tag{1}$$

In the initial stages of this work, we experimented with various DNN architectures, including U-Net, convolutional, and deconvolutional networks. However, as our research progressed and we incorporated physical constraints, we discovered that a highly lightweight fully connected architecture could achieve satisfactory results. The structure of the DNN is as follows:(1)The first layer is a linear transformation that projects the input from a 10-dimensional feature space to a 32-dimensional feature space. This transformation is followed by a ReLU(Rectified Linear Unit) activation function to introduce non-linearity.(2)The second layer takes the 32-dimensional output from the previous layer and projects it into a 64-dimensional space, followed by another ReLU activation function.(3)The third layer reduces the dimensionality from 64 back to 32, with a subsequent ReLU activation function.(4)The fourth and final layer maps the 32-dimensional features back to a 10-dimensional output space, matching the original input dimensionality. Notably, there is no activation function applied after this layer, meaning the output is in a linear form.

Experimental Setup: The setup utilized a laser with a wavelength of $532\,\text{nm}$, coupled with a beam expander to produce a homogeneous, parallel beam. Due to the polarization sensitivity of the setup's SLM[1–3], a polarizer was employed to modulate the light polarization. The DMD, equipped with a Texas Instruments DLPC900 chip, featured a resolution of 1920×1080 pixels, a pixel size of $7.56\,\mu\text{m}$, and a 92% fill factor. Two lenses, each with a focal length of $20\,\text{cm}$, were arranged to constitute a 4f system. The SLM, a Holoeye LETO model with a resolution of 1920×1200, a pixel size of $8\,\mu\text{m}$, a 95% fill factor, and a phase modulation range of $0 - 2\pi$ (equivalent to $0 - 255$ in digital steps), was utilized to upload the phase map. Captured two-dimensional light intensity images were obtained using a Daheng Imaging MER-130-30-UM CCD camera, with a resolution of 4096×3000.
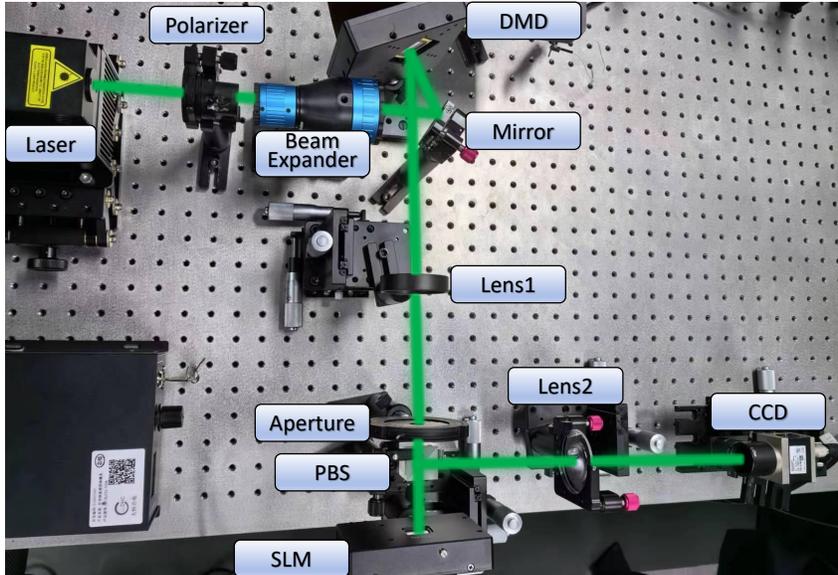
Figure 2: Experimental setup of an optical Fourier convolutional neural network with error compensation network. DMD: digital micromirror device; SLM: spatial light modulator; PBS: polarization beam splitter.

We first load the input image and the trained phase matrix *m*onto the DMD and SLM, respectively, and then proceed with the experiment. The laser, after collimation and beam expansion, illuminates the DMD. The reflected laser from the DMD carries the input image information and performs a

2

Fourier transform at the first lens. The beam then passes through a beamsplitter and impinges on the SLM. After phase modulation by the SLM, it undergoes another Fourier transform through a second lens and is finally captured by the camera. The beamsplitter does not modulate the laser; it simply ensures that the laser can vertically incident on the SLM and reflects the modulated light by 90 degrees. The actual experimental setup is illustrated in Fig. 2

Training Procedure: First, we disregard various experimental errors and train a phase matrix m_0 that can classify the MNIST dataset into ten categories using the ideal 4f transformation. This matrix is loaded onto the SLM. Simultaneously, we randomly select 1000 images from the training set, sequentially load them onto the DMD, and use a camera to capture the experimental output images of these 1000 input images.

Similarly, we input these 1000 images into the ideal 4f transformation and run a forward pass to obtain the computer-generated output images with the phase matrix set to m_0 . For both sets of images, we select ten regions of interest as shown in Fig 1 and calculate their mean values, resulting in two datasets with shapes of (1000, 10) for the experimental and simulated data.

The simulated data is used as the input for the DNN, and the experimental data is used as the ground truth to train the DNN. The DNN is trained($n \rightarrow n_1$) to fit the simulated data to the experimental data. The loss function for this training stage is shown in Eq 2.

$$Loss = MSE\left(Out_p - Out_s\right) \tag{2}$$

After completing the training of the DNN, we fix the parameters n_1 of the DNN and use the phase matrix m_0 as the initial parameter for the ideal 4f transformation. The DNN is then connected after the ideal 4f transformation, forming the Optical Deep Neural Network (Optical-DNN). We optimize the parameter m_0 to m_1 to enhance the classification accuracy of the Optical-DNN for input images. The loss function for this optimization stage is shown in Eq 3.This process is iterated continuously until the Optical-DNN converges.

$$Loss = ReLu\{W_{Gap} - |I_{max} - I_{2rdmax}|\} + \sum_{i=1}^{N} y_i log(\hat{y_i}) \tag{3}$$

Here, the W_{Gap} denotes the light intensity gap, N denotes the total number of classes, y represents a one-hot encoded vector that indicates the true class labels, with y_i being the i-th element of the vector y. The term $\hat{y}$ corresponds to the network's output probabilities, which are typically derived through the application of a softmax function, with $\hat{y_i}$ representing the probability that the model assigns to the likelihood that the sample pertains to class i.

Hardware and Software: For our model training, we utilized the NVIDIA RTX 4090 GPU, which features 24GB of GDDR6X memory, a 384-bit memory interface width, and a memory bandwidth of 1008 GB/s. The GPU is powered by the AD102-300-AI core, comprising 16,384 CUDA cores, 512 Tensor cores, and 128 RT cores, delivering a Tensor FP16 performance of 330 TFLOPS and a Tensor FP32 performance of 83 TFLOPS, with a power consumption of 450W. This powerful hardware setup allowed for efficient handling of large datasets and complex neural network architectures, significantly accelerating both the training and inference processes.

On the software side, we employed the PyTorch deep learning framework, known for its dynamic computation graph and user-friendly interface. The training process was optimized using the Adam optimizer (optim.Adam), which is well-suited for deep neural networks due to its adaptive learning rate, enhancing both convergence speed and stability. This combination of advanced hardware and sophisticated software tools provided a robust and efficient environment for our computational tasks, ensuring optimal performance throughout the training process.

B   convergence rate

Due to the inclusion of prior physical information in the Optical-DNN and the lightweight nature of the network, our convergence rate is very fast. For both the MNIST and Quickdraw16 datasets,

3

the model converges within 5 epochs. This demonstrates the importance of incorporating physical information into the Optical-DNN.
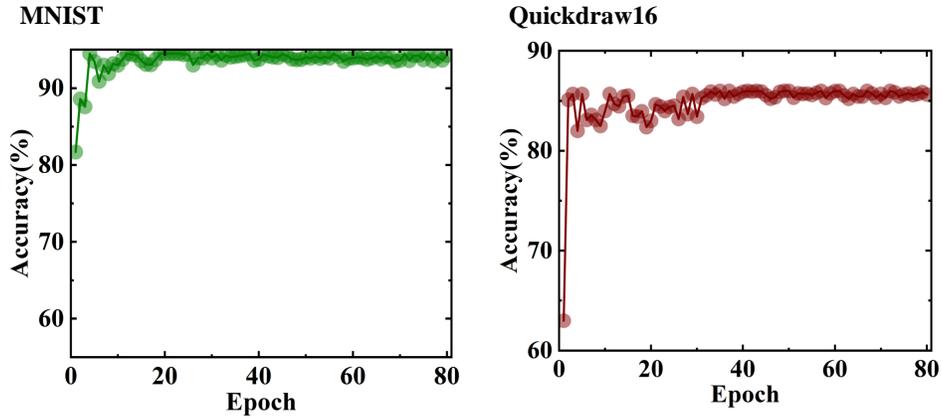
Figure 3: convergence curve of MNIST and Quickdraw16 dataset

C   Additional Experimental Results

For the MNIST dataset, Quickdraw 16, and FMNIST datasets, we present additional recognition results for input images, as well as the output images after error compensation. These results demonstrate that the error compensation network, which leverages physical priors, consistently narrows the gap between simulated output images and experimental output images. This improvement is evident across different types of input images. By effectively compensating for discrepancies between simulations and real-world experiments, our approach significantly enhances the recognition accuracy of the all-optical neural network.
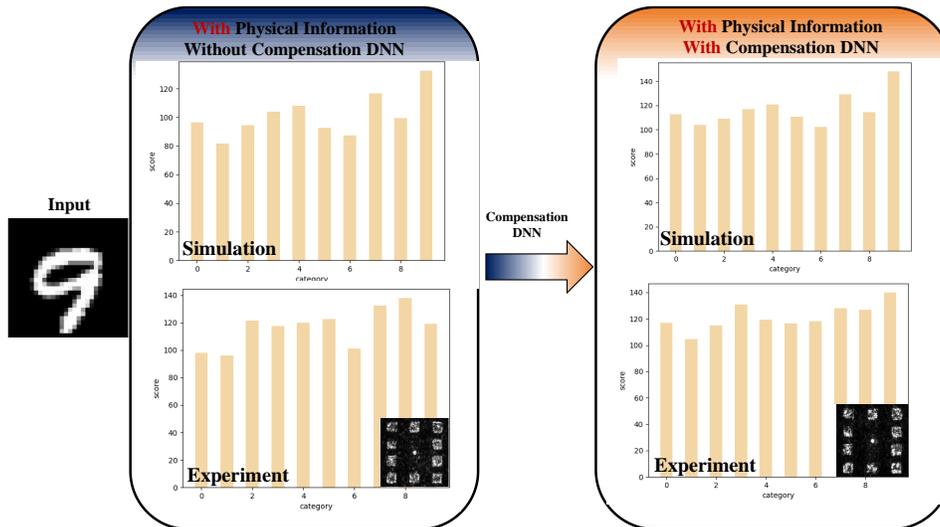
Figure 4: The result of MNIST dataset

4
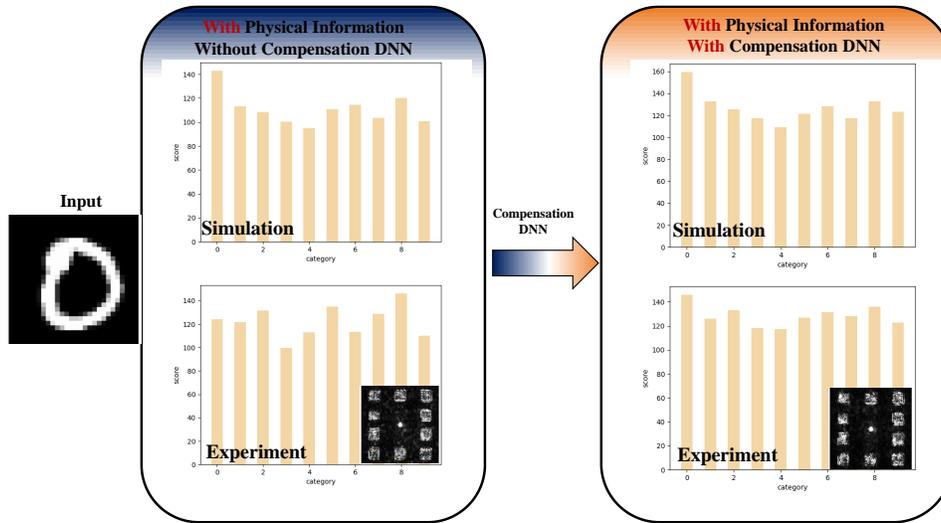
Figure 5: The result of MNIST dataset
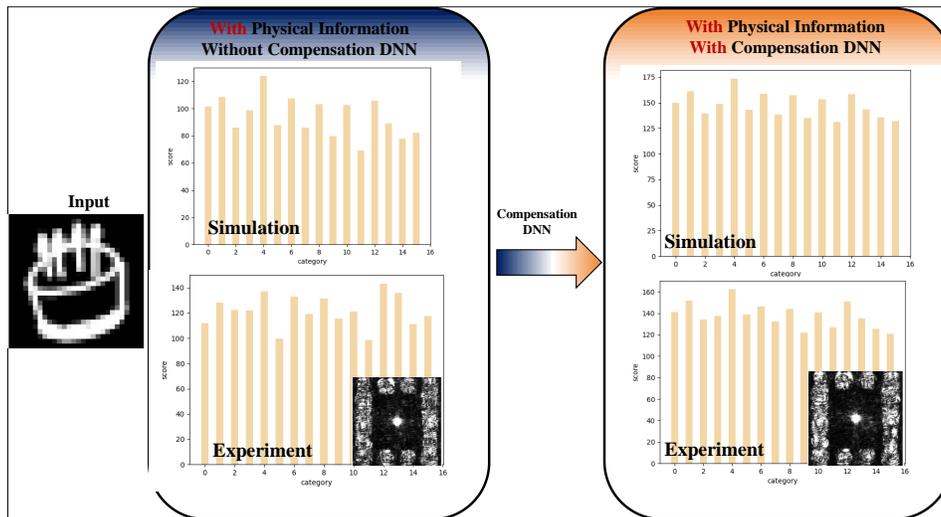
Figure 6: The result of Quickdraw 16 dataset

5
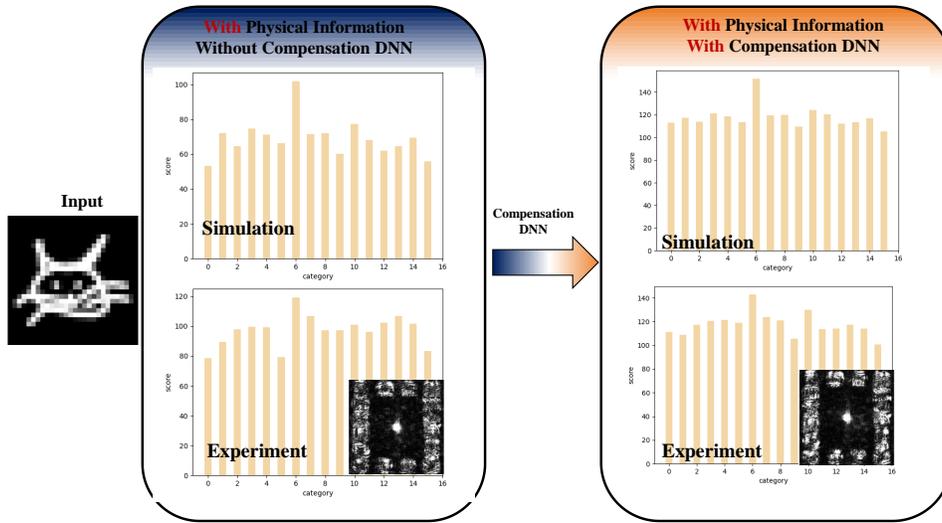
Figure 7: The result of Quickdraw 16 dataset
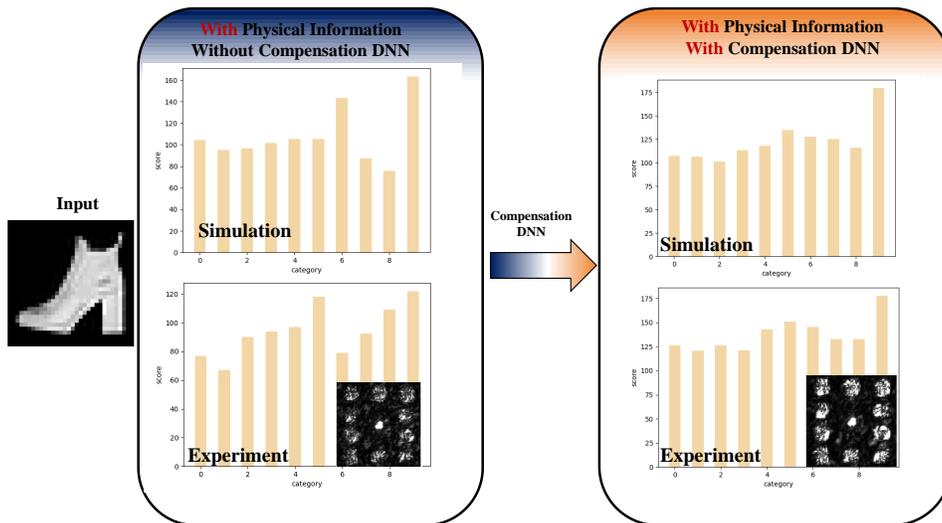
Figure 8: The result of FMNIST dataset

6
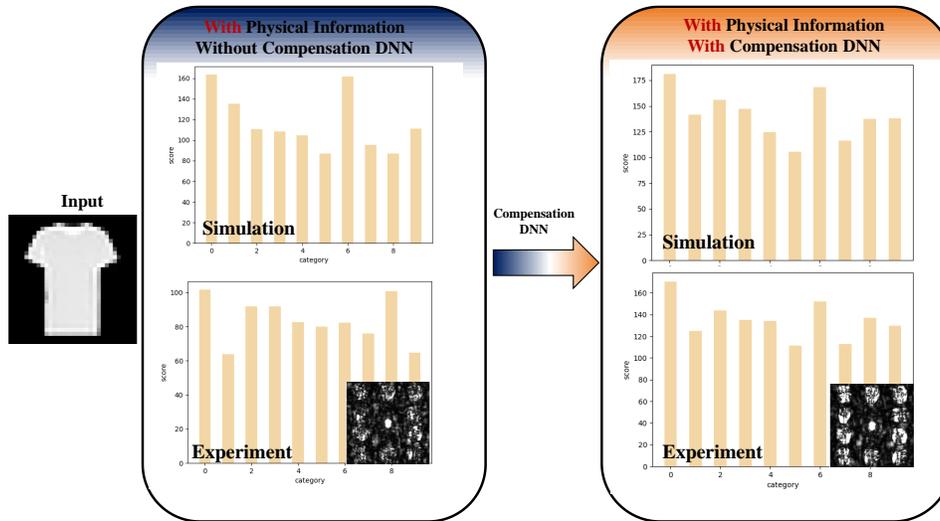
Figure 9: The result of FMNIST dataset

References

[1] J. F. De Boer, C. K. Hitzenberger, and Y. Yasuno, "Polarization sensitive optical coherence tomography–a review," *Biomedical optics express*, vol. 8, no. 3, pp. 1838–1873, 2017.

[2] J. E. Roth, J. A. Kozak, S. Yazdanfar, A. M. Rollins, and J. A. Izatt, "Simplified method for polarization-sensitive optical coherence tomography," *Optics Letters*, vol. 26, no. 14, pp. 1069–1071, 2001.

[3] A. Baumgartner, S. Dichtl, C. Hitzenberger, H. Sattmann, B. Robl, A. Moritz, A. Fercher, and W. Sperr, "Polarization–sensitive optical coherence tomography of dental structures," *Caries research*, vol. 34, no. 1, pp. 59–69, 2000.

7