



# Diffraction deep neural network adjoint assist or (DNA)<sup>2</sup>: a fast and efficient nonlinear diffractive neural network implementation

IGHODALO U. IDEHENRE,<sup>1,2</sup> ERIC S. HARPER,<sup>1,\*</sup>  AND MATTHEW S. MILLS<sup>1</sup> 

<sup>1</sup>Air Force Research Laboratory, 2179 12th St., Wright-Patterson Air Force Base, OH, USA

<sup>2</sup>Azimuth Corporation, 2970 Presidential Dr. # 200, Fairborn, OH 45324, USA

\*eric.harper.4@us.af.mil

**Abstract:** The recent advent of diffractive deep neural networks or D<sup>2</sup>NNs has opened new avenues for the design and optimization of multi-functional optical materials; despite the effectiveness of the D<sup>2</sup>NN approach, there is a need for making these networks as well as the design algorithms more general and computationally efficient. The work demonstrated in this paper brings significant improvements to both these areas by introducing an algorithm that performs inverse design on fully nonlinear diffractive deep neural network - assisted by an adjoint sensitivity analysis which we term (DNA)<sup>2</sup>. As implied by the name, the procedure optimizes the parameters associated with the diffractive elements including both linear and nonlinear amplitude and phase contributions as well as the spacing between planes via adjoint sensitivity analysis. The computation of all gradients can be obtained in a single GPU compatible step. We demonstrate the capability of this approach by designing several types of three layered D<sup>2</sup>NN to classify 8800 handwritten digits taken from the MNIST database. In all cases, the D<sup>2</sup>NN was able to achieve a minimum 94.64% classification accuracy with 192 minutes or less of training.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

## 1. Introduction

There are striking similarities between the mathematics and concepts governing optical wave propagation and those of artificial neural networks (ANNs) [1–4]. Perhaps the most straightforward example is illustrated in the recently reported diffractive deep neural networks (D<sup>2</sup>NN) – a specialized dense ANN prudently constrained to mimic diffractive-scale wave propagation [5]. In the D<sup>2</sup>NN perspective, the nodes, layers, and edges of the network directly correspond to discretized electric field values, propagation distances, and point-source wavelets, respectively. Known as the Huygens-Fresnel principle, electric field values at further distances are found by fully superimposing secondary wavelet contributions emanating from the current plane; this concept is mirrored in D<sup>2</sup>NNs by the repeated application of weighted sums to update node values layer-by-layer.

The benefit of the D<sup>2</sup>NN approach is realized with the introduction of training data as there is no simple analog in the wave propagation perspective. By training the D<sup>2</sup>NN on pairs of object/image plane electromagnetic field profiles to serve as input/output training data, one can effectively design a cascade of passive optical elements that can negotiate multiple propagation requirements [5,6]. Equipped with this powerful capability, the design of all-optical elements have been harnessed, generalized, and improved mostly in the context of classification schemes [7–10]. Beyond the confines of classification, D<sup>2</sup>NNs have also found utility in optics-focused applications including beam steering [11], frequency control [12], and mode differentiation [13–15]. In such a short period of time, the novel material design possibilities and application spaces inspired by D<sup>2</sup>NNs have expanded quickly – yet there still exists gaps in the original methodology. Currently, the main improvement zones fit into two categories: generalizing D<sup>2</sup>NNs and increasing the

computational accuracy of D<sup>2</sup>NNs. Examples of the former include addressing the need to handle true complex numbers rather than independently tracking amplitudes and phases [16], as well as the proper encoding of nonlinear activation functions [17]. Optical analogues to nonlinear activation functions and biases as seen in traditional ANN had been discussed in the supplemental materials of [5], with several promising avenues of study put forward. Despite the successful experimental demonstration of D<sup>2</sup>NN to the MNIST classifier problem, the absence of nonlinearity in the constructed network was a source of contention [18,19]. Many authors would later go on to explore and apply nonlinearity using photorefractive based approaches similar to those initially proposed in [5], resulting in Fourier space networks, residual networks, and in-situ trained networks [20–22]. In each case an increased computational accuracy compared to purely linear networks was observed.

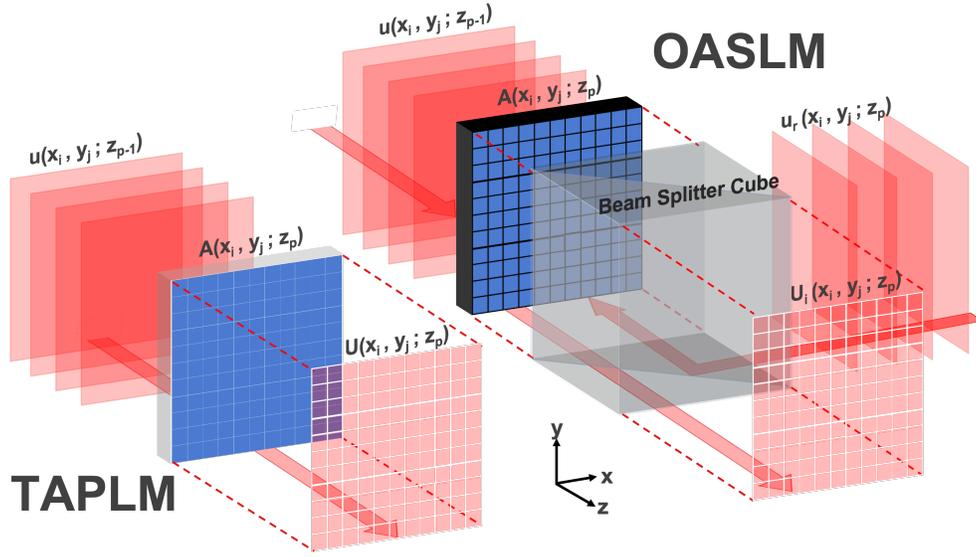
In this work, we demonstrate significant improvements to both the generalization and the computational efficiency of D<sup>2</sup>NNs by introducing an optimization algorithm for optimizing fully nonlinear complex-valued diffractive deep neural network computationally assisted by an adjoint method referred to here as (DNA)<sup>2</sup>. In the original procedure, the most computationally expensive step in each training iteration occurs during the gradient descent calculation. The gradient must be obtained by individually calculating the derivative of the loss function with respect to each complex-valued parameter at each node. These calculations involved repeated applications of a modified version of the the forward propagation algorithm the number of which scaled linearly with the number of parameters being optimized. Instead, (DNA)<sup>2</sup> is derived via an adjoint approach that returns the entire gradient using a single application of a GPU-compatible modified version of the forward propagation algorithm. Simply put, (DNA)<sup>2</sup> removes the linear dependence on the number of parameters entirely. Our adjoint procedure is an advance to those seen previously in D<sup>2</sup>NN [22], being a generalized complex-valued format with an arbitrary parameterized nonlinear activation function. Furthermore, it is compatible with both transmission amplitude and phase light modulators (TAPLM) and optically addressable spatial light modulators (OASLM). Optimization is both simple and efficient for any parameter associated with the linear and nonlinear response of the materials that compose each layer in a D<sup>2</sup>NN as well as the spacing between layers.

## 2. (DNA)<sup>2</sup> for transmissive amplitude and phase light modulators (TAPLM)

Our method applies the perturbative approach outlined in [23] to the problem of scalar diffraction theory. We begin our description of (DNA)<sup>2</sup> with forward propagation. In standard systems, the incident field impinges upon a TAPLM (see Fig. 1). In this regime, the behavior can be described using a discretized form of Rayleigh-Sommerfeld diffraction which models the evolution of scalar electric fields:

$$\begin{aligned}
 u(x_m, y_n; z_{p+1}) &= \sum_i^N \sum_j^N U(x_i, y_j; z_p) H_{mnij}(\Delta z_p) \Delta x \Delta y \\
 U(x_i, y_j; z_p) &= A(x_i, y_j; z_p) u(x_i, y_j; z_p) \\
 H_{mnij}(\Delta z_p) &= \left( \frac{e^{ikr_{mnij}}}{2\pi r_{mnij}} \right) \left( \frac{\Delta z_p}{r_{mnij}} \right) \left( \frac{1}{r_{mnij}} - ik \right)
 \end{aligned} \tag{1}$$

Here,  $u(x_i, y_j; z_p)$  is the 2D array of the transverse complex electric field values impinging upon the  $p^{\text{th}}$  layer of the network at a position  $z_p$ . This array has dimensions  $N \times N$  where  $N$  is the number of discretized nodal points along the  $\{x, y\}$  directions. Each entry is separated by a constant spatial resolution  $\{\Delta x, \Delta y\}$  and tracked by indices  $\{i, j\}$ . Eq. 1 provides the connection between this current network layer and the next hidden layer at a further position  $z_{p+1}$  separated by a propagation distance,  $\Delta z_p = z_{p+1} - z_p$ . The field impinging the next hidden layer,  $u(x_m, y_n; z_{p+1})$ , has the same dimensions and spatial resolution as before, but is tracked



**Fig. 1.** Working principle of the transmissive amplitude and phase modulator (TAPLM) vs the optically addressable spatial-light modulator (OASLM) for  $D^2NN$  applications. A TAPLM modulates the phase and/or amplitude of an incident field  $u(x_i, y_j; z_p)$  via linear or nonlinear processes allowing the optical field to potentially propagate directly from the previous layer to the next. The OASLM, on the other hand, uses an incident field from the previous layer to modulate the phase and/or amplitude of a separate read field  $u_r(x_i, y_j; z_p)$  that is instead propagated forward to then next layer. The majority of the incident field  $u(x_i, y_j; z_p)$  is assumed to be absorbed preventing any of the optical energy from propagating forward.

with a different set of indices  $\{m, n\}$ . Each impinging field encounters an adjustable complex aperture function,  $A(x_i, y_j; z_p)$ , which after interacting generates a field immediately after labeled  $U(x_i, y_j; z_p)$ . Connecting any two consecutive layers is the fixed Rayleigh-Sommerfeld diffraction kernel,  $H_{mnij}(\Delta z_p)$ . The latter is dependent on the radial spacing between subsequent layer nodes,  $r_{mnij} = [(x_i - x_m)^2 + (y_j - y_n)^2 + \Delta z_p^2]^{1/2}$ .

Imposing a convention, we define  $p = 1$  as the input layer at a distance  $z_1 = 0$  and  $p = \mathcal{P}$  as the output layer at a distance  $z_{\mathcal{P}}$ . Eq. 1 is then used to progress through each hidden layer resulting in a network depth equal to  $\mathcal{P} - 1$  layers with adjustable aperture functions at distances  $\{z_1, z_2, z_3, \dots, z_{\mathcal{P}-1}\}$ . Having defined the network architecture, we can introduce training data. The training set size is given by  $Q$  where each  $q \in Q$  contains an input field condition,  $u_q(x_i, y_j; z_1)$ , and a desired target output field condition,  $\bar{u}_q(x_i, y_j; z_{\mathcal{P}})$ . In this notation, the overhead bar conveys that this is the desired or supervised target value. Alternatively, we may also wish to optimize the output to a desired phase. This can be done holographically by interfering  $u_q(x_i, y_j; z_{\mathcal{P}})$  with a reference field  $u_{ref}(x_m, y_n; z_{\mathcal{P}})$ . The interferogram resulting from the superposition of the two fields labeled:

$$u_{int,q}(x_i, y_j; z_p) = u_q(x_m, y_n; z_{\mathcal{P}}) + u_{ref}(x_m, y_n; z_{\mathcal{P}}) \quad (2)$$

is then optimized corresponding to the desired output profile  $\bar{u}_q(x_i, y_j; z_{\mathcal{P}})$ . Lastly, we may want a system where the output power flowing through a  $v^{th}$  target region  $t_v(x_i, y_j; z_{\mathcal{P}})$  is higher than any other from a set of  $\mathcal{V}$  regions for a given  $u_q(x_i, y_j; z_1)$ . With  $v \in \mathcal{V}$ , the total power flowing

through the  $v^{\text{th}}$  specified target region  $P_{q,v}(z_{\mathcal{P}})$  is given by the formula:

$$P_{q,v}(z_{\mathcal{P}}) = \sum_i^N \sum_j^N t_v(x_i, y_j; z_{\mathcal{P}}) |u_q(x_i, y_j; z_{\mathcal{P}})|^2 \Delta x \Delta y \quad (3)$$

With our training parameters defined we can now progress to the core of the training routine (DNA)<sup>2</sup>. Using Eq. 1, for every input layer in  $Q$  we can propagate the evolved field to the output layer and obtain  $u_q(x_m, y_n; z_{\mathcal{P}})$ . We then assign an amount of error relative to the corresponding target via a figure of merit (FOM). The choice of FOM can be important in guiding convergence depending on the scenario; some commonly used intensity based FOM are given in Table 1. Choosing a FOM now guides the backpropagation process – an iterative procedure which adjusts network weights and biases such that the input conditions best match the target across every training pair. The cases of ID, PD, and GS based FOM are analogous to more traditional mean squared errors observed in ANN. The CCE based FOM as its name suggest, combines the softmax function  $\sigma_{q,v}(z_{\mathcal{P}})$  defined here as:

$$\sigma_{q,v}(z_{\mathcal{P}}) = \frac{e^{P_{q,v}(z_{\mathcal{P}})}}{\sum_{v'} e^{P_{q,v'}(z_{\mathcal{P}})}} \quad (4)$$

with categorical cross entropy to maximize the power flow through a given region relative to any other. This FOM is particularly useful for classifiers as it penalizes misclassification where the others (ID, PD, and GS) simply penalize differences between the output field intensity and the target field intensity.

**Table 1. Useful Figures of Merit (FOM) Based on Calculated and Target Fields.**

Name	FOM( $x_i, y_j; z_{\mathcal{P}}$ )
Intensity Difference (ID)	$\frac{1}{Q} \sum_q \left(  u_q(x_i, y_j; z_{\mathcal{P}}) ^2 -  \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) ^2 \right)^2$
Phase Difference (PD)	$\frac{1}{Q} \sum_q \left(  u_{int,q}(x_i, y_j; z_{\mathcal{P}}) ^2 -  \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) ^2 \right)^2$
Gerchberg-Saxton (GS)	$\frac{1}{Q} \sum_q ( u_q(x_i, y_j; z_{\mathcal{P}})  -  \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) )^2$
Categorical Cross Entropy (CCE)	$-\frac{1}{\mathcal{V}} \sum_v \ln(\sigma_{q,v}(z_{\mathcal{P}}))$

Regardless of the FOM in the D<sup>2</sup>NN context, the computational bottleneck is in retrieving the gradient of the FOM with respect to the parameters we wish to optimize such as phase delay and absorption. We denote this gradient function, keeping in mind that it was derived perturbatively, as  $\Delta FOM / \Delta \varrho$ . The variable  $\varrho$  in this context simply denotes the parameter we are trying to optimize and changes to it via our gradient descent algorithm must be sufficiently small for perturbation analysis to be valid [23]. In previous diffractive neural network approaches, the gradient function is cast with respect to the phase or amplitude modulation of each pixel in the previous  $z_{p-1}$  plane via the chain rule. The adjustments are then made by utilizing gradient descent. The standard approach for the backpropagation of errors in the context of D<sup>2</sup>NN is discussed in [5,11] which we will summarize.

The standard gradient descent algorithm is physically equivalent to a point-by-point aperture scan of a pixel sized pinhole coupled with a  $\pi/2$  phase shift at the opening over the pixels elements of a D<sup>2</sup>NN (see Eq.7 in [11]). The phase shifted pinhole is applied to the pixel that we wish to obtain gradients for. To this system, we apply the algorithm used in the forward propagation (i.e. solve Eq. 1) to calculate the corresponding field at the output plane  $\mathcal{P}$  for

a given input. This new field is equivalent to the gradient of the output field with respect to the pixel's phase delay. Similarly, one can obtain field gradients with respect to amplitude modulation of a given pixel by omitting the  $\pi/2$  phase shift on the pinhole. The output field gradient along with the output field itself is then used to calculate the gradient of the FOM with respect to that pixel (see Eq. 6 in [11]). In order to obtain the FOM gradients for every pixel in the  $D^2NN$  system, we must slide the pinhole over each pixel one by one and repeatedly use the forward propagation algorithm to calculate their respective field gradients. Additionally, this process must be repeated for every input image used to calculate the FOM. Thus in order to obtain all FOM gradients for every pixel ( $N^2$ ) on every layer ( $\mathcal{P} - 1$ ) for every input image used ( $Q$ ), the forward algorithm must be invoked  $N^2 \times Q \times (\mathcal{P} - 1)$  times. For optimizing large  $D^2NN$  this may require a large amount of computational resources and be very time consuming even for the most efficient of forward propagation routines.

Our procedure moving forward seeks to avoid this computational bottleneck while also being sufficiently general to incorporate nonlinearity. Adjoint methods excel in the electromagnetic inverse design space because for every input they **only** need a single invocation of the forward algorithm to obtain **all** the field gradients with respect to **all** design parameters regardless of the number being optimized [23–25]. This is done by backpropagating an adjoint field through the system, which like the standard approach, is combined with the fields calculated during the forward propagation process to obtain FOM gradients. By using this adjoint approach the number of times the forward routine needs to be invoked decreases from  $N^2 \times Q \times (\mathcal{P} - 1)$  to  $Q$ .

In the following, we drop the  $q$ -subscripts used in the fields and intensity terms with the implicit understanding that the network is only completely trained after it has been repeated for each  $q \in Q$ . The  $(DNA)^2$  formulation allows for both intensity dependent and independent amplitudes and phases at each hidden layer. This is encoded into the complex aperture function by:

$$\begin{aligned}
 A(x_i, y_j; z_p) &= A_1(x_i, y_j; z_p) * A_2(x_i, y_j; z_p) \\
 A_1(x_i, y_j; z_p) &= \exp[-if_1(x_i, y_j; z_p)] \\
 A_2(x_i, y_j; z_p) &= \exp[f_2(x_i, y_j; z_p)] \\
 f_1(x_i, y_j; z_p) &= g_1(h_1(x_i, y_j; z_p)) \\
 f_2(x_i, y_j; z_p) &= \ln[g_2(h_2(x_i, y_j; z_p))] \\
 h_1(x_i, y_j; z_p) &= w_1(x_i, y_j; z_p)I(x_i, y_j; z_p) + b_1(x_i, y_j; z_p) \\
 h_2(x_i, y_j; z_p) &= w_2(x_i, y_j; z_p)I(x_i, y_j; z_p) + b_2(x_i, y_j; z_p)
 \end{aligned} \tag{5}$$

Eq. 5 is first segmented into two intermediate exponential functions,  $A_1(x_m, y_n; z_p)$  and  $A_2(x_m, y_n; z_p)$ , exponent functions,  $f_1(x_m, y_n; z_p)$  and  $f_2(x_m, y_n; z_p)$ , which track aperture phases and amplitudes respectively. Defining  $A_2/f_2$  in this manner, though seemingly redundant, is useful for maintaining a mathematical symmetry between the phase and amplitude further on in the description. The intensity dependency is introduced by two auxiliary functions,  $h_1(x_i, y_j; z_p)$  and  $h_2(x_i, y_j; z_p)$ , which are immediately transformed by two activation functions,  $g_1$  and  $g_2$ . For our simulations, we have the most robust results with ReLU ( $g_1$ ) and clipped ReLU ( $g_2$ ) activation functions:

$$g_1(h_1(x_i, y_j; z_p)) = \begin{cases} 0 & h_1(x_i, y_j; z_p) \leq 0 \\ h_1(x_i, y_j; z_p) & 0 < h_1(x_i, y_j; z_p) \end{cases} \tag{6}$$

$$g_2(h_2(x_i, y_j; z_p)) = \begin{cases} 0 & h_2(x_i, y_j; z_p) \leq 0 \\ h_2(x_i, y_j; z_p) & 0 < h_2(x_i, y_j; z_p) \leq 1 \\ 1 & 1 < h_2(x_i, y_j; z_p) \end{cases} \tag{7}$$

While there exist numerous choices of  $g_1$  and  $g_2$ , this construction avoids phase ambiguities caused by negative values and the potential for field amplitudes to approach infinity. Additionally,

the simple form of  $g_1$  and  $g_2$  also extends to their derivatives allowing for faster calculations. The weights  $w_1(x_i, y_j; z_p)$  and  $w_2(x_i, y_j; z_p)$  and biases  $b_1(x_i, y_j; z_p)$  and  $b_2(x_i, y_j; z_p)$  are the terms that we ultimately seek to optimize. As their name suggest, these parameters are analogous to those used in ANN, with the key difference being that the weights are multiplied by real valued intensities as opposed to the complex valued fields. This choice was made because many of the most common and likely candidates for nonlinear materials such as inorganic/organic photorefractives [26,27] and photochromic dyes [28] are driven primarily by optical absorption phenomena. In the absence of any optical nonlinearity, the bias terms correspond to linear phase delay and linear absorption/reflection.

The formulation relating the FOM to the parameters each layer, can now be expressed as an adjoint aperture function  $\epsilon(x_i, y_j; z_p)$  defined as:

$$\epsilon(x_i, y_j; z_p) = \begin{cases} \frac{\partial \text{FOM}(x_i, y_j; z_p)}{\partial I(x_i, y_j; z_p)} & p = \mathcal{P} \\ 2\Re \left[ \frac{\partial A(x_i, y_j; z_p)}{\partial I(x_i, y_j; z_p)} u(x_i, y_j; z_p) \hat{u}(x_i, y_j; z_p) \right] & p \neq \mathcal{P} \end{cases} \quad (8)$$

Here,  $I(x_i, y_j; z_p) = |u(x_i, y_j; z_p)|^2$ , the layer's field intensity. Denoted with a hat,  $\hat{u}(x_i, y_j; z_p)$  is the backward-propagating adjoint field impinging upon the  $p^{\text{th}}$  layer. The key step now arrives: instead of performing the usual gradient descent, we define the backward-propagating adjoint field as:

$$\hat{U}(x_i, y_j; z_p) = A(x_i, y_j; z_p) \hat{u}(x_i, y_j; z_p) + \epsilon(x_i, y_j; z_p) u(x_i, y_j; z_p)^* \quad (9)$$

where  $\hat{U}(x_i, y_j; z_p)$  is the adjoint field immediately after the aperture functions on the  $p^{\text{th}}$  layer. It can be thought of as virtual field arrangements passing into a FOM error mask such that it carries information about the error backward to the plane of interest. The same Rayleigh-Sommerfeld formalism is employed to calculate adjoint fields as that used for the scalar electric fields. Note that at the last layer  $\hat{u}(x_i, y_j; z_{\mathcal{P}}) = 0$ . Propagating backwards, we can now define the adjoint fields using the adjoint Rayleigh-Sommerfeld equation:

$$\hat{u}(x_m, y_n; z_{p-1}) = \sum_i^N \sum_j^N \hat{U}(x_i, y_j; z_p) H_{mij}(\Delta z_{p-1}) \quad (10)$$

The diffraction kernel  $H_{mij}(\Delta z_{p-1})$ , given in Eq. 1, remains unchanged despite the reversal of propagation as a direct consequence of Helmholtz reciprocity. We obtain the sought out gradient by combining the information from the forward propagation (electric field) with the backward propagation (adjoint field). In this way, the previously mentioned costly point-by-point aperture scan is avoided. The derivative of the FOM with respect to any of the mask parameters ( $\varrho = \{w_1, w_2, b_1, b_2\}$ ) at a given position is:

$$\frac{\Delta \text{FOM}(x_i, y_j; z_p)}{\Delta \varrho(x_i, y_j; z_p)} = 2\Re \left[ \frac{\partial A(x_i, y_j; z_p)}{\partial \varrho(x_i, y_j; z_p)} u(x_i, y_j; z_p) \hat{u}(x_i, y_j; z_p) \right] \quad (11)$$

We pause here briefly to highlight some interesting observations on the nature of adjoint fields. As others have demonstrated [22,29], the adjoint field can be physically "simulated" using the electromagnetic field and used to optimize transmission masks. It should be noted however, that the physics of the adjoint field and the electromagnetic field are identical only in the linear case (i.e.  $\partial A / \partial I = 0$  and  $\epsilon(x_i, y_j; z_p) = 0$ ). The reason lies in the response of fields to boundary conditions, from which the Rayleigh-Sommerfeld equation was formulated to address. In the linear instance, the adjoint field responds to linear boundary conditions imposed by  $A(x_i, y_j; z_p)$  (see Eq. 9) in the same way the scalar electric field does (see Eq. 1). However, when  $A(x_i, y_j; z_p)$  has a nonlinear component, an additional term incorporating  $\epsilon(x_i, y_j; z_p)$  and the

complex conjugate of the scalar electric field must be included. Clearly, no such modification is needed when propagating the scalar electric field. In summary, the physics of propagation of the adjoint field and the scalar electric field are identical in free space (Eq. 1 and Eq. 10), but the material response which appears in the boundary conditions is not necessarily the same. Naturally these differences must be contained in the constitutive relations that govern both fields.

This has implications on real world attempts to perform all optical in-situ training. In principle the adjoint field can be “simulated” using the scalar electric field so long as it has the same phase and amplitude described in Eq. 1 is used. This synthetic adjoint field however, cannot propagate through the nonlinear components of the network in the manner required for the “real” adjoint field. In order to enforce the proper behavior additional measures are required to manage the discrepancy. For example in [22,29] the synthetic adjoint field is allowed to propagate through the linear components of the network only. When a nonlinear component is encountered, the synthetic adjoint field is calculated electronically on a computer using data collected from both the forward and backward propagation. The calculated synthetic adjoint field is then injected from an external optical source backward into the next linear component. The need for these awkward electronic calculations can be attributed to the direct transmission of the electromagnetic field between layers. Ideally, we would like to avoid such cumbersome electronic operations or replace them with passive optical analogues. In the next section we discuss (DNA)<sup>2</sup> optically addressable spatial light modulators, where the optical fields are not transmitted directly between layers. As we will show, the formulation of the adjoint field is greatly simplified serving as a possible alternative to published in situ methods.

### 3. (DNA)<sup>2</sup> for optically addressable spatial light modulators (OASLM)

In OASLM systems [30–32], the incident field is not transmitted through the mask (see Fig. 1). Instead, the incident field (“write beam”) is used to modulate the phase and/or amplitude of a separate reference field (“read beam”) that is propagated forward to a desired position. In this regime we use a modified form of the discretized Rayleigh-Sommerfeld diffraction equation:

$$u(x_m, y_n; z_{p+1}) = \sum_i^N \sum_j^N U(x_i, y_j; z_p) H_{nmij}(\Delta z_p) \Delta x \Delta y \quad (12)$$

$$U(x_i, y_j; z_p) = A(x_i, y_j; z_p) u_r(x_i, y_j; z_p)$$

Here,  $u_r(x_i, y_j; z_p)$  corresponds to a user controlled field that reads the hologram recorded by the OASLM via the interaction of the input field and the nonlinear complex aperture function  $A(x_i, y_j; z_p)$ . For simplicity, we have assumed this field to be a planewave at normal incidence (i.e.  $u_r(x_i, y_j; z_p) = 1$ ). We can use the adjoint Rayleigh-Sommerfeld formula given in Eq. 10 to calculate the adjoint fields as with the TAPLM case. In this scenario however, the adjoint aperture function  $\epsilon(x_i, y_j; z_p)$  is defined as:

$$\epsilon(x_i, y_j; z_p) = \begin{cases} \frac{\partial FOM(x_i, y_j; z_p)}{\partial I_q(x_i, y_j; z_p)} & p = \mathcal{P} \\ 2\Re \left[ \frac{\partial A(x_i, y_j; z_p)}{\partial I_q(x_i, y_j; z_p)} \hat{u}(x_i, y_j; z_p) \right] & p \neq \mathcal{P} \end{cases} \quad (13)$$

The backpropagating adjoint field immediately after the aperture function (i.e. side facing towards the  $z_1$  plane) is defined as:

$$\hat{U}(x_i, y_j; z_p) = \epsilon(x_i, y_j; z_p) u(x_i, y_j; z_p)^* \quad (14)$$

Lastly, the derivative of the FOM with respect to any of the mask parameters at a given position is:

$$\frac{\Delta FOM(x_i, y_j; z_p)}{\Delta \varrho(x_i, y_j; z_p)} = 2\Re \left[ \frac{\partial A(x_i, y_j; z_p)}{\partial \varrho(x_i, y_j; z_p)} \hat{u}(x_i, y_j; z_p) \right] \quad (15)$$

As discussed in the previous section, there is a possibility of in-situ training of optical neural networks by using a synthetic adjoint field. When we compare the adjoint field in the OASLM case (Eqs. 13–15) to the TAPLM case (Eqs. 8–11), a number of useful simplifications emerge. By decoupling the scalar electric fields in the forward propagation between layers, we no longer need to explicitly measure  $u(x_i, y_j; z_p)$  to calculate any of the gradients. The complex conjugate of the field is still technically needed to produce the adjoint field (Eq. 14). Even then, only the aperture function  $\epsilon(x_i, y_j; z_p)$  need be calculated as the role of generating  $u(x_i, y_j; z_p)^*$  could potentially be offloaded to a passive photorefractive phase conjugate mirror [33]. In this way OASLM can provide a simpler and faster architecture for in-situ learning for systems that incorporate nonlinearity.

#### 4. (DNA)<sup>2</sup> optimization of spacing between planes

The adjoint fields in either the TAPLM or OASLM geometries can also be used to optimize the spacing between planes  $\Delta z_p$  as well. This adjoint field, like those mentioned previously, backpropagates information about the error with respect to said spacing. This adjoint field when propagating to a prior plane is defined as  $\hat{u}_{\Delta z}(x_m, y_n; z_{p-1})$  and is obtained using the following formula:

$$\hat{u}_{\Delta z}(x_m, y_n; z_{p-1}) = \sum_i^N \sum_j^N \hat{U}(x_i, y_j; z_p) \frac{\partial H_{mij}(\Delta z_{p-1})}{\partial s(x_m, y_n; z_{p-1})} \Delta x \Delta y \quad (16)$$

$$\Delta z_p = Z(s(x_m, y_n; z_{p-1}))$$

Here  $s(x_m, y_n; z_{p-1})$  is a spacing parameter that can be optimized with respect to a spacing function  $Z$  and associated with each pixel in the plane  $p - 1$ . The spacing function is user defined and can be sigmoidal so that all possible values of  $\Delta z_p$  will be bounded between two distances  $\Delta z_{min}$  and  $\Delta z_{max}$ . In this paper we define the spacing function using a clipped ReLU:

$$Z(s(x_i, y_j; z_p)) = \begin{cases} \Delta z_{min} & s(x_i, y_j; z_p) \leq 0 \\ \Delta z_{min} + (\Delta z_{max} - \Delta z_{min}) s(x_i, y_j; z_p) & 0 < s(x_i, y_j; z_p) \leq 1 \\ \Delta z_{max} & 1 < s(x_i, y_j; z_p) \end{cases} \quad (17)$$

The adjoint field  $\hat{U}(x_i, y_j; z_p)$  used in the equation depends on the type of D<sup>2</sup>NN system and can either be that defined in Eq. 9 for TAPLMs or in Eq. 14 for OASLMs. The gradient of the FOM with respect to the spacing parameter is given by:

$$\frac{\Delta FOM(z_p)}{\Delta s(z_p)} = \frac{1}{N^2} \sum_i^N \sum_j^N \frac{\Delta FOM(x_i, y_j; z_p)}{\Delta s(x_i, y_j; z_p)} \quad (18)$$

$$\frac{\Delta FOM(x_i, y_j; z_p)}{\Delta s(x_i, y_j; z_p)} = 2\Re [\hat{u}_{\Delta z}(x_i, y_j; z_p)]$$

Note that because the Rayleigh-Sommerfeld model assumes all pixels in a TAPLM/OASLM element exist at the same value  $z_p$ . As such we can only have a single value for the spacing parameter defined here as  $s(z_p)$  where  $s(x_i, y_j; z_p) = s(z_p)$ . It is unlikely however, that each individual pixel will have the same optimal value for  $s(z_p)$ . A more likely scenario is that the gradient will have a spatial dependence owing to a non-uniform  $\hat{u}_{\Delta z}(x_i, y_j; z_p)$ . Thus the gradient is instead defined as the spatial average which resolves the matter.

#### 5. Adjoint aperture functions and various derivatives

As observed in the previous sections, the adjoint aperture functions  $\epsilon(x_i, y_j; z_p)$  exist for both the TAPLM and OASLM regimes and are used to propagate information about the error backwards

through the system. At the output plane  $\epsilon(x_i, y_j; z_{\mathcal{P}})$  is given by the derivative of the figure of merit  $FOM(x_i, y_j; z_{\mathcal{P}})$  with respect to the field intensity  $I(x_i, y_j; z_{\mathcal{P}})$ . A list of aperture functions associated with the various FOMs is given in Table 2. Note that in the case of PD we must substitute  $u(x_i, y_j; z_{\mathcal{P}})^* \rightarrow u_{int,q}(x_i, y_j; z_{\mathcal{P}})^*$  in Eq. 9 and Eq. 14. We note here that as the FOM becomes optimized with each iteration, the aperture function becomes more opaque resulting in ever weaker adjoint fields. This is to be expected, as the adjoint fields are directly related to the gradients (see Eqs. 11,14, and 18). In a perfectly optimized system  $\Delta FOM / \Delta \varrho = 0$  and thus for all adjoint fields  $\dot{U}(x_i, y_j; z_{\mathcal{P}}) = 0$ .

**Table 2. Aperture Functions for Various Figures of Merit (FOM).**

Name	$\epsilon(\mathbf{x}_i, \mathbf{y}_j; z_{\mathcal{P}})$
Intensity Difference (ID)	$2 \left(  u_q(x_i, y_j; z_{\mathcal{P}}) ^2 -  \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) ^2 \right)$
Phase Difference*(PD)	$2 \left(  u_{int,q}(x_i, y_j; z_{\mathcal{P}}) ^2 -  \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) ^2 \right)$
Gerchberg-Saxton (GS)	$\left( 1 - \frac{ \bar{u}_q(x_i, y_j; z_{\mathcal{P}}) }{ u_q(x_i, y_j; z_{\mathcal{P}}) } \right)$
Categorical Cross Entropy (CCE)	$\sum_{v'} t_{v'}(x_i, y_j; z_{\mathcal{P}}) \sigma_{q,v'}(z_{\mathcal{P}}) - t_v(x_i, y_j; z_{\mathcal{P}})$

At the intermediate planes where the elements of the TAPLM/OASLM lay,  $\epsilon(x_i, y_j; z_p)$  is related to the derivative of the complex aperture function  $A(x_i, y_j; z_p)$  with respect to  $I(x_i, y_j; z_p)$ :

$$\frac{\partial A(x_i, y_j; z_p)}{\partial I(x_i, y_j; z_p)} = \sum_{m=1}^2 a_l w_l(x_i, y_j; z_p) A(x_i, y_j; z_p) \frac{\partial g_l(h_m(x_i, y_j; z_p))}{\partial h_l(x_i, y_j; z_p)} \quad (19)$$

$$a_l = \begin{cases} -i & l = 1 \\ 1 & l = 2 \end{cases}$$

For the activation functions  $g_1(h_1(x_i, y_j; z_p))$  and  $g_2(h_2(x_i, y_j; z_p))$  given previously, the derivatives with respect to  $h_1(x_i, y_j; z_p)$  and  $h_2(x_i, y_j; z_p)$  are :

$$A(x_i, y_j; z_p) \frac{\partial g_1(h_1(x_i, y_j; z_p))}{\partial h_1(x_i, y_j; z_p)} = \begin{cases} 0 & h_1(x_i, y_j; z_p) \leq 0 \\ A(x_i, y_j; z_p) & 0 < h_1(x_i, y_j; z_p) \end{cases} \quad (20)$$

$$A(x_i, y_j; z_p) \frac{\partial g_2(h_2(x_i, y_j; z_p))}{\partial h_2(x_i, y_j; z_p)} = \begin{cases} 0 & h_2(x_i, y_j; z_p) \leq 0 \\ A_1(x_i, y_j; z_p) & 0 < h_2(x_i, y_j; z_p) \leq 1 \\ 0 & 1 < h_2(x_i, y_j; z_p) \end{cases} \quad (21)$$

In the TAPLM case, the complete absence of nonlinearity (i.e.  $w_1(x_i, y_j; z_p) = w_2(x_i, y_j; z_p) = 0$ ) results in a  $\epsilon(x_i, y_j; z_p) = 0$  at the given intermediate position. The implication is that only the linear component (the first term on the R.H.S. of Eq. 9) contributes to the gradient. In the OASLM case, there is no such linear term meaning that the gradient is necessarily zero. For this reason we cannot backpropagate any information about the error in the absence of nonlinearity beyond the last mask layer. As one would expect, having a system of purely linear OASLMs (i.e. just SLMs) with more than one layer is of no benefit.

The gradients with respect to the weight and bias parameters that we seek to calculate (Eq. 11 and Eq. 15) contain derivatives defined as:

$$\frac{\partial A(x_i, y_j; z_p)}{\partial w_l(x_i, y_j; z_p)} = a_l I(x_i, y_j; z_p) A(x_i, y_j; z_p) \frac{\partial g_l(h_l(x_i, y_j; z_p))}{\partial h_l(x_i, y_j; z_p)} \quad (22)$$

$$\frac{\partial A(x_i, y_j; z_p)}{\partial b_l(x_i, y_j; z_p)} = a_l A(x_i, y_j; z_p) \frac{\partial g_l(h_l(x_i, y_j; z_p))}{\partial h_l(x_i, y_j; z_p)} \quad (23)$$

The gradient equation for the spacing parameter Eq. 18 contains a derivative defined as:

$$\frac{\partial H_{mnij}(\Delta z_p)}{\partial s(x_i, y_j; z_p)} = \frac{\partial H_{mnij}(\Delta z_p)}{\partial \Delta z_p} \frac{\partial \Delta z_p}{\partial s(z_p)} \quad (24)$$

Having previously defined  $\Delta z_p$  using the spacing function  $Z(s(z_p))$  given by Eq. 17 we can then define  $\partial \Delta z_p / \partial s$  as:

$$\frac{\partial Z(s(z_p))}{\partial s(z_p)} = \begin{cases} 0 & s(z_p) \leq 0 \\ \Delta z_{max} - \Delta z_{min} & 0 < s(z_p) \leq 1 \\ 0 & 1 < s(z_p) \end{cases} \quad (25)$$

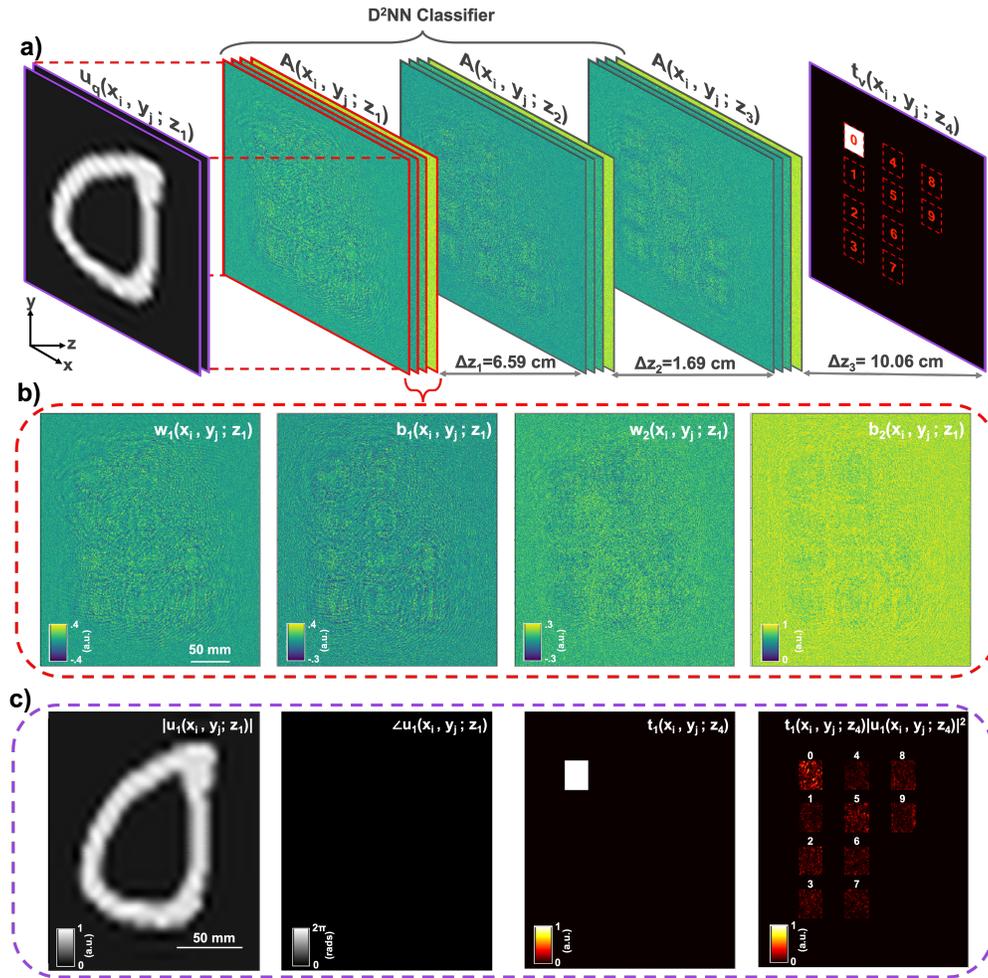
Here we do not give an explicit definition of  $\partial H_{mnij} / \partial \Delta z_p$ , though it can be easily obtained, because we used an FFT accelerated Angular-Spectrum method or FFT-AS [34] to solve both the Rayleigh-Sommerfeld and adjoint Rayleigh-Sommerfeld equations. In the frequency domain,  $H_{mnij}(\Delta z_p)$  is represented by a plane wave spectrum  $e^{ik_z \Delta z_p}$ . Therefore, the derivative with respect to  $\Delta z_p$  is readily calculated in the frequency domain using  $ik_z e^{ik_z \Delta z_p}$  in place of  $e^{ik_z \Delta z_p}$  for FFT-AS based approaches.

## 6. Example simulations and comparisons

In this section we demonstrate the (DNA)<sup>2</sup> technique for designing TAPLM and OASLM based D<sup>2</sup>NN, by applying them to the classifier problem for handwritten digits. Image data was drawn from the MNIST database [35] and the the physical parameters of the D<sup>2</sup>NN were taken from the supplemental material of [5]. Simulations for TAPLM and OASLM were done optimizing both linear and nonlinear phase and amplitude parameters as well as the spacing between planes (see Fig. 2). All simulation parameters described in this section are given in Table 3.

**Table 3. Summary of the Parameters Used in Our Simulations.**

Simulation Parameter Description	Symbol	Value
Resolution of $x$	$\Delta x$	400 $\mu\text{m}$
Resolution of $y$	$\Delta y$	400 $\mu\text{m}$
Simulation Window	$M \times M$	800 x 800
Mask/SLM Window	$N \times N$	400 x 400
Min Distance Between Planes	$\Delta z_{min}$	1 cm
Max Distance Between Planes	$\Delta z_{max}$	30 cm
Wavelength of Light	$\lambda$	749 $\mu\text{m}$
Number of Layers	$\mathcal{P}$	4
Optimization Parameter Description		Value
Mini-batch Size		20
Training Set Size		50000
Validation Set Size		4000
Testing Set Size		8800
Batch Size		1000
Iterations		100
Epochs		75
Learning Rate		$1 \times 10^{-3}$
ADAM Decay Rate		.9



**Fig. 2.** Example diffractive deep neural networks ( $D^2NN$ ) and results obtained using (DNA)<sup>2</sup>. a) A TAPLM2 based  $D^2NN$  numerical digit classifier comprised of 3 layers of nonlinear phase/amplitude activation functions. b) The weight and bias coefficients associated with the activation functions of each layer are trained via the (DNA)<sup>2</sup> algorithm using input data taken from the MNIST database. The algorithm optimizes the coefficients to perform classification between the input and output planes of the network. c) After this training phase, the  $D^2NN$  design is tested against a set of never before seen data. Note that in our study only the amplitude  $|u_q(x_i, y_j; z_1)|$  of the input field is modulated and normalized to a maximum value of unity. The phase profile ( $\angle u_q(x_i, y_j; z_1)$ ) is kept flat at 0. The maximum intensity of the output field has also been normalized to unity in the figure. As is typical for each  $D^2NN$ , the CCE FOM allows some energy to impinge all target regions in the output plane. However, the correct target region is consistently and visibly more intense than all others allowing for a 95.82% accuracy.

Additionally, simulations for a purely linear phase mask TAPLM system were done for comparison. In the purely linear TAPLM case, referred to as TAPLM1, only the linear phase parameters  $b_1(x_i, y_j; z_p)$  are optimized. The linear amplitude parameters of each layer are set to unity ( $b_2(x_i, y_j; z_p) = 1$ ) to allow for full transmission of the field, while the nonlinear phase and

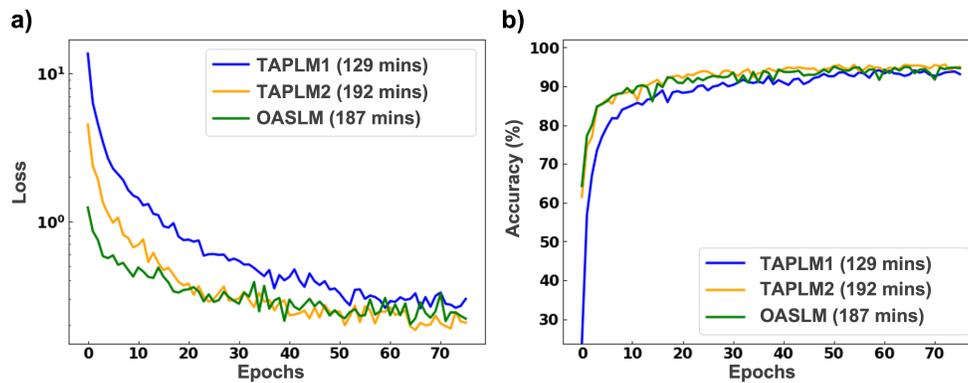
amplitude parameters are set to zero ( $w_1(x_i, y_j; z_p) = w_2(x_i, y_j; z_p) = 0$ ). In the transmission mask simulations referred to as TAPLM2, all parameters on each mask are optimized. The simulation parameters of the OASLM are likewise optimized. In all three cases, all parameters including those related to spacing were randomly initialized. The spacing between planes was constrained to a minimum value  $\Delta z_{min}$  set to 1 cm and a maximum value  $\Delta z_{max}$  set to 30 cm. Pixel resolutions for  $\Delta x$  and  $\Delta y$  were set to  $400\mu m$  with the number of pixels ( $N_x \times N_y$ ) in the input images up-sampled to  $400 \times 400$  pixels. The resulting images were normalized so that the maximum pixel value is always unity and the minimum zero. The amplitude of the input fields ( $|u_q(x_i, y_j; z_1)|$ ) are set equal to the input image, while the phase of the field ( $\angle u_q(x_i, y_j; z_1)$ ) is set to 0 everywhere. The total number of images in the training set, validation set, and testing set were 50000, 4000, and 8800 respectively. These numbers were chosen to maximize the MNIST data used while also ensuring that each of the 10 digit categories (0-9) contribute an equal number of images to the total in each set. The target regions are  $21.3 \mu m \times 21.3 \mu m$  squares assigned to each digit as shown in Fig. 2.

The method used to perform forward and backward propagation is a variant of the FFT-AS method described in [34]. This method zero-pads the simulation window to  $M \times M$  and apodizes the plane wave spectrum in the frequency domain to suppress aliasing. The simulation window used was  $800 \times 800$ . The region around the  $400 \times 400$  containing TAPLM/OASLM in the simulations were assumed to be opaque apertures where any incident light was fully absorbed. All parameters were optimized using a combination of mini-batch gradient decent and the Adaptive Moment Estimation (ADAM [36]) algorithms. The training set was subdivided into five batches of 1000 randomly selected images (100 from each category). A smaller mini-batch ( $Q = 20$ ) is randomly selected from the larger batch and a forward and backward simulation is performed and used to calculate the gradients. For both the TAPLM and OASLM cases, categorical cross entropy was used to calculate the loss. The gradients are averaged over the mini-batch and fed into the ADAM algorithm where it is used to update the simulation parameters. The learning rates and decay rates used in the ADAM algorithm for the momentum and velocity were  $10^{-3}$  and .9 respectively for all parameters. This process is repeated for 100 iterations for each batch with the the loss and accuracy of the  $D^2NN$  system calculated against the 4000 images in the validation set ( $Q = 4000$ ). Accuracy is determined by taking the ratio of correct predictions over total predictions for a given dataset ( $Q$ ). In our simulations, the predicted digit value ( $v$ ) for a given input field ( $q$ ) is determined by which target region has the highest optical power (i.e.  $\max_v P_{q,v}(z, \phi)$ ).

The validation set calculation defines the end of 1 epoch. A new batch of training data is generated and the aforementioned optimization process begins again. For our simulations the optimization process consisted of 75 epochs with the final  $D^2NN$  system selected from the epoch with the highest accuracy value when measured against the validation set. At the end of optimization the loss and accuracy of each  $D^2NN$  system is calculated against the 8800 images in the testing set ( $Q = 8800$ ). The code used in our simulation was written in Python and implemented on a AMD Ryzen 9 3900x 12 core CPU system with 64 GB of RAM and a NVidia RTX-2070 Super GPU with 8 GB of RAM. Simulation code was written in house using CuPy [37] to perform forward and backward simulations on the GPU.

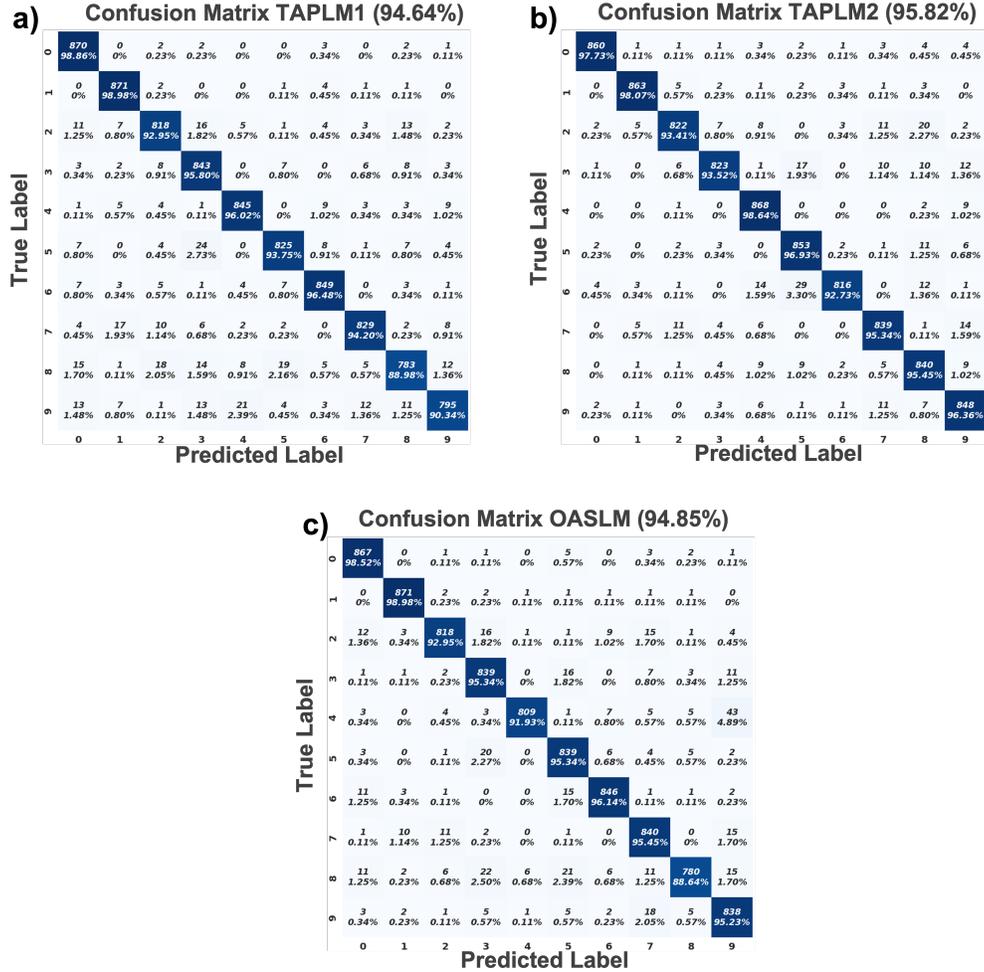
The evolution of the loss and accuracy against the validation set verses the simulation epochs of the TAPLM1, TAPLM2, OASLM are shown in Fig. 3. Included as well as are the total training and validation time of each scenario. All three of the simulations converged quickly to a solution with the loss and accuracy being roughly inversely proportional. The parameters that yielded the best accuracy results were selected for the final testing phase. The best accuracy values were 94.2% for TAPLM1, 95.7% for TAPLM2, and 95.1% for OASLM. The TAPLM1 case required the least amount of time (129 minutes) as it only required optimizing the 3 spacing parameters and linear phase components (3 layer's bias components). The TAPLM2 (192 minutes) and

OASLM (187 minutes) simulations, on the other hand, needed to optimize 3 spacing parameters and 12 weight and bias components of all the layers. Even with a roughly 4-fold increase in the number of parameters being optimized, the increase in simulation times for both TAPLM2 and OASLM were no more than 1.5 times that of the TAPLM1 case. This modest increase in simulation time despite the significant increase in design parameters highlights the main benefit of the adjoint methods. Another benefit that contributes to efficiency are the natural recursions present in the  $(DNA)^2$  algorithm, a feature common to most adjoint methods. Many terms such as the complex aperture functions and intensities are used in both the forward and backward propagation routines and thus only need to be calculated once. Additionally, terms that are unique to the backward calculations such as  $\partial A / \partial \rho$  and  $\partial A / \partial I$  are relatively fast to calculate due to the simple form of the activation functions  $g_1$  and  $g_2$ . As a final note, the OASLM were observed to slightly out perform the TAPLM2 in terms of simulation time which we attribute to the simpler nature of the OASLM calculations. As mentioned previously, when comparing the equations for the adjoint fields (Eq. 9 vs Eq. 14) and the gradients (Eq. 11 and Eq. 15), the OASLMs require fewer operations and thus should take less time.



**Fig. 3.** a) The categorical cross entropy loss (CCE) vs epochs and b) accuracy vs epochs for the TAPLM1, TAPLM2, and OASLM over the 4000 image validation set. Of the three, the TAPLM1 simulation required the least amount of time (129 minutes), followed by the OASLM (187 minutes), and TAPLM2 (192 minutes). On the other hand, the greatest accuracy is observed in the TAPLM2 (95.7%), followed by the OASLM (95.1%) and TAPLM1 (94.2%).

After optimization, the  $D^2NN$  systems were evaluated against the testing set resulting in accuracy values of 94.64% for TAPLM1, 95.82% for TAPLM2, and 94.85% for OASLM. Fig. 2(c) shows a sample result from an TAPLM2 optimized system that correctly predicted the value of the input and is typical of what was observed in all the  $D^2NN$  simulations. Although all target regions are illuminated, we can clearly see that the brightest region corresponds to the intended target region. The confusion matrices for each simulation, shown in Fig. 4, indicates that the accuracy of each  $D^2NN$  system are generally high across all numbers. The TAPLM1 and OASLM cases do struggle to classify certain digits such as 8 though both maintained accuracies above 88.98% and 88.64% respectively. Overall, the performance of each case were found to be very close with the nonlinear systems having a slight edge in terms of accuracy when classifying each number. This was especially clear in the TAPLM2 case with accuracy never falling below 92.73% for any digit. This of course comes at the expense of an increased optimization time. It is possible that these differences may decline with either an increase in the number of epochs, layers, and/or pixels.



**Fig. 4.** Confusion matrix representing  $D^2NN$  classification performance for the 8800 MNIST handwritten digit test set with overall averages for a) TAPLM1, b) TAPLM2, and b) OASLM. The overall accuracy for each system is shown at the top of each figure. They are 94.64% for the TAPLM1 case, 95.82% for the TAPLM2 case, and 94.85% for the OASLM case.

**7. Conclusion**

We have demonstrated significant improvements to both the generalization and computational efficiency of  $D^2NN$  inverse design algorithms by simulating a fully optimizable nonlinear complex-valued diffractive deep neural network utilizing the  $(DNA)^2$  optimization routine. The technique was used to develop a  $D^2NN$ -based digit classifier using linear transmission phase masks (TAPLM1), fully nonlinear amplitude/phase modulating transmission masks (TAPLM2), and fully nonlinear amplitude/phase modulating optically addressable spatial light modulators (OASLM) architectures. Using data taken from the MNIST database, weight and bias parameters associated the linear/nonlinear amplitude/phase response of each element as well as the spacing between planes were optimized for each of the aforementioned  $D^2NN$  systems. After 192 minutes or less of training and validation, each system was able to categorize a test set of 8800 never

before seen test digits, with an accuracy of 94.64% (TAPLM1), 95.82% (TAPLM2), and 94.85% (OASLM). While all systems were able to achieve a minimum 94% level of accuracy, the linear TAPLM1 D<sup>2</sup>NN architecture which was fastest to train (129 minutes) was the least able to classify certain digits with accuracy falling to as low as 88.98%. This pattern was followed by the OASLM (187 mins) which also struggled with certain digits with the lowest accuracy being 88.64%. The nonlinear cases TAPLM2 (192 min) taking the longest to train, performed the best with the accuracy never falling below 92.73% for any digit. We note however, that the average accuracy values are still very close (roughly 95%) and may be improved with increases in the number of pixels/layers and training epochs.

We anticipate that the adoption of (DNA)<sup>2</sup> in the space of diffractive optics and holography will facilitate design of more advanced systems. Such systems may require balancing the needs of multiple wavelengths using dispersion engineered materials (meta-atoms) or multiple objectives imposed by an imaging system. In these instances, the (DNA)<sup>2</sup> algorithm stands as an efficient alternative to the standard approach that could significantly accelerate inverse design process for gradient based approaches. Beyond the diffractive optics, (DNA)<sup>2</sup> may also find use as a flexible phase/amplitude retrieval algorithm in coherent optical imaging systems. Being based on the Rayleigh-Sommerfeld algorithm, our approach is able to operate seamlessly in nearfield or farfield conditions.

**Funding.** Air Force Research Laboratory under the MaRSS contract task order (13 FA8650-16-D-5404-0013.).

**Disclosures.** The authors declare no conflicts of interest.

**Data availability.** Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon request.

## References

1. X. Sui, Q. Wu, J. Liu, Q. Chen, and G. Gu, "A review of optical neural networks," *IEEE Access* **8**, 70773–70783 (2020).
2. D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Opt. Lett.* **10**(2), 98 (1985).
3. D. Psaltis, D. Brady, X. G. Gu, and S. Lin, "Holography in artificial neural networks," *Nature* **343**(6256), 325–330 (1990).
4. R. Xu, P. Lv, F. Xu, and Y. Shi, "A survey of approaches for implementing optical neural networks," *Opt. Laser Technol.* **136**, 106787 (2021).
5. X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science* **361**(6406), 1004–1008 (2018).
6. D. Mengu, Y. Luo, Y. Rivenson, and A. Ozcan, "Analysis of diffractive optical neural networks and their integration with electronic neural networks," *IEEE J. Sel. Top. Quantum Electron.* **26**(1), 1–14 (2020).
7. Y. Luo, D. Mengu, N. T. Yardimci, Y. Rivenson, M. Veli, M. Jarrahi, and A. Ozcan, "Broadband diffractive neural networks," in *Optics InfoBase Conference Papers*, (2020).
8. Y. Chen and J. Zhu, "An optical diffractive deep neural network with multiple frequency-channels," (2019).
9. J. Shi, M. Chen, D. Wei, C. Hu, J. Luo, H. Wang, X. Zhang, and C. Xie, "Anti-noise diffractive neural network for constructing an intelligent imaging detector array," *Opt. Express* **28**(25), 37686–37699 (2020).
10. J. Li, D. Mengu, Y. Luo, Y. Rivenson, and A. Ozcan, "Class-specific differential detection in diffractive optical neural networks improves inference accuracy," *Adv. Photon.* **1**(04), 1 (2019).
11. I. U. Idehenre and M. S. Mills, "Multi-directional beam steering using diffractive neural networks," *Opt. Express* **28**(18), 25915–25934 (2020).
12. Y. Luo, D. Mengu, N. T. Yardimci, Y. Rivenson, M. Veli, M. Jarrahi, and A. Ozcan, "Design of task-specific optical systems using broadband diffractive neural networks," *Light: Sci. Appl.* **8**(1), 112 (2019).
13. S. Zheng, X. Zeng, L. Zha, H. Shangguan, S. Xu, and D. Fan, "Orthogonality of diffractive deep neural networks," (2018).
14. Q. Zhao, S. Hao, Y. Wang, L. Wang, and C. Xu, "Orbital angular momentum detection based on diffractive deep neural network," *Opt. Commun.* **29**, 36936–36952 (2019).
15. Z. Huang, P. Wang, J. Liu, W. Xiong, Y. He, J. Xiao, H. Ye, Y. Li, S. Chen, and D. Fan, "All-optical signal processing of vortex beams with diffractive deep neural networks," *Phys. Rev. Appl.* **15**(1), 014037 (2021).
16. Y. L. Xiao, S. Li, G. Situ, and Z. You, "Unitary learning for diffractive deep neural network," *Opt Lasers Eng.* **139**, 106499 (2021).
17. Y. Sun, M. Dong, M. Yu, J. Xia, X. Zhang, Y. Bai, L. Lu, and L. Zhu, "Nonlinear all-optical diffractive deep neural network with 10.6  $\mu$  m wavelength for image classification," *Int. J. Opt.* **2021**, 1–16 (2021).

18. H. Wei, G. Huang, X. Wei, Y. Sun, and H. Wang, "Comment on all-optical machine learning using diffractive deep neural networks," (2018).
19. D. Mengü, Y. Luo, Y. Rivenson, X. Lin, M. Veli, and A. Ozcan, "Response to comment on all-optical machine learning using diffractive deep neural networks," (2018).
20. T. Yan, J. Wu, T. Zhou, H. Xie, F. Xu, J. Fan, L. Fang, X. Lin, and Q. Dai, "Fourier-space diffractive deep neural network," *Phys. Rev. Lett.* **123**(2), 023901 (2019).
21. H. Dou, Y. Deng, T. Yan, H. Wu, X. Lin, and Q. Dai, "Residual D 2 NN: training diffractive deep neural networks via learnable light shortcuts," *Opt. Lett.* **45**(10), 2688–2691 (2020).
22. T. Zhou, L. Fang, T. Yan, J. Wu, Y. Li, J. Fan, H. Wu, X. Lin, and Q. Dai, "In situ optical backpropagation training of diffractive optical neural networks," *Photonics Res.* **8**(6), 940–953 (2020).
23. C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," *Opt. Express* **21**(18), 21693–21701 (2013).
24. G. Veronis, R. W. Dutton, and S. Fan, "A new method for sensitivity analysis of photonic crystal devices," in *Photonic Crystal Materials and Devices III*, vol. 5733 (International Society for Optics and Photonics, 2005), pp. 348–355.
25. A. Michaels and E. Yablonovitch, "Leveraging continuous material averaging for inverse electromagnetic design," *Opt. Express* **26**(24), 31717–31737 (2018).
26. N. Kukhtarev, V. Markov, S. Odulov, M. Soskin, and V. Vinetskii, "Holographic storage in electrooptic crystals.: I. steady state," in *Landmark Papers On Photorefractive Nonlinear Optics*, (World Scientific, 1995), pp. 37–48.
27. P. A. Blanche, A. Bablumian, R. Voorakaranam, C. Christenson, W. Lin, T. Gu, D. Flores, P. Wang, W. Y. Hsieh, M. Kathaperumal, B. Rachwal, O. Siddiqui, J. Thomas, R. A. Norwood, M. Yamamoto, and N. Peyghambarian, "Holographic three-dimensional telepresence using large-area photorefractive polymer," *Nature* **468**(7320), 80–83 (2010).
28. H. Bouas-Laurent and H. Dürr, "Organic photochromism (IUPAC technical report)," *Pure Appl. Chem.* **73**(4), 639–665 (2001).
29. T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica* **5**(7), 864–871 (2018).
30. J. Grinberg, A. Jacobson, W. Bleha, L. Miller, L. Fraas, D. Boswell, and G. Myer, "A new real-time non-coherent to coherent light image converter the hybrid field effect liquid crystal light valve," *Opt. Eng.* **14**(3), 143217 (1975).
31. P. Aubourg, J.-P. Huignard, M. Hareng, and R. Mullen, "Liquid crystal light valve using bulk monocrystalline Bi<sub>12</sub>SiO<sub>20</sub> as the photoconductive material," *Appl. Opt.* **21**(20), 3706–3712 (1982).
32. U. Efron, J. Grinberg, P. Braatz, M. Little, P. Reif, and R. Schwartz, "The silicon liquid-crystal light valve," *J. Appl. Phys.* **57**(4), 1356–1368 (1985).
33. M. Cronin-Golomb, B. Fischer, J. White, and A. Yariv, "Theory and applications of four-wave mixing in photorefractive media," *IEEE J. Quantum Electron.* **20**(1), 12–30 (1984).
34. K. Matsushima and T. Shimobaba, "Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields," *Opt. Express* **17**(22), 19662 (2009).
35. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE* **86**(11), 2278–2324 (1998).
36. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).
37. R. Nishino and S. H. C. Loomis, "Cupy: A numpy-compatible library for nvidia gpu calculations," *31st conference on neural information processing systems* **151** (2017).