

Research Article

Ali Mohajer Hejazi and Vincent Ginis*

Deep neural networks for inverse design of multimode integrated gratings with simultaneous amplitude and phase control

<https://doi.org/10.1515/nanoph-2024-0667>

Received November 22, 2024; accepted January 26, 2025;

published online March 14, 2025

Abstract: We present a photonic mode converter based on a grating structure, modeled and inversely designed by deep neural networks. The neural network maps the physical parameters of the grating to the grating responses, i.e., complex scattering parameters representing the reflected modes from the grating structure. We design different neural networks to output the magnitudes and the phases of the scattering parameters associated with the multiple reflected modes. Following the training process, we use the trained networks to perform inverse design of the grating based on the desired magnitudes of the scattering parameters. The inverse design effort provides a full control on the magnitudes and the phases of the reflected modes from the mode converter. Our techniques help in creating a rich landscape of multiple interfering waves that provide higher control on optical near fields, complex resonators, and their relevant nanophotonic applications.

Keywords: artificial neural networks; inverse design; grating; mode converter

1 Introduction

Light–matter interactions in photonic devices bring about the emergence of several interdependent physical phenomena. The design of those devices relies on analytical knowledge, physical intuitions, and experiments.

In most design cases, practitioners resort to numerical simulations of the electromagnetic problems arising from light–matter interactions in photonic devices. Since last two decades, in addition to discovering and creating new functionalities in photonic systems, there have been vigorous efforts to maximize the capabilities of the devices through various optimization and inverse design techniques. Those efforts have been inevitable since novel photonic technologies often require a combination of small-size integration, sub-wavelength features, efficient utilization of optical non-linearity, broadband capabilities, etc. [1]–[6].

Inverse problems, in general, can be understood from the perspective of forward problems. Forward problems, particularly in photonics and electromagnetism, involve analytically and numerically solving the Maxwell's equations to obtain the responses of light interactions with physical domains such as waveguides, couplers, and nanostructures. These responses might be in the forms of scattering coefficients, radiation patterns, bandwidth, etc. On the other hand, in inverse problems, one or more desired responses are selected, then an optimization algorithm tries to achieve the desired response through systematic fine-tuning of the material and geometrical parameters of the physical domain [5]. Inverse problems often involve the maximization or minimization of one or multiple objective functions subject to some constraints. There are several techniques to achieve inverse optimal design of photonics devices. Among those techniques, we can mention evolutionary algorithms such as genetic algorithm [7]–[11] and particle swarm optimization [12]–[16], gradient-based methods for topology optimization [17]–[22], optimization via deep neural networks [23]–[26], etc.

Due to the power of deep learning techniques in tackling both forward and inverse problems, they have vigorously expanded since the last decade. Furthermore, another reason for the expansion is the emergence of computational hardware, like graphical processing units [27], [28], and software technologies [29] that have facilitated deep learning methods.

*Corresponding author: Vincent Ginis, Data Analytics Lab/Applied Physics, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium; and Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, 29 Oxford Street, Cambridge, MA 02138, USA, E-mail: vincent.ginis@vub.be. <https://orcid.org/0000-0003-0063-9608>

Ali Mohajer Hejazi, Applied Physics, Vrije Universiteit Brussel, 1050 Brussel, Belgium, E-mail: ali.mohajer.hejazi@vub.be. <https://orcid.org/0000-0003-1629-7913>

The deep neural networks, in forward problems, could be exploited to map a collection of feature vectors of variables – input independent variables – to the output vectors that are paired with the input vectors. A trained neural network could predict new outputs from new input provided that the input is within the space of the training data set. The capability of neural networks in creating maps from the input features to the output labels could be boosted by using a large number of data samples in the training process. The training mechanism in the neural network for a forward problem is initiated by feeding data at the network's input. Initially, the parameters of the network (weights and biases) are randomly generated. The network's output is estimated based on the generated parameters and the input data features. The estimated values of output would be compared with the actual labels from the dataset via a chosen loss function. The gradients of the loss function with respect to the weights and the biases are calculated, and the parameters of the network are adjusted according to the procedure stated in the gradient descent algorithm. This procedure would repeat until the value of the loss function becomes less than a defined threshold number.

On the other hand, an inverse design problem might be solved with the assistance of a trained deep neural network in which the weights and biases are fixed values. First, the desired values for the dependent variables are selected. Then, the mechanism is initiated by generating a random set of values as input features of the neural network. The trained neural network estimates the corresponding outputs. In this inverse design problem, gradient descent is also the core algorithm to achieve optimal results. A loss function is defined, and the error value between the desired output and the calculated output is obtained from the loss function. The derivatives of the loss function with respect to the input features are computed. The updated values of the input features can be obtained according to the algorithm. This procedure repeats until the value of the loss function becomes smaller than a defined threshold [5]. Two lucid examples of using the mentioned mechanism in nanophotonic inverse design are the studies conducted by Purifoy et al. [30] and Lenaerts et al. [31], where the former used deep neural networks for forward approximation of scattered fields from a multi-layered dielectric particle. Furthermore, they used the trained neural network in inversely design of the particle for desired scattered responses. In the latter research work, a deep neural network was utilized first for modeling the transmission spectra of a Fabry–Perot resonator. The trained model was exploited to optimize the physical parameters satisfying the desired transmission spectra.

This article uses inverse design for grating mode converters in order to precisely engineer the amplitudes and phases of multiple reflected modes at once. This is crucial in advanced applications such as designing the intricate near fields resulting from the interference of cascaded counter-propagating modes created by a succession of engineered mode converters [32], and creating novel optical resonators with unconventional electromagnetic properties [33].

All the modeling through deep neural networks and the inverse design process are performed by taking advantage of the codes provided in well-established Python library Tensorflow [34]. We use Python libraries Pandas [35] and Scikit-learn [36] for data manipulation and some statistical calculations.

2 Deep neural network modeling of the grating waveguide mode converter

Photonic waveguide gratings have been exploited in numerous applications such as optical couplers [37], [38], wavelength filtering [39], Bragg-reflection devices [40], mode-converting devices [41], field enhancement and nonlinear optical phenomena via resonant effects in grating dielectric waveguide [42], [43] etc. Cascaded mode-conversion [32], [33] is another intriguing application of waveguide gratings which is also the subject of this letter.

The mode-conversion capability of a waveguide grating emanates from the violation of mode orthogonality between guided modes due to the presence of the longitudinal periodic perturbation. The longitudinal phase matching condition required to be satisfied for the mode conversion is given by [41], [44]:

$$\beta_\mu + \beta_\nu - m\left(\frac{2\pi}{\Lambda}\right) = 0, \quad (1)$$

where β_μ and β_ν are the propagation constants of the excited mode and the converted mode respectively. The parameters Λ and m are the grating period and an arbitrary integer, respectively.

The dielectric waveguide is designed to guide the first three TE modes at wavelength 1,550 nm. Silicon, with refractive index equal to 3.48, has been selected as the dielectric material to be utilized for the waveguide and the grating. The dielectric portion of the waveguide is periodically removed to create the grating structure. The area surrounding the structure is determined to be vacuum. The whole structure is illustrated in Figure 2n on the bottom. The symbols Λ , d and t are respectively denoting the grating period, the depth of corrugations and the waveguide thickness.

The structure is simulated by exciting the first TE mode of the waveguide. The scattering coefficients S_{11} , S_{21} and S_{31} are calculated. The S_{11} , S_{21} and S_{31} are representing the reflection of the 1st mode, the reflection of the 2nd mode when the 1st mode is excited, and the reflection of the 3rd mode when the 1st mode is excited, respectively. In the simulation part, we sweep the parameters Λ , d , and duty cycle over a range of values and compute the scattering parameters. The duty cycle refers to the portion of the grating period that constitutes the grating material (Silicon). The caption of the Figure 2 provides more information about the specified ranges of the sweeping parameters. For the electromagnetic simulations of the grating mode converter, we employed COMSOL Multiphysics which is a finite element software. A data set comprising 50,545 samples is created from the simulations. Figure 1 illustrates the norms of the electric fields associated with three of these samples. Figure 1a–c depict the grating mode-converter for highly efficient conversion of the 1st mode to the 1st, the 2nd, and the 3rd mode, respectively.

The input of the deep neural network (DNN) consists of three nodes representing the physical features of the grating

i.e. the period, corrugation depth, and the duty cycle. This DNN is a supervised learning model, so that the labels corresponding to the input features are selected as two of the scattering parameters (for instance $|S_{21}|$ and $|S_{31}|$). The DNN architecture is illustrated in Figure 2o. The DNN possesses five hidden layers each with 600 nodes. The activation function designated for the first and the last two hidden layers is the ReLU function. The Sigmoid function is specified as the activation function of the hidden layer located at the middle of the DNN. In the training process, a batch size of 128 is used. The loss function for the forward DNN model is designated as the logarithm of hyperbolic cosine which is given by:

$$L = \frac{1}{2n} \sum_{i=1}^n \left(\log \left(\frac{e^{(y_{p_1}^i - y_{G_1}^i)} + e^{-(y_{p_1}^i - y_{G_1}^i)}}{2} \right) + \log \left(\frac{e^{(y_{p_2}^i - y_{G_2}^i)} + e^{-(y_{p_2}^i - y_{G_2}^i)}}{2} \right) \right), \quad (2)$$

where n denotes the number of samples from the training set. The symbols y_{p_1} and y_{p_2} indicate the DNN output

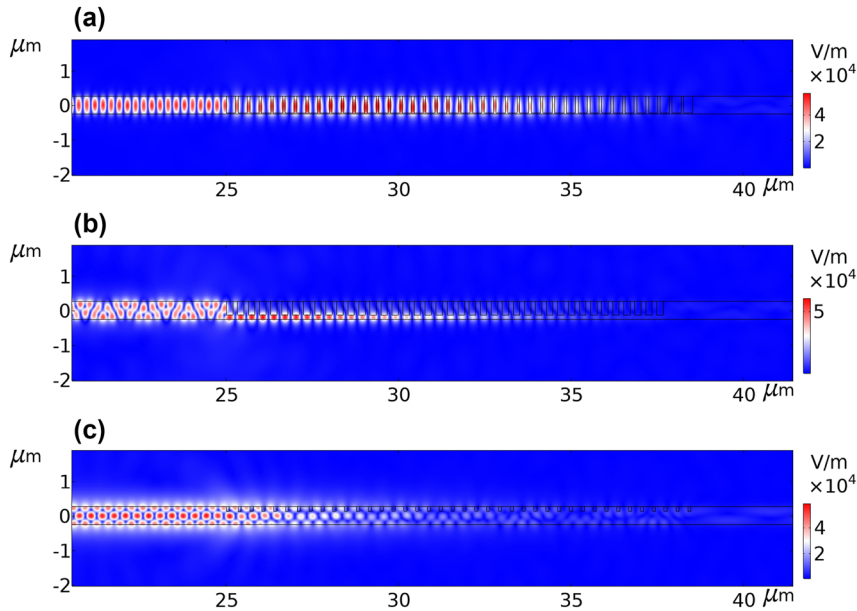


Figure 1: The plots illustrate the electric field norms in the presence of the grating mode-converter. The grating structure is distributed in the horizontal direction from $x = 25 \mu\text{m}$ to around $x = 38.5 \mu\text{m}$. The first waveguide mode is excited from the left side in all plots. In all three scenarios, the figures illustrate a substantial decrease in the electric field on the right side of the gratings. This observation indicates that a substantial fraction of the energy in the field is reflected by the grating mode-converter. (a) Demonstrates the reflection of the 1st mode. The magnitude of S_{11} is equal to 99 %. The geometrical properties that result in the value of $|S_{11}|$ are as follows: the period Λ is 340 nm, the corrugation depth d is 490 nm, and the duty cycle is 0.216. (b) The plot illustrates the conversion of the 1st mode to the 2nd mode when the 1st mode is excited. The converted mode is reflected back from the grating. The magnitude of S_{21} quantifies the extent to which the incident field is converted into the 2nd mode field. In this case, the magnitude of S_{21} is equal to 86.5 %. The grating's geometric properties that contribute to this outcome are as follows: the period (Λ) is 320 nm, the corrugation depth (d) is 390 nm, and the duty cycle is 0.417. (c) This plot shows the conversion of the 1st mode to the 3rd mode when the 1st mode is excited. The $|S_{31}|$ indicates the how much of the incident mode field is converted to the 3rd mode. In this case the parameter $|S_{31}|$ is equal to 97.5 %, and the relevant geometrical properties of the grating are as follows: the period Λ is 343 nm, the corrugation depth d is 130 nm, and the duty cycle is 0.74.

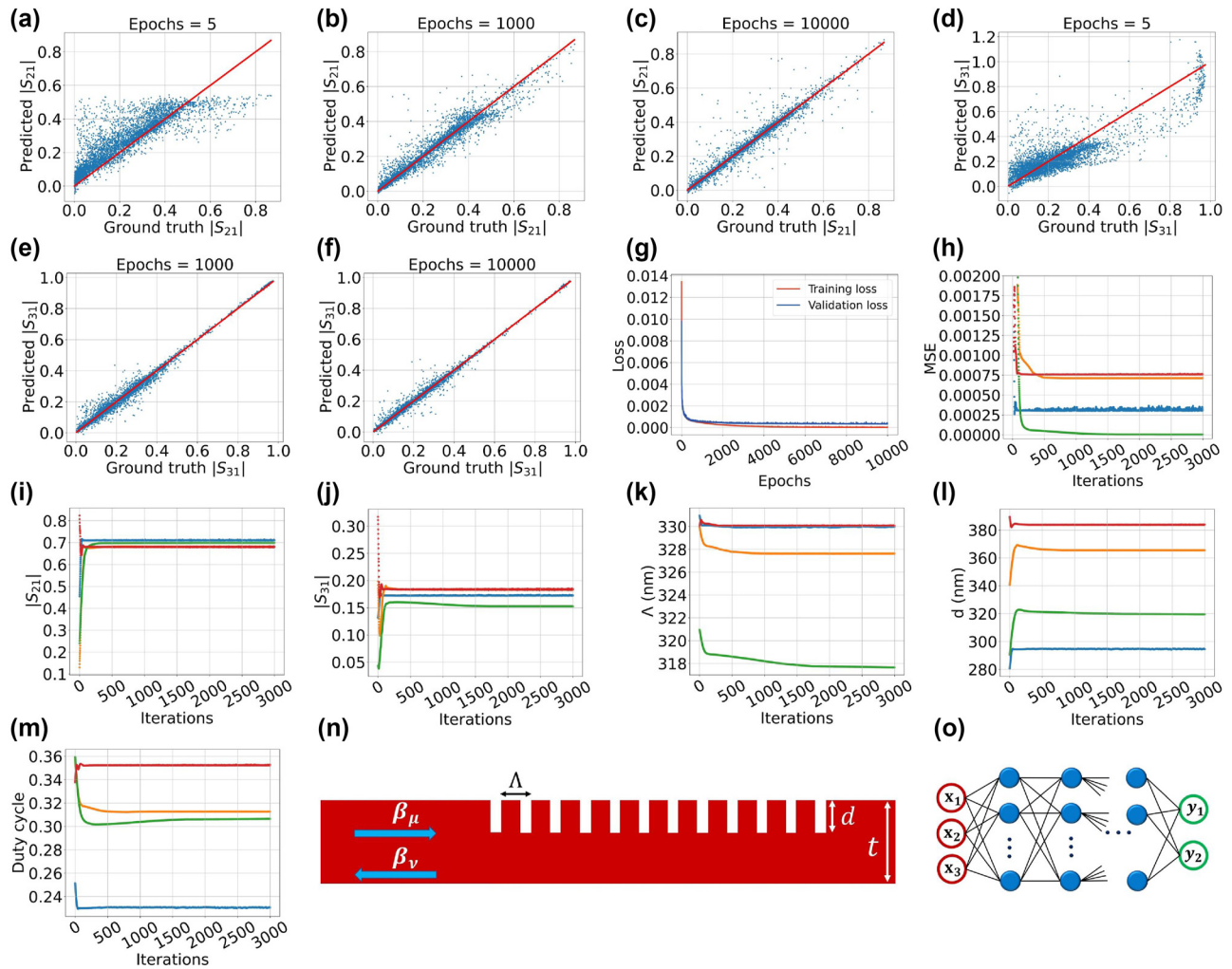


Figure 2: The plots are in three groups. The first group (a–g) shows the performance of a double-output neural network created to output the two scattering parameters' magnitudes. The second group (h–m) demonstrates the changes in various parameters during the inverse design process using the double-output neural network. The third group (n–o) shows the appearance of the grating mode converter and the utilized neural network. (a)–(c) Illustration of the scatter plots of the predicted $|S_{21}|$ over their actual values for different training epochs of 5, 1,000, and 10,000. Similar plots for $|S_{31}|$ are also depicted in (d)–(f), respectively. (g) Shows plot of the training (red) and validation (blue) losses over the 10,000 epochs. (h)–(m) Depict the results obtained from the inverse design of the grating mode converter via the trained neural network for desired magnitude of the scattering parameters $|S_{21}| = 0.7$ and $|S_{31}| = 0.15$. The inverse design algorithm is used many times leading to four different fairly accurate solutions. Each figure shows the evolutions of different variables such as value of the mean squared-error function, the scattering parameters and the physical features of the grating during optimization process. It is noteworthy that each colored curve in the figures is assigned to the same inverse design solution. (n) Illustration of the shape of the grating mode-converter connected to a waveguide. The waveguide thickness is equal to 520 nm. For the simulations to establish the data set, the ranges of the grating period, the corrugation depth, and the duty cycle are respectively chosen as 315–350 nm, 10–520 nm, and 0.1–0.9. (o) The neural network architecture used for the forward and the inverse problems is illustrated. It possesses 3 input nodes – period “ Λ ”, corrugation depth “ d ”, and duty cycle – and 2 output nodes being magnitudes of the scattering parameters $|S_{21}|$ and $|S_{31}|$.

predictions. The corresponding ground truth values are represented by y_{G_1} and y_{G_2} . The factor 2 in loss function relation 2 denotes the numbers of the output nodes of the DNN. The size of the training set is determined as 80 % of the number of samples in the data set.

The scatter plots of the predicted $|S_{21}|$ and $|S_{31}|$ over their actual values for various training epochs are depicted

in Figure 2a–f. The Figures show how the DNN's prediction performance improves with increasing epoch number. The points in the scatter plots are from a test set that include 10 % of the whole data set. Figure 2g shows the decreasing trend of the loss function with increasing epochs. The red curve depicts the loss-function values for the training set, which constitutes 80 % of the whole data set. The blue curve

Table 1: The list of the statistical measures for the assessment of the double-output DNN model for the prediction of $|S_{21}|$ and $|S_{31}|$. MAE, MSE, and EVS stand for mean-absolute error, mean-squared error, and explained-variance score respectively.

	MAE	MSE	R^2 score	EVS
For $ S_{21} $	0.014353	0.001106	0.960086	0.960177
For $ S_{31} $	0.008272	0.000293	0.991627	0.991627

demonstrates the loss function values associated with the validation dataset. Both curves suggest low values at their end points without significant divergence between them.

We calculate some statistical measures such as “Mean-squared error” (MSE), “ R^2 Score”, and “Explained variance score” (EVS) in order to quantitatively assess the DNN performance in prediction of $|S_{21}|$ and $|S_{31}|$. These statistical measures are listed in Table 1. The values in the Table are attributed to models depicted in Figure 2c and f. The R^2 score is a metric that measures how well the variability of the independent variables (DNN inputs) can account for the variability of the dependent variables (DNN outputs). In an ideal case, R^2 score is equal to unity that implies the trained model has zero error in its prediction ability. Another statistical measure mentioned in Table 1 is EVS indicating how much the prediction errors are dispersed in a regression model. EVS ranges from 0 to 1, with unity being the ideal number. A large value of EVS means that the variance of the prediction error is small compared to the variance of the actual labels which is desired.

3 Inverse design assisted by the trained neural network model

The trained DNN with fixed weights and biases could be exploited for the inverse design of the grating mode converter. We define the desired values of the two scattering parameters, for instance $|S_{21}|$ and $|S_{31}|$. The initial step in the inverse design algorithm is to randomly generate the physical features at the input nodes of the trained DNN. The trained DNN estimates the scattering parameters based on the features. The estimated scattering parameters are evaluated by comparing them with the desired scattering parameters using a loss function such as mean squared-error (MSE), that has the given mathematical form:

$$\mathbf{L} = \frac{1}{2} \left(\left(|S_{21}^m| - |\hat{S}_{21}| \right)^2 + \left(|S_{31}^m| - |\hat{S}_{31}| \right)^2 \right), \quad (3)$$

where the terms with the hat are associated with the constant desired parameters, and the ones with “ m ” superscript

are the DNN outputs which are the functions of the three physical features of the mode converter. Based on the procedure in the gradient descent algorithms, it is required to compute the derivatives of the loss function with respect to the DNN’s input features. Subsequently, the updated version of the features is computed based on the old values of the features and the values obtained from the derivatives. For the step size (learning rate) in the gradient descent, we use the Adam optimizer [45] with learning rate parameter equal to 0.001. This process continues until the value of the loss function decreases below a specified threshold.

Let the selected loss function and the output function of the trained neural network be represented by the symbols $\mathbf{L}$ and $\mathbf{J}$, respectively. The mathematical formulation for the update equations to update the values of the grating geometrical characteristics is as follows:

$$\mathbf{L} = \frac{1}{n} \sum (\mathbf{J}(\Lambda_m, d_m, t_m) - \mathbf{J}_{\text{desired}})^2, \quad (4)$$

$$\Lambda_{m+1} = \Lambda_m - \alpha \frac{\partial \mathbf{L}}{\partial \Lambda_m}, \quad (5)$$

$$d_{m+1} = d_m - \alpha \frac{\partial \mathbf{L}}{\partial d_m}, \quad (6)$$

$$t_{m+1} = t_m - \alpha \frac{\partial \mathbf{L}}{\partial t_m}, \quad (7)$$

where the symbol m denotes the iteration number in the gradient descent process. The $\mathbf{J}_{\text{desired}}$ is the target value for which we aim to achieve through inverse design. The trained DNN, represented by the symbol $\mathbf{J}$, is a function of its inputs: grating period (Λ), corrugation depth (d), and duty cycle (t).

The results of the inverse design for simultaneous desired magnitudes of the scattering parameters $|S_{21}| = 0.7$ and $|S_{31}| = 0.15$ are illustrated in Figure 2h–m. These curves are the results of running the optimization algorithm several times in which we obtain eight different satisfactory results that approximately meet the requirements. The Figure 2h–m shows four of these eight cases to keep the graph neat. We choose 3,000 as the iteration counts in the optimization process. Each time utilization of the optimization algorithm takes around 1 min. The changes in the MSE loss function, $|S_{21}|$ and $|S_{31}|$ are depicted in Figure 2h–j. Moreover, the gradual developments of the period, corrugation depth, and duty cycle from their initial random guess to their final optimal values are illustrated in Figure 2k–m. Table 2 lists the final optimal values of the grating physical features from the eight optimal cases. The table also includes four cases depicted in Figure 2, which are written in non-black colors. These features are also tested in COMSOL software and the results are listed in the

Table 2: This table lists the values of the grating physical features obtained from the inverse design with desired scattering parameters $|S_{21}| = 0.7$, $|S_{31}| = 0.15$. We use the COMSOL software to simulate the grating with the obtained physical features. The computed scattering parameters through the software, indicated by $|S_{21c}|$, $|S_{31c}|$, are listed beside the estimated scattering parameters, denoted by $|S_{21m}|$, $|S_{31m}|$, obtained from the double-output DNN model. The colored rows are related to the inverse design curves in Figure 2h–m. DC stands for duty cycle.

Λ	d	DC	$ S_{21m} $	$ S_{21c} $	$ S_{31m} $	$ S_{31c} $
329.97	294.46	0.23	0.71	0.71	0.17	0.18
327.63	365.43	0.31	0.68	0.69	0.18	0.19
334.12	353.7	0.27	0.685	0.66	0.166	0.17
323.36	332.1	0.289	0.696	0.696	0.167	0.168
324.96	300.47	0.26	0.695	0.688	0.16	0.16
317.65	319.6	0.3	0.699	0.72	0.15	0.17
322.14	379.28	0.4	0.685	0.665	0.16	0.17
330	383.87	0.35	0.681	0.688	0.18	0.2

Table 2. The computed scattering parameters in the software ($|S_{21c}|$, $|S_{31c}|$) are listed beside the calculated scattering parameters obtained from the DNN model ($|S_{21m}|$, $|S_{31m}|$).

4 Forward modeling and inverse design via triple-outputs DNN

We also conduct inverse design of the grating mode-converter to achieve desired values of $|S_{11}|$, $|S_{21}|$, and $|S_{31}|$ simultaneously. The first step is to train a new DNN model that consists of three output nodes. This model will be used to estimate the magnitude of the scattering parameters. The DNN also has five hidden layers, each consisting of 600 nodes, and utilizes the same activation functions ReLU and Sigmoid.

The key distinction between the double-output and the triple-output DNN, aside from the number of output nodes, is that the latter exploits a dropout layer after each hidden layer. The dropout layer helps prevent the DNN model from overfitting by randomly deactivating some nodes in each layer in both forward and backward propagation. In this case, the dropout layer randomly deactivate the 5 % of the nodes in each layer.

The loss function utilized in the training process of triple-output DNN is defined as follows:

$$L = \frac{1}{3n} \sum_{i=1}^n \left(\log \left(\frac{e^{(y_{p_1}^i - y_{G_1}^i)} + e^{-(y_{p_1}^i - y_{G_1}^i)}}{2} \right) + \log \left(\frac{e^{(y_{p_2}^i - y_{G_2}^i)} + e^{-(y_{p_2}^i - y_{G_2}^i)}}{2} \right) + \log \left(\frac{e^{(y_{p_3}^i - y_{G_3}^i)} + e^{-(y_{p_3}^i - y_{G_3}^i)}}{2} \right) \right), \quad (8)$$

Table 3: The list of the statistical measures for the assessment of the triple-output DNN model for the prediction of $|S_{11}|$, $|S_{21}|$, and $|S_{31}|$. MAE, MSE, and EVS stand for mean-absolute error, mean-squared error, and explained-variance score, respectively.

	MAE	MSE	R^2 score	EVS
$ S_{11} $	0.018118	0.001730	0.984159	0.984177
$ S_{21} $	0.015257	0.001088	0.960734	0.960734
$ S_{31} $	0.012638	0.000419	0.988031	0.988114

where the factor 3 in the denominator represents the number of nodes in the output layer of the DNN.

To evaluate the performance of the triple-output DNN, we calculate the aforementioned statistical measures which is listed in Table 3. The metrics EVS and R^2 score, for all three scattering parameters, are close to unity which is a sign of an accurate model. Figure 3a–c illustrate the scatter plots of the predicted $|S_{11}|$, $|S_{21}|$ and $|S_{31}|$ over their ground truth values for the training epoch of 10,000. These scatter plots are helpful for qualitatively graphical evaluation of the DNN model. The training loss and the validation loss curves are also depicted in Figure 3d. The curves are almost overlapped and both are extremely close to zero at their final points.

Similar to the inverse design of the mode converter based on the double-output DNN, we could use the same procedure for the triple-output DNN. First, we need to specify the desired values of the scattering parameters $|S_{11}|$, $|S_{21}|$, and $|S_{31}|$, where in this case they are selected to be 0.6, 0.6, and 0.3 respectively. We employ the optimization algorithm several times and obtain four cases that satisfy the constraints of the inverse design problem. The mathematical form of the inverse problem loss function is given by:

$$L = \frac{1}{3} \left(\left(|S_{11}^m| - |\hat{S}_{11}| \right)^2 + \left(|S_{21}^m| - |\hat{S}_{21}| \right)^2 + \left(|S_{31}^m| - |\hat{S}_{31}| \right)^2 \right) \quad (9)$$

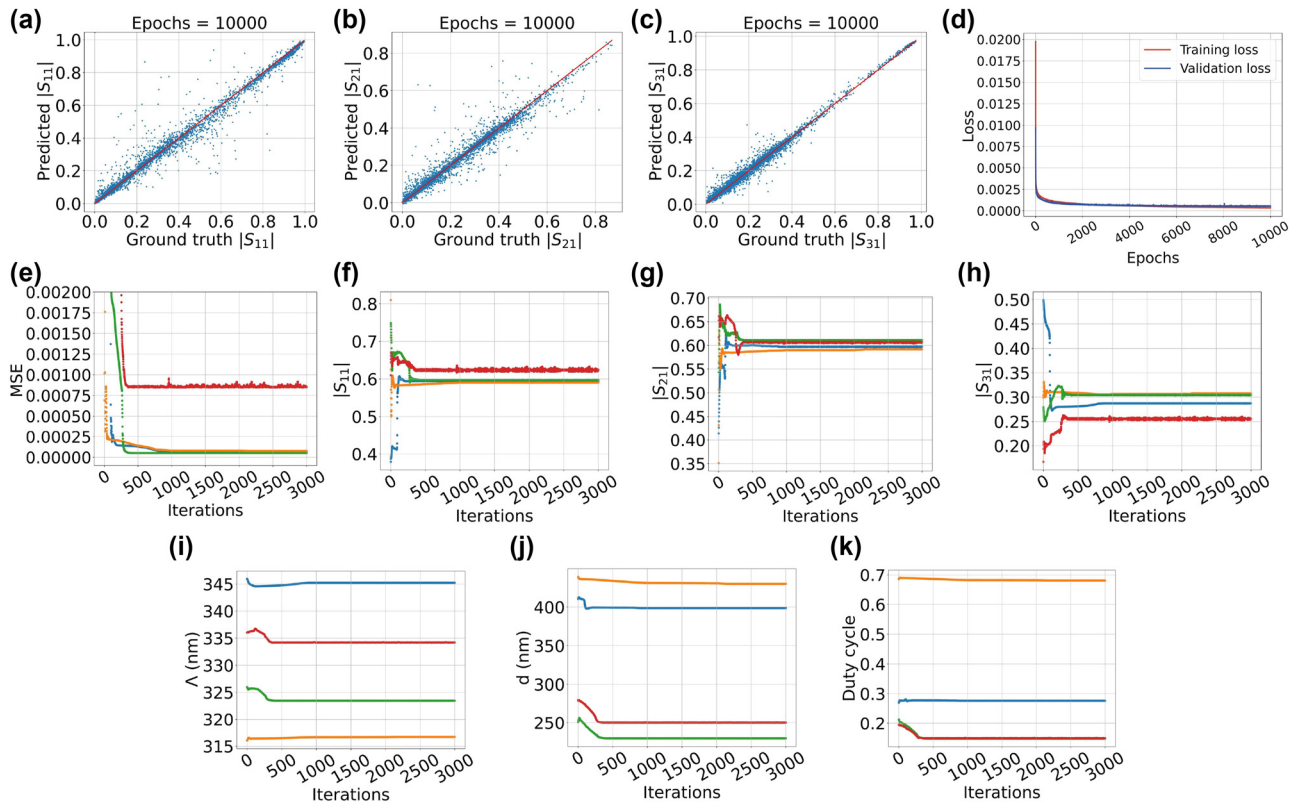


Figure 3: There are two groups of plots. The first group (a–d) demonstrates the performance of a triple-output neural network designed to produce the magnitudes of three scattering parameters. The second group (e–k) illustrates the inverse design process employing the triple-output network, which shows the gradual variations of various parameters during the process. (a)–(c) Demonstrate the scatter plots of the predicted scattering parameters $|S_{11}|$, $|S_{21}|$, and $|S_{31}|$ over their actual values for the training epoch number equal to 10,000. The blue points represent the test set including 10 % of the data set. (d) Shows the plot of the training (red) and validation (blue) losses over the 10,000 epochs. (e)–(k) Demonstrate results obtained from the inverse design of the grating mode converter for desired scattering parameters $|S_{11}| = 0.6$, $|S_{21}| = 0.6$, and $|S_{31}| = 0.3$.

The inverse design algorithm is used several times leading to four different solutions. Each figure shows the changes of different variables such as MSE function, the scattering parameters and the physical features of the grating during optimization process. Each colored curve in the figures is associated with the same inverse design solution.

Figure 3e–k depict the evolution of the MSE loss, the three scattering parameters, and the grating physical features, respectively.

We use the achieved results of the inverse design problem in COMSOL software to check the accuracy of them. The results are listed in Table 4. It is clear that the model

Table 4: This table lists the values of the grating physical features obtained from the inverse design with desired scattering parameters $|S_{11}| = 0.6$, $|S_{21}| = 0.6$, $|S_{31}| = 0.3$. The COMSOL software is exploited to simulate the grating mode converter with the obtained physical features. The computed scattering parameters through the COMSOL software are indicated by $|S_{11c}|$, $|S_{21c}|$, $|S_{31c}|$. The columns denoted by $|S_{11m}|$, $|S_{21m}|$, $|S_{31m}|$ are the estimated results from the triple-output DNN model. The colored rows are associated with the inverse design solution curves in Figure 3e–k.

Λ	d	DC	$ S_{11m} $	$ S_{11c} $	$ S_{21m} $	$ S_{21c} $	$ S_{31m} $	$ S_{31c} $
345.22	398.70	0.276	0.593	0.5	0.597	0.77	0.287	0.33
316.70	430.01	0.68	0.59	0.657	0.59	0.57	0.3	0.29
323.45	229.57	0.147	0.597	0.58	0.61	0.63	0.3	0.28
334.19	249.97	0.149	0.62	0.63	0.6	0.64	0.256	0.27

predictions and the results from the software are in agreement with one another.

5 Modeling and inverse design of the phase of the scattering parameters

The preceding sections examine the process of modeling the magnitudes of the scattering parameters using deep neural networks that include multiple output nodes. Moreover, we perform the inverse design of the mode converter based on the trained neural networks. Each scattering parameter is a complex number. We also tried to model the phase of the scattering parameters through various neural network architectures. However, the efforts have not led to proper convergence.

For instance, Figure 5a depicts the training and validation losses over epoch numbers associated with a double-output DNN. The neural network is used to map the physical attributes of the mode converter with the amplitude and phase of S_{21} . The ultimate values of the curves suggest that the losses are not sufficiently low. Furthermore, the curves diverge from each other that probably denote overfitting issue. Further evidence of the neural network's limitations is shown in Figure 5b, where the blue dots represent the DNN's predictions of the S_{21} phase compared to their actual values from a test set. Many of the blue points are distant from the red line, indicating the incapability of the DNN model. The statistical measures, including explained variance score (EVS), coefficient of determination (R^2 score), and mean squared error (MSE), for S_{21} phase in this example are equal to 0.679, 0.679, and 0.36, respectively. These values show the inadequacy of the DNN model. The reason behind this phase retrieval issue is partly the phase wrapping in the data set, partly the high sensitivity of the phase to the geometric features of the grating. Indeed, a sudden change in phase values causes a significant challenge for traditional neural networks to accurately model these values. In contrast to the phase, the amplitude is properly modeled by the double-output network, since the amplitude's variation in the data set possesses smooth quality. These different behaviors of the phases and amplitudes result in highly nonlinear relations between the inputs and outputs of the neural network. In Figure 5c, the scatter plot shows the double-output DNN's predictions of $|S_{21}|$ compared to their ground truth values for the test set. The EVS, R^2 score, and MSE values for $|S_{21}|$ are equal to 0.913, 0.913, and 0.0024, respectively.

We tried three other methods to tackle the phase retrieval. These include data augmentation, the use of alternative loss functions, and training a triple-output neural network mapping the geometrical features to the real part, imaginary part and the magnitude of the corresponding scattering parameter. We found that the latter method works better compared to the former two.

In the data augmentation method, the phase values in the data set are shifted by small amounts according to the following relation:

$$\Phi_{\text{shifted}} = \Phi_{S_{21}} + \mathcal{N}(\pi, \sigma^2) \times a, \quad (10)$$

where the noise to the phase $\Phi_{S_{21}}$ is a small value, multiplied with a sample from a Gaussian distribution ($\mathcal{N}(\pi, \sigma^2)$) centered around π with few degrees standard deviation. The parameter “ a ” thus gives us freedom to determine how much we intend to alter the phase value in percentage. It is probable that the data augmentation improves the network generalization ability. Figure 4c–f demonstrate the scatter plots of the predicted phase of S_{21} over the ground truth values resulting from this data augmentation method. The values of the parameter “ a ” (in percentage) and standard deviation (in degrees) are different for each subfigure.

Another method that we used to tackle the phase retrieval problem is to exploit an alternative loss function to alleviate the phase discontinuity. We used the cosine similarity function with mathematical formulation:

$$L = 1 - \cos(\Phi_{S_{21}}^{\text{pred}} - \Phi_{S_{21}}^{\text{ground}}). \quad (11)$$

The scatter plot associated with this method is shown in Figure 4g. The figure indicates that the DNN is not capable of properly representing the data. Finally, the result of combining the data augmentation and loss function alteration is also shown in Figure 4g. The plot shows small improvement, but generally it is not a very useful model for retrieving the phase. Table 5 includes the list of statistical measures for S_{21} phase models using the data augmentation and alternative loss function. It is evident from the table content that the augmentation method with $\sigma = 2^\circ$, and $a = 1\%$ possesses better statistical qualities, particularly compared to using cosine similarity loss function.

A different method for DNN modeling of the phase includes creating a neural network model that links the input features to the real part, imaginary part, and magnitude of the scattering parameter. In this scenario, the neural network model must possess a high level of accuracy in estimating both the real and imaginary components. This aids

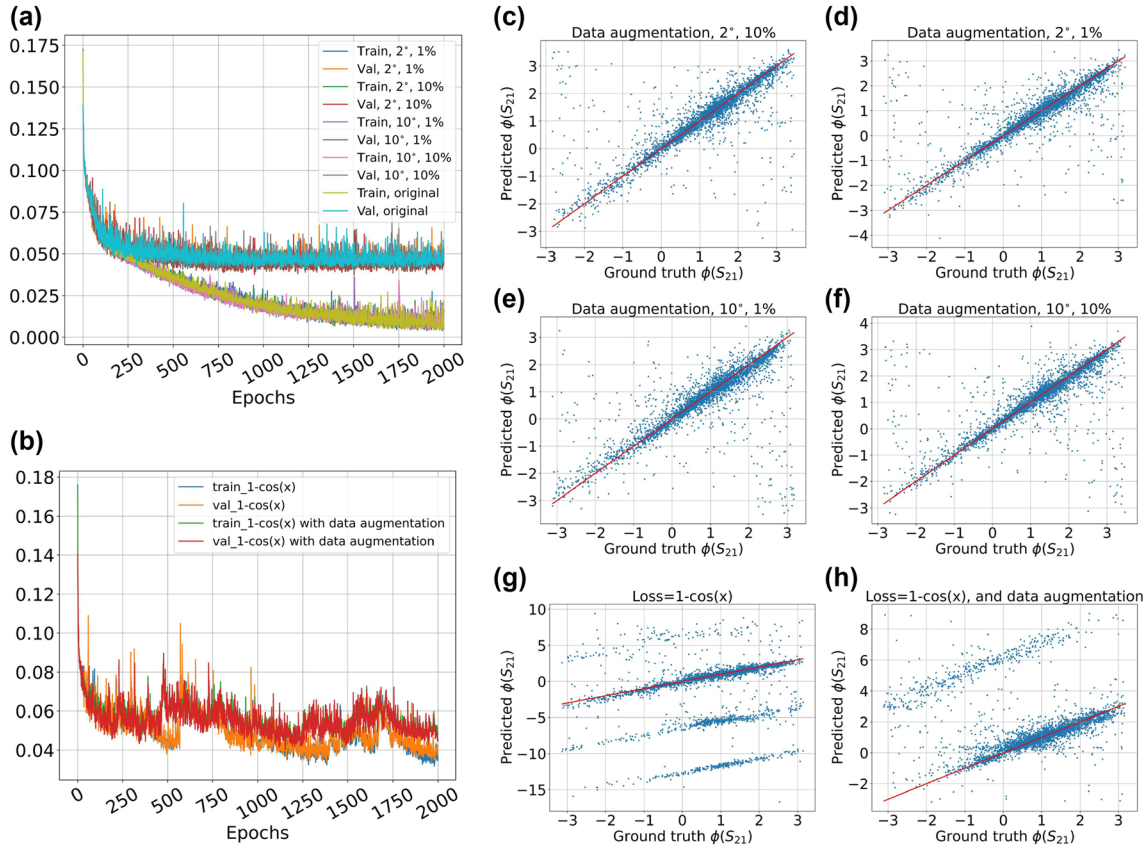


Figure 4: The subfigures demonstrate the performance of different methods to solve the scattering phase retrieval problem. (a) Demonstrates the train and validation loss curves of the data augmentation method to solve the problem of phase retrieval. In the figure the curves at the bottom indicate the training loss and the ones above are all showing validation loss. These curves do not align, pointing towards an overfitting problem, which is problematic for the inverse design purpose. (b) Shows the loss curves of another method in which we use a cosine similarity loss function ($1 - \cos(\Phi_{S_{21}}^{\text{pred}} - \Phi_{S_{21}}^{\text{ground}})$). Although, we do not face the problem overfitting problem here, the end points of the loss curves are not sufficiently low. (c)–(h) Demonstrate scatter plot of applying a test set to networks explained in (a) and (b). The majority of the points are gathered around the red bisector, however there is still a significant number of points, located too far away from this target.

Table 5: The list of the statistical measures for the assessment of the DNN model for the prediction of $\Phi(S_{21})$ by utilizing data augmentation and alternative loss function. MAE, MSE, and EVS stand for mean-absolute error, mean-squared error, and explained-variance score, respectively.

	MAE	MSE	R^2 score	EVS
Triple-output DNN	0.1871	0.3107	0.7256	0.7256
Data augmentation $\sigma = 2^\circ$, $a = 1\%$	0.215892	0.331471	0.707396	0.707733
Data augmentation $\sigma = 2^\circ$, $a = 10\%$	0.222337	0.343684	0.696625	0.696626
Data augmentation $\sigma = 10^\circ$, $a = 10\%$	0.230801	0.412481	0.635953	0.636698
Data augmentation $\sigma = 10^\circ$, $a = 1\%$	0.257033	0.432282	0.618382	0.622001
Loss = $1 - \cos(\Phi_{S_{21}}^{\text{pred}} - \Phi_{S_{21}}^{\text{ground}})$, and data augmentation	0.448349	0.958009	0.154323	0.196733
Loss = $1 - \cos(\Phi_{S_{21}}^{\text{pred}} - \Phi_{S_{21}}^{\text{ground}})$	0.650124	1.730549	−0.527622	−0.388227

in reducing the error in the ratio between the imaginary and real components. In this case, the DNN has the same configuration as the ones mentioned in the previous sections. Figure 5d demonstrates the training and validation losses over epoch number for the DNN model with output nodes

$\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$. The curves clearly exhibit low values for the losses in the final epochs. In Table 6, the statistical measures calculated for triple-output DNN are listed. All numbers are adequately qualified to claim that the DNN exhibits proper performance. Figure 5e–g demonstrate the

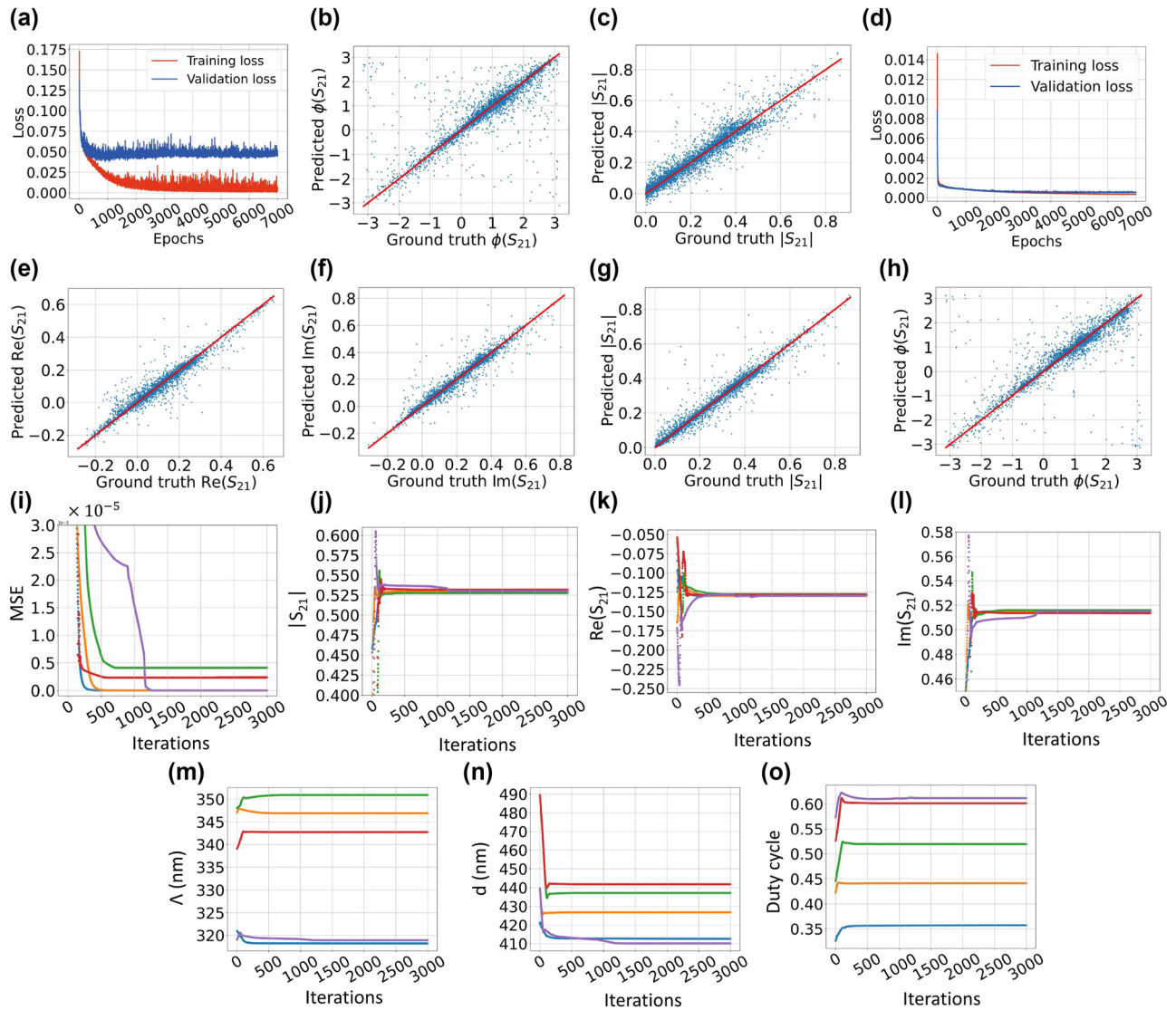


Figure 5: There are two groups of plots. The first group (a–h) shows the performance of a triple-output network designed to output the real part, imaginary part and the magnitude of one of the scattering parameters. The subfigures in the second group (i–o) illustrate the evolution of various parameters during the inverse design process for solving the phase retrieval problem. (a) Shows double-output DNN training and validation losses over epoch number. The neural network maps the mode converter physical properties to S_{21} amplitude and phase. (b) and (c) Show scatter plots predicted vs ground truth values of the S_{21} phase and amplitude for a test set applied to the neural network mentioned in (a). (d) Demonstrates the loss curves associated with the triple-output DNN. The output nodes are $\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$. (e)–(g) Illustrate the scatter plots of the test set applied to triple-output neural network mentioned in (c). These subfigures are representing the $\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$. (h) Shows the scatter plot of the phase $\Phi(S_{21})$ calculated from the scatter points in subfigures (e) and (f). The phase is calculated by taking inverse tangent of the real and imaginary parts of S_{21} . (i)–(o) Demonstrate results obtained from the inverse design of the grating mode converter for desired values $\text{Re}(S_{21}) = -0.13$, $\text{Im}(S_{21}) = 0.515$, and $|S_{21}| = 0.53$. The inverse design algorithm is used several times leading to five different solutions. Each colored curve in the figures is assigned to the same inverse design solution.

scatter plots of the predicted over the ground truth values of the test set for $\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$, respectively.

Exploiting the values shown in Figure 5e and f, we are able to calculate the scatter plot points for the S_{21} phase, which is shown in Figure 5h. Compared to the case illustrated in Figure 5b (associated with the double-output network), the Figure 5h exhibits improvement, since

more scattered points are closer the middle ideal line. In terms of the statistical measures, the EVS, R^2 score, and MSE values for this case are, respectively, equal to 0.72, 0.72, and 0.31. This shows some improvement compared to case of double-output DNN. Furthermore, the triple-output DNN also improves the statistical measures associated with $|S_{21}|$ (Table 6) in comparison with the

Table 6: The list of the statistical measures for the assessment of the triple-output DNN model for the prediction of $\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$. MAE, MSE, and EVS stand for mean-absolute error, mean-squared error, and explained-variance score, respectively.

	MAE	MSE	R^2 score	EVS
$\text{Re}(S_{21})$	0.0155	0.0009	0.9468	0.9468
$\text{Im}(S_{21})$	0.0156	0.0011	0.9559	0.9559
$ S_{21} $	0.0144	0.0009	0.9663	0.966370

double-output DNN (EVS = 0.913, R^2 score = 0.913, MSE = 0.0024). Table 5 contains all statistical information to compare the results from the triple-output DNN with the data augmentation and using alternative loss function. It is evident that triple-output exhibits better results. This observation is also found by comparing Figures 5h with 4c–h, where the scattered points are more distant from the ideal red line.

There are other advanced methods capable of modeling the complex behavior of such phase data. Physics-informed neural networks (PINNs) is a method that the Maxwell's equations get involved into the training process providing the physical constraints on the problem that ultimately assist in generalization capability of the network [46]. A hybrid utilization of adjoint methods and deep neural networks is also a powerful method, where the network performs rapid approximation, and the adjoint solver performs fine-tuning the phase accuracy [47]. Now, based on the trained triple-output DNN, it is possible to perform inverse design of the mode converter to achieve pre-defined $\text{Re}(S_{21})$, $\text{Im}(S_{21})$, and $|S_{21}|$. The method is the same as the one discussed in Sections 3 and 4. The primary advantage of this inverse design is the ability to choose the real and imaginary components of the associated scattering parameter depending on the desired phase value of the scattering parameter. Hence, the results of the inverse design process consist of the specific geometric characteristics of the grating that not only generate the intended real and imaginary components

of the scattering parameter, but also achieve the correct phase.

For the present case, the selected desired values are $\text{Re}(S_{21}) = -0.13$, $\text{Im}(S_{21}) = 0.515$, and $|S_{21}| = 0.53$. Based on the values of the real and imaginary parts, the desired phase of S_{21} is equal to $\phi(S_{21}) = 104.17^\circ$. Figure 5e–j demonstrate the results of the inverse design. Table 7 lists the numerical results of the inverse design. It is evident from the last column of the Table that the results are very close to the desired phase value $\phi(S_{21}) = 104.17^\circ$.

6 Conclusions

In this article, we exploit the power of deep neural networks to perform inverse design of the waveguide grating mode converter. For the inverse design goal, we first establish various forward models based on deep neural networks for the grating mode converter. The required data to train the neural networks is obtained through the simulation of the grating structure by the finite element software COMSOL Multiphysics. The input nodes of the neural networks are the grating period, corrugation depth, and the duty cycle. The neural networks outputs are considered to be magnitudes and the phases of the scattering parameters. The scattering parameters are the reflection coefficients of the converted modes reflected from the grating.

There are two layers of complexity to consider when scaling the proposed technique: As the number of supported optical modes increases, the number of possible mode conversions grows combinatorially, as discussed in previous work [32]. The second level of complexity involves the design and optimization of the individual mode converters and their integration within a deep neural network-based inverse design framework. This was the focus of our current study. The computational complexity of our inverse design methodology scales with the number of trainable parameters in the neural network. As the number of guided

Table 7: The following table presents the values of the physical characteristics of the grating (first 3 columns) determined using the inverse design process for the desired scattering values $\text{Re}(S_{21}) = -0.13$, $\text{Im}(S_{21}) = 0.515$, and $|S_{21}| = 0.53$. The computed scattering parameters through the COMSOL software are indicated by the columns denoted by the subscript “c”. The columns denoted by subscript “m” are the estimated results from the triple-output DNN model. The column denoted by $\phi(S_{21c})$ indicates the phase calculated by using the values in $\text{Re}(S_{21c})$ and $\text{Im}(S_{21c})$ columns. Moreover, the column indicated by $\phi(S_{21m})$ is the phase calculated from the values $\text{Re}(S_{21m})$ and $\text{Im}(S_{21m})$ columns. The desired value of the phase based on $\text{Re}(S_{21}) = -0.13$ and $\text{Im}(S_{21}) = 0.515$ is equal to 104.17° . The colored rows are relevant to the inverse design solution curves in Figure 5i–o.

Λ (nm)	d (nm)	Duty cycle	$\text{Re}(S_{21m})$	$\text{Re}(S_{21c})$	$\text{Im}(S_{21m})$	$\text{Im}(S_{21c})$	$ S_{21m} $	$ S_{21c} $	$\phi(S_{21m})$	$\phi(S_{21c})$
346.90	426.86	0.440	-0.1299	-0.1291	0.5149	0.50084	0.5300	0.51721	104.159°	104.45°
342.75	441.80	0.601	-0.1285	-0.15055	0.5136	0.4908	0.5317	0.5133	104.04°	107.05°
318.90	410.11	0.610	-0.1299	-0.1445	0.5149	0.5736	0.5299	0.5915	104.15°	104.14°
318.20	412.60	0.357	-0.1299	-0.1466	0.5150	0.4953	0.5299	0.5165	104.156°	106.49°
350.90	437.17	0.51900	-0.1277	-0.1314	0.5160	0.4649	0.5275	0.4831	103.90°	105.78°

optical modes increases, the dimensionality of the scattering parameter space also grows, requiring a more complex neural network architecture with a larger dataset for training. However, our approach benefits from the ability of deep learning models to generalize efficiently across large design spaces. Our trained models have demonstrated high accuracy even for multi-output regression tasks, as shown in the case of triple-output networks predicting multiple scattering parameters. The computational cost of each inverse design process is largely independent of the number of modes. However, generating sufficient training data through electromagnetic simulations (e.g., using COMSOL) will become more time-intensive as more modes are considered. This can be computationally expensive, especially for complex photonic structures where each data point requires solving Maxwell's equations with fine spatial resolution. While transfer learning and data augmentation can partially mitigate this issue, the initial cost of dataset generation remains a constraint. Computational cost is another factor, as deep neural networks, particularly those with large architectures, require significant GPU or cloud computing resources for training. However, once trained, the network provides near-instantaneous predictions, making it highly advantageous over traditional iterative optimization methods for large-scale inverse design problems.

Finally, there's also a subtlety related to the different design parameters. In particular the waveguide thickness is a crucial design parameter, because its variations influence the system in two fundamental ways. First, thickness fluctuations alter the effective refractive index of the guided modes, modifying their propagation constants. Since mode conversion relies on phase matching, even small deviations in thickness can cause phase mismatches and reduce conversion efficiency. In extreme cases, thickness variations may even cause certain modes to disappear, fundamentally changing the mode structure of the system. Second, the scattering properties of the grating-based mode converter depend sensitively on the waveguide thickness. A small shift in thickness alters the reflection/transmission coefficients and changes the relative amplitudes and phases of the converted modes. This can degrade the precision of the designed grating, leading to unexpected behavior in multi-mode interference and cascaded mode conversion systems. Thus, controlling and accounting for thickness variations is essential for ensuring robust inverse design. All libraries and codes developed and used in this manuscript are available in a link in Supplementary Materials. With these efforts, we want to provide the photonics community with a tool for controlling the phases the magnitudes of the

multiple waves reflected from integrated mode converters. This capability could be utilised in interference of multiple counter-propagating waves or advanced integrated photonic circuits.

Research funding: VG acknowledges support from Research Foundation Flanders under grant numbers G032822N and G0K9322N.

Author contributions: AMH initiated the project and developed the library and code presented in the manuscript with inputs from VG. AMH analyzed all the data and carried out the numerical simulations. VG supervised the project. Both authors contributed discussion, and writing of the manuscript.

Conflict of interest: Authors state no conflicts of interest.

Data availability: All data, both training libraries and codes, are available on the project's Zenodo repository at <https://zenodo.org/records/15000131>.

References

- [1] C. Kang, C. Park, M. Lee, J. Kang, M. S. Jang, and H. Chung, "Large-scale photonic inverse design: computational challenges and breakthroughs," *Nanophotonics*, vol. 13, no. 20, pp. 3765–3792, 2024.
- [2] A. Khaireh-Walieh, D. Langevin, P. Bennet, O. Teytaud, A. Moreau, and P. R. Wiecha, "A newcomer's guide to deep learning for inverse design in nano-photonics," *Nanophotonics*, vol. 12, no. 24, pp. 4387–4414, 2023.
- [3] J. Park, S. Kim, D. W. Nam, H. Chung, C. Y. Park, and M. S. Jang, "Free-form optimization of nanophotonic devices: from classical methods to deep learning," *Nanophotonics*, vol. 11, no. 9, pp. 1809–1845, 2022.
- [4] S. Mao, L. Cheng, C. Zhao, F. N. Khan, Q. Li, and H. Fu, "Inverse design for silicon photonics: from iterative optimization algorithms to deep neural networks," *Appl. Sci.*, vol. 11, no. 9, p. 3822, 2021.
- [5] J. Jiang, M. Chen, and J. A. Fan, "Deep neural networks for the evaluation and design of photonic devices," *Nat. Rev. Mater.*, vol. 6, no. 8, pp. 679–700, 2021.
- [6] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nat. Photonics*, vol. 12, no. 11, pp. 659–670, 2018.
- [7] A. Sharma, *et al.*, "Optimization of a programmable $\lambda/2$ -pitch optical phased array," *Nanophotonics*, vol. 13, no. 12, pp. 2241–2249, 2024.
- [8] R. Sabri and H. Mosallaei, "Inverse design of perimeter-controlled inas-assisted metasurface for two-dimensional dynamic beam steering," *Nanophotonics*, vol. 11, no. 20, pp. 4515–4530, 2022.
- [9] S. Jafar-Zanjani, S. Inampudi, and H. Mosallaei, "Adaptive genetic algorithm for optical metasurfaces design," *Sci. Rep.*, vol. 8, no. 1, p. 11040, 2018.
- [10] Y. Shi, W. Li, A. Raman, and S. Fan, "Optimization of multilayer optical films with a memetic algorithm and mixed integer programming," *ACS Photonics*, vol. 5, no. 3, pp. 684–691, 2017.

- [11] R. L. Haupt and D. H. Werner, *Genetic Algorithms in Electromagnetics*, Hoboken, New Jersey, John Wiley & Sons, 2007.
- [12] C. Yang, W. Cai, and Z. M. Zhang, “Tailoring full-Stokes thermal emission from twisted-gratings structures,” *Nanophotonics*, vol. 13, no. 5, pp. 803–811, 2024.
- [13] R. Wang, C. Wang, T. Sun, X. Hu, and C. Wang, “Simultaneous broadband and high circular dichroism with two-dimensional all-dielectric chiral metasurface,” *Nanophotonics*, vol. 12, no. 21, pp. 4043–4053, 2023.
- [14] J. Kim, J.-Y. Kim, J. Yoon, H. Yoon, H.-H. Park, and H. Kurt, “Experimental demonstration of inverse-designed silicon integrated photonic power splitters,” *Nanophotonics*, vol. 11, no. 20, pp. 4581–4590, 2022.
- [15] H. Farheen, T. Leuteritz, S. Linden, V. Myroshnychenko, and J. Förstner, “Optimization of optical waveguide antennas for directive emission of light,” *JOSA B*, vol. 39, no. 1, pp. 83–91, 2022.
- [16] M. Passoni, D. Gerace, L. Carroll, and L. Andreani, “Grating couplers in silicon-on-insulator: the role of photonic guided resonances on lineshape and bandwidth,” *Appl. Phys. Lett.*, vol. 110, no. 4, p. 041107, 2017.
- [17] M. F. Schubert, “Invs-gym: a toolkit for nanophotonic inverse design research,” *arXiv preprint arXiv:2410.24132*, 2024.
- [18] W. Ma, *et al.*, “Topologically-optimized on-chip metamaterials for ultra-short-range light focusing and mode-size conversion,” *Nanophotonics*, vol. 12, no. 6, pp. 1189–1197, 2023.
- [19] V. Nikkhah, M. J. Mencagli, and N. Engheta, “Reconfigurable nonlinear optical element using tunable couplers and inverse-designed structure,” *Nanophotonics*, vol. 12, no. 14, pp. 3019–3027, 2023.
- [20] Y. Zheng, *et al.*, “Designing high-efficiency extended depth-of-focus metalens via topology-shape optimization,” *Nanophotonics*, vol. 11, no. 12, pp. 2967–2975, 2022.
- [21] R. E. Christiansen and O. Sigmund, “Inverse design in photonics by topology optimization: tutorial,” *JOSA B*, vol. 38, no. 2, pp. 496–509, 2021.
- [22] M. Mansouree, A. McClung, S. Samudrala, and A. Arbabi, “Large-scale parametrized metasurface design using adjoint optimization,” *ACS Photonics*, vol. 8, no. 2, pp. 455–463, 2021.
- [23] C. Park, *et al.*, “Sample-efficient inverse design of freeform nanophotonic devices with physics-informed reinforcement learning,” *Nanophotonics*, vol. 13, no. 8, pp. 1483–1492, 2024.
- [24] Y. Jing, H. Chu, B. Huang, J. Luo, W. Wang, and Y. Lai, “A deep neural network for general scattering matrix,” *Nanophotonics*, vol. 12, no. 13, pp. 2583–2591, 2023.
- [25] Z. Zhang, C. Yang, Y. Qin, H. Feng, J. Feng, and H. Li, “Diffusion probabilistic model based accurate and high-degree-of-freedom metasurface inverse design,” *Nanophotonics*, vol. 12, no. 20, pp. 3871–3881, 2023.
- [26] S. So, T. Badloe, J. Noh, J. Bravo-Abad, and J. Rho, “Deep learning enabled inverse design in nanophotonics,” *Nanophotonics*, vol. 9, no. 5, pp. 1041–1057, 2020.
- [27] X. Shi, *et al.*, “Graph processing on gpus: a survey,” *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–35, 2018.
- [28] R. A. Bridges, N. Imam, and T. M. Mintz, “Understanding gpu power: a survey of profiling, modeling, and simulation methods,” *ACM Comput. Surv.*, vol. 49, no. 3, pp. 1–27, 2016.
- [29] B. J. Erickson, P. Korfiatis, Z. Akkus, T. Kline, and K. Philbrick, “Toolkits and libraries for deep learning,” *J. Digit. Imag.*, vol. 30, no. 4, pp. 400–405, 2017.
- [30] J. Peurifoy, *et al.*, “Nanophotonic particle simulation and inverse design using artificial neural networks,” *Sci. Adv.*, vol. 4, no. 6, p. eaar4206, 2018.
- [31] J. Lenaerts, H. Pinson, and V. Ginis, “Artificial neural networks for inverse design of resonant nanophotonic components with oscillatory loss landscapes,” *Nanophotonics*, vol. 10, no. 1, pp. 385–392, 2020.
- [32] V. Ginis, *et al.*, “Remote structuring of near-field landscapes,” *Science*, vol. 369, no. 6502, pp. 436–440, 2020.
- [33] V. Ginis, I.-C. Benea-Chelms, J. Lu, M. Piccardo, and F. Capasso, “Resonators with tailored optical path by cascaded-mode conversions,” *Nat. Commun.*, vol. 14, no. 1, p. 495, 2023.
- [34] M. Abadi, *et al.*, “Tensorflow: a system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [35] W. McKinney, *et al.*, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, vol. 445, Austin, TX, 2010, pp. 51–56.
- [36] F. Pedregosa, *et al.*, “Scikit-learn: machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [37] D. Taillaert, P. Bienstman, and R. Baets, “Compact efficient broadband grating coupler for silicon-on-insulator waveguides,” *Opt. Lett.*, vol. 29, no. 23, pp. 2749–2751, 2004.
- [38] T. Tamir and S.-T. Peng, “Analysis and design of grating couplers,” *Appl. Phys.*, vol. 14, no. 3, pp. 235–254, 1977.
- [39] H. Venghaus, *Wavelength Filters in Fibre Optics*, vol. 123, Heidelberg, Germany, Springer, 2006.
- [40] J. E. Carroll, J. Whiteaway, and D. Plumb, *Distributed Feedback Semiconductor Lasers*, vol. 10, Stevenage, UK, IET, 1998.
- [41] A. Yariv and M. Nakamura, “Periodic structures for integrated optics,” *IEEE J. Quant. Electron.*, vol. 13, no. 4, pp. 233–253, 1977.
- [42] G. Quaranta, G. Basset, O. J. Martin, and B. Gallinet, “Recent advances in resonant waveguide gratings,” *Laser Photonics Rev.*, vol. 12, no. 9, p. 1800017, 2018.
- [43] M. Neviere, E. Popov, R. Reinisch, and G. Vitrant, *Electromagnetic Resonances in Nonlinear Optics*, London, UK, CRC Press, 2000.
- [44] A. Yariv, P. Yeh, and A. Yariv, *Photonics: Optical Electronics in Modern Communications*, vol. 6, New York, Oxford University Press, 2007.
- [45] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [46] F. Tori and V. Ginis, “Phase space approach to solving higher order differential equations with artificial neural networks,” *Phys. Rev. Res.*, vol. 4, no. 4, p. 043090, 2022.
- [47] M. Chen, *et al.*, “High speed simulation and freeform optimization of nanophotonic devices with physics-augmented deep learning,” *ACS Photonics*, vol. 9, no. 9, pp. 3110–3123, 2022.