Inverse design of thin film stacks through a generative residual global optimization network Res-GLONet

Shadman Shahid

October 2021

Abstract

Despite the seeming simplicity of one dimensional (1D) multilayer dielectric stacks, their design process can be non-intuitive. The designer converges to a 'good' design for a given optical response through successive trail-and-error of numerous stacks. Thus optimization of such designs is highly tedious because for even as few as two design variables, such as layer thickness and refractive indices, the designer has to explore through a huge variety of possible combinations of design parameters in the design space. Traditional gradient based algorithms can be utilized in 'inverse design'-ing an optimized 1D stack from a given spectral response. However, such processes are computationally inefficient and slow to converge. Deep learning networks can be used for a faster design convergence in such an optimization problem, by gradually learning the space spanned by the design variables. In this regard I have implemented a generative neural network for inverse design of thin film stacks. The generator network, modelled by a residual multilayer perceptron (MLP) network is coupled to a transfer matrix method electromagnetic solver for 1D film stacks, acting as the discriminator. Even for a structure composed of 100s of layers, the model quickly converges to an optimized design within 10 minutes, in a personal laptop device with meager GPU capability.

1 Introduction

Inverse design is the process of obtaining a device design from the device response or some performance measure [1–3]. The optical response of a particular class of nanophotonic devices, such as metagratings, multi-layer thin film stacks, planar photonic waveguides etc. can be controlled by optimizing several device parameters. Contrary to forward design (device to optical response), inverse design (optical response to device) is a more efficient design method, as it reduces human effort and the level of expertise needed to design photonic devices. Forward design usually follows a trial-and-error process and is computationally very expensive since many repeated and often un-optimized series of electromagnetic simulations make it painstaking to converge on a good result. In inverse design, conventional optimization techniques such as particle swarm algorithm, genetic algorithm, and adjoint based topology-optimization are some of the employed methods. However, these methods require multiple electromagnetic simulations to converge on a good result and are thus computationally expensive. In this regard concepts of machine learning have recently been showed to be useful in photonic design process to reduce human and computational effort [4]. In photonic design enhanced through deterministic deep learning approaches, a training set of device configurations and their corresponding responses are used to train the machine learning model. Once properly trained, the model learns the relationship between the device design and output response. However, learning and optimizing the device design from such neural networks face the following key challenges

- **High dimensional data:** Even the simplest of photonic devices with few design parameters, have a high dimensional design space, i.e., the entire range of possible device configurations possible, is a high dimensional space. Obtaining accurate response from a high dimensional data is computationally expensive and require a lot of training samples.

- **Costly training:** Large number of training samples are re-

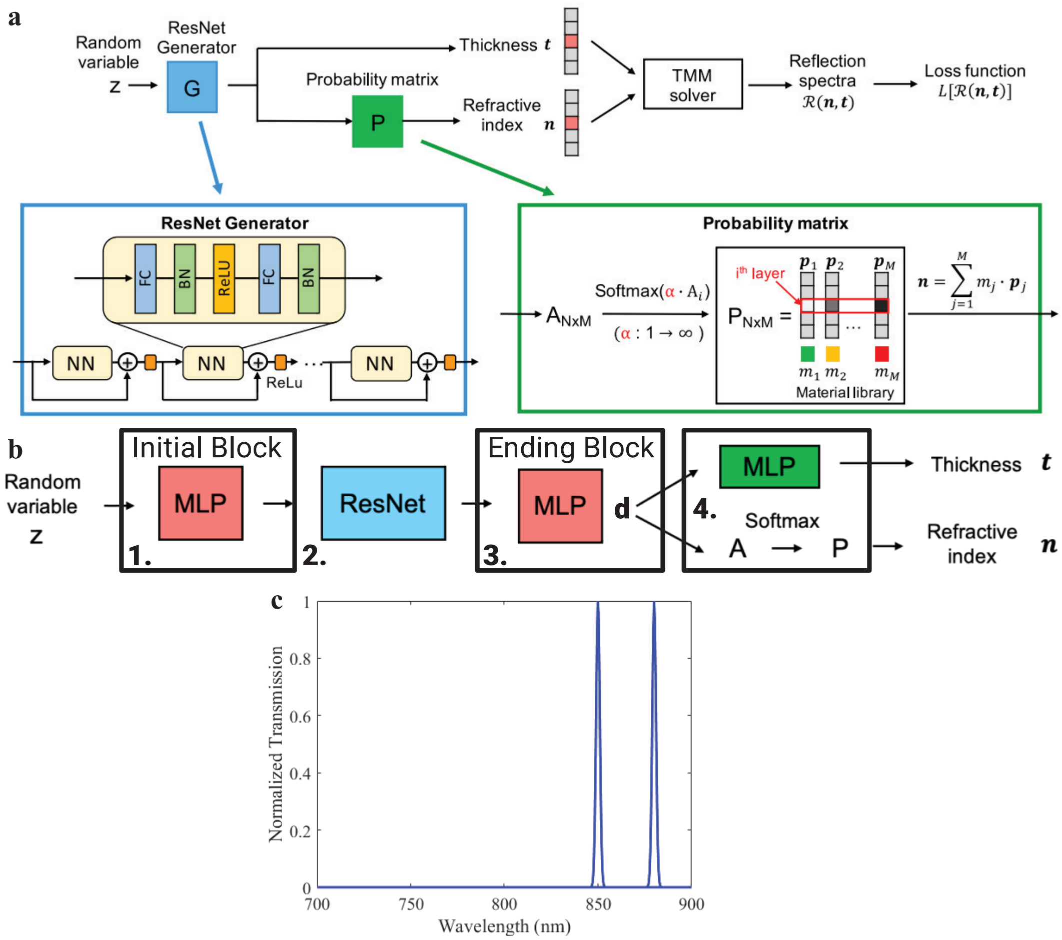

Figure 1: (a) Network architecture of Residual GLONet model as reported in Ref. 5. (b) The expanded network architecture shows constituent multilayer perceptron network blocks. The constituent blocks of the are numbered from 1 to 4 and described in greater detail in Fig. 2. (c) Target transmission spectrum for the thin film stack to be designed.

quired to properly learn the design space to be able to give good enough results. Generating training samples through electromagnetic simulators is time consuming and computationally expensive. Networks for structures described by even a few geometric parameters require tens to hundreds of thousands of devices for training.

2 Prior related work

Generative models coupled with EM solvers have been proposed as a solution to circumvent the problems associated with costly training. Jiaqi Jiang and Jonathan Fan of Stanford University have reported a generative model termed as global optimization network - GLOnet, in optimizing the design of a simple one dimensional metagrating [6, 7], free-form two dimensional metasurface [8] and a multilayer thin film stack [5]. The characteristic advantage of their proposed model is that a host of efficient devices are proposed. On the contrary, deterministic algorithms only learn one to one mapping, and are thus much slower to search through the design space. Mapping in deterministic algorithms is offered by the the ensemble of training samples [4]. The generative network, searches the entire design space directly with the help of the EM solver. However, with training the sub-space of the design space searched, converges towards the region of optimized design based upon the loss function. So, the design space search is guided by the loss function gradient extracted from the TMM solver, unlike the raw training samples, which may or may not span the entirety of design space. So, it is less computationally intensive to train the model.

In this report, the residual GLOnet - called ResNet - for optimization of thin film stacks has been implemented [5]. The model architecture is shown in Fig. 1(a,b). The objective was to design a thin film stack to mimic a certain reflection/transmission spectrum. The model parameters and architectures are obtained from Ref. 5 as reported in Fig. 1(a,b).

3 Approach

The design training pipeline consists of a generator network and a TMM solver. The generator network produces the design of the thin film stack, which is to be optimized. The TMM solver can be thought of as a discriminator or an extended layer of the generator network to compute the losses. So, for evaluation the generator is separated from the TMM solver. As design variables, thickness (**t**) vector and refractive indices **n** tensor is produced. The model is designed to optimize a stack of N-layers. The refractive index is actually a categorical variable, chosen from a material library of M-materials. A random noise vector is the input and the

Generator Network

A random variable $z \sim U^{16}(0,1)$, is an 16-dimensional vector with uniform random distribution between 0 and 1. This is the input to the first fully connected network, that maps the 16 dimensional z vector to a vector of s_{res} dimensions (Fig. 2(a)). This vector is fed into the ResNet generator, which is composed of 16 (or less) residual blocks of fully connected layers each with a residual connection (Fig. 2(b)). At the end of the ResNet lies an MLP that churns out an $N\times(M+1)$ dimensional vector (Fig. 2(c)). This $N\times(M+1)$ dimensional vector is reshaped to a tensor of $(N,M+1)$ size and split it to two tensors: a matrix $\mathbf{A}$ of size (N,M) and a vector $\mathbf{d}$ of N dimension (Fig. 2(d)). These two tensors form the precursor to the refractive index vector and layer thickness vector respectively for the designed thin film stack. The $\mathbf{d}$ vector is transformed to the thickness vector t by first passing through an FC network and then transforming the output (d^*) as per Eqn. 1 equation where t_u is the upper limit of layer thickness in the thin film stack.

$$t = \sigma(d^*) + t_u \tag{1}$$

$$P = \frac{e^{\alpha A_{ij}}}{\sum_{j=1}^{M} e^{\alpha A_{ij}}} \tag{2}$$

$$n_i = \sum_{j=1}^{M} m_j \cdot P_{ij} \tag{3}$$

The matrix $\mathbf{A}$ is transformed to a probability distribution P of the same size following Eqn. 2. Then Eqn. 3 maps these probabilities to a refractive index value in a layer. Over the course of training, the probability distribution becomes more discrete and points to a single material for a single layer. The thickness vector $\mathbf{t} = (t_1, t_2, t_3, \cdots t_N)^T$ of N dimension and the refractive index matrix $\mathbf{n} = (n_1(\lambda), n_2(\lambda), n_3(\lambda), \cdots n_N(\lambda))^T$ of size (N, N_λ) where N_λ is the number of frequency points (Fig. 2).

This generator network was coded using PyTorch framework's classes and functions based on the details provided in Ref. 5. Part of the code was taken from existing recipes of similar vanilla networks available in PyTorch's documentation and a similar project implemented in Ref. 6, 7.

TMM Solver and Loss functions

As a discriminator, a transfer matrix method (TMM) solver is used. A TMM solver is a one dimensional (1D) solver that computes layer by layer electric field propagation and transition in a multi layer thin film stack, solving the Fresnel equations. It can calculate the reflection transmission coefficients as well as average reflectivity and transmissivity. The TMM solver in this project was implemented based on the python library - tmm - by S.J. Byrnes [9]. The solver of S.J. Byrnes however, was implemented in python's NumPy module and was thus not suitable for integrating with the computation graph of pytorch for calcualtion and backpropagation of gradients. It was also not vectorized for multiple devices (i.e. batch size), wavelengths and incident angles. So, that tmm solver was modified and implemented in pytorch, enabling seamless integration with pytorch's automatic differentiation. To vectorize the solver in an efficient way, the transfer matrices as seen in Ref. 9 were converted to scattering matrices, following the equation in Ref. 10. The solver was vectorized for more than one batch

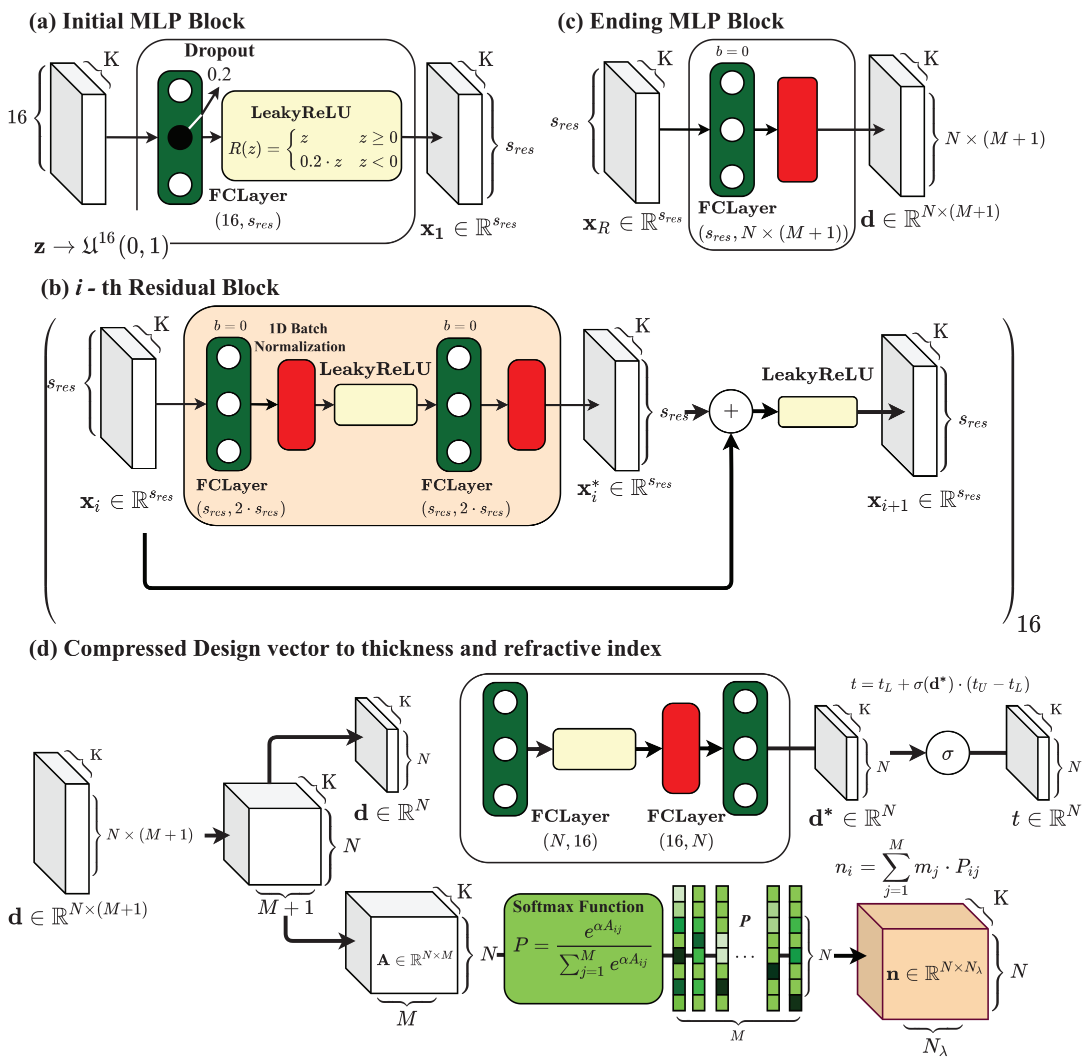

Figure 2: (a) The initial MLP block, indicated by block **'1'** in Fig. 1(b) produces a s_{res}-dimensional vector from a 16-dimensional random noise vector. (b) The residudal block has two fully connected layers with batch normalization and a skip connection with the input vector. $R = 16$ of such blocks are cascaded in the final network architecture in the ResNet block (block **'2'** in Fig. 1(b)). (c) The ending block (**'3'**) gives a high dimensional vector of size $N \times (M+1)$ where N is the number of layers and M is the number of materials in the material library. This vector is the design space vector $\mathbf{d}$ (d)The $\mathbf{d}$ vector is converted separately to a thickness vector t of N-dimension and a refractive index matrix of size (N, N_λ) via softmax conversion to a probability matrix P that chooses from the category of M materials. The sample size in all the data in the figure is K.

$$p_i = \begin{cases} 1, & s-\text{polarization} \\ -n_i^2, & p-\text{polarization} \end{cases} \tag{4}$$

$$M_i = \begin{pmatrix} \cos k_z t_i & j\frac{k}{k_z \cdot p_i}\sin k_z t_i \\ j\frac{k_z}{k \cdot p_i}\sin k_z t_i & \cos k_z t_i \end{pmatrix} \tag{5}$$

$$S = \begin{pmatrix} 1 & 1 \\ -\frac{k_z}{k \cdot p_{N+1}} & \frac{k_z}{k \cdot p_{N+1}} \end{pmatrix}^{-1} \cdot M_N \cdot \ldots \cdot M_2 \cdot M_1 \cdot \begin{pmatrix} 1 & 1 \\ -\frac{k_z}{k \cdot p_{N+1}} & \frac{k_z}{k \cdot p_{N+1}} \end{pmatrix} \tag{6}$$

In the above equations, for each of the i-th layer, p_i is the polarization multiplier factor for either s-or p-polarization. n_i is the refractive index of the i-th layer of thickness t_i. k_z is the component of the wavevector along the longitudinal axis, i.e. axis normal to the stack interface. Transfer matrix M_i for each of the layers are multiplied in ascending sequence to give to total **S**-matrix of the stack as shown in Eqn. 6. The first and the last matrix of the product series of Eqn. 6 are the transfer matrix of the last (i.e. $N+1$-th) layer (substrate) and the starting (i.e. 0-th) layer (superstrate). The reflectivity (or transmission) $\mathfrak{R}(\mathbf{n},\mathbf{t}|\lambda,\theta,\text{pol})$ obtained from the solver is used to compute the objective function to be minimized as shown in Eqn. 8, where $\mathfrak{R}^*(\lambda,\theta,\text{pol}))$ is the target spectrum. The objective is for the device to mimic the target spectrum $\mathfrak{R}^*$ as closely as possible. Consequently the loss function L amplifies this objective by mapping the function within an exponential function. K is the batch size and σ is a sensitivity hyper-parameter. The loss function in Ref. 5 had a mistake in the loss function L, that would cause the network to diverge. The problem was solved by incorporating a '-' sign in the loss function mentioned in the paper. The loss function used in this project is given in Eqn. 9

$$\mathfrak{R}(\mathbf{n},\mathbf{t}|\lambda,\theta,\text{pol}) = \left|\frac{S_{21}}{S_{22}}\right|^2 \tag{7}$$

$$O(\mathbf{n},\mathbf{t}) = (\mathfrak{R}(\mathbf{n},\mathbf{t}|\lambda,\theta,\text{pol})\bar{-}\mathfrak{R}^*(\lambda,\theta,\text{pol}))^2 \tag{8}$$

$$L \approx -\frac{1}{K}\sum_{k=1}^{K}\exp\left(-\frac{O(n^{(k)},t^{(k)})}{\sigma}\right) \tag{9}$$

The parameters of the model are optimized via Adam optimizer. We find that, for diminishing values of $O(n,t)$ L gradually converges to a value of -1. The network gradually converges on a subset of the total design space that give closest result to the target response. The network improves mapping of the random variable z to devices with optimized design for target response. The benchmark error rate for this project has been arbitrarily set to 5 percent. The error rate reported in different recent literature reporting similar works on photonic design via deep learning, vary from 3 % to 10 %.

4 Experiments and Data Acquisition

To implement and test the network, we designed two thin film stacks via the Res-GLONet: A simple three layer anti reflection (AR) coating and a narrowband filter with high delta-like transmission at 850 nm and 870 nm wavelengths for normal incidence. This generative network, works directly with the TMM solver. So, no explicit data or training samples necessary. The integration of the TMM solver with the model acts as the source of training. Over the course of training, the there is real time updating of the model wights propagated from the TMM solver.

Evaluation method

While the wavelength and angle range used in training is kept the same in evaluation of the network, the number of points and spacing between points are changed to make the results more general. The errors and average reflectivity reported here, were evaluated for this altered wavelengths and incidence angles vectors than used for training. The resulting thickness vector **t** and refractive indices **n** obtained from the trained network, is the input to the TMM solver, along with the wavelengths and incidence angles.

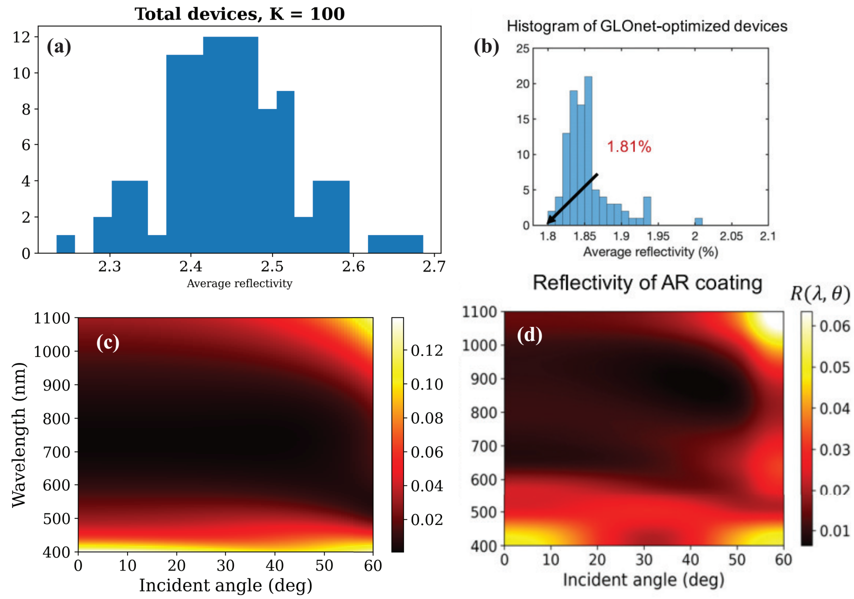

Figure 3: (a) The distribution of average reflectivity for a set of 100 randomly generated devices from the GLONet. (b) The same histogram as in (a) obtained from benchmark model of Ref. 5. (c) The reflection spectrum (averaged for both polarizations) for the best device design. (d) Benchmark spectrum for the best device generated from benchmark model. (e) Evolution of loss with number of iterations.

Table 1: Parameters of the network and TMM solver, for the three layer AR-coating design

Parameters	**Values**
Wavelength range (nm)	$\lambda = [400, 1100]$
Indcident angle range	$\theta_i = [0°, 60°]$
Polarization	Both
Material Library	$\{m_L, m_U\}$ = $\{1.09, 2.60\}$
Thickness range (nm)	$t \in [5, 200]$
Number of Layers (N)	3
Hyperparameters	
Batch Size (k)	50
Number of Iterations	100
σ	[0.3, 0.9] (0.8)
α	1 (fixed)
s_{res}	64
Learning rate	0.05

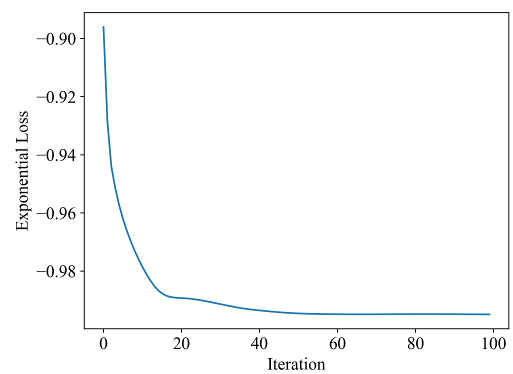

Figure 4: Evolution of loss with number of iterations for AR coating design network.

5 Analysis and Conclusion

Three layer AR-coating

A three layer AR-coating is implemented, where the refractive index **n** is considered to be a continuous variable (dispersionless refractive index), just like the thickness **t** of the layers. the index can vary continuously from 1.09 to 2.60. Air ($n = 1$) is selected as the superstrate and Silicon as the substrate. The network is trained to minimize reflection, i.e. target reflection is set to 0 for all wavelengths and incidence angles. The range of wavelengths and incidence angles is given in Table. 1. The batch size is important, because it tells the generator how much of the design space to search. A smaller batch size corresponds to the initial search of a smaller subset of the total design space. The hyper-parameter σ normalizes the objective function while higher value of α indicates greater discreteness (or selectivity) of the probability matrix used to sample refractive index from the set of the material library. Since $\mathbf{n}$ is a continuous variable and selection from the set of material library is not needed, α is fixed at 1. The results (i.e., reflectivity) are evaluated for a trained network, for the same range of λ and θ_i used for training, but with different number of points. The necessary parameters for training the generator and setting up the TMM solver is given in Table. 1. For a simple design such as this 3 layer stack, a shallow generator network of a single MLP network (i.e. no residual connections) give good evaluation results of up to a maximum of 6 % average reflectivity. Greater efficiency is observed when the residual blocks are included in the generator; average reflectivity of upto 2.1 % is observed (Fig. 3). The first column in Fig. 3 show the results obtained from our model, and the second column shows the results from the network implemented by Ref. 5. Our reported reflectivity is comparable within 1 % of the benchmarking paper.

Narrowband transmission filter:

To test the capability of the network, in designing a narrowband filter of a multilayer dielectric stack, we tested the Res - GLONet with the parameters mentioned in the Table. 2. Contrary to the previous case, the stack is now composed of a combination of seven materials from a given material library. So, the value of α is gradually increased from 1 to 1000 over the course of training, implying convergence of the probability matrix on a single material type for each of the $N = 50$ or $N = 100$ layers. At first the network was tested to design for 30 layers. But, the high deviation from the desired narrowband response, i.e., high loss function value even after multiple epochs of training, indicated that the device in the given design space simply did not posses the optical characteristics to mimic the narrowband response. So, the design space was consequently broadened, by increasing the layer number to 50, and then finally to 100. The loss initially seems to saturate at -0.62, but beyond 400 iterations further reduces to -0.95. Figure 5(b) shows that the minimum mean square error (MSE) of 10 percent is obtained. The spectrum could possibly be further improved by increasing the layer number. However, having low number of layers and consequently a small device footprint is also a competing objective of a good design. So, the layer number was not increased beyond 100. As seen in Fig. 5(c) a transmission response closely resembling the target spectrum was recorded for normal incidence.

Table 2: Parameters of the network and TMM solver, for the narrowband transmission filter design.

Parameters	**Values**
Wavelength range (nm)	$\lambda = [700, 900]$
Indcident angle range (for training)	$\theta_i = 0^\circ$
Polarization	TM
Material Library	$\{Al_22O_3$, MgF_2, TiO_2, SiC, SiN, SiO2, $HfO_2\}$
Thickness range (nm)	$t \in [0, 300]$
Number of Layers (N)	(50, 100)
Hyperparameters	
Batch Size (k)	500
Number of Iterations	1000
σ	0.2
α	$1 \rightarrow 1000$
s_{res}	64
Learning rate	0.05

6 Analysis and Conclusion

The high dimensional design space vector was decomposed via principal component analysis (PCA) and the the training samples gradually converge towards the optimized location in the design space at the end of training. It is to be noted, the convergence of various samples of $\mathbf{d}$ only depends on the training based upon the loss function and is independent of input random noise $\mathbf{z}$. Gradual evolution (top-to-bottom) of the principal component decomposed design space vector $\mathbf{d} \in \mathbb{R}^{N\times(M+1)}$ is seen for both the AR-coating

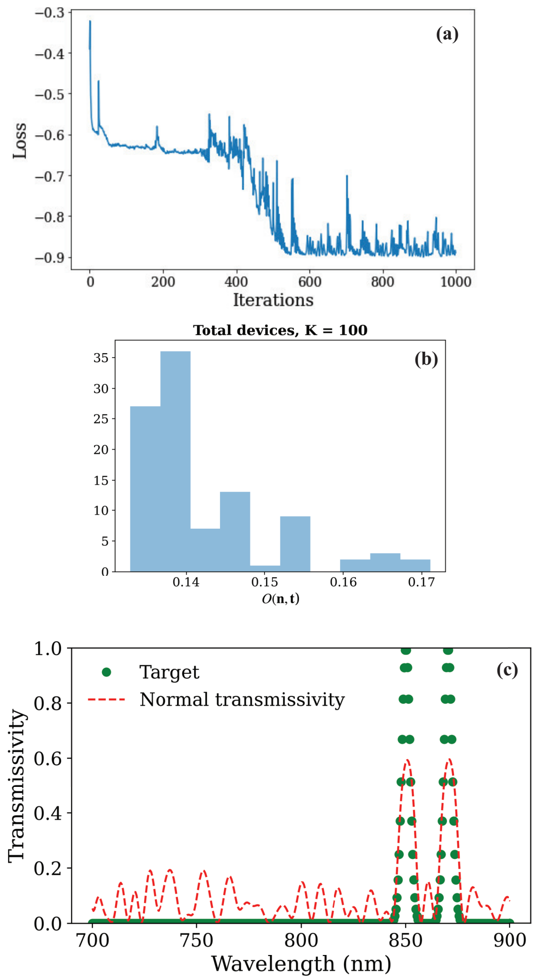

Figure 5: (a) Evolution of loss with number of iterations. (b) The distribution of MSE for a set of 100 narrowband filters randomly generated from the GLONet. (c) Target spectrum with the transmission spectrum of the narrowband filter film-stack.

and narrowband filter design problems. From start of training (top) to the final iteration (bottom) the gradual convergence of the samples to the optimized zone is easily noticeable in Fig. 6 and 7.

Further works

This project is a gateway prototype for further application of deep learning in the field of photonic design. A generative network has been implemented to converge on an optimized design through self-learning. This is advantageous as the burden of varying parameters of the EM solver is taken away from the designer. The model stochastically learns the characteristics of the device through the solver directly and hence no external data is needed. Further works in this domain can and should explore the effects and possibilities of different types of models as generators. Deep learning can and should also be used for augmenting numerical solvers for faster results to aid design and research of photonic design.

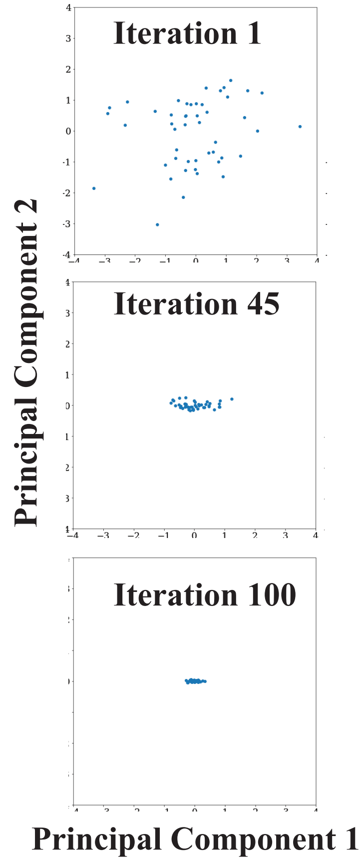

Figure 6: Evolution of the design space with the training of GLONet for AR-coating design problem.

References

[1] S. D. Campbell, D. Sell, R. P. Jenkins, E. B. Whiting, J. A. Fan, and D. H. Werner, "Review of numerical optimization techniques for meta-device design [Invited]," Optical Materials Express, vol. 9, no. 4, p. 1842, 2019.

[2] W. Li, F. Meng, Y. Chen, Y. f. Li, and X. Huang, "Topology Optimization of Photonic and Phononic Crystals and Metamaterials: A Review," Advanced Theory and Simulations, vol. 2, no. 7, p. 1900017, 2019.

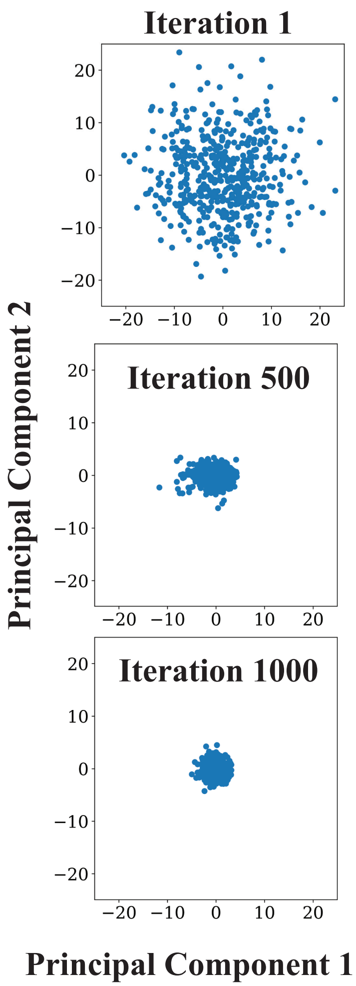

Figure 7: Evolution of the design space with the training of GLONet for AR-coating design problem.

[3] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, “Inverse design in nanophotonics,” Nature Photonics, vol. 12, no. 11, pp. 659–670, 2018.

[4] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, “Deep learning for the design of photonic structures,” Nature Photonics, vol. 15, no. 2, pp. 77–90, 2021.

[5] J. Jiang and J. A. Fan, “Multiobjective and categorical global optimization of photonic structures based on ResNet generative neural networks,” Nanophotonics, vol. 10, no. 1, pp. 361–369, 2020.

[6] J. Jiang and J. A. Fan, “Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network,” Nano Letters, vol. 19, no. 8, pp. 5366–5372, 2019.

[7] J. Jiang and J. A. Fan, “Simulator-based training of generative neural networks for the inverse design of metasurfaces,” Nanophotonics, vol. 9, no. 5, pp. 1059–1069, 2019.

[8] J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, “Free-Form Diffractive Metagrating Design Based on Generative Adversarial Networks,” ACS Nano, vol. 13, no. 8, pp. 8872–8878, 2019.

[9] S. J. Byrnes, “Multilayer optical calculations,” arXiv, 2016.

[10] M. Born and E. Wolf, Principles of optics: electromagnetic theory of propagation, interference and diffraction of light. Elsevier, 2013.