ORIGINAL ARTICLE

Supporting Information

Xiao Han[1,2]* | Ziyang Fan[1,2]* | Zeyang Liu[1]* | Chao Li[1,2] | L. Jay Guo[1]

1 | DETAILS ABOUT NEURAL NETWORK

We generated approximately 6500 pieces of data by RCWA[1]. 80 percent of them are used as training set and 20 percent as validation set. Each piece of data includes a shape, a period and the corresponding spectrum. The spectrum consists of 58 points, with half representing TE response and the other half TM response sampled at equal intervals between 400nm and 680nm. The shape is a 64×64 binary image. The period is an integer between 200 and 400nm.

The simulator consists of convolution layers extracting information from images and fully connected layers converting images into vectors. For the simulator, the data is augmented before being fed into the network. Adaptive moment estimation (Adam)[2] is used to update the gradient. Learning rate becomes smaller from 0.02 as the epoch becomes larger. We use a Telsa K80 GPU to train simulator with 500 epochs for approximately half an hour.

The generator consists of the deconvolution layers to generate images from sequences and the fully connected layers to obtain features from images. The shape is generated rst after deconvolution layers, then the period is produced after fully connected layers. Besides, we use a shortcut to make the training process more stable and fast. The input noise is sampled from a uniform distribution between 0 and 1. Adam is used to update the gradient, and the learning rate becomes smaller from 0.02 as the epoch becomes larger. We use a Telsa K80 GPU to train the generator with 1000 epochs for approximately an hour.

The detailed hyperparameters we used are listed in Table 1, and loss curves of simulator and generator are shown in Figure 1.

2 | ALGORITHM FOR GENERATING RANDOM STRUCTURES

`https://stackoverflow.com/questions/8997099/algorithm-to-generate-random-2d-polygon`

*Equally contributing authors.

TABLE 1 Hyperparameters used in the training of the simulator and the generator.

	Simulator	**Generator**
Epoch	500	1000
Batch size	1024	256
Optimizer	Adam (β_1=0.5, β_2=0.999)	Adam (β_1=0.5, β_2=0.999)
Learning rate	Initial=0.02, Step=100, γ=0.5	Initial=0.02, Step=200, γ=0.5
Initial strategy	Mean=0, Std=0.02	Mean=0, Std=0.02
Loss parameter	None	α=0.05, β=0

TABLE 2 The mapping relationship between color and function.

Color	Yellow	Cyan	Lavender	
Function	Convolution	Transpose Convolution	Full Connection	
Color	Blue	Orange	Purple	Red
Function	Normal Vector	Leaky ReLu (p=0.2)	Tanh	Sigmoid

3 | CONTRAST VECTORS

If we use the spectra consisting of 58 points instead of the contrast vectors, the loss curve of the generator suggests that the spectrum loss is smaller, with approximately 4% error rate. Nevertheless, if we feed an articial spectrum to it, the output results deteriorate a lot. As discussed in the paper, the suboptimal solution cannot be found because the network aims at ultimately minimizing MSE. In practice, the network pays more attention to the contrasts instead of inconsequential details of a spectrum, thus leading to a better result. Besides, the results are indifferent to the detailed features of the desired spectrum. The use of contrast vectors is essentially a trade-off for universality at the expense of network prediction accuracy.

REFERENCES

[1] Hugonin JP, Lalanne P, RETICOLO: Rigorous Coupled Wave Analysis for gratings with Matlab interface; 2005. `https://www.lp2n.institutoptique.fr/Membres-Services/Responsables-d-equipe/LALANNE-Philippe`.

[2] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 2014;.

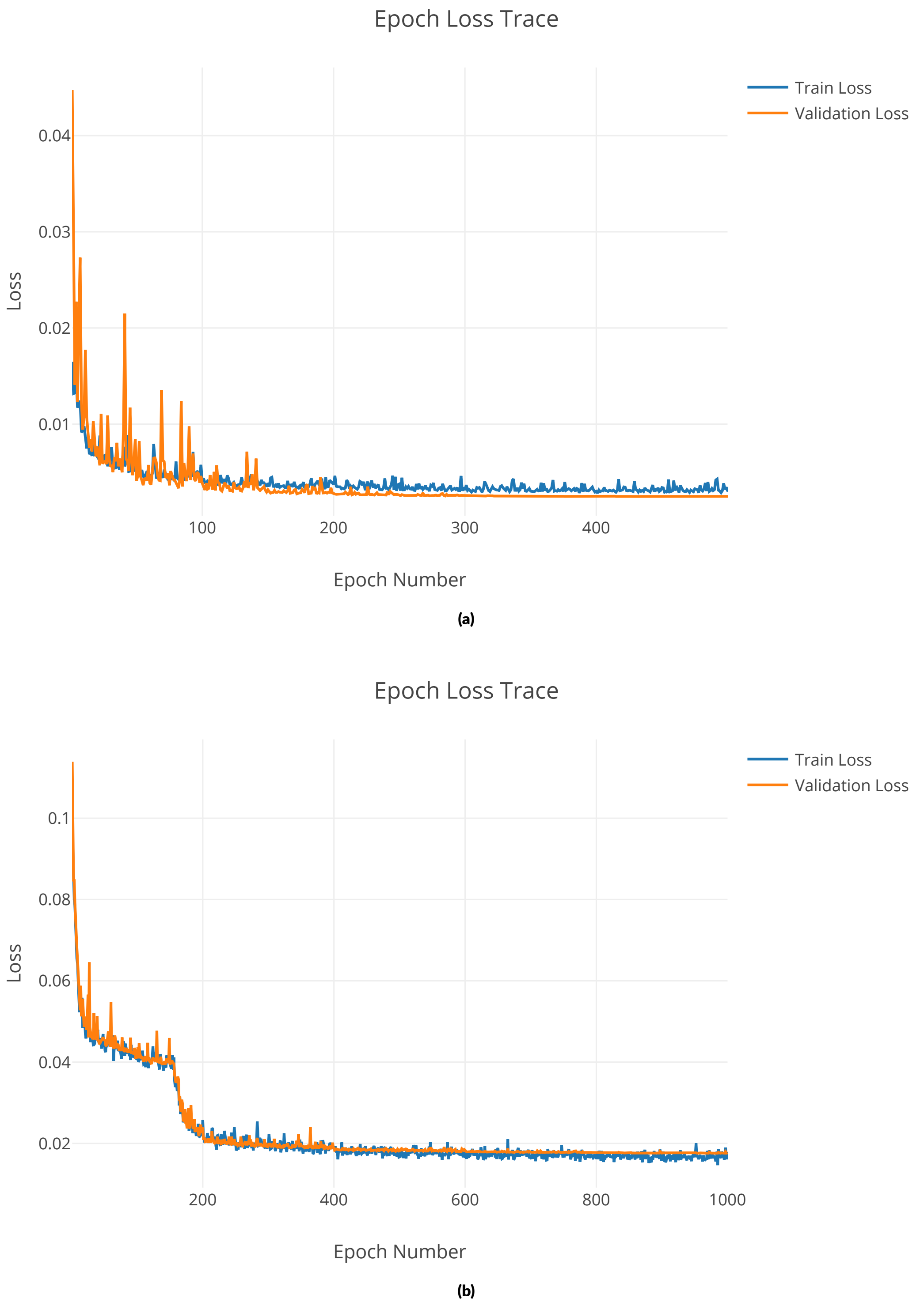

FIGURE 1 Loss curves of (a) simulator and (b) generator.