Mapping near field electromagnetic responses to sub-wavelength structures

Yixuan Shao
Stanford University
yixuans@stanford.edu

Abstract

When light interacts with an object, some of the wave is converted into the decaying mode, also known as the evanescent wave, which only exists at the surface of the object or the near field. Evanescent wave contains information of sub-wavelength structures. Conventionally, it's hard to interpret the complex near field and extract information about the object from it. Here, I train several deep learning network models to predict sub-wavelength structures' refractive index based on the input of the electric field at the surface of the object. The convolutional neural network (CNN) model could achieve a high prediction accuracy of 98%*. The saliency maps analysis also confirms that sub-wavelength information concentrates at the surface of the object.*

1. Introduction

Conventional optical imaging techniques' resolution is fundamentally restricted by the diffraction limit[1]. The electromagnetic wave component that contains sub-wavelength information, also known as the evanescent wave, decays exponentially over distance[2]. Therefore, cameras placed far from the scene can only capture features whose sizes are larger than the wavelength. So the best resolution for most optical imaging is on the order of the wavelength.

There are some existing techniques that can achieve higher resolution than conventional optical imaging, for example, photoactivated localization microscopy (PALM) [3] and scanning electron microscope (SEM)[4]. But PALM suffers from very long time of exposure and complicated sample preparation processes. It can also damage the sample. SEM usually requires the sample to be coated with conductive material, which is not always the case. These disadvantages make it hard to apply PALM and SEM widely.

In comparison, optical imaging is usually fast, convenient, and will not damage the sample. The idea is, that there's still rich information of the sub-wavelength features at the near field region, i.e. a few wavelengths distance from the object, where the evanescent wave doesn't decay a lot. As long as the detection probe is close enough to the object so that the evanescent wave is still strong, sub-wavelength information can be recovered from the near field[5]. This is useful as it provides a direct method to realize super-resolution imaging.

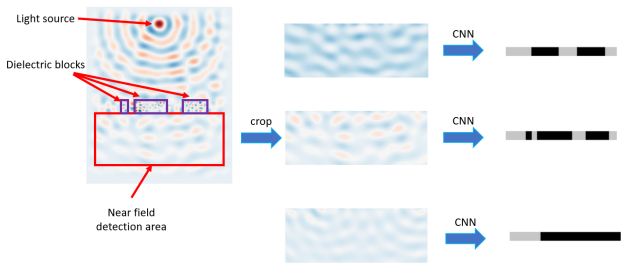

Figure 1. System setup. Some dielectric blocks are placed there and we would like to infer the position and length of each block from the detected electric field.

However, the iterations between light and matter are usually complicated and it's difficult to solve the sub-wavelength structure from the near field data. To solve this problem, here in this project I would like to use deep learning to solve the sub-wavelength structure from the near field electromagnetic response. The input for the model is the electric field scattered by the sample at the near field region. The output of the model is the refractive index profile of the sample. The detail will be discussed in the following paragraphs.

Figure 1 illustrates the system setup. A spherical wave source is placed at the top. It will emit light with a frequency ranging from 8 GHz to 12 GHz. The frequency range is chosen so that this idea could be experimentally demonstrated easily in the future if possible. There are several dielectric blocks placed at the horizontal plane 190 mm away from the source. The number of the blocks ranges from 1 to 6. The blocks can interact with the electromagnetic wave and produce complicated scattered fields. For practical reasons, we cannot detect the field inside the blocks. Instead, the electric field at the bottom region near

the block plane is detected. We would like to infer the position and length of each block from the detected electric field.

First I would like to discuss the input of my method. The detection region is 128 mm $\times$ 256 mm in size and sampled every 2 mm. So the detected electric field is an image of size 64×128. Note that the electric field is actually represented as complex numbers. In this project, I simply consider the real part and imaginary part of the electric field as two different channels. The electric field is sampled at 8 - 12 GHz with 1 GHz intervals. So there are 5 frequency channels. By combining the real part, imaginary part, and frequencies as one dimension, the electric field could be represented as an image of the size $64 \times 128 \times 10$.

Then I would like to discuss the output of my method. As the goal is to predict the position and length of each block, the output should contain information about them. I encode the block and air as binary numbers, with air represented as 0s and the dielectric blocks represented as 1s. The total length is 256 mm and is sampled every 1 mm. So we could use a 256-dimension vector composing binary numbers as the method output.

2. Related Work

This project is a novel work and there's no similar work before. As an alternative, I list some work that relates deep learning with electromagnetic wave here as a reference.

Deep network to predict electromagnetic field distributions from dielectric photonic structures[6, 7]. Although we could numerically solve Maxwell's equations to get the electromagnetic field distributions from dielectric photonic structures, this process is computationally expensive. In previous work [6], the researchers train a U-Net model that maps from dielectric structures to the electromagnetic response. The deep learning neural network could accelerate this computation process and support the backpropagation of the gradient. The drawback of deep learning is it suffers from large error, especially at the boundaries of structures. My work in this project is the inverse of this process, where I map the electromagnetic response back to the dielectric structures.

Designing dielectric structure distributions using generative adversarial networks[8, 9, 10, 11, 12]. This work [9] is similar to what I'm doing here, where both of us use a neural network to map the electromagnetic responses to dielectric structures. However, the difference is that the electric field I use here is measured by some probes and I would like to predict the structures that can produce such electromagnetic responses. I care more about the prediction accuracy. On the other hand, their work focuses on designing some unknown structures with some desired electromagnetic properties. They focus on optimizing the diffraction efficiency of the designed device. They apply a GAN model to transform random noise into the structure of optical grating. By optimizing the model, it could produce grating with higher and higher diffraction efficiency. This provides a new way to design optics.

Deep learning accelerated super-resolution localization microscopy[13, 14, 15]. In this work [13], deep learning is used to accelerate super-resolution single-molecule localization microscopy. The similarity is that both works realize optical super-resolution. However, they use far-field images as the input, which intrinsically loses the sub-wavelength information. As a compromise, the input images are required to be sparse so that super-resolution information could be preserved by the sparsity. The advantage of their method is that the accuracy is very high. In my project, I use near field as input, where sub-wavelength information is well preserved and the sparsity requirement is not required anymore. But as a compromise, the accuracy in this project may not be that high.

3. Dataset

The dataset is collected by solving the electric field distributions from given dielectric constant distributions, which could be considered an inverse of this problem. This process could be realized by numerically solving Maxwell's equations, which are a set of coupled partial differential equations that dictates the behavior of electromagnetic wave[2].

First I randomly generate 18,000 setups with varying numbers of blocks, block positions, and block lengths. The number of blocks ranges from 2 to 6. The minimum length of each block is set as 5 mm because too small blocks could have a negligible impact on the electromagnetic wave. For simplicity, the height of each block is set as a constant of 20 mm. As mentioned before, the blocks are encoded as 1s, and the air is encoded as 0s. The setup is represented as a 256-dimension vector composed of binary numbers. When generating the dataset, the vector is used to calculate the scattered electric field. But when training the deep neural network model, the vector will be the feature, or rather the output of the model.

Then each setup is interpreted as some dielectric constant distribution and the electric field distribution is solved from Maxwell's equations. Here I use Meep, an open-source software package that solves Maxwell's equations with the finite-difference time-domain (FDTD) method[16]. The near field electric field at 8, 9, 10, 11, and 12 GHz is extracted from the result at a sample rate of 0.5 $\text{mm}^{-}1$ with the resolution to be 64×128. The real part and imaginary part of the electric field are treated as two channels for each wavelength. So there are 10 channels for each image. Each image is of the shape $64 \times 128 \times 10$.

The dataset consisting of 18,000 datapoints is split into three parts, training set, validation set and test set. The train-

ing set has 14,000 examples, and the validation set and the test set each have 2,000 examples. Some of the examples from the training set are shown in Figure 1.

Finally, I preprocess the dataset. The electromagnetic wave is oscillating magnetic and electric fields. So the average value of the electric field is zero intrinsically. On the other hand, the standard deviation of the electric field represents the energy of the wave that depends on the spectrum of the light source. Because we hope our method can still work even if we use another light source in the experiment whose spectrum is different from the source used in the FDTD simulation, we need to normalize the standard deviation of the input electric field for each frequency.

4. Method

In this section, I will talk about the network architecture, loss function, and evaluation metrics I use for my model.

4.1. Architecture

As shown in Figure 1, the input images include features like ripples. Because the dielectric blocks have a different refractive index compared to air, the wave scattered by the block has a different phase profile. The waves scattered by different blocks interfere with each other, which ends up in the ripples in the images. So the ripples in the image contain much information about the blocks. It is well known that convolutional neural networks are good at identifying these features in the images. Therefore, it would be a good idea to learn these wavefront shapes using convolution kernels. I would like to finetune some well-known CNN architecture such as VGG and ResNet and apply them to this problem. As a benchmark, I would like to use a fully-connected network and compare it with the convolutional neural networks.

4.1.1 Fully-connected network

I use a two-layer fully-connected network as the baseline method. The input image is flattened to a vector of the shape 1×81920. The hidden layer consists of 4000 neurons. Because the output of the model should be a 256-dimension vector, I add another fully-connected layer whose output dimension is 256 at the end. In addition, because the prediction result is binary, a Sigmoid function is applied to the output.

The number of variables in the fully-connected network is as large as 328,716,256. Though the fully-connected network has much more variables compared to other CNN models I use in this project, later we will find that the fully-connected network performs worst among these methods. The reason is that the fully-connected network cannot make use of the spatial correlation between neighboring pixels.

4.1.2 Modified VGG network

Very deep convolutional networks are proven to be powerful for large-scale image recognition problems[17]. Different from the fully-connected network, the convolution operation can evaluate the correlation between neighboring pixels and thus can recognize translationally-invariant features in the image. I modify the VGG 11-layer model from [17] here. The network consists of two parts. The input image first goes through a few convolutional layers with 3×3 filters. Batch normalization, ReLU nonlinear layers and 2×2 max-pooling layers are added following the convolution layers. Note that there's no batch normalization layer in the vanilla VGG network. However, without batch normalization layers the network suffers a lot from gradient explosion and gradient vanishing problems, and that makes it very hard to find good hyper-parameters for the model. To solve this problem, I add batch normalization layers to the VGG model. Because the number of input channels is 10, I change the first convolution layer and set its input channel to be 10. In the second part, there are three fully-connected layers at the end of the model to map the result to the desired dimension. Similarly, as the output should be a 256-dimensional vector, I modify the last fully-connected layer and set its output size to be 256.

Although the network uses small filters of the size 3×3, by stacking convolution layers together, we can increase the effective receptive field. So the network can still recognize larger features.

4.1.3 Modified ResNet

To make a deeper model easier to train, ResNet is invented [18]. By adding identity mappings to the network as shortcuts, a deeper model can learn to be at least as good as a shallower model. The residual blocks try to learn a residual mapping instead of the desired underlying mapping. As ResNet performs very well in the image classification task, it is expected that it can also do well in this task. Here I modify an 18-layer ResNet structure to fit it into this problem's input and output size. To be specific, I change the first convolution layer and set its input channel to be 10 and set the last fully-connected layer's output size to be 256.

4.1.4 Implementation

The implementation of these network architectures is based on PyTorch [19]. I import the VGG model with batch normalization layers and the ResNet model from torchvision. I modify some layers to fit in my problem, as mentioned before. However, I don't use the pre-trained parameters in these models. Even though we can see the basic features in the input images are common in other image datasets, and the convolution kernels in the very first layer of VGG

or ResNet may be useful. But because the number of input channels is 10 instead of 3, it's hard to reuse the features in 3D space and apply them to 10D space. That's why transfer learning could be hard to be applied here.

4.2. Loss function

In this problem the model only needs to predict whether the structure is air or dielectric block at each position, which is a binary classification problem. I choose to use binary cross-entropy loss during the training process, which is defined as

$$Loss(\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log \hat{y}_i + (1-y_i)\cdot \log(1-\hat{y}_i)\,. \quad (1)$$

Here $\hat{y}_i$ is the prediction of the model on the ith example, which is a vector of 256 real numbers between 0 and 1. y_i is the groundtruth of the ith example, which is a vector of 256 binary numbers. N is the total number of examples.

4.3. Evaluation Metrics

I will use the average prediction accuracy to evaluate the accuracy of the model's prediction. For each input x_i, which is the ith example image of the size $64 \times 128 \times 10$, the prediction accuracy on this example is

$$\text{accuracy}_i = \frac{1}{256}\sum_{j=1}^{256}(y_{ij} == \hat{y}_{ij}) \quad (2)$$

The model could be evaluated using the average accuracy over all datapoints.

$$\text{average accuracy} = \frac{1}{N}\sum_{i=1}^{N}\text{accuracy}_i \quad (3)$$

5. Experiments/Results

5.1. Hyperparameters

Through experiments I find that the stochastic gradient descent algorithm (SGD) works better for the fully-connected network model. But Adam works better for the VGG model and the ResNet model. Adam combines the advantage of momentum and RMSProp. It's usually more robust and is less likely to be trapped in a saddle point [20]. I find the batch size doesn't have a big impact on the result and I use 50 for all models. To find the best learning rate for each model, I randomly choose different learning rates for each model and train each model for 50 epochs. The training accuracy curve is shown in Figure 2. We can see the accuracy gap between the training set and validation set is small, indicating that the model is not overfitting and the training set is large enough.

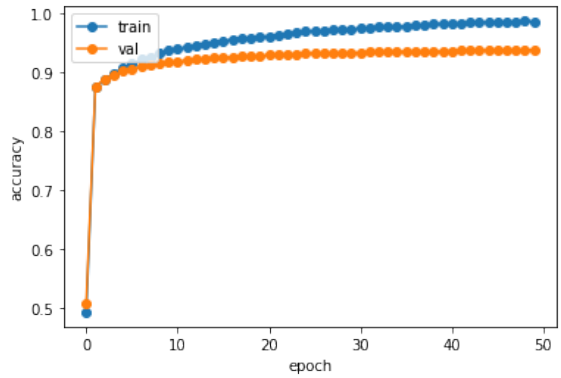

Figure 2. Training accuracy curve versus the training epoch on both training set and validation set.

	FC	VGG	ResNet
Accuracy	94.2%	96.7%	98.1%
# parameters	328,716,256	129,824,704	12,247,296
Best learning rate	5.0×10^{-3}	1.3×10^{-4}	2.8×10^{-4}

Table 1. Summary of the accuracy, number of parameters, and learning rate of the three models

Then I select the best learning rate based on the evaluation on the validation set. The best learning rates for fully-connected network model, VGG network model, and ResNet model are 5.0×10^{-3}, 1.3×10^{-4}, and 2.8×10^{-4}, respectively.

5.2. Prediction Accuracy

The prediction accuracy of the fully-connected network model, VGG network model, and ResNet model on the validation set are 94.2%, 96.7%, and 98.1%, respectively. Obviously, CNN-based models perform better than the fully-connected network model. This also validates that convolutional neural networks are good at identifying ripples-like features in the input images. In addition, the number of parameters in the three models are 328,716,256, 129,824,704, 12,247,296, respectively. This is interesting because the performance of the three models shows a negative correlation with the number of parameters. That's because the fully-connected layers are wasteful as they contribute a lot to the number of parameters but help less with the improvement of the model. The results are also summarized in Table 1.

5.3. Failure Cases Analysis

I include some prediction results and compare them with the ground truth in Figure 3. In the first example (figures in the first two rows), there are only two dielectric blocks in the ground truth, which is a relatively simple setup. We can see the predictions by both CNN models are very accurate, whereas the prediction by the fully-connected net-

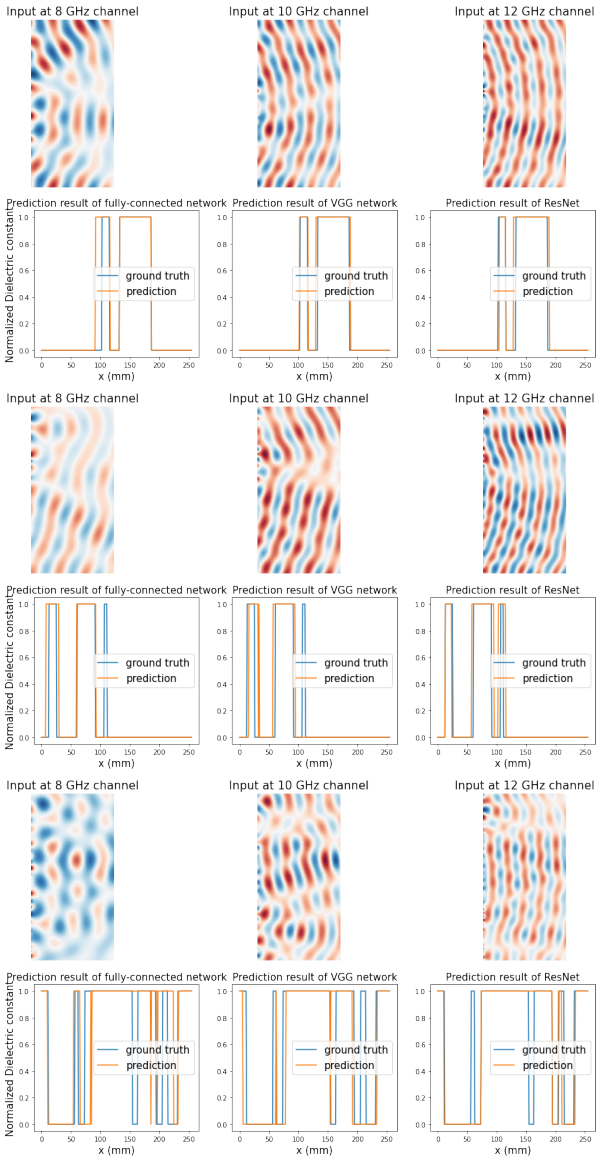

Figure 3. The input electric field images at 8 GHz, 10 GHz, and 12 GHz (upper), and the comparison of the prediction and ground truth (lower) on three different cases.

work model is close but not very accurate. We can see that the size of the first block is smaller than the prediction by the fully-connected network model. This indicates the fully-connected network model is not as good as CNN-based models at identifying spatially correlated features.

In the second example (the middle two figures), there are three dielectric blocks in the ground truth. The size of the third block from 105 mm to 111 mm is only 6 mm wide, which is only 0.2 wavelength. Both the fully-connected network model and VGG model fail to predict the existence of this small block. Although the ResNet model successfully predicts the existence of the block, its predicted block size is larger than the actual size of the block. Therefore the network models are not good at identifying very small blocks. This is understandable because such a small block can have little impact on the propagation of the electromagnetic wave, leaving very few clues in the input images.

A similar problem also happens in the third example (the two figures on the bottom). There're 6 blocks in this case and some of the blocks are very small. We can see all the three models fail to predict the accurate number of blocks. The common mistake is failing to predict the existence of one of the dielectric blocks or air blocks.

5.4. Saliency Maps Analysis

Saliency Maps describes which parts of the input images have the most impact on the classification result [21]. But different from the vanilla definition of the saliency maps for a single-label classification, in this problem there are 256 labels for each input image. So I cannot reuse the same definition of the saliency maps. In order to describe which parts of the input images are most important to all of the 256 prediction results, I backpropagate the network from the loss function and calculate the absolute value of the gradient of the input image, which is then the saliency maps in this problem.

Some of the examples of the saliency maps are shown in Figure 4. We can see that the most salient parts are always at the very left edge. So the prediction results are mostly based on the information from the left edge of the input. This surprising phenomenon actually accords very well with the physics nature of the evanescent wave. On the one hand, note that the wave propagates from left to right and the dielectric blocks are placed next to the left edge of the images. This indicates the wave that is just immediately scattered by the dielectric blocks contains most information of the blocks. On the other hand, the wave component that contains sub-wavelength information is the evanescent wave, which decays over distance. Therefore, the information flow of the block information also decays over distance. That's why we see the most salient part on the left, where the information doesn't decay much.

The analysis also provides me with some clues on how to improve the prediction accuracy of the model in practice. As the information of the sub-wavelength structures concentrates at the immediate surface of the block, the detection area should be as close as possible to the surface of the blocks. Experiments also show that if the detection area is detached from the block surface, the prediction accuracy decreases rapidly. Besides, it would be better if we have more sampling points at the left part of the input images. Then more information would be provided for the model and the prediction accuracy could be improved.

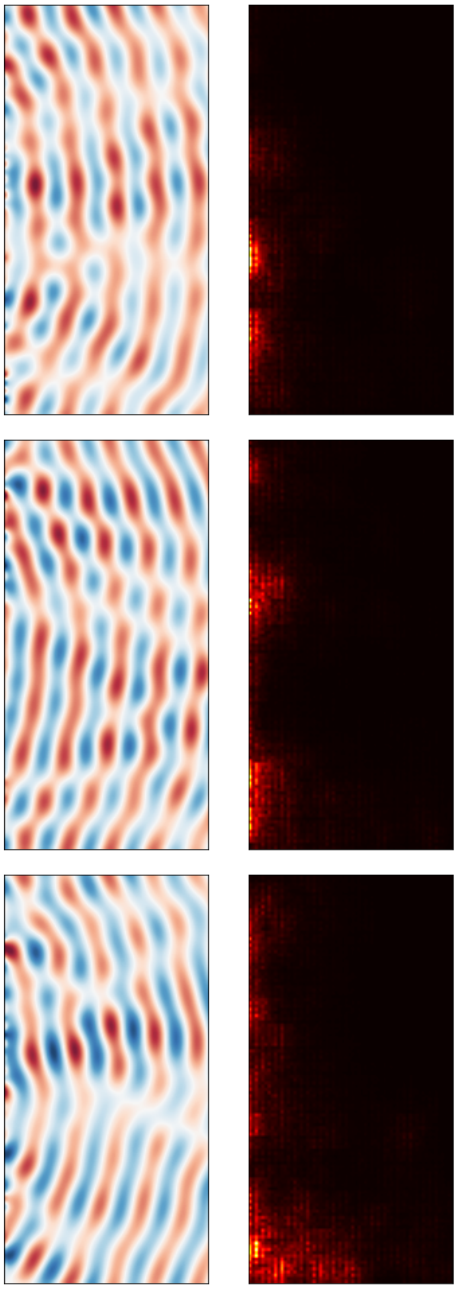

Figure 4. Left: input images at 12 GHz. Right: saliency maps of the input images

6. Conclusion

In this project, I implement a high-precision optical super-resolution imaging technique. By learning from the scattering pattern of dielectric structures, deep learning models could predict the sub-wavelength structures of the dielectric blocks based on the electric field. The experiment results show that the ResNet model performs best on this task, followed by the VGG model, and the fully-connected network model performs worst. The wave nature of the electric field makes the input image full of ripples-like features. Convolutional networks could effectively identify these features and extract information about blocks' structure from them. That's why CNN models can perform better than the fully-connected network model. I also analyze why the models don't give good predictions on some cases. Because the scattering by a very small (0.2 wavelength size) can be negligible to the electromagnetic wave, it's hard for the models to predict small features accurately. Finally, I analyze the saliency maps and connect them to the decaying property of the evanescent wave.

For future work, I would like to explore the possibility of applying this model to three-dimensional scenarios. In this project, the model only predicts the one-dimensional refractive index profile. In the real world, the refractive index could be a three-dimensional distribution we could also detect the electric field in three dimensions. There would be interesting scenarios in the three-dimensional space. For example, one material is embedded in another material. From the perspective of far field optical imaging, it's hard to see the embedded structure. But it would be possible to recover the embedded structure by analyzing the near field and modeling it.

7. Contributions and Acknowledgements

In this project, all the work is done by myself. PyTorch is used for building and training deep learning models. Meep is used for numerically solving electromagnetic responses and creating the dataset.

References

[1] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013. 1

[2] John David Jackson. Classical electrodynamics, 1999. 1, 2

[3] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *science*, 313(5793):1642–1645, 2006. 1

[4] D McMullan. Scanning electron microscopy 1928–1965. *Scanning*, 17(3):175–185, 1995. 1

[5] Eric Betzig, A Lewis, A Harootunian, M Isaacson, and E Kratschmer. Near field scanning optical microscopy (nsom): development and biophysical applications. *Biophysical journal*, 49(1):269–279, 1986. 1

[6] Mingkun Chen, Robert Lupoiu, Chenkai Mao, Der-Han Huang, Jiaqi Jiang, Philippe Lalanne, and Jonathan A. Fan. WaveY-Net: physics-augmented deep-learning for high-speed electromagnetic simulation and optimization. In Connie J. Chang-Hasnain, Jonathan A. Fan, and Weimin Zhou,

editors, *High Contrast Metastructures XI*, volume 12011, pages 63 – 66. International Society for Optics and Photonics, SPIE, 2022. 2

[7] Peter R Wiecha and Otto L Muskens. Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures. *Nano letters*, 20(1):329–338, 2019. 2

[8] Jiaqi Jiang, David Sell, Stephan Hoyer, Jason Hickey, Jianji Yang, and Jonathan A Fan. Free-form diffractive metagrating design based on generative adversarial networks. *ACS nano*, 13(8):8872–8878, 2019. 2

[9] Jiaqi Jiang and Jonathan A Fan. Global optimization of dielectric metasurfaces using a physics-driven neural network. *Nano letters*, 19(8):5366–5372, 2019. 2

[10] Mingkun Chen, Jiaqi Jiang, and Jonathan A Fan. Design space reparameterization enforces hard geometric constraints in inverse-designed nanophotonic devices. *ACS Photonics*, 7(11):3141–3151, 2020. 2

[11] Jiaqi Jiang and Jonathan A Fan. Simulator-based training of generative neural networks for the inverse design of metasurfaces. *Nanophotonics*, 9(5):1059–1069, 2020. 2

[12] Fufang Wen, Jiaqi Jiang, and Jonathan A Fan. Robust freeform metasurface design based on progressively growing generative networks. *ACS Photonics*, 7(8):2098–2104, 2020. 2

[13] Wei Ouyang, Andrey Aristov, Mickaël Lelek, Xian Hao, and Christophe Zimmer. Deep learning massively accelerates super-resolution localization microscopy. *Nature biotechnology*, 36(5):460–468, 2018. 2

[14] Elias Nehme, Lucien E Weiss, Tomer Michaeli, and Yoav Shechtman. Deep-storm: super-resolution single-molecule microscopy by deep learning. *Optica*, 5(4):458–464, 2018. 2

[15] Tianyang Zhou, Jianwen Luo, and Xin Liu. Deep learning for super-resolution localization microscopy. In *Optics in Health Care and Biomedical Optics VIII*, volume 10820, page 1082023. International Society for Optics and Photonics, 2018. 2

[16] Allen Taflove, Susan C Hagness, and Melinda Piket-May. Computational electromagnetics: the finite-difference time-domain method. *The Electrical Engineering Handbook*, 3, 2005. 2

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 5