



Deep learning for accelerated all-dielectric metasurface design

CHRISTIAN C. NADELL,¹ BOHAO HUANG,¹ JORDAN M. MALOF,¹
AND WILLIE J. PADILLA^{1,*}

¹Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27705, USA

*willie.padilla@duke.edu

Abstract: Deep learning has risen to the forefront of many fields in recent years, overcoming challenges previously considered intractable with conventional means. Materials discovery and optimization is one such field, but significant challenges remain, including the requirement of large labeled datasets and one-to-many mapping that arises in solving the inverse problem. Here we demonstrate modeling of complex all-dielectric metasurface systems with deep neural networks, using both the metasurface geometry and knowledge of the underlying physics as inputs. Our deep learning network is highly accurate, achieving an average mean square error of only 1.16×10^{-3} and is over five orders of magnitude faster than conventional electromagnetic simulation software. We further develop a novel method to solve the inverse modeling problem, termed fast forward dictionary search (FFDS), which offers tremendous controls to the designer and only requires an accurate forward neural network model. These techniques significantly increase the viability of more complex all-dielectric metasurface designs and provide opportunities for the future of tailored light matter interactions.

© 2019 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

All-dielectric metasurfaces (ADMs) have recently been shown to exhibit a number of interesting and useful properties, including max nullity absorption, [1] full color printing, [2] highly efficient arrays for tailored reflection, [3] and antireflective functionality [4]. Similar to metal-based metamaterials and metasurfaces, ADMs exhibit electric (EH_{111}) and magnetic (HE_{111}) modes, both of which are tunable with geometry, [5] or external stimuli [6]. However, ADMs offer further advantages which are of great utility compared to their metallic counterparts. For example, they lack Ohmic loss, may possess low thermal conductivity, can survive to high temperatures; and their operation and performance are not based on electrical conductivity, thus avoiding constraints set by the Wiedemann-Franz law. [5]

In principle these promising structures can be combined using supercell designs – like those in metal-based metasurfaces [7] – in order to achieve even greater functionality (see Fig. 1(a)). However, unlike metal-based metamaterials, structures which constitute ADMs possess an evanescent mode tail which couples strongly to neighboring cells, and is similar to those found in dielectric waveguides. For example, Figures 1(b) and (c) show the simulated frequency dependent transmission ($T(\omega)$) in the terahertz (THz) range for an infinite square array of cylindrical resonators embedded in vacuum, as shown in the insets of (b) and (c). The orange curve is for a cylinder with a radius and height of $r = 44.5\mu\text{m}$, $h = 30\mu\text{m}$, respectively, and the blue curve is $T(\omega)$ for a cylinder that is somewhat taller ($h = 42.5\mu\text{m}$), but with the same radius $r = 44.5\mu\text{m}$. In Fig. 1(c) we show the transmission that results from combining the two unit cells shown in (b), where we simply form a 2×2 super unit cell which uses three $h = 30\mu\text{m}$ cylinders and one with $h = 42.5\mu\text{m}$. As can be observed, there are several new modes, denoted by the black arrows, including cusp-like transmission, which is evident in the super unit-cell spectra, but importantly are not observed in either of the single unit-cell simulations. In theory, the complexity of these

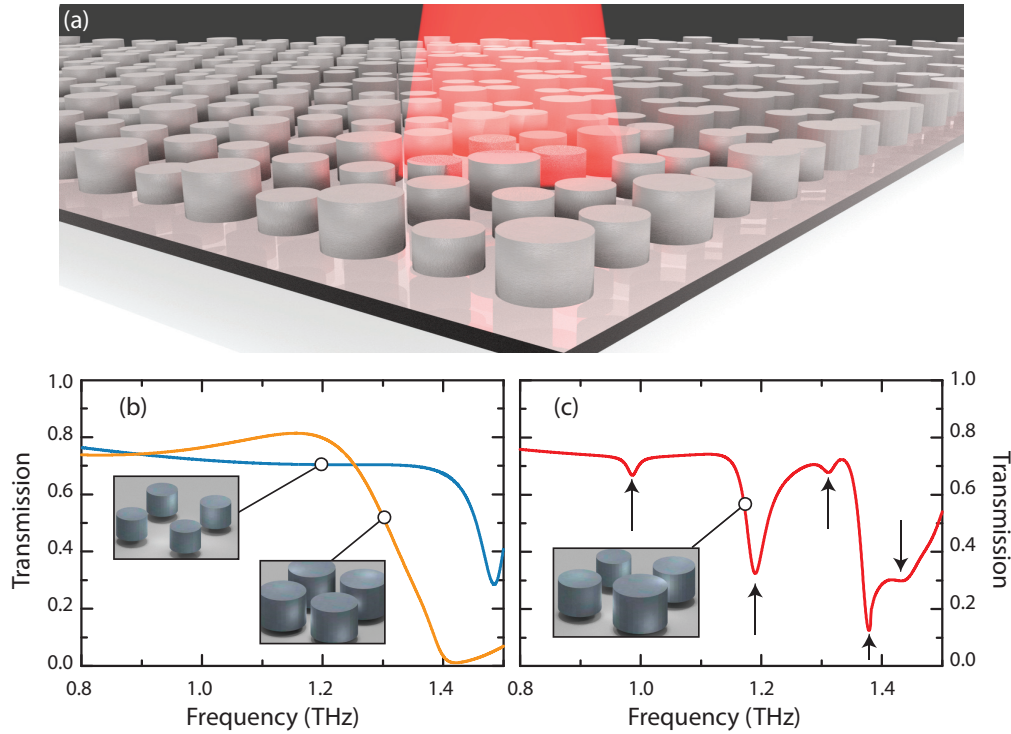


Fig. 1. (a) Illustration of the all-dielectric metasurface consisting of a square array of cylindrical resonators. (b) Simulated transmittance spectra for a single sub-unit cell structure consisting of a cylindrical square array (insets) with $r = 44.5\mu\text{m}$, $h = 30\mu\text{m}$ (orange curve), and $r = 44.5\mu\text{m}$, $h = 42.5\mu\text{m}$ (blue curve). (c) $T(\omega)$ of a 2×2 super unit cell (inset) consisting of three $r = 44.5\mu\text{m}$, $h = 30\mu\text{m}$ cylinders and one $r = 44.5\mu\text{m}$, $h = 42.5\mu\text{m}$ cylinder. Black arrows denote modes that result that were not present in (a).

spectral features can be utilized for more intricate frequency-dependent scattering designs, and further expanded if larger supercells are employed.

While supercells may yield rich and exotic scattering properties, designing ADMs to achieve a desired transmission – more generally a scattering parameter (S-parameter) – is currently difficult. Conventional methods for designing metasurfaces involve large trial and error searches over candidate metasurface geometries. Each candidate must be validated, and then optimized, using electromagnetic simulations. On conventional computational hardware these simulations are time-consuming, permitting only a fraction of candidate geometries to be explored. This is tolerable for simpler metasurfaces, which involve a limited set of geometric configurations. However, the space of geometric configurations in supercell ADMs grows rapidly with respect to the number of the supercells, resulting in a substantially larger set of candidate geometries to explore. This problem is exacerbated by the longer simulation times required to accurately model the complexity of supercell ADMs, meaning only a small fraction of designs can be considered.

In this work, we show that deep learning methods can be used to overcome the design challenges of supercell ADMs. We develop a deep neural network (DNN) that solves the *forward problem* of metasurface design – given any geometry (x), predict the corresponding S-parameters (s), i.e. $x \rightarrow s$. We also design and implement an approach to solve the *inverse problem* of metasurface design: given a desired S-parameter spectrum, denoted s^* , find the geometry, denoted x^* , that will yield s^* , i.e. $s^* \rightarrow x^*$. We also quantify the performance in terms of the average mean

squared error, and determine if our model learned the underlying physics or if it is simply a good interpolator.

In recent decades, DNNs and deep learning have made great leaps forward in learning complex functions [8, 9] e.g., solving computer vision [10–12] and natural language processing problems. [13–16] Based upon the available physics of ADMs, and our own observations from simulations, we expect the underlying relationship between x and s to be non-linear and highly complex. Therefore, due to their universal approximation capabilities [17, 18], and their recent successes solving complex problems, DNNs are a good class of machine-learning algorithms to infer the relationship. Our goal is to use deep learning to ascertain a function of the form $s = f(x; \theta)$ where $x \in \mathbb{R}^{N_x}$ is an N_x -dimensional vector parameterizing the geometry of our metasurface, and $s \in \mathbb{R}^{N_s}$ is the corresponding N_s -dimensional S-parameter of that metasurface. In our work the function f is a deep neural network parameterized by a set of weights θ . The values of θ control the precise form of the the function f and can be inferred automatically using a "training dataset" of (x, s) pairs obtained via simulations. During training the network's weights are adjusted incrementally in order to maximize the accuracy of f on the available training data. The performance of the trained network is evaluated on a "validation dataset" of (x, s) pairs that were not provided to it during training, thereby assessing its ability to make accurate predictions in a practical setting.

2. Metasurface design

Our work here focuses on a supercell ADM comprised of four cylindrical unit cells, as shown in Fig. 1. Each unit-cell cylinder is parameterized by a radius and a height (two parameters), yielding a 8-dimensional parameterization, x , for the geometry of the supercell. We select silicon (Si) for the metasurface material and, with no loss of generality, embed it within vacuum for the purposes of faster training set acquisition. Silicon represents a very common material often utilized in metasurface design, both for traditional metal-dielectric based metasurfaces, and for more recent demonstrations of all-dielectric metasurfaces.

Once we established effective simulation settings, we were able to collect the training dataset by running different x -values through the simulator to obtain their corresponding s values. In order to avoid favorably biasing the DNN's performance on any subset of geometries (and therefore negatively on others), we built up a training set by randomly sampling the full geometric hyperspace. Table 1 shows the discrete values from which geometry vectors were randomly sampled. The spacing between the discrete values was chosen to be dense enough to capture most spectral variability, while being distant enough to avoid substantial redundancy among the spectra of neighboring geometries. To identify this compromise we swept over the parameter space to observe how much spectral change is obtained for a given geometric change. Using the resulting discretized values in Table 1 yields approximately $13^8 \approx 815.7$ million possible geometries. We evaluated the MSE and found that it dropped to low values once our training set approached $\sim 21,000$ simulated spectra, and thus used a total dataset of this size for our investigation. Of this set, we randomly selected 3,000 pairs to act as a validation dataset. Our resulting training dataset therefore contained 18,000 pairs, or only about 0.0022% of all possibilities.

We also specify in Table 1 the scale of the metasurface geometry in comparison to the wavelength of light to which it is responding. In our study, we use a fixed supercell periodicity of $p = 217\mu\text{m}$, corresponding to a central frequency of $f_0 = c/\lambda_0 = c/p = 1.38\text{ THz}$, where $c = 2.99792 \times 10^8\text{ m/s}$ is the speed of light in vacuum, and λ_0 is the central wavelength. Thus, with regard to this central wavelength the minimum and maximum ratios of radius and height to p are $(r/p = 19\%, 24\%)$ and $(h/p = 14\%, 25\%)$, respectively. We seek a general framework for designing metasurfaces with deep learning, and therefore we intend these values as a means of comparison with other work that may fall in another region of the spectrum, as all-dielectric metasurfaces can easily be geometrically scaled to another band.

Table 1. Hyperspace parameter values for grid definition. All values are in units of microns.

r	r/p	h	h/p
42.0	0.19	30.0	0.14
42.8	0.20	32.0	0.15
43.7	0.20	34.0	0.16
44.5	0.21	36.0	0.17
45.3	0.21	38.0	0.18
46.2	0.21	40.0	0.18
47.0	0.22	42.5	0.20
47.8	0.22	44.0	0.20
48.6	0.22	46.0	0.21
49.5	0.23	48.0	0.22
50.4	0.23	50.0	0.23
51.2	0.24	52.0	0.24
52.0	0.24	55.0	0.25

3. Deep neural network architecture

Figure 2 shows an illustration of the overall DNN design we use in this work. As input to the network, we provide the 8-dimensional unit-cell parameters, along with ratios of those parameters, so that $x = \{h_1, h_2, h_3, h_4, r_1, r_2, r_3, r_4, \frac{r_1}{h_1}, \frac{r_1}{h_2}, \dots\}$. We hypothesize that the derived input quantities, i.e. $\frac{r_i}{h_j}$, are likely needed, or useful, to compute the S-parameter of the metasurface. While DNNs can still approximate these functions through training, doing so requires many extra network parameters [19], which tends to increase the difficulty of parameter inference (i.e., optimization), and thereby leading to yield poorer generalization. Therefore, pre-computing these quantities - motivated by knowledge of the underlying physics - may improve network performance. Indeed similar arguments provide theoretical support for the so-called "tensor layer" [20], which essentially pre-computes products of the input variables. Our inclusion of ratios as input to our DNN is intended to serve the same purpose, since we hypothesize that these functions are useful for computing the S-parameter, and may also be difficult for the DNN to approximate from training data. Indeed, we found that adding the ratios consistently improved the accuracy of our trained DNNs—on average about 12%. This approach is not necessarily beneficial if there is no apriori reason (e.g., based upon knowledge of physics) to believe that ratios will be useful statistics to estimate the spectra.

In the DNN, our augmented geometry vectors enter a sequence of nine fully connected layers of sizes $\{100, 500, 1000, 1500, 500, 2000, 1000, 500, 165\}$, all of which are subject to L_2 regularization. Since our goal is to achieve a good fit using the minimum amount of training data, we find L_2 regularization is necessary to mitigate the network overfitting to the training set, which would be detrimental to the performance of the validation set. The fully connected layer stack handles a large portion of the upsampling task, overall converting the length 24 input vector to a length 165 output vector. These fully connected layers are followed by three transposed-convolution layers [21], and finally a convolution layer [22] smooths out the

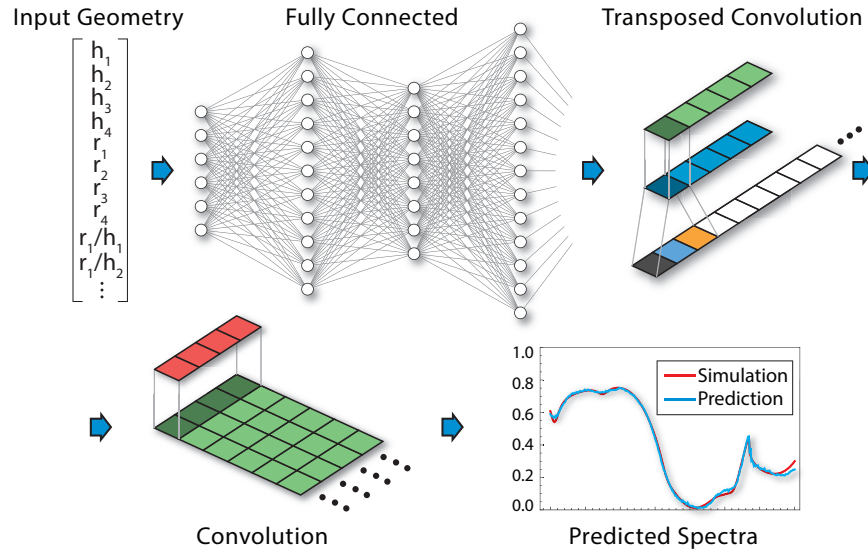


Fig. 2. An illustration of the neural network architecture. Fully connected layers are fed a set of geometric inputs. Data is then smoothed and upsampled in a learnable manner via transpose convolution layers (top row). The final layer (bottom row) is a convolutional layer, which produces a predicted spectra, shown as the blue curve, compared to the ground truth (red curve).

predicted spectrum. These convolution and transposed convolution layers perform the rest of the upsampling task, and allow points of similar frequency to be closely related in a learnable manner via the associated filters. Lastly, in order to mitigate edge effects due to zero-filling in the convolutional operation, we drop 15 points from either end of the output data tensor, producing a spectrum of 300 points.

4. Forward neural network model

Figure 3 displays examples of the network's predictions for a few input geometries from the validation set, *i.e.* for (a) through (c) the height and radii $\{h_i, r_i\}$ geometry vectors are given by $\{55, 50, 52, 48, 44.5, 49.5, 46.2, 47.8\}$, $\{50, 50, 36, 36, 43.7, 45.3, 50.4, 43.7\}$, and $\{50, 42.5, 48, 42.5, 42.8, 48.6, 47, 47\}$ in μm . Here we have chosen some of the best fits in order to demonstrate the capabilities of the network, and thus many exhibit MSE better than the mean – a value of 1.24×10^{-3} averaged over three models with this particular architecture. As can be observed, the predicted frequency dependent transmittance spectra (blue curves) match the simulated $T(\omega)$ spectra (red curves) well for a variety of different spectral shapes, including cusp behavior at ~ 1.4 THz in (d), and the $T \approx 0$ near 1.3 THz in (i). The predicted transmittance (T_{pred}) and simulated transmittance (T_{sim}) overlap significantly such that we plot, on the right axis (shaded gray area), the absolute value of the difference in transmittance, defined as $|T_{pred} - T_{sim}|$. Figure 3(m) shows a histogram of the MSE for evaluation set predictions, where the average mean squared error across all evaluated spectra was 1.16×10^{-3} for the highest performing model, and it is important to detail that 99% of the data has an error less than 6.2×10^{-3} , and 95% of the data has $\text{MSE} \leq 3.4 \times 10^{-3}$ indicated by the dashed vertical gray line.

Some work has been done recently using DNNs to solve the forward problem for material design problems, such as modeling local surface plasmonic resonance in metallic nanoparticles [23] and transmission in multilayer photonic films [24]. In prior work [20] good results were obtained through the use of a tensor module, which effectively generates more geometric inputs. However,

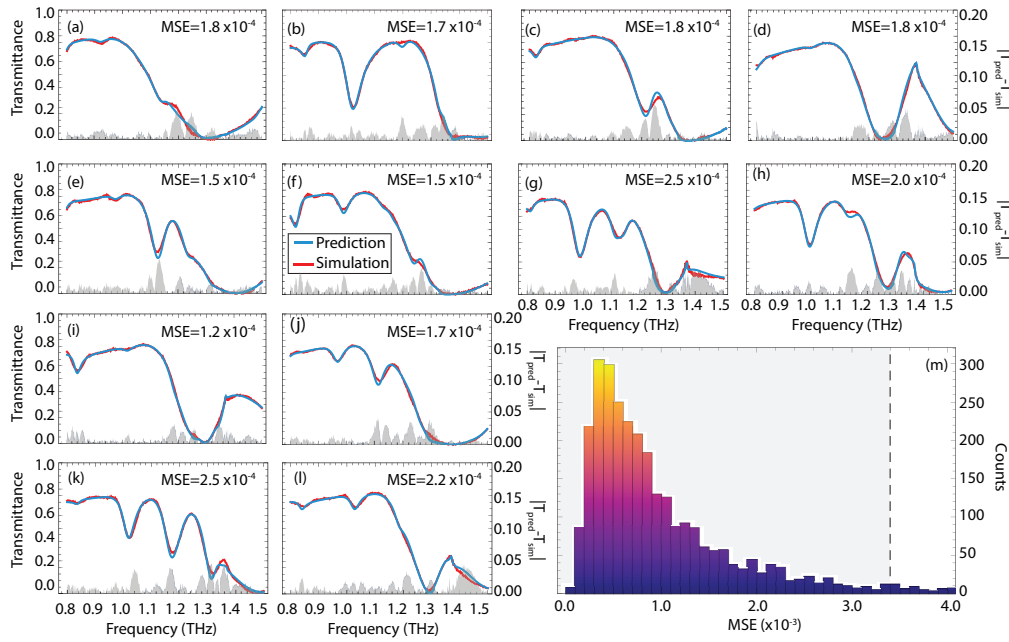


Fig. 3. (a)–(l) Network predictions of the frequency dependent transmittance (blue curves) and simulated $T(\omega)$ spectra (red curves) demonstrating excellent prediction accuracy for a variety of spectral features and input geometric parameters. The average MSE is listed in each sub-panel. The shaded gray area shows the absolute value of the difference in predicted and simulated transmittance, i.e. $|T_{pred} - T_{sim}|$, shown on the right vertical axis. In (m) a histogram of the MSE for all geometries in the evaluation set is shown, where 95% have $MSE \leq 3.4 \times 10^{-3}$, as indicated by the dashed vertical gray line.

for our particular problem, we found that despite varying the number of branches and the size of dense layers within a similar tensor module, we only achieved worse performance (see Table 2).

5. Inverse problem

The inverse problem presents several major challenges compared to the forward problem. First, for a given s^* specified by a designer, there may be no x that can realize s^* . Second, there is the one-to-many mapping problem [25, 26], i.e. the fact that many very different geometric structures can produce extremely similar spectra. Therefore, given some s^* , there may be multiple valid x^* that approximate s^* . Any solution to the inverse problem would ideally: (i) indicate if any valid solution exists, and (ii) provide a set of the designs that can most closely approximate the specified s^* . Work within the last year has made some progress towards building such algorithms - mostly DNNs - that can address these problems [27–31] however, they suffer from several drawbacks. First, most (but not all) require developing and validating a second or third DNN model, which requires a cross-validation process of the same magnitude as the forward model - a time-consuming process. Although some prior work avoids training a second model explicitly, [27, 30], to our knowledge these approaches provide no assurance of returning all (or even most) of the designs that may approximate a given s^* . Ideally we would like the inverse model to provide a list of the nearest realizable material configurations, even when s^* is not a realizable solution given the problem constraints. We next propose a novel solution to the inverse problem which does not suffer from any of these limitations.

6. Fast forward dictionary search

Our approach to solve the inverse problem – termed fast forward dictionary search (FFDS) – exploits the extremely efficient processing capabilities of deep neural networks on standard inexpensive desktop graphics processing units (GPUs). To solve the inverse problem, we generate the entire set of spectra given by all combinations of geometric parameters on the hypergrid – a total of $13^8 \approx 815.7$ million spectra. This is feasible because, on our Tesla Quadro M6000 GPU, we are able to compute $\sim 9,400$ spectra per second (sp/s), permitting us to find $T(\omega)$ for a massive number of geometric input values within a relatively short time. The result of this process is a dictionary, or lookup table, L , with individual entries of the form $l_i = (x_i, s_i)$. In order to find the geometry that yields s similar to s^* , we can simply search through our dictionary to find the $s_i \in L$ that most closely matches s^* . Mathematically, this search problem can be written as,

$$x^* = \arg \min_{x_i \in L} d(s_i, s^*) \quad (1)$$

where x^* is the candidate returned and d is the chosen distance operator, which in our case is given by the MSE. Note however that d can take numerous forms, providing tremendous flexibility to the metasurface designer. For example, d might be defined as the city-block distance (known also as L_1 norm), or it might only operate on a subset of the dimensions of s that are deemed most important. Furthermore, instead of returning the single best x -value, FFDS can return a list of (x, s) pairs, which can then be visually inspected in order to identify the best solution. Although the dictionary can be quite large, possibly making the evaluation of Eq. (1) slow, we can leverage highly-efficient evaluation algorithms that have been developed in recent years. [32] Therefore, benefiting from recent advances in hardware, deep learning, and computer sciences, FFDS provides a practical solution to solve inverse modeling problems.

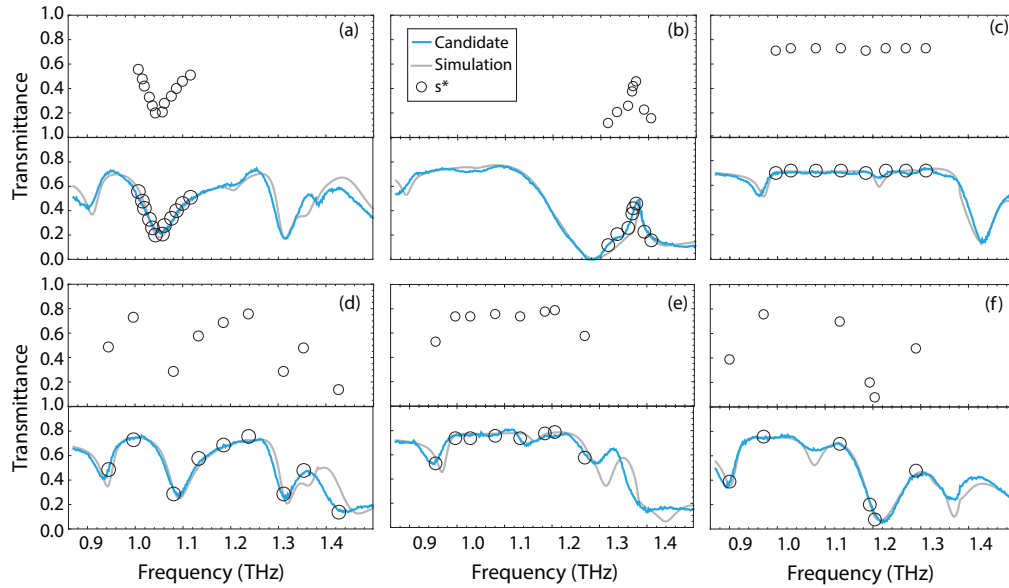


Fig. 4. Top panels of (a) – (c) show hand-picked T points (open symbols) demonstrating a trough, cusp behavior, and a flat transmission, respectively. The bottom (a) – (c) panels replot desired T points (open symbols) and show candidate spectra produced by FFDS (blue curves) compared to simulated $T(\omega)$, shown as the gray curves. Top and bottom panels (d) – (f) depict of other examples of desired $T(\omega)$ and the resulting FFDS result candidates and post-lookup simulated spectra.

In order to demonstrate the utility of our approach, in Fig. 4 we show several hand-picked frequency-dependent T points (ranging from six to twelve) in the spectral range from 0.9 - 1.5 THz (open black symbols), and find spectra that most closely match. We select different frequency dependent spectral features including a smooth minimum (Fig. 4(a)), cusp-maximum (Fig. 4(b)), flat transmission band (Fig. 4(c)), and other various combinations (Figs. 4(d)-(f)). Figure 4 shows candidate spectra (solid blue curves) for each, resulting from lookup table searches of the various spectral features. In addition, we have simulated the geometries returned by the lookup search, verifying that the network has made reasonable predictions. Simulated spectra from the geometry predicted by our FFDS algorithm are shown as the gray curves in Fig. 4, and show good agreement with the lookup table candidates (blue solid curves) generated by the model, as well as the hand-picked points (open symbols).

As is evident from the plots, s^* may be specified with any number of points greater than zero, bounded by the number of points in the predicted spectrum (in this case 300). Further, these points can be distributed at arbitrary indices on $[0, 299]$, (in our case corresponding to a frequency range of 0.86-1.5 THz), such that only regions of interest are specified. This provides greater flexibility in that certain areas can be excluded from affecting whether the search algorithm determines a good match with s^* has been found. The measure of distance can also be modified to suit the desired application. For example, a higher order L_p norm (for $p > 2$) can be used to penalize outliers more heavily, or a distance based on Huber loss could reduce their impact. Here however, we use the mean squared difference in order to achieve a balance between the two. We have specified the example s^* in Fig. 4 to showcase the metasurface structure's potential to fit a variety of transmission spectra, but obviously there are many s^* for which the fits would be very poor. As the number of geometrical degrees of freedom in a metasurface structure increases however, the breadth of achievable functionality expands rapidly and many more s^* become viable.

Although FFDS has many advantages, one important limitation is that L will grow rapidly with respect to the number of design parameters, N_x (i.e., the dimensionality of the geometry parameterization), potentially making it impractically slow to evaluate Eq. (1). However, FFDS is a special case of the nearest neighbor search problem, [32] for which there are extremely efficient approximate algorithms to evaluate Eq. (1). For example, methods such as the Randomized kd-forest and the hierarchical k-means priority search [32] can accelerate the evaluation of Eq. (1) by several orders of magnitude. Although these algorithms are approximate, they are highly accurate, and because FFDS can return a (short) list of solutions, any inaccuracies are largely mitigated. Finally, all approaches to evaluate Eq. (1) can readily be distributed over many computer nodes, providing approximately-linear speed increases with respect to the number of available nodes.

Although FFDS suffers as N_x grows, it should be noted that all machine-learning based approaches suffer in both speed and accuracy as N_x grows (e.g., the curse of dimensionality). We note that FFDS has several advantages over existing approaches for inverse design. First, it only requires training a forward DNN model, as opposed to training an inverse model, which is generally much more difficult and time-consuming. Furthermore, contemporary inverse modeling approaches typically produce just one solution, even though there may be many x values that each exhibit approximate solutions to a given s^* . FFDS can provide a list of all x that approximate s^* , allowing the designer to choose among multiple solutions with varying properties. Furthermore, if no inverse solution exists, FFDS will return the best approximate solutions.

Our current implementation of FFDS requires only about 23 hours to generate the entire dictionary using only a single desktop machine with one GPU, and non-optimized code for writing each geometry-spectrum pair to disk. Parallelizing the system with multiple machines could greatly enhance the speed of dictionary generation, making FFDS viable for even more complex physical systems. Even so, the supercell structure we have studied here is substantially

more complex and powerful than many metasurfaces in recent literature, which suggests a need for an inverse solution that can quickly design structures of this complexity. The FFDS approach offers speed and accuracy of design for a set of metasurfaces that are not easily accessible with conventional methods, as well as providing the flexibility and reliability that inverse DNN approaches currently lack. FFDS thus fulfills a significant design need in the field of metamaterials and exhibits distinct advantages over inverse DNN methods for a broad class of metasurfaces, although it may not scale well for very large problems, i.e., if N_x is very large.

7. Peering into the black box

Our final contribution is to investigate if our DNN learned a general underlying physical relationship between x and s , or whether it is simply interpolating between the data samples it was trained upon. Some work has been done in recent years to extract physical understanding of metamaterials from neural networks, [33] however, we assert that developing DNNs that "learn the physics" in this way is an important prerequisite if we are to ultimately use them to uncover physical relationships, and thereby advance the science. Towards this goal, we propose a simple experimental procedure to help answer this question, and we find that the answer, in our case, is that our DNN is largely a very good interpolator. This contrasts somewhat with the conclusions made in recent work, [30], using a more lenient testing criterion. Our results therefore suggest that much more work is needed by our research community to achieve this goal.

To explore the extent of the model's recognition of the physical patterns of the system, we perform cross-validation using the same network architecture described previously. We cut out a corner of the full eight-dimensional geometry hyperspace by constraining the training set to use only geometries with either $r_i \leq 48.6$, $i \in \{1, \dots, 4\}$ or $h_j \leq 46$, $j \in \{1, \dots, 4\}$ resulting in $\sim 8,600$ training samples, while the remaining $\sim 12,000$ are reserved for validation of the trained model. Removal of this training data provides us a measure of the model's extrapolation capability. To visualize the dependence of MSE on distance outside the training boundary (extrapolation), we project all geometry vectors in the validation set onto a two-dimensional space, as shown in Fig. 5. First, we project by taking the maximum of both the heights h_i and radii r_j , respectively, shown in Fig. 5(a). Due to the hypergrid spacing many points overlap in this projected space, and so we average the MSE of overlapping points and normalize to the resulting maximum. Evidently the model performs significantly better for geometries near the training set boundaries, while the MSE degrades with increased distance from the boundaries in a fairly smooth manner.

Secondly, we project onto two-dimensional space by taking the means of h_i and r_j respectively, shown in Fig. 5(b). This results in many of the geometry points mapping to radii and heights that are below the hyperspace training boundary values. Further, it illustrates a key point about the corner cross-validation technique: points in the validation set typically still have some values of r and h that the network has seen before during training. Additionally, we note that r_i/h_j values are not constrained for this cross-validation so their values may be shared between the training and validation sets. Nonetheless, Fig. 5 illustrates that while the model interpolates very efficiently, it is not capable of extrapolating very well.

Despite difficulties with designing generalizable network models, deep learning approaches to metasurface design may still prove to be more than blackbox models for generating desired spectra and geometries. First, it may be possible to identify regions of normalized wavelength where different physical models are required to accurately capture the relationship between geometry and spectra. This could be accomplished by holding out different regions of the geometry hyperspace and training models only on what remains. As our work here suggests, evaluating models on the hold-out data in this case could provide a measure of how well the regressed network function generalizes, and test how accurately it represents the underlying physical pattern. Further, computing feature extraction as in a tensor layer or single dense layer

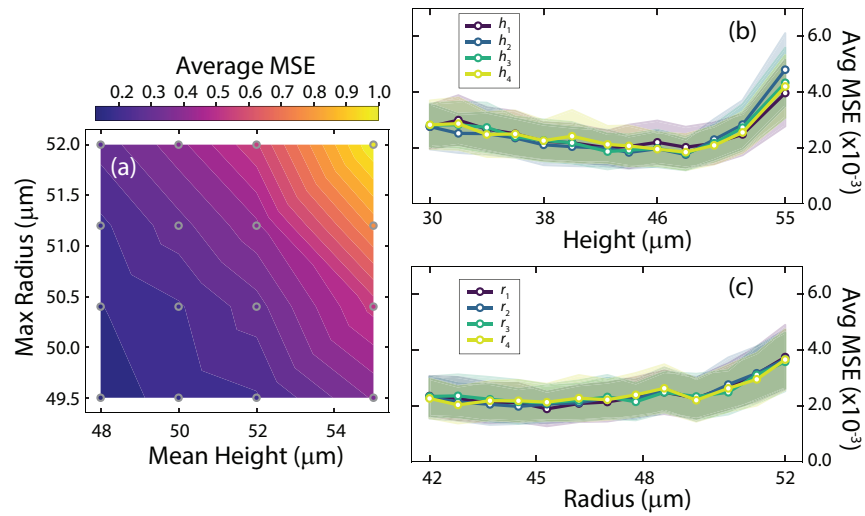


Fig. 5. (a) MSE error for geometrically-constrained cross-validation set after training, as described in the text. Color map shows normalized error for a max function projection (maximum MSE= 0.0057) with interpolation between the sampled points (open gray symbols). Shown in (b) and (c) are average MSE values for different values of input geometric parameter as they approach the training boundary at $h = 55\mu\text{m}$ (b), and $r = 55\mu\text{m}$ (c), with ribbons indicating standard deviation scaled by 1/3.

could prove useful in determining derived geometric quantities of interest. Strong dependence on particular network nodes can be interrogated via various deep learning interpretability tools, such as activation visualization histograms [34]. These may provide physical insights for even more complex systems that may be intractable with conventional analytical methods.

Our cross-validation results suggest that deep learning approaches to spectral prediction are able to interpolate very complex mapping functions with relatively little data, but they do not extrapolate well outside of the training space, which suggests that they do not learn physical principles of any generalizability. Previously it has been suggested that performing well on a validation set implies that neural networks can generalize the physics of a system [30]. At best this is too weak a requirement for claiming that underlying physical patterns have been learned. Good performance on validation data inside the training hyperspace is the minimum criterion for determining if a model has any value, while extrapolation and cross-validation performance are more representative measures of whether a model can generalize well. Further, the results of our extrapolation test in this work suggest that similar deep learning models – rather than learning generalizable physics – are limited to highly efficient nonlinear interpolators.

8. Network optimization

Our forward model results demonstrate high-accuracy predictions of spectra corresponding to a broad range of geometries. The trained regression model allows us to obtain spectra from geometry far more quickly than conventional electromagnetic simulations, albeit with lower fidelity. We found that results were notably improved by the inclusion of radii to height ratios, whereas without the r_i/h_j ratios, averaged over 3 trained models, we achieved an MSE of 1.41×10^{-3} . In addition, we explored the use of a tensor module in our neural network, both with and without using r_i/h_j ratios. We found the best (with no r_i/h_j ratios) results with a two-branch, 13-unit dense layer configuration; here the average MSE we achieved over 3 runs was 1.87×10^{-3} . In Table 2 we summarize the MSE for various configurations of our deep

Table 2. MSE averaged over three runs for various configurations of the network

MSE	Tensor Module	Geometric Ratios
1.24×10^{-3}	✗	✓
1.32×10^{-3}	✓	✓
1.41×10^{-3}	✗	✗
1.87×10^{-3}	✓	✗

neural net using r_i/h_j ratios, and the tensor module. The precise effect and best usage of the tensor module is unclear and presents an opportunity for further study, and our positive results with r_i/h_j ratios (rather than the $r_i h_j$, $r_i r_j$, and $h_i h_j$ products associated with the tensor layer) suggest that it may be worth pursuing for future metamaterial deep learning work. From a metasurface design perspective, the imperfect accuracy of our forward models is greatly mitigated by the FFDS method. Geometries corresponding to a very broad array of spectral shapes can be approximately found very quickly, and then finely tuned later with classical algorithms such as Nelder-Mead or trust region framework. This process can significantly decrease the design time of complex metasurface structures compared with conventional semi-analytic, numerical simulation, or physical intuition based methods.

9. Conclusion

We have designed and implemented a deep neural network capable of accurately predicting the frequency dependent transmission of an all-dielectric metasurface, given a set of eight geometrical parameters. The dielectric metasurface is an infinite square array – a super unit-cell which consists of four cylindrical structures, each of which may take on various values of radius and height – a problem which is challenging to solve with conventional computational approaches, and lacks a complete physical theory. Our DNN calculates spectra at a rate of ~ 9400 spectra/sec, and is thus $\sim 8.2 \times 10^5$ times faster than a conventional electromagnetic solver running on the same hardware. We have investigated various configurations of our DNN architecture, in order to reduce the amount of data necessary to achieve good performance with deep networks for metasurface design, including using derived geometric parameter inputs, permitting solution of the forward model using only 0.0022% of all total possibilities. The implemented neural network achieved an average mean squared error of 1.16×10^{-3} and 99% of the data had MSE less than 6.2×10^{-3} . In addition, we have introduced the FFDS as a method to solve the inverse materials design problem, overcoming difficult one-to-many mapping problems and greatly enhancing the viability of deep learning for metasurfaces design. Lastly, we have investigated the extrapolation capabilities of the trained model via a cross-validation approach, which indicates that a class of deep learning models likely exhibit merely efficient interpolation rather than generalizable physical pattern recognition, and in doing so we have identified an important line of inquiry for future work.

10. Experimental design

10.1. Numerical simulations and material properties

We model Si with a Drude model using a high frequency permittivity $\epsilon_\infty = 11.7$, plasma frequency $\omega_p = 7.3 \times 10^{12}$, and a collision frequency of $\gamma = 1 \times 10^{13}$. Prior work has shown that the above described Drude model is accurate for the THz range of the electromagnetic spectrum. [35, 36] We choose a frequency domain solver within the commercial simulation

software CST Microwave Studio in order to make use of periodic boundary conditions, which are essential for accuracy in cases like our present one where inter-cell interactions can be strong. To ensure accuracy in the output spectra we first engaged in a systematic study of the simulation settings in order to verify S-parameter convergence, including mesh cell size, solver order, and critically, the number of Floquet modes accepted at the boundary ports. The mesh is tetrahedral and on average comprises about 20,000 cells. We employ a minimum of 3 adaptive mesh passes for accuracy, and use the solver of 2nd order while requiring an accuracy of 1×10^{-5} . Crucially, we allow 10 Floquet modes to be accepted at each port, finding this is sufficient for good accuracy while fast enough to acquire large datasets in a reasonable amount of compute time. Simulated spectra are composed of 2,003 points evenly spaced over 0.8 to 1.5 THz, and are downsampled by pure decimation to 300 points for network training.

10.2. Deep neural network

The network architecture begins with a set of nine dense layers of sizes {100, 500, 1000, 1500, 500, 2000, 1000, 500, 165}. The transposed layers follow with dimensions {165, 165, 330} and filters of size {8, 4, 4}. The final convolution layer is built to match the transpose output, with dimensions of {1, 4, 1} and stride 1. Truncation then drops 15 points from either end of the data tensor. We train the network using Adam (an adaptive learning rate optimization algorithm), with a mean squared error loss function. This is paired with a learning rate adhering to TensorFlow's stepped decaying exponential with learning rate 1×10^{-4} , decay step of 20,000, and decay rate of 0.5. Training converged after about 45,000 training steps, or the equivalent of approximately 3 epochs.

Funding

Department of Energy (DOE) (DE-SC0014372); Alfred P. Sloan Foundation through the Duke University Energy Data Analytics Fellowship; Duke University Energy Initiative.

Acknowledgments

C. Nadell and W. Padilla acknowledge support from the Department of Energy (DOE) (DE-SC0014372). Support for Bohao Huang was provided by the Alfred P. Sloan Foundation through the Duke University Energy Data Analytics Fellowship. Support for Jordan Malof was by the Duke University Energy Initiative.

References

1. J. Y. Suen, K. Fan, and W. J. Padilla, "A Zero-Rank, Maximum Nullity Perfect Electromagnetic Wave Absorber," *Adv. Opt. Mater.* **1801632**, 1–6 (2019).
2. S. Sun, Z. Zhou, C. Zhang, Y. Gao, Z. Duan, S. Xiao, and Q. Song, "All-dielectric full-color printing with tio2 metasurfaces," *ACS Nano* **11**, 4445–4452 (2017). PMID: 28317376.
3. D. Headland, E. Carrasco, S. Nirantar, W. Withayachumnankul, P. Gutruf, J. Schwarz, D. Abbott, M. Bhaskaran, S. Sriram, J. Perruisseau-Carrier, and C. Fumeaux, "Dielectric resonator reflectarray as high-efficiency nonuniform terahertz metasurface," *ACS Photonics* **3**, 1019–1026 (2016).
4. C. C. Chang, L. Huang, J. Nogan, and H. T. Chen, "Invited Article: Narrowband terahertz bandpass filters employing stacked bilayer metasurface antireflection structures," *APL Photonics* **3** (2018).
5. X. Ming, X. Liu, L. Sun, and W. J. Padilla, "Degenerate critical coupling in all-dielectric metasurface absorbers," *Opt. Express* **25**, 24658 (2017).
6. J. Sautter, I. Staude, M. Decker, E. Rusak, D. N. Neshev, I. Brener, and Y. S. Kivshar, "Active tuning of all-dielectric metasurfaces," *ACS Nano* **9**, 4308–4315 (2015).
7. X. Liu, T. Tyler, T. Starr, A. F. Starr, N. M. Jokerst, and W. J. Padilla, "Taming the blackbody with infrared metamaterials as selective thermal emitters," *Phys. Rev. Lett.* **107**, 045901 (2011).
8. M. Ziatdinov, O. Dyck, A. Maksov, X. Li, X. Sang, K. Xiao, R. R. Unocic, R. Vasudevan, S. Jesse, and S. V. Kalinin, "Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations," *ACS Nano* **11**, 12742–12752 (2017).

9. F. Lussier, D. Missirlis, J. P. Spatz, and J.-F. Masson, "Machine learning driven surface-enhanced raman scattering optophysiology reveals multiplexed metabolite gradients near cells," *ACS Nano* **13**, 1403–1411 (2019).
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, (2012), pp. 1097–1105.
11. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).
12. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, (2016), pp. 21–37.
13. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems*, (NIPS, 2013), pp. 1–9.
14. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, (NIPS, 2017), Nips.
15. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805 (2018).
16. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**, 436 (2015).
17. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks* **2**, 359–366 (1989).
18. G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Contr. Sign. Syst.* **2**, 303–314 (1989).
19. H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?" *J. Stat. Phys.* **168**, 1223–1247 (2017).
20. W. Ma, F. Cheng, and Y. Liu, "Deep-Learning-Enabled On-Demand Design of Chiral Metamaterials," *ACS Nano* **12**, 6326–6334 (2018).
21. M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (IEEE, 2010), pp. 2528–2535.
22. V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," arXiv preprint arXiv:1603.07285 (2016).
23. C. Chen and S. Li, "Valence electron density-dependent pseudopermittivity for nonlocal effects in optical properties of metallic nanoparticles," *ACS Photonics* **5**, 2295–2304 (2018).
24. Y. Qu, L. Jing, Y. Shen, M. Qiu, and M. Soljačić, "Migrating knowledge between physical scenarios based on artificial neural networks," *ACS Photonics* **6**, 1168–1174 (2019).
25. J. L. Müller, *Linear and Nonlinear Inverse Problems with Practical Applications (Computational Science and Engineering)* (Society for Industrial and Applied Mathematics, 2012).
26. D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures," *ACS Photonics* **5**, 1365–1369 (2018).
27. I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, "Plasmonic nanostructure design and characterization via Deep Learning," *Light. Sci. Appl.* **7** (2018).
28. H. Kabir, Y. Wang, M. Yu, and Q. Zhang, "Neural Network Modeling and Applications to Microwave Design," *IEEE Trans. Microw. Theory Tech.* **56**, 867 (2008).
29. I. Sajedian, J. Kim, and J. Rho, "Predicting resonant properties of plasmonic structures by deep learning," arXiv preprint arXiv:1805.00312 (2018).
30. J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.* **4** (2018).
31. Y. Kiarashinejad, S. Abdollahrameazni, and A. Adibi, "Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures," arXiv preprint arXiv:1902.03865 (2019).
32. M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Analysis Mach. Intell.* **36**, 2227–2240 (2014).
33. Y. Kiarashinejad, S. Abdollahramezani, M. Zandehshahvar, O. Hemmatyar, and A. Adibi, "Deep learning reveals underlying physics of light–matter interactions in nanophotonic devices," *Adv. Theory Simulations* **0**, 1900088.
34. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," arXiv preprint arXiv:1506.06579 (2015).
35. X. Liu, K. Fan, I. V. Shadrivov, and W. J. Padilla, "Experimental realization of a terahertz all-dielectric metasurface absorber," *Opt. Express* **25**, 191–201 (2017).
36. K. Fan, J. Y. Suen, X. Liu, and W. J. Padilla, "All-dielectric metasurface absorbers for uncooled terahertz imaging," *Optica* **4**, 601–604 (2017).