ARTICLE OPEN

Active learning of deep surrogates for PDEs: application to metasurface design

Raphaël Pestourie¹✉, Youssef Mroueh^{2,3}, Thanh V. Nguyen⁴, Payel Das^{1,2,3}✉ and Steven G. Johnson¹

Surrogate models for partial differential equations are widely used in the design of metamaterials to rapidly evaluate the behavior of composable components. However, the training cost of accurate surrogates by machine learning can rapidly increase with the number of variables. For photonic-device models, we find that this training becomes especially challenging as design regions grow larger than the optical wavelength. We present an active-learning algorithm that reduces the number of simulations required by more than an order of magnitude for an NN surrogate model of optical-surface components compared to uniform random samples. Results show that the surrogate evaluation is over two orders of magnitude faster than a direct solve, and we demonstrate how this can be exploited to accelerate large-scale engineering optimization.

npj Computational Materials (2020)6:164; <https://doi.org/10.1038/s41524-020-00431-2>

INTRODUCTION

Designing metamaterials or composite materials, in which computational tools select composable components to re-create desired properties that are not present in the constituent materials, is a crucial task for a variety of areas of engineering (acoustic, mechanics, thermal/electronic transport, electromagnetism, and optics)¹. For example, in metalenses, the components are subwavelength scatterers on a surface, but the device diameter is often $>10^3$ wavelengths². Applications of such optical structures include ultra-compact sensors, imaging, and spectroscopy devices used in cell phone cameras and in medical applications². As the metamaterials become larger in scale and as the manufacturing capabilities improve, there is a pressing need for scalable computational design tools.

In this work, surrogate models were used to rapidly evaluate the effect of each metamaterial components during device design³, and machine learning is an attractive technique for such models^{4–7}. However, in order to exploit improvements in nano-manufacturing capabilities, components have an increasing number of design parameters and training the surrogate models (using brute-force numerical simulations) becomes increasingly expensive. The question then becomes: How can we obtain an accurate model from minimal training data? We present an active-learning approach—in which training points are selected based on an error measure—that can reduce the number of training points by more than an order of magnitude for a neural-network (NN) surrogate model of partial differential equations (PDEs). Further, we show how such a surrogate can be exploited to speed up large-scale engineering optimization by $>100\times$. In particular, we apply our approach to the design of optical metasurfaces: large (10^2 – 10^6 wavelengths λ) aperiodic nanopatterned ($\ll \lambda$) structures that perform functions such as compact lensing⁸.

Metasurface design can be performed by breaking the surface into unit cells with a few parameters each (Fig. 1) via domain-decomposition approximations^{3,9}, learning a surrogate model that predicts the transmitted optical field through each unit as a function of an individual cell's parameters, and optimizing the

total field (e.g. the focal intensity) as a function of the parameters of every unit cell³ (see “Results”). This makes metasurfaces an attractive application for machine learning because the surrogate unit-cell model is re-used millions of times during the design process, amortizing the cost of training the model based on expensive exact Maxwell solves sampling many unit-cell parameters. For modeling the effect of 1–4 unit-cell parameters, Chebyshev polynomial interpolation can be very effective³, but encounters an exponential curse of dimensionality with more parameters^{10,11}. In this paper, we find that an NN can be trained with orders of magnitude fewer Maxwell solves for the same accuracy with 10 parameters, even for the most challenging case of multi-layer unit cells many wavelengths ($>10\lambda$) thick. In contrast, we show that subwavelength-diameter design regions (considered by several other authors^{4–7,12,13}) require orders of magnitude fewer training points for the same number of parameters (Fig. 2), corresponding to the physical intuition that wave propagation through subwavelength regions is effectively determined by a few effective-medium parameters¹⁴, making the problems effectively low-dimensional. In contrast to typical machine-learning applications, constructing surrogate models for physical model such as Maxwell's equations corresponds to interpolating smooth functions with no noise, and this requires approaches to training and active learning as described in the “Results” section. We believe that these methods greatly extend the reach of surrogate model for metamaterial optimization and other applications requiring moderate-accuracy high-dimensional smooth interpolation.

Recent work has demonstrated a wide variety of optical-metasurface design problems and algorithms. Different applications¹⁵ such as holograms¹⁶, polarization^{17,18}, wavelength¹⁹, depth-of-field²⁰, or incident angle-dependent functionality²¹ are useful for imaging or spectroscopy^{22,23}. Pestourie et al.³ introduced an optimization approach to metasurface design using the Chebyshev polynomial surrogate model, which was subsequently extended to topology optimization ($\sim 10^3$ parameters per cell) with online Maxwell solvers²⁴. Metasurface modeling can also be composed with signal/image-processing stages for optimized end-to-end design^{25,26}. Previous work demonstrated NN surrogate

¹Department of Mathematics, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA. ²IBM Research AI, IBM Thomas J Watson Research Center, Yorktown Heights, Yorktown, NY 10598, USA. ³MIT-IBM Watson AI Lab, Cambridge, MA 02139, USA. ⁴Iowa State University, Ames, IA 50011, USA. ✉email: rpestour@mit.edu; daspa@us.ibm.com

Fig. 1 Unit cells for metasurface design. 3D unit cells: **a** fin unit cell with two parameters, **b** H-shape unit cell with four parameters. Unit cells (with independent sets of parameters) are juxtaposed to form a metasurface **c** in 3D and **d** in 2D, which is optimized to scatter light in a prescribed way. **e** 2D unit cell: multi-layer unit cell with holes with ten parameters. Each of the unit cell parameters are illustrated by red arrows. The transmitted field of the unit cell is computed with periodic boundary conditions. When the period is subwavelength, the transmitted field can be summarized by a single complex number—the complex transmission. Using the local periodic approximation and the unit cell simulations, we can efficiently compute the approximate source equivalent to the metasurface and generate the field anywhere in the far field.

Fig. 2 Test error with varying design-region diameters. Comparison of baseline training as we shrink the unit cell. **a** For the same number of training points, the fractional error (defined in “Methods”) on the test set of the small unit cell and the smallest unit cell are, respectively, one and two orders of magnitude better than the error of the main unit cell when using 1000 training points or more, which indicates that parameters are more independent when the design-region diameter is big ($\gg \lambda$), and training the surrogate model becomes harder. **b** Pictures of the unit cells to scale. Each color corresponds to the line color in the plot. For clarity, an inset shows the smallest unit cell enlarged 10 times.

models in optics for a few parameters^{27–29}, or with more parameters in deeply subwavelength-design regions^{4,12}. As shown in Fig. 2, deeply subwavelength regions pose a vastly easier problem for NN training than parameters spread over larger diameters. Another approach involves generative design, again typically for subwavelength^{6,7} or wavelength-scale unit cells³⁰, in some cases in conjunction with larger-scale models^{5,12,13}. A generative model is essentially the inverse of a surrogate function: instead of going from geometric parameters to performance, it takes the desired performance as an input and produces the geometric structure, but the mathematical challenge appears to be closely related to that of surrogates.

Fig. 3 Active-learning algorithm and the surrogate model. Diagram of the surrogate model (blue background) and the active-learning algorithm (orange background). The circle arrow signifies that the algorithm iterates T times. The fast evaluation of the surrogate is used both to create predictions of the surrogate model and to compute the error measure that selects the points to add to the training set.

Active learning is connected with the field of uncertainty quantification (UQ), because active learning consists of adding the most uncertain points to training set in an iterative way (Figs. 3 and 4) and hence it requires a measure of uncertainty. Our approach to UQ is based on the NN-ensemble idea of ref. ³¹ due to its simplicity. There are many other approaches for UQ^{32, Sec. 5]}, but ref. ³¹ demonstrated performance and scalability advantages of the NN-ensemble approach. This approach is an instance of Bayesian deep learning³². In contrast, Bayesian optimization relies on Gaussian processes that scale poorly ($\sim N^3$ where N is the number of training samples)^{33,34}. The work presented here achieves training time efficiency (we show an order of magnitude reduction sample complexity), design time efficiency (the actively learned surrogate model is at least two orders of magnitude faster than solving Maxwell’s equations), and realistic large-scale designs (due to our optimization framework³), all in one package.

Input: n_{init}, T, M, K
Result: the surrogate model $\tilde{t}(\mathbf{p})$ (μ_* and σ_*)
 $\mathcal{P}_0 = n_{\text{init}}$ points chosen from a random uniform distribution;
 Solve PDE for each point in $\mathcal{P}_0$; // expensive step
 Create the first iteration of the labeled training set $\mathcal{TS}_0$;
 Train the ensemble $\tilde{t}^0(\mathbf{p})$ on $\mathcal{TS}_0$;
for $i = 1:T$ **do**
    $\mathcal{R}_i = M \times K$ points chosen from a random uniform distribution ;
   Compute the error measures $\sigma_i^{t-1}(\mathbf{p})$ using $\tilde{t}^{i-1}, \forall \mathbf{p} \in \mathcal{R}_i$; // cheap step
    $\mathcal{P}_i$ = select K points in $\mathcal{R}_i$ with the highest error measures σ_i^{t-1} ;
   Solve PDE for each points in $\mathcal{P}_i$ and get $t(\mathbf{p}), \forall \mathbf{p} \in \mathcal{P}_i$; // expensive step
   Augment the labeled training set with new labeled data $\mathcal{TS}_i$;
   Train the ensemble $\tilde{t}^i(\mathbf{p})$ on $\mathcal{TS}_i$ with warm start of $\tilde{t}^{i-1}$;
end

Fig. 4 Algorithm for active learning of the surrogate model. The algorithm takes the input hyperparameters n_{init}, T, M, K , and returns the actively learned surrogate model (which outputs an estimate of the complex transmission coefficient and an error measure). The algorithm adds the training points iteratively, by filtering the randomly sampled points with the highest error measures.

RESULTS

Metasurfaces and surrogate models

In this section, we present the NN surrogate model used in this paper, for which we adopt the metasurface design formulation from ref. ³. The first step of this approach is to divide the metasurface into unit cells with a few geometric parameters $\mathbf{p}$ each. For example, Fig. 1 shows several possible unit cells: (Fig. 1a) a rectangular pillar (fin) etched into a 3d dielectric slab³⁵ (two parameters); (Fig. 1b) an H-shaped unit cell (four parameters) in a dielectric slab⁴; or (Fig. 1e) a multi-layered 2d unit cell with ten holes of varying widths considered in this paper. As depicted in Fig. 1c, d, a metasurface consists of an array of these unit cells. The second step is to solve for the transmitted field (from an incident plane wave) independently for each unit cell using approximate boundary conditions^{3,24,35,36}, in our case a locally periodic approximation (LPA) based on the observation that optimal structures often have parameters that mostly vary slowly from one unit cell to the next (ref. ³ has a detailed section and a figure about this approximation (Sec. 2.1, Fig. 2); other approximate boundary conditions are also possible⁹). For a subwavelength period, the LPA transmitted far field is entirely described by a single number—the complex transmission coefficient $t(\mathbf{p})$. One can then compute the field anywhere above the metasurface by convolving these approximate transmitted fields with a known Green's function—a near-to-far-field transformation³⁷. Finally, any desired function of the transmitted field, such as the focal-point intensity, can be optimized as a function of the geometric parameters of each unit cell³.

In this way, optimizing an optical metasurface is built on top of evaluating the function $t(\mathbf{p})$ (transmission through a single unit cell as a function of its geometric parameters) thousands or even millions of times—once for every unit cell, for every step of the optimization process. Although it is possible to solve Maxwell's equations online during the optimization process, allowing one to use thousands of parameters $\mathbf{p}$ per unit cell requires substantial parallel computing clusters²⁴. Alternatively, one can solve Maxwell's equations offline (before metasurface optimization) in order to fit $t(\mathbf{p})$ to a surrogate model:

$$\tilde{t}(\mathbf{p}) \approx t(\mathbf{p}), \quad (1)$$

which can subsequently be evaluated rapidly during metasurface optimization (perhaps for many different devices). For similar reasons, surrogate (or reduced-order) models are attractive for any design problem involving a composite of many components that can be modeled separately^{6,7,38}. The key challenge of the surrogate approach is to increase the number of design parameters, especially in non-subwavelength regions as discussed in Fig. 2.

In this paper, the surrogate model for each of the real and imaginary parts of the complex transmission is an ensemble of

Fig. 5 Test error for actively learned surrogate, baseline, and Chebyshev interpolation. The lower the desired fractional error, the greater the reduction in training cost compared to the baseline algorithm; the slope of the active-learning fractional error (−0.2) is about 30% steeper than that of the baseline (−0.15). The active-learning algorithm achieves a reasonable fractional error of 0.07 in 12 times less points than the baseline, which corresponds to more than one order of magnitude saving in training data. Chebyshev interpolation (surrogate for blue frequency only) does not compete well with this number of training points. Inset: Unit cell corresponding to the surrogate model.

$J = 5$ independent NNs with the same training data but different random batches³⁹ on each training step. Each of NN i is trained to output a prediction $\mu_i(\mathbf{p})$ and an error estimate $\sigma_i(\mathbf{p})$ for every set of parameters $\mathbf{p}$. To obtain these μ_i and σ_i from training data $y(\mathbf{p})$ (from brute-force offline Maxwell solves) we minimize³¹:

$$-\sum_{\mathbf{p}} \log p_{\Theta_i}(y|\mathbf{p}) \propto \sum_{\mathbf{p}} \left[\log \sigma_i(\mathbf{p}) + \frac{(y(\mathbf{p}) - \mu_i(\mathbf{p}))^2}{2\sigma_i(\mathbf{p})^2} \right] \quad (2)$$

over the parameters Θ_i of NN i . Equation (2) is motivated by problems in which y was sampled from a Gaussian distribution for each $\mathbf{p}$, in which case μ_i and σ_i^2 could be interpreted as mean and hetero-skedastic variance, respectively³¹. Although our exact function $t(\mathbf{p})$ is smooth and noise free, we find that Eq. (2) still works well to estimate the fitting error, as demonstrated in Fig. 5. Each NN is composed of an input layer with 13 nodes (10 nodes for the geometry parameterization— $\mathbf{p} \in [0, 1]^{10}$ —and 3 nodes for the one-hot encoding³⁹ of three frequencies of interest), three fully-connected hidden layers with 256 rectified linear units (ReLU³⁹), and one last layer containing one unit with a scaled hyperbolic-tangent activation function³⁹ (for μ_i) and one unit with a softplus activation function³⁹ (for σ_i). Given this ensemble of J NNs, the final prediction μ_* (for the real or imaginary part of $t(\mathbf{p})$) and its associated error estimate σ_* are combined as³¹

$$\mu_*(\mathbf{p}) = \frac{1}{J} \sum_{i=1}^J \mu_i(\mathbf{p}), \quad (3)$$

$$\sigma_*^2(\mathbf{p}) = \frac{1}{J} \sum_{i=1}^J (\sigma_i^2(\mathbf{p}) + (\mu_i^2(\mathbf{p}) - \mu_*^2(\mathbf{p}))). \quad (4)$$

Subwavelength is easier: effect of diameter

Before performing active learning, we first identify the regime where active learning can be most useful: unit-cell design volumes that are not small compared to the wavelength λ . Previous work on surrogate models^{4–7,12,13} demonstrated NN surrogates (trained with uniform random samples) for unit cells with $\sim 10^2$ parameters. However, these NN models were limited to a regime where the unit-cell degrees of freedom lay within a subwavelength-diameter

volume of the unit cell. To illustrate the effect of shrinking design volume on NN training, we trained our surrogate model for three unit cells (Fig. 2b): the main unit cell of this study is 12.5λ deep, the small unit cell is a vertically scaled-down version of the normal unit cell only 1.5λ deep, and the smallest unit cell is a version of the small unit cell further scaled down (both vertically and horizontally) by $10\times$. Figure 2a shows that, for the same number of training points, the fractional error (defined in “Methods”) on the test set of the small unit cell and the smallest unit cell are, respectively, one and two orders of magnitude better than the error of the main unit cell when using 1000 training points or more. (The surrogate output is the complex transmission $\hat{t}$.) That is, Fig. 2a shows that in the subwavelength-design regime, training the surrogate model is far easier than for larger design regions ($>\lambda$).

Physically, for extremely subwavelength volumes wave propagation is accurately approximated by an averaged effective medium¹⁴, so there are effectively only a few independent design parameters regardless of the number of geometric degrees of freedom. (Effective-medium theory, also called homogenization theory, arises from the fact that extremely subwavelength features affect waves only in an averaged sense, in the same way that light propagating through glass can be described using a refractive index rather than by explicitly modeling scattering from individual atoms.) Quantitatively, we find that the Hessian of the trained surrogate model (second-derivative matrix) in the smallest unit-cell case is dominated by only two singular values—consistent with a function that effectively has only two free parameters—with the other singular values being more than $100\times$ smaller in magnitude; for the other two cases, many more training points would be required to accurately resolve the smallest Hessian singular values. A unit cell with large design-volume diameter ($>>\lambda$) is much harder to train, because the dimensionality of the design parameters is effectively much larger.

Active-learning algorithm

Here, we present an online algorithm to choose training points that is significantly better at reducing the error than choosing points from a random uniform distribution. As described below, we select the training points where the estimated model error is largest, given the estimated error σ_* .

The online algorithm used to train each of the real and imaginary parts is outlined in Figs. 3 and 4. Initially we choose n_{init} uniformly distributed random points $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n_{\text{init}}}$ to train a first iteration $\hat{t}^0(\mathbf{p})$ over 50 epochs³⁹. Then, given the model at iteration i , we evaluate $\hat{t}^i(\mathbf{p})$ (which is orders of magnitude faster than the Maxwell solver) at $M \times K$ points sampled uniformly at random and choose the K points that correspond to the largest σ_*^2 . We perform the expensive Maxwell solves only for these K points, and add the newly labeled data to the training set. We train $\hat{t}^{i+1}(\mathbf{p})$ with the newly expended training set, using $\hat{t}^i$ as a warm start. We repeat this process T times.

Essentially, the method works because the error estimate σ_* is updated every time the model is retrained with an expended dataset. In this way, the model tells us where it does poorly by setting a large σ_* for parameters $\mathbf{p}$ where the estimation would be bad in order to minimize Eq. (2).

Active-learning results

Our algorithm achieves an order-of-magnitude reduction in training data.

We compared the fractional errors of a NN surrogate model trained using uniform random samples with an identical NN trained using an active-learning approach, in both cases modeling the complex transmission of a multi-layer unit cell with ten independent parameters (Fig. 5, inset). With the notation of our algorithm in Fig. 4, the baseline corresponds to $T = 0$, and n_{init} equal to the total

number of training points. This corresponds to no active learning at all, because the n_{init} points are chosen from a random uniform distribution. In the case of active learning, $n_{\text{init}} = 2000$, $M = 4$, and we computed for $K = 500, 1000, 2000, 4000, 8000, 16,000, 32,000, 64,000$, and $128,000$. Although three orders of magnitude on the log-log plot is too small to determine if the apparent linearity indicates a power law, Fig. 5 shows that the lower the desired fractional error, the greater the reduction in training cost compared to the baseline algorithm; the slope of the active-learning fractional error (-0.2) is about 30% steeper than that of the baseline (-0.15). The active-learning algorithm achieves a reasonable fractional error of 0.07 in 12 times less points than the baseline, which corresponds to more than one order of magnitude saving in training data (much less expensive Maxwell solves). This advantage would presumably increase for a lower error tolerance, though computational costs prohibited us from collecting orders of magnitude more training data to explore this in detail. For comparison and completeness, Fig. 5 shows fractional errors using Chebyshev interpolation (for the blue frequency only). Chebyshev interpolation has a much worse fractional error for a similar number of training points. Chebyshev interpolation suffers from the curse of dimensionality—the number of training points is exponential with the number of variables. The two fractional errors shown are for three and four interpolation points in each of the dimensions, respectively. In contrast, NNs are known to mitigate the curse of dimensionality⁴⁰.

Application to metamaterial design: we used both surrogates models to design a multiplexer—an optical device that focuses different wavelength at different points in space. The actively learned surrogate model results in a design that much more closely matches a numerical validation than the baseline surrogate (Fig. 6). As explained in the “Results” section, we replace a

Fig. 6 Application to metamaterial design. **a, b** We used **a** the actively learned and **b** the baseline surrogates models to design a multiplexer—an optical device that focuses different wavelength at different points in space. The actively learned surrogate model results in a design that much more closely matches a numerical validation than the baseline surrogate. This shows that the active-learning surrogate is better at driving the optimization away from regions of inaccuracy. **c** Resulting metastructure for the active-learning surrogate with 100 unit cells of 10 independent parameters each (one parameter per layer).

Maxwell's equations solver with a surrogate model to rapidly compute the optical transmission through each unit cell; a similar surrogate approach could be used for optimizing many other complex physical systems. In the case of our two-dimensional unit cell, the surrogate model is two orders of magnitude faster than solving Maxwell's equations with a finite difference frequency domain (FDFD) solver⁴¹. The speed advantage of a surrogate model becomes drastically greater in three dimensions, where PDE solvers are much more costly while the surrogate model remains the same.

The surrogate model is evaluated millions of times during a metastructure optimization. We used the actively learned surrogate model and the baseline surrogate model (uniform random training samples), in both cases with 514,000 training points, and we optimized a ten-layer metastructure with 100 unit cells of period 400 nm for a multiplexer application—where three wavelengths (blue: 405 nm, green: 540 nm, and red: 810 nm) are focused on three different focal spots (−10 μm, 60 μm), (0, 60 μm), and (+10 μm, 60 μm), respectively. The diameter is 40 μm and the focal length is 60 μm, which corresponds to a numerical aperture of 0.3. Our optimization scheme tends to yield results robust to manufacturing errors³ for two reasons: first, we optimize for the worst case of the three focal spot intensities, using an epigraph formulation³; second, we compute the average intensity from an ensemble of surrogate models that can be thought of as a Gaussian distribution $\tilde{t}(\mathbf{p}) = \mu_*(\mathbf{p}) + \sigma_*(\mathbf{p})\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$, and μ_* and σ_* are defined in Eq. (3) and Eq. (4), respectively,

$$\mathbb{E}|E(\mathbf{r})|^2 = \left| \int G\mu_* \right|^2 + \left| \int G\sigma_* \right|^2, \quad (5)$$

where G is a Green's function that generates the far field from the sources of the metastructure³. The resulting optimized structure for the active-learning surrogate is shown in Fig. 6c.

In order to compare the surrogate models, we validate the designs by computing the optimal unit cell fields directly using a Maxwell solver instead of using the surrogate model. This is computationally easy because it only needs to be done once for each of the 100 unit cells instead of millions of times during the optimization. The focal lines—the field intensity along a line parallel to the two-dimensional metastructure and passing through the focal spots—resulting from the validation are exact solutions to Maxwell's equations assuming the LPA (see “Results” section). Figure 6a, b shows the resulting focal lines for the active-learning and baseline surrogate models. A multiplexer application requires similar peak intensity for each of the focal spots, which is achieved using worst-case optimization³. Figure 6a, b shows that the actively learned surrogate has $\approx 3\times$ smaller error in the focal intensity compared to the baseline surrogate model. This result shows that not only is the active-learning surrogate more accurate than the baseline surrogate for 514,000 training points but also the results are more robust using the active-learning surrogate—the optimization does not drive the parameters towards regions of high inaccuracy of the surrogate model. Note that we limited the design to a small overall diameter (100 unit cells) mainly to ease visualization (Fig. 6c), and we find that this design can already yield good focusing performance despite the small diameter. In earlier work, we have already demonstrated that our optimization framework is scalable to designs that are orders of magnitudes larger⁴². In principle, a manufacturing uncertainty measure could also be incorporated into the metasurface design process via robust optimization algorithms⁴³, but in practice metasurface designs are already typically robust enough to manufacture, especially since multi-wavelength optimization is already a form of robustness³. Then robustness is robustness to any kind of error (including that of ML).

Previous work, such as ref. ⁴⁴—in a different approach to active-learning that does not quantify uncertainty—suggested iteratively

adding the optimum design points to the training set (re-optimizing before each new set of training points is added). However, we did not find this approach to be beneficial in our case. In particular, we tried adding the data generated from LPA validations of the optimal design parameters, in addition to the points selected by our active learning algorithm, at each training iteration, but we found that this actually destabilized the learning and resulted in designs qualitatively worse than the baseline. By exploiting validation points, it seems that the active learning of the surrogate tends to explore less of the landscape of the complex transmission function, and hence leads to poorer designs. Such exploitation–exploration trade-offs are known in the active-learning literature⁴⁵.

DISCUSSION

In this paper, we present an active-learning algorithm for composite materials which reduces the training time of the surrogate model for a physical response by at least one order of magnitude. The simulation time is reduced by at least two orders of magnitude using the surrogate model compared to solving the PDEs numerically. While the domain-decomposition method used here is the LPA and the PDEs are the Maxwell equations, the proposed approach is directly applicable to other domain-decomposition methods (e.g. overlapping domain approximation⁹) and other PDEs or ordinary differential equations⁴⁶.

We used an ensemble of NNs for interpolation in a regime that is seldom considered in the machine-learning literature—machine-learning models are mostly trained from noisy measurements, whereas here the data are obtained from smooth functions. In this regime, it would be instructive to have a deeper understanding of the relationship between NNs and traditional approximation theory (e.g. with polynomials and rational functions^{10,11}). For example, the likelihood maximization of our method forces σ_* to go to zero when $\tilde{t}(\mathbf{p}) = t(\mathbf{p})$. Although this allows us to simultaneously obtain a prediction μ_* and an error estimate σ_* , there is a drawback. In the interpolation regime (when the surrogate is fully determined), σ_* would become identically zero even if the surrogate does not match the exact model away from the training points. In contrast, interpolation methods such as Chebyshev polynomials yield a meaningful measure of the interpolation error even for exact interpolation of the training data^{10,11}. In the future, we plan to separate the estimation model and the model for the error measure using a meta-learner architecture⁴⁷, with expectation that the meta-learner will produce a more accurate error measure and further improve training time. We will also explore other ensembling methods that could improve the accuracy of our model^{48,49}. We believe that the method presented in this paper will greatly extend the reach of surrogate-model-based optimization of composite materials and other applications requiring moderate-accuracy high-dimensional interpolation.

METHODS

Training-data computation

The complex transmission coefficients were computed in parallel using an open-source FDFD solver for Helmholtz equation⁵⁰ on a 3.5 GHz 6-Core Intel Xeon E5 processor. The material properties of the multi-layered unit cells are silica (refractive index of 1.45) in the substrate, and air (refractive index of 1) in the hole and in the background. In the normal unit cell, the period of the cell is 400 nm, the height of the ten holes is fixed to 304 nm, and their widths vary between 60 and 340 nm, each hole is separated by 140 nm of substrate. In the small unit cell, the period of the cell is 400 nm, the height of the ten holes is 61 nm, and their widths vary between 60 and 340 nm, there is no separation between the holes. The smallest unit cell is the same as the small unit cell shrunk ten times (period of 40 nm, ten holes of height 6.1 nm, and width varying between 6 and 34 nm).

Metalens design problem

The complex transmission data are used to compute the scattered field off a multi-layered metastructure with 100 unit cells as in ref. ³. The metastructure was designed to focus three wavelengths (blue: 405 nm, green: 540 nm, and red: 810 nm) on three different focal spots (−10 μm, 60 μm), (0, 60 μm), and (+10 μm, 60 μm), respectively. The epigraph formulation of the worst-case optimization and the derivation of the adjoint method to get the gradient are detailed in ref. ³. Any gradient based optimization algorithm would work, but we used an algorithm based on conservative convex separable approximations⁵¹. The average intensity is derived from the distribution of the surrogate model $\tilde{t}(\mathbf{p}) = \mu_*(\mathbf{p}) + \sigma_*(\mathbf{p})\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$ and the computation of the intensity based on the local field as in ref. ³,

$$|E(\mathbf{r})|^2 = \left| \int_{\Sigma} G(\mathbf{r}, \mathbf{r}') (-\tilde{t}(\mathbf{p}(\mathbf{r}')) d\mathbf{r}' \right|^2, \quad (6)$$

$$= \int_{\Sigma} \bar{G}(\bar{\mu}_*(\mathbf{p}) + \bar{\sigma}_*(\mathbf{p})\epsilon) d\mathbf{r}' \int_{\Sigma} G(\mu_*(\mathbf{p}) + \sigma_*(\mathbf{p})\epsilon) d\mathbf{r}', \quad (7)$$

$$= \int \bar{G}\bar{\mu}_* \int G\mu_* + \epsilon^2 \int \bar{G}\bar{\sigma}_* \int G\sigma_* + 2\epsilon \text{Re} \left(\int \bar{G}\bar{\mu}_* \int G\sigma_* \right), \quad (8)$$

$$= \left| \int G\mu_* \right|^2 + \epsilon^2 \left| \int G\sigma_* \right|^2 + 2\epsilon \text{Re} \left(\int \bar{G}\bar{\mu}_* \int G\sigma_* \right), \quad (9)$$

where the $\bar{(\cdot)}$ notation denotes the complex conjugate, the notations $\int_{\Sigma}(\cdot) d\mathbf{r}'$ and $G(\mathbf{r}, \mathbf{r}')$ are simplified to $\int$ and G , and the notation $\mathbf{p}(\mathbf{r}')$ is dropped for concision. From the linearity of expectation,

$$\mathbb{E}|E(\mathbf{r})|^2 = \left| \int G\mu_* \right|^2 + \mathbb{E}(\epsilon^2) \left| \int G\sigma_* \right|^2 + 2\mathbb{E}(\epsilon) \text{Re} \left(\int \bar{G}\bar{\mu}_* \int G\sigma_* \right), \quad (10)$$

$$\mathbb{E}|E(\mathbf{r})|^2 = \left| \int G\mu_* \right|^2 + \left| \int G\sigma_* \right|^2, \quad (11)$$

where we used that $\mathbb{E}(\epsilon) = 0$ and $\mathbb{E}(\epsilon^2) = 1$.

Active-learning architecture and training

The ensemble of NN was implemented using PyTorch⁵² on a 3.5 GHz 6-Core Intel Xeon E5 processor. We trained an ensemble of 5 NNs for each surrogate models. Each NN is composed of an input layer with 13 nodes (10 nodes for the geometry parameterization and 3 nodes for the one-hot encoding³⁹ of three frequencies of interest), three fully-connected hidden layers with 256 ReLU³⁹, and one last layer containing one unit with a scaled hyperbolic-tangent activation function³⁹ (for μ_i) and one unit with a softplus activation function³⁹ (for σ_i). The cost function is negative-log-likelihood of a Gaussian as in Eq. (2). The mean and the variance of the ensemble are the pooled mean and variance from Eqs. (3) and (4). The optimizer is Adam⁵³. The parameters are initialized using PyTorch's default settings, i.e., sampled uniformly on a support inversely proportional to the square root of the number of input parameters. The starting learning rate is 0.001. After the tenth epoch, the learning rate is decayed by a factor of 0.99. Each iteration of the active-learning algorithm as well as the baseline were trained for 50 epochs. The choice of training points is detailed in the algorithm of Fig. 4. The quantitative evaluations were computed using the fractional error on a test set containing 2000 points chosen from a random uniform distribution. The fractional error FE between two vectors of complex values $\mathbf{u}_{\text{estimate}}$ and $\mathbf{v}_{\text{true}}$ is

$$\text{FE} = \frac{\|\mathbf{u}_{\text{estimate}} - \mathbf{v}_{\text{true}}\|}{\|\mathbf{v}_{\text{true}}\|}, \quad (12)$$

where $\|\cdot\|$ is the L2-norm for complex vectors.

For 128k training points and the surrogate NN architecture mentioned in this section, the time complexity breakdown of active learning is 5.4k seconds (including training and evaluation), and 27.8k seconds for Maxwell's simulations on a 3.5 GHz 6-Core Intel Xeon E5 processor. Maxwell's equations simulations are the most expensive part of the active learning process and account for 85% of the total computation time.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CODE AVAILABILITY

The code used for these findings is available upon reasonable request.

Received: 29 July 2020; Accepted: 23 September 2020;

Published online: 29 October 2020

REFERENCES

- Kadic, M., Milton, G. W., van Hecke, M. & Wegener, M. 3D metamaterials. *Nat. Rev. Phys.* **1**, 198–210 (2019).
- Khorasaninejad, M. & Capasso, F. Metalenses: versatile multifunctional photonic components. *Science* **358**, eaam8100 (2017).
- Pestourie, R. et al. Inverse design of large-area metasurfaces. *Opt. Express* **26**, 33732–33747 (2018).
- An, S. et al. A deep learning approach for objective-driven all-dielectric meta-surface design. *ACS Photonics* **6**, 3196–3207 (2019).
- Jiang, J. & Fan, J. A. Simulator-based training of generative neural networks for the inverse design of metasurfaces. *Nanophotonics* **9**, 1059–1069 (2020).
- An, S. et al. Generative multi-functional meta-atom and metasurface design networks. Preprint at <https://arxiv.org/abs/1908.04851> (2019).
- Jiang, J. et al. Free-form diffractive metagrating design based on generative adversarial networks. *ACS Nano* **13**, 8872–8878 (2019).
- Yu, N. et al. Light propagation with phase discontinuities: generalized laws of reflection and refraction. *Science* **334**, 333–337 (2011).
- Lin, Z. & Johnson, S. G. Overlapping domains for topology optimization of large-area metasurfaces. *Opt. Express* **27**, 32445–32453 (2019).
- Boyd, J. P. *Chebyshev and Fourier Spectral Methods*, 2nd edn. (Dover Publications, Inc., Mineola, 2001).
- Trefethen, L. N. *Approximation Theory and Approximation Practice* extended, edn. (SIAM, Philadelphia, 2019).
- Jiang, J., Chen, M. & Fan, J. A. Deep neural networks for the evaluation and design of photonic devices. Preprint at <https://arxiv.org/abs/2007.00084> (2020).
- Ma, W., Cheng, F. & Liu, Y. Deep-learning-enabled on-demand design of chiral metamaterials. *ACS Nano* **12**, 6326–6334 (2018).
- Holloway, C. L., Kuester, E. F. & Dienstfrey, A. Characterizing metasurfaces/metamaterials: the connection between surface susceptibilities and effective material properties. *IEEE Antennas Wirel. Propag. Lett.* **10**, 1507–1511 (2011).
- Maguid, E. et al. Photonic spin-controlled multifunctional shared-aperture antenna array. *Science* **352**, 1202–1206 (2016).
- Sung, J., Lee, G.-Y., Choi, C., Hong, J. & Lee, B. Single-layer bifacial metasurface: Full-space visible light control. *Adv. Opt. Mater.* **7**, 1801748 (2019).
- Arbabi, A., Horie, Y., Bagheri, M. & Faraon, A. Dielectric metasurfaces for complete control of phase and polarization with subwavelength spatial resolution and high transmission. *Nat. Nanotechnol.* **10**, 937–943 (2015).
- Mueller, J. B., Rubin, N. A., Devlin, R. C., Groever, B. & Capasso, F. Metasurface polarization optics: independent phase control of arbitrary orthogonal states of polarization. *Phys. Rev. Lett.* **118**, 113901 (2017).
- Ye, W. et al. Spin and wavelength multiplexed nonlinear metasurface holography. *Nat. Commun.* **7**, 11930 (2016).
- Bayati, E. et al. Inverse designed metalenses with extended depth of focus. *ACS Photonics* **7**, 873–878 (2020).
- Liu, W. et al. Metasurface enabled wide-angle Fourier lens. *Adv. Mater.* **30**, 1706368 (2018).
- Aieta, F., Kats, M. A., Genevet, P. & Capasso, F. Multiwavelength achromatic metasurfaces by dispersive phase compensation. *Science* **347**, 1342–1345 (2015).
- Zhou, Y. et al. Multilayer noninteracting dielectric metasurfaces for multi-wavelength metaoptics. *Nano Lett.* **18**, 7529–7537 (2018).
- Lin, Z., Liu, V., Pestourie, R. & Johnson, S. G. Topology optimization of freeform large-area metasurfaces. *Opt. Express* **27**, 15765–15775 (2019).
- Sitzmann, V. et al. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Trans. Graph.* **37**, 114 (2018).
- Lin, Z. et al. End-to-end inverse design for inverse scattering via freeform metastructures. Preprint at <https://arxiv.org/abs/2006.09145> (2020).
- Liu, D., Tan, Y., Khoram, E. & Yu, Z. Training deep neural networks for the inverse design of nanophotonic structures. *ACS Photonics* **5**, 1365–1369 (2018).
- Malkiel, I. et al. Plasmonic nanostructure design and characterization via deep learning. *Light Sci. Appl.* **7**, 1–8 (2018).
- Peurifoy, J. et al. Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci. Adv.* **4**, eaar4206 (2018).
- Liu, Z., Zhu, Z. & Cai, W. Topological encoding method for data-driven photonics inverse design. *Opt. Express* **28**, 4825–4835 (2020).

31. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. 31st International Conference on Advances in Neural Information Processing Systems*, 6402–6413 (NIPS, 2017).
32. Tagasovska, N. & Lopez-Paz, D. Single-model uncertainties for deep learning. In *Proc. 33rd International Conference on Advances in Neural Information Processing Systems*, 6417–6428 (NIPS, 2019).
33. Lookman, T., Balachandran, P. V., Xue, D. & Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *NPJ Comput. Mater.* **5**, 1–17 (2019).
34. Bassman, L. et al. Active learning for accelerated design of layered materials. *NPJ Comput. Mater.* **4**, 1–9 (2018).
35. Khorasaninejad, M. et al. Metalenses at visible wavelengths: diffraction-limited focusing and subwavelength resolution imaging. *Science* **352**, 1190–1194 (2016).
36. Yu, N. & Capasso, F. Flat optics with designer metasurfaces. *Nat. Mater.* **13**, 139–150 (2014).
37. Harrington, R. F. *Time-Harmonic Electromagnetic Fields*, 2nd edn. (Wiley-IEEE Press, New York, 2001).
38. Mignolet, M. P., Przekop, A., Rizzi, S. A. & Spottswood, S. M. A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures. *J. Sound Vib.* **332**, 2437–2460 (2013).
39. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, Cambridge, 2016).
40. Cheridito, P., Jentzen, A. & Rossmannek, F. Efficient approximation of high-dimensional functions with deep neural networks. Preprint at <https://arxiv.org/abs/1912.04310> (2019).
41. Champagne II, N. J., Berryman, J. G. & Buettner, H. M. FDFD: a 3D finite-difference frequency-domain code for electromagnetic induction tomography. *J. Comput. Phys.* **170**, 830–848 (2001).
42. Pestourie, R. *Assume Your Neighbor is Your Equal: Inverse Design in Nanophotonics*. Ph.D. thesis, Harvard University (2020).
43. Molesky, S. et al. Inverse design in nanophotonics. *Nat Photonics* **12**, 659–670 (2018).
44. Chen, C.-T. & Gu, G. X. Generative deep neural networks for inverse materials design using backpropagation and active learning. *Adv. Sci.* **7**, 1902607 (2020).
45. Hüllermeier, E. & Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: a tutorial introduction. Preprint at <https://arxiv.org/abs/1910.09457> (2019).
46. Rackauckas, C. et al. Universal differential equations for scientific machine learning. Preprint at <https://arxiv.org/abs/2001.04385> (2020).
47. Chen, T., Navrátil, J., Iyengar, V. & Shanmugam, K. Confidence scoring using whitebox meta-models with linear classifier probes. In *The 22nd Int. Conf. on Artificial Intelligence and Statistics*, 1467–1475 (AISTATS, 2019).
48. Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P. & Wilson, A. G. A simple baseline for Bayesian uncertainty in deep learning. In *Proc. 33rd Int. Conf. Advances in Neural Information Processing Systems*, 13153–13164 (NIPS, 2019).
49. Wilson, A. G. & Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. Preprint at <https://arxiv.org/abs/2002.08791> (2020).
50. Pestourie, R. *FDFD Local Field*. https://github.com/rpestourie/fdfd_local_field (2020).
51. Svanberg, K. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J. Optim.* **12**, 555–573 (2002).
52. Paszke, A. et al. Automatic differentiation in PyTorch. In *Proc. 31st International Conference on Advances in Neural Information Processing Systems Workshop on Automatic Differentiation* (NIPS, 2017).
53. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).

ACKNOWLEDGEMENTS

This work was supported in part by IBM Research, the MIT-IBM Watson AI Laboratory, the U.S. Army Research Office through the Institute for Soldier Nanotechnologies (under award W911NF-13-D-0001), and by the PAPP program of DARPA MTO (under award HR0011-20-90016).

AUTHOR CONTRIBUTIONS

R.P., Y.M., P.D., and S.G.J. designed the study, contributed to the machine-learning approach, and analyzed results; R.P. led the code development, software implementation, and numerical experiments; R.P. and S.G.J. were responsible for the physical ideas and interpretation; T.V.N. assisted in designing and implementing the training. All authors contributed to the algorithmic ideas and writing.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to R.P. or P.D.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020