APL Photonics

RESEARCH ARTICLE | OCTOBER 13 2025

Neural reparameterization for nonlocal metasurface topology optimization

Special Collection: Optical Computing Systems

Lincoln Clark ; Ann Roberts ; Lukas Wesemann

APL Photonics 10, 100803 (2025)
https://doi.org/10.1063/5.0283122

Neural reparameterization for nonlocal metasurface topology optimization

Cite as: APL Photon. 10, 100803 (2025); doi: 10.1063/5.0283122
Submitted: 29 May 2025 • Accepted: 26 September 2025 •
Published Online: 13 October 2025

Lincoln Clark,[a] Ann Roberts, and Lukas Wesemann

AFFILIATIONS

School of Physics, University of Melbourne, ARC Centre of Excellence for Transformative Meta-optical Systems, Victoria 3010, Australia

Note: This paper is part of the Special Topic on Optical Computing Systems.
[a]**Author to whom correspondence should be addressed:** clarklj@student.unimelb.edu.au

ABSTRACT

Metasurfaces are emerging as a paradigm for performing all-optical, energy efficient analog computing and image processing in an ultra-compact form-factor. While various operations have been demonstrated with simple metasurface geometries, broadening the range of available processing outputs will require non-trivial and non-intuitive designs. Topology optimization (TO) is a powerful inverse design technique capable of designing such metasurfaces with significantly enhanced optical properties. The two most important design considerations are the quality of the optical performance and the minimum feature size in the resulting device. Larger features can reduce the mismatch between design and realized device performance, increasing fabrication speed and extending the scalability of the fabrication. Metasurface TO is typically performed on a pixel grid to determine the material distribution at each point. Here, we compare two pixel-based methods to neural-network based parameterizations, where the parameters of a neural-network are trained to output a metasurface design. We produce 150 different computational metasurface design tasks to compare the performance of each parameterization with regard to the performance and feature sizes. In comparison to the best pixel-based approach, a hybrid neural-network and pixel method produced designs with a median minimum feature size 2.8 times larger with only a 1.4 times increase in RMS error. Compared to a pixel method capable of producing similar critical dimensions, the optical performance of the hybrid approach is improved by 39%. Finally, as a proof of concept, we use this neural-network based parameterization to design an angular bandpass filter with relatively large critical dimensions and non-trivial performance.

I. INTRODUCTION

Optical metasurfaces are gaining significant interest as ultra-compact, engineered structures that control light through interaction with precisely designed nanostructured surfaces.[1] In contrast to traditional optics, which rely on diffraction and refraction in bulk-optical components, metasurfaces manipulate light through tailored phase, amplitude, and polarization responses of two-dimensional arrangements of resonators and thin films. Metasurfaces have enabled a broad range of ultra-compact optical components, including flat lenses[2] and beam steering devices.[3] More recently, their application in all-optical image processors,[4,5] including hardware accelerators for machine vision applications,[6–8] has attracting increasing interest.

Early optical metasurface designs typically consisted of arrays of simple resonators, such as discs or rods, where tuning a limited set of geometric parameters allowed for basic control over their optical response.[9] While it enabled the implementation of devices with unprecedented optical properties, this design approach encounters significant challenges when aiming to achieve highly specific and complex optical responses.[10]

Topology optimization (TO) is a powerful tool that originated in the field of structural mechanics and has been successfully applied to the inverse design of metasurfaces with complex optical responses, enhanced performance, and robustness to fabrication errors.[11–13] Unlike traditional parameter-based tuning, TO explores a large design space by iteratively refining material distributions to achieve a desired optical response. This approach has, for example,

enabled advances in high efficiency flat lenses,[11] multifunctional metasurfaces,[14] plasmonic optical tweezers,[15] optical computing,[16] image processing,[17] and nonlinear metasurfaces.[18]

The primary constraint on practical metasurface design arises from the limitations of the fabrication techniques used to produce them, such as electron beam lithography (EBL), which is typically used for high-resolution prototyping.[19] For this reason, the minimum feature size of structures, voids, and gaps in the design is a key indicator of the feasibility of a device realized via TO. Reducing the resolution constraints opens up a range of more scalable fabrication methods. Although EBL and scanning probe lithography can produce sub-10 nm features,[20] they are not suitable for mass fabrication.[21] Scalable methods that can produce large-area metasurfaces, such as nanoimprint lithography (critical dimension of 25 nm) or DUV lithography (critical dimension of 65 nm), have been used to mass-produce metasurfaces.[22] The feature sizes that can be produced are not limited entirely by the lithography process; liftoff and etching processes may also limit the minimum feature sizes. Furthermore, larger features are generally easier to fabricate, as they are further from the fundamental resolution limits of the lithography process used, improving the quality of fabricated devices, leading to reduced artifacts and improved performance.

In conventional density based TO for metasurface design, the pattern that defines the material distribution is parameterized into

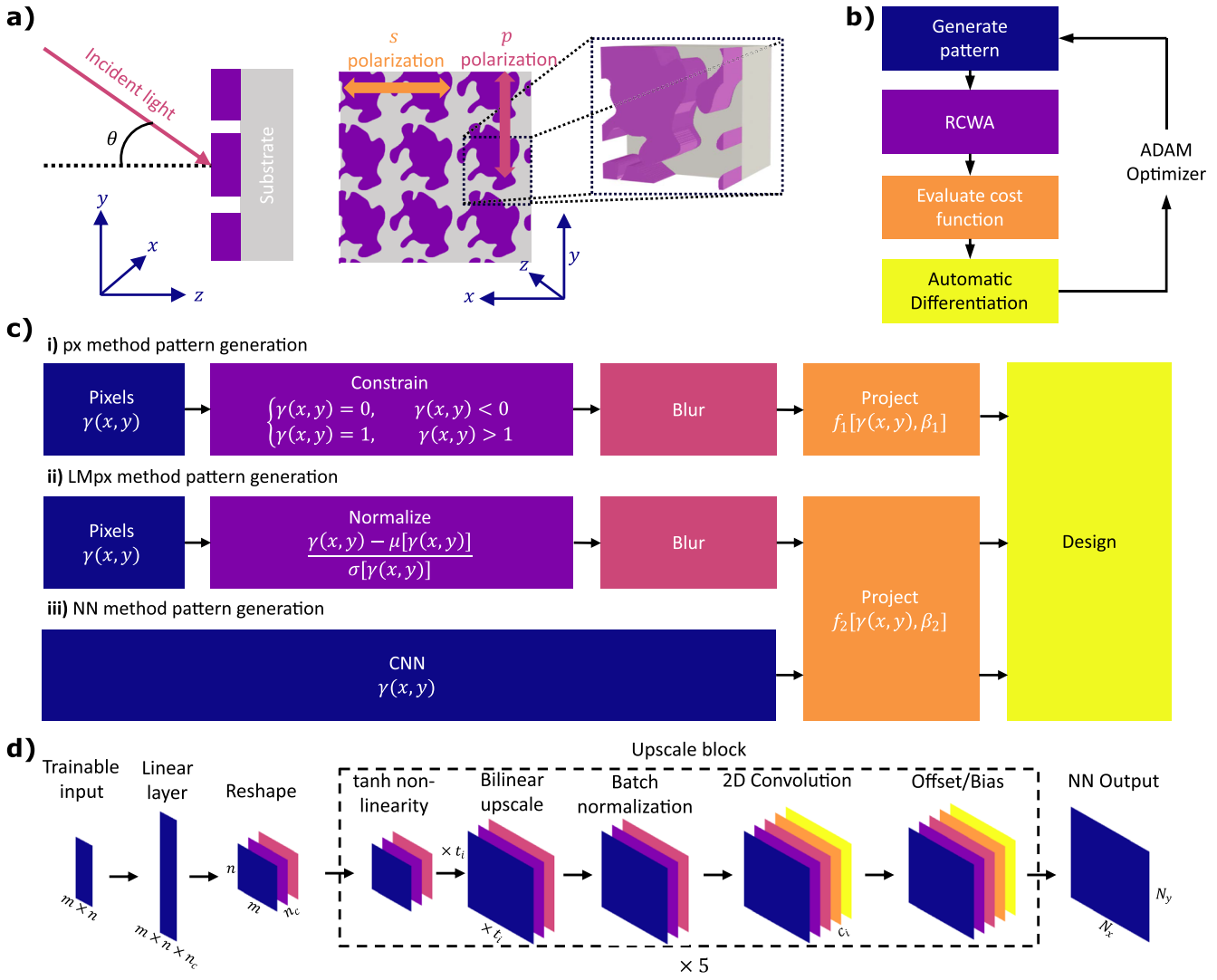

FIG. 1. Schematic of optimization process and parameterization approaches. (a) Schematic of a single unit cell in the optimization process, showing how the axes, angle of incidence, and polarizations are defined. (b) Flow chart of the optimization algorithm. (c) Pattern generation process for (i) the *px* method, (ii) the *LMpx* method, and (iii) the *NN* method. (d) Neural network architecture.

a pixel grid on which the algorithm directly performs the optimization.[23] However, it has been shown that reparameterizing topology optimization problems using the weights and biases of neural networks can lead to better designs.[24] As of yet, this variation of topology optimization has not been applied to metasurface design.

An important finding, originally demonstrated in Ref. 25, is that the inherent assumptions and constraints of an image generator implied by its neural network architecture, the "deep image prior," make a significant contribution to the quality of the solutions it produces prior to any training on data. Building on this, a paper by Google Research investigated how the deep image prior can be leveraged for applications beyond natural images to produce better designs in topology optimization.[26] For optimization problems in structural mechanics, they found that replacing the typical pixel parameterization with a convolutional neural network (CNN) produced designs with superior performance and simpler geometries. Instead of optimizing the design density on a pixel-by-pixel basis, the weights and biases of a CNN are optimized to produce the device design.[24] The impact of the choice of neural network architecture used for reparameterization on the path of the optimizer through the parameter landscape and the final solution in neural-TO is still an active area of research. Sanu *et al.*[27] have recently investigated this quantitatively using problems in structural mechanics, comparing the CNN architecture to feedforward neural networks. Here, we focus on the design of nonlocal periodic metasurfaces for optical computation, where the angular spectrum of an incident field is directly manipulated. Although other neural network architectures are possible, the category of metasurfaces on which we focus consists of an array of repeating unit cells, which require periodicity and continuity of the design across the boundaries. This makes CNNs with periodic boundary conditions a natural choice.

The CNN is used exclusively as a parameterization tool in TO, with no ability to extrapolate to other optimization problems. This approach is fundamentally different from other neural-network based metasurface inverse design methods, where the neural-network is trained to replace the simulation step and extrapolate to other situations, enabling ultra-fast metasurface designs.[28–30] Training these neural networks to predict the electromagnetic response of a metasurface requires a large computational investment to generate the training and validation data needed to learn the laws of physics. Neural networks and physics informed neural networks (PINNs) typically use geometric parameterizations of a metasurface, for example, rectangular or cylindrical meta-atoms. These parameters are fed into the model, and it calculates the optical response.[31,32] In PINN based methods, differential equations such as Maxwell's equations are used in combination with neural networks in order to reduce the amount of training data required to develop a model.[33] Neural networks can also be used to improve the accuracy of simple equivalent circuit models of metasurfaces.[34] Compared with machine learning optimization approaches, TO uses a fully fledged electromagnetic solver; there is no upfront investment to train a neural-network, and it gives more degrees of freedom to design complex free-form metasurfaces.[35] It can be flexibly applied to different optical problems with only minor changes. If the large degrees of freedom that TO offers are not required, then training a neural network to calculate the optical response of a meta-atom platform may be worth the increased computational cost. Rather than replacing the physics simulation step with neural networks, a study by Jiang *et al.*[36] shows that generative adversarial networks can be trained to generate free-form metasurfaces that diffract light into the +1 order highly efficiently. Combined with a TO refinement step, they can rapidly produce high performing metasurface designs; however, this approach requires generating a training set of topology-optimized metasurfaces and is inflexible to changing the optical problem being solved.

Here, we compare the CNN reparameterization of TO to pixel based approaches for the design of periodic metasurfaces. To the best of our knowledge, we believe that this has not been reported previously. In particular, we focus on designing metasurfaces to perform analog optical computing operations. We evaluate the designs produced concerning two key performance indicators: the size of the smallest feature in the design, i.e., the "minimum feature size," and how well the design performs the specific task, as measured by a cost function. We focus on optimization problems commonly encountered in metasurface design. The performance of each of the approaches investigated is evaluated on 150 benchmark problems where we optimize the transmission response of metasurfaces $T(\theta, \lambda = \lambda_0)$ as a function of incident angle θ at a fixed wavelength λ_0 to match a target function T_t. Engineering the angular response of a metasurface is directly related to designing metasurfaces that perform specific spatial computing operations on images, such as differentiation, edge detection, and denoising.[4] Finally, we support our findings by designing an angular bandpass filter, a device that transmits plane waves incident over a limited range of angles, using the neural-TO method.

II. METHODS

Metasurfaces can be designed using aperiodic patterns or via periodic repetition of an identical unit cell. An example of a single unit cell is given in Fig. 1(a), where the full metasurface is a two dimensional array constructed from this building block. By tailoring the geometry of the unit cell in the array, we can control how light interacts with the device as a whole. To design a metasurface, we thus need to determine the distribution of material in the unit cell that gives us the desired optical response. How well a particular metasurface geometry achieves the desired response can be quantified by a cost function. Optimization algorithms can then be used to refine the metasurface design, i.e., the distribution of material in the unit cell, in order to minimize this cost function.[37] The distribution of material in a single unit cell can be described by a material density, $\gamma(x, y)$, where $\gamma = 0$ corresponds to air and $\gamma = 1$ corresponds to the presence of material, silicon in this case. Values of γ satisfying $0 < \gamma < 1$ smoothly interpolate between the permittivity of the two materials used in the optimization. TO deals with determining the specific material density that minimizes the cost function, with minimal constraints on the geometry. An overview of the TO optimization algorithm is given in Fig. 1(b). A candidate unit cell pattern is generated, and the optical properties are simulated using rigorous coupled wave analysis (RCWA). Once we know the optical properties of the candidate pattern, the cost function is evaluated, and we use a gradient based optimization algorithm to successively update the pattern to converge toward a final design. The key to this approach is the use of an implementation of RCWA that uses the PyTorch library. This makes the method differentiable and, therefore, permits us to differentiate the cost function with regard to the

material density. Here, we explore different methods for generating the material density γ in metasurface TO.

A. Spatial filtering

The optical transfer function (OTF) of an optical system is a complex valued function that describes how the amplitude and phase of the different plane wave components that make up an optical field are altered upon transmission through the optical system. By designing metasurfaces with specific OTFs, we can perform analog optical computation.[4] For a coherent, linear, and spatially invariant optical system, the transmitted scalar optical field, $U(x,y)$, is given by

$$U(x,y) = \mathcal{F}^{-1}[H(k_x,k_y)\,\mathcal{F}[U_0(x,y)]], \tag{1}$$

where $U_0(x,y)$ is the incident field, $\mathcal{F}$ represents the Fourier transform, $\mathcal{F}^{-1}$ is the inverse Fourier transform, k_x and k_y are known as the spatial frequencies, and $H(k_x,k_y)$ is the OTF.[38] This particular expression ignores any polarization effects, in which the OTF is a tensor. The spatial frequencies are the transverse wavevector components of the plane waves that make up the optical field and are related to the angle of incidence through the relationship

$$k_x = k_0 \sin(\theta)\cos(\phi), \tag{2}$$

$$k_y = k_0 \sin(\theta)\sin(\phi), \tag{3}$$

where θ is the elevation angle of incidence, ϕ is the azimuthal angle, $k_0 = 2\pi/\lambda$ is the wavenumber, and λ is the free space wavelength. In an image, specific information is associated with different spatial frequencies. For example, edges and rapidly varying details in the field are carried by large spatial frequencies at large angles to the optical axis. Hence, edge enhancement or detection can be performed by blocking low spatial frequencies and transmitting large spatial frequencies.[39,40] An OTF with a quadratic dependence on spatial frequency, as a further example, can be used to perform edge detection. We can selectively modify the spatial frequencies in the object plane by controlling the angular response of a metasurface to perform optical computation.[4] An important metric for designing OTFs is the numerical aperture (NA). The NA characterizes the range of spatial frequencies over which the OTF performs the operation of interest.

B. Pattern generation

We compare four different methods to parameterize the unit cell of a metasurface in topology optimization. Two parameterizations are pixel based, where $\gamma(x,y)$ is broken up into a discrete grid: a constrained pixel parameterization *px* and an unconstrained pixel parameterization *LMpx*, a latent matrix pixel based approach. A third optimization method referred to as *NN* is parameterized entirely by the weights and biases of a convolutional neural network that outputs a pixel based grid as its final layer, and the fourth method is a hybrid between *NN* and *LMpx* called *NNpx*. A schematic detailing how each parameterization generates a specific material density is given in Fig. 1(c). In the *px* method, we optimize the material density, $\gamma(x,y)$, directly. During the optimization, we ensure that the density is constrained between 0 and 1. Values either above or below are clipped. For the *LMpx* method, the density is no longer constrained; we allow $-\infty < \gamma(x,y) < \infty$. During each iteration, we normalize the density by subtracting its mean and dividing it by its standard deviation to stop the values from becoming too large. To generate a valid pattern, we use a sigmoid projection function to interpolate from $-\infty < \gamma(x,y) < \infty$ to $0 \le \gamma(x,y) \le 1$. Details regarding the formulation and role of the projection functions are given below. Compared with the *px* representation, the *LMpx* representation features no clipping of values, which may produce better performing designs. The *LMpx* method is inspired by the studies in Refs. 41 and 42. For the *NN* method, the convolutional neural network outputs a 2D array representing the material density. Similar to the *LMpx* method, the values in this array are unconstrained, and we use the same projection method to interpolate the output into a valid material density. The *NNpx* method is a combination of the *NN* and *LMpx* methods, where we perform two optimizations. The first optimization is performed using *NN* for half the total number of optimization steps. The density produced by the neural network is then used as the starting point for an *LMpx* optimization.

In metasurface TO, it is common to have a projection step, which is used to promote convergence to binary designs. The material density smoothly interpolates between the permittivity of air and the material of which the metasurface is composed; intermediate values are unphysical. The material density is fed into a projection function each iteration before the RCWA simulation. As the iterations progress, these projection functions slowly approach a step function, mapping their input to a binary output design. We use two different projection functions, f_1 for the *px* method and f_2 for all other methods,

$$f_1(\gamma,\beta_1) = \frac{\tanh(0.5\beta_1) + \tanh(\beta_1(\gamma - 0.5))}{2\tanh(0.5\beta_1)}, \tag{4}$$

$$f_2(\gamma,\beta_2) = \frac{1}{1+e^{-\beta_2\gamma}}, \tag{5}$$

where β_1 and β_2 control the degree of binarization in the pattern. The function f_2 is a sigmoid function and serves the additional purpose of interpolating from unconstrained values into valid constrained material densities, which is required for all methods except for the *px* method. As β_1 and β_2 increase, the rate of binarization provided by f_1 and f_2 differs. In order to ensure consistency between methods, we set $\beta_2 = 2\beta_1$. To force binarization toward the end of the optimization, we smoothly increase the value of β_2 each iteration, $\beta_2(i) = 1.4^{i/15}$, where i is the current iteration number. We found that by increasing the binarization at this rate, the final designs are completely binarized.

Typically, pixel based TO uses blurring to enforce minimum feature sizes and prevent the appearance of single pixel features.[43] Blurring is a convolution between the blurring kernel $w(x,y;\,r)$, in our case a cone with radius r, and the pixel values $\gamma(x,y)$. We use the kernel from Ref. 43, which is given below in Eq. (6). For efficiency, the convolution is performed as a multiplication in Fourier space,

$$w(x,y;r) = \begin{cases} \dfrac{r-\sqrt{x^2+y^2}}{1/3\pi r^3}, & \sqrt{x^2+y^2} < r, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

TABLE I. Neural network parameters.

Parameter	Value
$m = n$	20
n_c	16
t_i	$\{1, 2, 2, 3, 1\}$
c_i	$\{128, 64, 32, 16, 1\}$
Offset (O)/bias (B)	$\{O, O, O, B, B\}$
$N_x = N_y$	240

Here, each *px* and *LMpx* optimization is performed with four different levels of blurring, corresponding to different radii of the blurring kernel, r in Eq. (6). Optimization is performed without blurring, with $r = 0.05\lambda$ nm blurring, 0.1λ nm blurring, and 0.2λ nm blurring, where λ is the wavelength of light. For the pixel part of the *NNpx* method, only a 0.1λ nm radius blurring kernel is used. The blurring levels are chosen to produce a range of outcomes, from generating features that are too small to features that are too large, inhibiting device performance.

The neural network used for the *NN* approach is an upscaling CNN, similar to the one used by Hoyer *et al.*[26] A schematic of the neural network is given in Fig. 1(d). The input to the neural network is a trainable 1D vector of shape $(1, 1, m \times n)$, which is passed into a linear layer, producing a vector with shape $(1, 1, n \times m \times n_c)$, which is subsequently reshaped into a 3D array of shape (m, n, n_c). This is followed by five repetitions of an "upscaling" block, which consists of tanh non-linearity, a bilinear upscaling by a factor t_i for block i, normalization by subtracting the mean and dividing by standard deviation, a 2D convolutional layer with 5×5 kernels and c_i channels, and finally either an offset or bias layer. An offset layer is a trainable additive value that is different for each element and channel, whereas a bias layer is constant but is different across all channels. Following these upscaling blocks, the output of the neural network is a 2D array of shape (N_x, N_y) which, after passing through the projection function, yields the physical distribution of material in the metasurface unit cell. To ensure comparability between the neural network and pixel based approaches, the pixel based methods are set to operate on an $N_x \times N_y$ grid. The parameters for our neural network are given in Table I.

C. Design of benchmarks

To evaluate the performance of the different parameterization methods, we perform optimization on 150 different OTF matching tasks. We construct a set of benchmark optimizations to test the different parameterization methods. These tasks cover different wavelengths, polarizations, and numerical apertures (NA). In this context, NA refers to the range of normalized spatial frequencies (corresponding to angles of incidence) over which the optimization is performed; in particular, we optimize over 6 evenly spaced angles between 0° and $\arcsin(\mathrm{NA})$. The performance of the device for angles of incidence outside of the NA is not considered in the optimization. For simplicity, we focus on tailoring the 1D co-polarized angular response of the metasurface to be either quadratic, i.e., matching $Q(\theta;\ N_{\mathrm{NA}})$, or flat, i.e., matching $I(\theta;\ N_{\mathrm{NA}})$, where θ is the angle of incidence and N_{NA} is the NA,

$$Q(\theta; N_{\mathrm{NA}}) = \frac{1}{N_{\mathrm{NA}}^2}\sin^2\theta, \tag{7}$$

$$I(\theta; N_{\mathrm{NA}}) = 0.8. \tag{8}$$

This permits a design with the operation of interest carried by one polarization channel. Here, we will use the linear *s* and *p*-polarization basis. Independent OTFs can be encoded into the different polarization channels to multiplex multiple functionalities into a single metasurface. We perform the polarization dependent optimizations given in Table II.

TABLE II. Polarization dependent OTF matching tasks.

Opt. task	Requirement on *s*-pol	Requirement on *p*-pol
SQ	$Q(\theta;\ N_{\mathrm{NA}})$	⋯
PQ	⋯	$Q(\theta;\ N_{\mathrm{NA}})$
SQPQ	$Q(\theta;\ N_{\mathrm{NA}})$	$Q(\theta;\ N_{\mathrm{NA}})$
SQPF	$Q(\theta;\ N_{\mathrm{NA}})$	$I(\theta;\ N_{\mathrm{NA}})$
SFPQ	$I(\theta;\ N_{\mathrm{NA}})$	$Q(\theta;\ N_{\mathrm{NA}})$

TABLE III. Wavelength, period, and thickness combinations used in the optimizations.

Wavelength (nm)	Period ($x \times y$) (nm)	Silicon thickness (nm)
800	530 × 530	250
900	600 × 600	300
980	650 × 650	330
1300	860 × 860	500
1550	1000 × 1000	550

The task names describe the shape of the OTF for a given polarization. For example, "SQ" refers to optimizing for a quadratic *s*-polarized response, and "SFPQ" refers to a flat response for *s*-polarization and a quadratic response for *p*-polarization. In optimizations SQ and PQ, only a single polarization is being considered, and the performance of the metasurface for the other polarization can be arbitrary. We perform these five polarization dependent OTF matching tasks for $N_{\mathrm{NA}} = 0.05, 0.1, 0.15, 0.2, 0.3, 0.5$ and for the wavelength, period, and patterned layer thicknesses given in Table III. Each period is ~2/3 of the wavelength, preventing diffraction in the substrate, and the thickness is chosen to produce slightly less than a 2π phase shift, creating 150 benchmark tasks and a total of 1500 optimizations to run.

D. Optimization

Our metasurfaces are composed of periodically repeated unit cells; therefore, RCWA can be used to perform the optical simulations. RCWA is a semi-analytical solution to Maxwell's equations for periodic structures,[44,45] in which the electric and magnetic fields are expanded in a series of Floquet modes that represent propagating and evanescent diffracted orders. The fields inside each layer are decomposed into a superposition of eigenmodes, which depend on

the distribution of the material in the layer. Mode amplitudes and phases can be propagated between layers using scattering matrices to enforce the electromagnetic boundary conditions. The decomposition of the fields into a Fourier series, the Floquet modes, is what limits the accuracy of RCWA; the mode expansion needs to be truncated for computational efficiency. For the design of sub-wavelength metasurfaces, we use 19 × 19 Fourier orders in the RCWA expansion; increasing the number of orders increases both the time and the amount of memory required. We use TORCWA,[46] in particular, to benefit from the automatic differentiation engine in PyTorch. Rather than using adjoint methods,[47–49] automatic differentiation provides the gradient of the cost function with respect to our parameterization in a single simulation.

The specific angle of incidence over which we are optimizing, θ, is a rotation about the x axis [$\phi = \pi/2$ in Eqs. (2) and (3)], i.e., optimizing along k_y; s-polarization refers to the electric field vector being perpendicular to the plane of incidence (normal to the y-z plane), and p-polarization refers to the electric field vector being parallel to the plane of incidence (in the y-z plane). We focus on the optimization of amorphous silicon metasurfaces (optical constants taken from Ref. 50) on fused silica (optical constants taken from Ref. 51), and we assume that the substrate is semi-infinite. Light is incident on the air side, and the amplitude transmission is measured in the substrate.

Our cost function, $c(t, T)$, is the root mean square (RMS) error between a target OTF and the OTF of the device, defined in the following equation:

$$c(t,T) = \sqrt{\frac{1}{N_{\text{pol}} N_{\text{ang}}} \sum_{\text{pol}} \sum_{\text{ang}} (t_i - T_i)^2}, \tag{9}$$

where N_{pol} is the number of polarizations being considered and N_{ang} is the number of angles at which the OTF is sampled. The current metasurface amplitude transmission is t, and T is the target amplitude transmission for angle and polarization i.

Once the material density is generated, all optimizations follow the same optimization procedure. We calculate the angular response using RCWA and then the cost function and the gradient of the cost function with respect to the parameters using automatic differentiation. The gradient is fed into a gradient based optimization algorithm, ADAM,[52] to successively update the design. We perform the optimization over 300 iterations with a learning rate of 0.03 for both pixel parameterizations, 0.02 for the pixel part of the *NNpx* optimization, and 0.001 for the *NN*. Optimization for all methods is performed on a virtual machine with 8 vCPU cores (Intel Xeon Platinum 8562Y+), 61 GB of RAM, and a GPU with 12 GB of VRAM (NVIDIA L40). The *px* method is initialized with all gray values, $\gamma(x, y) = 0.5$, and the *LMpx* method is randomly initialized using a Gaussian distribution. Similarly, the weights and biases of the neural network are randomly initialized using a Gaussian distribution.

After the 300 optimization iterations, a threshold filter as given in Eq. (10) is applied to the pattern to binarize the design before the final value of the cost function and the OTF is calculated,

$$\begin{cases} 1, & \gamma(x,y) > 0.5, \\ 0, & \gamma(x,y) \le 0.5. \end{cases} \tag{10}$$

The minimum feature size of the final design is extracted using an algorithm by Chen *et al.*,[53] which uses a morphological opening operation to determine the minimum feature sizes in the solid and void regions. This operation erodes and then dilates the image, removing features that are smaller than the amount of erosion. By analyzing how the pattern changes as the amount of erosion is increased, the minimum feature size can be inferred. To compare optimizations at different wavelengths, the feature sizes are normalized by the wavelength, d_{fs}/λ.

III. RESULTS

A. Benchmark optimizations

Full tables containing the RMS error and minimum feature sizes for each of the benchmark optimizations are available in the supplementary material.

A specific example of one of the benchmark optimizations is given in Fig. 2. This is an SQPF optimization for 980 nm across an NA of 0.5. For this optimization, *LMpx* 5% blur, *NNpx*, and *LMpx* 10% are the top performing parameterizations with RMS errors of 0.0244, 0.0249, and 0.0469, respectively. Of these, the minimum feature size of the *NNpx* method is the largest at 24.4 nm, followed by *LMpx* 10% at 21.7 nm and *LMpx* 5% at 13.5 nm. Figure 2(a) contains a bar chart showing the RMS error and the normalized minimum feature size produced by each method.

The *NN* and *NNpx* parameterizations produce qualitatively simple structures with large features, as shown in Fig. 2(b). Although *LMpx* with 10% blurring has a similar minimum feature size as *NNpx*, the resulting structure is less simple, with many disjointed features. Apart from a small hole in the *NNpx* design, the remaining structure is quite large, whereas *LMpx* 10% produces islands of material, and many features are close in size to the critical dimension. We can see from Fig. 2(c) that the OTFs produced by *NNpx*, *LMpx* 5%, and *LMpx* 10% match the target OTF quite well.

The pure *NN* parameterization gives a critical dimension of 46 nm; however, the performance is degraded, with an RMS error of 0.172. For optical computing applications, the performance of the *NN* structure is unsuitable. In particular, the transmission of the *NN* structure at normal incidence, $ky/k_0 = 0$, is too high, as shown in Fig. 2(c). If used to perform edge enhanced imaging, it would result in poor extinction of the background light, making the edges harder to see. The *NNpx* method is a good compromise between the pixel and *NN* approaches, inheriting the large features from the *NN* method and good performance from the *LMpx* methods.

During the optimization process, the *NN* method is able to converge much more rapidly toward a design than pixel based methods. Figure 2(d) shows the evolution of the cost function during the optimization process. For the *NN* parameterization, the RMS error becomes less than 0.1 in less than 20 iterations, faster than the *LMpx* methods. Unfortunately, this method diverges from a well-performing design toward the end of the optimization, producing a sub-optimal metasurface. Small changes to the weights and biases of the neural network are able to rapidly change the geometry of the metasurface, which results in this rapid convergence. While not implemented here, task-specific learning rate scheduling and early

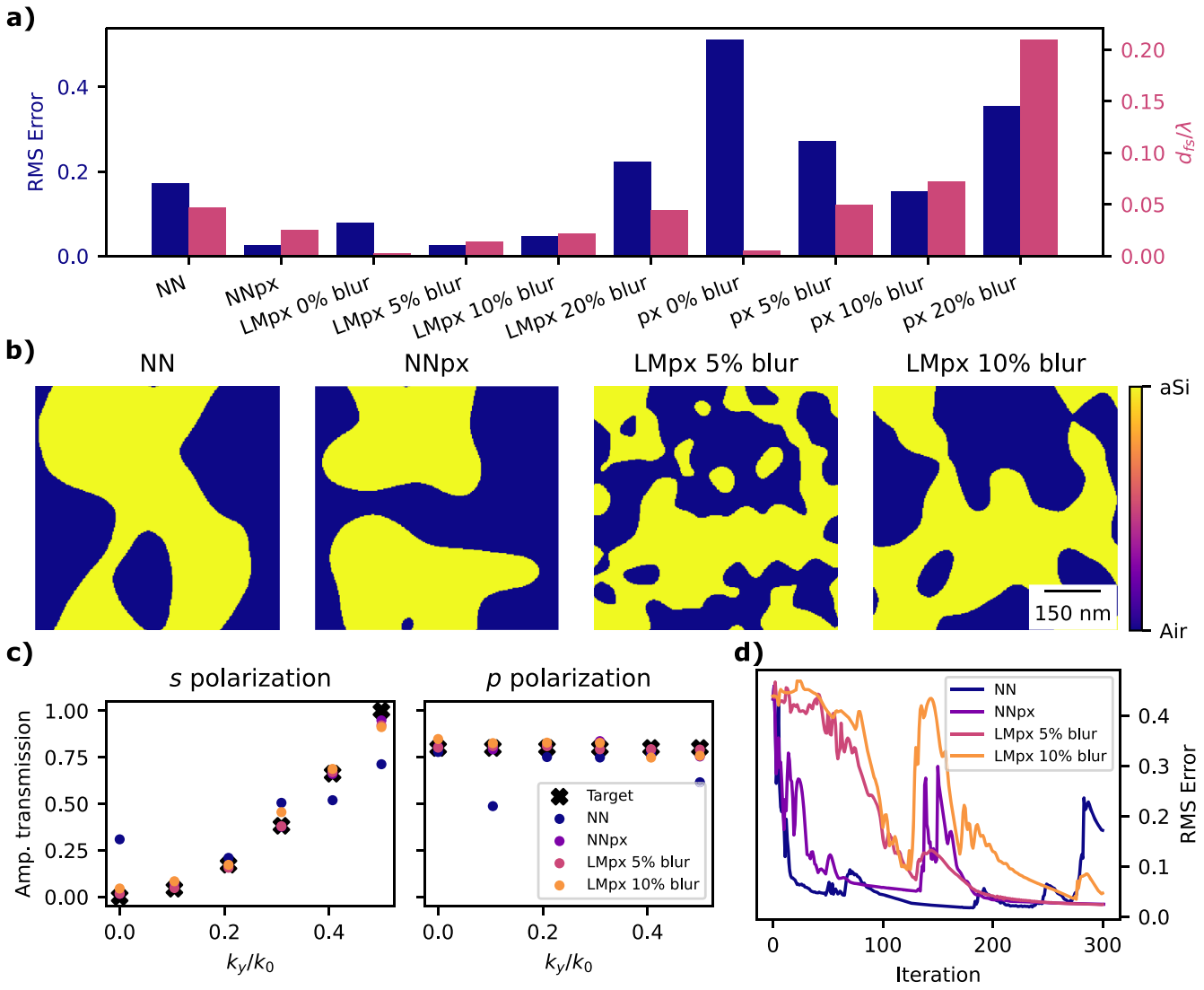

FIG. 2. Example benchmark optimization. The target is a quadratic angular response for *s*-polarized light and a flat angular response for *p*-polarized light at 980 nm across a numerical aperture of 0.5. The period of the unit cell is 650 nm. (a) Histogram comparing the RMS error of a final design with the minimum feature size, (b) Examples of designs produced, (c) OTFs of the designs in (b) compared with the target, and (d) cost function evolution.

stopping for cost function values below a given threshold could prevent this.

Since the *NN* and *NNpx* methods are identical for the first 150 iterations, we would expect the cost functions to follow the same trajectory; however, there is some slight variability. Although we set the same seed for the random number generation, differences in the design can occur due to the non-deterministic algorithms used in the automatic differentiation engine. At iteration 150, the *NNpx* method switches from *NN* to *LMpx*, and this is accompanied by a small increase in the cost function, which then slowly decreases until the end of the optimization. The *LMpx* optimizations tend to converge slowly at first before rapidly converging to a design.

The key results from the 150 benchmark tasks can be visualized by plotting them on a 2D histogram, as shown in Fig. 3. The *y* axis is the normalized feature size on a log scale, d_{fs}/λ, and the *x* axis is the value of the cost function. Ideally, the points would be clustered in the top-left of the plot, which represents designs with large features and a small cost function. Designs along the bottom, with the smallest feature sizes, may contain single pixel features, which make them impractical to fabricate. The average feature sizes and cost function produced by each parameterization for the design of computational metasurfaces are given in Table IV. If the cost function is too large, then the OTF may be unsuitable for optical computation either due to a lack of contrast or by not matching the desired shape of the

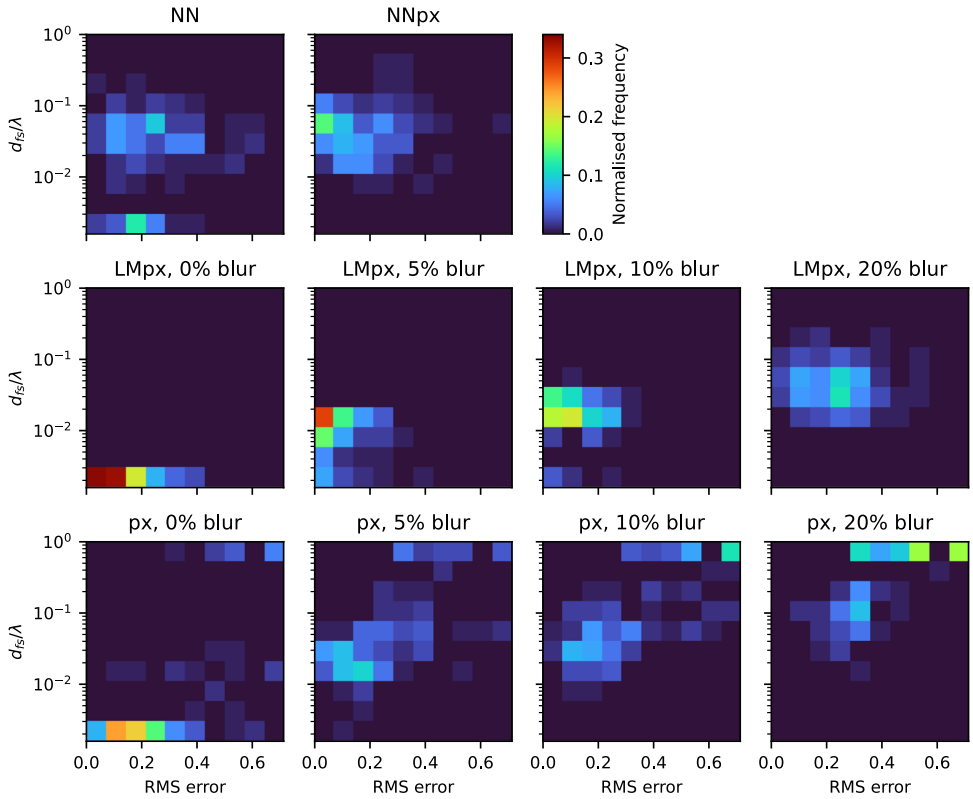

FIG. 3. Normalized histogram for each parameterization method comparing over the 150 benchmark optimization tasks. Along the *x* axis is the RMS error from the target OTF, and the *y* axis is the minimum feature size normalized by the wavelength. Note that the scale along the *y* axis is non-linear.

OTF. Contrast in this case refers to the difference between the maximum and minimum transmittance value of the OTF. By maximizing the contrast, for a quadratic OTF, the computational operation performed by the metasurface will have a higher SNR and be more robust.

When the amount of blurring increases, the minimum feature size tends to increase. Increasing the blurring from 5% to 20%, there is an increase in the minimum feature size but a decrease in performance. For both the *px* and *LMpx* methods, as the blurring increases, the cluster of optimizations on the histogram moves toward the upper-right corner. With more blurring, the variance in the final cost function is greater, and the parameterization becomes less consistent at producing well performing devices. Simply blurring the design is not an ideal approach for increasing the minimum feature size due to this decrease in performance. Interestingly, a small amount of blurring performs better than no blurring. The median RMS error for *LMpx* decreases by 42% when comparing no blurring to 5% blurring. Physically, this could mean that features that are too small do not interact strongly with light, and a small amount of blurring causes larger regions to form that are of a size where they can interact more strongly with light. In addition, features that are too small may give erroneous optical properties when using RCWA since the small details may not be accommodated in the Fourier series truncation.

TABLE IV. Statistical properties of the different parameterizations.

Method	Mean d_{fs}/λ	Median d_{fs}/λ	Mean RMS error	Median RMS error
NN	0.0346	0.0304	0.2253	0.2009
NNpx	0.0539	0.0469	0.1609	0.1383
LMpx 0%	0.0027	0.0028	0.1220	0.0965
LMpx 5%	0.0097	0.0110	0.0822	0.0560
LMpx 10%	0.0177	0.0166	0.1157	0.0973
LMpx 20%	0.0495	0.0403	0.2246	0.2282
px 0%	0.0783	0.0028	0.2558	0.1904
px 5%	0.1367	0.0314	0.2529	0.2093
px 10%	0.2289	0.0718	0.3376	0.2769
px 20%	0.4435	0.6452	0.4316	0.4132

The *NN* approach has a large variance in both the size of the features produced and the RMS error. Although the feature sizes produced by *NN* are quite large, with a median critical dimension of 3.04% of the wavelength, the median RMS error is 0.201. The ability of *NN* to produce large features could be ascribed to the upscaling in the network architecture, allowing the optimization to occur on several length scales simultaneously. In some example optimizations, such as SQPF NA 0.5 with 980 nm illumination (Fig. 2), the *NN* ends up diverging from a solution with good performance. In addition, the *NN* approach can sometimes converge to designs with single pixel features. This cluster of single pixel designs along the bottom of the histogram is disconnected from the cluster of large feature designs. Unlike the pixel methods, the *NN* parameterization

has no explicit mechanism to prevent these single pixel features from forming. In structural mechanics, where the *NN* parameterization produced notably simpler designs than the pixel counterparts,[26] single pixels and discontinuous features are avoided since they cannot support any load. In photonics, however, these sub-wavelength features can strongly interact with light and cannot be naturally discarded.

By combining the *NN* method with a pixel based method with blurring to form *NNpx*, this single pixel feature problem can be overcome to produce well-performing large structures. None of the benchmark optimizations have single pixel features, and the performance is superior to the "vanilla" *NN* parameterization. In the 150 benchmarks, *NNpx* produced no single pixel features. Even if the *NN* portion of the method produces single pixel features, the refinement by *LMpx* in the second half of the process removes them, producing larger features in the final design. The most populated bin in the histogram for the *NNpx* method has feature sizes similar to *LMpx* 20% but with a smaller error. The median critical dimension for *NNpx* is 4.69% of the wavelength, which is 16% larger than the critical dimension produced by *LMpx* 20% with a median RMS error 39% lower than *LMpx* 20%.

The *LMpx* approach is generally superior to the *px* method. The mean and median RMS errors for *LMpx* are lower than the *px* method with equivalent blurring. Increasing the blurring for *px* increases the chance of the optimization diverging, heading to the top-right corner of the histogram. The points in the top-right corner represent divergence to a thin film design, i.e., a homogeneous material distribution. This situation occurs only for the *px* method and becomes more likely as the amount of blurring increases. Although the average feature size for *px* 0% is 7.8% of the wavelength, this is skewed by designs that diverge to become planar films of material or voids. Looking at the histogram in Fig. 3, the best performing *px* 0% metasurfaces are along the bottom of the histogram with single pixel features. These problems do not occur for *LMpx*. Although the performance degrades as blurring increases, the drop in performance is not nearly as large. The superiority of *LMpx* to *px* can be attributed to the fact that the design density can always be updated in each iteration since it is unbounded, whereas changes to *px* can be clipped, which prevents improvements to the design from being made.

The best performing methods are *LMpx* 10% and *NNpx*. Both these parameterizations are able to produce metasurfaces with low RMS error and feature sizes large enough to ensure they remain possible to fabricate. The median RMS error for *LMpx* 10% is 1.4 times smaller than that of *NNpx*. Although better performing than *NNpx*, the median minimum feature size is only 1.7% of the wavelength, 2.8 times smaller than the median feature sizes produced by *NNpx*. Comparing the histograms of *NNpx* and *LMpx* 10% in Fig. 3, the *LMpx* 10% results are clustered closer together; the chance of producing a high performance metasurface is higher. Generally, *NNpx* produces metasurfaces with larger critical dimensions and slightly degraded performance. The inconsistency in the performance of *NNpx* designs could be minimized by tweaking the optimization hyperparameters for each specific optimization task.

The pixel part of *NNpx* has a 10% blur applied; the difference in feature sizes between *LMpx* 10% and *NNpx* can be attributed to the starting point of the optimization. The *NN* method converges rapidly to designs with large features, which provides an ideal initial design for a pixel based approach. The larger features could be due to the upscaling layers in the neural network architecture, which permit the *NN* method to optimize on different length scales simultaneously. Due to this rapid convergence, perhaps *NNpx* could be improved by performing 50 iterations of *NN* and 250 iterations of *LMpx* 10% to produce better performing freeform computational metasurfaces with large critical dimensions.

The average time to design a metasurface with the *NN* approach was 855 s, which is similar to the 832 s for *LMpx* 10%. Although more operations occur during the forward and backward passes of the *NN*, the time difference between the neural network and pixel based approaches is negligible. The most time intensive step is the RCWA simulation; the parameterization step is insignificant in comparison.

B. Angular bandpass filter

As a final test of the neural network reparameterization, we design a polarization switchable angular bandpass filter for 1550 nm illumination. Due to the relation between spatial frequency and angle of incidence, such a metasurface acts as a bandpass filter for spatial frequency information. The goal is to design a metasurface that is transparent to light incident between +5° and +10° and opaque otherwise when illuminated with *s*-polarized light and transparent to light traveling between −5° and −10° for *p*-polarized light. This kind of optical response is highly non-trivial to design using geometric optical resonators. We choose a period of 1500 × 1500 nm^2 and use 550 nm for the thickness of the patterned layer. An asymmetry about normal incidence in the zeroth order can be generated through diffraction, whereby energy in that order is removed by asymmetric diffraction into other propagating diffracted orders. Since the period is larger, we use 29 × 29 orders in the RCWA simulation.

In terms of optical performance, the three best angular bandpass filters are produced by *px* 0%, *LMpx* 0%, and *NNpx* with cost functions of 0.243, 0.244, and 0.269, respectively. Of these, the *NNpx* design has the largest critical dimension of 43.75 nm, compared to the 6.25 nm of both the pixel methods. The final designs of these methods are given in Fig. 4(b). Although sub-nm precision in the fabrication process is unrealistic, these feature sizes are multiples of a single pixel, which represents a 6.25 × 6.25 nm^2 area. Qualitatively, the *NNpx* design is much simpler, with large continuous structures, and the designs produced by the pixel methods are permeated by holes and contain isolated islands of material that are impractical for fabrication. Although the features in *LMpx* 5% appear much larger than those in *LMpx* 0%, single pixel features appear in the design, and they have the same critical dimension.

The *NN* designed structure, see the supplementary material, has poor performance and single pixel features occur in the final design. Toward the end of the optimization, the neural network diverges from a design with performance on par with *NNpx*. We note that the single pixel features in *NN* form during the divergence. Prior to the divergence, the edges of the silicon regions are smooth. After the divergence, these edges become jagged and extremely thin, lowering the feature size by an order of magnitude.

The optical performance for *s*-polarization matches the desired response quite well for all methods, although reaching a maximum

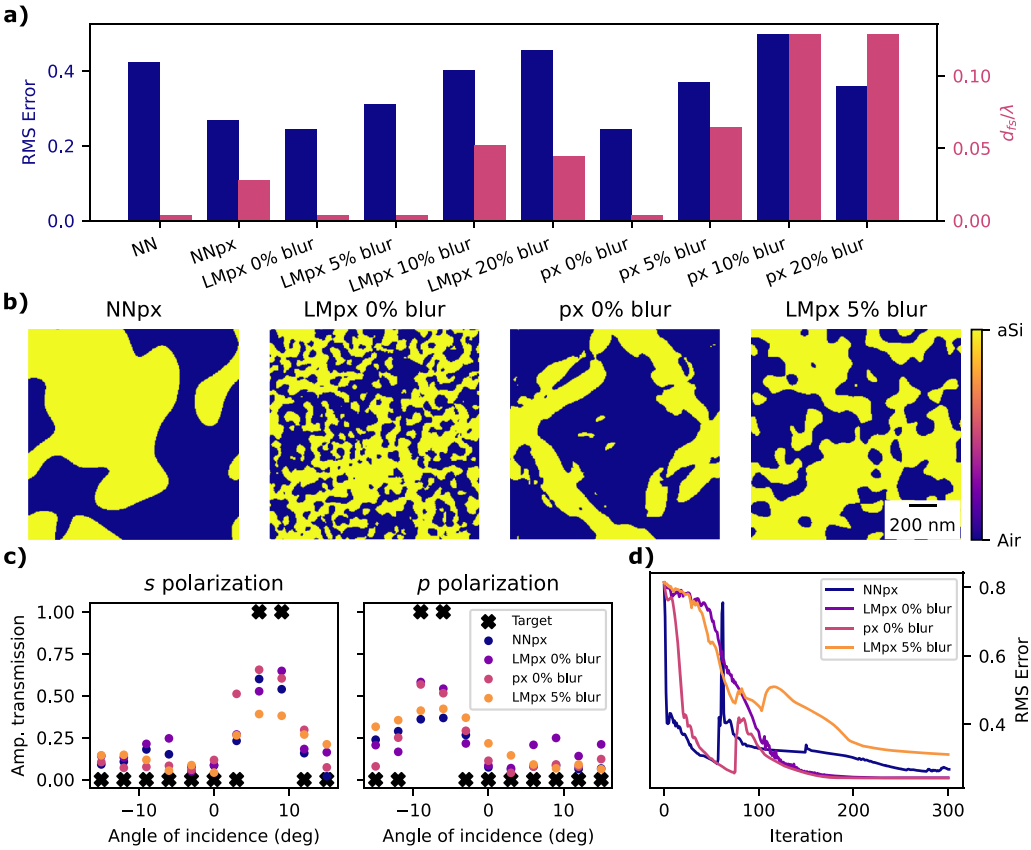

FIG. 4. Optimization of an angular bandpass filter. (a) Comparison of the RMS error and the normalized feature size for each optimization method, (b) final designs produced by selected parameterization, (c) the angular transmission of these selected designs, and (d) the evolution of the cost function during these optimizations.

transmission of only ≈ 70% and the extinction in the stopband is suitable. The major differences between the target and actual performance of the methods are in the transmission in the passband; the maximum passband transmission for *NNpx*, *LMpx* 0%, and *px* 0% is all quite similar. The widths of the passbands for *NNpx* and *LMpx* 0% are similar and match the target response quite well, whereas *px* 0% and *LMpx* 5% have a wider passband window than desired.

Compared to the *s*-polarization performance, the performance of the metasurface designs for *p*-polarization is worse. The transmission in the passband is lower than for *s*-polarization, and the width of the passband is larger than desired. The extinction in the stopband from -15° to -10° for all methods could be improved, particularly for *NNpx* and *LMpx* 5%; however, the extinction in the other stopband for these methods, from -5° to 15°, matches the target. The lower RMS error in the *px* 0% parameterization for this optimization task comes from its superior performance for *p*-polarization compared to the other methods.

We note that the target function for this optimization was quite unrealistic. Setting a smoother transition from passband to stopband may prove beneficial to ensure the convergence of all parameterizations to this unique optical response.

The evolution of the *NNpx* design throughout the optimization is displayed in Fig. 5(a). The starting point for *NN* is noise; within 15 iterations, the design is almost completely binary with large features. Iteration 150 is the final iteration of *NN*, and iteration 151 is the first iteration of the *LMpx* part of *NNpx*. We can see that the features produced by *NN* become smoother and small features disappear, and this is accompanied by a small increase in the cost function, as shown in Fig. 4(d). By iteration 250, the design is practically identical to the final design. Most of the design is generated using the *NN*, and the pixel method subsequently refines it.

We use a finite element method (FEM) solver (COMSOL Multiphysics) to validate the optical performance of the *NNpx* design, the details of which are in the supplementary material. As shown in Fig. 5(b), there is a slight discrepancy between the performance of the metasurface obtained through the RCWA optimization and from the FEM simulation of the optimized design. There is a small blueshift in the operational wavelength from 1550 to 1542.5 nm; however, the angular bandpass functionality is still present at 1550 nm but with lower contrast. At 1542.5 nm, the optical properties calculated with FEM agree with the optimized response quite well, with the largest discrepancies occurring in the stopbands.

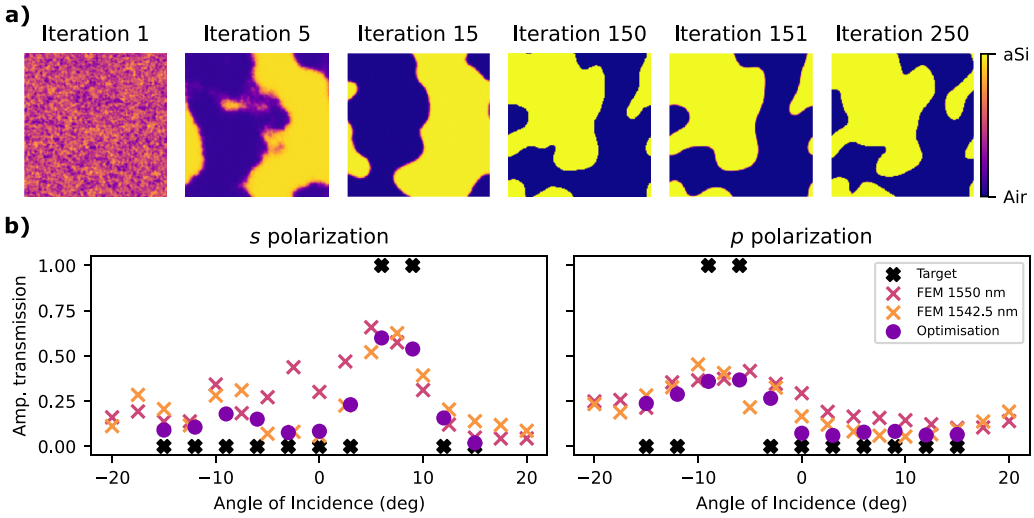

FIG. 5. Properties of the *NNpx* angular bandpass filter. (a) Design at different iterations during the optimization. (b) Comparison between the optical properties from the optimization (RCWA) and the finite element method.

Although there are discrepancies between RCWA and FEM, RCWA remains valuable due to its speed and lower memory consumption. If used in the optimization, FEM would require four simulations, forward and adjoint simulations for each polarization, compared to the single simulation with automatically differentiable RCWA. Sampling the OTF of this metasurface with *s* and *p* polarizations using FEM takes 4805 s compared to the 21 s of RCWA. FEM is performed on a machine with 32 vCPU cores and 128 GB of RAM.

This kind of metasurface could be used for optical security. In a noisy image, this metasurface would be able to spatially filter very specific angular frequencies, which could be used to store information. Information stored at particular spatial frequencies can be isolated by this metasurface, enabling a signal to be recovered. In image processing, this metasurface could perform directional edge detection in the object plane. With *s*-polarization, the top edges in an image would be revealed, whereas for *p*-polarization, the bottom edges would be enhanced. Finally, a metasurface with this angular bandpass response could also be used for single shot quantitative phase recovery.[54]

To illustrate the ability of the *NNpx* method to produce large minimum feature sizes, we compare it with the metasurface presented by Sell *et al.*,[55] which maximizes the diffracted light into the +1 diffracted order, which travels at 75° to the optical axis. Using the *NNpx* method, we produce a design with similar performance but with a minimum feature size 70% larger (see the supplementary material). The metasurface produced by Sell *et al.* has robustness considerations incorporated into their optimization process. Although not a perfect comparison, this illustrates the ability of the *NNpx* method to produce metasurfaces with relatively large critical dimensions.

With enough adjustment of the optimization parameters, each parameterization approach could produce high performance designs with suitable feature sizes for fabrication. Certain methods, however, are more consistent in producing high quality designs with large features, such as *NNpx* and *LMpx*. For each specific problem, the number of iterations, learning rates, starting conditions, and binarization rates can be changed to generate metasurfaces with suitable performance for the optical computation operations considered here. Furthermore, the period and thickness of the metasurface may be sub-optimal and could be refined for each specific optimization task to increase the performance of a particular design. Performing optimizations for different thicknesses of silicon could produce a thinner device with better efficiency.[56] For the *NN* approach, a different architecture or starting seed may improve the performance of the resulting metasurfaces and prevent single pixel features from occurring. A common technique used in machine learning is to dynamically decrease the learning rate to avoid divergence at the end of the training. Although not explored in this study, this may improve the ability of all the parameterization methods to produce well-performing metasurfaces. In particular, the *NN* approach tends to diverge toward the end of the optimization; by steadily lowering the learning rate, this may be prevented. Early stopping may also prevent divergence of the *NN* method. Once the cost function is below a certain threshold and the design is fully binarized, the optimization could be stopped prematurely to give a metasurface with good performance. During this study, only one blurring kernel was used for the pixel methods; perhaps a different kernel, such as a Gaussian, may be more effective at evolving toward larger features in the metasurface design without compromising the performance quality. We did not explore the effect of symmetry requirements on the parameterization methods. By forcing the resulting metasurfaces to have an axis of symmetry, the resulting structures may be even simpler or have larger feature sizes.

Feature size is used as a proxy for how appropriate a particular inversely designed metasurface is for fabrication. Another important consideration is the tolerance of the design to small imperfections in the fabrication process, commonly referred to as robustness. Robustness to dilated and eroded designs can be added into TO

algorithms separately from the parameterization process.[13] More complex robustness considerations, derived from the fundamental fabrication processes (spot size and step sizes of the lithography process), can also be considered to ensure free-form devices satisfy the constraints of the fabrication processes.[57,58] These can be incorporated into the pattern generation steps of the TO processes outlined in this study. Whether the parameterization affects the overall robustness of the metasurface is an interesting avenue for future study.

During this study, we optimize for the 1D OTF, but a natural extension of this study would be optimizing for the full 2D OTF, which requires sampling the OTF at more spatial frequency points and would greatly increase the computational cost of the optimization. This method could also be applied to design large area aperiodic metasurfaces, such as lenses, by breaking up the metasurface into small sections that can be designed with topology optimization with periodic boundary conditions.[11]

IV. CONCLUSION

We have investigated neural network reparameterization in topology optimization for the design of optical metasurfaces. Specifically, we replaced the conventional pixel parameterization with a convolutional neural network. Based on 150 example problems, we found that a hybrid pixel-neural-network approach produced designs 2.8 times larger than the best performing pixel-based method, with only a 1.4 times decrease in the optical performance, significantly reducing fabrication complexity. Compared with a pixel method capable of producing similar critical dimensions, the optical performance of the pixel-neural-network approach is improved by 39%.

We envisage that further optimization of the approach is possible by using more advanced neural network architectures or tuning the neural network hyperparameters for each specific problem, with extension to more complex computing operations.

SUPPLEMENTARY MATERIAL

In the supplementary material, we include details about the simulations performed using COMSOL Multiphysics, present all the designs produced in the angular bandpass filter section, compare the *NNpx* method with an existing metasurface design in the literature, and include the tables that contain the benchmark optimization data.

ACKNOWLEDGMENTS

This study was supported by the Australian Government through the Australian Research Council Center of Excellence Grant (No. CE200100010). L.C. would like to acknowledge the support of the Australian Government Research Training Program Scholarship.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Lincoln Clark: Conceptualization (lead); Formal analysis (lead); Investigation (lead); Software (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Ann Roberts**: Conceptualization (supporting); Funding acquisition (lead); Supervision (lead); Writing – review & editing (equal). **Lukas Wesemann**: Conceptualization (equal); Project administration (lead); Writing – original draft (supporting); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material. The code used during this study is made publicly available at https://github.com/LincolnClark/NN-vs-Px-top-opt.

REFERENCES

[1] A. I. Kuznetsov, M. L. Brongersma, J. Yao, M. K. Chen, U. Levy, D. P. Tsai, N. I. Zheludev, A. Faraon, A. Arbabi, N. Yu *et al.*, "Roadmap for optical metasurfaces," ACS Photonics **11**, 816–865 (2024).
[2] A. Arbabi and A. Faraon, "Advances in optical metalenses," Nat. Photonics **17**, 16–25 (2023).
[3] P. Berini, "Optical beam steering using tunable metasurfaces," ACS Photonics **9**, 2204–2218 (2022).
[4] L. Wesemann, T. J. Davis, and A. Roberts, "Meta-optical and thin film devices for all-optical information processing," Appl. Phys. Rev. **8**, 031309 (2021).
[5] D. N. Neshev and A. E. Miroshnichenko, "Enabling smart vision with metasurfaces," Nat. Photonics **17**, 26–35 (2023).
[6] L. Li, H. Ruan, C. Liu, Y. Li, Y. Shuang, A. Alù, C.-W. Qiu, and T. J. Cui, "Machine-learning reprogrammable metasurface imager," Nat. Commun. **10**, 1082 (2019).
[7] H. Zheng, Q. Liu, Y. Zhou, I. I. Kravchenko, Y. Huo, and J. Valentine, "Meta-optic accelerators for object classifiers," Sci. Adv. **8**, eabo6410 (2022).
[8] H. Zheng, Q. Liu, I. I. Kravchenko, X. Zhang, Y. Huo, and J. G. Valentine, "Multichannel meta-imagers for accelerating machine vision," Nat. Nanotechnol. **19**, 471–478 (2024).
[9] N. Yu and F. Capasso, "Flat optics with designer metasurfaces," Nat. Mater. **13**, 139–150 (2014).
[10] Z. Li, R. Pestourie, Z. Lin, S. G. Johnson, and F. Capasso, "Empowering metasurfaces with inverse design: Principles and applications," ACS Photonics **9**, 2178–2192 (2022).
[11] T. Phan, D. Sell, E. W. Wang, S. Doshay, K. Edee, J. Yang, and J. A. Fan, "High-efficiency, large-area, topology-optimized metasurfaces," Light: Sci. Appl. **8**, 48 (2019).
[12] Z. Li, R. Pestourie, J.-S. Park, Y.-W. Huang, S. G. Johnson, and F. Capasso, "Inverse design enables large-scale high-performance meta-optics reshaping virtual reality," Nat. Commun. **13**, 2409 (2022).
[13] E. W. Wang, D. Sell, T. Phan, and J. A. Fan, "Robust design of topology-optimized metasurfaces," Opt. Mater. Express **9**, 469–482 (2019).
[14] P. Dainese, L. Marra, D. Cassara, A. Portes, J. Oh, J. Yang, A. Palmieri, J. R. Rodrigues, A. H. Dorrah, and F. Capasso, "Shape optimization for high efficiency metasurfaces: Theory and implementation," Light: Sci. Appl. **13**, 300 (2024).
[15] D. Nelson, S. Kim, and K. B. Crozier, "Inverse design of plasmonic nanotweezers by topology optimization," ACS Photonics **11**, 85–92 (2023).
[16] A. Cordaro, B. Edwards, V. Nikkhah, A. Alù, N. Engheta, and A. Polman, "Solving integral equations in free space with inverse-designed ultrathin optical metagratings," Nat. Nanotechnol. **18**, 365–372 (2023).
[17] P. Pearson, G. Roberts, and A. Faraon, "Inverse-designed metasurfaces for multifunctional spatial frequency filtering," Optica **12**, 1090 (2025).

[18]N. Li, J. Zhang, D. N. Neshev, and A. A. Sukhorukov, "Inverse design of nonlinear metasurfaces for sum frequency generation," Nanophotonics **13**, 3363 (2024).
[19]B. Leng, Y. Zhang, D. P. Tsai, and S. Xiao, "Meta-device: Advanced manufacturing," Light: Adv. Manuf. **5**, 117–132 (2024).
[20]H. Duan, H. Hu, K. Kumar, Z. Shen, and J. K. W. Yang, "Direct and reliable patterning of plasmonic nanostructures with sub-10-nm gaps," ACS Nano **5**, 7593–7600 (2011).
[21]S. Fruncillo, X. Su, H. Liu, and L. S. Wong, "Lithographic processes for the scalable fabrication of micro- and nanostructures for biochips and biosensors," ACS Sens. **6**, 2002–2024 (2021).
[22]J. Seong, Y. Jeon, Y. Yang, T. Badloe, and J. Rho, "Cost-effective and environmentally friendly mass manufacturing of optical metasurfaces towards practical applications and commercialization," Int. J. Precis. Eng. Manuf.-Green Technol. **11**, 685–706 (2023).
[23]R. E. Christiansen and O. Sigmund, "Inverse design in photonics by topology optimization: Tutorial," J. Opt. Soc. Am. B **38**, 496–509 (2021).
[24]A. Chandrasekhar and K. Suresh, "TOuNN: Topology optimization using neural networks," Struct. Multidiscip. Optim. **63**, 1135–1149 (2021).
[25]D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2018), pp. 9446–9454.
[26]S. Hoyer, J. Sohl-Dickstein, and S. Greydanus, "Neural reparameterization improves structural optimization," arXiv:1909.04240 [cs.LG] (2019).
[27]S. M. Sanu, A. M. Aragon, and M. A. Bessa, "Neural topology optimization: The good, the bad, and the ugly," arXiv:2407.13954 [cs.LG] (2024).
[28]Y. Deng, S. Ren, K. Fan, J. M. Malof, and W. J. Padilla, "Neural-adjoint method for the inverse design of all-dielectric metasurfaces," Opt. Express **29**, 7526 (2021).
[29]W. Li, H. Barati Sedeh, D. Tsvetkov, W. J. Padilla, S. Ren, J. Malof, and N. M. Litchinitser, "Machine learning for engineering meta-atoms with tailored multipolar resonances," Laser Photonics Rev. **18**, 2300855 (2024).
[30]Y. Deng, K. Fan, B. Jin, J. Malof, and W. J. Padilla, "Physics-informed learning in artificial electromagnetic materials," Appl. Phys. Rev. **12**, 011331 (2025).
[31]L. Jiang, X. Li, Q. Wu, L. Wang, and L. Gao, "Neural network enabled metasurface design for phase manipulation," Opt. Express **29**, 2521 (2021).
[32]S. An, C. Fowler, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, J. Ding, A. M. Agarwal, C. Rivero-Baleine, K. A. Richardson, T. Gu, J. Hu, and H. Zhang, "A deep learning approach for objective-driven all-dielectric metasurface design," ACS Photonics **6**, 3196–3207 (2019).
[33]Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," Opt. Express **28**, 11618 (2020).
[34]B. Wu, G. Wang, K. Liu, G. Hu, and H.-X. Xu, "Equivalent-circuit-intervened deep learning metasurface," Mater. Des. **218**, 110725 (2022).
[35]W. Ji, J. Chang, H.-X. Xu, J. R. Gao, S. Gröblacher, H. P. Urbach, and A. J. L. Adam, "Recent advances in metasurface design and quantum optics applications with machine learning, physics-informed neural networks, and topology optimization methods," Light: Sci. Appl. **12**, 169 (2023).
[36]J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, "Free-form diffractive metagrating design based on generative adversarial networks," ACS Nano **13**, 8872–8878 (2019).
[37]S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," Nat. Photonics **12**, 659–670 (2018).
[38]J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed. (McGraw-Hill Publishing, 1996).
[39]L. Wan, D. Pan, S. Yang, W. Zhang, A. A. Potapov, X. Wu, W. Liu, T. Feng, and Z. Li, "Optical analog computing of spatial differentiation and edge detection with dielectric metasurfaces," Opt. Lett. **45**, 2070 (2020).
[40]A. Komar, R. A. Aoni, L. Xu, M. Rahmani, A. E. Miroshnichenko, and D. N. Neshev, "Edge detection with Mie-resonant dielectric metasurfaces," ACS Photonics **8**, 864–871 (2021).
[41]M. Chen, J. Jiang, and J. A. Fan, "Design space reparameterization enforces hard geometric constraints in inverse-designed nanophotonic devices," ACS Photonics **7**, 3141–3151 (2020).
[42]C. Kim, J. Hong, J. Jang, G.-Y. Lee, Y. Kim, Y. Jeong, and B. Lee, "Freeform metasurface color router for deep submicron pixel image sensors," Sci. Adv. **10**, eadn9000 (2024).
[43]F. Wang, B. S. Lazarov, and O. Sigmund, "On projection methods, convergence and robust formulations in topology optimization," Struct. Multidiscip. Optim. **43**, 767–784 (2010).
[44]M. G. Moharam and T. K. Gaylord, "Rigorous coupled-wave analysis of planar-grating diffraction," J. Opt. Soc. Am. **71**, 811 (1981).
[45]M. G. Moharam, T. K. Gaylord, E. B. Grann, and D. A. Pommet, "Formulation for stable and efficient implementation of the rigorous coupled-wave analysis of binary gratings," J. Opt. Soc. Am. A **12**, 1068 (1995).
[46]C. Kim and B. Lee, "TORCWA: GPU-accelerated Fourier modal method and gradient-based optimization for metasurface design," Comput. Phys. Commun. **282**, 108552 (2023).
[47]J. S. Jensen and O. Sigmund, "Topology optimization for nano-photonics," Laser Photonics Rev. **5**, 308–321 (2010).
[48]C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," Opt. Express **21**, 21693 (2013).
[49]A. C. R. Niederberger, D. A. Fattal, N. R. Gauger, S. Fan, and R. G. Beausoleil, "Sensitivity analysis and optimization of sub-wavelength optical gratings using adjoints," Opt. Express **22**, 12971 (2014).
[50]D. T. Pierce and W. E. Spicer, "Electronic structure of amorphous Si from photoemission and optical studies," Phys. Rev. B **5**, 3017–3029 (1972).
[51]I. H. Malitson, "Interspecimen comparison of the refractive index of fused silica," J. Opt. Soc. Am. **55**, 1205 (1965).
[52]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 [cs.LG] (2017).
[53]M. Chen, R. E. Christiansen, J. A. Fan, G. Işiklar, J. Jiang, S. G. Johnson, W. Ma, O. D. Miller, A. Oskooi, M. F. Schubert, F. Wang, I. A. D. Williamson, W. Xue, and Y. Zhou, "Validation and characterization of algorithms and software for photonics inverse design," J. Opt. Soc. Am. B **41**, A161 (2024).
[54]J. Liu, H. Wang, Y. Li, L. Tian, and R. Paiella, "Asymmetric metasurface photodetectors for single-shot quantitative phase imaging," Nanophotonics **12**, 3519–3528 (2023).
[55]D. Sell, J. Yang, S. Doshay, R. Yang, and J. A. Fan, "Large-angle, multifunctional metagratings based on freeform multimode geometries," Nano Lett. **17**, 3752–3757 (2017).
[56]J. Yang and J. A. Fan, "Analysis of material selection on dielectric metasurface performance," Opt. Express **25**, 23899 (2017).
[57]A. Y. Piggott, E. Y. Ma, L. Su, G. H. Ahn, N. V. Sapra, D. Vercruysse, A. M. Netherton, A. S. P. Khope, J. E. Bowers, and J. Vučković, "Inverse-designed photonics for semiconductor foundries," ACS Photonics **7**, 569–575 (2020).
[58]M. F. Schubert, A. K. C. Cheung, I. A. D. Williamson, A. Spyra, and D. H. Alexander, "Inverse design of photonic devices with strict foundry fabrication constraints," ACS Photonics **9**, 2327–2336 (2022).