Article

Shaping freeform nanophotonic devices with geometric neural parameterization

Check for updates

Tianxiang Dai[1], Yixuan Shao[1], Chenkai Mao[1], Yu Wu[1], Sara Azzouz[1], You Zhou[2] & Jonathan A. Fan[1] ✉

Nanophotonic freeform design has the potential to push the performance of optical components to new limits, but there remains a challenge to effectively perform optimization while reliably enforcing design and manufacturing constraints. We present Neuroshaper, a framework for freeform geometric parameterization in which nanophotonic device layouts are defined using an analytic neural network representation. Neuroshaper serves as a qualitatively new way to perform shape optimization by capturing multi-scalar, freeform geometries in an overparameterized representation scheme, enabling effective optimization in a smoothened, high dimensional geometric design space. We show that Neuroshaper can enforce constraints and topology manipulation in a manner where local constraints lead to global changes in device morphology. We further show numerically and experimentally that Neuroshaper can apply to a diversity of nanophotonic devices. The versatility and capabilities of Neuroshaper reflect the ability of neural representation to augment concepts in topological design.

Free space and on-chip nanophotonic systems have emerged as revolutionary platforms for controlling, routing, filtering, and transducing photons using subwavelength-scale structured media[1–10]. The immense versatility of these platforms largely derives from the strong relationship between device geometry and optical response at subwavelength scales, which enables an exponentially large design space for optical manipulation. Traditionally, these structure-function relations have derived from the use of simple, physically intuitive geometries in the form of nanoscale waveguides and resonators, together with their tailored coupling[11–16]. Such simplifications have led to robust and scalable design rules for many classes of photonic systems, however, these concepts also pose fundamental limitations in performance and device footprint. To overcome these limits, a wide range of freeform inverse design algorithms have been proposed that utilize a much larger design space of possible device geometries to achieve an objective[17–20]. The most advanced and computationally efficient algorithms utilize gradients calculated using the adjoint variables method or automatic differentiation to perform iterative optimization[21–27].

A core feature of all inverse design methodologies is the specification of a geometric parameterization scheme that describes the device layout. The choice of parameterization scheme is critical to reliably enforce constraints to the device geometry, in a manner that enables reliable device manufacturability and electromagnetic mode engineering control, and to ensure proper convergence of the optimization algorithm. Constraints of interest range from feature size and curvature tolerances to specification of the layout topology and connectivity. There are currently three primary classes of parameterization schemes used in freeform inverse design. One is pixel-based representations[28,29], in which the device is described as an ultra-fine grid of pixels, each with a distinctly tailored dielectric constant value. The second is level sets[30,31], in which layouts are implicitly represented as iso-contours of scalar functions[31]. The third is explicit geometric parameterization[32–34], in which simple analytic functions are defined to map latent variables to constrained geometric shapes.

While these parameterization concepts have been utilized effectively in certain design limits, each presents distinct trade-offs between optimization capability and constraints compliance. Pixel-based methods are the most straightforward to adapt to gradient-based optimizers such as the adjoint variables method. However, the use of spatial filters[35], density filters, and projection methods[36] to impose constraints lead to either poor fabrication compliance or the imposition of overly restrictive constraints. Traditional level set methods offer improved topology control, but they struggle with multi-scale features, require careful numerical schemes to maintain stability[37], and are limited in performance by their reliance on local shape derivatives[38]. Explicit parameterization schemes have the benefit of hard coding strict design constraints in the parameterization functions themselves, but they are ultimately overly restrictive and suffer from limited topology control. Qualitatively new geometric parameterization concepts are needed to generally and robustly enforce multiple competing objectives and constraints in the device, while maintaining access to the large freeform design landscape.

In this work, we present Neuroshaper, a neural network-based approach to freeform geometric representation that can streamline with existing gradient-based freeform optimizers. Neuroshaper is inspired by

[1]Department of Electrical Engineering, Stanford University, Stanford, CA, USA. [2]Department of Physics and Optical Science, University of North Carolina at Charlotte, Charlotte, NC, USA. ✉e-mail: jonfan@stanford.edu

recent work in computational imaging and computer graphics on neural representations[39,40] and specifically the development of multi-resolution techniques for representing complex signals[41]. Unlike traditional deep learning models that are trained on large datasets, Neuroshaper functions as a differentiable geometric parameterization scheme; its neural network parameters directly define the device geometry and are optimized using physics-based gradients and constraint penalties. It addresses limitations with existing parameterization schemes in photonics in multiple ways. First, it takes advantage of the immense expressivity of neural networks paired with learnable coordinate encoding to capture a wide range of multi-scalar freeform shapes. Second, it is differentiable and uses network loss function engineering and backpropagation to robustly and flexibly enforce constraints. Third, by capturing geometric device parameters using large numbers of neural network weights, Neuroshaper serves as an over-parameterized representation scheme that captures a smoothened, high-dimensional geometric design space. In this manner, gradient-based optimizers are able to more easily traverse the design space and detour around forbidden zones when constraints are encountered (Fig. 1a). An example of this optimization trajectory is included in Supplementary Movie 1. Fourth, Neuroshaper captures a global representation of geometric shapes, such that local geometric constraints map onto global modifications to the device layout. With these collective advantages, devices designed using Neuroshaper exhibit better handling and enforcement of constraints, and also similar or better performance, compared to conventionally parameterized freeform devices.

The full framework for Neuroshaper-enabled freeform inverse design is illustrated in Fig. 1b. Freeform layouts are produced using a novel neural architecture that combines multi-resolution hash encoding with multilayer perceptrons to generate continuous level set functions. These functions undergo level set operations to produce binary or grayscale patterns suitable for both constraint evaluation and simulation. The ability for Neuroshaper to support multi-resolution representations enables high resolution representations to be used for geometric constraint evaluation (M_1) and low resolution versions for electromagnetic simulations (M_2). Global loss functions accounting for feature size, curvature, and connectivity constraints are aggregated together with electromagnetic performance to properly balance these different competing objectives. All aspects of the algorithm are differentiable, enabling direct compatibility with auto-differentiation packages on GPU hardware and efficient gradient-based optimization of the complete system. The 'electromagnetic performance' objective function ($J(M_2)$) is flexible and can be tailored to various goals, including maximizing transmission/conversion efficiency (as shown in Fig. 5), minimizing reflection, or optimizing for phase-dependent characteristics (e.g., specific diffraction orders, wavefront shaping), provided the chosen simulation method allows for gradient computation with respect to the device geometry.

Results

Neuroshaper architecture
Schematics detailing the specification of nanophotonic geometries with Neuroshaper are presented in Fig. 2 and consist of two parts, level set encoding and level set translation to a dielectric distribution. We focus on the encoding of level sets as an intermediary to dielectric layouts, as opposed to the direct encoding of dielectric layouts, due to the uniquely flexible and expressive nature of level sets in topology optimization. To specify level sets, we propose the use of a neural network-encoded hash table description that combines discrete feature vectors with continuous neural networks and allows for both local and global control of the design space while maintaining differentiability (Fig. 2a).

More specifically, we first find the nearest lattices of the input coordinates (x, y) at every resolution level, represented as $\mathbf{x_{1,2,3,4}}$. In each level, the lattice point $\mathbf{x}$ is mapped to an entry in the lookup table by the hash function $h(\mathbf{x})$, which stores learnable feature vectors φ. These vectors from each hash table are then interpolated and concatenated to produce an encoded, high-dimensional feature vector, $\mathcal{H}_\varphi(x, y)$, which serves as an input to a multi-layer perceptron (MLP) that maps these features to scalar level sets. The neural network mapping is described by the transformation $\mathcal{N}_\omega$, where ω are weights of the network. Overall, the collective mapping of input coordinate to level set is specified by:

$$F_\theta(x, y) = \mathcal{N}_\omega(\mathcal{H}_\varphi(x, y)) \tag{1}$$

where $\theta = \varphi, \omega$ captures the hash table feature parameters and neural network parameters. It is important to note that this neural network serves as a *differentiable geometric representation* whose parameters θ are directly optimized based on simulation results and constraints, rather than a predictive model trained on a pre-existing dataset. These parameters are learned using autodifferentiation and gradient-based updating with tailored loss functions, which will be discussed later. Detailed mathematical derivations and implementation specifics are provided in "Methods" Section.

To translate the level set to a dielectric distribution (Fig. 2b), differentiable operations are applied, and the Neuroshaper platform is capable of readily generating a range of grayscale and binary dielectric representations. To produce a grayscale dielectric structure layout, $M_g(x, y)$, scaling and clipping operations are used. Binary dielectric structures, $M_b(x, y)$, are generated using direct thresholding.
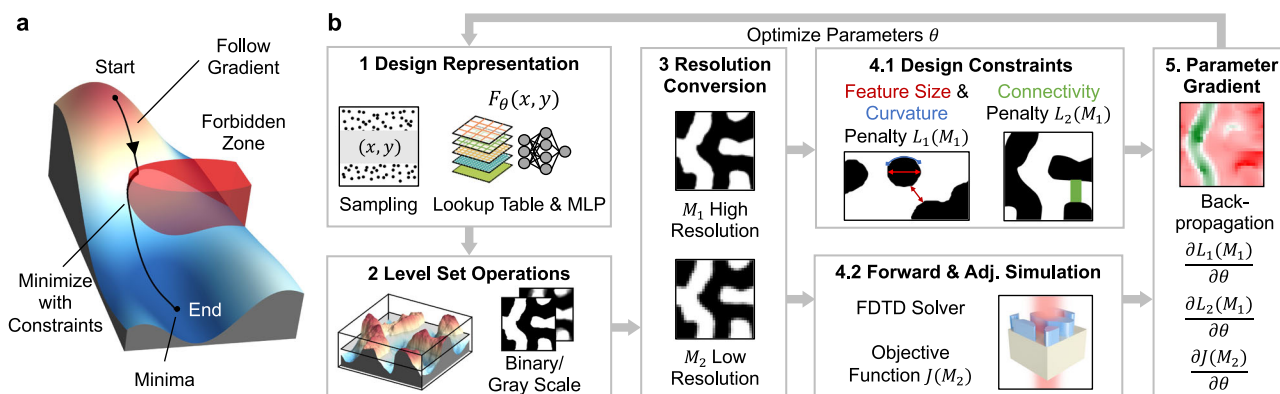
Fig. 1 | Neuroshaper overview. a Visualization of gradient-based optimization in the high dimensional Neuroshaper parameterization space. The optimizer avoids forbidden zones imposed by constraints to reach a suitable local minima. **b** Overview of the Neuroshaper framework for nanophotonic device design. The pipeline consists of: (1) Neural design representation combining sampling points with lookup tables and multilayer perceptrons to generate continuous level set functions; (2) Level set operations to obtain binary/grayscale patterns; (3) Multi-resolution conversion between high (M_1) and low (M_2) resolutions; (4) Design evaluation based on geometric constraints and electromagnetic performance; (5) Parameter gradient computation via backpropagation.

a **Functional Representation**

b **Level Set Operations**
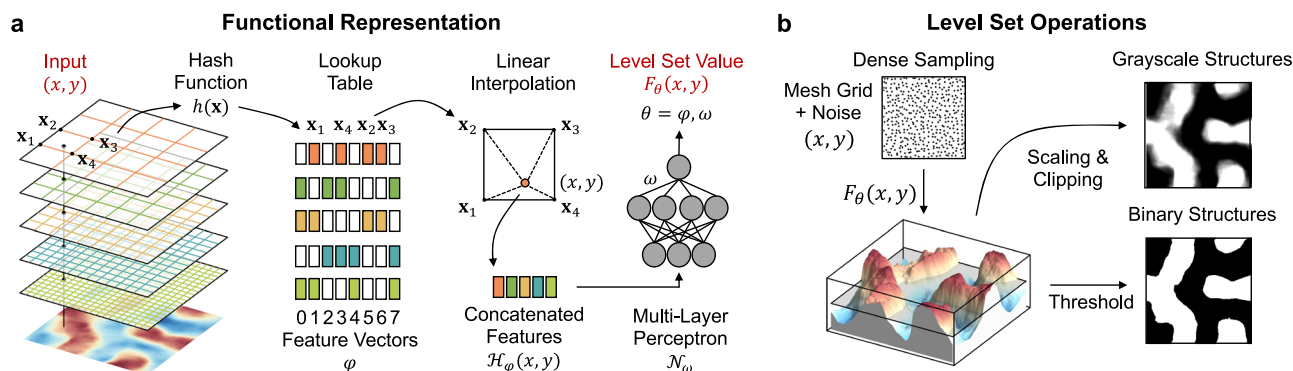
Fig. 2 | The neural level set representation model. a Functional representation showing the multi-resolution hash encoding process. Input coordinates (x, y) are mapped through parallel hash functions at different resolution levels, each accessing a dedicated lookup table of trainable feature vectors. Features are interpolated based on the relative position within grid cells and concatenated. The resulting feature vector is processed by a multi-layer perceptron (MLP) to produce the final level set function value $F_\theta(x, y)$. **b** Level set operations showing the conversion from continuous functions to binary and grayscale structures through dense sampling and thresholding operations.

Multi-resolution hash encoding addresses fundamental limitations posed by traditional level set methods. The hash encoding effectively acts as an adaptive grid, automatically allocating more representational capacity to regions with important features through collision-based feature sharing. As such, we achieve natural scale separation while maintaining differentiability. Our encoding concept also enables efficient parameter updates during optimization, as each input affects only a small subset of feature vectors and a small MLP network, but still produces both global and local changes in geometry due to the multiple resolution levels. In addition, the differentiability of our representation enables straightforward computation of gradients and curvature at each point.

Initialization and constraints

Neuroshaper features a comprehensive system of methods for enforcing constraints while maintaining optimization flexibility, and we provide an overview of these methods below. The methods can be used in different combinations depending on the problem with no restrictions. Mathematical and implementation details are provided in "Methods" Section.

Initialization: to initialize Neuroshaper, the parameters θ are randomly initialized from a normal distribution, and the corresponding level can serve as a random starting point for grayscale topology optimization. Neuroshaper can also be initialized to a desired shape using a two-stage process (Fig. 3a). Here, the level set for the desired initial shape is first generated as an approximate level set function using a distance transform. The Neuroshaper parameters are then optimized using an L1 loss function to approximate this initial level set. This approach provides controlled initialization while also introducing slight randomness that breaks the symmetry of the level set and associate gradient calculations.

Periodicity and reflection symmetry: Neuroshaper is capable of explicitly enforcing periodicity and reflection symmetry constraints by the implementation of a triangle wave coordinate mapping of the x and y coordinates prior to hash table encoding (Fig. 3b). Triangle waves are specifically chosen for their constant gradient magnitudes, which preserve the multi-resolution characteristics of our design representation. This is in contrast with sinusoidal mappings, whose varying gradients can distort the multi-resolution nature of the representation. Different types of triangle waves can be judiciously chosen to specify the reflection symmetry axes or to specify periodicity without any reflection symmetry.

Feature size and curvature constraints: feature size and curvature control are enforced through gradient-based boundary analysis (Fig. 3c). Given a minimum feature size d_m and a sampled boundary point $\mathbf{p}$ with a normal vector $\mathbf{n}$ into the void region, a loss function is defined such that all points from $\mathbf{p}$ to $\mathbf{p}_+ = \mathbf{p} + d_m\mathbf{n}$ are specified to have positive values and those from $\mathbf{p}$ to $\mathbf{p}_- = \mathbf{p} - d_m\mathbf{n}$ have negative values. We also require that the level set is similar to a signed distance function of the boundary, which

guarantees that after level set thresholding, all physical features in both material and void regions do not violate our minimum feature size constraint and are minimally modified each iteration. For curvature control, we leverage the differentiability of our shape representation to compute local curvature and add a loss function penalty term to discourage small values for local radii.

Connectivity: connectivity control is achieved by formulating connectivity as an auxiliary heat conduction problem (Fig. 3d) in which regions requiring connectivity are treated as heat sources while the target connection points act as heat sinks[42]. The steady-state heat equation defines the temperature distribution:

$$-\nabla \cdot (k\nabla u) = f \tag{2}$$

where u represents the temperature field, k is the thermal conductivity determined by the binary design, and f describes heat sources and sinks placed at desired connection points. Upon solving the heat equation, the resulting temperature distribution identifies disconnected regions through high temperature values, and the temperature gradient represents optimal paths for establishing connectivity and is utilized in the connectivity constraint loss function. The connectivity constraint can be used to control device topology, be specified to global or local features, and can be applied to both material and void regions.

Multi-resolution scaling: Neuroshaper can produce device layouts with different spatial resolutions (Fig. 3e), which is useful when considering that geometric constraint evaluation demands high-resolution structures (M_1) for accurate feature analysis while electromagnetic simulations can be efficiently and accurately performed at lower spatial resolutions (M_2).

To derive device layouts with different resolutions from the continuous function $F_\theta(x, y)$, we first sample this function at an extremely high base resolution M_b that far exceeds both the geometric analysis and simulation requirements. From this high-fidelity binary representation, we derive both the geometric analysis resolution M_1 and the simulation resolution M_2 through average pooling operations with different downsampling factors. This enables sub-pixel averaging, which is crucial for maintaining the accuracy of electromagnetic simulations as it provides effective material properties that are consistent with the finite difference formalism and that well approximate the structural layout.

Our approach additionally overcomes issues pertaining to the introduction of aliasing artifacts upon changes in resolution. Since our representation remains analytically continuous throughout the optimization process, resolution changes merely affect the sampling density rather than the underlying structure. The difference becomes particularly evident in designs requiring highly precise geometric features.
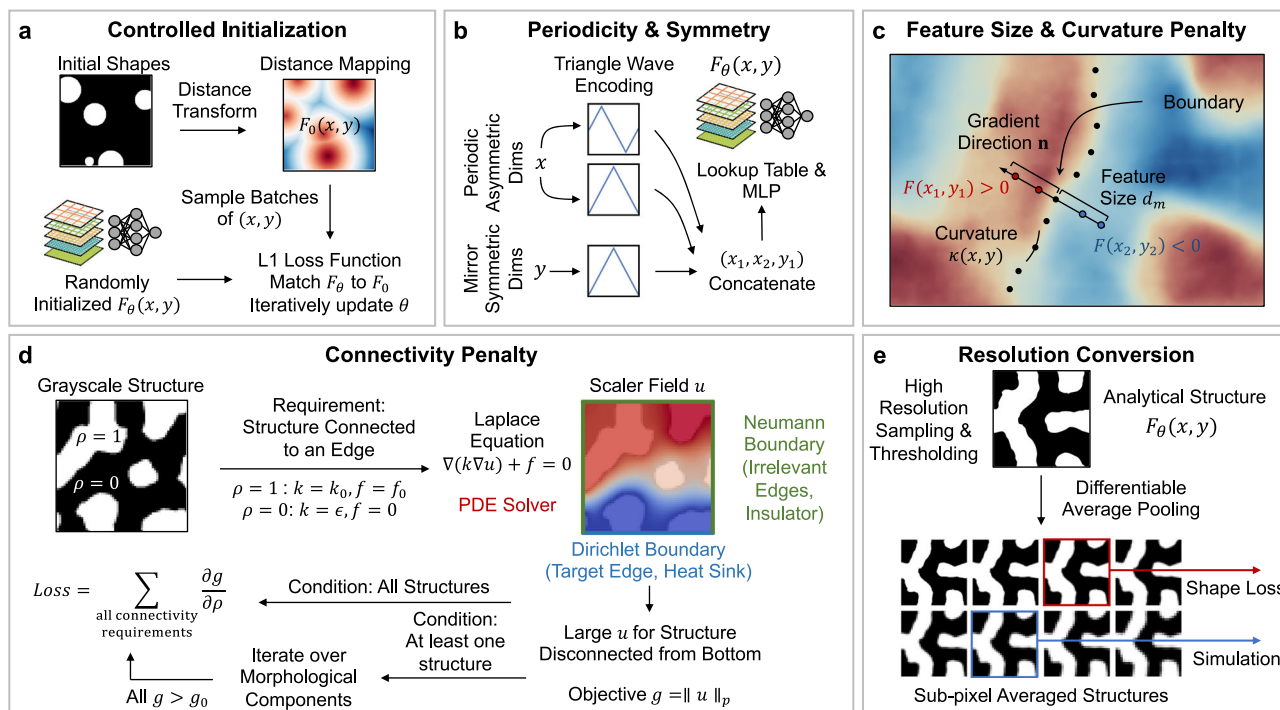
Fig. 3 | Implementation of initialization methods, constraints and resolution conversions. a Controlled initialization showing the transformation from initial shapes to distance mapping and the training process through randomly sampled batches. **b** Implementation of periodicity and symmetry constraints using triangle wave functions and coordinate mapping. **c** Feature size and curvature penalty calculation demonstrating gradient-based measurement of local geometric properties. **d** Connectivity penalty implementation using PDE-based approach with specified boundary conditions. **e** Resolution conversion process illustrating the relationship between high-resolution reference structures and differentiable sub-pixel averaged structures at different resolution for simulation.

Demonstrations

To demonstrate the different features of Neuroshaper and its ability to produce manufacturable nanophotonic devices, we design and experimentally implement freeform nonlocal metasurfaces serving as narrowband wavelength filters. Nonlocal metasurfaces are devices that support the coupling of free space light into guided mode resonances[43–45], enabling light-matter interactions that are inherently sensitive to wavelength and incidence angle. We focus our demonstration on near-infrared periodic silicon-based metasurfaces that feature zeroth-order nonlocal responses. To perform this optimization, we consider a modal framework for nonlocal optical phenomena in which the strong coupling of incident light to a vertically oriented magnetic dipole mode within a metasurface unit cell leads to a nonlocal response[46]. To explicitly enforce this mode behavior, we perform near-field optimization to maximize the magnitude of the H_z field component at a single point just above the metasurface, to avoid diverging values when performing adjoint gradient computation (Fig. 4a). For devices supporting multi-wavelength responses, multi-objective optimization is performed to specify modes at different wavelengths.

Details pertaining to our optimization implementation can be found in "Methods" section. To summarize, we consider periodic devices with a unit cell period of 500 nm and normally incident light with linear polarization along the x-axis. The adjoint variables method is used to perform near-field freeform optimization, and Tidy3D[47] is used as the solver. The iterative optimization process contains two parts (Fig. 4b). First, grayscale optimization is performed in which the continuous level set function is used to specify device layouts with grayscale dielectric values. This device parameterization enables a thorough exploration of the design space and gradual convergence toward promising topologies. Second, iterative optimization of layouts with binary dielectric values is performed to fine-tune these devices into final configuration.

Our optimization pipeline utilizes many of the constraints and topology control features in Neuroshaper to ensure the final devices support clean nonlocal behavior. Triangle wave encoding is used to enforce periodic boundary conditions along both in-plane axes. Minimum feature size and curvature constraints are implemented to limit the feature size and radii of curvature to 100 nm and 50 nm, respectively, which ensures that the device can be reliably fabricated. With the connectivity constraint, we specify the device topology to be isolated silicon island structures, which helps to suppress the presence of laterally distributed optical modes that can spoil the background. We note that our method for imposing this constraint is less restrictive than simply specifying the unit cell boundaries to be void, which excludes island structures that span across unit cell boundaries. We additionally specify the device to be only 120 nm thick, which is much smaller than the target wavelength and which suppresses the presence of higher order modes spanning the device height. The small thickness also enhances manufacturing feasibility.

Single and double-dip metasurface designs are shown in Fig. 4c, d. To fabricate these devices, we deposit a thin film of silicon onto a glass substrate by chemical vapor deposition, pattern a hydrogen silsesquioxane mask on the film using electron beam lithography, and transfer the pattern into the film using reactive ion etching. Electron microscopy images of both devices (Fig. 4e, f) reveal excellent fabrication fidelity and accurate matching with the design layouts. To characterize the devices, transmission spectra are obtained by polarizing and illuminating each device under normal incidence using a tuneable laser and scanning wavelength in 1 nm increments. The measured transmission spectra (Fig. 4g, h) show good agreement with simulations. The single dip device shows a sharp transmission dip near its design wavelength of $\lambda = 815$ nm and a clean background, while the double-dip device shows sharp transmission dips close to its design wavelengths of 800 nm and 860 nm. Additional details pertaining to the fabrication and characterization steps can be found in Section "Experimental setup".

Finally, we explore the ability and efficacy of Neuroshaper to broadly apply to the design of a wide range of free space and on-chip nanophotonic devices. A summary of device type, objective, added constraints, and result is presented in Fig. 5. Benchmark results based on conventional pixel-based adjoint-based freeform design are also presented, using conventional spatial
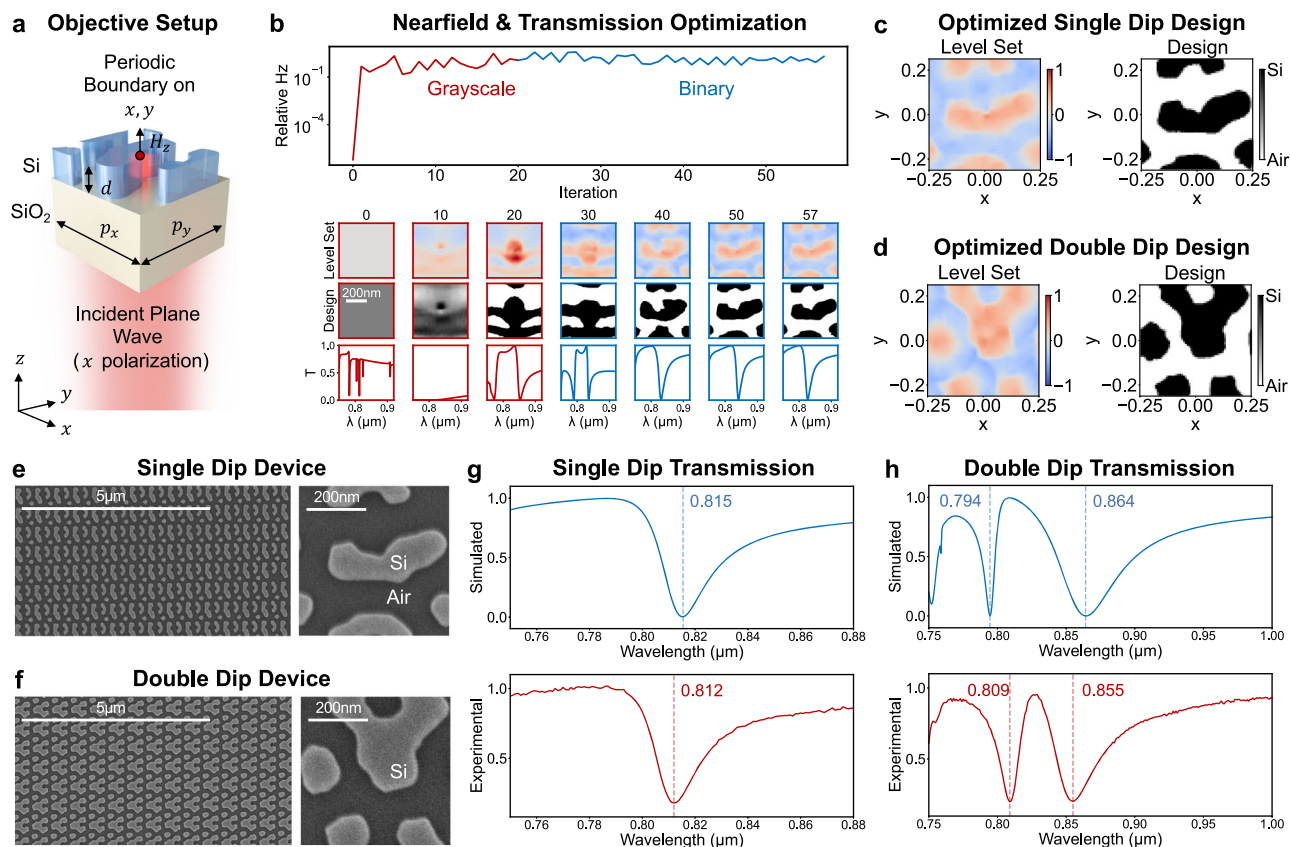
Fig. 4 | Optimization setup and experimental validation for Neuroshaper-designed nonlocal metasurfaces. a Schematic of the simulation setup showing a periodic silicon (Si) metasurface of thickness d on a silica (SiO$_2$) substrate. **b** Evolution of the optimization process showing grayscale (red) and binary (blue) design regimes. The lower panels show the level set function (top), corresponding physical designs (middle), and transmission spectra (bottom) at different iterations. **c, d** Level set function and binary silicon structure for the optimized **c** single-dip and double-dip design. SEM images of the fabricated **e** single-dip and **f** double-dip devices. Simulated (top) and experimental (bottom) transmission spectra for the **g** single-dip and **h** double-dip devices.

filtering techniques to enforce constraints. These devices use the same initial starting point as the Neuroshaper result to provide a direct comparison. These diverse benchmarks include common nanophotonic components: a 2D mode converter, bandpass filter, wavelength multiplexer, 3D grating coupler, and 3D plasmonic nanoantenna. Each was optimized with specific objectives (e.g., efficiency, contrast, field enhancement) and relevant constraints such as feature size, connectivity, and symmetry using Neuroshaper. For direct comparison, conventional pixel-based adjoint optimization using standard filtering techniques was performed starting from identical initial conditions. Detailed parameters for each case, including materials (typically Si/SiO2 or Au), dimensions, and specific constraints, are available in the "Methods" section.

Based on these results and benchmark comparisons, we observe the following. First, Neuroshaper-designed devices display more robust control over feature size and curvature. This control enforces the trade-off between geometric constraint and performance and better ensures that the devices can be fabricated. Second, for chip-based devices, the connectivity constraint enables seamless material transitions from waveguides to freeform structures, showing how non-trivial constraints can be enforced. For instance, while the Neuroshaper-designed 3D grating coupler (Fig. 5d) exhibits slightly lower peak efficiency (60.6%) compared to the conventional benchmark (70.9%), this is attributed to the enforcement of stricter manufacturability-focused constraints within our framework, including simultaneous adherence to waveguide connectivity, and boundary disconnection requirements (see "Methods" section for details on the 3D Grating Coupler benchmark). The conventional design shown achieves higher theoretical efficiency but visibly violates minimum feature size limits (note the penalty values in Fig. 5d: 0.02 for our result vs. 0.12 for the

conventional design) even without these stricter constraints. This highlights a key capability of Neuroshaper: balancing performance optimization with robust constraint satisfaction, which is crucial for practical device fabrication. Third, the Neuroshaper-designed devices generally have comparable or better performance than the conventional designed devices, even while featuring more total design constraints. To more systematically probe this observation, we perform an ensemble of 2D mode converter optimizations using both Neuroshaper and conventional methods, and we fix the optimization conditions (i.e., initial random dielectrics, feature size and curvature constraints, connectivity constraints) to be the same in both cases to enable a fair comparison. The resulting distributions of device efficiency are shown in Fig. 6 and show that on average the Neuroshaper-designed converters have better overall performance than conventionally designed devices. These data suggest that Neuroshaper's ability to account for constraints in a global manner can lead to an overall improved optimization result.

Discussion

In summary, we present Neuroshaper as a robust neural architecture representation framework for level set-based nanophotonic device optimization. Our method addresses several long-standing challenges in photonic freeform inverse design. First, it yields fully analytic, differentiable, and stable geometric representations in a manner that enables streamlining with gradient-based inverse design algorithms and accurate specification of device layouts at different scales and dielectric binarization conditions. Second, it introduces a modular approach to incorporating multiple constraints, including competing constraints. By specifying individual constraints in the form of loss functions, constraints involving feature size,
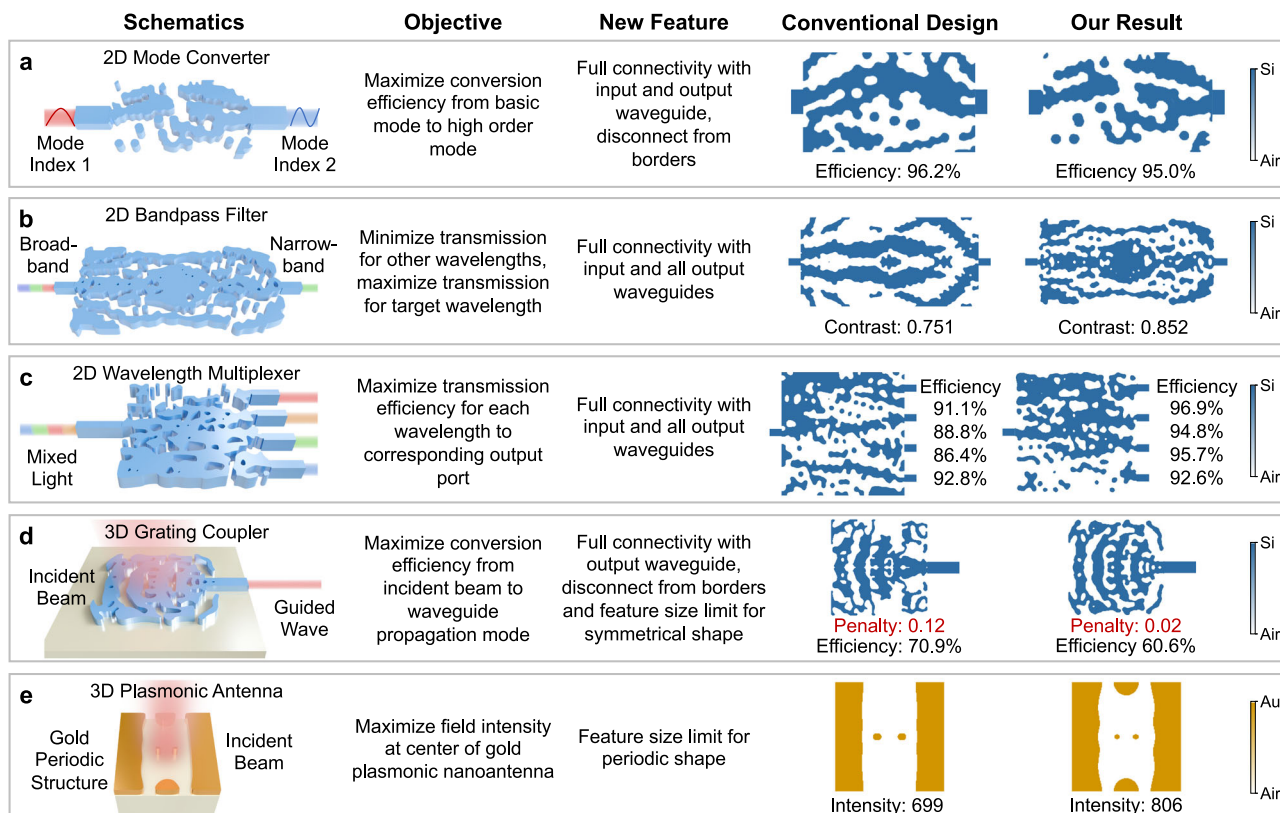
Fig. 5 | Application of the Neuroshaper framework to various nanophotonic devices. a 2D mode converter design demonstrating efficient conversion between different waveguide modes. The optimized structure maintains smooth connection with both input and output waveguides and is disconnected from the boundary boarders. **b** 2D bandpass filter showing improved contrast ratio compared to conventional approaches while maintaining structural connectivity across all waveguide interfaces. **c** 2D wavelength multiplexer design exhibiting superior transmission efficiencies compared to conventional designs while ensuring robust waveguide connectivity. **d** 3D grating coupler optimization incorporating multiple constraints: connectivity with the output waveguide, boundary disconnection, and symmetry-preserving feature size limits. The conventional design violates feature size limits. **e** 3D plasmonic nanoantenna optimization with periodic boundary conditions and feature size constraints, demonstrating enhanced field intensity compared to conventional designs. Simulated field distributions for these devices can be found in Supplementary Figs. 3 and 4.
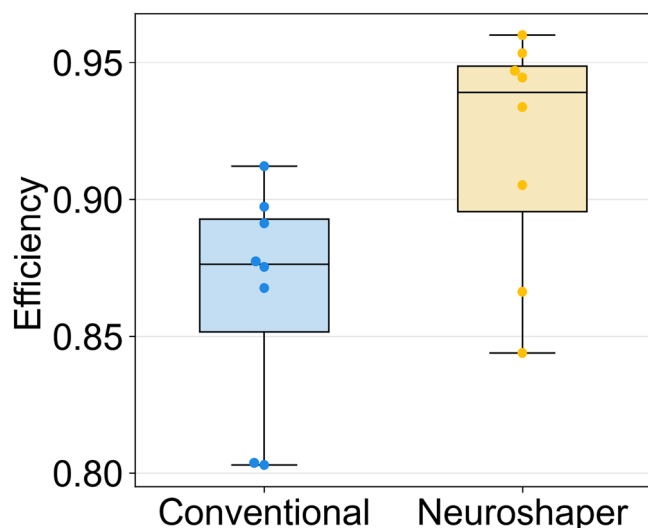
Fig. 6 | Comparison of efficiencies for randomly initialized 2D mode converter devices. The chart displays the resulting distributions of device efficiency for eight different optimizations using Neuroshaper and conventional pixel-based methods, designed under identical conditions including initial random dielectrics, feature size and curvature constraints.

feature curvature, and connectivity can be readily imposed by appending the desired loss functions to the global loss function for network training and updating. Third, Neuroshaper utilizes local constraints information to globally modify and evolve device layouts, providing a more robust and global framework for balancing the enforcement of constraints with device performance. Numerical experiments indicate that Neuroshaper consistently outperforms conventional pixel-based representations in constrained freeform design tasks, and an experimental proof-of-concept demonstration indicates the viability for freeform Neuroshaper-designed devices to be experimentally translated.

One potential limitation arises when applying Neuroshaper to very large design areas relative to the minimum feature size. The global nature of the neural representation, while advantageous for constraint handling, can sometimes lead to overly complex interdependencies across distant regions, potentially affecting optimization stability. Future work could explore architectural modifications or optimization strategies to mitigate this in large-scale problems. Looking forward, this work opens several promising directions for future research. We anticipate that Neuroshaper can apply to other classes of passive and active photonic devices, and more generally, to other domains of the physical sciences that leverage strong relationships between geometry and physical response. We also anticipate that Neuroshaper can be augmented with other neural network-based algorithms[48–51], such as high speed deep surrogate solvers[52–54], generative models[55–57], and large language model interfaces, to further streamline and accelerate the

computer aided design of physical infrastructure. We see potential for neuroparameterization to encode objects other than device geometries, such as electromagnetic fields or scattering profiles, which would expand the scope of how devices and their associated attributes are modeled. Finally, with the burgeoning growth of neurorepresentation research in the computer vision community, we anticipate a wide diversity of neural network-based representation models to be developed with practical use in the physical sciences.

Methods

Multi-resolution hash encoding
The hash encoding $\mathcal{H}\phi$ operates across L resolution levels, with each level maintaining a dedicated table of feature vectors[41]. For each level l, input coordinates are mapped to grid vertices through:

$$\mathbf{v}_l = \lfloor \mathbf{x} \cdot N_l \rfloor, \quad N_l = N_{\min} \cdot b^l \quad (3)$$

where N_l is the grid resolution at level l, $N_{\min}$ is the size of the most coarse grid, and b is a geometric progression factor. These integer coordinates are mapped to feature indices through a spatial hash function:

$$h_l(\mathbf{v}) = \left(\bigoplus_{i=1}^{d} \pi_i v_i \right) \mod T_l \quad (4)$$

where π_i are large prime numbers ($\pi_1 = 1$, $\pi_2 = 2{,}654{,}435{,}761$, $\pi_3 = 805{,}459{,}861$, $\pi_4 = 2{,}097{,}152$, $\pi_5 = 3{,}231{,}375$, $\pi_6 = 4{,}278{,}255{,}361$) and T_l is the hash table size at level l. This hash function effectively XORs the results of per-dimension linear congruential permutations, decorrelating the effect of dimensions on the hashed value. For better cache coherence, we set $\pi_1 = 1$, while the other prime numbers are chosen to be large and mutually coprime to minimize hash collisions in higher dimensions.

The encoding concatenates interpolated feature vectors from all levels:

$$\mathcal{H}_\phi(x, y) = [\, \text{Interp}\,(\phi_1[h_1(\mathbf{v})]), \ldots, \text{Interp}\,(\phi_L[h_L(\mathbf{v})]) \,] \quad (5)$$

Hash collision handling
A unique aspect of our encoding is its treatment of hash collisions. Rather than implementing explicit collision resolution through conventional methods like chaining or open addressing, our approach allows collisions to naturally occur and relies on the training process to handle them appropriately. When multiple spatial locations hash to the same feature vector, their gradients during optimization are effectively averaged. This creates an implicit form of importance sampling, where locations generating stronger gradients (typically corresponding to important features or boundaries) have a greater influence on the shared parameters. For coarse levels where $(N_l + 1)^d < T_l$, the mapping is one-to-one and no collisions occur. At finer levels, collisions become more frequent but are distributed pseudo-randomly across space. Since each input is encoded through multiple resolution levels simultaneously, it is statistically unlikely for two distant points to collide at all levels, maintaining the encoding's ability to distinguish distinct spatial locations. To produce a grayscale dielectric structure layout, $M_g(x, y)$, scaling and clipping operations are used:

$$M_g(x, y) = \min\{\max\{\alpha F_\theta(x, y) + \beta, 0\}, 1\} \quad (6)$$

α and β control the transition sharpness and bias.

Performance considerations
The multi-resolution structure provides several computational advantages:
- Parameter updates are sparse and efficient, as each input affects only a small subset of feature vectors at each level.
- The hash table structure allows for efficient GPU implementation with predictable memory access patterns.

- The geometric progression of resolution levels ensures coverage of all relevant spatial scales while maintaining a manageable parameter count.

The total number of parameters for the hash table is:

$$\text{Total Parameters} = L \cdot P \cdot T \quad (7)$$

where L is the number of levels, P is the feature dimension per level, and T is the hash table size per level. The hash table size T provides a direct trade-off between memory usage, computational cost, and reconstruction quality.

Gradient computation
The complete differentiability of our representation enables computation of important geometric properties. The gradient of the level set function is given by:

$$\nabla F_\theta(x, y) = \frac{\partial F_\theta}{\partial x} \mathbf{i} + \frac{\partial F_\theta}{\partial y} \mathbf{j} \quad (8)$$

The mean curvature can be computed as:

$$\kappa = \nabla \cdot \frac{\nabla F_\theta}{|\nabla F_\theta|} \quad (9)$$

These quantities are essential for implementing geometric constraints and ensuring fabrication-ready designs. The gradients are computed efficiently through automatic differentiation, propagating through both the neural network and the multi-resolution hash encoding.

Initialization implementation
The initialization process in our framework consists of two distinct stages that work together to provide both controlled starting points and diverse optimization trajectories. Given an initial shape defined by a binary mask $M_0(x, y)$, we first generate an approximate level set function through a signed distance transform:

$$F_0(x, y) = \text{DT}\,(M_0(x, y)) \quad (10)$$

This transform computes the signed Euclidean distance from each point to the nearest boundary of the initial shape, with positive values inside the shape and negative values outside. The distance transform is implemented using a fast marching method that ensures accuracy while maintaining computational efficiency.

The network parameters are then initialized using a truncated normal distribution:

$$\omega_i \sim \mathcal{N}(0, \sigma^2), \quad \sigma = \sqrt{\frac{2}{n_{\text{in}} + n_{\text{out}}}} \quad (11)$$

where n_{in} and n_{out} are the input and output dimensions of each layer respectively. This initialization scheme, known as Kaiming initialization, helps maintain consistent gradient magnitudes throughout the network during the early stages of training[58].

The optimization process minimizes the L1 loss between the neural network output and the initial level set function:

$$\mathcal{L}_{\text{init}} = \| F_\theta(x, y) - F_0(x, y) \|_1 \quad (12)$$

During this initialization phase, we employ stochastic sampling of points within the design domain to compute the loss:

$$\mathcal{L}_{\text{batch}} = \frac{1}{B} \sum_{i=1}^{B} \| F_\theta(x_i, y_i) - F_0(x_i, y_i) \|_1 \quad (13)$$

where B is the batch size (typically 65,536 in our implementation). This stochastic approach introduces beneficial noise during initialization while maintaining computational efficiency. The initialization process uses Adam optimization with learning rate = 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

We typically run the initialization optimization for 300 steps, which we found sufficient to capture the general structure of the initial shape while maintaining enough flexibility for subsequent optimization.

For pure random initialization, we skip the distance transform step and instead initialize the network parameters directly.

Periodicity and symmetry implementation

For periodic boundaries, we apply two triangle wave mappings and then concatenating the results:

$$x_{\text{periodic,1}} = 1 - 2\left|\left\{\frac{x}{p}\right\} - 0.5\right| \tag{14}$$

$$x_{\text{periodic,2}} = 1 - 2\left|\left\{\frac{x}{p} + 0.25\right\} - 0.5\right| \tag{15}$$

where p is the period length and $\cdot$ denotes the fractional part. The concatenated coordinates $[x_{\text{periodic,1}}, x_{\text{periodic,2}}]$ ensure periodicity while avoiding symmetry in the mapped domain.

For symmetry constraints, we apply a single triangle wave transformation:

$$x_{\text{symmetric}} = 1 - 2\left|\frac{x}{p} - 0.5\right| \tag{16}$$

These mappings maintain several crucial properties. The gradient magnitude of the triangle wave remains constant except at the vertices:

$$\left|\frac{d}{dx}x_{\text{periodic,i}}\right| = \frac{2}{p}, \quad \left|\frac{d}{dx}x_{\text{symmetric}}\right| = \frac{2}{p} \tag{17}$$

This uniform gradient preserves the multi-resolution characteristics of our design representation, unlike sinusoidal mappings which introduce varying gradients:

$$\left|\frac{d}{dx}\sin\left(\frac{2\pi x}{p}\right)\right| = \frac{2\pi}{p}\left|\cos\left(\frac{2\pi x}{p}\right)\right| \tag{18}$$

The complete coordinate transformation for a design requiring both periodicity and symmetry can be expressed as:

$$\mathbf{x}_{\text{transformed}} = \begin{bmatrix} x_{\text{periodic,1}} & x_{\text{periodic,2}} & y_{\text{symmetric}} \end{bmatrix} \tag{19}$$

This transformation preserves feature size constraints as they operate on the raw coordinates before mapping:

$$L_{\text{feature}}(F_\theta(\mathbf{x}_{\text{transformed}})) = L_{\text{feature}}(F_\theta(\mathbf{x})) \tag{20}$$

This ensures simultaneous satisfaction of periodicity, symmetry, and geometric requirements. The uniformity of the triangle wave gradients prevents edge artifacts while maintaining the effectiveness of our multi-resolution representation across the entire design domain.

Feature size and curvature constraint implementation

The feature size control implementation begins by identifying boundary points where the level set function crosses zero. For each boundary point $\mathbf{p}$, to numerically evaluate the penalty value, we employ Monte Carlo sampling with these 12 randomly distributed evaluation points along the normal direction for each of 65,536 boundary points. This stochastic sampling strategy provides robust feature size control while avoiding potential artifacts that could arise from regular sampling patterns. We compute a series of

randomly sampled evaluation points along the normal direction $\mathbf{n}$:

$$\mathbf{p}_i = \mathbf{p} + t_i d\mathbf{n}, \quad t_i \sim \mathcal{U}(-1, 1), \quad i = 0, \ldots, 11 \tag{21}$$

where d is the minimum feature size, $\mathcal{U}(-1, 1)$ denotes uniform random sampling in the interval $[-1, 1]$, and the normal direction is computed from the gradient of the level set function:

$$\mathbf{n} = \frac{\nabla F_\theta}{|\nabla F_\theta|} \tag{22}$$

The feature size constraint combines both minimum feature size enforcement and signed distance function maintenance across all evaluation points:

$$L_{\text{feature}}(F_\theta) = \sum_{p \in \mathcal{P}} \sum_{i=0}^{11} \Big[\text{ReLU}\left(-F_\theta(\mathbf{p}_i) \cdot \text{sign}(t_i)\right) \\ + \alpha(|F_\theta(\mathbf{p}_i)| - |t_i|d)^2 \Big] \tag{23}$$

where $\mathcal{P}$ represents the set of boundary points and α is a weighting parameter typically set to 0.1. The first term in the summation enforces the minimum feature size by ensuring proper sign of the level set function at each evaluation point, while the second term maintains the signed distance property by encouraging the function values to match the expected distance at each evaluation point.

The curvature at boundary points is computed directly from the level set function:

$$\kappa = \nabla \cdot \frac{\nabla F_\theta}{|\nabla F_\theta|} \tag{24}$$

The curvature penalty is formulated as a direct sum over boundary points:

$$L_{\text{curve}}(F_\theta) = \sum_{p \in \mathcal{P}} \kappa(p) \tag{25}$$

The complete geometric constraint combines both feature size and curvature terms:

$$L_{\text{geometric}} = L_{\text{feature}} + \lambda L_{\text{curve}} \tag{26}$$

where λ is typically set between 0.01–0.1 depending on the specific application requirements. This random sampling approach ensures robust feature size control while avoiding systematic biases that might arise from regular sampling patterns.

Heat equation formulation for connectivity

Our framework implements connectivity constraints through a heat equation-based approach that enables precise control over structural connectivity while maintaining compatibility with gradient-based optimization[42]. The method can enforce multiple simultaneous connectivity requirements between arbitrary points or boundaries in the design space, making it particularly valuable for creating fabrication-ready metasurface designs.

The heat equation governing the connectivity constraint is formulated as:

$$-\nabla \cdot (k\nabla u) = f \tag{27}$$

where u represents the scalar temperature field, k denotes the spatial conductivity distribution, and f represents the source/sink distribution. The connectivity constraint g is evaluated using a normalized p-norm of the temperature field:

$$g = \left(\frac{\| u \|_p}{\text{thresh}}\right) - 1 \tag{28}$$

where $\| u \|_p = \left(\sum_i |u_i|^p \right)^{1/p}$ is the p-norm of u.

The framework enables sophisticated connectivity control through several key mechanisms:

First, boundary conditions can be flexibly specified by setting appropriate source (f) and conductivity (k) values at desired locations. Any point or edge in the design can be designated as a connection target through these boundary conditions. The conductivity field k is directly derived from the design variables ρ, with high conductivity in material regions (k_s) and low conductivity in void regions (k_v).

Second, multiple connectivity requirements can be enforced simultaneously by solving multiple heat equations with different boundary conditions. For instance, to ensure connectivity between multiple waveguide interfaces or to maintain connections to multiple boundaries, separate temperature fields are computed and combined into a joint constraint. The framework automatically handles the interaction between these multiple constraints through gradient computation.

Third, the method incorporates morphological analysis to adaptively apply connectivity constraints. Before computing the heat equation solution, the current design is analyzed to identify disconnected components. If existing connections already satisfy the requirements, the constraint gradients are zeroed in those regions, focusing the optimization on areas where connectivity needs to be established or improved.

To compute the gradient $\frac{\partial g}{\partial \rho}$ for optimization, we employ the adjoint method:

$$\frac{\partial g}{\partial \rho} = \frac{\partial g}{\partial u} \frac{\partial u}{\partial \rho} + \frac{\partial g}{\partial k} \frac{\partial k}{\partial \rho} + \frac{\partial g}{\partial f} \frac{\partial f}{\partial \rho} \tag{29}$$

We introduce an adjoint variable λ satisfying:

$$-\nabla \cdot (k\nabla\lambda) = \frac{\partial g}{\partial u} \tag{30}$$

The final gradient expression becomes:

$$\frac{\partial g}{\partial \rho} = -\lambda^T \left((k_s - k_v)\nabla u \cdot \nabla u + (f_s - f_v) \right) \tag{31}$$

where $k_s - k_v$ represents the difference between solid and void conductivities, and $f_s - f_v$ denotes the difference between solid and void source terms.

The numerical implementation employs finite differences for spatial derivatives and supports both direct and iterative solvers for the resulting linear systems. The heat equation is discretized using central differences, and the conductivity at material interfaces is computed using harmonic averaging to ensure numerical stability. This approach provides smooth, well-behaved gradients suitable for optimization while accurately capturing connectivity requirements.

This formulation has proven particularly effective for metasurface design, where maintaining proper connectivity is crucial for both fabrication and optical performance. The method successfully ensures that optimized structures maintain continuous paths between specified boundaries or points while allowing the topology to evolve freely in other regions. This capability is demonstrated in our results, where complex metasurface designs consistently exhibit proper connectivity to waveguide interfaces and maintain specified periodic boundary conditions.

Resolution conversion and sub-pixel averaging details
Starting from our continuous level set function $F_\theta(x, y)$, we implement a three-stage resolution conversion process. The first stage involves high-resolution sampling, where we sample the continuous function at an extremely high base resolution M_b that significantly exceeds both geometric

analysis and simulation requirements:

$$M_b(i, j) = \begin{cases} 1 & \text{if } F_\theta(x_i, y_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

where (x_i, y_j) represents the spatial coordinates of each high-resolution pixel.

In the second stage, we derive both the geometric analysis resolution M_1 and simulation resolution M_2 through average pooling operations from this high-fidelity binary representation:

$$M_k(i, j) = \frac{1}{n_k^2} \sum_{p=0}^{n_k-1} \sum_{q=0}^{n_k-1} M_b(n_k i + p, n_k j + q) \tag{33}$$

Here, $k \in 1, 2$ denotes the resolution level and n_k represents the downsampling factor for level k, with M_1 used for geometric constraint evaluation and M_2 used for electromagnetic simulation.

The third stage involves effective medium approximation for electromagnetic simulation, where the averaged values are interpreted as effective medium properties:

$$\epsilon_{\text{eff}}(i, j) = \epsilon_1 M_2(i, j) + \epsilon_2(1 - M_2(i, j)) \tag{34}$$

In this equation, $\epsilon_{\text{eff}}(i, j)$ represents the effective permittivity at position (i, j), while ϵ_1 and ϵ_2 are the permittivities of the constituent materials, and $M_2(i, j)$ represents the material fraction from average pooling.

The sub-pixel averaging approach provides several critical advantages for electromagnetic simulation. First, it offers improved accuracy by providing a more accurate representation of material boundaries than binary discretization, and this effective medium approximation also minimizes artificial resonances and scattering that can arise from staircased boundaries in binary discretization:

$$\text{Staircasing Error} \propto \left| \frac{\partial \epsilon}{\partial x} \right|_{\text{discrete}} - \left| \frac{\partial \epsilon}{\partial x} \right|_{\text{continuous}} \tag{35}$$

Our framework also supports dynamic resolution adaptation during the optimization process. The base resolution can be adjusted throughout the optimization according to:

$$M_b^{\text{iter}}(i, j) = M_b(i \cdot s^{\text{iter}}, j \cdot s^{\text{iter}}) \tag{36}$$

where s^{iter} is a scaling factor that evolves throughout the optimization:

$$s^{\text{iter}} = s_0 \cdot \alpha^{\lfloor \text{iter}/N \rfloor} \tag{37}$$

This approach enables rapid exploration of the design space in early iterations using lower resolutions, followed by progressive refinement of geometric details as optimization converges, ultimately leading to high-resolution optimization for fabrication-ready designs.

Hyperparameters of optimization pipeline for nonlocal metasurface
The optimization was performed using PyTorch (v2.2) on a single NVIDIA RTX A6000 GPU. Each optimization step, including constraint evaluation and backpropagation through the Neuroshaper network, typically completed in under 2 s, representing minimal overhead compared to the FDTD simulation time, which takes roughly 2 min. The hash encoding is implemented with a hierarchical structure consisting of $L = 10$ levels, each encoding features of dimensionality $P = 4$. The base resolution begins at $N_{\text{min}} = 4$ and scales geometrically between consecutive levels by a factor of 1.5. Each level maintains an independent hash table of size 2^{10} entries. The complete encoding transforms 2D spatial coordinates into a high-dimensional feature space through parallel lookup operations at multiple resolutions.

The neural network architecture comprises a sequence of fully connected layers with dimensions [128, 64, 64], employing LeakyReLU activation functions with $\alpha = 0.1$ between layers. All linear transformations are implemented without bias terms. The network maps from the hash-encoded feature space to a scalar level set value, maintaining end-to-end differentiability throughout the optimization process. The algorithm identifies boundary points through iterative optimization of $N = 65{,}536$ sample points. The optimization process alternates between gradient descent on the level set function and point resampling. At each iteration, points are categorized based on their squared SDF values, with the update strategy retaining 50% of the best-performing points, incorporating 25% from regular grid sampling, and introducing 25% new random samples. Feature size control is implemented through our gradient-based sampling around boundary points. The maximum SDF value is constrained to 0.5. The feature size is considered 0.2 normalized units, for 100 nm feature size in a 500 nm device. We set $\lambda = 0.1$ for the curvature penalty.

The nonlocal metasurface simulation employs a uniform grid with 60 points per unit cell. The physical dimensions correspond to a period of 0.5 µm in both directions, with the structure thickness set to 0.12 µm. Material properties are defined by refractive indices $n_{Si} = 3.75$ for silicon and $n_{SiO2} = 1.45$ for the substrate. The simulation domain is terminated with perfectly matched layers (PML) of thickness 1.0 µm, and excitation is provided at normal incidence ($\theta = 0°$) with the H_z field component monitored for optimization. The framework maintains two distinct resolution levels: a high-resolution representation of 240×240 pixels for geometric constraint evaluation, and a lower resolution 60×60 grid for FDTD simulation. Conversion between resolutions is accomplished through area averaging, maintaining differentiability throughout the optimization process. We did not change the resolution over the optimization process as in the current size it already runs sufficiently fast.

The optimization employs the Adam algorithm with initial learning rates of $\eta = 0.001$ for optimization. Batch sizes are set to 65,536 for geometric constraint evaluation and derived from the simulation resolution for physics optimization. The implementation maintains pure gradient-based updates without additional regularization terms.

Experimental setup

For fabrication of the device, a 120 nm thick layer of hydrogenated amorphous silicon (a-Si:H) is deposited on a fused silica substrate of 500 µm using plasma-enhanced chemical vapor deposition (PECVD, PlasmaTherm CCP-Dep) at a chamber temperature of 200 °C. Next, Surpass 3000 is spun on the wafer to promote adhesion of the e-beam resist 3% hydrogen silsesquioxane (HSQ) and the latter is then spun and baked. Next, a charging dissipation solution (Electra 92) is spun. The sample is then exposed using e-beam lithography and developed in the 25% TMAH developer, while the Electra 92 layer is solved in water. The metasurface pattern is then transferred to the a-Si:H film through dry etching (Oxford III-V Etcher) using a mixture of hydrogen bromide (HBr) and chlorine (Cl_2) gases. The etch quality is controlled by adjusting the chamber pressure and forward power of the RF and ICP generators in the inductively-coupled plasma reactive-ion etching (ICP-RIE) system.

The experimental setup for measuring the transmission spectra of the nonlocal metasurfaces is depicted in Fig. 7. A collimated beam from a supercontinuum white light laser (NKT Photonics, SuperK EXTREME EXW-12) was used as the light source, with a tunable bandpass filter (NKT Photonics, SuperK LLTF) applied at the output to select specific wavelengths within the target range of 750 nm to 1000 nm. The incident beam was linearly polarized along the designed polarization of the metasurface using a polarizer and then focused through a 40 mm focal length lens to shrink the beam spot to match the dimension of the metasurface. The sample was mounted on an x-y translational stage to ensure precise alignment of the beam with individual devices. The transmitted power was measured using a power meter (Thorlabs PM100D) placed behind the sample along the optical axis. The experiment involved two sequential measurements: the transmitted power through the bare substrate was first recorded as a
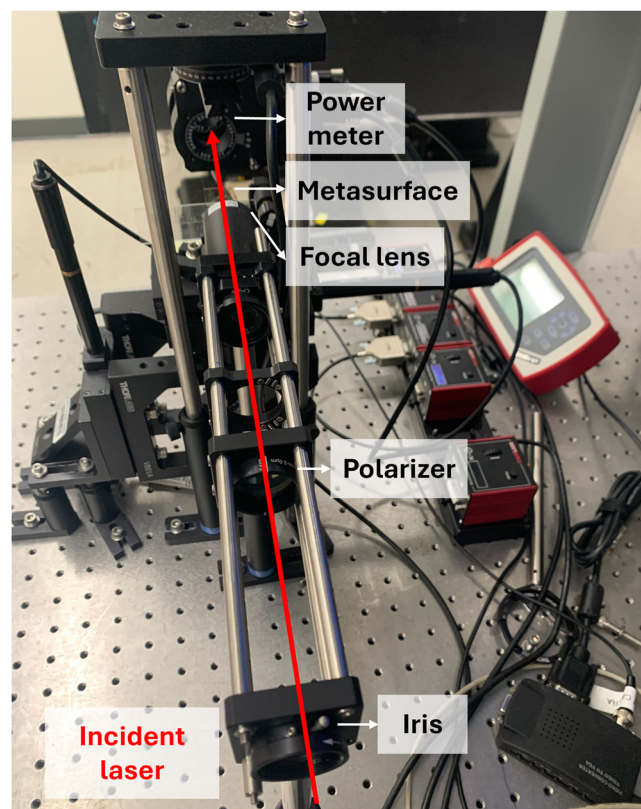
Fig. 7 | Photograph of the experimental transmission spectra collection setup. The image shows the arrangement of optical components, including a super-continuum white light laser, tunable bandpass filter (iris), polarizer, focusing lens, sample stage, and power meter, used for characterizing the fabricated metasurfaces.

reference, then the beam was positioned through the metasurface, where the transmission spectrum of the metasurface is calculated as the ratio of the transmitted power through the metasurface to the reference measurement. This setup provides a robust and accurate method to characterize the transmission performance of the metasurface across a broad wavelength range.

Details for representative nanophotonic problems

Here we provide detailed setup information for the five benchmark problems used to demonstrate Neuroshaper's capabilities across different application domains. The key parameters for the 2D problems are summarized in Table 1, and for the 3D problems in Table 2.

For each benchmark, the design region is discretized according to specified resolution, and grid resolution satisfies minimum 20 pixels per wavelength in all materials. Except for periodic dimensions, the boundary conditions are perfectly matched layers (PML). For conventional optimization, feature size constraints are enforced using erosion-dilation penalty with conic filter, and the topology optimization is performed using Adam optimizer with learning rate 0.01. A specific beta parameter for projection is ramped from 1 to 30 over optimization, which projects the parameters using the following relation.

$$f(x; \beta) = \frac{\tanh\left(\frac{\beta}{2}\right) + \tanh\left(\beta\left(x - \frac{1}{2}\right)\right)}{2\tanh\left(\frac{\beta}{2}\right)} \tag{38}$$

For comparison of the two methods, we reduced the y design region from 3 µm to 2 µm and removed the constraints regarding connectivity to enable a fair comparison.

Table 1 | Detailed setup information for the 2D benchmark nanophotonic problems

Parameter	Mode converter	Bandpass filter	Wavelength mux
General design parameters			
Wavelength(s)	1.0 μm	0.8–1.2 μm	1.27, 1.29, 1.31, 1.33 μm
Design region size	5 × 3 μm² (Eval)	10 × 5 μm²	4.5 × 4.5 μm²
	5 × 2 μm² (Ens)		
Resolution	10 nm	15 nm	15 nm
Min. feature size	240 nm	240 nm	200 nm
Symmetry	None	None	None
Performance metric	Mode conv. eff.	Trans. contrast	Avg. trans. eff.
Field visualization	E_z	H_z	E_z
Material properties (relative permittivity, ϵ_r)			
Background ϵ_r	1.0	1.0	1.0
Waveguide core ϵ_r	2.75^2	2.0^2	3.49^2
Waveguide/port parameters			
Input Wg. width	0.7 μm	0.4 μm	0.3 μm
Output Wg. width	Same as input	0.4 μm	0.3 μm
Output Wg. spacing	–	–	0.8 μm
Input mode	Fund. TE	Fund. TM	Fund. TE
Output mode	2nd order TE	Fund. TM	Fund. TE
Other device-specific parameters			
Bandwidth	–	100 nm	–

"Eval" refers to initial evaluation and "Ens" to ensemble comparison.

Table 2 | Detailed setup information for the 3D benchmark nanophotonic problems

Parameter	3D grating coupler	3D plasmonic nanoantenna
General design parameters		
Wavelength	1.55 μm	910 nm
Design region footprint	4 × 4 μm²	500 × 500 nm²
Design layer material	Si	Au
Design layer thickness	220 nm (Si)	50 nm (Au)
Resolution	20 nm	3 nm
Min. feature size	160 nm	48 nm
Symmetry	Mirror (xz-plane)	Mirror (xz, yz-planes)
Performance metric	Coupling eff.	Max. field enhancement
Material properties (relative permittivity, ϵ_r)		
Background ϵ_r	1.0	1.0
Design layer ϵ_r	3.48^2 (Si)	From J&C (Au)
Substrate material ϵ_r	1.44^2 (Oxide)	1.44^2 (SiO₂)
Substrate thickness	1.6 μm	728 nm
Device-specific waveguide/optical parameters		
Input waveguide width	0.5 μm (Si wg.)	–
Output mode	Fiber mode	–
Spot diameter	2.5 μm	–
Beam tilt angle	10°	–

"J&C" refers to Johnson and Christy data for gold permittivity.

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

Code availability
The custom code and scripts used for generation and analysis of data in this study are available from the corresponding author upon reasonable request. Information on software versions includes Python (v3.10), PyTorch (v2.2), Tidy3D (v2.7). Specific variables and parameters are detailed in the "Methods" section.

References
1. Yu, N. & Capasso, F. Flat optics with designer metasurfaces. *Nat. Mater.* **13**, 139–150 (2014).
2. Kamali, S. M., Arbabi, E., Arbabi, A. & Faraon, A. A review of dielectric optical metasurfaces for wavefront control. *Nanophotonics* **7**, 1041–1068 (2018).
3. Shrestha, S., Overvig, A. C., Lu, M., Stein, A. & Yu, N. Broadband achromatic dielectric metalenses. *Light Sci. Appl.* **7**, 85 (2018).
4. Mohammadi Estakhri, N., Edwards, B. & Engheta, N. Inverse-designed metastructures that solve equations. *Science* **363**, 1333–1338 (2019).
5. Guo, C., Wang, H. & Fan, S. Squeeze free space with nonlocal flat optics. *Optica* **7**, 1133–1138 (2020).
6. Chen, A. & Monticone, F. Dielectric nonlocal metasurfaces for fully solid-state ultrathin optical systems. *ACS Photonics* **8**, 1439–1447 (2021).
7. Hua, X. et al. Ultra-compact snapshot spectral light-field imaging. *Nat. Commun.* **13**, 2732 (2022).
8. Zou, X. et al. Pixel-level bayer-type colour router based on metasurfaces. *Nat. Commun.* **13**, 3288 (2022).
9. Zaidi, A., Rubin, N.A., Meretska, M.L. et al. Metasurface-enabled single-shot and complete Mueller matrix imaging. *Nat. Photon.* **18**, 704–712 (2024).
10. Du, Z. et al. Ultracompact and multifunctional integrated photonic platform. *Sci. Adv.* **10**, eadm7569 (2024).
11. Yu, N. et al. Light propagation with phase discontinuities: generalized laws of reflection and refraction. *Science* **334**, 333–337 (2011).
12. 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).
13. Koshelev, K., Lepeshov, S., Liu, M., Bogdanov, A. & Kivshar, Y. Asymmetric metasurfaces with high-q resonances governed by bound states in the continuum. *Phys. Rev. Lett.* **121**, 193903 (2018).
14. Rubin, N. A. et al. Matrix fourier optics enables a compact full-stokes polarization camera. *Science* **365**, eaax1839 (2019).
15. Chen, W. T., Zhu, A. Y., Sisler, J., Bharwani, Z. & Capasso, F. A broadband achromatic polarization-insensitive metalens consisting of anisotropic nanostructures. *Nat. Commun.* **10**, 355 (2019).
16. Overvig, A. C., Malek, S. C. & Yu, N. Multifunctional nonlocal metasurfaces. *Phys. Rev. Lett.* **125**, 017402 (2020).
17. Whiting, E. B., Campbell, S. D., Kang, L. & Werner, D. H. Meta-atom library generation via an efficient multi-objective shape optimization method. *Opt. Express* **28**, 24229–24242 (2020).
18. Chen, M., Jiang, J. & Fan, J. A. Algorithm-driven paradigms for freeform optical engineering. *ACS Photonics* **9**, 2860–2871 (2022).
19. Shai Gertler et al. Many photonic design problems are sparse QCQPs. *Sci. Adv.* **11**, eadl3237 (2025).

20. Chen, M. et al. Validation and characterization of algorithms and software for photonics inverse design. *J. Opt. Soc. Am. B* **41**, A161–A176 (2024).

21. Lalau-Keraly, C. M., Bhargava, S., Miller, O. D. & Yablonovitch, E. Adjoint shape optimization applied to electromagnetic design. *Opt. Express* **21**, 21693–21701 (2013).

22. Sell, D., Yang, J., Doshay, S., Yang, R. & Fan, J. A. Large-angle, multifunctional metagratings based on freeform multimode geometries. *Nano Lett.* **17**, 3752–3757 (2017).

23. Yang, J. & Fan, J. A. Topology-optimized metasurfaces: impact of initial geometric layout. *Opt. Lett.* **42**, 3161–3164 (2017).

24. Yang, J. & Fan, J. A. Analysis of material selection on dielectric metasurface performance. *Opt. Express* **25**, 23899–23909 (2017).

25. Yang, J., Sell, D. & Fan, J. A. Freeform metagratings based on complex light scattering dynamics for extreme, high efficiency beam steering. *Ann. der Phys.* **530**, 1700302 (2018).

26. Chung, H. & Miller, O. D. High-na achromatic metalenses by inverse design. *Opt. Express* **28**, 6945–6965 (2020).

27. Shao, Y. et al. Multifunctional spaceplates for optical aberration correction. *ACS Photonics* **11**, 1753–1760 (2024).

28. Jensen, J. & Sigmund, O. Topology optimization for nano-photonics. *Laser Photonics Rev.* **5**, 308–321 (2011).

29. Piggott, A. Y. et al. Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. *Nat. Photonics* **9**, 374–377 (2015).

30. Osher, S. J. & Sethian, J. A. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988).

31. Wang, M. Y., Wang, X. & Guo, D. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Eng.* **192**, 227–246 (2003).

32. Chen, M., Jiang, J. & Fan, J. A. Design space reparameterization enforces hard geometric constraints in inverse-designed nanophotonic devices. *ACS Photonics* **7**, 3141–3151 (2020).

33. Gershnabel, E. et al. Reparameterization approach to gradient-based inverse design of three-dimensional nanophotonic devices. *ACS Photonics* **10**, 815–823 (2023).

34. Zhou, Y., Shao, Y., Mao, C. & Fan, J. A. Inverse-designed metasurfaces with facile fabrication parameters. *J. Opt.* **26**, 055101 (2024).

35. Sigmund, O. Morphology-based black and white filters for topology optimization. *Struct. Multidiscip. Optim.* **33**, 401–424 (2007).

36. Guest, J. K., Prévost, J. H. & Belytschko, T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int. J. Numer. Methods Eng.* **61**, 238–254 (2004).

37. Osher, S. & Fedkiw, R. P. Level set methods: an overview and some recent results. *J. Comput. Phys.* **169**, 463–502 (2001).

38. Allaire, G., Jouve, F. & Toader, A.-M. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.* **194**, 363–393 (2004).

39. Mildenhall, B. et al. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **65**, 99–106 (2021).

40. Park, J. J., Florence, P. R., Straub, J., Newcombe, R. A. & Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 165–174 (IEEE Computer Society / Computer Vision Foundation, 2019).

41. Müller, T., Evans, A., Schied, C. & Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph* **41**, 1–15 (2022).

42. Li, Q., Chen, W., Liu, S. & Tong, L. Structural topology optimization considering connectivity constraint. *Struct. Multidiscip. Optim.* **54**, 971–984 (2016).

43. Zhou, Y., Guo, S., Overvig, A. C. & Alù, A. Multiresonant nonlocal metasurfaces. *Nano Lett.* **23**, 6768–6775 (2023).

44. Kwon, H., Sounas, D., Cordaro, A., Polman, A. & Alù, A. Nonlocal metasurfaces for optical signal processing. *Phys. Rev. Lett.* **121**, 173004 (2018).

45. Shastri, K. & Monticone, F. Nonlocal flat optics. *Nat. Photonics* **17**, 36–47 (2023).

46. Zhou, Y., Shao, Y., Dai, T., Mao, C. & Fan, J. Freeform high-q metasurfaces enabled by tailored optical nearfields. In *CLEO: Fundamental Science*, FM3L–2 (Optica Publishing Group, 2024).

47. Flexcompute Inc. Tidy3d: Fast electromagnetic solver (fdtd) at scale. (2021).

48. Ma, W. et al. Deep learning for the design of photonic structures. *Nat. Photonics* **15**, 77–90 (2021).

49. Campbell, S. D., Jenkins, R. P., O'Connor, P. J. & Werner, D. The explosion of artificial intelligence in antennas and propagation: How deep learning is advancing our state of the art. *IEEE Antennas Propag. Mag.* **63**, 16–27 (2021).

50. Wang, Z. et al. Metasurface on integrated photonic platform: from mode converters to machine learning. *Nanophotonics* **11**, 3531–3546 (2022).

51. Tahersima, M. H. et al. Deep neural network inverse design of integrated photonic power splitters. *Sci. Rep.* **9**, 1368 (2019).

52. Wiecha, P. R. & Muskens, O. L. Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures. *Nano Lett.* **20**, 329–338 (2020).

53. Chen, M. et al. High speed simulation and freeform optimization of nanophotonic devices with physics-augmented deep learning. *ACS Photonics* **9**, 3110–3123 (2022).

54. Mao, C., Lupoiu, R., Dai, T., Chen, M. & Fan, J. Towards general neural surrogate solvers with specialized neural accelerators. In *Forty-first International Conference on Machine Learning*, 34693–34711 (Proceedings of Machine Learning Research, 2024).

55. Jiang, J. et al. Free-form diffractive metagrating design based on generative adversarial networks. *ACS Nano* **13**, 8872–8878 (2019).

56. Jiang, J. & Fan, J. A. Global optimization of dielectric metasurfaces using a physics-driven neural network. *Nano Lett.* **19**, 5366–5372 (2019).

57. Ma, W. et al. Pushing the limits of functionality-multiplexing capability in metasurface design based on statistical machine learning. *Adv. Mater.* **34**, 2110022 (2022).

58. Kaiming, H. et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In *Proc. IEEE International Conference on Computer Vision*, 1026–1034 (IEEE, 2015).

Acknowledgements

Author contributions

T.D., Y.S., Y.Z. and J.F. developed the idea, designed experiments, and wrote the manuscript. T.D. designed and implemented the core algorithm, performed optical simulations, processed the results and prepared Figs. 1–6. Y.S., C.M. and S.A. performed the sample preparation. Y.W. built up the optical system and performed the optical measurement. All authors have read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information