

---

# BACK-PROPAGATION OPTIMIZATION AND MULTI-VALUED ARTIFICIAL NEURAL NETWORKS FOR HIGHLY VIVID STRUCTURAL COLOR FILTER METASURFACES

---

**Arthur Clini de Souza**

Université Côte d'Azur, Inria, CNRS, LJAD, 06902 Sophia Antipolis Cedex, France  
 Laboratory of Applied and Computational Electromagnetism (LEMAC),  
 School of Electrical and Computer Engineering (FEEC),  
 University of Campinas (UNICAMP), Campinas, São Paulo, Brazil  
 Solnil, 95 Rue de la République, 13002 Marseille, France

**Stéphane Lanteri**

Université Côte d'Azur, Inria, CNRS, LJAD,  
 06902 Sophia Antipolis Cedex, France

**Hugo Enrique Hernandez-Figueroa**

Laboratory of Applied and Computational Electromagnetism (LEMAC),  
 School of Electrical and Computer Engineering (FEEC),  
 University of Campinas (UNICAMP), Campinas, São Paulo, Brazil

**Marco Abbarchi**

Solnil, 95 Rue de la République,  
 13002 Marseille, France  
 Université Aix Marseille, CNRS,  
 Université de Toulon, IM2NP, UMR 7334,  
 F-13397 Marseille, France

**David Gross**

Solnil, 95 Rue de la République,  
 13002 Marseille, France  
 Université Aix Marseille, CNRS,  
 Université de Toulon, IM2NP, UMR 7334,  
 F-13397 Marseille, France

**Badre Kerzabi**

Solnil, 95 Rue de la République,  
 13002 Marseille, France

**Mahmoud Elsawy\***

Université Côte d'Azur, Inria, CNRS, LJAD,  
 06902 Sophia Antipolis Cedex, France  
 mahmoud.elsawy@inria.fr

## ABSTRACT

We introduce a novel technique for designing color filter metasurfaces using a data-driven approach based on deep learning. Our innovative approach employs inverse design principles to identify highly efficient designs that outperform all the configurations in the dataset, which consists of 585 distinct geometries solely. By combining Multi-Valued Artificial Neural Networks and back-propagation optimization, we overcome the limitations of previous approaches, such as poor performance due to extrapolation and undesired local minima. Consequently, we successfully create reliable and highly efficient configurations for metasurface color filters capable of producing exceptionally vivid colors that go beyond the sRGB gamut. Furthermore, our deep learning technique can be extended to design various pixellated metasurface configurations with different functionalities.

## Introduction

Optical color filters are structures or materials designed to discriminate and manipulate distinct light wavelengths through the selective transmission or reflection of particular colors while simultaneously absorbing or attenuating undesired colors [1, 2]. Conventional color filters rely on the manipulation of chemical composition to achieve the desired optical properties, which can lead to issues such as absorption losses, thermal effects, and alterations in chemical characteristics [3]. An alternative approach involves the utilisation of structural color filters, offering distinct advantages and applications in diverse fields such as photorealistic color printing, color holography, anti-counterfeiting devices, and much more [4, 5, 6].

Metasurfaces have emerged as a promising platform for structural color filters [7, 8], owing to its peculiar capability of controlling all the light properties at the nanoscale, enabling a plethora of applications [9, 10, 11, 12]. Dielectric metasurfaces play a crucial role in color filter applications, especially within the visible spectrum range where the plasmonic counterpart based on metals is less performing owing to intrinsic optical losses. The limited losses of dielectrics (e.g.  $\text{Si}_3\text{N}_4$ ,  $\text{GaN}$ ,  $\text{TiO}_2$ ,  $\text{ZrO}_2$ ,  $\text{HfO}_2$ ) make them highly desirable for designing efficient devices with sharp resonance responses [13, 14, 15, 16]. Resonant dielectric metasurfaces achieve precise control over the phase of reflected and transmitted light by leveraging various resonant phenomena (e.g. Mie resonances) [17, 18]. Through meticulous engineering of the resonators, selective interaction with different wavelengths is enabled, leading to efficient and vivid color filters. Such kind of metasurfaces offer exceptional phase control, high-quality factors, and sharp resonances, resulting in enhanced color purity and spectral selectivity [19, 20, 21]. Yet, the design of an ideal color filter demands capability to selectively filter all colors across the optical spectrum. In other words, at each desired wavelength, it is crucial to eliminate any background resonances in order to achieve a pure color response characterized by sharp reflection or transmission amplitudes. Given the fabrication constraints, finding the appropriate resonator shape to achieve a desired response, is a challenging task that has garnered significant attention in the research community. Numerous studies explored this area, employing sophisticated optimization algorithms including advanced Deep Learning (DL) approaches to tackle the inherent complexity of the problem [21, 22, 23, 24, 25, 26, 20, 27, 28, 29]. However, relying on classical optimization approaches requires several costly simulations when optimizing various color targets simultaneously [30, 31, 32]. A viable solution for the design of vivid metasurface color filters is one-shot optimization using Artificial Neural Network (ANN). However, it is not straightforward owing to the presence of several designs with similar optical response whereas regular ANN has only a single output [33].

Here we present a novel data-driven methodology for efficiently designing fabrication constrained color filter metasurfaces. Our approach combines the ability to find suitable designs of Multi-Valued Artificial Neural Networks (MVANNs) with the solution refinement of back-propagation optimization. Thereby overcoming the fundamental limitations of relying on either the latter, which will lead to an undesirable local minima, or solely on a MVANN leading to poor performance associated with extrapolation [33, 23, 24]. In our case, relying solely on 585 simulations and by varying four parameters allows for the optimization of a continuous spectrum of objectives and the identification of highly vivid metasurface color filters. The optimized geometries exhibit a single sharp resonance response representing all the colors across the visible regime. To the best of our knowledge, our research outcomes exceed the previous findings documented in the literature, positioning our color filter as the foremost advancement in terms of vividness and overall performance [34, 35, 36, 8, 29].

## Geometry and surrogate model

Figure 1 represents the considered metasurface geometry composed of slanted ridges of titanium dioxide ( $\text{TiO}_2$ ) on top of a silicon dioxide ( $\text{SiO}_2$ ) substrate with refractive index  $n_s = 1.45$ . The refractive index of  $\text{TiO}_2$  is determined through ellipsometry (available in the supplementary information section). The inclusion of slanted gratings in the metasurface design introduces additional degrees of freedom, enabling the appearance of sharp resonances in the reflection spectrum. This is achieved by breaking the symmetry in the z-direction, resulting in high-quality resonance modes [37, 38]. Four parameters, namely the metasurface period ( $P$ ), resonator height ( $H$ ), base width ( $W$ ), and the ratio between the top and base widths of the resonator ( $S$ ), are optimized in this study.

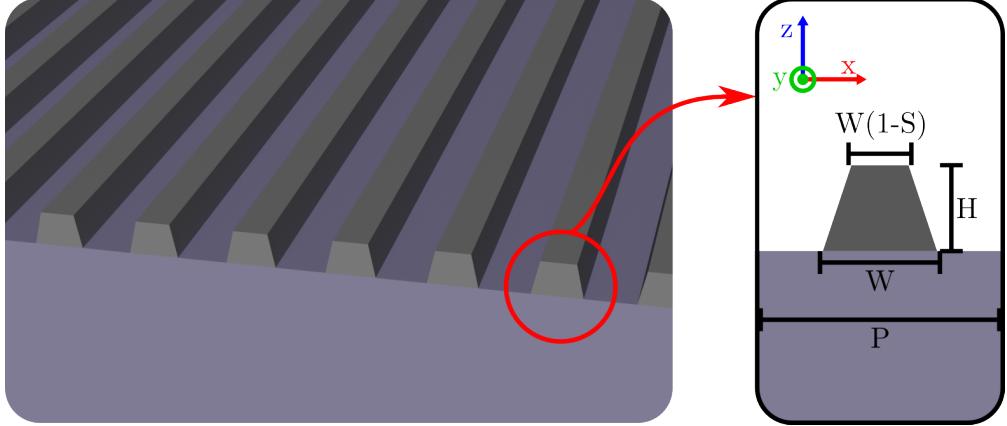


Figure 1: Schematic representation of the simulated structure. The light is injected from top with normal incidence with electric field polarized along the x-direction. The inset refers to the geometrical parameters associated to the single unit-cell surrounded with periodic boundary conditions.

The first step to train the ANN is to generate a dataset. One important aspect that we considered to build it is the fabrication constraint that imposes a maximum aspect ratio of 2 between the height and width of the resonators. The dataset consists of 585 simulations generated by uniformly sweeping the four parameters indicated in Figure 1. The period  $P$  ranges from 250 nm and 510 nm. For the height  $H$ , the range spans from 100 nm to 425 nm. In order to ensure that the ratios  $H/W$  and  $H/(P-W)$  remain less than 2, the width  $W$  varies between  $H/2$  and  $(2P-H)/2$ , and therefore, the height  $H$  must obey the inequality  $H/2 < (2P-H)/2$ . Additionally, the parameter  $S$  ranges from 0 to 0.8, with intervals of 0.1. The simulations were performed on an Intel® Xeon® W-2125 Processor operating at 4.0 GHz and allocating 4 threads, taking a total of 33 minutes and 40 seconds. The description of the 2D Python MEEP Finite-Difference Time-Domain (FDTD) simulations are described in the first section in the supplementary information.

Subsequently, a feedforward ANN surrogate model was trained to forecast the reflection spectral response, considering the resonator's geometry as input. In this study, we leverage the surrogate model for two distinct purposes. Firstly, it enables rapid estimation of the reflection spectrum, achieving significant computational speed-ups compared to full wave FDTD simulations. Secondly, the surrogate model is crucial in the inverse design process, playing a critical role in computing optimal solutions. As a result, the surrogate model's performance is of paramount importance, as it must deliver exceptional precision to provide the most accurate approximations possible. By fulfilling these requirements, the surrogate model accuracy significantly contributes to the effectiveness and success of this study. Our surrogate model is a classical Multilayer Perceptron (MLP) model with fully connected layers. In this scenario, there are several ways to configure this MLP, and we will compare two different strategies to map the input geometries into a spectrum.

The first approach uses the geometric parameters as inputs and generates a multi-dimensional vector as output. Each dimension of this vector corresponds to a distinct wavelength in the reflection spectrum. The second approach involves incorporating the wavelength as an input parameter while outputting a single dimension representing the reflection amplitude at that particular wavelength (Figure 2). Although both methods initially seems to be identical, the second is more convenient for this study. By incorporating the wavelength as an input, each point in the spectrum becomes a unique data point for the surrogate model. In contrast, the first method contains only 585 data points available for training the ANN. Conversely, the second method significantly enhances the availability of data samples, yielding a total of 292,500 individual data points resulting from sampling the spectrum across 500 different wavelengths.

Figure 2 represents the architecture of the surrogate model. In general, we consider the Gaussian Error Linear Unit (GELU) activation function which is a gating activation with a continuous derivative. Considering GELU implies the continuity of the output and its derivative, providing smoother response compared to the classical Rectified Linear Unit (ReLU) activation [39]. It is worth mentioning that the last layer contains a single linear neuron to compute a singular output dimension aligning with the reflection observed at the designated wavelength as indicated by the red arrow in Figure 2.

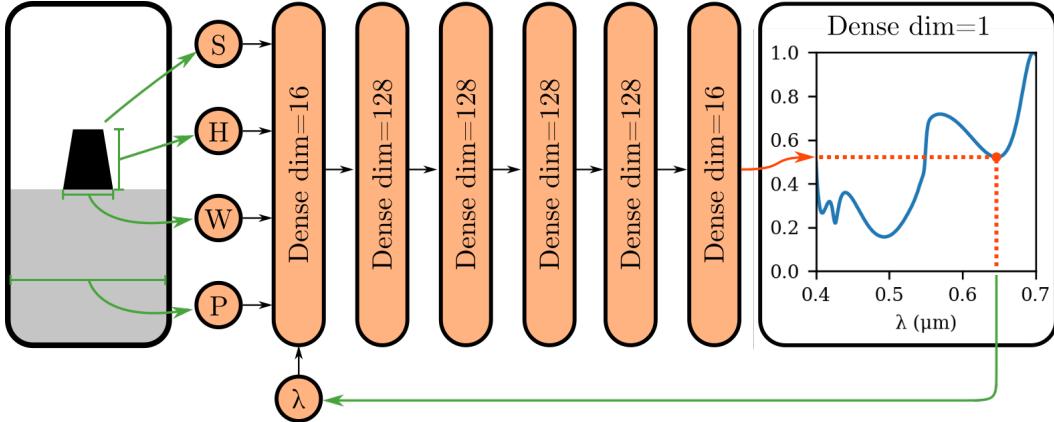


Figure 2: The diagram represents the architecture of the surrogate model ANN. The green arrows represent the input parameters. The surrogate model is designed to calculate a single output dimension, which corresponds to the reflection at the desired wavelength as shown by the red arrow. On the left side, a typical example of the resonator's geometry is provided, while on the right side, an example spectrum is displayed. It is important to note that these examples are purely for illustrative purposes and do not represent any real simulations. All the layers in this architecture are initialized using Glorot normal initializers as discussed in Ref. [40]. The model was trained using the TensorFlow framework, employing a holdout methodology [41]. The dataset is divided into two subsets: 555 simulations to update the weights of the ANN and 30 simulations for validation. The optimizer of choice was Adam with MSE loss and batch size of 512 [42]. The training stopped at epoch 82 due to the lack of improvement of the validation loss within the last 20 epochs. The final MSE loss was  $2.6996 \times 10^{-4}$  and validation loss,  $3.2745 \times 10^{-4}$ . The graph showing the loss evolution at each epoch can be found in section 2 of the supplementary information document. The total training time was 9 min 48s on a Google colab's CPU.

Figure 3 illustrates the generalization capacity of the surrogate model. Despite being trained on a relatively small dataset, the model is able to reproduce the spectra associated to the geometries that lie beyond the training dataset. This accomplishment can be attributed to the integration of wavelength as an input variable and the meticulous training methodology employed to mitigate overfitting. The demonstrated efficiency of this approach underscores the potential for constructing dependable surrogate models even when confronted with restricted training data.

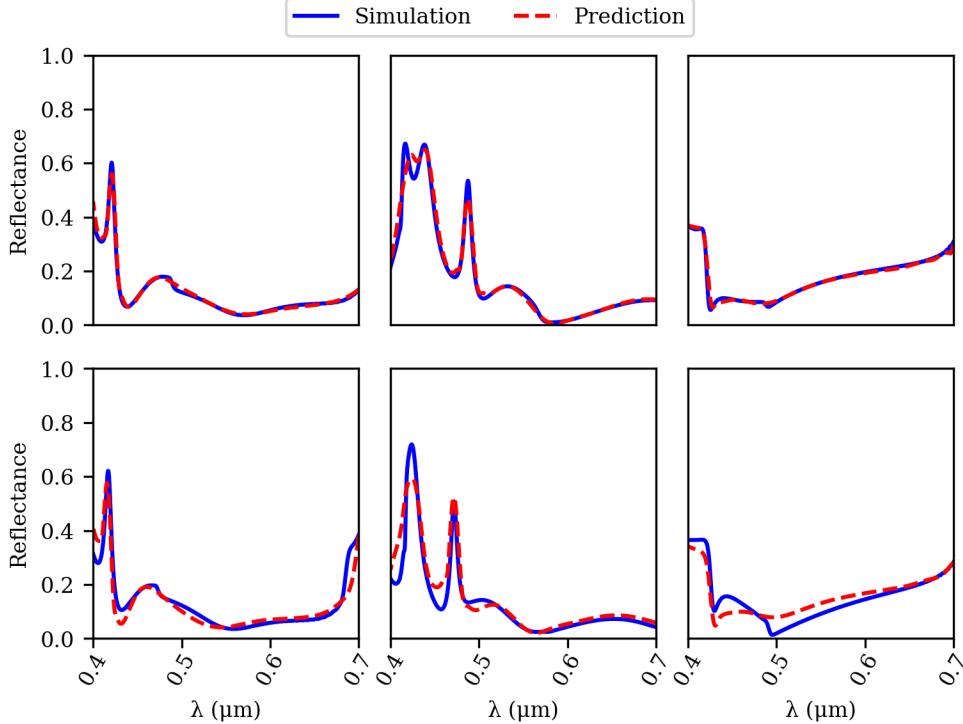


Figure 3: Comparison between fullwave simulation and the prediction from the surrogate model. Top figures indicate the reflection spectra of three randomly selected samples from the dataset. The bottom figures corresponding to the same top geometries while introducing a small random deviation to the geometrical parameters. Notably, the surrogate model demonstrates robust generalization capabilities despite the inherent sparsity of the dataset.

## Inverse Design

In this particular section, an inverse design methodology is formulated for metasurface color filter. The primary objective is to determine the corresponding set of parameters based on the desired spectrum response. To initiate the process, the target line shape spectrum must be established, and in our research, a Lorentzian function shape resembling a Fano resonance-like shape is employed:

$$L(f) = \frac{\omega^2}{\omega^2 + 4(f - f_0)^2}, \quad (1)$$

where  $\omega$  represents the full width at half maximum (FWHM) while  $f_0$  corresponds to the central frequency. The units employed are in accordance with MEEP's configuration, where both frequency and FWHM are measured in  $\mu\text{m}^{-1}$ . Additionally, it is worth noting that throughout the optimization process, the FWHM remains unchanged for all targets. This means that the Quality Factor (QF) of the desired spectra undergoes variations as a function of  $f_0$ , in accordance with Equation 2.

$$QF := \frac{f_0}{\omega} \quad (2)$$

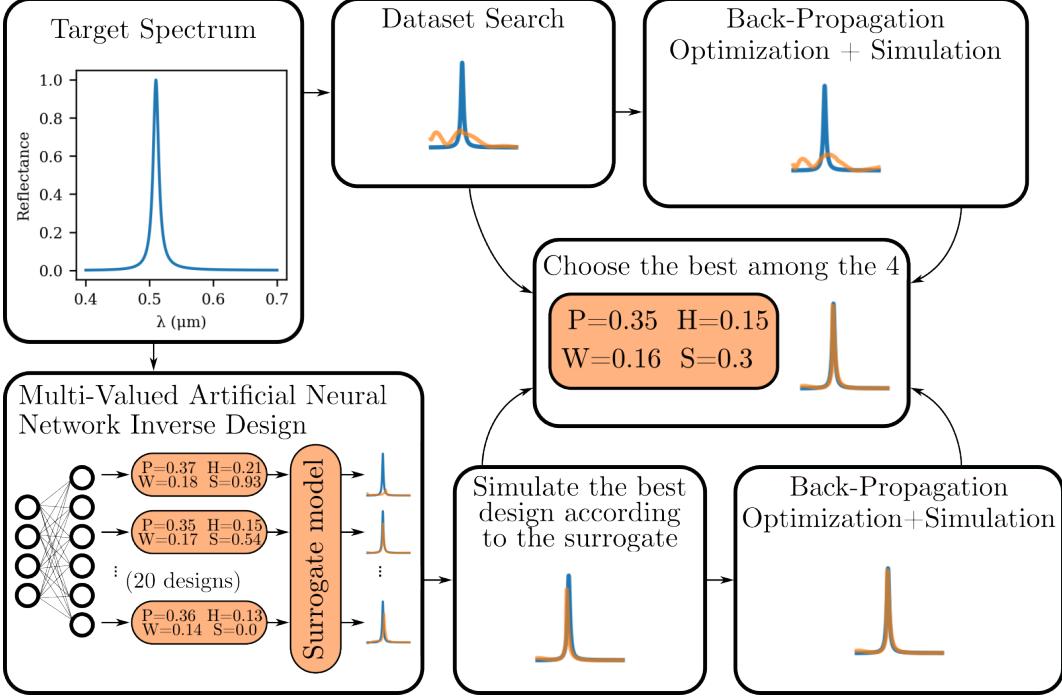


Figure 4: Optimization diagram for the inverse design methodology, where the target reflection spectrum is specified in the cyan curve. To achieve the desired outcome, two complementary paths are identified. The first path involves exploiting the dataset scan and searching for the best design that will be subjected to a back-propagation optimization to fine-tune the parameters, as indicated in the top blocks. On the other hand, the second path involves passing the target through the MVANN to obtain 20 designs. These designs are then validated using the surrogate model, and the solution that presented the least MSE compared to the target is chosen. Finally, a back-propagation optimization is performed on that design. Ultimately, the best solution among the four selected designs from the two paths is chosen. Further details can be found in the text.

The optimization scenario illustrated in Figure 4 depicts the inverse design process. The desired spectrum is represented by the cyan curve in the top left corner. In this stage, we employ two complementary approaches for inverse design. The first approach involves a straightforward search through a dataset, represented by the arrow pointing to the right of the target spectrum. For each simulation in the dataset, we calculate the MSE loss between the target spectrum and the simulated spectrum. By doing so, we can identify the design that produces the lowest MSE. Although this design is the best within the dataset, it fails to accurately match the desired response, as indicated in Figure 5. To address this limitation, we utilize a fully differentiable surrogate model and perform gradient optimization to finely adjust the parameters. As it can be seen in Figure 5 running the back-propagation yields interesting designs. However, still this approach is not able to produce vivid colors along all the visible regime. Further the resonances are mainly associated with higher order modes appearing at shorter wavelengths (see last column) that allows for non-pure color. The optimized solution, along with the best design found within the dataset, are both stored for later comparison. A detailed explanation of the methodology used to perform back-propagation is displayed in section 3 of the supplementary information.

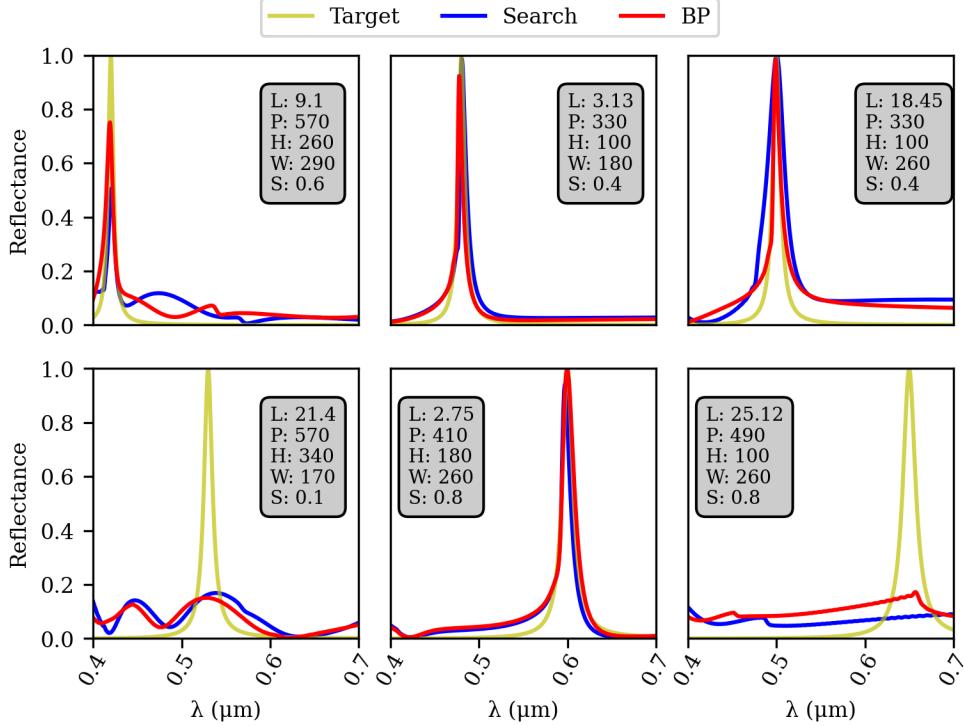


Figure 5: Results from the dataset search using a Lorentzian lineshape in frequency as target for the dataset search for the wavelengths of 420 nm, 480 nm, 500 nm, 530 nm, 650 nm, 650 nm with  $\omega = 0.01$  from left to right, top to bottom. The metric used to evaluate the results was the MSE (L) in  $10^{-3}$  between the target spectrum and the closest one in the dataset, while the geometric parameters (P, H, W and S) of the device found on the search were described in Figure 1. Blue and red curves represent the best design without and with back-propagation optimization on best design found in the dataset. A comparison of two different resonance shapes with different  $\omega$  values are also tested in figure S5 provided in the supplementary information.

The second complementary approach, indicated by the downward arrow from the target spectrum in Figure 4, involves the utilization of a MVANN to generate robust solutions for the inverse design problem. A MVANN is essentially an ANN capable of producing multiple solutions based on a single input. In the context of our study, the adoption of the MVANN proves highly advantageous as it effectively addresses the challenge of multiple parameter responses corresponding to a given objective target [33]. To ensure reasonable accuracy, we choose to employ a model that yields 20 outputs. The detailed architecture and training procedure of the MVANN are provided in the supplementary information section 5. After acquiring the set of 20 distinct designs, a validation process is conducted using the surrogate model. Among these designs, the one that exhibits the lowest MSE loss when compared to the target is initially stored for future comparison. To further improve the results, this best design is subjected to back-propagation optimization. By employing a combination of the dataset search method indicated by the top blocks in Figure 4 and the MVANN approach, followed by back-propagation indicated by the bottom blocks in Figure 4, we successfully identify four robust designs. This entire process, encompassing both approaches, takes approximately 100 seconds to complete. Finally, the design exhibiting the lowest MSE loss compared to the target is selected as the final choice. We refer to section 6 in the supplementary information for more details regarding the performance of MVANN as a function of the dataset size. Further, a detailed comparison between MVANN and a state-of-the art algorithm can be found in section 7 in the supplementary information.

By employing the aforementioned inverse design methodology illustrated in Figure 4, we perform a total of 100 optimization iterations, as depicted in Figure 6 (a). These iterations involve sweeping the desired wavelength across the range of 400 nm to 700 nm, thereby encompassing the entire spectrum of colors. The target function is defined as a Lorentzian spectrum with a Full Width at Half Maximum (FWHM) value of  $\omega = 0.01$ . Subsequently, we calculate the perceived color values ( $x$  and  $y$ ) on the chromaticity diagram using the CIE 1931 2° standard observer color matching functions and the "Equal energy" illuminant, which ensures that incident light of all wavelengths possesses uniform power [43]. As depicted Figure 6 (a), we successful optimized metasurface designs that generate colors closely

resembling our target (gray curve). Consequently, a significant number of these optimized designs fall outside the sRGB zone, represented by the dashed triangle in Figure 6 (a), expanding the color gamut achievable with metasurfaces beyond the limitations of conventional display technologies.

In Figure 6 (b), we select six distinct designs to demonstrate the spectral response of our optimized configurations, as indicated by the black arrows in Figure 6 (a). Our inverse design approach has led to metasurface designs that exhibit a reflection spectrum response that is consistent with the target Lorentzian spectrum across all desired wavelengths. Moreover, the background response in the spectrum is nearly flat, providing a vibrant color response for our optimized configurations. A typical example of the field profile is presented in the supplementary information. Additionally, a numerical comparison between FDTD calculations and COMSOL has been included to verify our results. Furthermore, we have conducted a full-wave simulation with finite structure to identify the number of required periods to retrieve the results of a single unitcell simulation with periodic boundary conditions (for more details, the reader can refer to the supplementary information). To the best of our knowledge, the optimized metasurface designs yield the most vivid color filter reported so far in the literature [34, 35, 36, 8, 29]. It is worth mentioning that the entire optimization process took 2 hours, 37 minutes, and 45 seconds. Among this time, 1 hour, 26 minutes, and 20 seconds were dedicated to simulations, 1 hour, 2 minutes, and 6 seconds were allocated for back-propagation optimization, 8 minutes and 27 seconds were spent on predictions using both the surrogate model and the MVANN, and 250 milliseconds were utilized for searching the best design within the dataset.

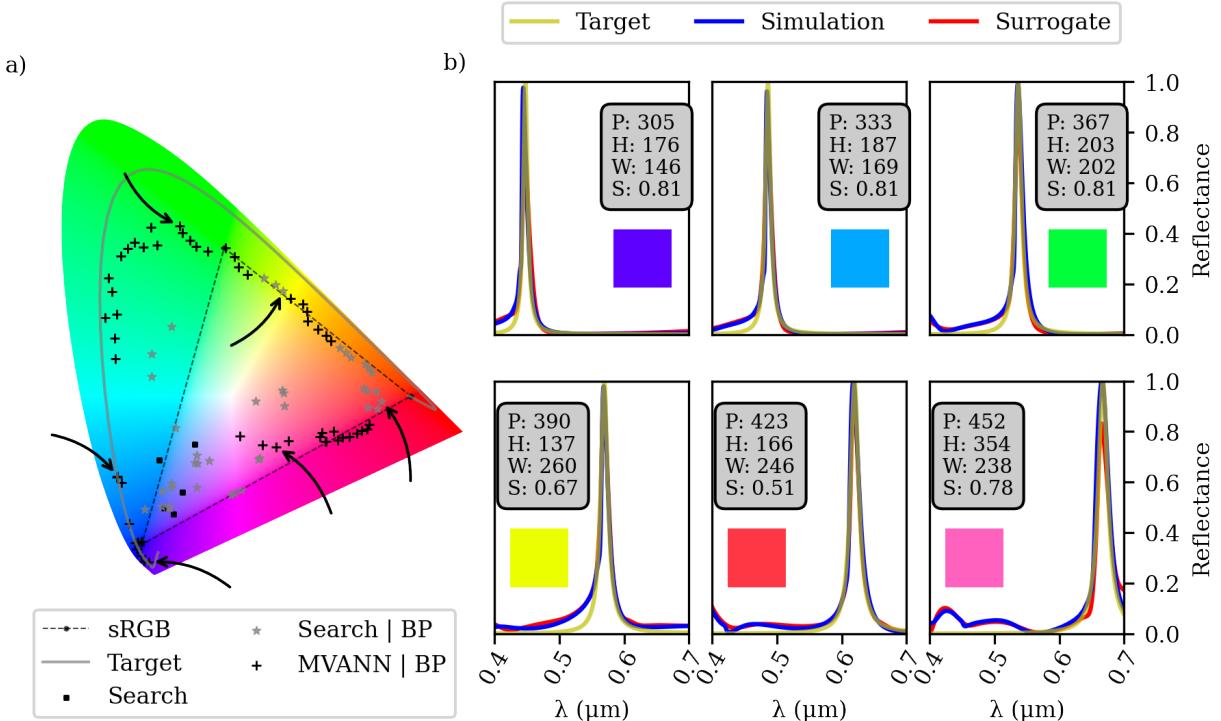


Figure 6: (a) Chromaticity diagram depicting the outcomes of the 100 optimization results. The arrows in the diagram indicate the corresponding designs showcased in (b). The gray line represents the calculated perceived color corresponding to the target spectrum. (b) detailed examination of 6 different optimizations, displaying their dimensions (in nm) of each geometrical parameter and the corresponding sRGB color representation of the simulated spectrum.

The results presented in Figure 6 show that the optimized designs have varying heights. However, considering the complexity associated with fabricating structures of different heights, it is preferable to find designs with a fixed height. Therefore, we introduce a constraint to the back-propagation optimization process, requiring all designs to have equal heights. To achieve this, we modified the inverse design workflow as follows: Initially, we obtained the lowest 5 losses from the MVANN for each of the 6 different target designs, each representing a different color and having a different height. Subsequently, we identified designs that were closer in their height values. We calculated the mean height of

these selected designs for each color and used this mean value as an initial guess for optimizing each target using the back-propagation optimization process.

To accommodate the height constraint, we introduced a dedicated layer in the surrogate model. This layer connected the height input solely to the initial guess and consisted of a single weight and bias. This shared layer was employed throughout all optimization iterations. By optimizing all 6 models simultaneously within the same batch, we ensured that the loss for each individual model progressively decreased with each training epoch. Consequently, we transformed the fixed height constraint into an optimized parameter, allowing us to determine the most suitable height value for achieving the desired outcomes across all six colors.

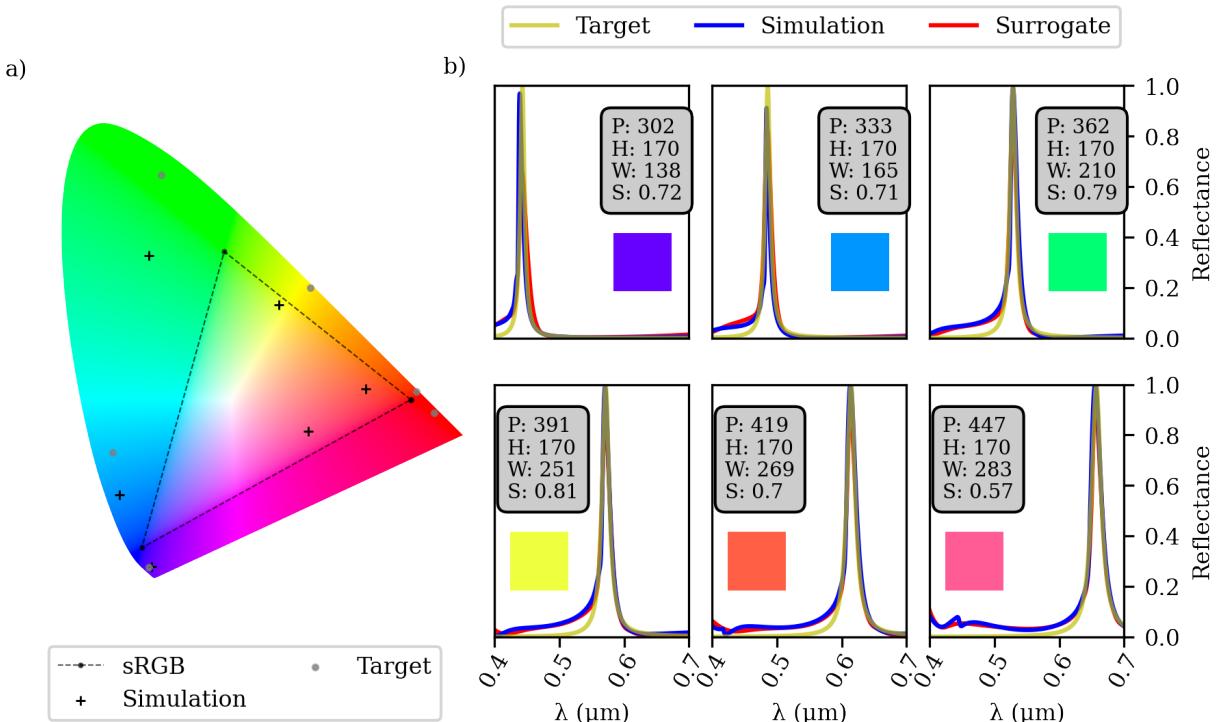


Figure 7: Optimization results based on a fixed height configurations. (a) refers to the chromaticity diagram of 6 optimized designs. The simulated spectra is given in (b) and the time used for training, and optimizing is depicted in the supplementary information section.

The results presented in Figure 7 demonstrate the effectiveness of our approach in achieving favorable outcomes for most of the considered colors. Our optimization approach allows us to explore various line shapes using the same dataset, as depicted in Figure 8. Notably, our algorithm excels at identifying optimized designs that seamlessly match the desired response. It is important to mention that our study focuses on achieving pure colors across the entire spectrum, rather than emphasizing a single color. Yet, interestingly, our algorithm is capable of optimizing designs to replicate a vibrant red color response, a task that traditionally involves complex optimization processes (see [20, 44]). However, our design approach, coupled with advanced inverse design techniques, simplifies the process of identifying designs that mimic this vivid red color response, all while relying on the same dataset. Furthermore, it is worth noting that prior studies employed half the number of simulations to optimize a single color compared to our methodology.

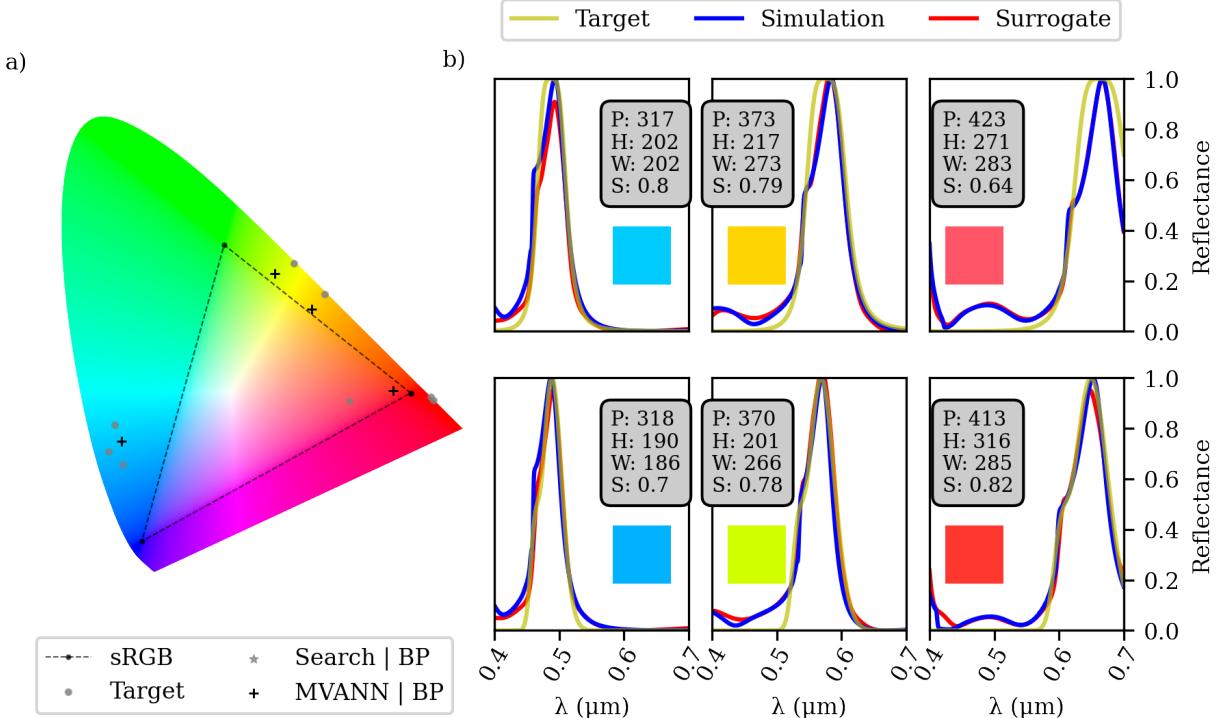


Figure 8: Optimization results targeting a broader Lorentzian ( $\omega^2/(\omega^2 + 4(f - f_0)^6)$ ) on the top row and a combination of Gaussians to mimic a strong secondary resonance on the bottom row. (a) refers to the chromaticity diagram of 6 optimized designs. The simulated spectra is given in (b). It can be seen that we achieved the most vivid red.

## Conclusion

In conclusion, our research presents a novel DL methodology for optimizing resilient designs of vivid color filter metasurfaces. By utilizing a surrogate model constructed from a dataset of only 585 simulations, our approach demonstrates exceptional efficiency in the optimization process. The numerical tool developed in this study enables the cost-effective fabrication of structural color filters by exploring a wide range of narrow line shapes that exhibit high-quality resonances, aligning with desired spectral reflection responses. Notably, our methodology expands the color gamut beyond the conventional RGB colors, offering unprecedented versatility in color generation. Furthermore, our DL approach successfully respects fabrication constraints, ensuring practical feasibility. The achievements of our research significantly contribute to the field of optical device design. By pushing the boundaries of metasurface optimization, we open up new possibilities for the development of advanced optical devices. The proposed methodology holds promise for various applications, such as display technologies, data encoding, and artistic expression. These notable advancements not only enhance the understanding of metasurface design principles but also provide valuable insights for future research endeavors.

## References

- [1] Craig F Bohren and Donald R Huffman. *Absorption and scattering of light by small particles*. John Wiley & Sons, 2008.
- [2] Milton Kerker. *The scattering of light and other electromagnetic radiation*. Elsevier, 2016.
- [3] Mark A Brown and Stephen C De Vito. Predicting azo dye toxicity. *Critical reviews in environmental science and technology*, 23(3):249–324, 1993.
- [4] Hyoki Kim, Jianping Ge, Junhoi Kim, Sung-eun Choi, Hosuk Lee, Howon Lee, Wook Park, Yadong Yin, and Sunghoon Kwon. Structural colour printing using a magnetically tunable and lithographically fixable photonic crystal. *Nature Photonics*, 3(9):534–540, 2009.

- [5] Wei Ting Chen, Kuang-Yu Yang, Chih-Ming Wang, Yao-Wei Huang, Greg Sun, I-Da Chiang, Chun Yen Liao, Wei-Lun Hsu, Hao Tsun Lin, Shulin Sun, et al. High-efficiency broadband meta-hologram with polarization-controlled dual images. *Nano letters*, 14(1):225–230, 2014.
- [6] Haibo Hu, Qian-Wang Chen, Jian Tang, Xian-Yi Hu, and Xu-Hui Zhou. Photonic anti-counterfeiting using structural colors derived from magnetic-responsive photonic crystals with double photonic bandgap heterostructures. *Journal of Materials Chemistry*, 22(22):11048–11053, 2012.
- [7] Fei Ding, Anders Pors, and Sergey I Bozhevolnyi. Gradient metasurfaces: a review of fundamentals and applications. *Reports on Progress in Physics*, 81(2):026401, 2017.
- [8] Egor Khaidarov, Damien Eschimese, Keng Heng Lai, Aihong Huang, Yuan Hsing Fu, Qunying Lin, Ramon Paniagua-Domínguez, and Arseniy I Kuznetsov. Large-scale vivid metasurface color printing using advanced 12-in. immersion photolithography. *Scientific Reports*, 12(1):14044, 2022.
- [9] Arseniy I Kuznetsov, Andrey E Miroshnichenko, Mark L Brongersma, Yuri S Kivshar, and Boris Luk'yanchuk. Optically resonant dielectric nanostructures. *Science*, 354(6314):aag2472, 2016.
- [10] Guoxing Zheng, Holger Mühlenbernd, Mitchell Kenney, Guixin Li, Thomas Zentgraf, and Shuang Zhang. Metasurface holograms reaching 80% efficiency. *Nature nanotechnology*, 10(4):308–312, 2015.
- [11] Ehsan Arbabi, Seyedeh Mahsa Kamali, Amir Arbabi, and Andrei Faraon. Full-stokes imaging polarimetry using dielectric metasurfaces. *Acs Photonics*, 5(8):3132–3140, 2018.
- [12] Andreas Tittl, Aleksandrs Leitis, Mingkai Liu, Filiz Yesilkoy, Duk-Yong Choi, Dragomir N Neshev, Yuri S Kivshar, and Hatice Altug. Imaging-based molecular barcoding with pixelated dielectric metasurfaces. *Science*, 360(6393):1105–1109, 2018.
- [13] Yuanmu Yang, Ivan I Kravchenko, Dayrl P Briggs, and Jason Valentine. All-dielectric metasurface analogue of electromagnetically induced transparency. *Nature communications*, 5(1):5753, 2014.
- [14] Amir Arbabi, Yu Horie, Mahmood Bagheri, and Andrei Faraon. Dielectric metasurfaces for complete control of phase and polarization with subwavelength spatial resolution and high transmission. *Nature nanotechnology*, 10(11):937–943, 2015.
- [15] Mikhail I Shalaev, Jingbo Sun, Alexander Tsukernik, Apra Pandey, Kirill Nikolskiy, and Natalia M Litchinitser. High-efficiency all-dielectric metasurfaces for ultracompact beam manipulation in transmission mode. *Nano letters*, 15(9):6261–6266, 2015.
- [16] Dianmin Lin, Pengyu Fan, Erez Hasman, and Mark L Brongersma. Dielectric gradient metasurface optical elements. *science*, 345(6194):298–302, 2014.
- [17] Manuel Decker, Isabelle Staude, Matthias Falkner, Jason Dominguez, Dragomir N Neshev, Igal Brener, Thomas Pertsch, and Yuri S Kivshar. High-efficiency dielectric huygens' surfaces. *Advanced Optical Materials*, 3(6):813–820, 2015.
- [18] Ye Feng Yu, Alexander Y Zhu, Ramón Paniagua-Domínguez, Yuan Hsing Fu, Boris Luk'yanchuk, and Arseniy I Kuznetsov. High-transmission dielectric metasurface with  $2\pi$  phase control at visible wavelengths. *Laser & Photonics Reviews*, 9(4):412–418, 2015.
- [19] Han-Don Um, Deokjae Choi, Amit Solanki, Emerald Huang, Kwanyong Seo, and Fawwaz Habbal. Dynamic selection of visible wavelengths using resonant  $\text{TiO}_2$  nanostructures. *Nanophotonics*, (0), 2023.
- [20] Ronghui Lin, Vytautas Valuckas, Thi Thu Ha Do, Arash Nemati, Arseniy I Kuznetsov, Jinghua Teng, and Son Tung Ha. An efficient neural optimizer for resonant nanostructures: demonstration of highly-saturated red silicon structural color. *arXiv preprint arXiv:2304.13516*, 2023.
- [21] Xiao Shang, Jiebin Niu, He Li, Longjie Li, Huakui Hu, Cheng Lu, and Lina Shi. Polarization-sensitive structural colors based on anisotropic silicon metasurfaces. In *Photonics*, volume 10, page 448. MDPI, 2023.
- [22] Wei Ma, Feng Cheng, Yihao Xu, Qinlong Wen, and Yongmin Liu. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Advanced Materials*, 31(35):1901111, 2019.
- [23] John Peurifoy, Yichen Shen, Li Jing, Yi Yang, Fidel Cano-Renteria, Brendan G DeLacy, John D Joannopoulos, Max Tegmark, and Marin Soljačić. Nanophotonic particle simulation and inverse design using artificial neural networks. *Science advances*, 4(6):eaar4206, 2018.
- [24] Takashi Asano and Susumu Noda. Optimization of photonic crystal nanocavities based on deep learning. *Optics express*, 26(25):32704–32717, 2018.
- [25] Jiaqi Jiang and Jonathan A Fan. Global optimization of dielectric metasurfaces using a physics-driven neural network. *Nano letters*, 19(8):5366–5372, 2019.

- [26] Fufang Wen, Jiaqi Jiang, and Jonathan A Fan. Robust freeform metasurface design based on progressively growing generative networks. *ACS Photonics*, 7(8):2098–2104, 2020.
- [27] Peter R Wiecha and Otto L Muskens. Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures. *Nano letters*, 20(1):329–338, 2019.
- [28] Peter R Wiecha, Arnaud Arbouet, Christian Girard, and Otto L Muskens. Deep learning in nano-photonics: inverse design and beyond. *Photonics Research*, 9(5):B182–B200, 2021.
- [29] Omid Hemmatyar, Sajjad Abdollahramezani, Yashar Kiarashinejad, Mohammadreza Zandehshahvar, and Ali Adibi. Full color generation with fano-type resonant hfo 2 nanopillars designed by a deep-learning approach. *Nanoscale*, 11(44):21266–21274, 2019.
- [30] Mahmoud MR Elsawy, Stéphane Lanteri, Régis Duvigneau, Gauthier Brière, Mohamed Sabry Mohamed, and Patrice Genevet. Global optimization of metasurface designs using statistical learning methods. *Scientific reports*, 9(1):1–15, 2019.
- [31] Dong Xu, Yu Luo, Jun Luo, Mingbo Pu, Yaxin Zhang, Yinli Ha, and Xiangang Luo. Efficient design of a dielectric metasurface with transfer learning and genetic algorithm. *Optical Materials Express*, 11(7):1852–1862, 2021.
- [32] Samad Jafar-Zanjani, Sandeep Inampudi, and Hossein Mosallaei. Adaptive genetic algorithm for optical metasurfaces design. *Scientific reports*, 8(1):11040, 2018.
- [33] Chao Zhang, Jing Jin, Weicong Na, Qi-Jun Zhang, and Ming Yu. Multivalued neural network inverse modeling and applications to microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 66(8):3781–3797, 2018.
- [34] Hongjun Liu, Huiren Peng, Ke Li, Liping Lu, Junhong Deng, Yanjun Liu, Chengfeng Qiu, Guixin Li, and Xing Cheng. Transfer printing of solution-processed 3d zno nanostructures with ultra-high yield for flexible metasurface color filter. *Advanced Materials Interfaces*, 9(21):2101963, 2022.
- [35] Soumyashree S Panda and Ravi S Hegde. Transmission-mode all-dielectric metasurface color filter arrays designed by evolutionary search. *Journal of Nanophotonics*, 14(1):016014–016014, 2020.
- [36] Yicheng Wang, Weikai Huang, Yu-Sheng Lin, and Bo-Ru Yang. A tunable color filter using a hybrid metasurface composed of zno nanopillars and ag nanoholes. *Nanoscale Advances*, 4(17):3624–3633, 2022.
- [37] D Barton III, M Lawrence, and J Dionne. Wavefront shaping and modulation with resonant electro-optic phase gradient metasurfaces. *Applied Physics Letters*, 118(7):071104, 2021.
- [38] Mark Lawrence, David R Barton III, Jefferson Dixon, Jung-Hwan Song, Jorik van de Groep, Mark L Brongersma, and Jennifer A Dionne. High quality factor phase gradient metasurfaces. *Nature Nanotechnology*, 15(11):956–961, 2020.
- [39] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [40] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [41] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Osdi*, volume 16, pages 265–283. Savannah, GA, USA, 2016.
- [42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [43] János Schanda. *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007.
- [44] Zhaogang Dong, Lei Jin, Soroosh Daqiqeh Rezaei, Hao Wang, Yang Chen, Febiana Tjiptoharsono, Jinfa Ho, Sergey Gorelik, Ray Jia Hong Ng, Qifeng Ruan, et al. Schrödinger’s red pixel by quasi-bound-states-in-the-continuum. *Science Advances*, 8(8):eabm4512, 2022.

## Author contributions statement

A.C.D developed the DL architecture and performed the numerical simulation. ME proposed the structure and coordinated the work. M.E, A.C.D, S.L discussed the numerical results. M.E, A.C.D, H.E.H.F, and M.A discussed the physical results. All authors contributed to the writing and reviewed the manuscript. The authors declare that they have no competing interests.

## **Acknowledgment**

Hugo EHF would like to thank the support of Brazilian government agencies CAPES and CNPq (through project: 312714/2019-0), and the São Paulo Research Foundation (FAPESP) agency through the projects: 2021/11380-5 (CCD/CPTEn) and 2021/06506-0.

---

## SUPPLEMENTARY INFORMATION

---

### 1 Simulation setup

The simulations are realized using python's MEEP library [1]. The default resolution is 150 pixels per  $\mu\text{m}$ , and therefore, 105 pixels per longest wavelength. However, if the cell size of the simulation is not an integer number of pixels, instead of adapting the resolution, the default MEEP procedure is to change the cell size. In our case, we require fine control over the period. That is why we instead used a value close to 150 nm that resulted in the cell dimensions to be an integer number of pixels. MEEP stills warns that the grid volume is not an integer number of pixels and changes slightly the cell size due to the truncation of some values. Each simulation only evaluates a single resonator and there is periodic boundary condition on each boundary of the simulation. Perfectly Matched Layer (PML) layers with  $0.8 \mu\text{m}$  length were positioned on the extremes of the simulation except from the sides. MEEP is known to have issues regarding periodic boundary conditions and PMLs, and the longer the interface between them, the more artifacts the simulation acquires and easier it becomes to diverge. The other option was using Absorber layers, the problem is that they require much more space to fully extinguish the fields and increases the simulation time. Some reaching 20 minutes. As the length of the PML is slightly larger than the longest wavelength, it extinguishes the fields while having imperceptible simulation artifacts. The same simulation that would last 20 minutes with the absorber layer, now takes only 14 seconds. The total height of the simulation is  $3.1 \mu\text{m}$ . The height of the cell is adjusted if the resolution is updated. The light is injected from top with normal incidence with electric field polarized in the x-direction. There are 5 uniformly distributed in frequency Gaussian sources to guarantee that floating point errors does not interfere with the results. The transmission spectrum was sampled from 400 nm to 700 nm with 500 points on the bottom of the simulation. An electric filed sampling on the x-direction was positioned between the sources and the middle of the simulation. It records the maximum value it ever encounters and then it stops the simulation if the field recorded on that point is  $10^{-8}$  smaller than the maximum value it found. It then performs a Fast Fourier Transform (FFT) to compute the transmission spectrum. As there are no losses in the simulation, the reflection spectrum is calculated using the transmission spectrum. In order to calculate the transmission spectrum, firstly it is required to validate the "raw" transmission without any structure. Therefore, computing the spectrum of the source. By varying the period of the resonator, we observe changes the intensity of the transmission as seen in Figure 1 a).

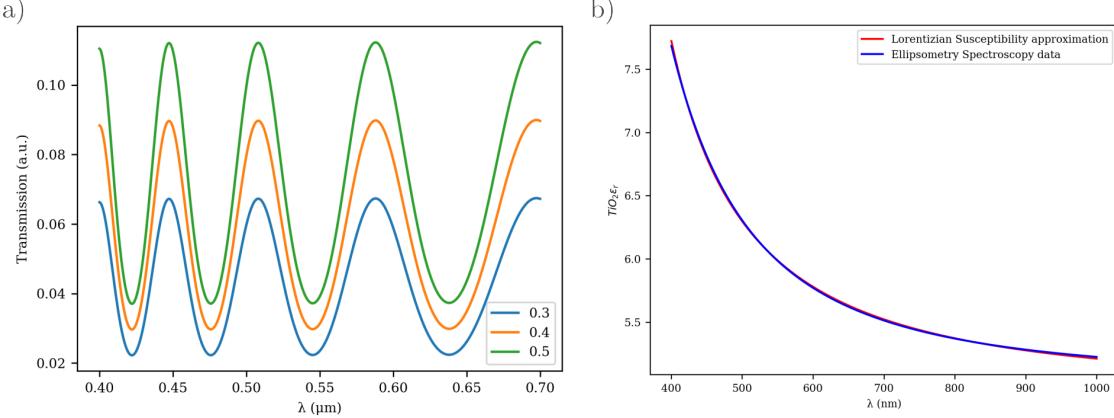


Figure 1: a) Transmission spectrum without any structure. Each curve represent a different resonator period. b) Comparative analysis of the measured dispersion of  $TiO_2$  and the approximation generated through modeling using MEEP. The experimental data capturing the dispersion characteristics were obtained through Ellipsometry spectroscopy detailed below. Subsequently, the collected data were fitted to a Lorentzian susceptibility function. Notably, the approximation achieved through the modeling process closely aligns with the measured dispersion, demonstrating the efficacy of the approach in accurately representing the dispersion behavior of  $TiO_2$

## $TiO_2$ Ellipsometry Spectroscopy

The characterization of the two-dimensional flat layers made of  $TiO_2$  was performed in order to determine their optical constants. The measurements were carried out with a Wollam M200V ellipsometer (350-1000 nm) on a layer obtained via sol-gel chemistry and dip-coating on a bulk Si substrate[2]. The ellipsometric data ( $\Delta$  and  $\Psi$ ) are fitted by a Cauchy model to extract refractive index  $n$  and extinction coefficient  $k$  (being the latter equal to zero in the considered wavelength range). These layers are of optical quality being their surface roughness below 0.1 nm [3].

Measuring the refractive index evolution of a  $TiO_2$  thin layer kept in a ellipsometry-porosimetry chamber with controlled humidity (atmosphere where air and water were progressively mixed) we assess the adsorption/desorption of water. From the corresponding isotherms we obtain the mesoporosity of the layers[4] (pore volume, size distribution, and interconnection through adapted Kelvin models). Measurements on dense  $TiO_2$  ( $n > 2.4$ ) account for a porosity well below 10%

The refractive index of the  $TiO_2$  was fitted by a Lorentzian susceptibility function using scipy's optimize curve\_fit function with initial guess 1.1 for epsilon, 3.85 of frequency and 3.75 of  $\sigma$ . The final parameters values are 1.87681265 for epsilon, 3.63269272 for frequency and 3.08109585 for  $\sigma$ . The difference between the ellipsometry data and the fitted curve can be seen in Figure 1 b).

To calculate the spectrum source for every period, we assumed a linear correlation between the period and the magnitude of the transmittance, and performing a linear interpolation for every point in the spectrum, obtaining the source vector  $\mathbf{a}$ . The linear coefficient was close to 0 for all 500 points, and therefore, considered to be exactly 0. And therefore, to obtain the source spectrum of a resonator with period  $P$ , we can simply multiply the source vector  $\mathbf{a}$  with the period of the metasurface.

A comparison of MEEP Finite-Difference Time-Domain (FDTD) simulation and COMSOL multiphysics Finite Element Method (FEM), has been and shows on Figure 2 a) and b).

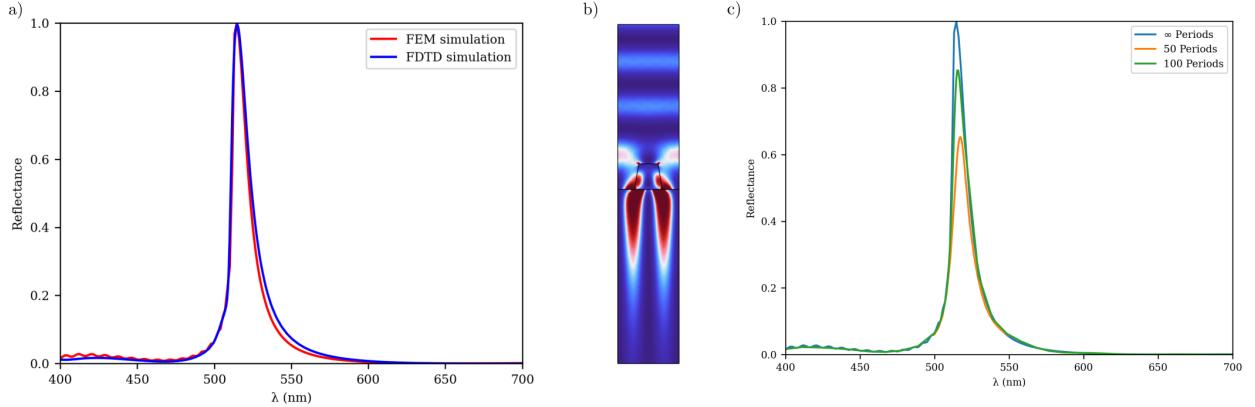


Figure 2: Comparison between the FDTD MEEP simulations and FEM COMSOL simulations. With  $P = 353$  nm,  $H = 147$  nm,  $W = 159$  nm,  $S = 0.3$ . a) Reflection spectrum and b) Electric field normalized at the wavelength corresponding to the peak reflectance. c) Difference between different array sizes.

Figure 2 c) shows how the reflectance, transmittance and loss responds to different array sizes. As the period is 353 nm, the last configuration with 100 repetitions is 35.3  $\mu\text{m}$  in length. And as seen, the resonance requires many repetitions to achieve the desired performance. The reason for this behavior is the mode excited by the resonant frequency. Which propagates in the x-axis in the substrate as seen by Figure 3.

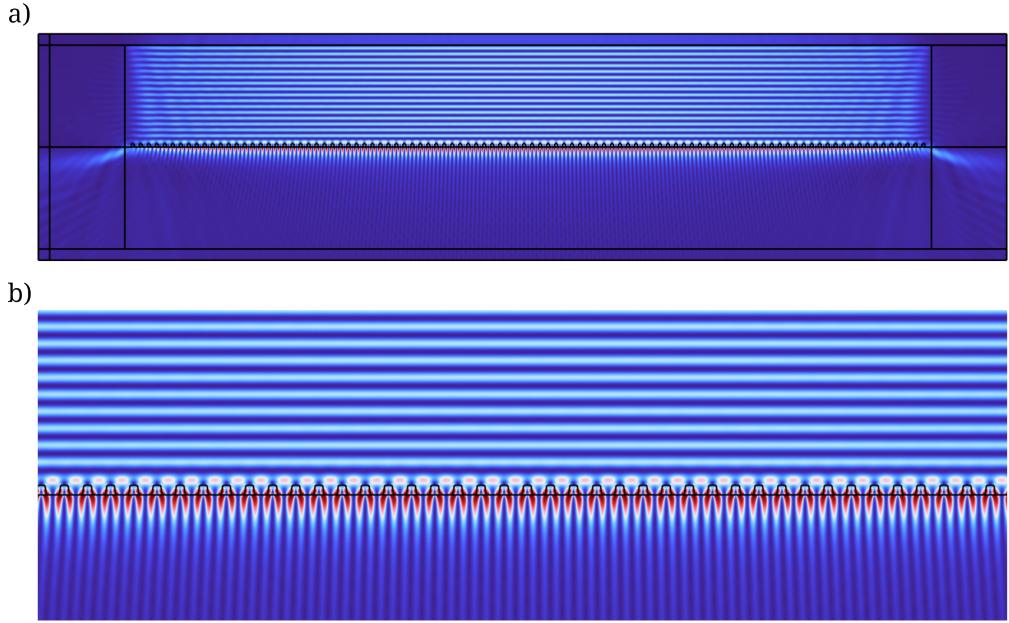


Figure 3: Fields computed at the wavelength of 515 nm for the scenario of 100 repetitions. a) Electric field normalized of the entire simulation domain. b) Zoom at the center of the simulation domain to obtain a better view of the local fields near the resonators.

## 2 Surrogate Training

Figure 4 displays the loss evolution at each epoch.

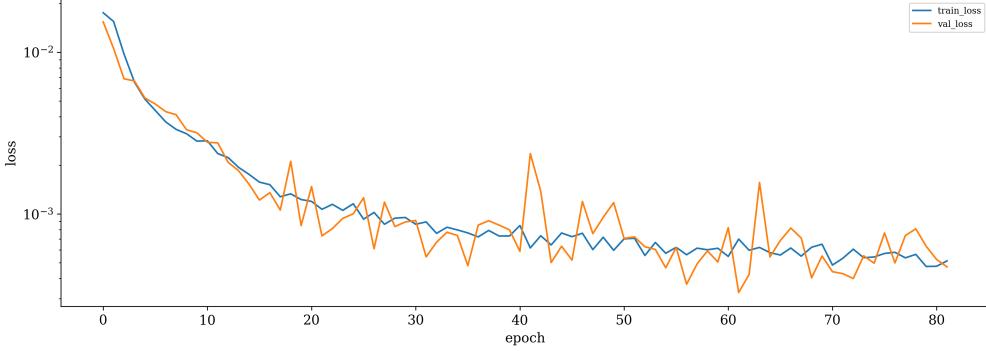


Figure 4: Mean Square Error (MSE) loss of the training and validation sets in function of the epoch for the surrogate model.

### 3 Back-propagation optimization methodology

By disabling the training of the weights of the surrogate model and added one layer before the input. The added layer has 5 inputs and the weights were initialized with the identity. The 4 first inputs (geometry parameters) are fully connected with the first 4 inputs of the surrogate model and the last input ( $\lambda$ ) is solely connected to the surrogate model input referent to the wavelength. The training of this connection is disabled. The activation function for the described layer was Rectified Linear Unit (ReLU) to prevent negative values. The most important part is the training. We used full batch Adam optimizer with initial learning rate of  $10^{-5}$  with exponential decay with 1000 decay steps and decay rate of 0.98. These choices proved necessary due to the needed small adjustments and the gradients grow after passing through the entire Artificial Neural Network (ANN) without any Batch Normalization (BN) layer. The training spanned 1800 epochs with patience of 6 with minimum delta of  $10^{-6}$ , meaning if the loss decrease less than  $10^{-6}$  in 6 epochs, the training stops. We present the Python code using tensorflow and keras framework to run the back-propagation optimization.

```

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=1e-5,
    decay_steps=1000,
    decay_rate=0.98) # Learning rate schedule
opt = tf.keras.optimizers.Adam(learning_rate=lr_schedule) # Using Adam

# Patience
callback = tf.keras.callbacks.EarlyStopping(monitor='loss',
                                             patience=6,
                                             min_delta=1e-6)

# Define the model to be trained
Input_all = Input(shape=(5,)) # Input that will receive the initial guess

# Layer responsible for finding the geometrical parameters.
d1 = Dense(4, kernel_initializer='identity', activation='relu')(Input_all[:, :-1])

# Layer that will pass the wavelength.
# Note that we do not want to optimize this layer.
d2 = Dense(1, kernel_initializer='identity', trainable=False)(Input_all[:, -1:])
d3 = Concatenate()([d1, d2])
inverse_designer = Model(inputs=Input_all, outputs=d3)

Surrogate_model.trainable = False # Freeze the weight of the surrogate model.
Full_model_outputs = Surrogate_model(inverse_designer.output)

# Creating the model comprising the surrogate and our layer
Full_model = Model(inverse_designer.input, Full_model_outputs)
Full_model.compile(optimizer=opt, loss='mse')

```

```

# Initial guess here can be from the MVANN or from the Search
# Its simply a list in the following format: [P, H, W, S]
ipt_i_tolist = initial_guess.tolist()

# We then create a dataset comprising the combination of
# same input and all the wavelengths
# wl is a list of all wavelengths
eng_inp = np.array([ipt_i_tolist + [w] for w in wl])

# The next step is training the new layer. Note that we want full batch
# optimization

# t_f is the target response
history = Full_model.fit(eng_inp,
                          np.array(t_f),
                          epochs=1800,
                          callbacks=[callback],
                          verbose=0,
                          batch_size=500)

# Then, to obtain the optimized result, we call it.
bp_result = inverse_designer.predict(eng_inp, verbose=0)

```

We decided to incorporate an additional layer for optimization rather than directly computing and applying gradients to the design space. This decision was primarily driven by the fact that the framework already had features like early stopping and learning rate schedules in place, which we found beneficial for our approach.

## 4 Optimization for lower quality factor resonance targets

By performing the dataset search with relaxed objectives that concerns a broader Full Width at Half Maximum (FWHM), a reasonable solution can be found on the dataset. Turning the Remaining optimization methodology unnecessary as seen by Figure 5.

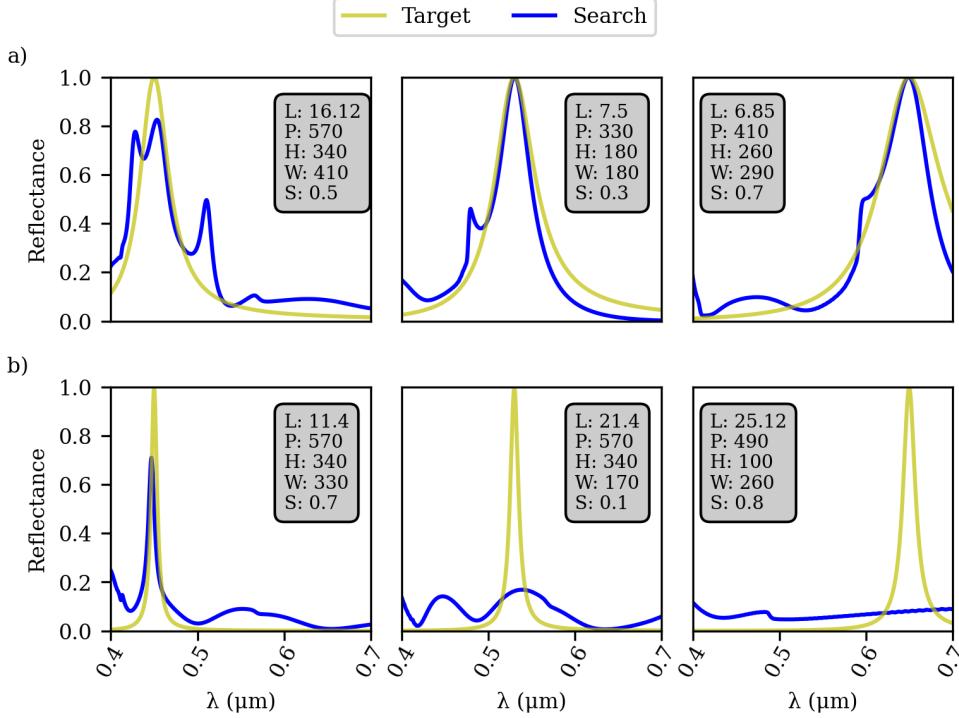


Figure 5: Results from the dataset search using a Lorentzian lineshape in frequency as target for the dataset search for the wavelengths of 450 nm, 560 nm and 650 nm with a)  $\omega = 0.05$  and b)  $\omega = 0.01$ . L is the MSE between the target spectrum and the closest one in the dataset, and P, H, W and S are the geometric parameters. As seen, unlike the low quality target objectives give in (a), the dataset search could not find a good solution when targeting sharp resonances as it is depicted in b).

As our objective is relaxed, there is no need to refine the solutions found by searching the dataset.

## 5 MVANN

A Multi-Valued Artificial Neural Network (MVANN) is simply a ANN capable of outputting multiple solutions given a single input. The adoption of the MVANN in this context is particularly advantageous due to its ability to mitigate the issue of multiple parameter responses for a given objective target [5]. This feature ensures that the MVANN provides reliable and consistent results in non-unique scenarios where multiple solutions can arise from a single input target spectrum.

One of the greatest challenges even for a MVANN is the fact that we are extrapolating, because the data that is going to be used after training is different from the data used to train it. Although looking similar to some training data, a pure Lorentzian curve does not exist in the simulated data, and a small difference between it and a pure Lorentzian spectrum can cause the ANN to not generalize well, because ANN are used to interpolate functions and therefore, should not be applied to extrapolation [6]. There are several solutions to this problem. One of them consists in having a Deep Learning (DL) model capable of extrapolating well. One such example are Physics-Informed Neural Networks (PINNs) that hard codes the physics of the problem into the ANN and can extrapolate extremely well [7]. The problem for our case is that we cannot extract simple differential equations that correlates the input to the output, and therefore, PINNs cannot be considered. The last solution and the least efficient is to apply regularization to the training parameters. By doing so, we manipulate the bias-variance trade off, and by having a lower variance model with higher bias our ANN is able to extrapolate with the downside of losing interpolation performance.

Matching the correct output given a spectrum becomes an arduous task, as it necessitates accurately aligning the corresponding parameters across all wavelengths encompassed within the spectrum. Training with only 810 data points is not enough to obtain reasonable results. A solution is to generate data with the surrogate model to be used as training for the MVANN. We generated 7839 input-output pairs in 4 min and 13 seconds

on a Google colab’s Central Processing Unit (CPU). The geometry limits are the same as for the simulations, the only difference is the step, which now is 25 nm instead of 65 nm.

The architecture of the MVANN is represented by Figure 6. We opted to use residual blocks as they usually grant us good generalization with a reasonable depth [8, 9]. On top of that, we also used L<sub>1</sub> regularization, by summing all the weights on the network, and multiplying by the penalization strength, on this case, 10<sup>-7</sup>, as the ANN had 2 023 808 trainable parameters, meaning an average penalization of 0.2 per parameter. All activation functions were Gaussian Error Linear Unit (GELU) except for the last layers of the output block and the kernel initialiser was Glorot normal for all parameters [10, 11]. BN Layers and dropout layers with high rate were employed to enhance the generalization and extrapolation capacity of the ANN.

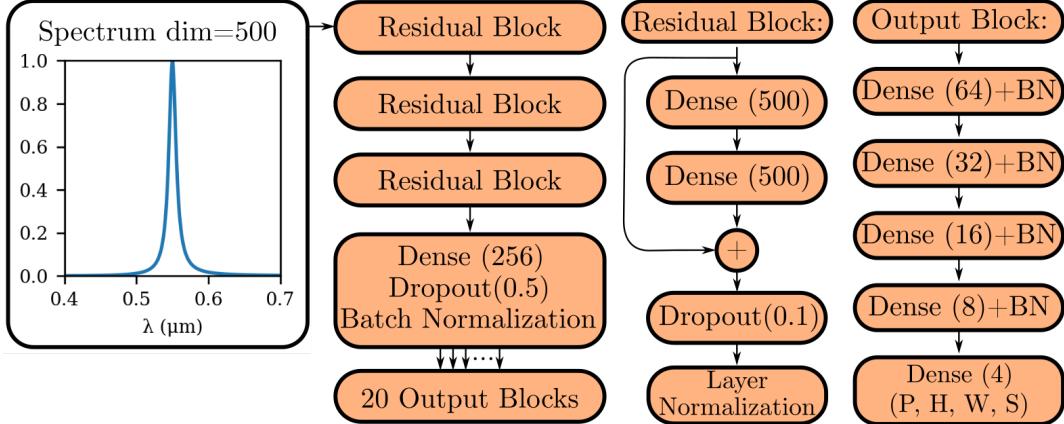


Figure 6: Schematic of the MVANN Architecture. We used 3 residual blocks with 500 neurons per layer followed by a layer with 256 neurons, a dropout layer with rate = 0.5 and batch normalization. Each output block consists of 5 layers, each followed by a batch normalization layer. The first layer starts with 64 neurons and geometrically decays to 4.

The model was trained with tensorflow for 100 epochs with patience of 10 [12]. However, differently from the surrogate model, we have not used holdout.

The decision to exclude the use of holdout training for the MVANN may be viewed as controversial, but it yielded superior initial estimates for the subsequent back-propagation optimization process. Identifying concrete reasons for this phenomenon can be challenging, as it involves various factors, such as the small size of the dataset used for training, the potentially inadequate number of outputs, or the inherent nature of the MVANN in requiring a vast amount of training data for optimal performance in this particular scenario. Additionally, the notion of withholding data for assessing overfitting becomes less crucial in this context, as each additional data point could provide relevant information due to the high sensitivity of the dataset. It is important to note that the results obtained using the MVANN are going to be used as initial guesses for back-propagation optimization and do not necessitate perfection.

The chosen optimizer is Adam and the batch size has been fixed to 32 [13]. The training stopped at epoch 100 with final loss of  $2.3093 \times 10^{-4}$ . The total training time was 46 min 8 s on a google colab’s CPU.

The loss function of the MVANN illustrates a crucial characteristic of the behaviour of the model. This being said, that the presence of a single low loss output solution has a significant impact on the final loss value. This implies that the attainment of a single high-quality solution effectively determines the calculated loss, thereby compelling the specialization of each individual output. Subsequently, we exploit the advantageous properties of the surrogate model ANN, which is fully differentiable. This enables us to propagate gradients through the model using back-propagation, allowing for efficient refinement of the solutions initially provided by the MVANN. For this specific study, we employed a configuration with 20 outputs, which proved to be sufficient in offering reasonable starting guesses for the back-propagation optimization. The loss function for the MVANN is explained in the original material [5]. A small modification of the loss function is used in this study and displayed on the supplementary information section S2.

To quantify the performance of the MVANN, we calculated the loss function  $L(\mathbf{w})$  over a batch of size  $N$ , as represented by Equation (1).

$$L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{B_i(\mathbf{w})} \quad (1)$$

The weights  $\mathbf{w}$  are involved in the calculation of the loss, and  $B_i(\mathbf{w})$  represents the inverse of the loss for a single batch, defined as the sum of the inverse losses for each output, as:

$$B_i(\mathbf{w}) = \sum_{j=1}^{N_{outs}} \frac{1}{S_j(\mathbf{w})} \quad (2)$$

Here  $S_j(\mathbf{w})$  is the  $L_2$  distance between the output  $j$  of the MVANN,  $\mathbf{G}_{\text{ANN},j}(\mathbf{x}, \mathbf{w})$  given the input  $\mathbf{x}$  and the desired response  $\mathbf{d}$ ,

$$S_j(\mathbf{w}) = \|\mathbf{G}_{\text{ANN},j}(\mathbf{x}, \mathbf{w}) - \mathbf{d}\|_2^2 \quad (3)$$

The motivation for the utilization of such loss function is to lower the losses for conflicting outputs and is explained in the original material [5]. However, we give here a quick explanation. If a single output matches very well,  $S_j(\mathbf{w}) \rightarrow 0 \implies B_i(\mathbf{w}) \rightarrow \infty$  which will imply that the loss for that point of data is going to tend to 0 even if there is a  $S_j(\mathbf{w}) \rightarrow \infty$  [5]. The only difference between the original MVANN and the used on this paper is that we haven't divided by 2 the  $L_2$  distance represented by Equation (3) [5].

Figure 7 displays the loss evolution at each epoch. The validation loss considers the same dataset used to train it, and it is lower than the train loss due to the use of dropout, batch normalization layers and most importantly, the L1 regularization term that is only present in the training loss and not in the validation.

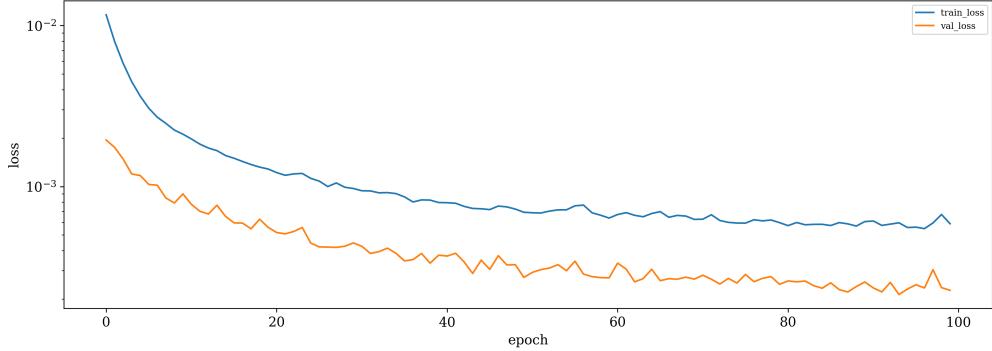


Figure 7: MSE loss of the training and validation sets in function of the epoch for the MVANN model.

## 6 Study of the efficacy of the MVANN with different dataset sizes

In this section, we investigate how the dataset size affects the performance of the MVANN. In Figure 8, we present the MSE between the target function and the optimized design at various wavelengths. The orange curve corresponds to the search with back-propagation, while the blue curve represents the MVANN with back-propagation for four different dataset sizes. Notably, the MVANN with back-propagation consistently outperforms the search method. The results from the 585 simulations demonstrate the lowest MSE for the MVANN with back-propagation. This case, involving 585 simulations, essentially represents the limit beyond which further dataset size increases do not yield performance improvements. In other words, increasing the dataset size beyond this point does not enhance performance. Also, it can be observed that it is more common for the MVANN to outperform the search.

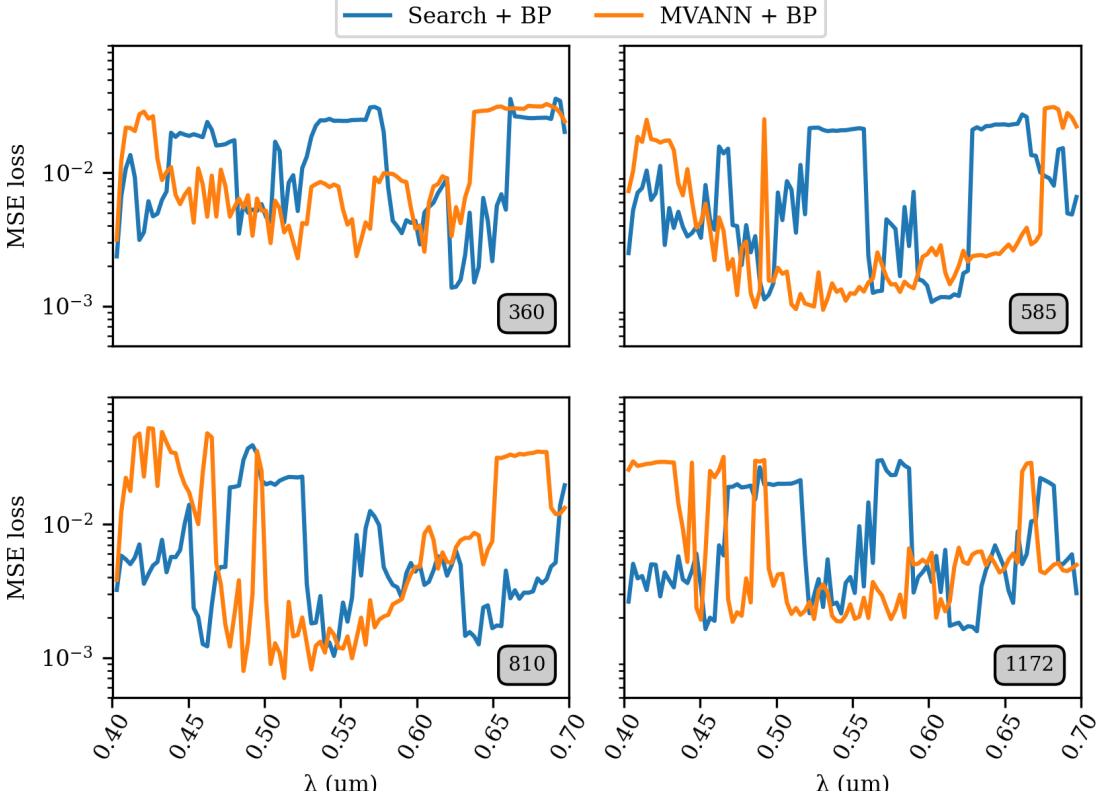


Figure 8: MSE between target function and optimized design at each wavelength. The search scan with back-propagation is represented by the orange curve, while the MVANN with back-propagation is given by the blue curve for 4 different dataset sizes.

## 7 Comparison between MVANN and state-of-the-art algorithms

In Figure 9, we compared the MVANN with a  $\beta$  conditional Variational Autoencoder ( $\beta$ -cVAE) [14, 15]. The  $\beta$ -cVAE model utilized a similar architecture to the MVANN, comprising 3 645 288 trainable parameters and having 2 latent dimensions. Training was conducted over 100 epochs using AdamW with a learning rate of  $3 \times 10^{-4}$  and MSE loss. The training process took 41 minutes and 40 seconds. For comparison, training the MVANN on the same CPU typically requires around 50 minutes. The results can be seen in Figure 9. One of the challenges associated with the  $\beta$ -cVAE is that it necessitates an optimization technique for determining the latent variables, unlike the MVANN where the solution is typically present in at least one of the outputs. In our case, we applied a brute-force optimization by randomly guessing 20 values to make the comparison with the MVANN more equitable. Furthermore, similar to the MVANN, we employed back-propagation optimization for the design variables. The  $\beta$ -cVAE encoder consists of a first layer with 500 neurons, followed by 3 residual layers, and then 128 neurons. The decoder follows the same architecture, with the addition of a layer with 4 neurons after the one with 128 neurons. All layers utilize GELU activation with Glorot normal initialization, except for the layers responsible for computing the mean and the logarithm of the variance before the sampling layer. In conclusion, as demonstrated in Figure 9, both methods generally exhibit similar performance. However, the  $\beta$ -cVAE has two notable drawbacks when compared to the MVANN. Firstly, it still requires an optimization technique to find the latent variables, while in the MVANN, the solution is typically present in at least one of the outputs. Secondly, the implementation of the  $\beta$ -cVAE is less straightforward and more convoluted compared to the MVANN.

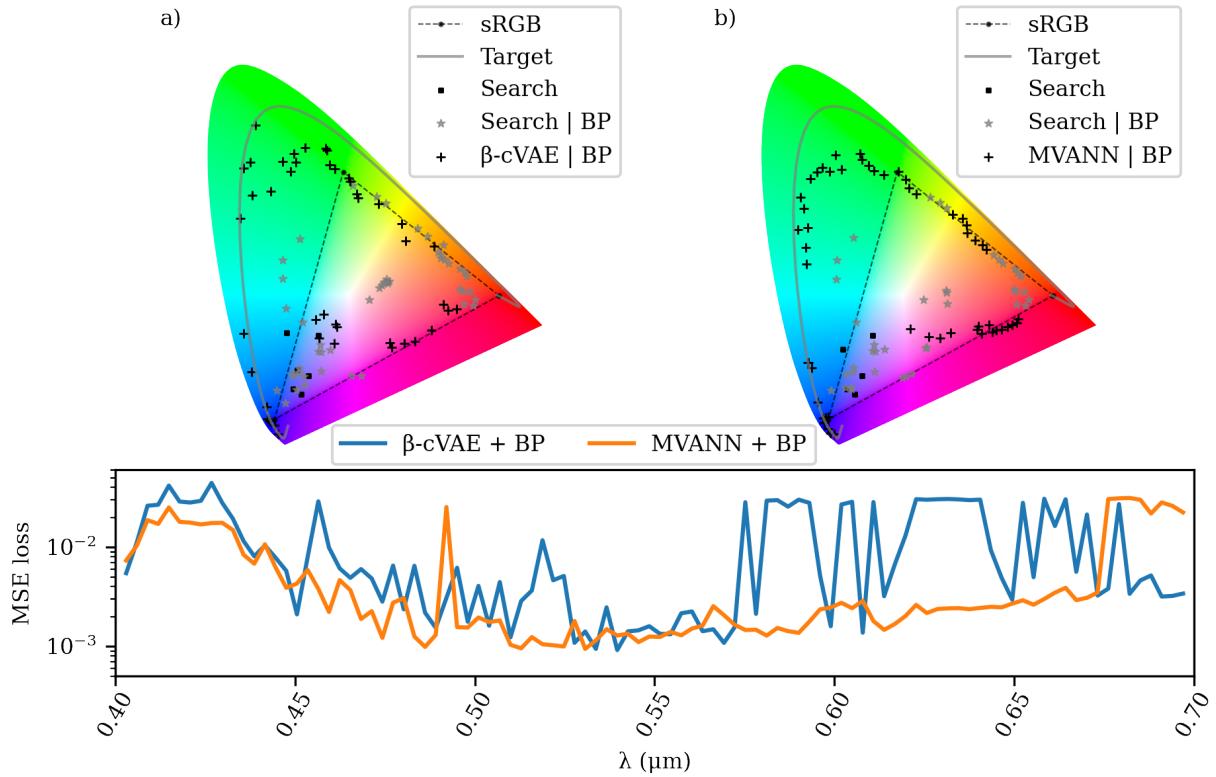


Figure 9: Chromaticity diagram and MSE loss comparing the target with the simulated spectrum of the optimized design for the a)  $\beta$ -cVAE and b) MVANN after back-propagation for a dataset of size 585. They exhibit similar performance in terms of MSE and the chromaticity diagram, except for targets above 575 nm.

One metric to benchmark the MVANN and the Search after back-propagation is to compare the average MSE and standard deviation. However, it does not tell the whole story, as it is observed a bi-stable regime, where it oscillates between two values, and therefore, the minimum and maximum values are also taken into account. Table 1 benchmarks the different dataset sizes.

Table 1: Table representing the mean, standard deviation ( $\sigma$ ), minimum and maximum values for the MSE loss between the simulated and target spectrum for different dataset sizes and methods. The values are multiplied by  $10^3$ .

		Values in $10^{-3}$	Mean	$\sigma$	min	max
360	MVANN + BP	12.8	10.37	2.28	32.82	
	Search + BP	13.92	9.8	1.37	35.96	
585	MVANN + BP	6.19	8.26	0.94	31.16	
	Search + BP	9.65	8.23	1.07	27.40	
	$\beta$ -cVAE + BP	12.06	11.98	0.91	44.14	
810	MVANN + BP	13.78	15.15	0.70	52.74	
	Search + BP	7.94	8.42	1.03	39.30	
	$\beta$ -cVAE + BP	16.1	15.51	0.81	56.86	
1172	MVANN + BP	9.37	10.26	1.86	32.15	
	Search + BP	9.55	8.53	1.58	30.37	

It can be seen that the best case scenario is the dataset with 585 simulations, showing the MVANN after back-propagation to have the least mean, however, for 810, the minimum is lower, and considering Figure 8, that minimum lasts from 475 nm to 575 nm, which is mostly the whole green and cyan areas. However,

this region exhibits instability and spikes. Other conclusion is the whole stability of the MVANN with back-propagation in the scenario with 585, outperforming the search in multiple areas.

## 8 Time table

Table 2: Time required for various stages, including data generation, model training, and optimization runs. The simulations were conducted on the Google Colab platform equipped with an Intel® Xeon® E5 v4 Processor operating at 2.2 GHz. It should be noted that determining the optimization time per design in the fixed height case can be misleading, as the back-propagation time does not scale linearly with the number of optimizations due to training on the same batch. Furthermore, the time needed to identify designs with the closest heights exhibits a significant increase as the number of simultaneous optimizations rises. For instance, while 6 designs required a mere 0.8 seconds, this duration is extended to 5 minutes and 30 seconds when dealing with 10 designs.

Simulations		Models Training	
Total	33 min 40 s	Surrogate	9 min 48 s
Average	3.4 s	Generation	4 min 13 s
		MVANN	46 min 8 s
		Total	1h 0 min 9 s
Free height optimization (1 design)		Fixed height optimization (6 designs)	
Simulations	51.8 s	Simulations	2 min 57 s
Back-propagation	37.3 s	Back-propagation	1 min 18 s
Predictions	5.1 s	Predictions	15 s
Search	2.5 ms	Total	4 min 30 s
Total	1 min 34 s	Per design total	35 s

## References

- [1] Ardavan F Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, John D Joannopoulos, and Steven G Johnson. Meep: A flexible free-software package for electromagnetic simulations by the fdtd method. *Computer Physics Communications*, 181(3):687–702, 2010.
- [2] Elisa Bindini, Guillaume Naudin, Marco Faustini, David Grosso, and Cédric Boissière. Critical role of the atmosphere in dip-coating process. *The Journal of Physical Chemistry C*, 121(27):14572–14580, 2017.
- [3] Zeinab Chehadi, Mohammed Bouabdellaoui, Mehrnaz Modaresialam, Thomas Bottein, Marco Salvalaglio, Monica Bollani, David Grosso, and Marco Abbarchi. Scalable disordered hyperuniform architectures via nanoimprint lithography of metal oxides. *ACS Applied Materials & Interfaces*, 13(31):37761–37774, 2021.
- [4] Jérôme Loizillon, Bettina Baumgartner, Christophe Sinturel, Marco Abbarchi, Bernhard Lendl, and David Grosso. In-depth study of coating multimodal porosity using ellipsometry porosimetry in desorption scanning mode. *The Journal of Physical Chemistry C*, 123(38):23464–23479, 2019.
- [5] Chao Zhang, Jing Jin, Weicong Na, Qi-Jun Zhang, and Ming Yu. Multivalued neural network inverse modeling and applications to microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 66(8):3781–3797, 2018.
- [6] Didulani Acharige and Eric Johlin. Machine learning in interpolation and extrapolation for nanophotonic inverse design. *ACS omega*, 7(37):33537–33547, 2022.
- [7] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [8] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12):5349–5362, 2020.
- [9] A Emin Orhan and Xaq Pitkow. Skip connections eliminate singularities. *arXiv preprint arXiv:1701.09175*, 2017.
- [10] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

- [11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [12] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Osdi*, volume 16, pages 265–283. Savannah, GA, USA, 2016.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [14] Hai Lin, Junjie Hou, Yumei Wang, Rongxin Tang, Xintong Shi, Yuze Tian, Weilin Xu, et al. Machine-learning-assisted inverse design of scattering enhanced metasurface. *Optics Express*, 30(2):3076–3088, 2022.
- [15] Rui Yu, Yuanyuan Liu, and Lu Zhu. Inverse design of high degree of freedom meta-atoms based on machine learning and genetic algorithm methods. *Optics Express*, 30(20):35776–35791, 2022.