



Research article

Zhaxylyk A. Kudyshev*, Alexander V. Kildishev, Vladimir M. Shalaev and Alexandra Boltasseva*

Machine learning–assisted global optimization of photonic devices

<https://doi.org/10.1515/nanoph-2020-0376>

Received July 4, 2020; accepted October 5, 2020;

published online October 28, 2020

Abstract: Over the past decade, artificially engineered optical materials and nanostructured thin films have revolutionized the area of photonics by employing novel concepts of metamaterials and metasurfaces where spatially varying structures yield tailorable “by design” effective electromagnetic properties. The current state-of-the-art approach to designing and optimizing such structures relies heavily on simplistic, intuitive shapes for their unit cells or metaatoms. Such an approach cannot provide the global solution to a complex optimization problem where metaatom shape, in-plane geometry, out-of-plane architecture, and constituent materials have to be properly chosen to yield the maximum performance. In this work, we present a novel machine learning–assisted global optimization framework for photonic metadvice design. We demonstrate that using an adversarial autoencoder (AAE) coupled with a metaheuristic optimization framework significantly enhances the optimization search efficiency of the metadvice configurations with complex topologies. We showcase the concept of physics-driven compressed design space engineering that

introduces advanced regularization into the compressed space of an AAE based on the optical responses of the devices. Beyond the significant advancement of the global optimization schemes, our approach can assist in gaining comprehensive design “intuition” by revealing the underlying physics of the optical performance of metadvice with complex topologies and material compositions.

Keywords: machine learning; metasurface; optimization; thermal emitter.

1 Introduction

Multiconstrained optimization of metamaterials [1] and metasurfaces [2–5] requires intensive computational efforts. The main goal of such optimization is to determine the distribution of constituent materials within the computational domain, which assures the best performance of the metadvice while satisfying all the constraints of the problem. Recently, various gradient-based [6–10] and metaheuristic algorithms [11, 12] (evolutionary, swarm based) have been adapted to advance nanophotonic design problems. However, even the simplest realizations of these optimization frameworks depend heavily on computationally expensive three-dimensional (3D) full-wave direct electromagnetic solvers, thus making the proposed frameworks very time-consuming and inefficient.

Moreover, the computational costs of conventional optimization methods increase with the number of additional constraints, thus making these methods less practical for highly constrained problems. On the other hand, with the development of novel material platforms and advances in nanofabrication techniques, there is a growing interest in the multiconstrained optimization of such metastructures, which can be decisive in addressing critical problems in the fields of space exploration [13], quantum technology [14], energy [15], and communication [16]. There is a critical demand for efficient optimization frameworks capable of performing global optimization searches within highly dimensional parametric domains with complex optimization landscapes.

***Corresponding authors: Zhaxylyk A. Kudyshev and Alexandra Boltasseva**, School of Electrical and Computer Engineering, Birck Nanotechnology Center and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, IN, 47906, USA; and The Quantum Science Center (QSC), A National Quantum Information Science Research Center of the U.S. Department of Energy (DOE), Oak Ridge, USA, E-mail: zkudyshev@purdue.edu (Z.A. Kudyshev), aeb@purdue.edu (A. Boltasseva). <https://orcid.org/0000-0002-6955-0890> (Z.A. Kudyshev)

Alexander V. Kildishev, School of Electrical and Computer Engineering, Birck Nanotechnology Center and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, IN, 47906, USA

Vladimir M. Shalaev, School of Electrical and Computer Engineering, Birck Nanotechnology Center and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, IN, 47906, USA; and The Quantum Science Center (QSC), A National Quantum Information Science Research Center of the U.S. Department of Energy (DOE), Oak Ridge, USA

Due to its versatility and efficiency, machine learning (ML) algorithms have been successfully applied to different areas of photonics and optoelectronics. Various ML techniques have demonstrated their potential to address the bottlenecks of the conventional methods. For example, machine and deep learning models have been used in microscopy [17], quantum optics [18–20], and laser physics [21]. Recently, discriminative networks have been applied to various direct and inverse electrodynamics problems in nanophotonics [22–33]. The main advantage of the data-driven frameworks over the conventional electrodynamic simulation methods is the ability of the neural networks (NNs) to identify hidden correlations in the large data sets during the training phase and utilizing the retrieved “knowledge” to provide instantaneous solution searches, without costly computations [34, 35].

Along with the optimization of geometrical parameters and prediction of the optical response of metastructures with simplistic shapes, the advanced deep learning algorithms have been used to perform optimization of the metadevices with complex topologies. Thus, specific classes of generative networks, such as generative adversarial networks (GANs) [36, 37] and autoencoders [38–40], have been applied to nanoantenna design optimization. Recently, GANs and AAEs have been coupled with adjoint topology optimization (TO) technique for optimizing the diffractive dielectric gratings [41, 42], as well as thermal emitters [43]. It has previously been demonstrated that by coupling the AAE network with a conventional adjoint TO formalism, it is possible to get ~4900-time speedup in thermal emitter optimization compared with conventional TO [34].

Within this work, we extended the AAE-based optimization framework beyond the random sampling of the designs from the AAE network. Specifically, we have developed a methodology to perform the multiparametric global optimization (GO) directly within the compressed design space, via coupling the conditional AAE (c-AAE) network with a differential evolution (DE) optimizer. The proposed approach allows us to not only determine the antenna shape/topology but also optimize the geometrical parameters of the unit cell (e.g., the thickness of the dielectric spacer and the array periodicity). Moreover, we demonstrated that supervised training of the c-AAE network allows adding physics-driven regularization to compressed design space during the training phase, which in turn leads to better GO searches. To showcase the performance of the proposed AAE-based GO technique, we optimized the thermal emitter design with two different methods: (i) the c-AAE network coupled with a DE optimizer (c-AAE + DE) and (ii) DE optimization utilizing the compressed design space with physics-driven regularization (c-AAE + rDE).

Section 2 describes the main optimization problem under consideration, while Section 3 introduces the main data generation framework. Specifically, Section 3 describes the structure, training, and design generation process based on conditional AAEs, as well as rapid efficiency estimation process via pretrained conditional convolution NNs. Within Section 4, we focus on the GO scheme based on the developed AAE framework. Section 5 showcases the concept of physics-driven compressed space engineering via supervised training of the conditional AAE network, as well as demonstrates the GO within the regularized compressed design space. Section 6 concludes the work.

2 Optimization problem

Without loss of generality, we focus on the optimization of thermal emitters for thermophotovoltaics (TPVs) utilizing GaSb photovoltaic (PV) cells with a working band ranging from $\lambda_{\min} = 0.5 \mu\text{m}$ to $\lambda_{\max} = 1.7 \mu\text{m}$ (shaded area in Figure 1a). To maximize the generation of electric power from the TPV system, the emissivity of the thermal emitter should maximize in-band radiation (the red shaded area in Figure 1a) and minimize the out-of-band radiation (the blue shaded area in Figure 1a). Hence, the emissivity of the ideal thermal emitter is a step function with $\varepsilon(\lambda_{\min} \leq \lambda \leq \lambda_{\max}) = 1$ and zero elsewhere (depicted as the dashed blue contour in Figure 1a). Recently, gap-plasmon structures have been proposed as a viable solution to the thermal emission reshaping problem [44–47]. However, due to the simplistic, nonoptimal antenna designs, the efficiency of such thermal emitters has been limited.

Within this work, we consider a thermal emitter comprising the titanium nitride (TiN) back reflector [48], a silicon nitride (Si_3N_4) spacer, and a TiN plasmonic antenna in the top layer (Figure 1b). The main goal of the optimization is to determine the shape/topology of the top antenna, as well as the optimal configuration of the entire device, i.e., its 2D periodicity and spacer thickness that would drive the spectral emissivity to the step-like emissivity of the ideal emitter. Here, we do not focus on the details of the TO technique; instead, the paper is centered around the AAE-assisted global optimization framework. More details on the TO used for training set generation can be found in our prior work [43].

For assessing the performance of our designs, we define the efficiency of the thermal emitter as a product of in-band (eff^{in}) and out-of-band (eff^{out}) efficiencies, which is given as follows:

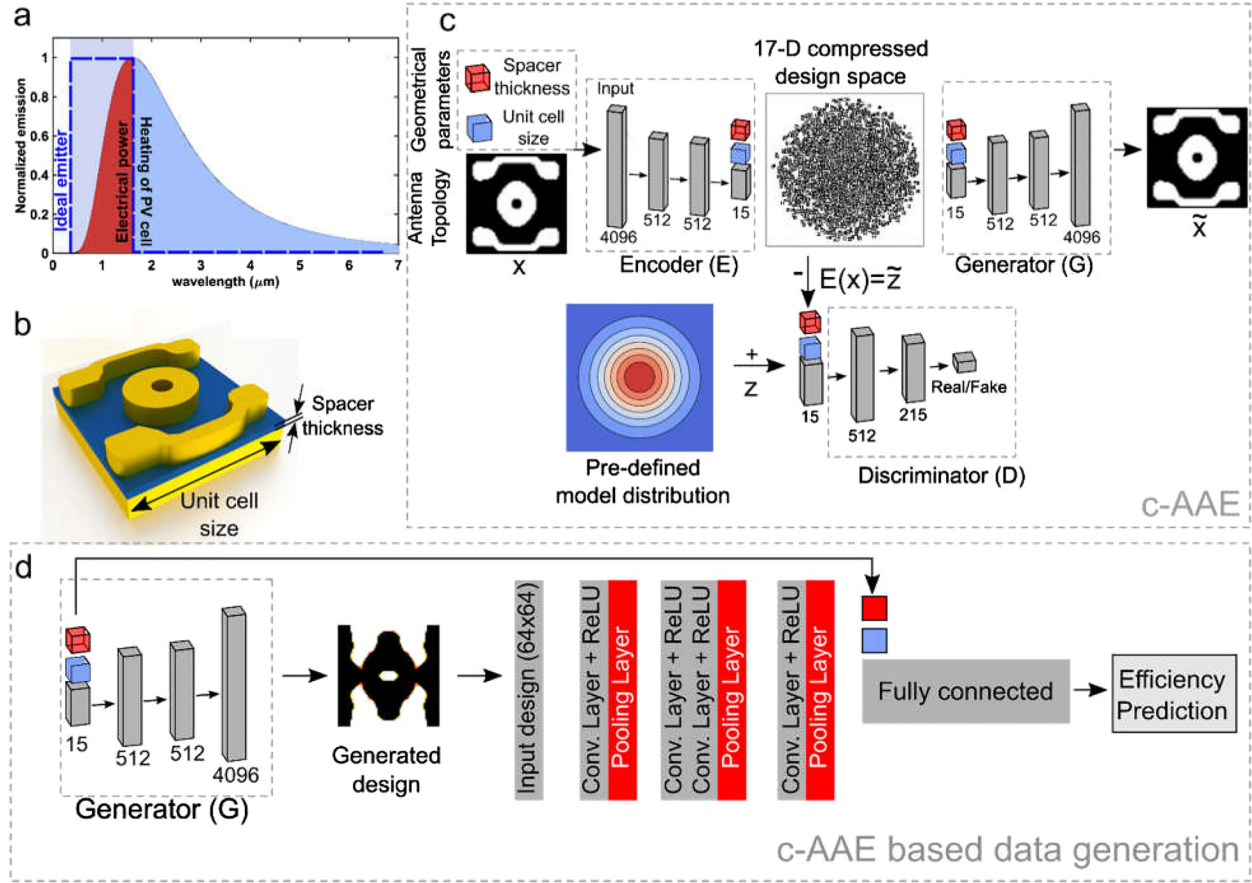


Figure 1: Conditional adversarial autoencoder-based data generation.

(a) Blackbody radiation of a bare heater (solid black curve) corresponding to the emission of a blackbody at 1800 C. The gray rectangular region highlights the GaSb photovoltaic (PV) cell working band. Only in-band radiation is converted into electrical power (red area), while out-of-band radiation causes unwanted heating of the PV cell (blue area). The dark blue dashed contour corresponds to an ideal thermal emitter's emissivity/absorption spectrum. (b) The gap-plasmon thermal emitter design with a metallic TiN back reflector, a Si_3N_4 dielectric spacer, and a top TiN plasmonic antenna. The topology optimization (TO) aims to optimize the top layer with a TiN/air mixer to match the step function emissivity pattern of an ideal emitter. (c) Training of the c-AAE network: Along with the antenna topology, the c-AAE is trained on a conditional vector with geometrical parameters of the unit cell (unit cell size, spacer thickness). (d) c-AAE-assisted rapid design generation: The trained G network is coupled with a conditional Visual Geometry Groupnet (VGGnet) type network (c-VGGnet) for rapid efficiency estimation. The geometrical parameters of the unit cell are used as constrained labels for generation of the designs, as well as estimation of the efficiencies. c-AAE, conditional AAE; AAE, adversarial autoencoder.

$$eff = eff^{in} \cdot eff^{out},$$

here

$$eff^{in} = \frac{\int_{\lambda_{min}}^{\lambda_{max}} \varepsilon(\lambda) B_{\omega}(\lambda, T) d\lambda}{\int_{\lambda_{min}}^{\lambda_{max}} B_{\omega}(\lambda, T) d\lambda},$$

$$eff^{out} = \frac{\int_{\lambda_{max}}^{\infty} \varepsilon_{\text{TiN}}(\lambda) B_{\omega}(\lambda, T) d\lambda}{\int_{\lambda_{max}}^{\infty} \varepsilon(\lambda) B_{\omega}(\lambda, T) d\lambda}.$$

where the Planck law, $B_{\omega}(\lambda, T) = 2hc\lambda^{-3} (e^{hc/(\lambda k_B T)} - 1)^{-1}$, gives the spectral radiance of the blackbody at a given temperature T and wavelength λ ; the fundamental constants include the Planck constant h , the Boltzmann constant k_B , and the speed of light in free space c . In (1) $\varepsilon(\lambda)$ and $\varepsilon_{\text{TiN}}(\lambda)$ denote the spectral emissivities of the optimized

- (1) emitter and a bare TiN back reflector, respectively, T is the working temperature of the emitter, wavelengths λ_{min} , λ_{max} are, respectively, the lower and upper bounds of the PV cell operation band.

eff^{in} is an in-band radiance of the emitter normalized to the in-band radiance of an ideal emitter at 1800°C, while out-of-band efficiency eff^{out} is defined as a ratio of the out-of-band radiance of a back reflector and radiance of the TO design. Such definition of the out-of-band efficiency is dictated by the fact that the response of the gap-plasmon structures in the long-wavelength limit is fully determined by the material properties of the back reflector, and out-of-band emissivity is fundamentally limited by the optical losses of TiN.

3 Conditional AAEs for rapid design generation

To include all the design parameters into the optimization framework, we couple the c-AAE network with TO (Figure 1c) [49]. The c-AAE network is a generative model, which consists of the encoder (E), the decoder/generator (G), and the discriminator (D). The E network is coupled with G network aiming at compression and decompression of the input design (x) through the so-called compressed design space. During the training phase, the E and G networks are trained to minimize the reconstruction loss between input (x) and the generated (\tilde{x}) designs by forming the 17-dimensional compressed design space. After the training process, the G network can be used to generate new designs based on the input compressed space vector $\tilde{z} = E(x)$, which is a 15D coordinate vector appended with two conditional labels l (unit cell size and spacer thickness). The dimensionality of the compressed space and conditional vector is defined by the main objective of the optimization problem. It can be further enlarged according to the requirements of the problem under consideration. The regularization of the compress design space is achieved through the adversarial training process via coupling to the D network. The D network is trained to distinguish between samples from the compressed design space distribution $q(\tilde{z})$ and the predefined model $p(\tilde{z})$. During the training process, the E network is trained to reshape the compressed design space such that the D network cannot distinguish between samples generated from the compressed design space and predefined model. The adversarial training forces the compressed design space to have the same data distribution as the user-defined model $p(\tilde{z})$. So, the main goal of the E and G networks is to learn the main geometrical features of the antenna designs in the training set, while the D network assures the regularization of the formed compressed design space defined by the predefined model. This is achieved via the optimization of the AAE network weights (E , G , and D networks) during the training phase by minimizing the following loss function:

$$L = L_{\text{adv}} + L_{\text{rec}} \quad (2)$$

The regularization of the compressed space is achieved via a minmax game between E and D networks which aims to minimize the adversarial loss term L_{adv} [49]:

$$L_{\text{adv}} = \min_E \max_D [\log(D(E(x), l)) + \log(1 - D(\tilde{z}, l))] \quad (3)$$

while the training of the E and G networks is done through to minimize the reconstruction loss L_{rec} [49]:

$$L_{\text{rec}} = -\min_{E,G} [\log p(x|G(E(x), l))]. \quad (4)$$

Once the c-AAE is trained, the G network can be used as a separate generative network that samples the new thermal emitter designs based on the input compressed design space vector and the geometrical parameters of the unit cell (Figure 1d). To rapidly estimate the performance of the designs, we couple the G network with a c-VGGnet [50] that estimates the efficiency of the designs based on the input binary image of the antenna (top view), thus avoiding time-consuming full-wave simulations altogether. To realize the conditional estimation, the conditional vector is coupled to the first fully connected layer after the feature extraction part of the convolutional neural networks (CNN). We give the details on the c-AAE and c-VGGnet network architecture in the supplementary materials.

The initial training set for the c-AAE network is obtained by performing TO of the thermal emitter designs for different cases of the unit cell sizes and spacer thicknesses. Specifically, the TO optimization is used to generate 100 designs for each of the period-thickness combinations: the period was chosen to be 250, 280, and 300 nm and spacer thicknesses 30 and 50 nm, thus yielding 600 designs in total. To train the c-AAE network, we use the data augmentation technique employed in the study by Kudyshev et al. [43] and increased the training set up to 24,000 designs. The periodic nature of the thermal emitters allows us to expand the initial data set by applying lateral and rotational perturbations to the original design. Here, we expand the data set by applying 20 lateral shifts and 90° rotation to each design of the original set. The enlarged design set, as well as corresponding periodicity and spacer thickness labels, has been used to train the c-AAE network. The predefined model distribution is set to be Gaussian that is centered at the origin of 15D compressed space.

Figure 2a shows the adversarial (blue) and reconstruction (red) losses evolution of the c-AAE network as a function of training epochs. The reconstruction loss of the E and G coupled network decreases with the training and saturates at <0.1 value, which indicates that the G network can reconstruct input design correctly from the compressed design space. The adversarial loss decreases at the beginning of the training, which corresponds to the fact that the E network fails to generate the desired compressed space distribution at the early steps of the training. However, with the increasing number of the training epochs, adversarial loss saturates at relatively high values (~0.8), highlighting the ability of the E network to “fool” the D network and passing the sample from the compressed space as “real” through it. This fact indicates that the constructed compressed design space has data distribution close to the

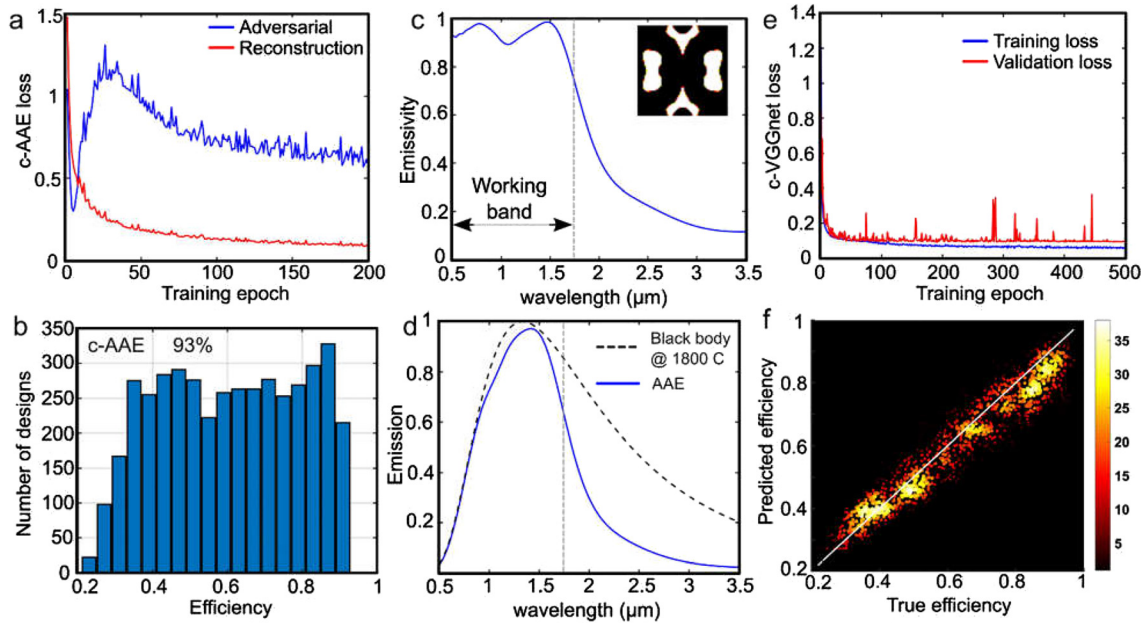


Figure 2: Training of c-AAE and c-VGGnet networks.

(a) Evolution of the adversarial (blue line) and reconstruction (red line) losses during the c-AAE network training. (b) The efficiency distributions of the design set generated by the c-AAE network via random sampling. (c) Spectral emissivity of the best design in the generated set. The dashed vertical black line shows the upper bound of the PV cell's working band. The inset shows the unit cell of the best design in the set (white color: TiN, black: air, unit cell period (x and y) 280 nm; spacer thickness, 30 nm). (d) Emission spectrum of the blackbody (dashed black line) and thermal emitter (solid blue line) at 1800°C. (e) Evolution of the training (blue line) and validation (red line) losses of the c-VGG predictive network during the training process. (f) The regression results performed by c-VGGnet on the testing data set. White line shows the regression line; the colormap shows the number of designs tested. c-AAE, conditional AAE; AAE, adversarial autoencoder; PV, photovoltaic; c-VGGnet, conditional VGGnet type network.

predefined one. Once the training of the c-AAE network is done, we have generated 4500 thermal emitter designs via random sampling of the compressed space coordinates. Specifically, the 300 designs for each of the combinations of the periodicity (from 200 to 280 nm with 20 nm step) and spacer thicknesses (30, 40, and 50 nm) have been generated. The efficiency of each design has been assessed by Finite-difference time-domain (FDTD) simulation (Lumerical FDTD solver). Figure 2b shows the statistics of the generated data set. The best design in the set has 93% efficiency of thermal emission reshaping and corresponds to 280-nm unit cell size and the 30-nm spacer thickness. Figures 2c and 2d show the corresponding absorption/emissivity spectra, as well as the gray body emission at 1800°C from the best design in the set (blue curves). The inset in Figure 2c shows the antenna design. The complex topology of the c-AAE antenna design enables 94% mean in-band absorption and substantially suppresses the out-of-band absorption spectrum (23% mean out-of-band absorption). This absorption behavior of the thermal emitter leads to its high in-band emission and the rapid decay of the out-of-band emission.

To perform the filtering of highly efficient design, as well as to avoid time-consuming full-wave analysis, the

c-AAE network is coupled to the c-VGGnet regression network, which estimates the performance of the design based on the input design and unit cell parameters (Figure 1d). The c-VGGnet regression network is trained on the design set, which is generated by the c-AAE network (Figure 2b). This training is done due to the high designs' variance and a broader range of the unit cell parameters vs. the TO generated set that allows for more efficient training of the c-VGG network. Employing the generated design set, unit cell parameters, and corresponding efficiency values as ground truth, the c-VGGnet is trained by using mean absolute percentage error loss function. Eighty percentage of the designs are used for training, while 20% is used for the validation loss estimation. The loss evolution during the training process is shown in Figure 2e. The figure demonstrates that the regression loss decreases and saturates at 9%, indicating that c-VGGnet is capable of retrieving the efficiency values based on the binary image of the design with high accuracy. For assessing the performance of the regression network, we calculate the coefficient of determination r^2 that quantifies the ability of the network to predict the variance of the true data. While in the ideal case r^2 should be equal to 100%, a sufficiently high value of r^2 (87%) is achieved.

Figure 2f shows the dependence of the predicted values by the network vs. the true values. In the ideal case, the point on the colormap should coincide with the regression line (white, solid line). The integrated scheme of the c-AAE generator with c-VGGnet opens the possibility to perform rapid prototyping and efficiency estimation of the metadevices (Figure 1d). This combination is a crucial step for the realization of various, ML-assisted GO schemes for highly constrained problems. The next two sections highlight one of the possible c-AAE-assisted GO. The proposed approach can be integrated with the other metaheuristic and/or gradient-based optimization frameworks. The next section describes the c-AAE-based GO technique based on differential evolution optimization.

4 AAE-assisted global optimization

Due to the nonconvex nature of many optimization problems, the design spaces usually correspond to the rapidly changing figure of merit (FOM) landscapes that make the brute-force approach inefficient in the quest for the global solution. This issue becomes even more significant for multiconstrained problems since the exhaustive search within highly dimensional parametric landscapes would be extremely resource heavy with no guarantee of retrieving the most optimal solution. Hence, it is crucial to develop a global optimization framework that is capable of using the “best of both worlds”: (i) efficiency and scalability of the c-AAE-based optimization framework and (ii) the ability of metaheuristic algorithms to perform global optimization searches most efficiently. To address the issues mentioned above, we develop a c-AAE-assisted GO scheme, which is capable of performing global search directly within the compressed design space.

Within this work, the DE algorithm [51] has been coupled to the c-AAE-based data generation approach for retrieving the global maximum inside the compressed design space. The DE framework is a population-based metaheuristic algorithm, which uses multiple agents to probe the solution space and evolutionally converge to global extremum. At optimization step, the positions of each of the agents in the population are updated by adding the weighted difference between two randomly selected agents from a given population to the agent with the best efficiency at the current iteration:

$$\tilde{z}_n^{i+1} = \tilde{z}_{\text{best}}^i + F(\tilde{z}_{r_1}^i - \tilde{z}_{r_2}^i) \quad (5)$$

Here, $\tilde{z}_{\text{best}}^i$ is a compressed design space coordinate of the agent with the best performance at i th iteration, F is a mutation parameter, $r_{1,2} = \text{rand}(1, N)$ are the random

indices, and N is a total number of agents within a given population.

The coordinates of the agents at the next optimization step are updated with (5) or kept unchanged. This choice is made with a binomial distribution by generating a random number, $r = \text{rand}(0, 1)$, and comparing it vs. a predefined recombination constant.

Within the developed optimization framework, the DE optimizer sends the set of compressed design space vectors $\tilde{z}_{1,N}^i$ to the c-AAE generator. The G network generates the designs, and the c-VGG-net estimates the efficiencies $\text{eff}_{1,N}^i$ of each design in the set. Once these efficiencies are sent back to the DE optimizer, the coordinates of the agents are updated at the next optimization step employing the algorithm shown above. At the end of the DE optimization run, we use the best $\tilde{z}_{\text{best}}^{\text{final}}$ to generate the antenna design and retrieve the corresponding unit cell configuration. For this, we take two last elements of $\tilde{z}_{\text{best}}^{\text{final}}$ corresponding to the unit cell size and spacer thickness, encoded during the constrained training of the c-AAE network. The c-AAE-based GO framework assures an extremely flexible framework that addresses highly constrained optimization problems by enlarging the compressed design space with a larger number of conditional labels during the c-AAE training. Most importantly, the c-VGGnet regression network removes the need for time-consuming full-wave simulations at the optimization search stage. Additionally, since the proposed approach uses the global optimizer as a black box, the developed c-AAE-assisted framework can be coupled to any global optimization techniques.

The DE optimization is implemented using the SciPy library [52]. The total population size of the DE optimizer is set to 20 agents, with a maximum iteration number of 80. The mutation and recombination coefficients are set to 0.5 and 0.7, respectively. The optimization objective function aims at minimizing the value of $1 - \text{eff}$. Figure 3b shows the typical conversancy curve of the c-AAE-assisted DE optimization (cAAE + DE). The DE optimization starts with the relatively high value of the objective function. However, with the evolution of the optimization, the DE algorithm converges to better efficiencies. The optimization stops when the objective function gets saturated or the maximum iteration is reached. We have performed 60 c-AAE + DE optimization runs. Figure 3c depicts the statistics of the efficiencies obtained from the runs. The analysis shows that the c-AAE + DE approach assures much better performance in comparison with the set generated directly from the c-AAE network. The mean efficiency of the distribution is 85%, while the best obtained design has 95.9% efficiency vs. 93% of the c-AAE set (Figure 2b). Figure 3d

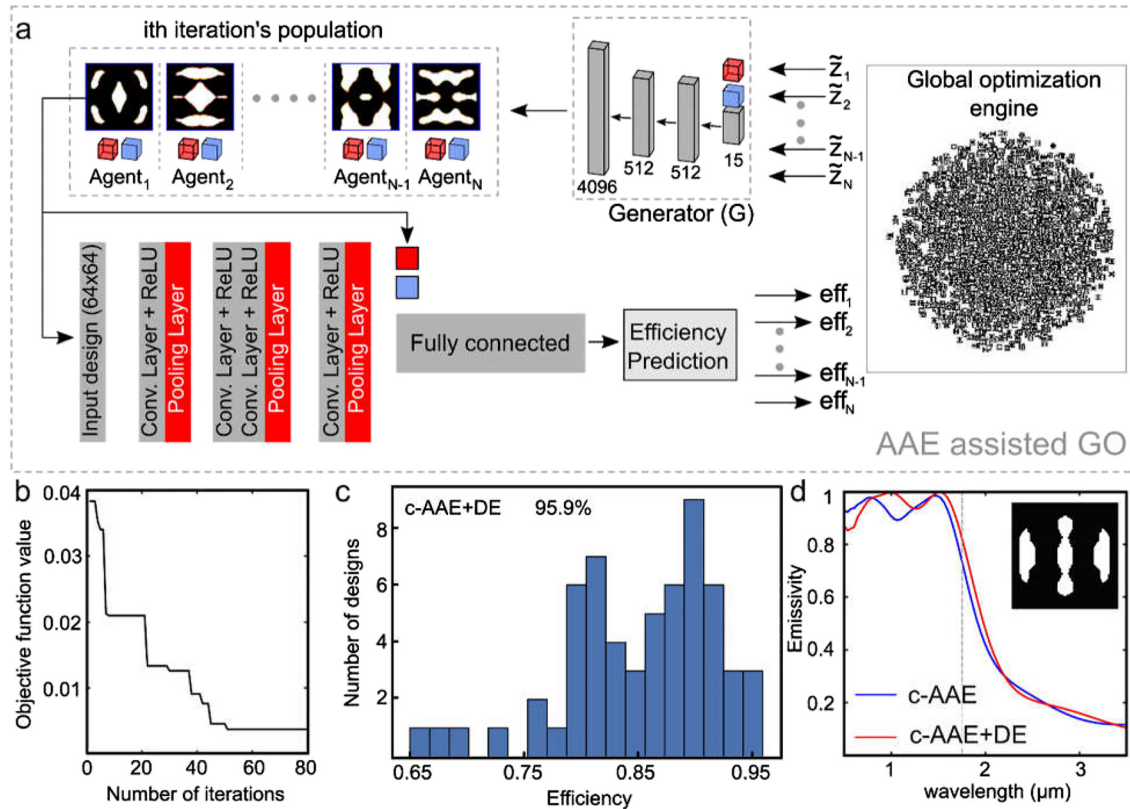


Figure 3: c-AAE-assisted global optimization.

(a) Scheme of the c-AAE based GO algorithm. The GO engine is used as a black box that generates the agents' coordinates within the compressed design space and pass them to the generator of the c-AAE network. The generator samples the design, and the corresponding efficiencies are rapidly assessed via the c-VGGnet network. These efficiencies are returned to the GO engine to update the positions of the agents at the current iteration. (b) Conversancy plot of the c-AAE + DE optimization framework. (c) Efficiency distribution for 60 designs generated by the c-AAE + DE. The legend indicates the maximum efficiencies obtained via random search within the compressed design space (93%, gray) and by GO (95.9%, black). (d) Spectral emissivity/absorption of the best designs generated by random search (blue) and GO (red). Thin vertical dashed line shows the upper bound of the PV cell's working band. The inset depicts the best unit cell design in the set (white color: TiN, black: air; unit cell period (x and y), 190 nm; spacer thickness, 45 nm). c-AAE, conditional AAE; AAE, adversarial autoencoder; PV, photovoltaic; DE, differential evolution; GO, global optimization.

shows the absorption spectra of the best design in the set (red line), while the blue curve indicates the efficiency of the best design generated directly from the c-AAE network (93%). The inset illustrates the best design generated with the c-AAE + DE approach. The c-AAE + DE optimization framework leveraged on the connection of the shape/topology of the metadevices and their optical responses. However, it is highly desirable to construct the physics-driven compressed design space by incorporating available knowledge on the performance of the device into the training process. By doing so, it is possible to regularize the compressed design space for more efficient GO searches. The next section describes the physics-driven compressed space engineering framework and demonstrates the performance of the GO within such design domains.

5 Physics-driven compressed space engineering

The training of the c-AAE network on the TO data set leads to the compressed design spaces, constructed based on geometrical features of the metadevices, omitting the available information regarding the optical responses of the designs. However, using available information about the essential physics of the metadevices, it is possible to preengineer the compressed design space for improved performance of the GO search. Such regularization of the compressed design spaces can be introduced by choosing the physics-driven predefined model of the c-AAE network connected with the FOMs of the metadevices.

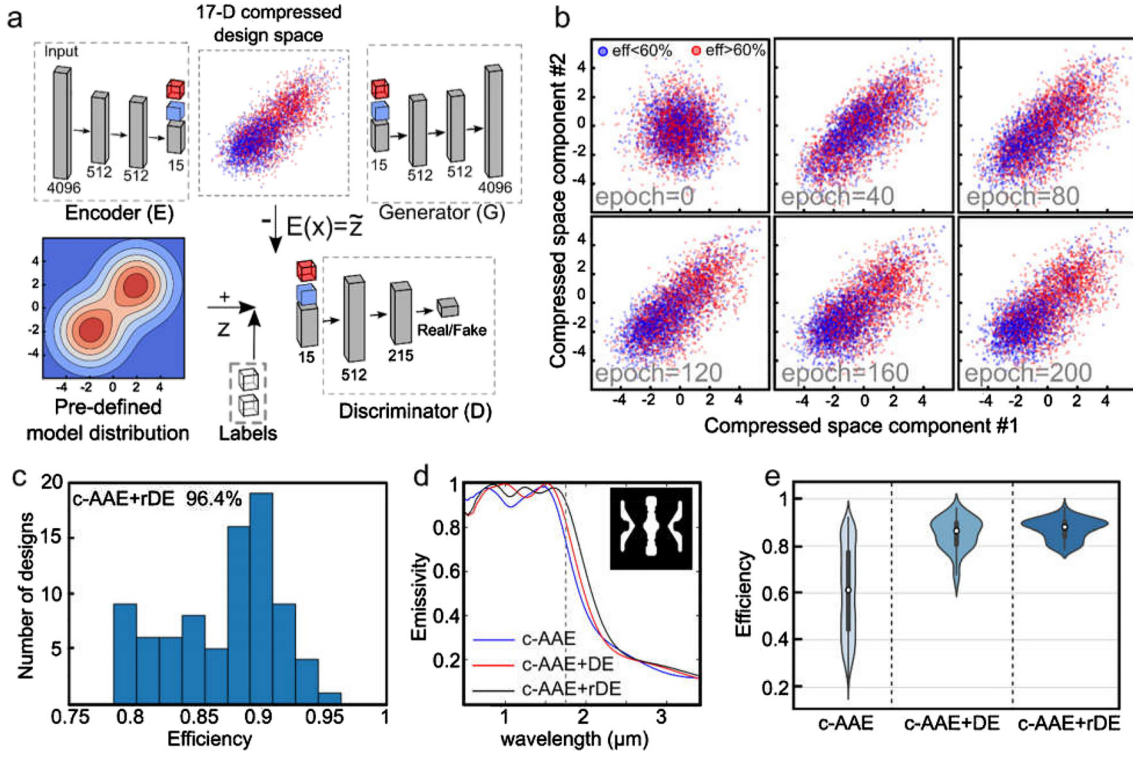


Figure 4: Physics-driven compressed design space engineering for GO.

(a) Scheme of the supervised training of the c-AAE network for physics-driven compressed space construction. The predefined model is set to be a combination of two 15-D Gaussian distributions symmetrically shifted from the origin. The additional “hot” binary vectors are used during the training for regularization of the compressed space during training. (b) Evolution of the compressed design space within the training process. The scatter plots show the distribution of the designs used in the training within 2D plane cut along the first two coordinates of the 17D compressed space. The blue markers depict the LE class data; red markers show the distribution of the HE class. (c) Efficiency distribution of 80 designs globally optimized with the c-AAE + rDE. (d) Spectral emissivity/absorption of the best designs generated by random search (blue dashed), c-AAE + DE (solid red), and c-AAE + rDE (black solid). The thin dashed vertical line shows the upper bound of the PV cell’s working band. The inset shows the unit cell of the best design in the set (white color: TiN, black: air, period (x and y) 290 nm, spacer thickness 45 nm). (e) The “violin plot” of three design sets generated by, (i) random search within unregularized compressed design space (left gray pattern, c-AAE), (ii) DE optimization within the unregularized compressed design space (center light blue pattern, c-AAE + DE), and (iii) DE optimization within the regularized compressed design space (right dark blue pattern, c-AAE + rDE). LE, low efficiency; HE, high efficiency; c-AAE, conditional AAE; AAE, adversarial autoencoder; PV, photovoltaic; DE, differential evolution; GO, global optimization.

This technique can be realized based on the supervised training of the c-AAE network by adjusting the predefined model and passing an additional binary regularization vector into the D network at the training stage (see, Figure 4a). For demonstrating the physics-driven compressed space engineering, we use two 15D Gaussian distributions in a predefined model and introduce an additional 2D hot vector as a label to the D network (Figure 4a). In more detail, the designs in the training data set are divided into two classes based on their efficiencies: (i) high efficiency (HE, $eff > 60\%$) and (ii) low efficiency (LE, $eff < 60\%$) classes. The predefined model distribution of the c-AAE model has been defined as follows:

$$z_i = \begin{cases} N(\mu = 2, \sigma^2 = 1), & eff_i > 60\% \\ N(\mu = -2, \sigma^2 = 1), & eff_i < 60\% \end{cases}, \quad (6)$$

here, z_i is a sampling of the predefined model for i th design in the set and $N(\mu, \sigma^2)$ is the random normal distribution with mean μ and variance σ^2 .

With this approach, we construct the compressed design space, with two clusterization regions separated according to the two efficiency-level classes. By applying the DE optimization within the HE region, we can obtain the thermal emitters with better efficiencies. Figure 4b shows the evolution of the compressed design space throughout the training process. The figure shows two first coordinates of the 17-D compressed design space. The red markers correspond to the HE class, while the blue ones represent the LE designs. Initially, all designs are sampled as a mixture of HE and LE classes. With the training, the clusterization of both classes progressively appears. The final state of the

compressed design space is shown in Figure 4b (epoch = 200). Once the training is done, the DE optimization is applied to the HE designs' region of the compressed space. We perform 80 runs with c-AAE + rDE optimization technique. Figure 4c shows the efficiency statistics of the optimized designs. The additional regularization leads to better performance of the DE optimizer vs. the unregularized case (c-AAE + DE). Thus, the c-AAE + rDE ensures 87% mean efficiency, with the best design in the set providing 96.4% efficiency of thermal emission reshaping. Figure 4d shows the emissivity spectrum of the best design in the set. The optimized design delivers 96% of in-band emissivity and substantially suppresses the out-of-band emission. The inset shows the design of the best thermal emitter in the set.

Figure 4e shows the back-to-back comparison of the efficiency distributions of three c-AAE-based optimization frameworks, (i) the set generated from c-AAE by random search (gray), (ii) c-AAE + DE (light blue) case, as well as (iii) c-AAE + rDE (dark blue). A sampling of the emitter designs from the random search (the c-AAE network with no postselection) leads to a broad range of the efficiencies with a mean efficiency of 61% and a maximum efficiency of 93%. It can also be seen that the efficiencies are uniformly distributed along the entire range (almost uniform width of the shadowed area, left subplot in Figure 4e). In contrast, the GO performed inside the unregularized compressed design space enables the design generation with much better efficiencies distribution with data concentrated within [65%, 95.9%] efficiency range and the maximum designs sampled around 90% efficiency (light blue pattern, central subplot in Figure 4e). The regularization of the compressed space coupled with the GO search leads to even better efficiencies distribution within [79%, 96.4%] (dark blue pattern, right subplot in Figure 4e). This analysis clearly shows that the physical regularization of the compressed design space allows adapting the design space configuration to perform a better GO search.

It is important to compare the proposed global optimization vs. previously demonstrated AAE-based "local" optimization frameworks (AAE + TO and AAE + VGGnet) [43]. The first approach (AAE + TO) utilizes a random sampling of the designs from the AAE network and then uses them as the initial condition for the TO refinement. This approach generated a thermal emitter design with the highest efficiency (97.9%) while taking ~31 min per design [43]. Such a relatively high computational time was the result of time-consuming design refinements performed via additional TO iterations. We note that the computation time would be even higher in the case of multiobjective

optimization. This leads to poor scalability of the approach for multiobjective optimization. The second approach (AAE + VGGnet) relies on the VGGnet network-based filtering of the highly efficient, robust designs within the randomly sampled AAE design set. While the second approach generated the thermal emitter designs with an efficiency of 95.5%, its computational cost was extremely low (1.2 s per design) [43].

The AAE + DE and AAE + rDE approaches generate the designs with the efficiencies of 95.9 and 96.4%, respectively, while the computational cost for both of these new GO methods is 14 min per design. Thus, both of the new methods provide higher efficiencies in comparison to the random sampling approach based on the AAE + VGGnet method and generate the designs 2.2 times faster than AAE + TO. The optimization efficiency and the speedup of the AAE + rDE method could be further increased by choosing the different types of global optimizers, as well as through more comprehensive physics-driven regularization of the compressed design space. Such flexibility of the proposed c-AAE-based global optimization framework might be instrumental for addressing multiobjective optimization problems with a high number of constraints. All the information on the computation environment is provided in Appendix 3. Detailed information regarding the AAE-assisted local optimization frameworks (AAE + TO, AAE + VGGnet) can be found in the study by Kudyshev et al. [43].

6 Conclusion

In conclusion, we have developed a global optimization framework utilizing a c-AAE network that can be applied to a wide range of highly constrained optimization problems in nanophotonics and plasmonics, as well as in biology, chemistry, and quantum optics. We show that by applying the differential evolution optimization directly to the compressed design space, it is possible to achieve efficient optimization of the metadevices with complex topology. We numerically demonstrate advanced compressed space engineering by utilizing physics-driven regularization of the compressed design space via supervised training of the c-AAE network. The proposed physics-driven design space compression leads to significant improvement in the GO search. We also show that physics-driven regularization of the compressed design space leads to a more intuitive way of performing the GO search within the compressed space, which, in turn, leads to the almost perfect performance of the optimized metadevices.

Preengineering of the compressed design spaces of metastructures can be used in combination with diverse ML algorithms such as principal component analysis [53] and cluster analysis [28] both to retrieve the best possible solution of the problem and to gain hidden knowledge about the physics of the metastructure with complex topologies. For example, analyzing the eigenmodes of the structures sampled from the high-efficiency cluster, it is possible to gain additional intuition regarding the electrodynamic mode components that lead to the optimal metadevices. This technique would allow us to generalize the physics requirements to the device design for achieving the best possible performance and reconstruct the antenna designs based on the first principles approach to the problem.

Author contribution: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: The authors acknowledge partial support from the NSF ECCS award “Machine-Learning-Optimized Refractory Metasurfaces for Thermophotovoltaic Energy Conversion”, Army Research Office MURI award no. W911NF-19-1-0279 and DARPA/DSO Extreme Optics and Imaging (EXTREME) Program award no. HR00111720032.

Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

Appendix 1: c-AAE for design production

An adversarial autoencoder (AAE) consists of three coupled NNs: the encoder, decoder/generator, and discriminator. Figure A1 shows a detailed description of the neural networks.

Encoder

The encoder takes a 4096-dimensional vector (that corresponds to a 64×64 binary design pattern) as an input. We use two fully connected layers as the hidden layers of the encoder and a 17-neuron layer as an output layer of the encoder so that each of the hidden layers has 512 neurons. The last two elements are the geometrical parameters of the unit cell used as a conditional label, while the rest is a coordinate of the 15D compressed design space. For hidden layers, the rectified linear unit (ReLU) activation function is used, and one batch

normalization layer is coupled to the second linear layer.

Decoder

The decoder has the same architecture as the encoder but with the reversed sequence. The decoder generates a 4096-element output vector based on 17-dimensional input (15D coordinate vector + 2 conditional labels). For the output layer, we use the tanh activation function.

Discriminator

The discriminator takes a 17-dimensional latent vector as an input and performs binary classification (fake/real). Hence the output of the discriminator is one neuron. Here, we have used 2 hidden linear layers with 512 and 256 neurons. The activation function for two hidden layers is the ReLU and for the output layer is the sigmoid function.

The compressed space vector consists of 15D latent coordinate and additional 2 labels with geometrical parameters of the unit cell. The training of the conditional AAE network has been realized according to the previously proposed joint disentanglement technique [54].

Appendix 2: c-VGGnet structure

CNN takes 64×64 image of the design as an input and passes it through three hidden layers, which consist of convolutional layers with ReLU activation functions. Each hidden layer is followed by the max. pooling layer, which ensures the downsampling of the feature maps. The stack of convolutional layers is followed by one fully connected layer, which is paired with 2 conditional labels corresponding to the unit cell geometrical parameters. The base VGGnet architecture is followed by “linear” activation function with “mean squared error” loss function for efficiency prediction (regression). A detailed description of the VGGnet is shown in Figure A2.

Appendix 3: Time requirement for training set generation, training of the networks

The generation of the original 600 TO designs (training set) was performed on three cluster nodes in parallel,

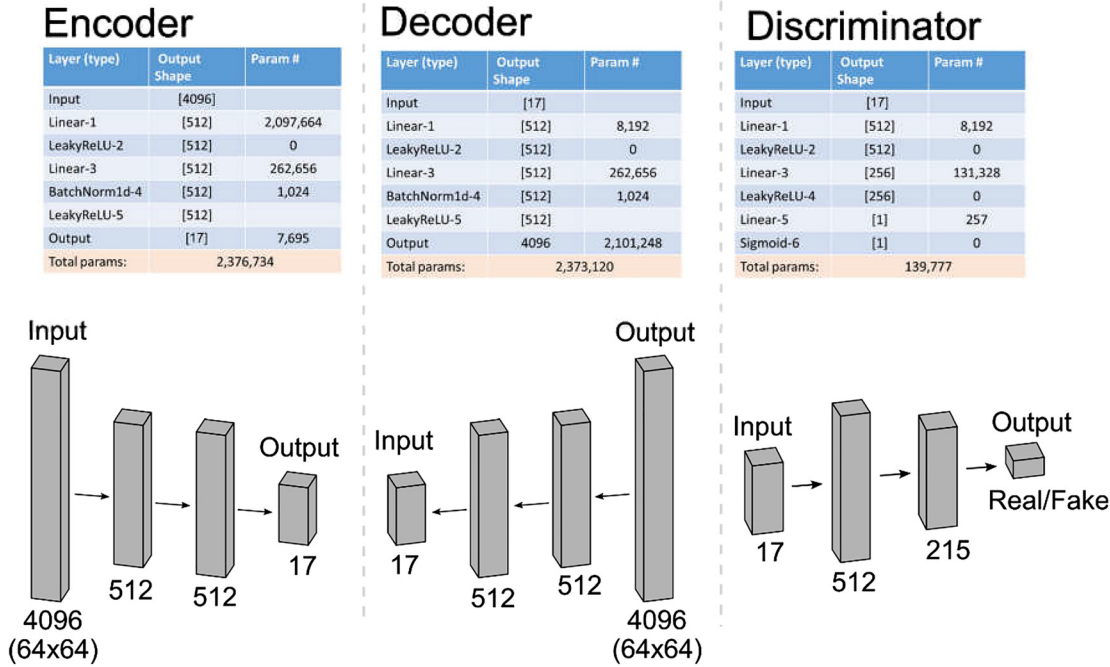


Figure A1: Structure of the c-AAE network.
c-AAE, conditional AAE; AAE, adversarial autoencoder.

Layer (type)	Output Shape	Param #
Input	[64,64,3]	
conv2d_1	[64, 64, 32]	896
activation_1	[64, 64, 32]	0
batch_normalization_1	[64, 64, 32]	128
max_pooling2d_1	[21, 21, 32]	0
dropout_1	[21, 21, 32]	0
conv2d_2	[21, 21, 64]	18496
activation_2	[21, 21, 64]	0
batch_normalization_2	[21, 21, 64]	256
conv2d_3	[21, 21, 64]	36928
activation_3	[21, 21, 64]	0
batch_normalization_3	[21, 21, 64]	256
max_pooling2d_2	[10, 10, 64]	0
dropout_2	[10, 10, 64]	0
conv2d_4	[10, 10, 128]	73856
activation_4	[10, 10, 128]	0
batch_normalization_4	[10, 10, 128]	512
conv2d_5	[10, 10, 128]	147584
activation_5	[10, 10, 128]	0
batch_normalization_5	[10, 10, 128]	512
max_pooling2d_3	[5, 5, 128]	0
dropout_3	[5, 5, 128]	0
flatten_1	[3200]+[2]	0
dense_1	[1024]	3277824
activation_6	[1024]	0
batch_normalization_6	[1024]	4096
dropout_4	[1024]	0
dense_2	[4]	4100
dense_3	[1]	5
Total params:		3,565,449

Figure A2: C-VGGnet structure.

which took 328 h. The training of the c-AAE network on topology optimized designs took 35 min, while the training of the c-VGGnet for efficiency regression took 48 min. All the training of the networks, as well as AAE + DE and AAE + rDE optimizations, were done on a cluster node with two 12-core Intel Xeon Gold “Sky Lake” processors @ 2.60 GHz (24 cores per node) and 96 GB of RAM. The initial TO was performed on three cluster nodes:

(i) two 12-core Intel Xeon Gold “Sky Lake” processors @ 2.60 GHz with 96 GB of RAM; (ii) two Haswell CPUs @ 2.60 GHz (20 cores per node) and 128 GB of RAM, and (iii) two Haswell CPUs @ 2.60 GHz (20 cores per node) and 256 GB of RAM. Direct full-wave simulation at each iteration was done in parallel, while the filtering, calculation of gradients, and material distribution updates were performed sequentially.

References

- [1] V. M. Shalaev, "Optical negative-index metamaterials," *Nat. Photonics*, vol. 1, pp. 41–48, 2007.
- [2] N. Yu and F. Capasso, "Flat optics with designer metasurfaces," *Nat. Mat.*, vol. 13, p. 139, 2014.
- [3] A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, "Planar photonics with metasurfaces," *Science*, vol. 339, no. 6125, pp. 12320091–12320096, 2013.
- [4] O. Quevedo-Teruel, H. Chen, A. Díaz-Rubio, et al., "Roadmap on metasurfaces," *J. Opt.*, vol. 21, p. 073002, 2019.
- [5] M. Song, Z. A. Kudyshev, H. Yu, A. Boltasseva, V. M. Shalaev, and A. V. Kildishev, "Achieving full-color generation with polarization-tunable perfect light absorption," *Opt. Mater. Express*, vol. 9, p. 779, 2019.
- [6] D. Sell, J. Yang, S. Doshay, R. Yang, and J. A. Fan, "Large-angle, multifunctional metagratings based on freeform multimode geometries," *Nano Lett.*, vol. 17, pp. 3752–3757, 2017.
- [7] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nat. Photonics*, vol. 12, pp. 659–670, 2018.
- [8] Z. Lin, V. Liu, R. Pestourie, and S. G. Johnson, "Topology optimization of freeform large-area metasurfaces," *Opt. Express*, vol. 27, p. 15765, 2019.
- [9] L. F. Frellsen, Y. Ding, O. Sigmund, and L. H. Frandsen, "Topology optimized mode multiplexing in silicon-on-insulator photonic wire waveguides," *Opt. Express*, vol. 24, p. 16866, 2016.
- [10] R. E. Christiansen, J. Michon, M. Benzaouia, O. Sigmund, and S. G. Johnson, "Inverse design of nanoparticles for enhanced Raman scattering," *Opt. Express*, vol. 28, p. 4444, 2020.
- [11] S. Jafar-Zanjani, S. Inampudi, and H. Mosallaei, "Adaptive genetic algorithm for optical metasurfaces design," *Sci. Rep.*, vol. 8, p. 11040, 2018.
- [12] D. Z. Zhu, E. B. Whiting, S. D. Campbell, D. B. Burckel, and D. H. Werner, "Optimal high efficiency 3D plasmonic metasurface elements revealed by lazy ants," *ACS Photonics*, vol. 6, pp. 2741–2748, 2019.
- [13] O. Ilic and H. A. Atwater, "Self-stabilizing photonic levitation and propulsion of nanostructured macroscopic objects," *Nat. Photonics*, vol. 13, pp. 289–295, 2019.
- [14] S. I. Bogdanov, A. Boltasseva, and V. M. Shalaev, "Overcoming quantum decoherence with plasmonics," *Science*, vol. 364, pp. 532–533, 2019.
- [15] A. Lenert, D. M. Bierman, Y. Nam, et al., "A nanophotonic solar thermophotovoltaic device," *Nat. Nanotechnol.*, vol. 9, pp. 126–130, 2014.
- [16] R. C. Devlin, A. Ambrosio, N. A. Rubin, J. P. B. Mueller, and F. Capasso, "Arbitrary spin-to-orbital angular momentum conversion of light," *Science*, vol. 358, pp. 896–901, 2017.
- [17] H. Wang, Y. Rivenson, Y. Jin, et al., "Deep learning enables cross-modality super-resolution in fluorescence microscopy," *Nat. Methods*, vol. 16, pp. 103–110, 2019.
- [18] Z. A. Kudyshev, S. I. Bogdanov, T. Isacson, A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, "Rapid classification of quantum sources enabled by machine learning," *Adv. Quantum Technol.*, vol. 3, no. 2000067, 2020, <https://doi.org/10.1002/qute.202000067>.
- [19] A. M. Palmieri, E. Kovlakov, F. Bianchi, et al., "Experimental neural network enhanced quantum tomography," *npj Quant. Inf.*, vol. 6, p. 20, 2020.
- [20] R. Santagati, A. A. Gentile, S. Knauer, et al., "Magnetic-field learning using a single electronic spin in diamond with one-photon readout at room temperature," *Phys. Rev. X*, vol. 9, p. 021019, 2019.
- [21] T. Zahavy, A. Dikopoltsev, D. Moss, et al., "Deep learning reconstruction of ultrashort pulses," *Optica*, vol. 5, p. 666, 2018.
- [22] W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," *ACS Nano*, vol. 12, pp. 6326–6334, 2018.
- [23] J. Peurifoy, Y. Shen, L. Jing, et al., "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, p. eaar4206, 2018.
- [24] I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, "Plasmonic nanostructure design and characterization via deep Learning," *Light Sci. Appl.*, vol. 7, p. 60, 2018.
- [25] P. R. Wiecha and O. L. Muskens, "Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3D nanostructures," *Nano Lett.*, vol. 20, pp. 329–338, 2020.
- [26] L. Jin, Y.-W. Huang, Z. Jin, et al., "Dielectric multi-momentum meta-transformer in the visible," *Nat. Commun.*, vol. 10, p. 4789, 2019.
- [27] C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, "Deep learning for accelerated all-dielectric metasurface design," *Opt. Express*, vol. 27, p. 27523, 2019.
- [28] Y. Kiarashinejad, M. Zandehshahvar, S. Abdollahramezani, O. Hemmatyar, R. Pourabolghasem, and A. Adibi, "Knowledge discovery in nanophotonics using geometric deep learning," *Adv. Intell. Syst.*, vol. 2, p. 1900132, 2020.
- [29] Y. Kiarashinejad, S. Abdollahramezani, M. Zandehshahvar, O. Hemmatyar, and A. Adibi, "Deep learning reveals underlying physics of light-matter interactions in nanophotonic devices," *Adv. Theor. Simulat.*, vol. 2, p. 1900088, 2019.
- [30] Y. Kiarashinejad, S. Abdollahramezani, and A. Adibi, "Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures," *npj Comput. Mater.*, vol. 6, p. 12, 2020.
- [31] I. Sajedian, J. Kim, and J. Rho, "Finding the optical properties of plasmonic structures by image processing using a combination of convolutional neural networks and recurrent neural networks," *Microsyst. Nanoeng.*, vol. 5, p. 27, 2019.
- [32] Z. Liu, D. Zhu, K. Lee, A. S. Kim, L. Raju, and W. Cai, "Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques," *Adv. Mater.*, vol. 32, p. 1904790, 2020.
- [33] H. Ren, W. Shao, Y. Li, F. Salim, and M. Gu, "Three-dimensional vectorial holography based on machine learning inverse design," *Sci. Adv.*, vol. 6, no. 16, eaaz4261, 2020. <https://doi.org/10.1126/sciadv.aaz4261>.
- [34] R. S. Hegde, "Deep learning: a new tool for photonic nanostructure design," *Nanoscale Adv.*, vol. 2, pp. 1007–1023, 2020.
- [35] K. Yao, R. Unni, and Y. Zheng, "Intelligent nanophotonics: merging photonics and artificial intelligence at the nanoscale," *Nanophotonics*, vol. 8, pp. 339–366, 2019.
- [36] Z. Liu, D. Zhu, S. P. Rodrigues, K. T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Lett.*, vol. 18, pp. 6570–6576, 2018.

- [37] S. So and J. Rho, "Designing nanophotonic structures using conditional deep convolutional generative adversarial networks," *Nanophotonics*, vol. 8, pp. 1255–1261, 2019.
- [38] W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy," *Adv. Mater.*, vol. 31, p. 1901111, 2019.
- [39] Z. Liu, Z. Zhu, and W. Cai, "Topological encoding method for data-driven photonics inverse design," *Opt. Express*, vol. 28, p. 4825, 2020.
- [40] W. Ma and Y. Liu, "A data-efficient self-supervised deep learning model for design and characterization of nanophotonic structures," *Sci. China Phys. Mech. Astron.*, vol. 63, p. 284212, 2020.
- [41] J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, "Free-form diffractive metagrating design based on generative adversarial networks," *ACS Nano*, vol. 13, pp. 8872–8878, 2019.
- [42] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," *Nano Lett.*, vol. 19, pp. 5366–5372, 2019.
- [43] Z. A. Kudyshev, A. V. Kildishev, V. M. Shalaev, and A. Boltasseva, "Machine-learning-assisted metasurface design for high-efficiency thermal emitter optimization," *Appl. Phys. Rev.*, vol. 7, p. 021407, 2020.
- [44] M. G. Nielsen, A. Pors, O. Albrektsen, and S. I. Bozhevolnyi, "Efficient absorption of visible radiation by gap plasmon resonators," *Opt. Express*, vol. 20, p. 13311, 2012.
- [45] F. Ding, Y. Yang, R. A. Deshpande, and S. I. Bozhevolnyi, "A review of gap-surface plasmon metasurfaces: fundamentals and applications," *Nanophotonics*, vol. 7, pp. 1129–1156, 2018.
- [46] J. T. Heiden, F. Ding, J. Linnet, Y. Yang, J. Beermann, and S. I. Bozhevolnyi, "Gap-Surface plasmon metasurfaces for broadband circular-to-linear polarization conversion and vector vortex beam generation," *Adv. Opt. Mater.*, vol. 7, p. 1801414, 2019.
- [47] W.-Y. Tsai, C.-M. Wang, C.-F. Chen, et al., "Material-assisted metamaterial: A new dimension to create functional metamaterial," *Sci. Rep.*, vol. 7, p. 42076, 2017.
- [48] H. Reddy, U. Guler, Z. Kudyshev, A. V. Kildishev, V. M. Shalaev, and A. Boltasseva, "Temperature-dependent optical properties of plasmonic titanium nitride thin films," *ACS Photonics*, vol. 4, pp. 1413–1420, 2017.
- [49] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, *Adversarial Autoencoders*, preprint at arXiv.org, arXiv: 1511.05644, 2015.
- [50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," preprint at arXiv.org, arXiv:1409.1556, 2015.
- [51] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, 1997.
- [52] P. Virtanen, R. Gommers, T. E. Oliphant, et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020.
- [53] D. Melati, Y. Grinberg, M. Kamandar Dezfouli, et al., "Mapping the global design space of nanophotonic components using machine learning pattern recognition," *Nat. Commun.*, vol. 10, p. 4775, 2019.
- [54] D. Polykovskiy, A. Zhebrak, D. Vetrov, et al., "Entangled conditional adversarial autoencoder for de Novo drug discovery," *Mol. Pharm.*, vol. 15, pp. 4398–4405, 2018.