Supplementary Information for

Deep Learning-Assisted Inverse Design of Metasurfaces for Active Color Image Tuning

Qiang Weng and Yanjun Bao*

Guangdong Provincial Key Laboratory of Nanophotonic Manipulation, Institute of Nanophotonics, College of Physics & Optoelectronic Engineering, Jinan University, Guangzhou 511443, China

*Corresponding Authors: Y.B. (yanjunbao@jnu.edu.cn)

Table of Contents

S1. Implement of the boundary loss L_b

Our proposed hexagonal nanopillar structure has 13 geometric degrees of freedom (DOFs): 12 vertex parameters ($x_1, x_2, ..., x_6, y_1, y_2, ..., y_6$) and a period P. For simplicity, we denote all these parameters as x_i (where i=1,2,…,13). Each of these parameters is normalized to a range between 0 and 1. Consequently, the input parameters for the trained forward neural network (FNN) are valid only within this domain.

When using the inverse adjoint neural network (IANN), if no boundary loss is applied, the resulting geometric parameters x_i may fall outside the valid range, leading to errors. To ensure that all x_i values remain within the boundaries of x_{min}=0 and x_{max}=1, we introduce a boundary loss component to the total loss function. The boundary loss is defined as:

$$\sum_{i=1}^{13} \mathrm{ReLU}(x_i - x_{max}) + \sum_{i=1}^{13} \mathrm{ReLU}(x_{min} - x_i) \tag{S1}$$

This equation penalizes any geometric parameters that exceed the specified boundaries.

Figure S1 illustrates a comparison of the geometric parameters obtained with and without the boundary loss. When the boundary loss is applied, all geometric parameters are confined within the range of 0 to 1. In contrast, without the boundary loss, some parameters fall outside this range, resulting in inaccuracies.

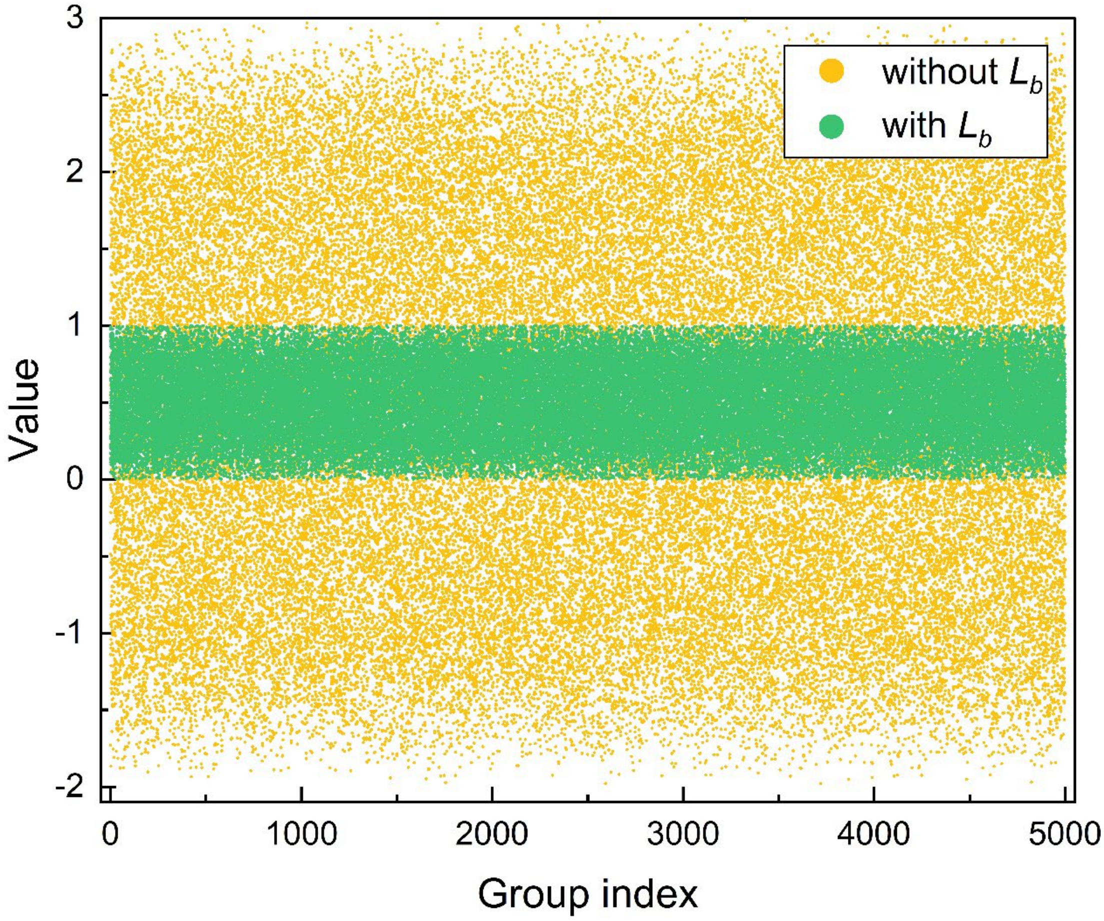

Figure S1. Comparison of 5000 sets of structural parameters obtained with and without boundary loss. The yellow circle indicates no boundary loss, and its value is outside the limit. The green circle is a bounded loss, and its value is limited between 0 and 1.

S2. Influence of Different Weights in the Multi-Objective Loss Function

In the inverse adjoint neural network (IANN), the loss is defined as the mean squared error (MSE) between the predicted XYZ values and the target X'Y'Z' values in the three environments, as

$$L = w_1 \cdot \left|\mathrm{XYZ}|_a\text{-}\mathrm{X}'Y'Z'|_a\right|^2 + w_2 \cdot \left|\mathrm{XYZ}|_w\text{-}\mathrm{X}'Y'Z'|_w\right|^2 + w_3 \cdot \left|\mathrm{XYZ}|_c\text{-}\mathrm{X}'Y'Z'|_c\right|^2 \quad (S2)$$

In the main text, the weight coefficients w_1=w_2=w_3=1 were used. To investigate the influence of different weights on the predicted images, we tested three different sets of weight coefficients, with the results shown in Figure S2. The results indicate that the pixel-level errors persist and do not improve, suggesting that our proposed hexagonal structures are unable to simultaneously match specific color combinations highlighted in the yellow circular domains.

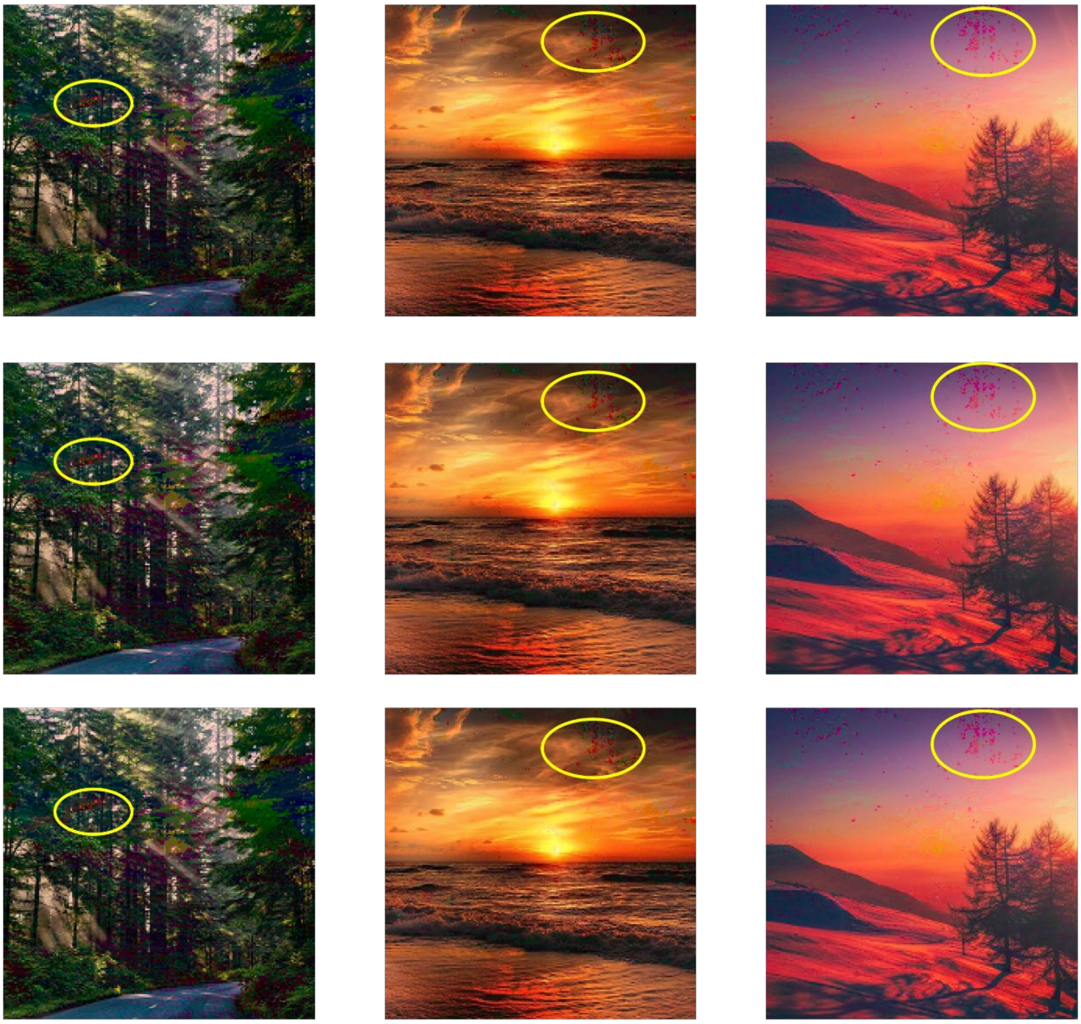

Figure S2. Predicted Images with different weighting coefficients of (a) w_1=1, w_2=1.5, w_3=1.5, (b) w_1=1.5, w_2=1.0, w_3=1.5 , (c) w_1=1.5, w_2=1.5, w_3=1.0.

S3. Structural Similarity Index Measure (SSIM) loss

Structural Similarity Index Measure (SSIM) is an index used to measure the similarity between two images. Different from the traditional pixel-level error indicators (such as mean square error, MSE), SSIM pays more attention to the structural information of the image and changes in brightness and contrast. The SSIM index calculates the similarity of brightness, contrast and structure, and synthesizes these three factors to get the overall similarity score. Specifically, for two images x and y ,SSIM is calculated as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{S3}$$

where:

- μ_x and μ_y are the mean luminance of images x and y respectively.
- σ_x^2 and σ_y^2 are the variances of images x and y, representing their contrast.
- σ_{xy} is the covariance between images x and y, reflecting their structural similarity.
- C_1 and C_2 are small constants used to avoid division by zero, typically defined as:

$$C_1 = (k_1L)^2, C_2 = (k_2L)^2 \tag{S4}$$

where L is the dynamic range of pixel values (e.g., L=255 for 8-bit images), and k_1 and k_2 are small constants, commonly set to k_1 = 0.01 and k_2= 0.03. Here, we explored the use of SSIM loss for active color image design. Figure S3 shows the predicted images generated using SSIM loss across different environments: air, water, and CS_2.

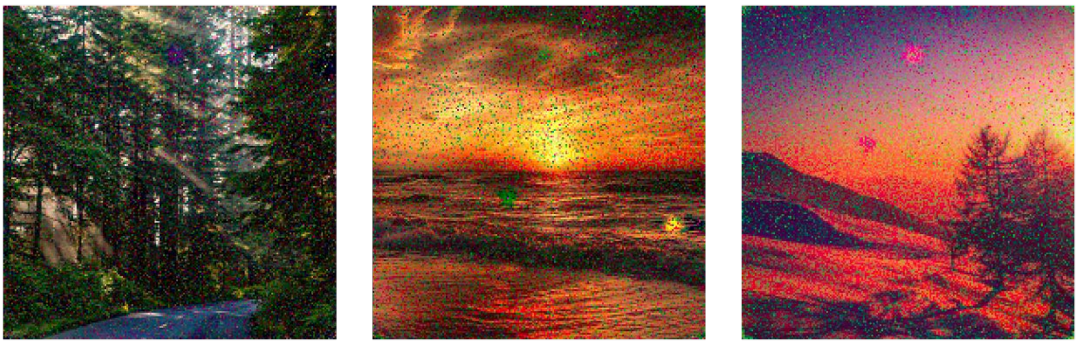

Figure S3. Predicted images using SSIM loss for active color image design (from left to right are air, water, CS_2 environment, respectively).

As shown in Figure S3, the use of SSIM loss led to pixel-level errors spreading across the entire images, resulting in an overall effect that was inferior to that achieved with

MSE loss. This may be attributed to the fact that SSIM is a measure of the overall perceived quality of the image rather than a pixel-by-pixel comparison. In contrast, MSE is designed to evaluate each pixel individually, allowing multiple structural parameters to be initialized in batches by the GPU to find the optimal structure for each pixel.