

# Contextual Hypernetwork for Adaptive Prediction of Laser-Induced Colors on Quasi-Random Plasmonic Metasurfaces

Thibault Girardin<sup>1,3</sup> (✉), Nathalie Destouches<sup>1,2</sup>, and Amaury Habrard<sup>1,2,3</sup>

<sup>1</sup> Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School,  
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT- ETIENNE, France

{thibault.girardin,nathalie.destouches,amaury.habrard}@univ-st-etienne.fr

<sup>2</sup> Institut Universitaire de France (IUF)

<sup>3</sup> Inria

**Abstract.** Laser processing is a rapid, versatile, and low-cost technology to print images on large surfaces. When applied to very thin films embedded with disordered metallic nanoparticles, known as quasi-random plasmonic metasurfaces, it generates colors that vary with the observation mode, making it valuable for visual security applications. Predicting these colors in different modes from the knowledge of laser processing parameters and the initial state of the metasurface can accelerate the industrialization process. However, there is no general physical model able to make this prediction accurately in various modes. In order to address this issue, this paper proposes a data-driven approach for learning deep models on experimental data able to predict the colors observed in different environments for a large range of laser processing parameters. We leverage a framework that learns jointly a shared latent space for multiple environments together with a contextual representation specific to each. This contextual representation is generated by an hypernetwork conditioned on an interpretable context vector. This context vector can be learned from few data allowing fast adaptation to new environments. This approach demonstrates that a single model can learn to predict a large range of colors across different environments. Its effectiveness is demonstrated through its ability to rapidly adapt to new scenarios with minimal data and to serve as an improved weight initializer for fine-tuning when larger datasets are available. Source code and datasets are available on Gitlab <sup>4</sup>.

**Keywords:** Deep Learning · Hypernetwork · Domain Adaptation · Nanoplasmonic · Color science.

## 1 Introduction

Metasurfaces are ultrathin optical surfaces made of periodic metallic or dielectric nanostructures that strongly interact with light, and have found applica-

---

<sup>4</sup> <https://gitlab.univ-st-etienne.fr/gt101872/ECML25-Hypernetwork-ColorPrediction-metasurface>

tions such as nano-antennas, sensors or optical filters [3,4,25]. Introducing different types of disorder into optical metasurfaces can enhance their performance or create unique optical properties [14]. The use of lasers to control the opto-geometrical properties of random optical metasurfaces provides the flexibility to shape their optical properties at the micrometer scale over large areas in a rapid process [27]. Laser processing is mainly used on plasmonic metasurfaces since metallic nanostructures can be reshaped and reorganized through photo-induced physical and chemical mechanisms [17,19]. Laser-induced self-organization mechanisms can also introduce order in initially random distribution of metallic nanostructures as demonstrated with continuous-wave [6,15], and femtosecond lasers [16]. The resulting quasi-random plasmonic metasurfaces feature dichroic colors that have been recently used to create multiplexed images observable under white light [7]. Image multiplexing is based on the ability to create sets of metasurfaces that show the same colors in some observation modes and different colors in others, and to organize these metasurfaces appropriately on the surface to display different images in the different observation modes. This laser-induced printing technology has great promise in visual security applications for the protection of secure documents from counterfeiting [5].

The industrial application of this technology requires the ability to reproduce on different substrates the laser-induced colors observed in different modes with an accuracy that is below the smallest color difference perceived by the human eye. Since the laser-induced colors depend on the initial state of the random plasmonic metasurface, and since this state may vary slightly over time due to uncontrolled variations in the fabrication process, it is useful to predict accurately the colors that can be produced on a given sample from a limited number of measurements. Unfortunately, due to the complex physical and chemical processes that occur in the metasurface during laser processing, physical models do not precisely predict the morphological transformations of the metasurface and thus the colors it may ultimately display in different observation modes.

One solution is to consider data driven models. Deep learning-based methods have been successfully used in nanophotonics to infer possible designs from given optical responses [11,12,18,21,22,30] or to predict optical properties [1,8,10,26]. In the context of color prediction, one solution has been developed in [20] with a deep learning model for predicting color spectra from laser parameters. However, this model is not able to generalize to new environment such as novel initial state of the substrate, requiring to learn a specific model for each setting.

In order to overcome this drawback, we propose in this paper a new framework for efficient prediction of laser-induced colors on plasmonic metasurfaces. Our contribution is three-fold. We first propose to learn a deep learning model that predicts the colors directly in the CIE LAB color space from laser parameters. This model is used to provide a shared latent space among training environments. Second, we use a contextual hypernetwork to adapt the shared representation to new environments. Inspired by previous works on adaptive dynamics [13], this hypernetwork is parametrized by a context vector modeling the peculiarities of a particular environment and that can be learned efficiently from

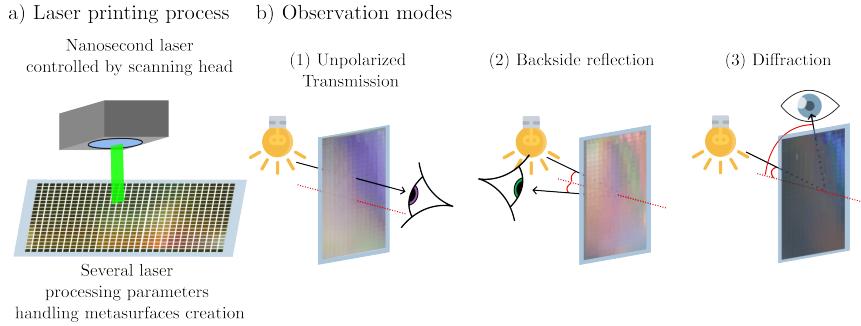


Fig. 1: Color production rationale. (a) Colors are produced by scanning a laser over the thin film. (b) Processed surface are observed in different modes such as transmission (1), specular reflection (2) or diffraction (3) displaying different ranges of colors.

few data. Finally, we provide an experimental study on real data showing the interest of the proposed setup for multi-environment color prediction.

The paper is organized as follows. The problem definition and the related work are introduced in Section 2. We present in Section 3 our hypernetwork-based deep learning architecture. Section 4 is dedicated to our experimental study. We conclude in Section 5.

## 2 Problem Definition and Related Work

### 2.1 Color Production Rationale

Quasi-random plasmonic metasurfaces of square shape and micrometer area are created by laser processing of a thin layer of TiO<sub>2</sub> containing silver nanoparticles [5]. Each square is laser printed with a specific set of laser parameters and the color of the squares is measured in different observation modes (see Fig. 1). Production rationale is detailed in Appendix Sec.2.

### 2.2 Color Prediction Problem

The color prediction problem consists in learning a model capable of mapping the produced color of a quasi random metasurface from  $d$  laser processing parameters. Colors are computed, with respect to the D65 illuminant, from the measured and normalized spectra in several observation modes which are expressed in CIE LAB space, a 3-dimensional perceptually uniform color space (See Appendix Sec.1 for further details on color theory and color spaces). The learning problem can then be formalized as follows. Let  $\mathcal{X} \subset \mathbb{R}^d, \mathcal{Y} \subseteq \mathbb{R}^3$  be respectively the space of laser processing parameters and the color coefficients in the CIE LAB space. We denote by  $\mathcal{D}_e$  a dataset of colors from  $\mathcal{X} \times \mathcal{Y}$  that can

Reference	LAB(50, -1, 52)	LAB(50, -1, 52)	LAB(50, -1, 52)	LAB(50, -1, 52)
Sample	LAB(50, 1.5, 52)	LAB(50, -3, 52)	LAB(45.8, -1.64, 52.3)	LAB(50, 15, 52)
$\Delta E94$	0.28	1.12	4.22	8.93

Fig. 2: Evolution of measured color shifts with  $\Delta E94$  for similar but different pair of colors modeled by their LAB coordinates, illustrating the necessity to be close to 1 to minimize perceived color differences (best viewed in color).

be obtained from a particular observation environment  $e$ . We denote by  $\mathcal{E}$ , the set of all possible environments, each environment corresponding to a particular experimental configuration where the colors are measured.

The objective is to learn a neural network model  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ , parameterized by a weight vector  $\theta$ . Given a dataset of  $N$  elements  $\mathcal{D}_e = \{(x_{e,i}, y_{e,i})\}_{i=1}^N$ , the loss function is defined with respect to the  $\Delta E94$  measure, a weighted RMSE loss allowing to measure human perceived color differences.

$$\mathcal{L}(\theta, \mathcal{D}_e) = \sum_{i=1}^N \Delta E94(y_{e,i}, f_\theta(x_{e,i})). \quad (1)$$

Illustrated in Fig. 2, the goal is to learn a model able to make target color prediction with a small  $\Delta E94$ . A value below or at least close to 1 corresponds to situations where the colors are indistinguishable by a human eye.

### 2.3 Related Work

Deep Learning is widely used to solve design and properties prediction problem in nanophotonics applications, mainly from simulated datasets. As highlighted in [2] in the context of plasmonic color prediction, learning from experimental data is difficult. Data acquisition is by nature costly and subject to experimental noise which may lead to small datasets of low quality. In this context, transfer learning has already been used for simulated optical properties [24] or inverse design prediction [32]. However, there is no widely known work in this community considering a learning process based entirely on experimental data and transfer learning. By the lack of simulation model for our random plasmonic metasurfaces, we aim at solving a fully experimental data-driven transfer problem.

In domain adaptation, feature-based methods such as DANN [28] or DA-DETR [31], aims at learning robust intermediate representations across different domains or tasks to improve transfer accuracy. While popular, these approaches are not directly adapted to our setting since we have a small number of input features that remain the same across the different environments. Instance-based adaptation methods such as WANN [23] learns a reweighting model to correct

the co-variate shift between domains or samples but this assumption is not directly adapted to our setting. Multi-task learning has the objective to deal with multiple tasks but adaptation to new tasks exists essentially for classification [29]. Meta-learning approaches such as [9], which aims to adapt to a new domain with few (gradient) adaptation steps, are very appealing in the context of low data regimes. Recently, CoDA [13] proposed a generalization of these approaches with the use of an hypernetwork allowing efficient adaptation with few samples to new domains in the context of dynamical models. In this paper, our objective is to adapt this strategy to our real problem of color prediction across different environments.

### 3 Hypernetwork for Contextual Adaptation

This section introduces the adaptation rule used to train our hypernetwork and how parameters are updated in the training and adaptation phase.

#### 3.1 Architecture and Adaptation Principles

Our model follows a similar strategy as CoDA [13]. The principle is to allow the color prediction model  $f_\theta$  to adapt to colors induced in different environments. The idea is to condition the model on observed samples from environment  $\mathcal{D}_e$ . This conditioning is done by using a network  $A_\vartheta$  parametrized by a set  $\vartheta$  containing weights variation for  $f_{\theta^e}$  to adapt to an environment  $e$  with respect to:

$$\theta^e = A_\vartheta(\mathcal{D}_e) = \theta^s + \delta\theta^e, \quad \vartheta = \{\theta^s, \{\delta\theta^e\}_{e \in \mathcal{E}}\}, \quad (2)$$

where  $\theta^s \in \mathbb{R}^{d_\theta}$  is a vector of shared parameters and  $\delta\theta^e \in \mathbb{R}^{d_\theta}$  corresponds to a vector of parameters specific to an environment  $\mathcal{D}_e$ . The set  $\vartheta$  represents the parameters to be learned by the network. The environment-specific parameters will be obtained by a contextual hypernetwork introduced in the next subsection.

#### 3.2 Contextual Hypernetwork

The Hypernetwork principle is illustrated in Fig.3. The idea is to estimate  $\delta\theta^e$  from the linear decoding of a context vector specific to an environment  $\mathcal{D}_e$  and denoted  $\kappa_e \in \mathbb{R}^{d_\kappa}$ . Let  $W \in \mathbb{R}^{d_\theta \times d_\kappa}$  the matrix parameter of the linear decoder, the adaptation problem can be rewritten as:

$$A_\vartheta(\mathcal{D}_e) = \theta^s + W\kappa_e, \quad \vartheta = \{W, \theta^s, \{\kappa_e\}_{e \in \mathcal{E}}\}. \quad (3)$$

We consider  $d_\kappa \ll d_\theta$ . This allows us to learn a low-dimensional subspace of specific parameters which is better suited to limit overfitting, in particular in low data regimes. From an adaptation point of view, considering that the shared parameters  $\theta^s$  and  $W$  have been learned, adapting to a new environment  $e$  implies to learn only  $\kappa_e$  which represents very few parameters to infer to adapt the color prediction model to this environment.

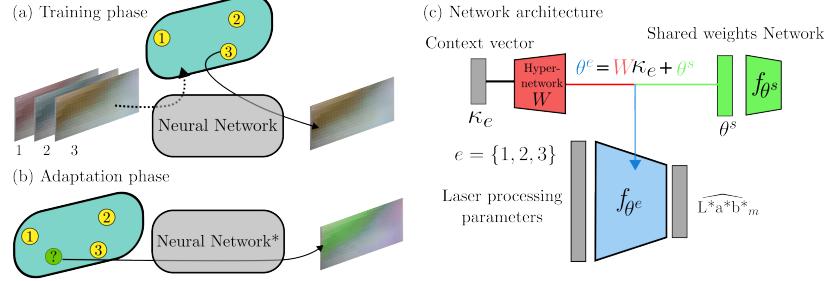


Fig. 3: Principle of the approach. During the training phase (a) the model learns parameters from all the training environments. In the adaptation phase, the model can explore the induced latent space freely to find the best context vector coordinates to adapt to a new task. Adapting to a given task is always obtained by combining the shared parameters between all the environments with the addition of a specific variation conditioned by the context vector  $\kappa_e$  (c).

### 3.3 Learning and Adaptation Loops

In this part, we introduce two important aspects of the method: the training phase and the adaptation one.

From given training samples of a set of training environments  $\mathcal{E}_{tr} \subseteq \mathcal{E}$ , the training phase aims at learning the shared initial weights, the shared decoder and the context vectors according to the following optimization problem in Eq. 4:

$$\min_{\theta^s, W, \{\kappa_e\}_{e \in \mathcal{E}_{tr}}} \sum_{e \in \mathcal{E}_{tr}} (\mathcal{L}(\theta^s + W\kappa_e, \mathcal{D}_e) + \lambda_\kappa \|\kappa_e\|_2^2 + \lambda_W \sum_{i=1}^{d_\theta} \|W_{i,:}\|^2) \quad (4)$$

with an  $L_2$  regularization on the context vectors and a mixed  $L_{2,1}$  regularization on the  $W$  matrix to induce sparsity,  $\lambda_\kappa$  and  $\lambda_W$  serve as hyperparameters.

Once the model is trained,  $\theta^s$  and  $W$  are fixed, so that only the context vectors  $\kappa_e$  of a set of new environments  $\mathcal{E}_{adapt} \subseteq \mathcal{E}$  need to be learned during the adaptation phase with Eq. 5:

$$\min_{\{\kappa_e\}_{e \in \mathcal{E}_{adapt}}} \sum_{e \in \mathcal{E}_{adapt}} \mathcal{L}(\theta^s + W\kappa_e, \mathcal{D}_e). \quad (5)$$

Where  $\mathcal{L}$  refers to the  $\Delta E94$  introduced in Eq. 1. The pseudo code Alg. 1 shows how the model is trained. Note that in practice we initialize the shared weights  $\theta^s$  by optimizing the loss  $\sum_{e \in \mathcal{E}_{tr}} \mathcal{L}(\theta^s, \mathcal{D}_e)$  leading to the initialization  $\theta^{init}$  which can be seen as a pre-trained network without contextual information.

## 4 Experiments

This section is devoted to our experimental study. We first introduce the used datasets, how they are pre-processed and prepared for the different learning ex-

**Algorithm 1** Model training and adaption pseudo-codes**Training:**

- 1: **Input:**  $\mathcal{E}_{\text{tr}} \subset \mathcal{E}$ ,  $\{\mathcal{D}_{e_{\text{tr}}}\}_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}}$
- 2:  $\vartheta = \{W, \theta^s, \{\kappa_{e_{\text{tr}}}\}_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}}\}$  where  $W \in \mathbb{R}^{d_\theta \times d_\kappa}$ ,  $\theta^s \in \mathbb{R}^{d_\theta}$
- 3: Randomly initialize  $W$ , and set  $\forall e_{\text{tr}} \in \mathcal{E}_{\text{tr}}, \kappa_{e_{\text{tr}}} = \mathbf{0} \in \mathbb{R}^{d_\kappa}, \theta^s = \theta^{\text{init}} \in \mathbb{R}^{d_\theta}$
- 4: **while** stopping criterion is not reached **do**
- 5:      $\vartheta \leftarrow \vartheta - \eta \nabla_{\vartheta} \left( \sum_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}} \left( \mathcal{L}(\theta^s + W\kappa_{e_{\text{tr}}}, \mathcal{D}_{e_{\text{tr}}}) + \lambda_\kappa \|\kappa_{e_{\text{tr}}}\|_2^2 + \lambda_W \sum_{i=1}^{d_\theta} \|W_{i,:}\|^2 \right) \right)$
- 6: **end while**

**Adaptation on a new environment:**

- 1: **Input:**  $e_{\text{adapt}} \in \mathcal{E}_{\text{adapt}} \subset \mathcal{E}$ ;  $\mathcal{D}_{e_{\text{adapt}}}$
- 2: Trained  $W \in \mathbb{R}^{d_\theta \times d_\kappa}$ ,  $\theta^s \in \mathbb{R}^{d_\theta}$  and  $\kappa_{e_{\text{adapt}}} = \mathbf{0} \in \mathbb{R}^{d_\kappa}$
- 3: **while** stopping criterion is not reached **do**
- 4:      $\kappa_{e_{\text{adapt}}} \leftarrow \kappa_{e_{\text{adapt}}} - \eta \nabla_{\kappa_{e_{\text{adapt}}}} \left( \mathcal{L}(\theta^s + W\kappa_{e_{\text{adapt}}}, \mathcal{D}_{e_{\text{adapt}}}) \right)$
- 5: **end while**

periments. Then, we present our learning setup before presenting the results, which are then analyzed both quantitatively and qualitatively. Finally, we propose an ablation study and a context size study to substantiate our parameters selection. Note that the code and the dataset are available as supplementary material and on Gitlab .

#### 4.1 Dataset

We use the dataset presented in [20] that contains a set of real colors measurements obtained from the principle introduced in Section 2.1. Among the existing observation modes, we consider **two modes** presented in Fig.1: the unpolarized transmission (Tunpol) and the Backside Reflection (BR). The data are obtained by varying 5 laser parameters: the laser power, the laser repetition rate (frep), the laser scan speed, the interline distance between laser lines and the laser polarization<sup>5</sup> state. Both datasets contain 9600 samples with the exact same laser parameters linked to their respective color L\*a\*b\* coefficients. Additional information are given in Appendix Sec.2.

We define different sub-environments of observation according to the laser polarization state parameter which corresponds to 4 possible laser polarization angles: 0, 30, 60 and 90 degrees. Then, the laser prediction consists in predicting the colors in the CIE LAB space according to the other 4 laser parameters across the different laser polarization environments. For each of the two datasets, we define 4 specific environment subsets:  $\mathcal{D}_{\{\text{Tunpol, BR}\}^{\{0, 30, 60, 90\}}}$ , each of them containing 2400 samples. As illustrated in Fig.4, the color shifts between different subsets is different enough to require an adaptation, we can note that the BR mode is more challenging than the Tunpol one.

To minimize the influence of the experimental noise during the learning process, a two-step pre-processing method is applied on the L\*a\*b\*/scan-speed

<sup>5</sup> Note that the laser polarization used to process the metasurface should not be confused with unpolarized transmission which refers to the non polarization of the illuminant used to acquire the dataset.

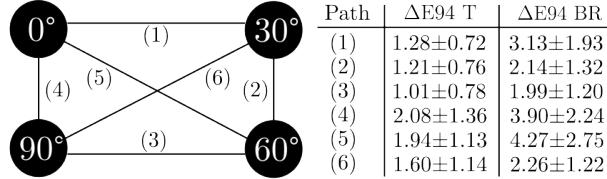


Fig. 4: Mean color shifts and their respective standard deviation between each sub-dataset in each observation mode with respect to the laser parameters. A path ( $i$ ) indicates the average  $\Delta E94$  between the colors obtained with respect to the two laser polarization angles linked by the path, for each of the two observation modes. For example (2) compares the colors between the  $30^\circ$  and  $60^\circ$  polarization angles for the same laser parameters.

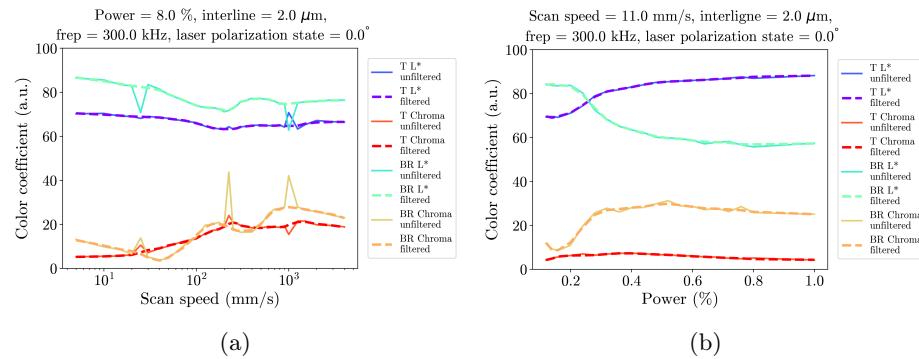


Fig. 5: Evolution of color coefficients over Scan speed (a) and Power (b). Chroma is defined as  $C = \sqrt{a^{*2} + b^{*2}}$ .

correlation curves. A Hampel filter (Alg.2) removes outliers and a polynomial filter (Alg. 3) enforces smoothness to color coefficients according to their laser processing parameters. Illustrated in Fig.5, outliers within correlation curves are removed while preserving the natural evolution of the colors. Laser processing parameters are minmax-normalized (with scan-speed log-normalized) and the color coefficients are normalized according to the defined bounds of each channel with  $L^* \in [0, 100]$  and  $a^*, b^* \in [-128, 128]$  allowing adaption on partially known environments. Model learning is done by a train/validation/test split following a 80/10/10 ratio for each specific environment dataset. Splitting is done at the laser parameters level ensuring that the same parameters appear in the same split for each environment. Each experiment is repeated 4 times.

#### 4.2 Experimental Setup

We present in this section the different learning setups considered in our study.

**Adaptation setup.** Given one observation mode, we consider the 4 settings where three of the four datasets are given as training environments and the last one is considered for the adaptation phase corresponding to the target environment.

**Base-architecture.** The architecture of our color prediction model corresponds to a 8 layer feed forward neural networks with 128 neurons on each layer. The size of the context vector is fixed to 3.

**Pre-training.** We first begin to pre-train the shared parameters  $\theta^s$  on a sample where we associate to each laser parameters the mean of the colors of the 3 datasets used in training<sup>6</sup>. Training is achieved over 800 epochs and a batch size of 64, using an Adam optimizer with a step scheduler after 300 epochs and a learning rate of 0.0015.

**Hypernetwork training and adaptation** (Hnet). For the training phase,  $\theta^s$  is initialized with pre-trained weights mentioned above. The hypernetwork parameters and context vectors are then adjusted with the mean response of the three datasets and then frozen<sup>7</sup>. The training uses the same learning parameters as pre-training with  $\lambda_k = 1e-3$  and  $\lambda_W = 1e-6$  for the regularization hyperparameters.

Adaptation is performed with a small data regime on the adaptation dataset with increasing data size drawn from its associated training sample: [5,10,20,40,80,160,320], these include laser parameters already seen in training phase by construction. Here, in addition to the test error that measures the generalization over unseen laser parameters during training, we also measure the error on half of the training sample of the adaptation environment that is not used during adaptation to measure the generalization to laser parameters unseen during adaptation but considered during training. Adaption is made on 100 epochs using Adam with no scheduling and a learning rate of 0.003.

**0-shot and Fine-Tuning** (0-shot, FT). 0-shot consists in evaluating each pre-trained model without updating any weights. We also perform a fine-tuning on all the models parameters for 500 epochs using Adam and step scheduling after 300 epochs. We use the same increasing data as for the adaptation phase of the hypernetwork.

**Hypernetwork adaptation and Fine-tuning** (HnetFT). Models obtained after the adaptation phase of the hypernetwork are also fine-tuned following the aforementioned protocol using exactly the same datafold as in adaption. Epochs are reduced to 400 with scheduling starting at 200 to align with fine-tuning tasks.

**Training from target with low data** (Target). A model is trained using only the data available in the adaptation phase with 40 and 320 samples, following the pre-training learning setup.

**Instance based WANN** (WANN). An adaptation is performed using the WANN model [23] and is tested using 40 and 320 samples from the target environment of the adaptation phase. The weighted network used is a 3 layer network with 64 neurons each using ReLU activation. Learning parameters are the same as in fine-tuning.

**Metrics.** To monitor the performance of the model in each adaptation scenario three metrics are considered: the accuracy defined as the percentage of color predictions that have a  $\Delta E94$  below to 1.5 with respect to the ground truth, the average  $\Delta E94$  mean values over the test sample and the associated  $\Delta E94$  standard deviation. Results are averaged on the 4 repetitions.

---

<sup>6</sup> On Tunpol dataset, using only  $\Delta E94$  could occassionaly make the model diverging. As a workaround model's parameters are updated with MSE during the first epoch and the rest of the training is done with  $\Delta E94$ .

<sup>7</sup> We have noticed nevertheless that it is more efficient to freeze the shared weights after the pre-training than learning them together with the other two sets of parameters.

### 4.3 Unpolarized Transmission

Results comparing FT models and Hnet models in Tunpol are displayed in Fig.6 and 7. In the very low data regimes, the Hnet based models are able to provide the best or a competitive performance. Fine-tuning the different models improves the results when the number of target data increases. The HnetFT model provides always for the best result for all data sizes. It must be noticed that using the Hnet model only reaches a plateau justifying the need for a subsequent model fine-tuning, but we can notice that the Hnet models provide a relevant initialization for fine-tuning. Generalization to unseen laser parameters during training is rather good for all models. Tab.1 provides numerical values on two problems for fixed target learning data sizes. The HnetFT models lead to the best accuracy with a variance reduction trend in terms of  $\Delta E_{94}$ . The evolution of the mean  $\Delta E_{94}$  in validation and test shown in Fig.8 highlights a reduction of overfitting for Hnet and HnetFT compared to FT making them more invariant to the used data-folds. Finally, the performance decrease between the best model with 320 data folds and the fully train one (Full results in Appendix Sec.3) is of 73% for  $0^\circ$  and 60% for  $60^\circ$ .

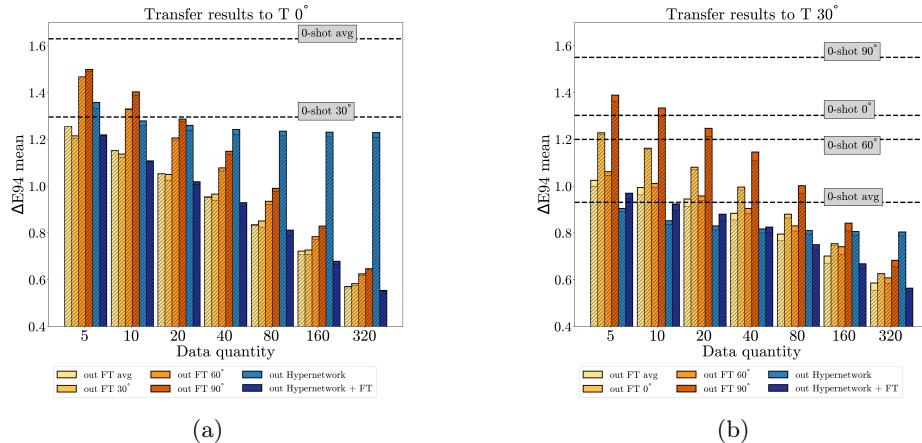


Fig. 6: Transfer results in Tunpol for  $0^\circ$  and  $30^\circ$  laser polarization states. Smooth bars correspond to results on test samples, hatched ones to unseen laser parameters during adaptation phase.

### 4.4 Backside Reflection

Results comparing FT models and Hnet models in BR are displayed in Fig.9 and 10. The same observations made for Tunpol can be made for BR, *i.e.* HnetFT always provides the best adaption model. Numerical results in Tab.2 confirm that HnetFT leads to best accuracy with better variance. Test and validation curves in Fig.11 show the same benefits as depicted in Tunpol. The performance decrease between the best model with 320 data folds and the fully train one is similar to Tunpol with 73% for  $0^\circ$

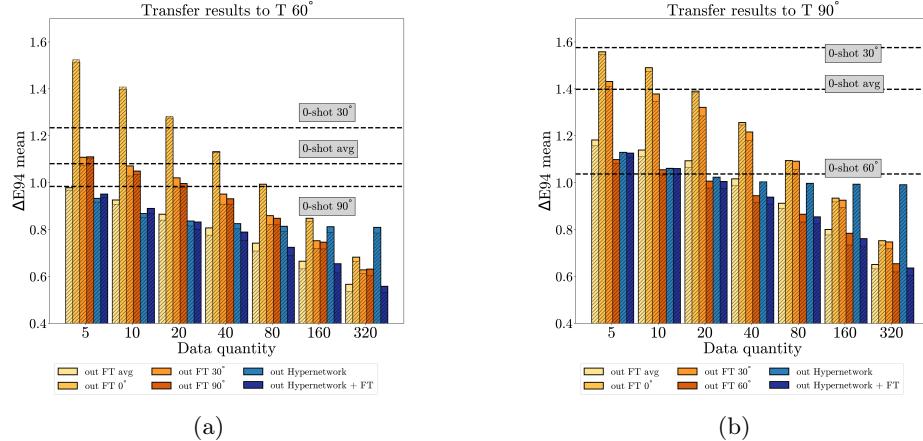


Fig. 7: Transfer results in Tunpol for  $60^\circ$  and  $90^\circ$  laser polarization states. Smooth bars with thick edge-lines correspond to results on test samples, hatched with thin lines ones to unseen laser parameters during adaptation phase.

Exp Name	To $0^\circ$			To $60^\circ$		
	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)
0-shot	$1.30 \pm 0.03$	$0.74 \pm 0.04$	$69.90 \pm 1.42$	$0.98 \pm 0.04$	$0.79 \pm 0.013$	$84.69 \pm 1.33$
Target-320	$0.86 \pm 0.07$	$0.83 \pm 0.19$	$56.98 \pm 3.23$	$0.92 \pm 0.07$	$0.85 \pm 0.12$	$71.09 \pm 3.78$
FT-320	$0.57 \pm 0.03$	$0.52 \pm 0.12$	$95.28 \pm 1.86$	$0.57 \pm 0.05$	$0.47 \pm 0.08$	$95.21 \pm 2.12$
Hnet-320	$1.23 \pm 0.04$	$0.75 \pm 0.08$	$73.61 \pm 2.27$	$0.80 \pm 0.02$	$0.57 \pm 0.03$	$89.34 \pm 0.55$
<b>HnetFT-320</b>	<b><math>0.55 \pm 0.05</math></b>	<b><math>0.45 \pm 0.03</math></b>	<b><math>96.04 \pm 0.90</math></b>	<b><math>0.56 \pm 0.03</math></b>	<b><math>0.45 \pm 0.07</math></b>	<b><math>95.76 \pm 1.33</math></b>
WANN-320	$1.26 \pm 0.03$	$0.71 \pm 0.05$	$71.66 \pm 2.52$	$1.05 \pm 0.03$	$0.85 \pm 0.10$	$82.93 \pm 1.33$
Target-40	$1.98 \pm 0.10$	$1.57 \pm 0.19$	$49.01 \pm 6.05$	$2.08 \pm 0.27$	$1.64 \pm 0.32$	$46.46 \pm 5.97$
FT-40	$0.95 \pm 0.06$	$0.78 \pm 0.12$	$85.00 \pm 2.69$	$0.81 \pm 0.05$	$0.47 \pm 0.08$	$89.69 \pm 2.42$
Hnet-40	$1.24 \pm 0.04$	$0.76 \pm 0.09$	$73.39 \pm 2.27$	$0.82 \pm 0.02$	$0.58 \pm 0.03$	$89.30 \pm 0.96$
<b>HnetFT-40</b>	<b><math>0.93 \pm 0.09</math></b>	<b><math>0.73 \pm 0.06</math></b>	<b><math>85.44 \pm 2.51</math></b>	<b><math>0.79 \pm 0.05</math></b>	<b><math>0.58 \pm 0.07</math></b>	<b><math>90.48 \pm 2.32</math></b>
WANN-40	$1.27 \pm 0.03$	$0.71 \pm 0.05$	$71.67 \pm 2.57$	$1.05 \pm 0.03$	$0.85 \pm 0.10$	$82.93 \pm 1.50$

Table 1: Test results of different models in Tunpol for adaptation to  $0^\circ$  and  $60^\circ$  degrees environments. Evaluation is done between Hnet and the finetuned pre-trained models. WANN uses the weights resulting to the best 0-shot adaptation.

and 54% for  $60^\circ$  (Full results in Appendix Sec.3). The results are nevertheless less good than the ones obtained in transmission confirming the difficulty of the task.

#### 4.5 Qualitative Analysis on the learned Environments

In this section, we provide a qualitative analysis of the context vectors learned with Hnet models by displaying their projection in 2D in Fig.12 and Fig.13. The heatmap indicates the adaptation error in terms of  $\Delta E94$  in the considered space. For each adaptation task, even in the smallest data regime, the learned context vector is close to the one corresponding to the best  $\Delta E94$  (green diamond). As the data quantity increases, context vectors get closer to the optimal one while becoming less and less scattered across the latent space. Two behaviors can be identified: one where the op-

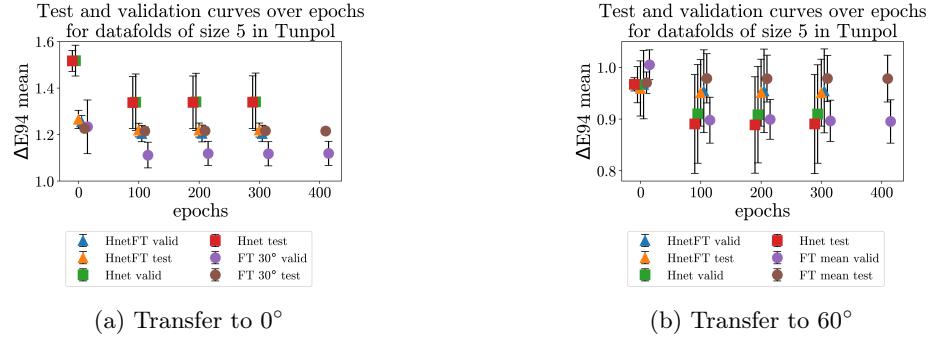


Fig. 8: Evolution of  $\Delta E94$  in test and validation in Tunpol every 100 epochs. Hnet epochs are extended up to 300. More results in Appendix Sec.3.

Exp Name	To 0°			To 60°		
	ΔE94 mean	ΔE94 std	Accuracy (%)	ΔE94 mean	ΔE94 std	Accuracy (%)
0-shot	3.00±0.19	1.81±0.11	22.29±2.42	2.00±0.09	1.18±0.04	38.12±5.72
Target-320	1.68±0.18	1.30±0.12	56.98±3.23	1.60±0.18	1.30±0.12	60.21±2.47
FT-320	1.22±0.04	0.87±0.07	72.19±1.66	1.15±0.04	0.84±0.08	75.87±2.93
Hnet-320	3.00±0.18	1.79±0.10	21.88±2.70	1.78±0.02	1.06±0.05	47.01±1.47
<b>HnetFT-320</b>	<b>1.21±0.06</b>	<b>0.86±0.06</b>	<b>74.44±3.53</b>	<b>1.13±0.04</b>	<b>0.84±0.10</b>	<b>76.32±2.78</b>
WANN-320	3.09±0.13	1.86±0.13	19.60±2.58	2.02±0.03	1.32±0.08	38.75±1.78
Target-40	6.23±1.03	4.13±0.92	8.19±2.45	5.88±0.65	3.73±0.44	5.65±2.71
FT-40	2.14±0.13	1.38±0.14	39.08±3.98	1.78±0.09	1.08±0.09	48.08±4.53
Hnet-40	3.05±0.17	1.79±0.12	20.79±3.11	1.82±0.06	1.07±0.05	45.35±3.02
<b>HnetFT-40</b>	<b>2.08±0.14</b>	<b>1.35±0.15</b>	<b>40.76±4.41</b>	<b>1.65±0.09</b>	<b>1.01±0.09</b>	<b>52.71±4.40</b>
WANN-40	3.10±0.13	1.87±0.13	18.75±2.95	2.03±0.03	1.32±0.09	37.92±2.54

Table 2: Test results for different model in BR for adaptation to  $0^\circ$  and  $60^\circ$  polarization states. Evaluation is done between Hnet and the finetuned pre-trained models. WANN uses the weights resulting to the best 0-shot adaptation.

tum is between two training environments as in Fig.12b or Fig.13b, and one where the optimum is close to one training environment such as in Fig.12a or in Fig.13a. These behaviors are directly related to the proximity of the training environments to the target ones. For example in Tunpol from Fig.6, when transferring to  $0^\circ$  the best model performs closely to the fine-tuned  $30^\circ$  model and this is what is found in Fig.12a, while the reverse is not true for adaptation to  $30^\circ$  (See Appendix Sec.4.). This shows that learned context vectors can be interpreted to give some insights on the closeness of different environments. Predicted colors are also closer to the expected color variations using Hnet as seen in Fig.14a and b for low data regime adaptation. With only five datapoints, Hnet has already reproduced some of the mapping compared to the Finetuned model. In BR, 0-shot delivers better performance which is consistent with previous numerical results. This highlights the capability of the model to quickly match new color trends.

## 4.6 Ablation Studies

The study is performed to highlight the impact of the introduced regularization and pre-training on the adaptation quality. The adaption task used is toward  $60^\circ$  in Tunpol

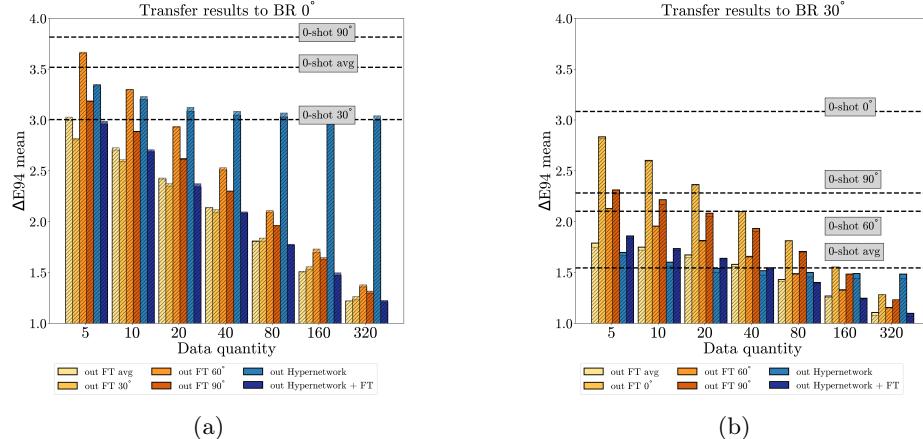


Fig. 9: Transfer results in BR for  $0^\circ$  and  $30^\circ$  laser polarization angles. Smooth bars with thick edge-lines correspond to results on test samples, hatched with thin lines ones to unseen laser parameters during adaptation phase.

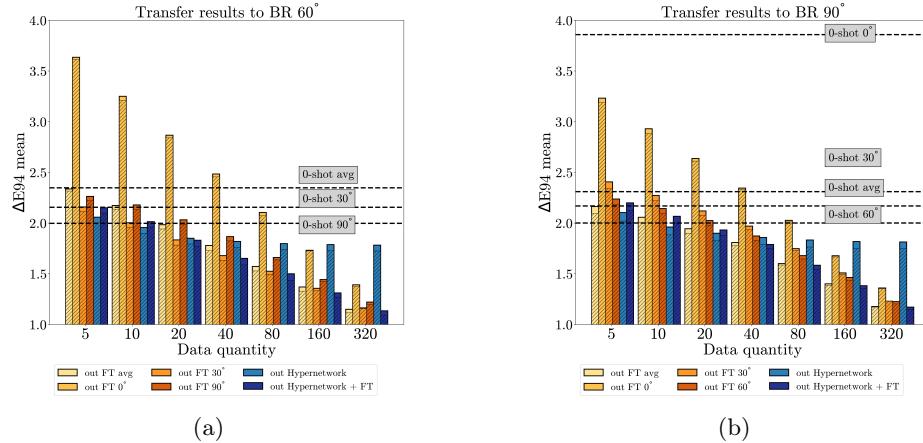


Fig. 10: Transfer results in BR for  $60^\circ$  and  $90^\circ$  laser polarization angles. Smooth bars with thick edge-lines correspond to results on test samples, hatched with thin lines ones to unseen laser parameters during adaptation phase.

and BR. Evaluation is done on Hnet with datafolds of size 320. Results are reported in Tab.3. If it is clear to see the benefits of pretraining and freezing on the adaptation results, it is hardly the case for the different regularizations. However, adding a regularization during the training helps to reduce the variance of each metrics, so that adaptation becomes more invariant to the datasets used to generate the latent space.

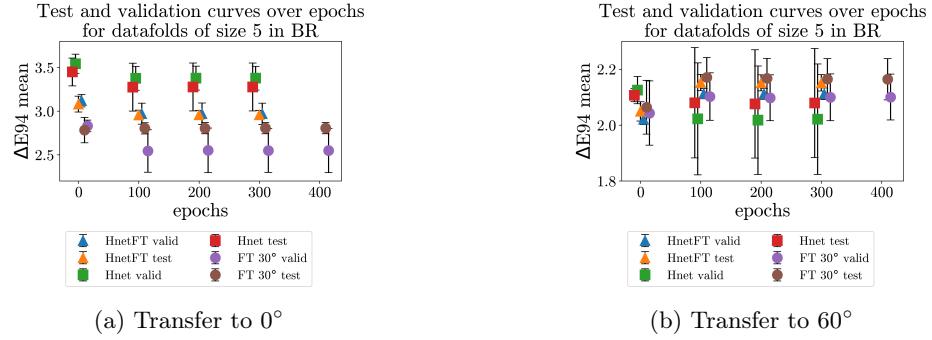


Fig. 11: Evolution of  $\Delta E94$  in test and validation in BR every 100 epochs. Hnet epochs are extended up to 300. More results in Appendix Sec.3.

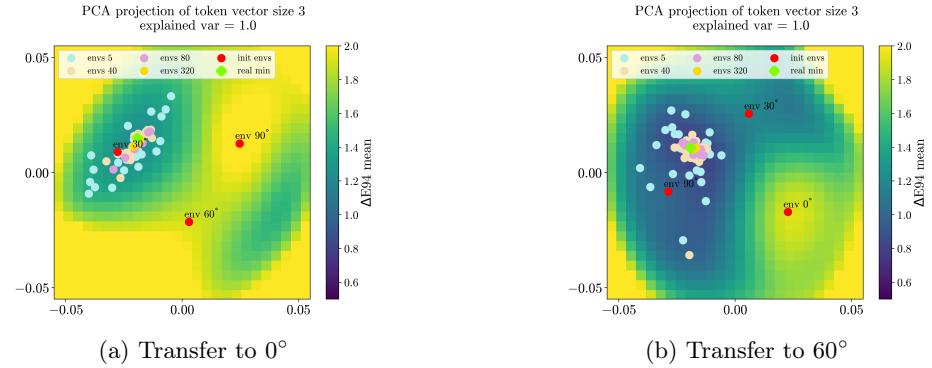


Fig. 12: Shared latent space visualization in Tumpol using PCA projection. More results in Appendix Sec.4.

It is chosen to apply a regularization on both decoder and context vector but a more in-depth hyperparameters tuning could enhance adaptation quality.

#### 4.7 Context Vectors Size Study

We use the same learning condition as for the ablation study on frozen pretrained weights. Context size varies from 1 to 4 and results are showed in Tab.4. As performance increases an optimum is found at size 3 with less variance on the metrics compared to size 2 and 4.

### 5 Conclusion

This work introduces adaptative color prediction models for nanophotonics applications based on a contextual hypernetwork. In a real-world experimental case scenario, the proposed model demonstrates its ability to adapt quickly and automatically to unseen environments in low data regimes while preventing strong overfitting. Our study

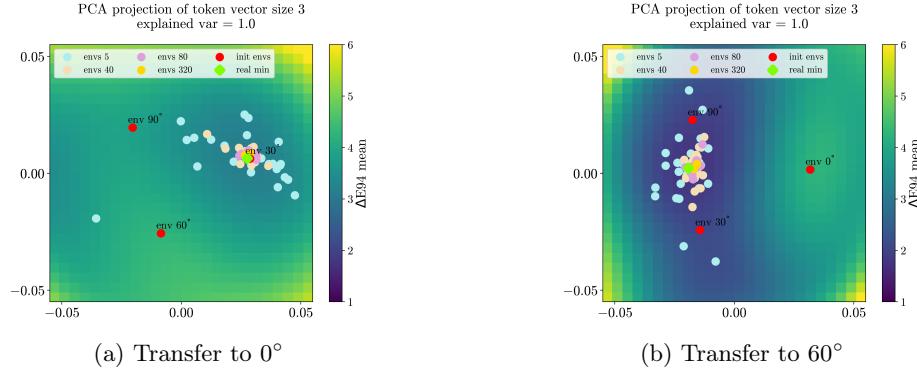


Fig. 13: Shared latent space visualization in BR using PCA projection. More results in Appendix Sec.4.

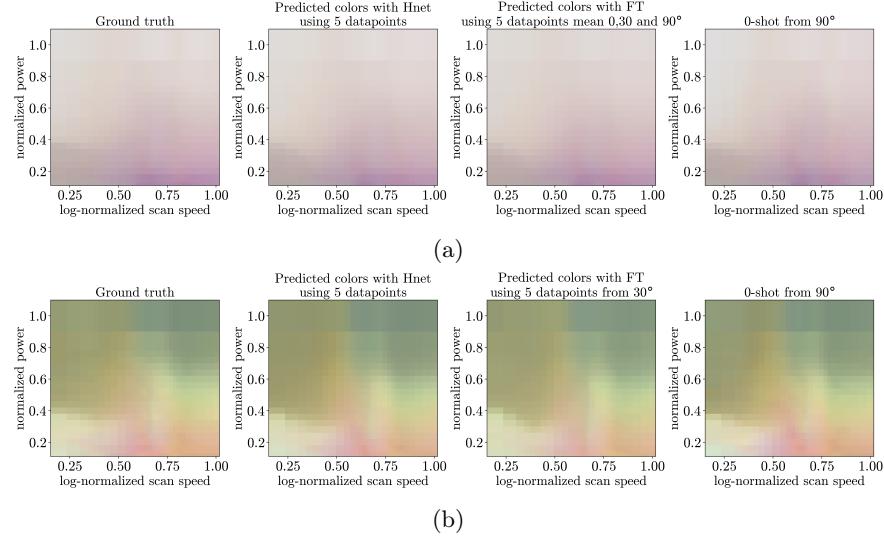


Fig. 14: Tunpol (a) and BR (b) colors comparison between real data and predicted ones given at 60° for repetition rate of 300kHz and interline distance of 2 $\mu$ m. More results in Appendix Sec.4.

indicates that the model needs to be refined by a fine-tuning procedure but its quality is good enough to ensure a good adaptation. The visualization of the context vectors can be interpreted to identify similar environments which could help the practitioner to identify close tasks. In low data-regime, the model matches more closely to ground truth's color variations than the other methods. Overall, this work provides a general framework for color adaptation that can offer many application perspectives for learning adaptive models. Future work aims at integrating a priori physical knowledge to help the model to adapt faster to a wider range of environments.

Exp Name	Tunpol			BR		
	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)
no-pretraining						
no reg	7.88 $\pm$ 0.23	5.18 $\pm$ 0.37	3.44 $\pm$ 2.32	8.04 $\pm$ 3.40	4.20 $\pm$ 1.31	4.20 $\pm$ 3.66
Only $\lambda_W$	10.24 $\pm$ 7.69	3.30 $\pm$ 2.18	10.31 $\pm$ 12.42	8.55 $\pm$ 1.52	3.98 $\pm$ 1.52	5.07 $\pm$ 7.11
Only $\lambda_\kappa$	11.58 $\pm$ 6.69	10.66 $\pm$ 1.66	1.94 $\pm$ 1.54	11.74 $\pm$ 8.02	3.19 $\pm$ 1.67	7.78 $\pm$ 12.33
$\lambda_W$ and $\lambda_\kappa$	9.53 $\pm$ 7.62	2.93 $\pm$ 2.16	19.62 $\pm$ 31.75	11.00 $\pm$ 1.03	4.98 $\pm$ 0.33	0.83 $\pm$ 0.59
pre-training						
no reg	1.16 $\pm$ 0.20	0.77 $\pm$ 0.17	75.73 $\pm$ 9.70	1.84 $\pm$ 0.06	1.02 $\pm$ 0.08	43.06 $\pm$ 4.18
Only $\lambda_W$	3.17 $\pm$ 2.91	2.26 $\pm$ 2.20	47.64 $\pm$ 30.29	1.84 $\pm$ 0.03	1.01 $\pm$ 0.06	43.06 $\pm$ 2.49
Only $\lambda_\kappa$	1.55 $\pm$ 0.66	1.03 $\pm$ 0.35	64.69 $\pm$ 20.59	1.91 $\pm$ 0.05	1.07 $\pm$ 0.10	41.08 $\pm$ 1.66
$\lambda_W$ and $\lambda_\kappa$	1.70 $\pm$ 0.24	1.00 $\pm$ 0.07	50.00 $\pm$ 11.86	1.82 $\pm$ 0.04	1.02 $\pm$ 0.06	44.51 $\pm$ 1.34
frozen-pretraining						
no reg	0.81 $\pm$ 0.01	0.57 $\pm$ 0.17	90.21 $\pm$ 0.52	1.74 $\pm$ 0.02	1.04 $\pm$ 0.05	49.41 $\pm$ 3.47
Only $\lambda_W$	0.80 $\pm$ 0.02	0.57 $\pm$ 0.03	90.07 $\pm$ 0.99	1.77 $\pm$ 0.04	1.04 $\pm$ 0.08	46.42 $\pm$ 2.49
Only $\lambda_\kappa$	0.79 $\pm$ 0.01	0.55 $\pm$ 0.02	91.18 $\pm$ 0.77	1.75 $\pm$ 0.01	1.03 $\pm$ 0.06	48.12 $\pm$ 2.33
$\lambda_W$ and $\lambda_\kappa$	0.80 $\pm$ 0.02	0.57 $\pm$ 0.03	89.34 $\pm$ 0.55	1.78 $\pm$ 0.02	1.06 $\pm$ 0.05	47.01 $\pm$ 1.47

Table 3: Test results for ablation study.

Exp Name	Tunpol			BR		
	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)	$\Delta E94$ mean	$\Delta E94$ std	Accuracy (%)
context size : 1	0.99 $\pm$ 0.04	0.61 $\pm$ 0.05	83.82 $\pm$ 2.80	1.78 $\pm$ 0.04	1.02 $\pm$ 0.04	45.17 $\pm$ 3.26
context size : 2	0.80 $\pm$ 0.02	0.56 $\pm$ 0.02	90.42 $\pm$ 0.62	1.77 $\pm$ 0.03	1.07 $\pm$ 0.06	47.71 $\pm$ 2.68
context size : 3	0.80 $\pm$ 0.02	0.57 $\pm$ 0.03	89.34 $\pm$ 0.55	1.78 $\pm$ 0.02	1.06 $\pm$ 0.05	47.01 $\pm$ 1.47
context size : 4	0.80 $\pm$ 0.02	0.57 $\pm$ 0.03	90.17 $\pm$ 1.48	1.75 $\pm$ 0.03	1.05 $\pm$ 0.07	47.64 $\pm$ 3.06

Table 4: Test results for context Vector Size study.

**Algorithm 2** Hampel filter algorithm $X, k$  refer respectively to a 1D or 2D signal and the sliding window size.

- ```

1: Input:  $X = \{x_i\}_{i \in \{0, \dots, N\}, N \in \mathbb{N}}, k \in \mathbb{N}^*$ 
2: for  $x_i \in X$  do ▷ Boundary conditions are taken into account for  $i > N + k$  and  $i < k + 1$ 
3:    $m \leftarrow \text{median}(x_{i-k}, \dots, x_i, \dots, x_{i+k})$ 
4:    $\sigma \leftarrow \kappa \text{median}(|x_{i-k} - m|, \dots, |x_i - m|, \dots, |x_{i+k} - m|)$  ▷ here  $\frac{\sigma}{\kappa}$  refers to the median absolute deviation and  $\kappa \approx 1.4826$ 
5:   if  $|x_i - m| > \sigma$  then  $x_i \leftarrow m$  endif
6: end for

```

**Algorithm 3** Savitsky-Golay filter algorithm  $X, k, P$  refer respectively to a 1D or 2D signal, the sliding window size and the maximum polynomial order.

- ```

1: Input:  $X = \{x_i\}_{i \in \{0, \dots, N\}, N \in \mathbb{N}}, (k, P) \in \mathbb{N}^{*2}$ 
2:  $\hat{y} \leftarrow []$ 
3: for  $x_i \in X$  do
4:    $y \leftarrow \sum_{p=0}^P \alpha_p x^p$ 
5:   Minimize  $\sum_{i=-k}^k (y_{k+i} - \sum_{p=1}^P \alpha_p x_{k+i}^p)^2$ 
6:    $\hat{y}_k \leftarrow \sum_{p=0}^P \alpha_p x_k^p$ 
7: end for

```

*Impact Statement* Our work focuses on a machine learning model for the prediction of laser-induced colors in nanophotonics. This work is essentially an applicative work for an engineering task. There are many potential consequences of using our work on other domains or tasks, none which we feel must be specifically highlighted here.[28]

**Acknowledgments.** This work has been funded by a public grant from the French National Research Agency under the “France 2030” investment plan, which has the reference EUR MANUTECH SLEIGHT - ANR-17-EURE-0026.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. An, S., Zheng, B., Julian, M., Williams, C., Tang, H., Gu, T., Zhang, H., Kim, H.J., Hu, J.: Deep neural network enabled active metasurface embedded design. *Nanophotonics* **11**(17), 4149–4158
2. Baxter, J., Calà Lesina, A., Guay, J.M., Weck, A., Berini, P., Ramunno, L.: Plasmonic colours predicted by deep learning. *Sci Rep* **9**(1), 8074
3. Bibbò, L., Khan, K., Liu, Q., Lin, M., Wang, Q., Ouyang, Z.: Tunable narrowband antireflection optical filter with a metasurface. *Photon. Res.* **5**(5), 500–506 (2017)
4. Casaletti, M., Valerio, G., Quevedo-Teruel, O., Burghignoli, P.: An overview of metasurfaces for thin antenna applications. *Comptes Rendus. Physique* **21**(7-8), 659–676 (2020)
5. Dalloz, N., Le, V.D., Hebert, M., Eles, B., Flores Figueroa, M.A., Hubert, C., Ma, H., Sharma, N., Vocanson, F., Ayala, S., Destouches, N.: Anti-Counterfeiting White Light Printed Image Multiplexing by Fast Nanosecond Laser Processing. *Advanced Materials* **34**(2), 2104054
6. Destouches, N., Crespo-Monteiro, N., Vitrant, G., Lefkir, Y., Reynaud, S., Epicier, T., Liu, Y., Vocanson, F., Pigeon, F.: Self-organized growth of metallic nanoparticles in a thin film under homogeneous and continuous-wave light excitation. *J. Mater. Chem. C* **2**, 6256–6263 (2014)
7. Destouches, N., Sharma, N., Vangheluwe, M., Dalloz, N., Vocanson, F., Bugnet, M., Hébert, M., Siegel, J.: Laser-Empowered Random Metasurfaces for White Light Printed Image Multiplexing. *Advanced Functional Materials* **31**(18), 2010430
8. Dinsdale, N.J., Wiecha, P.R., Delaney, M., Reynolds, J., Ebert, M., Zeimpekis, I., Thomson, D.J., Reed, G.T., Lalanne, P., Vynck, K., Muskens, O.L.: Deep Learning Enabled Design of Complex Transmission Matrices for Universal Optical Components. *ACS Photonics* **8**(1), 283–295
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Precup, O., Teh, Y.W. (eds.) *International Conference on Machine Learning*. vol. 70, pp. 1126–1135. PMLR (06–11 Aug 2017)
10. Jia, Y., Qian, C., Fan, Z., Cai, T., Li, E.P., Chen, H.: A knowledge-inherited learning for intelligent metasurface design and assembly. *Light Sci Appl* **12**(1), 82
11. Jiang, J., Chen, M., Fan, J.A.: Deep neural networks for the evaluation and design of photonic devices. *Nat Rev Mater* **6**(8), 679–700
12. Khoram, E., Wu, Z., Qu, Y., Zhou, M., Yu, Z.: Graph Neural Networks for Metasurface Modeling. *ACS Photonics* **10**(4), 892–899
13. Kirchmeyer, M., Yin, Y., Dona, J., Baskiotis, N., Rakotomamonjy, A., Gallinari, P.: Generalizing to New Physical Systems via Context-Informed Dynamics Model. In: *International Conference on Machine Learning*. pp. 11283–11301. PMLR
14. Lalanne, P., Chen, M., Rockstuhl, C., Sprafke, A., Dmitriev, A., Vynck, K.: Disordered optical metasurfaces: basics, properties, and applications. *Adv. Opt. Photon.* **17**(1), 45–113 (Mar 2025)
15. Le, V.D., Eles, B., Dalloz, N., Flores Figueroa, M.A., Vocanson, F., Hubert, C., Destouches, N.: Understanding and exploiting the optical properties of laser-induced quasi-random plasmonic metasurfaces. *ACS Applied Optical Materials* **2**(3), 373–385 (2024)

16. Liu, Z., Siegel, J., Garcia-Lechuga, M., Epicier, T., Lefkir, Y., Reynaud, S., Bugnet, M., Vocanson, F., Solis, J., Vitrant, G., Destouches, N.: Three-dimensional self-organization in nanocomposite layered systems by ultrafast laser pulses. *ACS Nano* **11**(5), 5031–5040 (2017), pMID: 28471649
17. Liu, Z., Vitrant, G., Lefkir, Y., Bakhti, S., Destouches, N.: Laser induced mechanisms controlling the size distribution of metallic nanoparticles. *Phys. Chem. Chem. Phys.* **18**, 24600–24609 (2016)
18. Liu, Z., Zhu, D., Rodrigues, S.P., Lee, K.T., Cai, W.: Generative Model for the Inverse Design of Metasurfaces. *Nano Lett.* **18**(10), 6570–6576
19. Ma, H., Bakhti, S., Rudenko, A., Vocanson, F., Slaughter, D.S., Destouches, N., Itina, T.E.: Laser-generated ag nanoparticles in mesoporous tio2 films: Formation processes and modeling-based size prediction. *The Journal of Physical Chemistry C* **123**(42), 25898–25907 (2019)
20. Ma, H., Dalloz, N., Habrard, A., Sebban, M., Sterl, F., Giessen, H., Hebert, M., Destouches, N.: Predicting Laser-Induced Colors of Random Plasmonic Metasurfaces and Optimizing Image Multiplexing Using Deep Learning. *ACS Nano* **16**(6), 9410–9419 (2022)
21. Ma, W., Liu, Z., Kudyshev, Z.A., Boltasseva, A., Cai, W., Liu, Y.: Deep learning for the design of photonic structures. *Nat. Photonics* **15**(2), 77–90
22. Majorel, C., Girard, C., Arbouet, A., Muskens, O.L., Wiecha, P.R.: Deep Learning Enabled Strategies for Modeling of Complex Aperiodic Plasmonic Metasurfaces of Arbitrary Size. *ACS Photonics* **9**(2), 575–585
23. de Mathelin, A., Richard, G., Mougeot, M., Vayatis, N.: Adversarial weighting for domain adaptation in regression. *CoRR* **abs/2006.08251** (2020)
24. Peng, R., Ren, S., Malof, J., Padilla, W.J.: Transfer learning for metamaterial design and simulation. *Nanophotonics* **13**(13), 2323–2334
25. Qin, J., Jiang, S., Wang, Z., Cheng, X., Li, B., Shi, Y., Tsai, D.P., Liu, A.Q., Huang, W., Zhu, W.: Metasurface micro/nano-optical sensors: Principles and applications. *ACS Nano* **16**(8), 11598–11618 (2022), pMID: 35960685
26. Sadeghi Dizaji, P., Habibiyan, H.: Machine learning with knowledge constraints for design optimization of microring resonators as a quantum light source. *Sci Rep* **15**(1), 372
27. Sharma, N., Destouches, N., Florian, C., Serna, R., Siegel, J.: Tailoring metal-dielectric nanocomposite materials with ultrashort laser pulses for dichroic color control. *Nanoscale* **11**(40), 18779–18789
28. Sicilia, A., Zhao, X., Hwang, S.J.: Domain adversarial neural networks for domain generalization: when it works and how to improve. *Mach. Learn.* **112**(7) (2023)
29. Wang, H., Zhao, H., Li, B.: Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In: ICML (2021)
30. Xu, P., Lou, J., Li, C., Jing, X.: Inverse design of a metasurface based on a deep tandem neural network. *J. Opt. Soc. Am. B, JOSAB* **41**(2), A1–A5
31. Zhang, J., Huang, J., Luo, Z., Zhang, G., Lu, S.: DA-DETR: domain adaptive detection transformer by hybrid attention. In: CVPR (2023)
32. Zhu, R., Qiu, T., Wang, J., Sui, S., Hao, C., Liu, T., Li, Y., Feng, M., Zhang, A., Qiu, C.W., Qu, S.: Phase-to-pattern inverse design paradigm for fast realization of functional metasurfaces via transfer learning. *Nature Communications* **12**(1), 2974 (May 2021)