

# Physics-Informed PointNets for Modeling Electromagnetic Scattering from All-Dielectric Metasurfaces with Inclined Nanopillars

Leon Armbruster , Vlad Medvedev and Andreas Rosskopf

**Abstract**—Metasurfaces are innovative planar optical structures capable of manipulating incident light properties. Accurate and computationally efficient modeling of such metasurfaces, particularly those with irregular geometries, remains a challenge for conventional solvers. In this work, we present a mesh-free Physics-Informed PointNet (PIPN) to model electromagnetic scattering from all-dielectric metasurfaces that feature spatially varying nanopillars. Our approach uses the PointNet architecture to directly encode spatially varying material properties into the Physics-Informed Machine Learning (PIML) framework. We demonstrate the generalization capability of our PIPN through evaluations on datasets; these datasets are generated with varying refractive indices representing common dielectric materials. Furthermore, the inclination angles are varied within each dataset, which represent expected manufacturing defects. Overall, our method provides a promising, mesh-free framework for accurate and efficient modeling of complex optical structures represented by irregular geometries.

**Index Terms**—physics-informed machine learning, helmholtz equation, mesh free, weakly supervised, hybrid method

## I. INTRODUCTION

Rapid advancements in nanofabrication techniques have led to the development of sophisticated nano-optic devices, with features on the scale of (or even smaller than) the dimensions of the operating wavelength. In particular, metasurfaces have emerged as planar optical structures capable of manipulating incident light in diverse ways.

Metasurfaces are artificially engineered planar materials composed of meta-atoms with subwavelength thickness. Because they allow precise control over the properties of the electromagnetic waves they interact with, metasurfaces are an important advancement in the field of nano-optics [1]–[4]. By scattering light from resonant nanostructures, properties such as phase, amplitude, and polarization-can be finely tuned, as opposed to conventional optical elements that reflect, scatter, or refract light in a less controlled manner. This unlocks novel applications in fields such as: biomedical imaging and sensing [5], full color holography [6], multiplexed communications [7], and all-optical diffractive deep neural networks [8].

(Corresponding author: Leon Armbruster) The authors are with the Fraunhofer Institute for Integrated Systems and Device Technology, 91058 Erlangen, Germany (e-mail: leon.armbruster@iisb.fraunhofer.de; vlad.medvedev@iisb.fraunhofer.de; andreas.rosskopf@iisb.fraunhofer.de).

Manuscript received Month Day, Year; revised Month Day, Year.

Due to these new applications, demand has increased for accurate and efficient modeling techniques to predict the electromagnetic behavior of metasurfaces and optimize their performance. To model light diffraction from these sub-wavelength nanostructures and capture complex optical effects, researchers have traditionally relied on electromagnetic field solvers. These traditional solvers numerically approximate Maxwell's equations by utilizing algorithms such as the finite-difference/element methods [9] and rigorous coupled wave analysis [10]. However, difficulties can arise when the dimensionality of the problem increases or the governing equations involve highly nonlinear or multi-scale behavior. These limitations are exacerbated when simulating curvilinear or irregular nano-optic geometries, where a high degree of spatial discretization is required to resolve the interactions between light and the nanoscale features [11].

These flaws have led to the development of alternative approaches capable of maintaining the high accuracy numerical solvers provide while reducing computational time and complexity. Recent advancements in machine learning—particularly deep learning—have introduced a new paradigm to address optical modeling challenges. Early applications of data driven deep learning in computational optics was proven to enable faster predictions and enhanced generalization. These studies demonstrate Neural Networks' (NNs) ability to predict electromagnetic responses [12], [13] (such as far-field spectra) with high accuracy. However, these data-driven approaches often require extensive training datasets generated through costly simulations or obtained by experimental measurements. Moreover, they operate as 'black-box' models, offering limited insight into the underlying physics and often struggling with interpolation or extrapolation tasks. To overcome these limitations, hybrid approaches [14] have gained traction, combining data-driven methods with physics-based constraints. PIML has emerged as a prominent extension of traditional data-driven machine learning approaches [15], [16]. So called Physics-Informed Neural Networks (PINNs) incorporate the governing equations of physical systems directly into the NN training process, constraining the network to produce physically plausible solutions, thereby reducing overfitting and improving interpretability.

Nonetheless, PINNs can incorporate observational data directly into the loss function, allowing for additional constraints in settings where data-driven regularization is necessary [17].

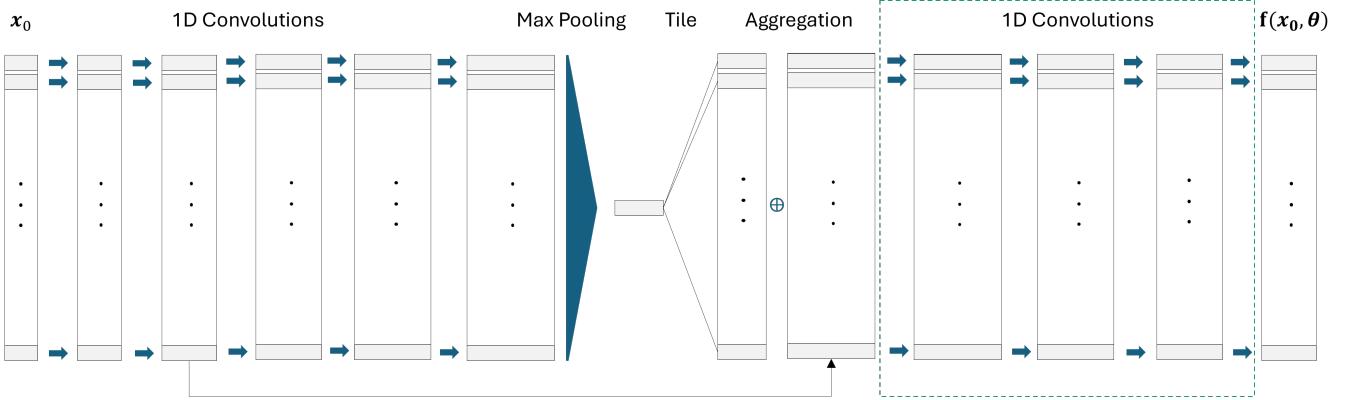


Fig. 1: A schematic of the PointNet architecture used in this work. The PointNet's design leverages 1D convolutional kernels applied to each point individually to process local features, as well as a symmetric aggregation function (such as max pooling or average pooling) to generate a global feature vector representation that is invariant to the input order. These global and local feature vectors are combined to produce the output point cloud. The dotted square represents the second half of convolutional layer which can be replaced by other layer types (App. VI-B).

This can be especially useful when sparse observational or experimental data is available, or when the underlying physics constraints are not exactly known. Despite these advantages, PINNs introduce their own set of challenges, such as slow convergence, difficulty in balancing different loss terms, and sensitivity to hyperparameter tuning [18], [19]. Ultimately, while traditional numerical methods remain state-of-the-art for many well-posed Partial Differential Equation (PDE) problems, PINNs and related PIML techniques offer promising alternatives in complex real world scenarios [14], [20], [21].

Early applications of PINNs in nano-optics have shown promise in solving inverse scattering problems involving interacting nanostructures [22]–[24], designing photonic devices [25], and lithography mask simulations [26]–[28]. However, many existing studies have relied on scalar approximations and grid-based methods that limit their ability to efficiently capture electromagnetic interactions on complex domains.

To overcome these limitations, we employed a PIPN framework. Unlike grid-based methods, PIPN frameworks use a point cloud representation of the domain, which enables efficient encoding of varying domain geometries. To the best of our knowledge, this is the first work that encodes spatially varying PDE parameters into the PIPN training framework. Based on that, we can train NNs that learn the connection between the material properties of the metasurface and the corresponding electromagnetic response. This allows for subsequent fast inference on previously unseen domains. By extending the PIPN framework, we aim to develop a robust and scalable mesh-free method for predicting both near-field and far-field electromagnetic responses of nano-optical devices.

In Sec. II-A we introduce the PointNet architecture and highlight its advantage over mesh-based approaches. Then, in Sec. II-B, we describe the physical model we used as well as our reason for including reference solutions in our training paradigm. We can then establish how the training and testing domains are constructed, and how the reference solutions are calculated. In Sec. II-C, we construct the loss functions used during training to highlight the addition of a reference loss

term. In Sec. III, we lay out the training hyperparameters and discuss the metrics used for testing. In Sec. IV, the main results are shown; the implications of the results are discussed in Sec. V and concluded in Sec. VI.

## II. PROBLEM FORMULATION

### A. PointNets

The PointNet (Fig. 1) is a NN architecture originally developed for processing unordered point clouds for tasks in computer vision, including object classification [29], segmentation [30]. This architecture was extended for use in PIML by [31]–[33], which introduced and popularized PIPNs. The schematic in Fig. 1 illustrates how the PointNet processes its input.

Our input point cloud is structured as vector  $\mathbf{x}_0 = (B \times N_t \times (d+1))$ , where  $B$  denotes the batch size and  $N_t$  denotes the total number of points in the point cloud. The  $(d+1)$  component combines the spatial coordinates (determined by dimension  $d$ ) with the spatially dependent permittivity. For the two-dimensional problem, the network produces an output point cloud  $\mathbf{f}(\mathbf{x}_0; \theta) = (B \times N_t \times 2)$ , where the final channel dimension represents the real and imaginary parts of the electric field. In the three-dimensional case, the output expands to  $\mathbf{f}(\mathbf{x}_0; \theta) = (B \times N_t \times 6)$ , capturing the real and imaginary components of all three electric field vectors ( $E_x$ ,  $E_y$ , and  $E_z$ ).

Furthermore, as long as the underlying architecture is not changed, varying the number of convolutional layers and/or the size of the 1D convolutional kernels does not notably impact the final error. A modification to this network structure investigated in App. VI-B is to replace the 1D convolutional layers after the aggregation (as indicated by the dashed square in Fig. 1) either with dense multilayer perceptron layers or a Kolmogorov-Arnold Network (KAN). This is viable because the material distribution and geometry have already been encoded after the feature aggregation.

PIPNs offer several distinct advantages compared to mesh-based physics-informed NNs, which typically process binary images where pixels encode different regions (e.g., solid objects and fluid flow [34]).

**Advantages over mesh-based approaches:** First, for such Convolutional NNs to capture the high frequency aspects of the input and output, the overall node density of the grid must be increased throughout the entire computational domain. This leads to an increase in computational time. Working with point clouds theoretically solves this problem, as sampling more points in regions where finer detail is needed allows for adaptive resolution since one can simply sample more points in regions where finer detail is needed to allow for adaptive resolution.

Second, defining boundary conditions can be challenging when working with irregular geometries in mesh-based approaches, as a single pixel may contain both boundary and interior regions. This ambiguity complicates loss function implementation. It requires decision-making about whether a given pixel contributes to boundary constraints or PDE residuals. This issue is also present in the case of our inclined pillars, where separating between the pillars and surrounding air can be challenging due to numerical artifacts present when deciding which pixel belongs to which medium. PointNets (Fig. 1), on the other hand, eliminate this issue, as they naturally distinguish between boundary and interior points.

Another limitation of mesh-based approaches is their poor performance when dealing with sparse data. Since convolutions are applied over the entire grid regardless of where meaningful information is located, this can lead to significant inefficiencies. In scenarios where most of the input consists of empty, irrelevant, or noisy regions, a large portion of the computational effort is wasted. This makes mesh-based NNs less suitable for applications involving highly sparse or irregularly distributed data.

### B. Governing Equation

The Helmholtz equation is often used in scattering problems because it simplifies the analysis of wave propagation in a medium under steady-state conditions. While Maxwell's equations describe the complete time-dependent behavior of electromagnetic fields, the Helmholtz equation arises under the assumption of time-harmonic fields ( $e^{-i\omega t}$ ), transforming the problem into a spatial boundary value problem. By focusing on the spatial variation of the fields, the Helmholtz equation allows the modeling of wave interactions with scatterers (e.g. obstacles, particles, or surfaces) without explicitly tracking time-dependent field evolution. This assumption significantly simplifies both computational and analytical approaches. Specifically, for linear, isotropic, non-magnetic, and non-dispersive materials in regions without free current density, Maxwell's equations reduce to the Helmholtz equation. Assuming that the electric field varies sinusoidally in time, given by  $\mathbf{E}(\mathbf{x}, t) = \mathbf{E}(\mathbf{x})e^{-i\omega t}$  where  $\omega = \frac{2\pi}{\lambda}$  is the angular frequency and  $\lambda$  is the wavelength of the incident light, a frequency-domain formulation can be derived. Substituting this time-harmonic ansatz into Maxwell's curl equations for non-magnetic materials leads to the vectorial Helmholtz equation:

$$\nabla \times (\nabla \times \mathbf{E}(\mathbf{x})) + \omega^2 \varepsilon(\mathbf{x}) \mathbf{E}(\mathbf{x}) = 0, \quad (1)$$

where  $\mathbf{x} = (x, z)$  for the two-dimensional case and  $\mathbf{x} = (x, y, z)$  for the three-dimensional case. We furthermore assume lossless media with no absorption or amplification effects, ensuring energy conservation by setting  $\text{Im}(\varepsilon(\mathbf{x})) = 0$ , where  $\varepsilon(\mathbf{x})$  is the spatially dependent permittivity. In addition to (1), we impose periodic boundary conditions in  $x$ - and  $y$ -direction. In the  $z$ -direction, non-reflective boundary conditions are assumed. Adding the boundary conditions to (1) gives us the problem formulation of the Helmholtz equation for the three-dimensional case.

For the two-dimensional case, we can effectively decompose the two-dimensional incident field into two primary polarization states: Transverse Electric (TE) and Transverse Magnetic modes. In our simulation, we have a TE-polarized monochromatic plane wave traveling in the  $z$ -direction. In the TE polarization, the electric field is entirely perpendicular to the plane of incidence ( $xz$ -plane). This implies that the electric field  $\mathbf{E}$  has only a single non-zero component,  $\mathbf{E}_y$ , which oscillates along the  $y$ -axis. As a result, we obtain the following scalar formulation of the Helmholtz equation:

$$\begin{aligned} \nabla^2 \mathbf{E}_y(\mathbf{x}) + \omega^2 \varepsilon(\mathbf{x}) \mathbf{E}_y(\mathbf{x}) &= 0 & \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ \mathbf{E}_y(\mathbf{x}_{\Gamma_a}) - \mathbf{E}_y(\mathbf{x}_{\Gamma_b}) &= 0 & \mathbf{x}_{\Gamma_{a/b}} \in \partial\Omega \subset \mathbb{R}^{d-1}, \\ \nabla \mathbf{E}_y(\mathbf{x}_{\Gamma_a}) - \nabla \mathbf{E}_y(\mathbf{x}_{\Gamma_b}) &= 0, \end{aligned} \quad (2)$$

where  $\mathbf{x}_{\Gamma_{a/b}}$  denote corresponding points on the boundaries. This is the scalar Helmholtz equation for the TE polarization case. Well-posedness for problem (2) in the sense of Hadamard is shown in [35]. For readability's sake, we set  $\mathbf{E}(\mathbf{x}) := \mathbf{E}_y(\mathbf{x})$  from here on out.

Furthermore, since no source is defined, problem (2) admits a trivial solution where  $\mathbf{E}(\mathbf{x}) = 0$  everywhere in  $\Omega$ . While mathematically valid, this solution is physically meaningless unless justified by specific boundary conditions or constraints. One way to circumvent this limitation would be to decompose the electric field into two parts: one representing the scattered electric field and the other representing the incident electric field [36]. We on the other hand add points into the training point clouds for which the reference solution is known, thereby explicitly encoding the source term into the problem formulation.

### C. Physics-Informed Machine Learning

Unlike traditional supervised learning that relies solely on input-output pairs, PIML requires a specialized loss function that enforces compliance with the governing PDEs, boundary conditions, and any available reference data throughout the computational domain. This physics-constrained optimization ensures that the learned solution is physically meaningful and generalizable beyond the training data.

The total PIML loss function is then composed as a weighted sum of these residuals over their respective domains:

$$\mathcal{L}(\theta) = \lambda_i \mathcal{L}_i + \lambda_b \mathcal{L}_b + \lambda_r \mathcal{L}_r. \quad (3)$$

Each category corresponds to a specific loss— $\mathcal{L}_i$  for interior points,  $\mathcal{L}_b$  for boundary points, and  $\mathcal{L}_r$  for reference points.

The weighting factors  $\lambda_i, \lambda_b, \lambda_r$  balance the contributions of the PDE, boundary, and reference losses respectively. The optimization problem is then given by:

$$\begin{aligned} \theta^* = \arg \min_{\theta} \mathcal{L}(\theta) &= \frac{1}{|\Omega|} \int_{\Omega} |\mathcal{R}_i(\mathbf{x}^i, \varepsilon(\mathbf{x}^i); \theta)|^2 d\mathbf{x}^i \\ &+ \frac{1}{|\partial\Omega|} \int_{\partial\Omega} |\mathcal{R}_b(\mathbf{x}^b; \theta)|^2 d\mathbf{x}^b \\ &+ \frac{1}{N_r} \sum_{j=1}^{N_r} |\mathcal{R}_r(\mathbf{x}_j^r, \varepsilon(\mathbf{x}_j^r); \hat{\mathbf{E}}(\mathbf{x}_j^r), \theta)|^2. \end{aligned} \quad (4)$$

Since exact integration is often computationally infeasible, we approximate the integral terms using numerical quadrature or Monte Carlo sampling over a discrete set of collocation points  $\{\mathbf{x}_j^i\}_{j=1}^{N_i} \subset \Omega$  for the PDE residual, and  $\{\mathbf{x}_j^b\}_{j=1}^{N_b} \subset \partial\Omega$  for the boundary residual. Similarly, we have a set of points  $\{\mathbf{x}_j^r\}_{j=1}^{N_r} \subset \Omega$  at which the reference solutions are enforced. Thus, the total number of points in each point cloud is  $N_t = N_i + N_b + N_r$ . A representative two-dimensional training point cloud with  $N_r = 224$ ,  $N_b = 128$ , and  $N_i = 1696$ , for a total of  $N_t = 2048$  can be seen in Fig. 2(a).

The approximation of  $\mathbf{E}$  by the NN depends on the spatial coordinates  $\mathbf{x}$ , the permittivity distribution  $\varepsilon(\mathbf{x})$ , and the set of trainable network parameters  $\theta$ . We denote this approximation as  $\hat{\mathbf{E}}(\mathbf{x}; \theta)$ , thereby expanding our previous definition of  $\mathbf{x}$  to also include the permittivity at the given coordinate. The corresponding discrete optimization problem is then given by:

$$\begin{aligned} \theta^* = \arg \min_{\theta} \mathcal{L}(\theta) &\approx \frac{\lambda_i}{N_t} \sum_{j=1}^{N_i} |\mathcal{R}_i(\mathbf{x}_j^i, \varepsilon(\mathbf{x}_j^i); \theta)|^2 \\ &+ \frac{\lambda_b}{N_b} \sum_{j=1}^{N_b} |\mathcal{R}_b(\mathbf{x}_j^b; \theta)|^2 \\ &+ \frac{\lambda_r}{N_r} \sum_{j=1}^{N_r} |\mathcal{R}_r(\mathbf{x}_j^r; \hat{\mathbf{E}}(\mathbf{x}_j^r), \theta)|^2 \end{aligned} \quad (5)$$

where, for the two-dimensional case, we have

$$\begin{aligned} \mathcal{R}_i(\mathbf{x}_j^i, \varepsilon(\mathbf{x}_j^i); \theta) &= \nabla^2 \mathbf{E}(\mathbf{x}_j^i; \theta) + \omega^2 \varepsilon(\mathbf{x}_j^i) \mathbf{E}(\mathbf{x}_j^i, \theta) \\ \mathcal{R}_b(\mathbf{x}_j^b; \theta) &= \sum_{\alpha=0}^1 \partial^\alpha \mathbf{E}(x_j^{\Gamma_\alpha}, z_j^b; \theta) - \partial^\alpha \mathbf{E}(x_j^{\Gamma_b}, z_j^b; \theta) \\ \mathcal{R}_r(\mathbf{x}_j^r; \hat{\mathbf{E}}(\mathbf{x}_j^r), \theta) &= \mathbf{E}(\mathbf{x}_j^r; \theta) - \hat{\mathbf{E}}(\mathbf{x}_j^r) \end{aligned} \quad (6)$$

where  $\hat{\mathbf{E}}(\mathbf{x}_j^r)$  is the reference solution at the given point  $\mathbf{x}_j^r$ . For the three-dimensional case,  $\mathcal{R}_i(\mathbf{x}_j^i, \varepsilon(\mathbf{x}_j^i); \theta)$  is substituted by the relevant formulation given by (1).

#### D. Domain Geometry

Given (1), we employed a rigorous-coupled wave analysis solver based on the waveguide method [10] to compute reference solutions. In this work, we simulate transmission-type metasurfaces that are illuminated by a monochromatic light with  $\lambda = 750 \text{ nm}$ . The plane wave is incident on the substrate, and subsequently propagates through the meta-atom layer. The domain size remains constant, with the  $x$ -dimension

set to 1500 nm (simulated by three units cells each with a period of 500 nm) and the  $z$ -dimension set to 1125 nm. In the three-dimensional case, the  $y$ -dimension is set to 500 nm. The typical height and width for each of the pillars are 500 nm and 250 nm, respectively. The basis configuration is then three pillars equally spaced between one another on top of a substrate. The spacing between the bottom of the domain and the substrate, as well as the substrate thickness, remain fixed across all simulations. Due to numerical inaccuracies, minor fluctuations in the pillar width and height may occur on a scale of tens of nanometers. The spatial resolution of our dataset corresponds to  $\frac{750}{120}$  nanometers per 'pixel'. Two- and three-dimensional representative domains can be seen in Fig. 2(a) and (b).

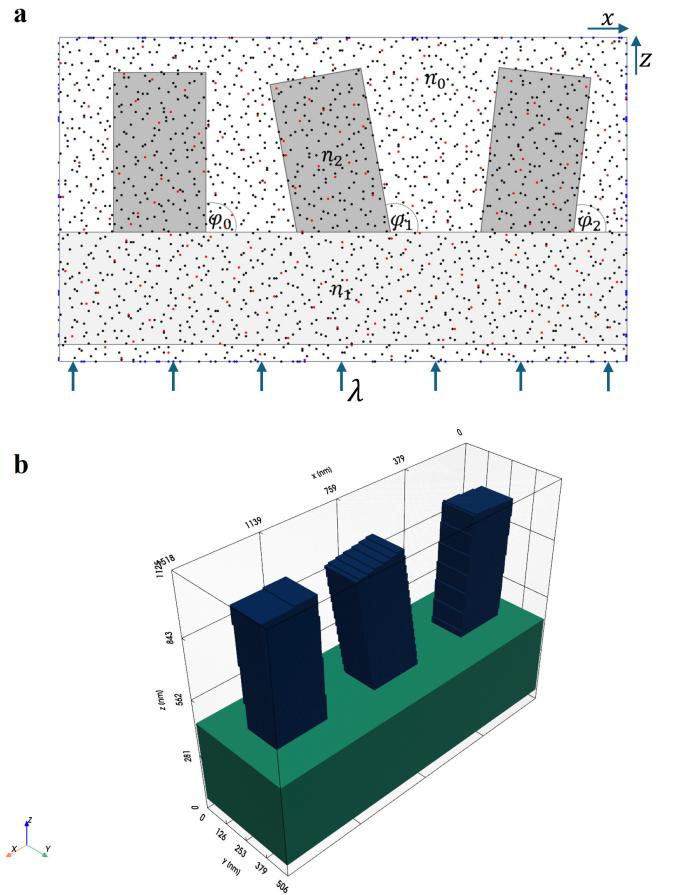


Fig. 2: A representative schematic of a (a) two-dimensional and (b) three-dimensional metasurface domain where  $n_0, n_1$  and  $n_2$  are the refractive indices for the different materials,  $(\varphi_0, \varphi_1, \varphi_2)$  are the inclination angles for the pillars, and  $\lambda$  is wavelength of the source plane wave. In (a) a representative training point cloud ( $N_t = 2048$ ) is drawn with boundary (blue), interior (black), and reference (red) points.

The metasurfaces are surrounded by air, which has a refractive index of  $n_0 \approx 1.0$ . For visible and near-infrared metasurface applications, dielectric materials such as silicon dioxide ( $SiO_2$ ) and titanium dioxide ( $TiO_2$ ) are often used because of their low loss at those wavelengths compared to metals. Thus, three different cases are considered:

- (I) For the two-dimensional  $SiO_2$  case both the pillars and substrate are made of  $SiO_2$  with refractive index  $n_1 =$

$n_2 = 1.45$  (see Sec. IV-Case I).

- (II) The same as (Case I), but in three-dimensions (see Sec. IV-Case II).
- (III) The two-dimensional  $\text{TiO}_2$ -on- $\text{SiO}_2$  metasurface case features  $\text{TiO}_2$  pillars ( $n_2 = 2.53$ ) on a  $\text{SiO}_2$  substrate ( $n_1 = 1.45$ ), corresponding to a practical high-index-contrast implementation (see Sec. IV-Case III).

To generate the datasets for the different cases, all geometric parameters of the domains are held constant except for the inclination angles of the three pillars, represented by the angles ( $\varphi_0, \varphi_1, \varphi_2$ ). These inclination angles are varied randomly, allowing the generation of distinct dataset configurations. Each inclination angle ranges from  $-10^\circ$  to  $10^\circ$  away from the substrate normal (aligned with the z-axis); this simulates realistic fabrication imperfections.

Each dataset then consists of a total of  $|\mathcal{D}| = 500$  samples split apart into  $\mathcal{D}_t \cup \mathcal{D}_v$  with training set  $\mathcal{D}_t$  of size 450 and a testing set  $\mathcal{D}_v$  of size 50.

### III. IMPLEMENTATION

Before training, a single point cloud is selected for each domain in  $\mathcal{D}_t$ , representing a unique material distribution. The point clouds were sampled using Halton sequences, with reference point locations fixed and shared across all domains in  $\mathcal{D}_t$ . In contrast, during testing, each domain in  $\mathcal{D}_v$  is subdivided into multiple point clouds. This leads to smoother, more consistent validation errors and images.

The Mean Absolute Percentage Error (MAPE) of the amplitude of the predicted electric field was used as a near-field validation metric. When looking at pointwise differences of the near-field, we use the Absolute Error (AE) and Mean Absolute Error (MAE) to maintain consistency with the visualizations of the near-field errors. The Absolute Percentage Error (APE) was employed when finding errors between different diffraction efficiencies of transmitted light. Since the period of the metasurface is smaller than the exposure wavelength, there are only one or a few propagating diffraction orders in the transmitted light. Subsequently, we analyze the diffraction efficiency of  $0^{\text{th}}$  diffraction order by near-to-far-field transformation using the fourier transform. Furthermore, the sign ( $\pm$ ) indicates standard deviation over the distribution.

Unless otherwise stated, each PIPN is trained using the Adam optimizer [37] for 50,000 iterations. Testing errors were computed at the end of each training. A batch size of 8 was used. A polynomial learning rate decay was employed, starting from an initial learning rate of  $10^{-3}$  and decaying to a final learning rate of  $8 \times 10^{-5}$ , with a polynomial decay coefficient of 4.5. A sine activation function was used in all layers except the final one. The weighting factors specified in Sec. II-C were set to 1, 1, and 10 for  $\lambda_i, \lambda_b$ , and  $\lambda_r$  respectively, as consistent with prior studies [38]. The implementation was done in Python using TensorFlow 1.x, and

all training was conducted on an NVIDIA Quadro P5000 GPU with 16 GB of VRAM.

## IV. RESULTS

To establish the optimal network configuration used in subsequent analyses, several key architectural and preprocessing considerations were investigated. First, we examined variations to the network structure by replacing the 1D convolutional layers after aggregation with dense layers (PointNetMLP) or a Chebychev-KAN (PointNetCheby) App. VI-B. Here we find that a PointNetMLP with  $N_\theta = 252,248$  trainable parameters strikes a good balance between computational cost and testing error. Furthermore, the specific input scaling constant necessary for achieving converging results in our experiments are thoroughly discussed in App. VI-A. Here we find that a scaling factor of approximately 21 is necessary for our problem to give plausible approximations. Lastly, the selection of input point clouds is another critical factor for the performance and computational efficiency of PIPNs. Our investigation (presented in App. VI-C Fig. 9) reveals that optimal performance is achieved with  $N_t = 2048$ ,  $N_r = 224$  and  $N_b = 128$ , which yields the lowest near-field MAPE.

### Case I - 2D $\text{SiO}_2$ Metasurface

Fig. 3 depicts the near-field of a representative test domain. Columns show reference, predicted electric fields and AE between the two; rows show the real, imaginary, amplitude and phase.

The near-field MAPE for the test domain in Fig. 3 is 1.59%, whereas the near-field MAPE for  $\mathcal{D}_v$  is  $1.69 \pm 0.43\%$ . Similarly the diffraction efficiency APE for all test domains is  $1.95 \pm 1.1\%$ . The training took around 2 hours and inference for an entire testing domain is on the scale of 100ms. The MAE for  $\mathcal{D}_v$  is around 0.02 with regions around the material interface and boundaries reaching values as high as 0.08. A detailed plot showing the dependence of the MAE with respect to the distance from the nearest material interface can be seen in App. VI-C in Fig. 10. The results demonstrate that the pointwise AE decreases linearly with increasing distance to the nearest material interface, reducing by approximately 50% from 0.028 to 0.015 as the distance approaches 200 nm. Conversely, the pointwise AE exhibits an exponential increase with distance to the nearest training reference point, more than doubling from approximately 0.018 to 0.042 as the distance reaches 130 nm (see Fig. 10).

### Case II - 3D $\text{SiO}_2$ Metasurface

The network output now includes the  $E_x$ ,  $E_y$ , and  $E_z$  fields (instead of just  $E_y$  for the two-dimensional case) according to (1). The training iterations are increased to 100,000, and the point cloud is doubled in size (i.e  $N_t = 4096$ ,  $N_b = 256$  and  $N_r = 448$ ) to account for the larger domain. Furthermore we increase the network size to  $N_\theta = 448 \times 10^3$ . Due to this, the memory requirements went up significantly and the batch size had to be decreased to 2 for us to still be able to train on a 16GB VRAM GPU. The training time increased by an order

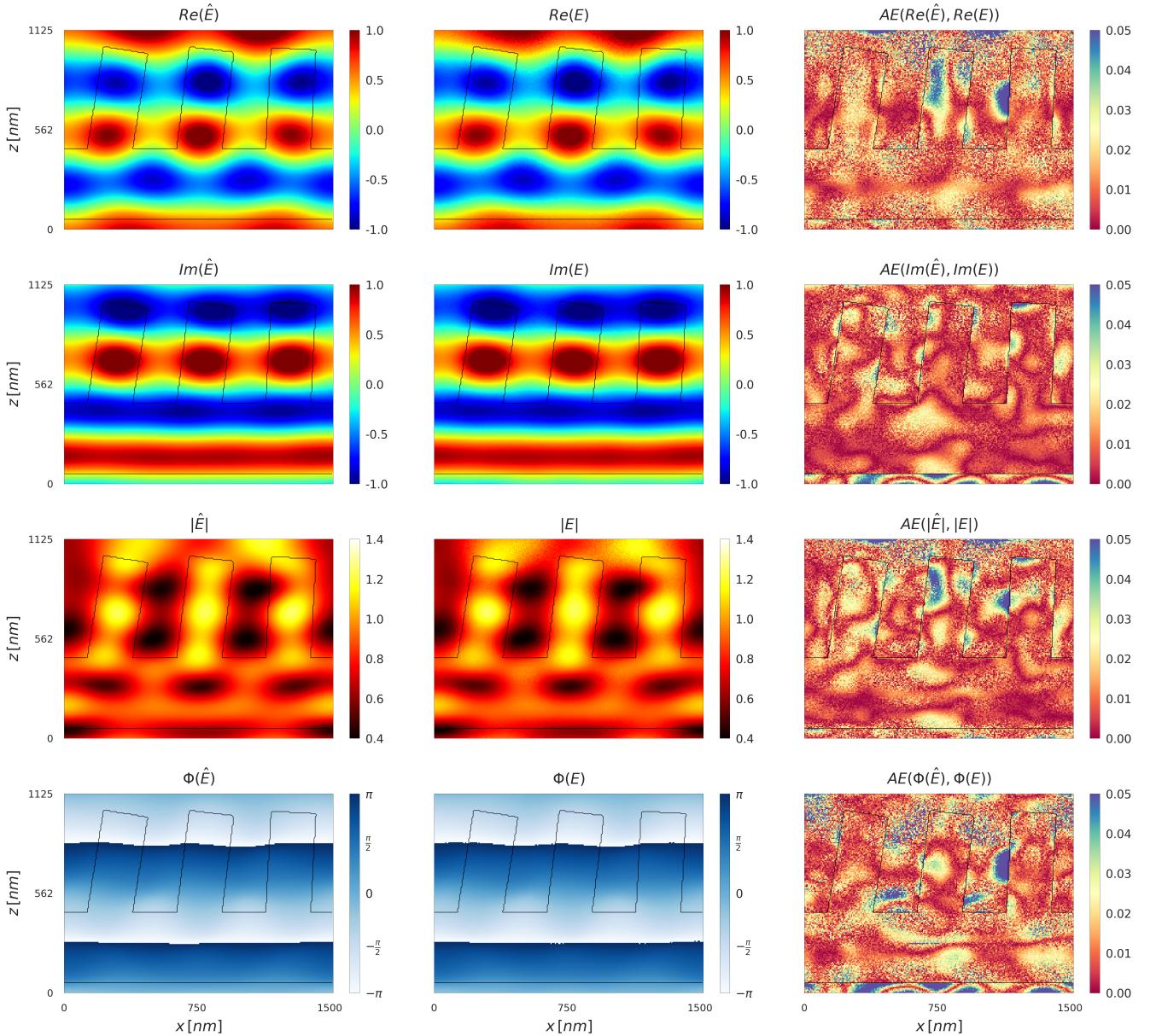


Fig. 3: A representative approximate solution (near-field MAPE = 1.59%), the reference solution, and its errors of a representative test domain of the *SiO<sub>2</sub>* dataset. The first column shows the real, imaginary, amplitude, and phase of the reference electric field. The second column shows the PIPN solution, and the third column shows the MAE between them. The metasurface contour is highlighted in black.

of magnitude to approximately 24 hours. Fig. 4 illustrates the reference solution and predicted near-fields of a representative test domain slice ( $y \approx 250 \text{ nm}$ ). The entire domain has a near-field MAPE of 4.04%, whereas set  $\mathcal{D}_v$  has a near-field MAPE of  $3.04 \pm 0.09\%$ , and a diffraction efficiency APE of  $1.12 \pm 0.28\%$ . The near-field MAPE distribution over all slices in the  $x$ -,  $y$ -, and  $z$ -directions can be seen in Fig. 6. The increased error, starting at the material interfaces and going into the pillars, is observable in Fig. 6(a). No error jump on the material interface is discernible, due to the averaging over  $\mathcal{D}_v$  (i.e., the angles  $\varphi_0, \varphi_1, \varphi_2$  in the  $x$ -direction of the pillars are different, and thus no clear transition can be seen). The error profile is symmetric, reflecting the symmetry of the material properties with respect to the pillar center in the  $y$ -direction (Fig. 6(b)).

A steep decrease in near-field MAPE below approximately 125 nm and above 375 nm coincides with the substrate/pillar interface.

### Case III - 2D TiO<sub>2</sub>-on-SiO<sub>2</sub> Metasurface

We use a network of size  $N_\theta = 447,594$ . Besides this, no network parameters were changed when compared to Case I. The training took 3.92 hours. The near-field approximation of a representative test domain with MAPE = 10.01% is shown in Fig. 5. The testing domains  $\mathcal{D}_v$  have a near-field MAPE of  $10.72 \pm 5.64\%$  and a diffraction efficiency APE of  $6.6 \pm 3.37\%$ . The largest point wise AE is found in the regions between the pillars and around the material interface. Moreover, the predicted solution appears grainy compared

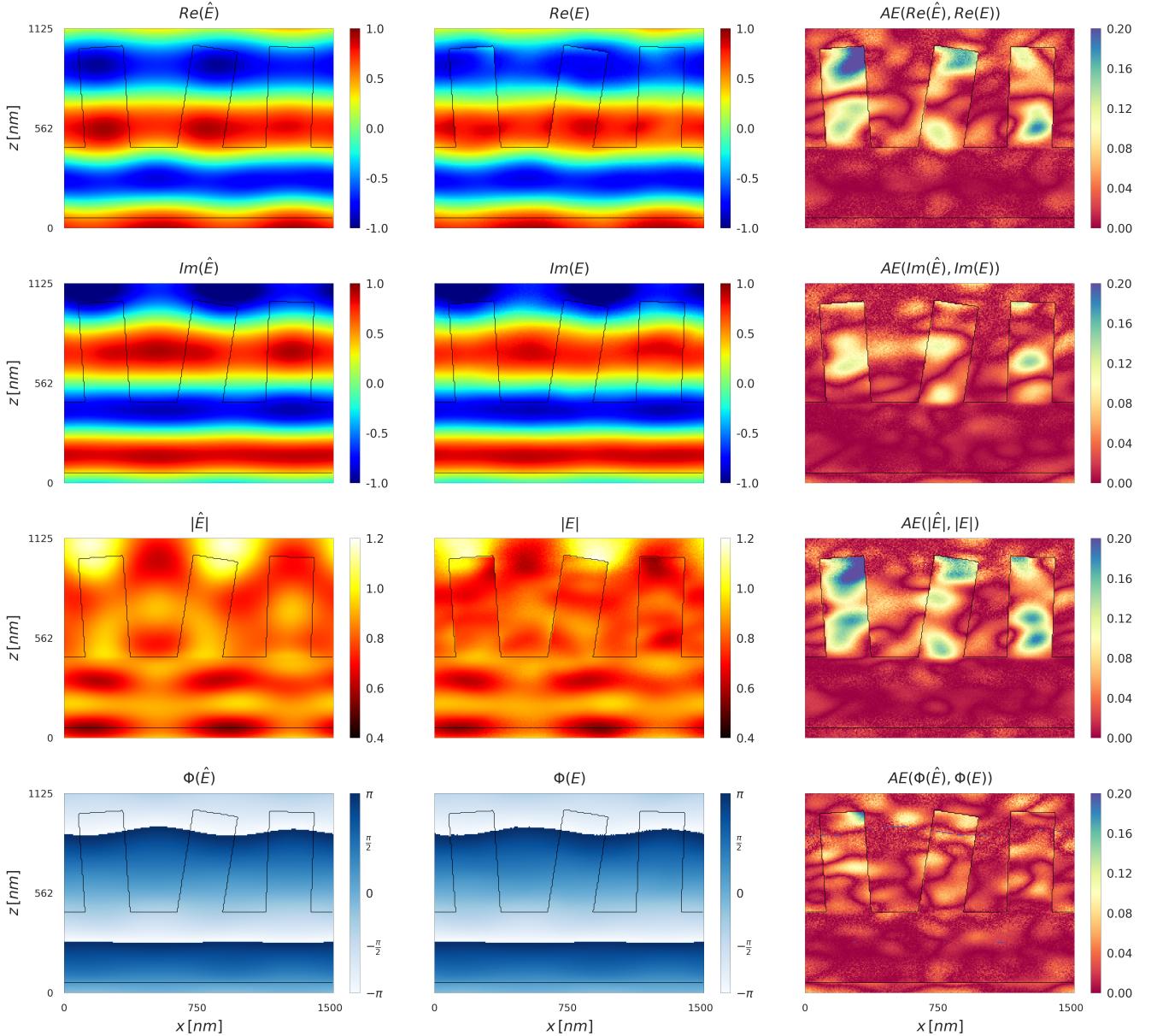


Fig. 4: Representative solution for the 3D  $SiO_2$   $E_y$  field at  $y \approx 250$  nm, with near-field MAPE = 4.04%. Metasurface geometry and material as indicated in Fig. 2(b).

to Fig. 3, suggesting that this artifact is exaggerated in the presence of larger material contrast. The MAE for the testing dataset is around 0.04 with select regions around the material interface and boundaries reaching values as high as 0.2.

## V. DISCUSSION

The results presented in Sec. IV demonstrate the robust generalization capabilities of the PIPN framework for metasurfaces with varying refractive indices. This establishes that variable PDE parameters can be effectively encoded within the PIPN architecture. Future investigations should examine the performance characteristics of PIPNs that incorporate spatially and temporally dependent PDE parameters across a wide variety of PDE problems. Additionally, a particular area of future study for PIPN applications should be the

generalization across diverse domain geometries, as opposed to the square domains exclusively considered in this study. Tab. I summarizes the results obtained in Sec. IV. Across all cases, a clear correlation between material contrast and prediction accuracy is evident: the  $SiO_2$  (Case I + II) dataset achieved the best performance, while the higher contrast  $TiO_2$  (Case III) system exhibited significantly increased errors.

This was to be expected, as PINN generally tend to do better when approximating smoother functions. We note that the largest error is observed near and within the meta-atom layer, where local field enhancement and strong scattering effects cause significant field perturbations (see Fig. 6(c)). Therefore, the concentration of points around the meta-atom layer is not guaranteed to be large enough to successfully capture the interface. Future work should investigate adaptive

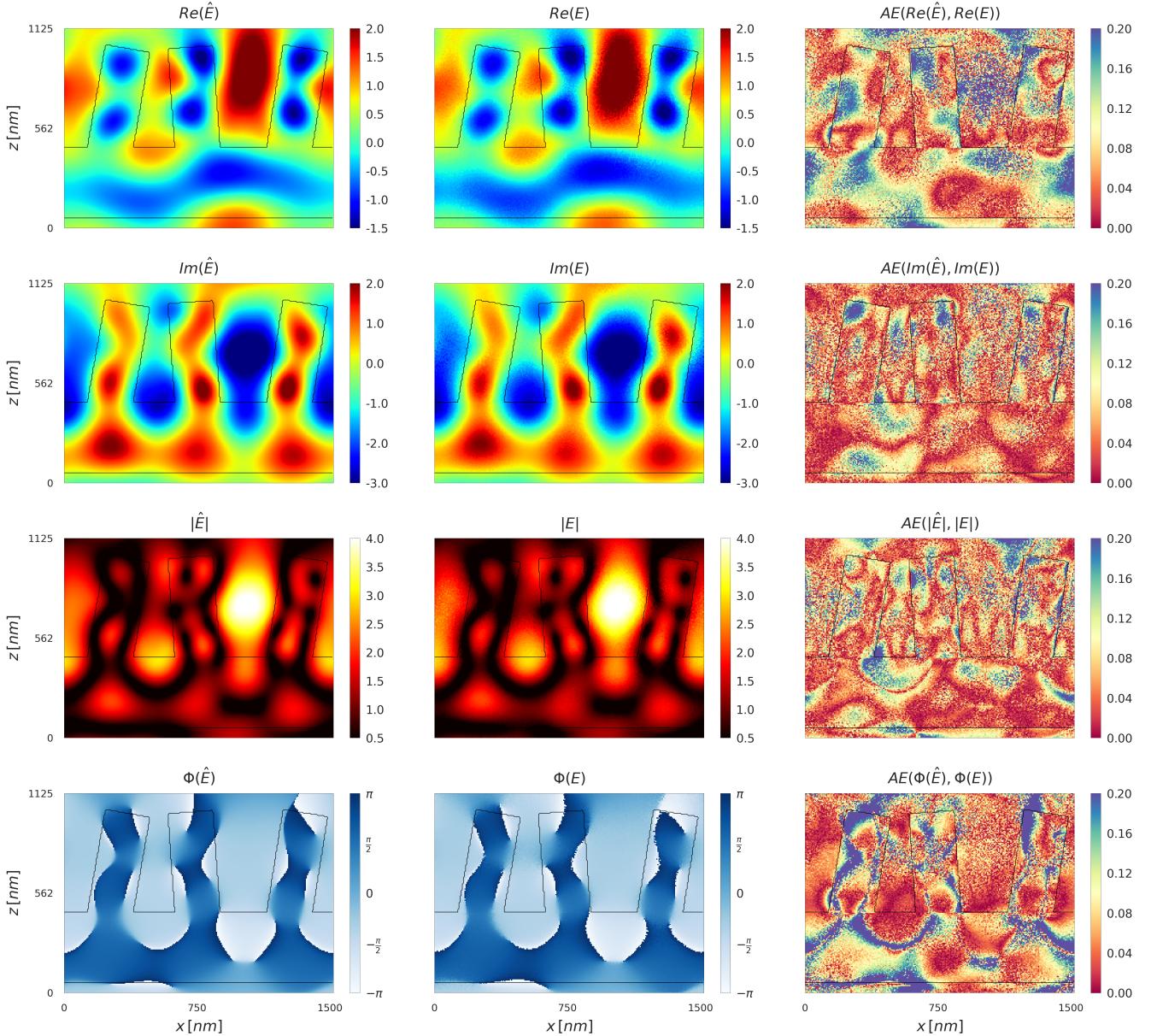
Fig. 5: Representative solution for the 2D  $TiO_2$  dataset, with near-field MAPE = 10.01%.

TABLE I: Summary of results presented in Sec. IV. Error metrics represent the mean and standard deviation computed over the testing dataset  $\mathcal{D}_v$  for both near-field predictions and diffraction efficiency calculations.

Dataset	$SiO_2$		$TiO_2$
	Case I 2D	Case II 3D	Case III 2D
Network Size [ $N_\theta$ ]	252,248	448,846	447,594
Training Time [hours]	1.96	24.6	3.92
Near-Field MAPE [%]	$1.69 \pm 0.43$	$3.84 \pm 0.13$	$10.72 \pm 5.64$
Diffraction Efficiency APE [%]	$1.95 \pm 1.10$	$3.91 \pm 4.18$	$6.60 \pm 3.37$

point sampling strategies [39], which have shown promise in mitigating this issue.

The two-dimensional problems (Case I + III) have a small training time of only a few hours. Here, the computational requirements were low enough in both cases that with a decrease

in batch size and training point cloud size, the networks could have been trained on 8GB VRAM GPUs without a significant increase in testing error. A steep increase in training time was observed for Case II; training took around 24 hours due to the increased computational requirements, which stemmed from the vectorial formulation of the Helmholtz equation (1). Nonetheless, our network shows great promise in terms of the required computational resources compared to state-of-the-art grid-based physics-informed NN approaches [26]. While testing errors between the two are comparable, we are able to train the PIPN on a NVIDIA Quadro P5000 GPU with 16 GB of VRAM in a day compared to the week-long training on a NVIDIA A80 80GB VRAM GPU required for its grid-based counterpart.

Additionally, for all three cases, an increase in error can be seen at the edges of the domain. This can be explained by

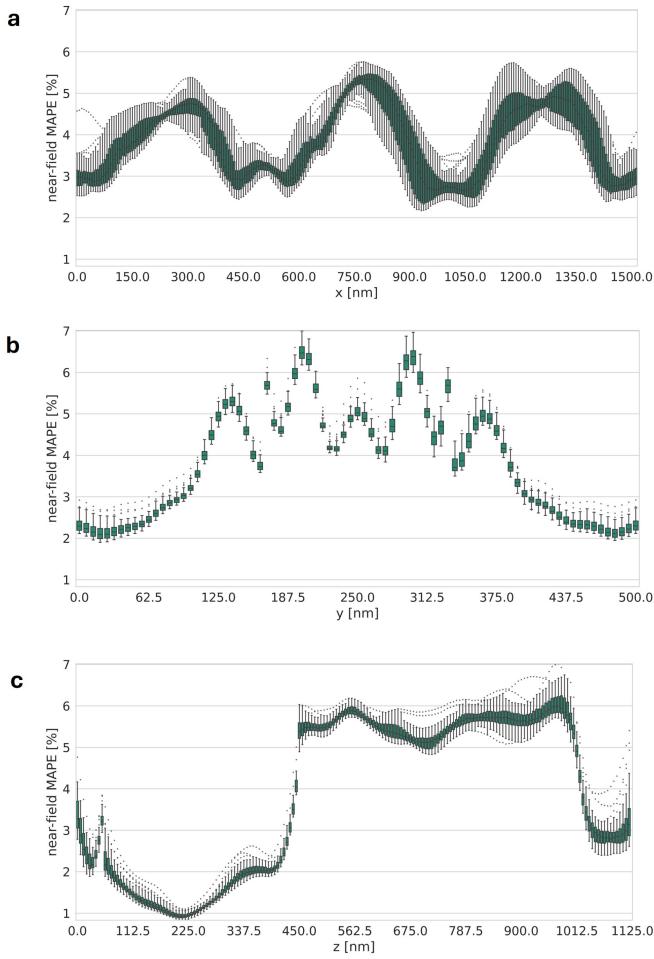


Fig. 6: The near-field MAPE for the three-dimensional test dataset when (a) when slicing in the  $x$ -dimension (i.e. looking at the results in the  $yz$ -plane), (b) when slicing in the  $y$ -dimension and (c) when slicing in the  $z$ -dimension. Metasurface geometry and material as indicated in Fig. 2(b).

the fact that the point density is smallest at the edge. Future work—similar to App. VI-B—should be done on adapting the PointNet architecture to further mitigate computational requirements; also, to test whether or not the proposed architectural changes hold up for other PDE problems. We have tested architectures involving Fourier-KAN [40], RBF-KAN [41], and B-Spline-KAN [42], as well as one implementation that replaces the convolutional kernels used in the PIPN framework by B-Splines. No converging results were obtained, and thus were omitted from this work. However, when used as a standalone network to solve the ungeneralized problem, Fourier-KAN showed great promise with lower training time, network size, and testing errors.

Lastly, further investigation is required to characterize the failure modes of PIPNs. As demonstrated in App. VI-A, the convergence of PIPN to a physically meaningful approximation exhibits strong sensitivity to the input coordinate scaling. Consequently, conventional normalization strategies that scale NN inputs to magnitudes comparable to the outputs [14], [18], [43] may not be universally applicable across all PIML formulations.

## VI. CONCLUSION

This work examined how well PIPNs perform when solving the optical scattering problem involving all-dielectric metasurfaces that contain nanopillars with varying inclinations and refractive indices. The proposed method is a mesh-free, PIML approach that achieves competitive accuracy and generalizability while relying only on weak supervision from sparse reference solution while requiring a fraction of the computational resources of its grid-based counterparts [44]. In particular, our PIPN enables us to encode the spatially varying PDE parameter, which allows the network to differentiate domains based on the underlying material distribution. Furthermore, although not an intrinsic requirement, the inclusion of reference points may prove fruitful for optical applications where the source is not fully defined, but experimentally-collected reference data is available. While the framework exhibited promising performance in various scenarios, it encountered notable limitations, especially in regards to domains characterized by large material contrast. These limitations produced noticeable artifacts and increased the prediction errors, both of which highlight the challenges that PIPNs face when approximating sharp transitions. Hyperparameter tuning provided valuable insights into optimal input point cloud configuration and PointNet architectures. Although further hyperparameter optimization was outside the scope of this study, the limited improvements observed suggest that the current parameters already achieve near-optimal performance.

## APPENDIX

### A. Error Dependence on the Input Scale

Fig. 7 shows the near-field MAPE distribution over all testing domains when the input coordinates and wavelength are scaled by different constants.

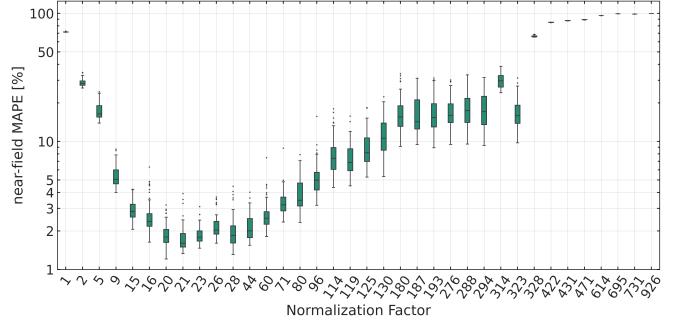


Fig. 7: This figure illustrates the relationship between the normalization factor (of the input spatial coordinates and wavelength) and the near-field MAPE of the testing domains.

The best results are achieved with a scaling factor around 21 (note the non-uniform  $x$ -axis). Using this, the input  $x$ -coordinates get scaled to around  $\pm 35$ , and the  $z$ -coordinates are scaled to around  $\pm 18$ . This is still an order of magnitude larger than the output of the network (real and imaginary parts of the electric field); that output is on a scale of around  $\pm 1.1$  for the two- and three-dimensional  $SiO_2$  datasets, and -3.7 to 2.3 for the  $TiO_2$ -on- $SiO_2$  dataset. The  $x$ - and  $z$ -coordinates are an order of magnitude larger than the permittivity, which

ranges from 1.0 to 2.1 for the  $SiO_2$  datasets and 1.0 to 6.4 for the  $TiO_2$ -on- $SiO_2$  dataset.

### B. Comparison of Assorted PointNet Architectures

**Fig. 8** compares the performance of three PointNet architecture variations across different network sizes, where each network size is scaled by proportionally decreasing the number of nodes in each layer. PointNetCNN (orange) represents the unchanged architecture depicted in **Fig. 1**; PointNetMLP (green) replaces the second half of convolutional kernels (as indicated by the dashed squared in **Fig. 1**) with a three-layer dense neural network. PointNetCheby (blue) similarly substitutes the second half with a three-layer KAN that employs Chebyshev basis functions.

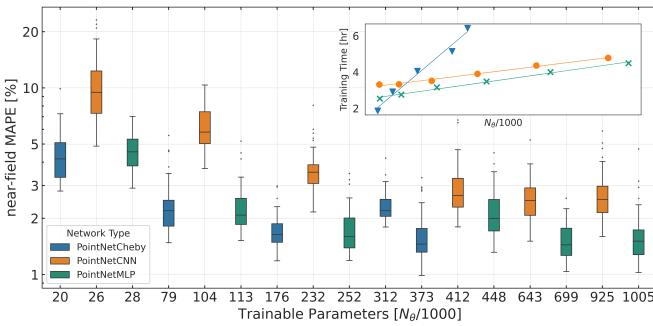


Fig. 8: Box plots of the test near-field MAPE versus the network size ( $N_\theta$ ) for three different variations of the PointNet architecture. The inset plot shows the relative training time (hours) versus the same network sizes.

All three architectures demonstrate continuous improvement in mean near-field MAPE with increasing network size, though they exhibit distinct performance characteristics. PointNetCheby achieves the best accuracy across most network sizes, reaching a minimum mean near-field MAPE of  $1.71 \pm 0.37\%$  at approximately  $373 \times 10^3$  parameters before GPU VRAM limitations are encountered. PointNetMLP shows comparable performance to PointNetCheby at smaller network sizes, achieving  $1.66 \pm 0.53\%$  mean near-field MAPE at  $252 \times 10^3$  parameters. In contrast, PointNetCNN consistently exhibits the highest error rates, with its best performance of  $2.69 \pm 0.94\%$  mean near-field MAPE occurring at the largest tested network size:  $995 \times 10^3$  parameters.

The inset plot reveals linear scaling of training time with network size across all architectures. The training times range from 3 to 6.5 hours, with the PointNetCheby demonstrating the worst scaling behavior and the longest training times. PointNetMLP and PointNetCNN exhibit similar scaling behavior to each other, with PointNetMLP consistently requiring about 0.5 hours less training time and reaching maximum training times of approximately 4.5 hours.

### C. Error Dependence on Reference Points and Distance to Material Interface

When working with point clouds, the selection of points can significantly impact performance. The three key components within our point cloud are: boundary points, interior points,

and reference points. Intuitively, the optimal configuration would be achieved by maximizing the number of points in each category, thus giving the NN the largest possible amount of information about the system. However, increasing the number of reference points raises dependence on a priori data which can be costly to obtain.

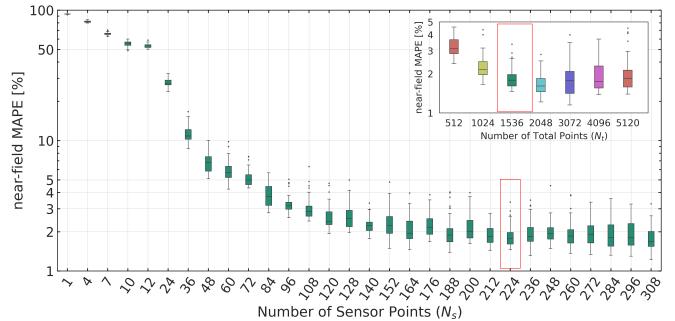


Fig. 9: Box plots of the test near-field MAPE versus the number of reference points ( $N_r$ ) used during training. The inset plot shows how varying the number of total points ( $N_t$ ) used for training affects the near-field MAPE at a constant  $N_r = 224$ .

**Fig. 9** shows the dependence of the near-field MAPE on the number of reference points and total points used during training. The figure depicts the MAPEs distribution over the 50 testing domains when varying the number of reference points used during training ( $N_t = 1536$ ,  $N_b = 128$  and  $N_i = 1536 - 128 - N_r$ ). Results improve until  $N_r = 224$ , when a near-field MAPE of 1.88% is obtained; after that, the mean MAPE does not change significantly. The inset plot depicts the error distribution when varying  $N_t$  ( $N_r = 224$ ,  $N_b = 128$  and  $N_i = N_t - 128 - 224$ ). The average MAPE decreases until  $N_t = 2048$ , at which a near-field MAPE of 1.67% is obtained. After that point, it increases slightly until  $N_t = 5120$ . This is likely due to small network size, which prevents the network from effectively leveraging additional information. However, the plateau could also be explained by the network overfitting on the reference points and neglecting physics.

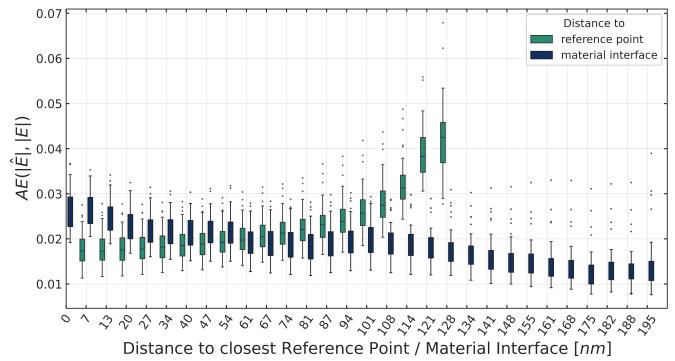


Fig. 10: Correlation between the near-field MAE (averaged over the testing set  $D_v$  from Case I - 2D  $SiO_2$  Metasurface) at each test domain point and its minimum distance from the corresponding training reference points (green) and material interface (blue).

**Fig. 10** shows the near-field MAE distribution at each point versus the minimum distance to the closest material

interface and reference training points. The MAE increases continuously as the distance to the closest reference point increases. Most points are  $60\text{ nm}$  away from the nearest reference solution, with decreasing numbers for points that are either closer or further away from reference points. The near-field MAE distribution at each point continuously decreases as the distance to the closest material interface increases. Most points in the point cloud are within  $100\text{ nm}$  of a material interface.

### ACKNOWLEDGMENT

We would like to thank Dr. Daniel Tenbrinck and Dr. Andreas Erdmann for their guidance and valuable assistance in the selection of this research topic.

### REFERENCES

- [1] D. Neshev and I. Aharonovich, "Optical metasurfaces: new generation building blocks for multi-functional optics," *Light: Science & Applications*, vol. 7, no. 1, p. 58, Aug. 2018, publisher: Nature Publishing Group.
- [2] A. I. Kuznetsov, M. L. Brongersma, J. Yao, M. K. Chen, U. Levy, D. P. Tsai, N. I. Zheludev, A. Faraon, A. Arbabi, N. Yu *et al.*, "Roadmap for optical metasurfaces," *ACS photonics*, vol. 11, no. 3, pp. 816–865, 2024.
- [3] N. Yu, P. Genevet, F. Aieta, M. A. Kats, R. Blanchard, G. Aoust, J.-P. Tetienne, Z. Gaburro, and F. Capasso, "Flat optics: controlling wavefronts with optical antenna metasurfaces," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 3, pp. 4700423–4700423, 2013.
- [4] J. Qin, S. Jiang, Z. Wang, X. Cheng, B. Li, Y. Shi, D. P. Tsai, A. Q. Liu, W. Huang, and W. Zhu, "Metasurface micro/nano-optical sensors: principles and applications," *ACS nano*, vol. 16, no. 8, pp. 11598–11618, 2022.
- [5] H. Kim, H. Yun, S. Jeong, S. Lee, E. Cho, and J. Rho, "Optical Metasurfaces for Biomedical Imaging and Sensing," *ACS Nano*, vol. 19, no. 3, pp. 3085–3114, Jan. 2025, publisher: American Chemical Society.
- [6] Y. Hu, X. Luo, Y. Chen, Q. Liu, X. Li, Y. Wang, N. Liu, and H. Duan, "3D-Integrated metasurfaces for full-colour holography," *Light: Science & Applications*, vol. 8, no. 1, p. 86, Sep. 2019, publisher: Nature Publishing Group.
- [7] S. Nie and I. F. Akyildiz, "Metasurfaces for multiplexed communication," *Nature Electronics*, vol. 4, no. 3, pp. 177–178, Mar. 2021, publisher: Nature Publishing Group.
- [8] C. Liu, Q. Ma, Z. J. Luo, Q. R. Hong, Q. Xiao, H. C. Zhang, L. Miao, W. M. Yu, Q. Cheng, L. Li, and T. J. Cui, "A programmable diffractive deep neural network based on a digital-coding metasurface array," *Nature Electronics*, vol. 5, no. 2, pp. 113–122, Feb. 2022, publisher: Nature Publishing Group.
- [9] F. L. Teixeira, "Time-Domain Finite-Difference and Finite-Element Methods for Maxwell Equations in Complex Media," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2150–2166, Aug. 2008.
- [10] P. Evanschitzky and A. Erdmann, "Fast near field simulation of optical and EUV masks using the waveguide method," in *Proceedings of the SPIE, Volume 6533, article id. 65330Y*, vol. 6533, Feb. 2007, p. 65330Y, conference Name: 23rd European Mask and Lithography Conference ADS Bibcode: 2007SPIE.6533E.OYE.
- [11] C. Kang, C. Park, M. Lee, J. Kang, M. S. Jang, and H. Chung, "Large-scale photonic inverse design: computational challenges and breakthroughs," *Nanophotonics*, vol. 13, no. 20, pp. 3765–3792, 2024.
- [12] S. An, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, Y. Dong, M. Haerinia, A. M. Agarwal, C. Rivero-Baleine, M. Kang, K. A. Richardson, T. Gu, J. Hu, C. Fowler, and H. Zhang, "Deep Convolutional Neural Networks to Predict Mutual Coupling Effects in Metasurfaces," *Advanced Optical Materials*, vol. 10, no. 3, p. 2102113, 2022.
- [13] C. Li, J. Chen, Q. Lin, and Y. Han, "PISC-Net: A Comprehensive Neural Network Framework for Predicting Metasurface Infrared Emission Spectra," *ACS Applied Materials & Interfaces*, vol. 16, no. 32, pp. 42816–42827, Aug. 2024, publisher: American Chemical Society.
- [14] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What's Next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Jul. 2022.
- [15] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations," Nov. 2017, arXiv:1711.10566 [cs].
- [16] J. D. Toscano, V. Oomen, A. J. Varghese, Z. Zou, N. Ahmadi Daryanari, C. Wu, and G. E. Karniadakis, "From PINNs to PIKANs: recent advances in physics-informed machine learning," *Machine Learning for Computational Science and Engineering*, vol. 1, no. 1, p. 15, Mar. 2025.
- [17] V. Gopakumar, S. Pamela, and D. Samaddar, "Loss landscape engineering via Data Regulation on PINNs," *Machine Learning with Applications*, vol. 12, p. 100464, Jun. 2023.
- [18] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An Expert's Guide to Training Physics-informed Neural Networks," Aug. 2023, arXiv:2308.08468 [cs].
- [19] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3055–A3081, Jan. 2021, publisher: Society for Industrial and Applied Mathematics.
- [20] J. Tu, C. Liu, and P. Qi, "Physics-Informed Neural Network Integrating PointNet-Based Adaptive Refinement for Investigating Crack Propagation in Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2210–2218, Feb. 2023.
- [21] D. H. Nguyen, T. H. Nguyen, K. D. Tran, and K. P. Tran, "Physics-Informed Machine Learning for Industrial Reliability and Safety Engineering: A Review and Perspective," in *Artificial Intelligence for Safety and Reliability Engineering: Methods, Applications, and Challenges*, K. P. Tran, Ed. Cham: Springer Nature Switzerland, 2024, pp. 5–23.
- [22] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Optics Express*, vol. 28, no. 8, pp. 11618–11633, Apr. 2020.
- [23] Q. Deng, H. Tang, R. Li, W. Li, M. Zhang, Z. Shi, and Y. Zhang, "Application of PINNs in PNJ Research," in *2024 Photonics & Electromagnetics Research Symposium (PIERS)*. IEEE, 2024, pp. 1–8.
- [24] Y. Wang and S. Zhang, "Multi-receptive-field physics-informed neural network for complex electromagnetic media," *Opt. Mater. Express*, vol. 14, no. 11, pp. 2740–2754, Nov. 2024.
- [25] K. Kojima, T. Koike-Akino, Y. Tang, and Y. Wang, "Inverse Design for Integrated Photonics Using Deep Neural Network," in *Integrated Nanophotonics: Platforms, Devices, and Applications*. Wiley, 2023, pp. 209–243.
- [26] V. Medvedev, A. Erdmann, and A. Rosskopf, "3D EUV mask simulator based on physics-informed neural networks: effects of polarization and illumination," in *Computational Optics 2024*, vol. 13023, International Society for Optics and Photonics. SPIE, 2024, p. 1302304.
- [27] V. Medvedev, A. Erdmann, and A. Rosskopf, "3D mask simulation and lithographic imaging using physics-informed neural networks," in *Optical and EUV Nanolithography XXXVII*, vol. 12953. SPIE, Apr. 2024, pp. 208–224.
- [28] V. Medvedev, A. Erdmann, and A. Rosskopf, "Physics-informed deep learning for 3D modeling of light diffraction from optical metasurfaces," *Opt. Express*, vol. 33, no. 1, pp. 1371–1384, Jan 2025.
- [29] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escalano, M. Cazorla, and J. Azorin-Lopez, "PointNet: A 3D Convolutional Neural Network for real-time object class recognition," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 1578–1584, iSSN: 2161-4407.
- [30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.
- [31] A. Kashefi and T. Mukerji, "Physics-informed PointNet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries," *Journal of Computational Physics*, vol. 468, p. 111510, Nov. 2022.
- [32] A. Kashefi, L. J. Guibas, and T. Mukerji, "Physics-informed PointNet: On how many irregular geometries can it solve an inverse problem simultaneously? Application to linear elasticity," Sep. 2023, arXiv:2303.13634 [cs].
- [33] A. Kashefi and T. Mukerji, "Physics-informed pointnet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries," *J. Comput. Phys.*, vol. 468, p. 111510, 2022.
- [34] Z. Zhang, "A physics-informed deep convolutional neural network for simulating and predicting transient darcy flows in heterogeneous reservoirs without labeled data," *Journal of Petroleum Science and Engineering*, vol. 211, p. 110179, 2022.
- [35] I. Graham, O. Pembery, and E. Spence, "The Helmholtz equation in heterogeneous media: A priori bounds, well-posedness, and resonances,"

- Journal of Differential Equations*, vol. 266, no. 6, pp. 2869–2923, Mar. 2019.
- [36] J. Lim and D. Psaltis, “MaxwellNet: Physics-driven deep neural network training based on Maxwell’s equations,” *APL Photonics*, vol. 7, no. 1, p. 011301, Jan. 2022.
  - [37] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Jan. 2017, arXiv:1412.6980 [cs].
  - [38] A. Kashefi and T. Mukerji, “Prediction of fluid flow in porous media by sparse observations and physics-informed PointNet,” *Neural Networks*, vol. 167, pp. 80–91, Oct. 2023.
  - [39] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, “A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115671, Jan. 2023.
  - [40] A. A. Imran and M. F. Ishmam, “FourierKAN outperforms MLP on Text Classification Head Fine-tuning,” Sep. 2024, arXiv:2408.08803 [cs].
  - [41] Z. Li, “Kolmogorov-Arnold Networks are Radial Basis Function Networks,” May 2024, arXiv:2405.06721 [cs].
  - [42] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, “KAN: Kolmogorov-Arnold Networks,” Jun. 2024, arXiv:2404.19756 [cond-mat, stat].
  - [43] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256, iSSN: 1938-7228.
  - [44] V. Medvedev, A. Erdmann, and A. Roskoppf, “Physics-informed deep learning for 3D modeling of light diffraction from optical metasurfaces,” *Optics Express*, vol. 33, no. 1, p. 1371, Jan. 2025.

**Leon Armbuster** received his MSc in Applied and Computational Mathematics from Friedrich Alexander-Universität, Erlangen, Germany, in 2025. He is currently working in the AI-Augmented Simulation research group at the Fraunhofer IISB in Erlangen, Germany.



**Vlad Medvedev** received his MSc in Advanced Optical Technologies from Friedrich Alexander-Universität, Erlangen, Germany, in 2022. He is currently working towards a PhD in physics-informed deep learning for computational lithography and design of nano-optical devices with the Computational Lithography and AI-Augmented Simulation research groups at the Fraunhofer IISB in Erlangen, Germany. His research interests include the implementation of computer vision and deep learning solutions to enhance lithographic output.



**Dr. Andreas Roßkopf** received his Diploma in Applied Mathematics and the PhD in Engineering from Friedrich-Alexander-Universität, Erlangen, Germany, in 2008 and 2018, respectively. He has worked in numerous positions in the automotive, mobility, and energy industries. Since 2018, he has been leading the research group AI-Augmented Simulation at the Fraunhofer IISB in Erlangen, Germany. His research interests include coupled numerical simulation, design automation, and optimization of engineering processes through data and AI.

