Deep Inverse Modeling the Near Field Response of Optical Metasurfaces

Saad Lahrichi[a], Ethan J. Mick[a], Marshall B. Lindsay[a], Scott D. Kovaleski[a], Derek T. Anderson[a], Jordan M. Malof[a], Stanton R. Price[b], and Steven R. Price[c]

[a]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA
[b]US Army Rapid Capabilities and Critical Technologies Office, Huntsville, AL, USA
[c]US Army Engineer Research and Development Center, Vicksburg, MS, USA

ABSTRACT

In this work, we investigate the problem of real-time design of electromagnetic (EM) metamaterials to achieve custom scattering properties: a type of inverse modeling problem. To address this problem, we investigate a class of DNN-based models that are specially designed to address inverse problems, termed deep inverse models (DIMs). DIMs have recently shown tremendous promise for solving material design problems, however, relatively less work has been done for high-dimensional problems, such as near-field design. In this work, we performed 1500 simulations of a metasurface with a 3x3 array of meta-atom pillars, where we independently and randomly-varied the radii of each pillar and recorded the resulting electric near-field values. We then used this dataset to train and evaluate several data-driven inverse models, including several variations of a recently-successful DIM, termed the *Tandem*. Our results indicate that the Tandem is capable of making relatively accurate design predictions in this challenging high-dimensional settings, and doing so in real-time (e.g., roughly 4 ms). We find that the choice of model architecture significantly impacts the accuracy of the inverse model, and even higher accuracy can be achieved with further improvements to the Tandem's design.

Keywords: inverse design, metamaterials, physics inspired neural networks, deep learning

1. INTRODUCTION

Electromagnetic (EM) Metamaterials are a class of materials whose properties primarily depend upon their physical structure, or *geometry*, rather than the properties of their bulk building materials. With the proper geometric design, metamaterials can exhibit powerful properties that are difficult or impossible to achieve with conventional materials; some examples include negative refractive indices,[1] cloaking,[2] and perfect absorption.[3] Despite their success, it is plausible that even more powerful properties might be achieved, given a sufficiently complex design. This has motivated researchers to pursue the modeling and design of increasingly complex metamaterials in recent years.

In the metamaterial design problem, we assume that the metamaterial is governed by some relationship of the form $y = f(x)$, where $x \in \mathcal{X}$ refers to the design of the metamaterial and y refers to the properties of interest. For example, x is often a numerical vector characterizing the geometric structure of the material (e.g., radii or height of a cylindrical metamaterial,[4,5] or thickness of a material layer[5]). Meanwhile, y often encodes some EM properties of interest, such as an absorption or transmission spectrum (e.g., y is a real-valued vector),[4] or the spatial distribution of electric or magnetic fields (e.g., y is a complex-valued matrix).[6] Given this formalism, the goal of metamaterial design is to infer a geometry, denoted x^* with the property that $f(x^*) \approx y^*$, where y^* denotes some desired property.

Metamaterial design is a widely-studied problem, and a large number of methods have been developed to solve it, depending upon the precise characteristics of the design problem.[7] For complex metamaterials, however, the relationship f can be highly intricate or completely unknown, rendering many conventional design methods

Corresponding author: Saad Lahrichi
E-mail: saad.lahrichi@missouri.edu

Advanced Optics for Imaging Applications: UV through LWIR X, edited by Jay N. Vizgaitis, Peter L. Marasco, Jasbinder S. Sanghera, Proc. of SPIE Vol. 13466, 1346609
2025 Published by SPIE · 0277-786X · doi: 10.1117/12.3053476

ineffective or inapplicable. Even for highly complex metamaterials, it is often feasible to evaluate their properties given a specific design using numerical simulation techniques. Consequently, it is possible to collect a dataset of I input/output pairs, denoted $d = (x_i, y_i)_{i=1}^{I}$, that can then be utilized to train data-driven models, such as deep neural networks (DNNs), to approximate f. In recent years, DNNs have been shown capable of accurately approximating highly complex metamaterials where x and y are high dimensional, and the underlying function f is highly non-linear.[7] Once trained, DNNs can be utilized to greatly accelerate iterative "forward" design processes,[7] wherein the properties of candidate designs (i.e., settings of x) are sequentially evaluated in search of a satisfying design (i.e., one where $f(x) \approx y^*$).

In this work, we investigate a relatively new class of DNN-based models, termed Deep Inverse Models (DIMs), which attempt to learn a model of the form $x = g_{\theta_g}(y)$, so that the model takes as input a desired set of material properties then directly predicts the design needed to achieve those properties. Here, θ_g refers to learnable parameters of the inverse model, which control the relationship between its input and output. However, learning inverse mappings - such as g_{θ_g} here - is challenging because the inverse learning problem is often ill-posed. A problem is said to be ill-posed if it violates any one of the three so-called Hadamard conditions: uniqueness, existence, and smoothness (see[7] for further discussion). Most DIMs have focused on addressing violations of the uniqueness condition, which implies that there are multiple designs that yield similar metamaterial properties. The violation of this condition will cause many conventional supervised learning methods to fail. These challenges led to the recent development of a variety of DIMs that can overcome the non-uniqueness problem, and learn accurate inverse models, where we desire that $f(g_{\theta_g}(y^*)) \approx y^*$. Some examples of recent DIMs are conditional invertible neural networks (cINNs), mixture density networks, the Tandem model, and the neural adjoint. We refer readers to Ren et al. 2022[8] for a review of recent DIMs and a benchmark of their performance on metamaterial problems.

1.1 Contributions of this Work

Although DIMs have shown substantial success for the design of metamaterials,[9] they have typically been applied to (relatively) low-dimensional problems, where the goal has been to predict the spectrum of a metamaterial (e.g., absorption, transmission, reflection) so that typically $y \in \mathbb{R}^N$ where $N \leq 10^3$. Yet, in many real-world metamaterial design problems, our goal is to control the near-field patterns. In this case, the properties of the metamaterial are characterized by 3-dimensional complex-valued tensors, which are substantially higher-dimensional than a spectrum. Furthermore, near-field EM scatterings may have a much more complex dependency on the metasurface design than spectra, potentially making the inverse problem more difficult as well. It is unclear whether, and to what extent, DIMs will be effective in this more challenging scenario. Therefore, in this work, we investigate the effectiveness of modern DIMs for designing a cylindrical metasurface to achieve a desired near-field pattern. To make the problem more tractable, we only consider the total electric field along a 2-dimensional plane. We use numerical simulation to create a large dataset of design and simulation pairs, which we use to train and evaluate the performance of several recent DIMs for their ability to identify designs to achieve a custom desired near-field pattern.

The remainder of this paper is organized as follows. Sec. 2 describes our dataset and the DIMs utilized in our experiments. Sec. 3 presents our experimental results. Lastly Sec. 4 discusses our conclusions.

2. METHODOLOGY

In this section, we outline the methods that we use to support our investigation. Specifically, we describe the process that we used to generate our dataset of design-property pairs, which we used to train and evaluate inverse models. Then, we review the performance metrics that we use to quantitatively evaluate the prediction accuracy of our models. Lastly, we present the two inverse models we consider: the naive approach, and the Tandem.

2.1 The Dataset

Our dataset is derived from a collection of finite-difference time-domain simulations conducted using the open source MEEP software package.[10] We illustrate the simulation environment in Fig. 1, which comprises a 3x3 arrangement of cylindrical pillars. Each of our simulations was identical, aside from changing the nine radii of each of our pillars. For each simulation, the the radius of each pillar was randomly sampled from the interval 75 nm to

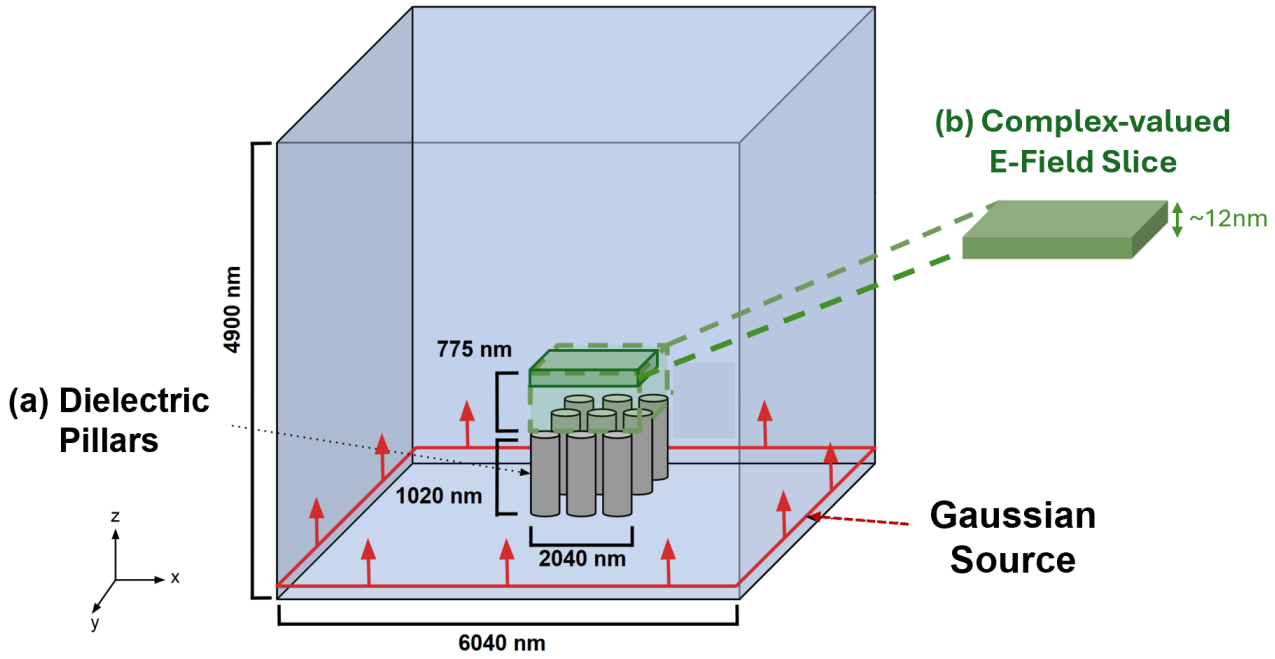

Figure 1. The meta-pillar neighborhood. A Gaussian source with center frequency 1550 nm is incident in $+\hat{z}$ on the 3x3 configuration of meta pillars, with a spatial buffer of 4000 nm in x and y. We characterize the EM properties of the metasurface using a slice of the complex electric field at approximately 775 nm above the top of the metasurface.

250 nm. The pillars are composed of amorphous silicon ($n = 3.48$), they are embedded in polydimethylsiloxane (PDMS, $n = 1.4$), and they reside on a fused silica substrate ($n = 1.44$). Block periodic boundary conditions are applied in the lateral (x, y) directions, while perfectly matched layers are used in z direction. To mitigate unwanted interactions from the wave crossing at the boundary in the $\pm x$ or $\pm y$ directions, a 4 μm buffer is placed around the pillar array, ensuring that scattered fields do not "wrap around" in the periodic cell. The spatial resolution is set to 80 pixels/μm to adequately resolve the meta-atom geometry and wave interactions.

We illuminate the metasurface with a linearly polarized gaussian source (with $\vec{E}$ oriented along the y-axis) that is normally incident in the $+z$ direction and a peak amplitude near 1550 nm. The raw output of the simulation comprises a complex-valued electric field distribution, which we use a DFT to evaluate at $\lambda = 1550$ nm: the wavelength of the peak amplitude of the Gaussian illumination. The raw output data is further processed to encompass a subset of the full simulation volume.

This region of interest extends from the top of the pillars to 775 nm, corresponding to a half wavelength in the z-direction with respect to the Gaussian source's central frequency. We extract a single 2-Dimensional "slice" from each simulation that is approximately 775 nm above the metasurface latic in the z direction, and spans the spatial extent of the 3x3 metasurface array, as illustrated in 1. In total, 1500 unique simulations were performed, which each correspond to a distinct set of nine pillar radii, which comprise the geometric design of each metasurface, x. The target property that we wish to manipulate consists of a 2D complex slice of the electric field, $y \in \mathbb{C}^{166\times166}$.

2.2 Evaluation Metrics

We use two primary performance measures in this work: mean squared error (MSE), and resimulation MSE (reMSE). MSE is widely used for quantitatively evaluating the accuracy of regression models, and is defined as follows:

$$e_{MSE} = \frac{1}{I}\sum_{i=1}^{I}\left|\left|y_i - \hat{y}_i\right|\right|_2^2 \tag{1}$$

Here, it is assumed that we have a collection of I ground truth vectors (or matrices) and associated predictions; y_i refers to the i^{th} ground truth vector (or matrix) and $\hat{y}_i$ refers to a corresponding prediction. The reMSE, can be seen as an adaptation of MSE for the purpose of evaluating inverse models, and has been utilized recent studies on inverse models.[4,8,9] In reMSE we aim to evaluate inverse model predictions by evaluating whether the design predicted by the inverse model yields the desired properties that were originally passed into it as input. Mathematically, reMSE is defined as follows

$$e_{reMSE} = \frac{1}{I}\sum_{i=1}^{I}\left|\left|y_i - f(\hat{x}_i)\right|\right|_2^2 \tag{2}$$

Here, we have a collection of I settings of some target outcome y_i, and associated predictions of $\hat{x}_i$. In an inverse modeling context, we assume that $\hat{x}_i = g_{\theta_g}(y_i)$. Note that Equation 2 assumes that we have access to the true forward model and that we evaluate it for novel settings of x. This is impractical in our case, where our forward model is a numerical simulator. Therefore, we utilize our neural network approximation of the forward model, f_{θ_f} to approximate f in Equation 2, which can result in somewhat optimistic estimates of design accuracy.

2.3 The Forward Model, f_{θ_f}

Although our goal is to evaluate DIMs, some of the DIMs we discuss (e.g., the Tandem) rely upon first having a neural network that has been trained to approximate the underlying forward function, f. As discussed in Sec. 2.2, we also need a fast approximation of the forward model to quantitatively evaluate the accuracy of our DIMs using reMSE. For these reasons, we train a deep neural network, denoted f_{θ_f} to approximate the forward model.

2.4 Inverse Models, g_{θ_g}

Naive Inverse Model (Naive) In some cases, inverse learning problems are relatively well-posed, so that conventional learning machine learning methods work well, or even much better than specialized DIMs.[8] Recent work in Ren et al.[8] suggested that it is useful, as a baseline, to train a conventional DNN-based model where we simply reverse the role of input and output, so that the model is trained to predict x from y using MSE; we adopt this approach and call the resulting model our *Naive* inverse model. We utilize the same architecture as our forward model except that we train the model to predict x using y as input.

The Tandem Model (Tandem) The Tandem architecture consists of two neural networks. One network approximates the underlying forward function, f, while the second network serves as an inverse model. We train a Tandem model in two steps, illustrated in Fig. 2. First, we train a DNN-based forward model, denoted f_{θ_f}, on a collection of input/output pairs (e.g., of the form d in Sec. 1), using a mean-squared error loss, denoted $\mathcal{L}_{MSE}$, and given as follows:

$$\mathcal{L}_{MSE} = \sum_{i=1}^{I}\left|\left|y_i - f_{\theta_f}(x_i)\right|\right|_2^2 \tag{3}$$

After training the forward model, its parameters θ_f are fixed, or "locked", before estimating the inverse model g_{θ_g} by minimizing a reMSE given by:

$$\mathcal{L}_{reMSE} = \sum_{i=1}^{I}\left|\left|f_{\theta_f}(g_{\theta_g}(y_i)) - y_i\right|\right|_2^2 \tag{4}$$

In addition to the loss described in 4, we investigate the use of a boundary loss proposed by Ren et al.[9] This loss was introduced to mitigate a challenge in Tandem, where it frequently converged to solutions outside the training data domain. The authors hypothesized that this occurs because the forward model becomes highly inaccurate outside of the training data domain (for points x), and erroneously predicts that x-values outside of this domain are good solutions. To address this problem, an additional loss term, termed the boundary loss, was included to encourage models to identify solutions within the bounds of the training data (i.e., within the space $\mathcal{X}$). This loss term, $\mathcal{L}_{bnd}$, is formulated as:

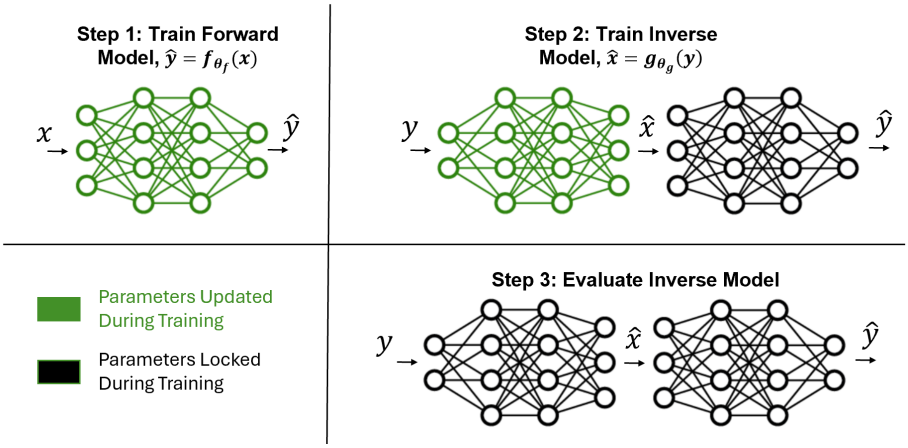

Figure 2. Illustration of the Tandem Model training procedure (Steps 1 and 2), and its evaluation (Step 3). In Step 1, we train a forward model. In Step 2, we train the inverse model using a resimulation loss, which is estimated using the forward model. In this step, we lock the weights of the forward model and only update those of the inverse model.

$$\mathcal{L}_{bnd}(x) = \begin{cases} x - x_{max}, & \text{if } x \geq x_{max} \\ 0, & \text{if } x_{min} \leq x \leq x_{max} \\ x_{min} - x, & \text{if } x \leq x_{min} \end{cases} \tag{5}$$

which is equivalent to $\mathcal{L}_{bnd} = \text{ReLU}\left(|x - \mu_x| - \frac{1}{2}R_x\right)$ for uniform distributions, where μ_x and R_x are the mean and range of the training data, respectively, and ReLU is the rectified-linear-unit commonly used as an activation function in neural networks. The authors demonstrated that including $\mathcal{L}_{bnd}$ in the training of Tandem inverse models also improves their performance. Thus, when training the inverse component of a Tandem model, we use a combination loss from Equations 4 and 5:

$$\mathcal{L}_{inv} = \mathcal{L}_{reMSE} + \gamma \mathcal{L}_{bnd} \tag{6}$$

where the hyperparameter γ is used to balance the relative importance of the two losses in the combination. Once the Tandem is trained, we can extract the inverse component of the model, g_{θ_g} and use it to make inverse predictions.

3. EXPERIMENTS

Both our forward and inverse models consist of Complex-Valued Neural Networks (CVNN). We adopt the implementation in,[11] which allows common complex-valued neural network operations and architectures in PyTorch. Specifically, we use ComplexLinear layers and the ComplexReLU activation, to map our input (as described in Table 1), to the desired output. We also consider adding two ComplexConv2D layers of 16, then 32 filters before our MLPs. These layers use a kernel size of 5 and a stride of 5 and are used to reduce the dimensionality of our electric field tensors (166x166).

For forward models, we simply convert the real-valued radii, $x \in \mathbb{R}^9$, to a complex tensor with null imaginary part, and get a complex output, which represents the real and imaginary parts of the electric field. For inverse models, we build the complex tensor from the real and imaginary channels of the electric field tensor and only preserve the real part of the predicted radii.

For example, the Forward model described in Table 1 consists of two ComplexLinear hidden layers: the first maps the 9-dimensional complex-valued input (the radii) to a 64-dimensional complex representation, and the second maps from 64-d to 512-d complex representation, with ComplexReLU activations being applied in between layers. Finally, a ComplexLinearFinal layer maps the 512-d representation to the output size, producing both real and imaginary parts of the predicted electric field.

Table 1. Details of trained models. Architecture represents the sizes of each layer used, loss represents the loss function used. The Forward model takes as input the radii and returns the last slice of the electric field. All other models do the inverse operation. "FC" refers to fully connected layers, while "Conv" refers to convolutional layers.

Name	**Architecture**	**Loss**	**Input Shape,** $\in \mathbb{C}$	**Output Shape,** $\in \mathbb{C}$
Forward	FC:[64, 512]	MSE	(9)	(166, 166)
Naive	FC: [512, 64]	MSE	(166, 166)	(9)
Tandem	FC:[512, 64]	reMSE	(166, 166)	(9)
Tandem-BD	FC:[512, 64]	reMSE + Boundary	(166, 166)	(9)
Tandem-Conv	Conv:[16, 32], FC:[512, 64]	reMSE + Boundary	(166, 166)	(9)

We partition our available simulation data into a training set with 80% of the data (1200 instances), and the remaining 20% for a validation set (300 instances). We train all of our models on a single NVIDIA-A100 GPU for 100 epochs, using a batch size of 16, an initial learning rate of $1e-3$, which we decay using Cosine Annealing. Our models are optimized using Adam, and we monitor the validation loss to stop the training early if it does not improve after 50 epochs. We also save the model with the lowest validation loss to use for testing.

As outlined in Sec. 2.2 and Table 1, the Forward model and the Naive models compute the mean squared error at each step: measuring the difference between the model's prediction (electric fields in the case of Forward, and radii in the case of Naive) and the ground truth labels. The Tandem uses the reMSE, which passes the predicted inverse solution (i.e., a setting of radii) through the frozen forward model, and computes the MSE between the resimulated electric fields, and the ones input to the inverse model (Equation 4). Tandem-BDY is a variant of the Tandem that adds a boundary loss term to the resimulation loss. The boundary loss, described in 5, ensures that our predictions are confined within the bounds of the training data (for this problem, 75 and 250 nm).

Table 2. Mean Squared Error achieved by each model on the validation set. The error of the Forward model is computed directly between its predictions and labels using MSE; while for Naive, Tandem and Tandem-BD, we use the reMSE.

Name	**MSE/reMSE**
Forward	$1.3e-4$
Naive	$5.5e-2$
Tandem	$5.6e-2$
Tandem-BD	$4.1e-2$
Tandem-Conv	$1.6e-2$

3.1 Results and Discussion

Forward Model. We report the mean squared error, e_{MSE}, of our forward model on the validation set in Table 2. The results indicate that our forward model achieves an MSE of $1.188e-4$, which is competitive with other recent reported results in the literature for models focused on simpler 1D problems (e.g.,[4,5,12]). Fig. 3 presents three instances from our validation set, where we present the ground truth electric field magnitude and phase, respectively, as well as the corresponding predictions of these quantities made by our forward model. The visualizations in Fig. 3 indicate that the forward model makes predictions that are visually very similar to the ground truth, corroborating the quantitative MSE measure.

Inverse Models. We report the resimulation MSE, e_{reMSE}, for our four inverse models - Naive, Tandem, Tandem-BD, and Tandem-Conv - on the validation set in Table 2. The results indicate that the inverse models generally achieve substantially higher error rates than the forward model: roughly two order of magnitude greater for all four inverse models. Recall that we use reMSE to evaluate the inverse models, which takes the designs predicted by the inverse models and passes them through our trained forward model (i.e., $\hat{y} = f_{\theta_f}(g_{\theta_g}(y))$), as discussed in Sec. 2.2). Therefore, the error of the inverse models includes error contributed by the inverse model as well as the forward model. In this case, however, the forward model error is substantially lower than the total error, suggesting that most of the error is being contributed by the inverse models. Fig. 3 presents the

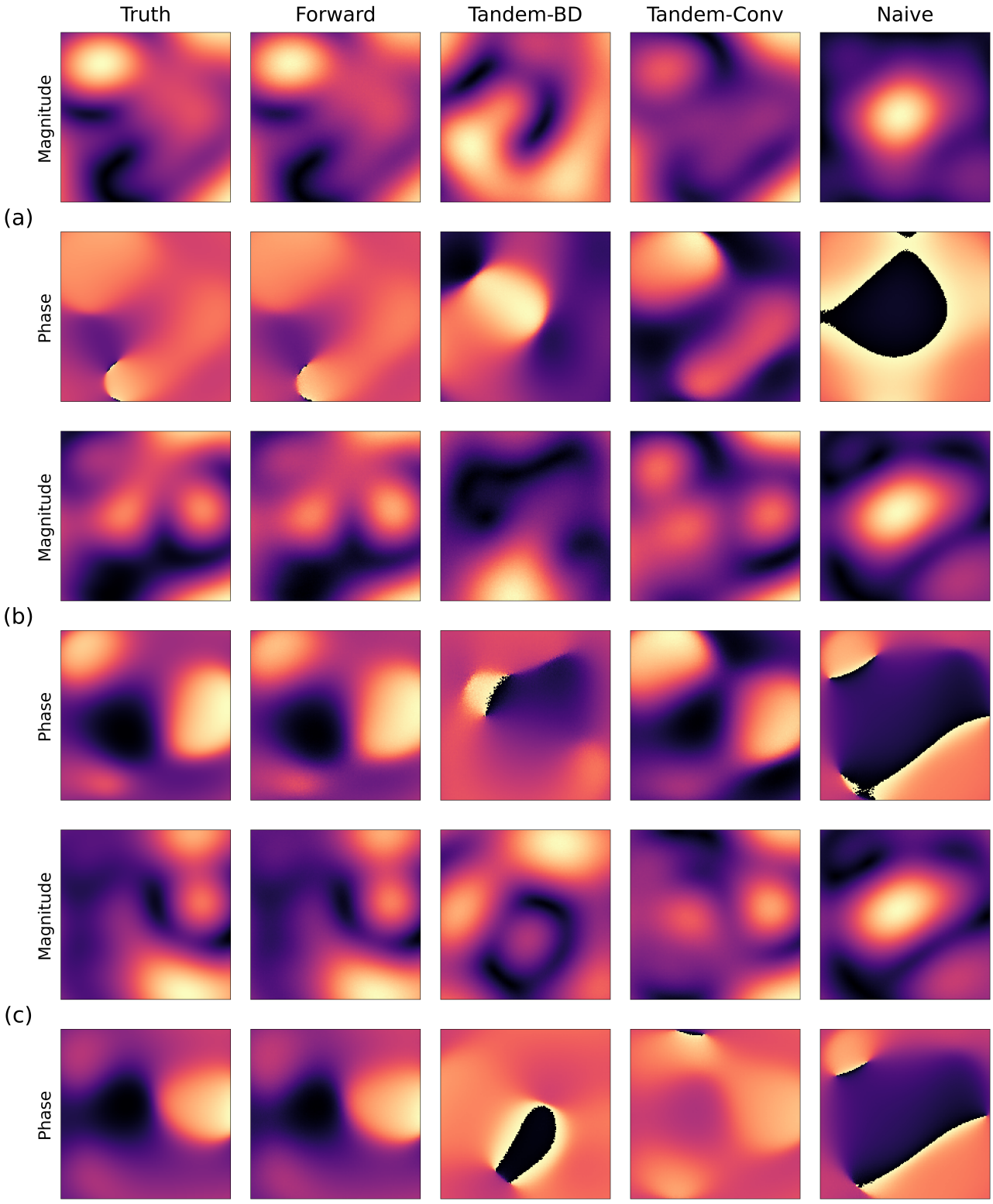

Figure 3. Sample ground truth magnitudes and phases alongside corresponding predictions made by the Forward, Tandem-BD, Tandem-Conv, and Naive models. These examples correspond to those with the lowest MSE for the Tandem-Conv model; they are still relatively representative of Tandem-Conv predictions though because there is little variance in the Tandem-Conv errors, compared to the variance in error rates across the different inverse models, as shown in Fig. 4

magnitudes and phases that result from re-simulating the designs predicted by three inverse models: Tandem-BD, Tandem-Conv, and Naive models. Fig. 3 indicates that there are substantial visual differences between the ground truth fields and the designed field patterns.

The results indicate that the Naive model achieves somewhat lower error than the Tandem model. As found in recent work however,[9] we find that the addition of the boundary loss to Tandem is highly beneficial, resulting here in a 27% reduction in its error, and leading to a 25% accuracy advantage over the Naive inverse model. This performance advantage is apparent in Fig. 3 where we see some instances where the Tandem-BD model is visually much more similar to the ground truth than the Naive model. Because the target data has spatial structure, we also investigate the addition of convolutional layers into the Tandem model, resulting in the Tandem-Conv model (which also includes the boundary loss). We found that the Tandem-Conv performs substantially better, and reduces the error of the Tandem-BD model by approximately 61%. This performance advantage is clearly apparent in Fig. 3 where, in contrast to the other inverse models, Tandem-Conv achieves field patterns that resemble the targeted patterns, albeit with some clear differences.

Is the Problem Ill-Posed? As discussed in Ren et al., 2022,[8] if an inverse problem exhibits a substantial amount of non-uniqueness (i.e., where two or more x exhibit similar output, y) then specialized inverse models (e.g., the Tandem) should exhibit substantial performance advantages over a Naive inverse model. Based upon the relatively similar error of the Naive and Tandem (and Tandem-BD) models, we hypothesize that the inverse problem we study here may exhibit relatively little non-uniqueness. If the problem were ill-posed, we would expect the Tandem models to exhibit a significant performance advantage since they are designed specifically to address that issue. Although the Tandem-Conv model performs substantially better, this may be caused because it has the benefit of convolutional layers, and more parameters, than the Naive model, rather than the problem being ill-posed.

This finding is corroborated by an analysis of the distribution of errors made by the Naive and Tandem-based inverse models, which are presented in Fig. 4. The histograms indicate that the error distributions of the Tandem and Naive model are very similar. If non-unique test instances existed in the dataset, we would expect to see unusually high error for those samples for the Naive model, but we do not find any such instances. Although the Tandem-BD has a clear outlier, this is unlikely to be due to non-uniqueness since the Tandem-BD is specifically designed to mitigate this problem. Based upon the similarity of the Tandem and Naive distributions, as well as the similarity of their quantitative error rates, we hypothesize that this inverse problem exhibits relatively little non-uniqueness. Consequently, we hypothesize that substantial further improvements in inverse accuracy are likely possible through the introduction of generic improvements in the modeling, rather than adding specific modeling adaptations for non-uniqeness.

4. CONCLUSION AND FUTURE WORK

In this work, we investigate the problem of real-time design of electromagnetic (EM) metamaterials to achieve a custom scattering properties: a type of inverse modeling problem. To address this problem we investigate a class of DNN-based models that are specially designed to address inverse problems, termed deep inverse models (DIMs). DIMs have recently shown tremendous promise for solving material design problems, however, relatively less work has been done for high-dimensional problem, such as near-field design. In this work, we performed 1500 simulations of a metasurface with a 3x3 array of meta-atom pillars, where we independently and randomly-varied the radii of each pillar and recorded the resulting electric near-field values. We then used this dataset to train and evaluate several data-driven inverse models, including several variations of a recently-successful DIM, termed the *Tandem*. From our investigation we draw the following main conclusions:

- The Tandem is capable of making relatively accurate design predictions in this challenging high-dimensional setting.
- The Tandem can make predictions in real time, being capable of predicting designs within roughly 4 ms.
- We find evidence that our particular problem may not be ill-posed, and therefore that a similar level of accuracy may be achievable using a conventional supervised learning model rather than a specialized DIM such as the Tandem.

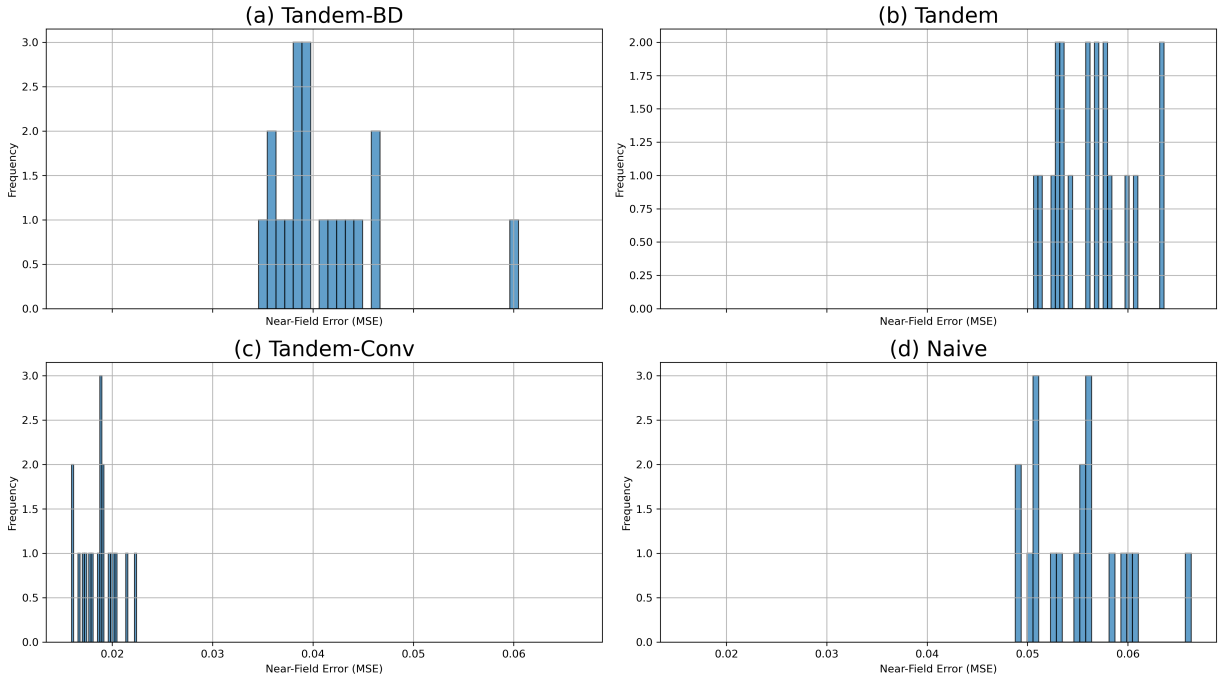

Figure 4. Histograms of inverse resimulation mean squared error (MSE) on the validation dataset for four inverse models: (a) Tandem-BD, (b) Tandem, (c) Tandem-Conv, and (d) Naive.

Future Work We found that the design of the Tandem itself (e.g., its architecture and loss functions) had a significant impact on its accuracy, and we hypothesize that much higher inverse accuracy may be feasible with further investigation into the Tandem's design. We also investigated the ability of the Tandem to identify designs for field patterns that were present in our validation dataset, so that we knew apriori that there existed some design that could achieve the desired input field pattern. In the future, it may be useful to investigate inverse design for arbitrary field patterns, for which it is unknown whether a suitable design exists.

REFERENCES

[1] Smith, D. R., Pendry, J. B., and Wiltshire, M. C., "Metamaterials and negative refractive index," *science* **305**(5685), 788–792 (2004).

[2] Schurig, D., Mock, J. J., Justice, B., Cummer, S. A., Pendry, J. B., Starr, A. F., and Smith, D. R., "Metamaterial electromagnetic cloak at microwave frequencies," *Science* **314**(5801), 977–980 (2006).

[3] Landy, N. I., Sajuyigbe, S., Mock, J. J., Smith, D. R., and Padilla, W. J., "Perfect metamaterial absorber," *Physical review letters* **100**(20), 207402 (2008).

[4] Deng, Y., Ren, S., Fan, K., Malof, J. M., and Padilla, W. J., "Neural-adjoint method for the inverse design of all-dielectric metasurfaces," *Optics Express* **29**, 7526 (Feb. 2021).

[5] Deng, Y., Dong, J., Ren, S., Khatib, O., Soltani, M., Tarokh, V., Padilla, W., and Malof, J., "Benchmarking data-driven surrogate simulators for artificial electromagnetic materials," (2021).

[6] Wiecha, P. R. and Muskens, O. L., "Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures," *Nano letters* **20**(1), 329–338 (2019).

[7] Khatib, O., Ren, S., Malof, J., and Padilla, W. J., "Deep learning the electromagnetic properties of metamaterials—a comprehensive review," *Advanced Functional Materials* , 2101748 (2021).

[8] Ren, S., Mahendra, A., Khatib, O., Deng, Y., Padilla, W. J., and Malof, J. M., "Inverse deep learning methods and benchmarks for artificial electromagnetic material design," *Nanoscale* **14**(10), 3958–3969 (2022).

[9] Ren, S., Padilla, W. J., and Malof, J. M., "Benchmarking deep inverse models over time, and the neural-adjoint method," in [*Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*], Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., eds. (2020).

[10] Oskooi, A., Roundy, D., Ibanescu, M., Bermel, P., Joannopoulos, J., and Johnson, S., "Meep: A flexible free-software package for electromagnetic simulations by the fdtd method," *Computer Physics Communications* **181**, 687–702 (03 2010).

[11] Matthès, M. W., Bromberg, Y., De Rosny, J., and Popoff, S. M., "Learning and avoiding disorder in multimode fibers," *Physical Review X* **11**(2), 021060 (2021).

[12] Nadell, C. C., Huang, B., Malof, J. M., and Padilla, W. J., "Deep learning for accelerated all-dielectric metasurface design," *Optics Express* **27**, 27523 (Sept. 2019).