Benchmarking Deep Learning Architectures for Artificial Electromagnetic Material Problems - Supplementary Materials

Yang Deng*
yang.deng@duke.edu

Juncheng Dong*
juncheng.dong@duke.edu

Simiao Ren*
simiao.ren@duke.edu

Omar Khatib
omar.khatib@duke.edu

Mohammadreza Soltani
mohammadreza.soltani@duke.edu

Vahid Tarokh
vahid.tarokh@duke.edu

Willie J. Padilla
willie.padilla@duke.edu

Jordan M. Malof
jordan.malof@duke.edu

Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

1 Dataset & Code

Our codebase for the MLP, Mixer, and Transformer architectures can be found at the following remote repository: `https://github.com/ydeng-MLM/ML_MM_Benchmark`

1.1 All-dielectric metasurface [2]

This dataset has a total of nearly 60,000 pairs of g, s[1]. The geometry input g has 14 dimensions, and the scattering output s has 2001 dimensions, which are 2001 frequency points from 100-500 THz. The geometry is randomly selected from the grid points defined in table 1.

We generated the 60,000 ADMs datasets on CEMS utilizing CST Microwave Studio distributed computing for two months. The all-dielectric metasurface geometry dataset in our benchmark tasks consists of four SiC elliptical resonators in one supercell as illustrated in Fig.2. The 14 dimensional-geometry parameters make our all-dielectric metasurface rather complex with exotic scattering responses. The difficulty of the problem requires the use of computational electromagnetic simulation (CEMS). We use CST Microwave Studio to carry out the simulations for our all-dielectric metasurface supercells. Each simulation roughly takes 5-10 minutes to compute, and the total data collection time is around two months. To the best of our knowledge, there is currently no ethical problem in our physics simulations.

1.2 Nanophotonics particles [2]

This dataset has a total of 50,000 pairs of g, s. The geometry input g has 8 dimensions, and the scattering output s has 201 dimensions, which are 201 frequency points from 400-800 nm. The multi-layer nanophotonics particle has alternating layers of TiO_2 and silica with a silica core. Each layer's thickness of the nanophotonic particle range from 30 to 70 nm. The geometry parameters

---

*Authors contributed equally

[2] Licensed by CC0 1.0: `https://doi.org/10.7924/r4jm2bv29`

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

for the nanophotonic particle dataset are randomly sampled within the defined geometry boundary, following a uniform distribution.

The nanophotonics particle is the first few geometries that were used to demonstrate the capability of neural networks in accelerating the simulation of AEMs. The original work used the MLP architecture and established the nanophotonics particle problem as one of the most popular MLP-based AEM problems. The complexity of the problem increases as the number of layers increases in the nanophotonics particle. To generate an adequate difficulty level for the neural network, we set the number of layers at eight for our benchmark tasks. We simulate the scattering responses from the nanophotonics geometry parameter using the transfer matrix method included in Peurifoy et al. [2]. We have not identified ethical issues during the data collection process.

Table 1: Grid definition for the 14-dimensional input geometry parameters. h, p, and r are in units of microns. θ is in unit of degrees.

Step	h	p	r_{x_n}/r_{y_n}	θ_n
1	0.49	1	0.1	-45
2	0.4975	1.125	0.1125	-22.5
3	0.45	1.25	0.125	0
4	0.525	1.375	0.1375	22.5
5	0.6	1.5	0.15	45
6	-	-	0.1625	-
7	-	-	0.175	-
8	-	-	0.1875	-
9	-	-	0.25	-

1.3 Color filter [3]

This dataset consists of multiple collections of color filter data. We chose to use the 100,000 pairs of g, s dataset. The geometry input g has 3 dimensions, and the scattering output s has 3 dimensions, which are 3 values defining CIE XYZ color space. The geometry of the color filter consists of three layers: One layer of SiO_2 sandwiched by two Ag layers. The center SiO_2 layers have a thickness range from 0-1000nm, and the two silver layers have thickness range from 0-50nm. The geometry parameters are sampled randomly from a uniform distribution within the geometry boundaries.

The design iteration in the color filter is a lot faster than the ADMs and nanophotonic particles without requiring full-wave simulation or long computational time[3]. Nonetheless, the color filter dataset brings a unique perspective. In Fig 2, the three-dimensional geometry input of the color filter has three-dimensional color space output instead of scattering responses. The color space is derived from the scattering response. Thus, the problem expanded from $s^{'} = f^{'}(g)$ to $c^{'} = \theta(s^{'}) = f^{'}(g)$. Then the model $f^{'}$ in the color filter forward problem needs to cover an extra step of deriving color space from scattering responses. The geometry follows a uniform distribution within the defined geometry space. We adopted the dataset from the data repository provided by the work from [3].

2 Transformers

When Vaswani et al. [4] introduced transformer, it was a brand new structure specifically aiming to replace the previous state-of-the-art sequence to sequence [5] architectures (like long short-term memory network [6] and gated recurrent neural network [7]) in Natural Language Processing (NLP). By introducing the attention mechanism, applying a learned weighted sum of the meaning of each word in the sequence, and a couple of new changes to the original sequenced idea, transformers architectures dominate the field of natural language processing in the past few years [8].

Transformers did not stop at natural language processing. Shortly after Parmar et al. [9] and Ramachandran et al. [10] applied the self-attention mechanism on computer vision (CV) task, which was heavily dominated by convolution neural networks [11, 12, 13, 14] and got similar performance with significantly fewer trainable parameters. Dosovitskiy et al [15] later demonstrated that using

---

[3] Licensed by CC by 4.0 : `http://dx.doi.org/10.5258/SOTON/D1686`

visual transformers (ViT) they achieved state-of-the-art performance in multiple image recognition benchmarks with much fewer parameters.

Apart from NLP and CV, transformers are also being applied to other application fields biological structure [16] and protein prediction [17]. Currently, there exists little exploration using transformer architecture in AEM design and we aim to provide our first attempt at applying vanilla transformer modules to the AEM design task.

Here we use the original architecture of the transformer [4] implemented built-in in the Pytorch framework [18], with as little modification as possible. Below we list some of the discrepencies about the assumption of the original transformer model with our problem and the way we circumvent these problems:

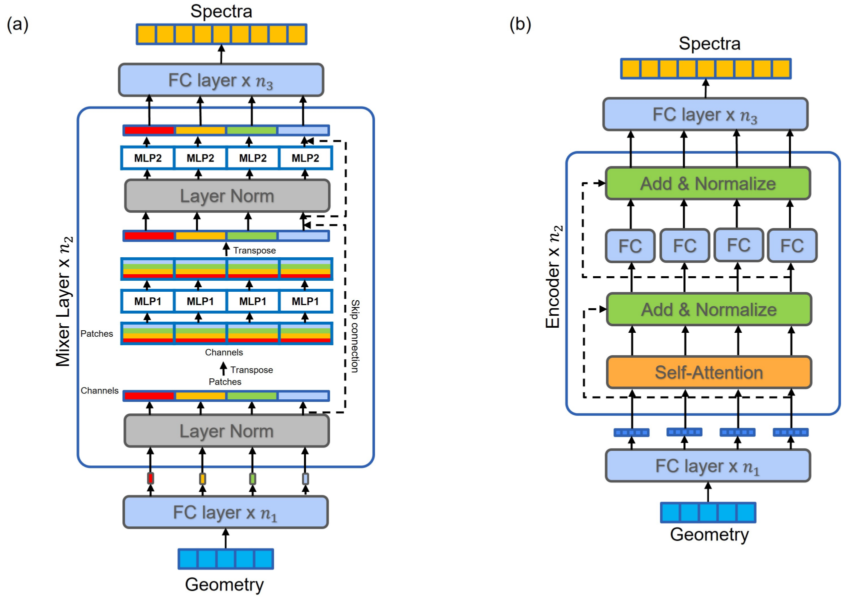

Figure 1: Architecture of the transformer used in the work and the tweaks we made. The input of all our AEM benchmarks are D_g x1 dimension and output are D_S x 1 dimension, where we use fully connected layers to connect with the input and output of the transformer structure. The structure is copied n_2 times and there are n_1 and n_3 layers of fully connected layers, all being hyper-parameters that we grid search on validation set.

Table 2: Adaptation of original (ori.) Transformer/Mixer architecture to AEM modeling, dimension of n and n refers to sentence length and embedding dimension for transformer, number of patches and patch size for Mixer.

Property	Ori. Method	Our problem	Our Adapation
Ordered input	Transformer (Sentence)	No (Geometry)	Delete positional encoding
Sequenced output	Transformer (Sentence)	No (Spectra)	Only take the Encoder (Decode using MLP)
Sequenced input	Transformer/Mixer (dim: $n * d$)	No (dim: $len_d * 1$)	Append tokenizer (Tokenize using MLP)

3 Testset error distribution

We also examined the loss distribution of each problem's testset for all the architecture. To eliminate any possible odd geometry inputs that have unreasonable high loss, we plotted the histogram of our testset loss distribution so that we can visualize any oddity directly as shwon in Fig 2.

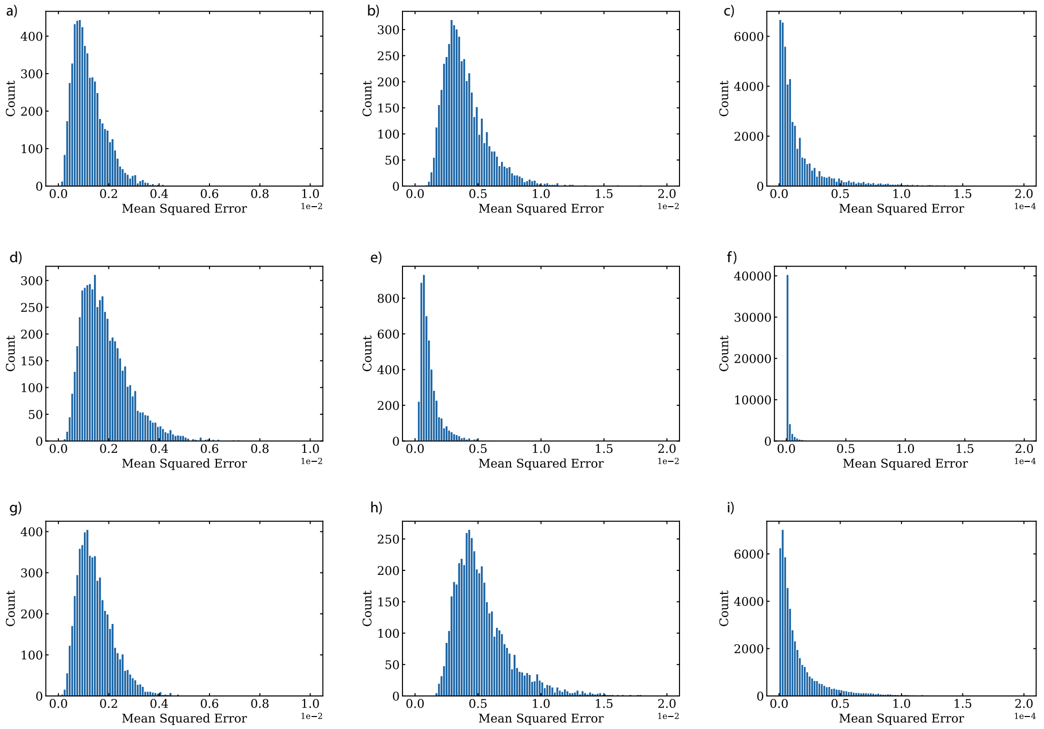

Figure 2: Histograms of the testset loss distribution for each architecture on all-dielectric metasurface dataset (a)-(c), nanophotonics particle dataset (d)-(f), color filter dataset (g)-(i).

4 Baseline model hyperparameters

We build the baseline models based on the hyperparameters provided by the original papers for each dataset [19, 2, 3].

All three datasets implemented the MLP architecture except that the all-dielectric metasurface problem added one-dimensional transpose convolutional and convolutional layers. The baseline model for the ADM problem consists of twelve fully connected hidden layers, each with 1000 neurons, followed by four 1D transpose convolutional layers for upsampling and one convolutional layer for smoothing. The transpose convolutional layers have kernel size [16, 16, 33, 33] and filter size [4, 4, 4, 4]. The final convolutional layer has kernel size one and strides at 1. Leaky ReLU is the activation function, and batch normalization is applied at each hidden layer. The learning rate is set at $1e^{-4}$, and the L2 regularization is set at $1e^{-4}$, and batch size at 1024.

The baseline model consists of four fully connected hidden layers for the nanophotonics particles dataset, each with 250 neurons. We use ReLU as the activation function, and batch normalization is also applied in our base model. The learning rate is set at $1e^{-4}$, and the L2 regularization is set at $1e^{-3}$, and batch size at 1024.

The baseline model consists of seven fully connected hidden layers for the color filter dataset, each with 250 neurons. We use ReLU as the activation function, and batch normalization is also applied in our base model. The learning rate is set at $1e^{-4}$, and the L2 regularization is set at $1e^{-4}$, and batch size at 1024.

5 Main results summary

We also provide a tabular format of the main results here for easy reference.

Table 3: Summary of architectures' testset performance on each dataset

	All-dielectric Metasurface	Nanophotonics particle	Color filter
Baseline	$1.20e^{-3}$	$1.12e^{-2}$	$5.00e^{-5}$
MLP	$\mathbf{1.23e^{-3}}$	$4.04e^{-3}$	$2.22e^{-5}$
Mixer	$1.87e^{-3}$	$\mathbf{1.22e^{-3}}$	$\mathbf{1.86e^{-6}}$
Transformer	$1.47e^{-3}$	$5.47e^{-3}$	$1.76e^{-5}$

6 Non-deep learning models performance

In addition to the three deep learning architectures we benchmarked in our main paper, we also benchmarked three non deep learning models: Linear Regression(LR), Random Forests (RF), Support Vector Regressor(SVR). Due to the difficulty of a comparable optimization (the DL models are optimized using GPU whereas our non-DL models are running on CPUs), we did not do a hyper-parameter search for these non-DL models. We report the default hyper-parameters that we use here:

- Linear Regression: Basic linear regression with an intercept. No penalty on the weights for regularization purposes. For each data point in the spectra, we run one linear regressor taking all the geometry points as input.
- Random Forests: An ensemble of decision tree regressors. The total number of trees is 100. The criteria for making the split is MSE. The maximum depth of a tree is 10. The minimal samples to be split in each split is 2. Bootstrapped samples are used to construct individual trees.
- Support Vector Regression: Used linear kernel for the support vector construction due to running time. The regularization term is kept at default 1. The maximum iteration is 1000 iterations, we did discover due to the large size of our dataset, a lot of times the SVR is not converging.

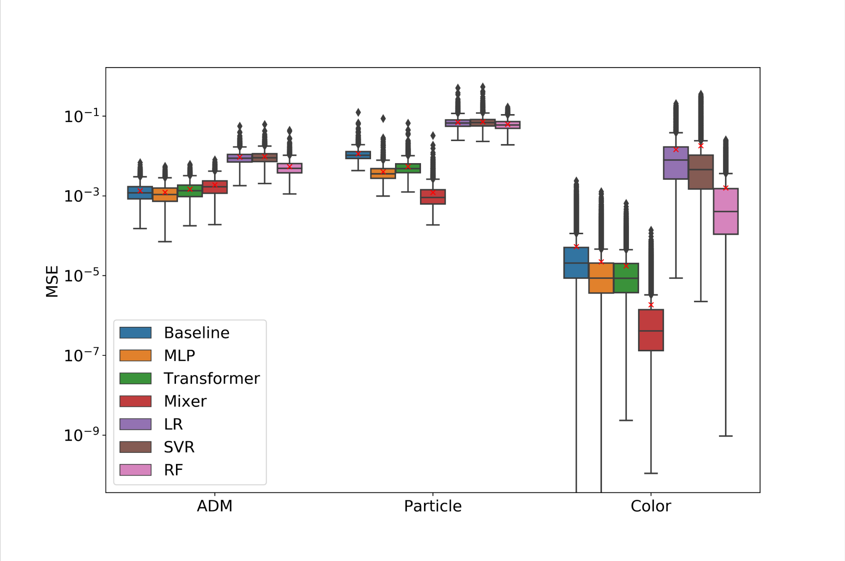

Figure 3: The MSE performance plot including the non deep learning

7 Other performance metrics

In addition to the MSE metrics that were reported in our main manuscript, here we also report the ranking performance of our DL models using Kendall's Tau and Spearman's Rho.

Metrics	MLP	Transformer	Mixer	Dataset
Spearman's Rho	0.94654	0.91829	0.90054	ADM
Kendall's Tau	0.84606	0.78598	0.75707	
Spearman's Rho	0.99909	0.99914	0.99982	Color
Kendall's Tau	0.97714	0.97732	0.99122	
Spearman's Rho	0.97890	0.97180	0.98646	Particle
Kendall's Tau	0.90674	0.88718	0.92896	

We also plot the critical difference plot here. Note that limited by the number of datasets presents here, the critical distance is fairly large. However, the rank can be seen clearly that the deep learning-based models outperform the non-DL ones by large margins.

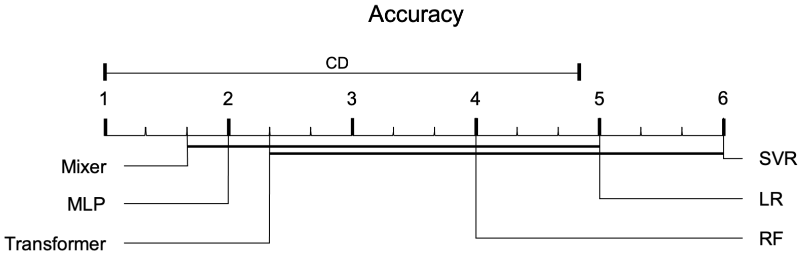

Figure 4: The critical difference plot for benchmarked methods on MSE including the non-DL models

8 Limitations

The optimization for all the datasets and architectures is human-guided grid search. Although an automatic model-based optimization method may provide a less biased (by eliminating the human component) comparison between models, techniques such as Bayesian optimization are not as reliable as human expertise in hyperparameters tuning for varying circumstances. In applying Bayesian optimization to our benchmark tasks, the different setup values can still introduce unfairness into our optimization processes. Therefore, we chose to perform the optimization process with manual searches of the optimal hyperparameters. The manual grid search in our benchmark tasks supplied clear trends indicating the local optima. The manual search also allows us to receive real-time feedback and observations of the architecture behaviors under numerous hyperparameters, giving us intriguing insights into the parallel study of the three architectures of interest. Samples of our optimization process can be found in the supplementary.

References

[1] Y. Deng, S. Ren, K. Fan, J. M. Malof, and W. J. Padilla, "Neural-adjoint method for the inverse design of all-dielectric metasurfaces," *Optics Express*, vol. 29, no. 5, pp. 7526–7534, 2021.

[2] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Science advances*, vol. 4, no. 6, p. eaar4206, 2018.

[3] P. Dai, Y. Wang, Y. Hu, C. de Groot, O. Muskens, H. Duan, and R. Huang, "Accurate inverse design of fabry–perot-cavity-based color filters far beyond srgb via a bidirectional artificial neural network," *Photonics Research*, vol. 9, no. 5, pp. B236–B246, 2021.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[8] “Papers with code, machine translation benchmarks.” `https://paperswithcode.com/task/machine-translation`. Accessed: 2021-08-27.

[9] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, “Image transformer,” in *International Conference on Machine Learning*, pp. 4055–4064, PMLR, 2018.

[10] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, “Stand-alone self-attention in vision models,” *arXiv preprint arXiv:1906.05909*, 2019.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[16] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, *et al.*, “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences,” *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[17] A. Nambiar, M. Heflin, S. Liu, S. Maslov, M. Hopkins, and A. Ritz, “Transforming the language of life: transformer neural networks for protein prediction tasks,” in *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–8, 2020.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[19] Y. Deng, S. Ren, K. Fan, J. M. Malof, and W. J. Padilla, “Neural-adjoint method for the inverse design of all-dielectric metasurfaces,” *Optics Express*, vol. 29, p. 7526, Feb. 2021.