DCU
Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

School of Electronic Engineering

CB54: Machine Learning Algorithms for EM Wave Scattering Problems

Conference Paper

Anthony James McElwee
ID Number: 20211330

August 2023

MEng in Electronic and Computer Engineering

Supervised by Dr Conor Brennan

Declaration

I hereby declare that, except where otherwise indicated, this document is entirely my own work and has not been submitted in whole or in part to any other university.

Signed: Anthony James McElwee   Date: 2023/08/20

Machine Learning Algorithms for EM Wave Scattering Problems

Anthony James McElwee, *MEng Student, DCU*[1]

***Abstract* – This paper details the construction and evaluation of a deep learning emulator, Prescient2DL, to assist a Method of Moments (MoM) iterative solver, SolverEMF2, in generating solutions to two-dimensional electromagnetic scattering problems at 10 MHz. The acceleration of conventional solvers at this frequency is of interest to the medical community where existing methods face computational challenges due to the high-contrast nature of the scenes. Recent works report successes in the area of applying machine learning to electromagnetic scattering problems, however, there lacks documentation as to the potency of combining such models with conventional methods. This paper outlines a statistical experiment to assess the impact of Prescient2DL on SolverEMF2. Experimental evidence indicates a considerably lower initial error than that associated with purely conventional solvers. However, negligible impact on metrics associated with conventional solvers is also reported. Finally, the paper records two simple tests of generalizability where results indicate a degradation in model performance.**

Index Terms* – *computational electromagnetics, deep learning, VEFIE, Transverse Electric, Contrast-Source Integral Equations, U-net, scientific emulation, forward problem, frequency domain

I. Introduction

Development of medical segmentation models to accurately differentiate between benign and malignant biological tissue require the generation of large volumes of frequency-domain electromagnetic scattering simulations. The development of such simulations necessitates considerable learning investment and is considered uneconomical [1]. As extolled in [2], benefits to medical practitioners could arise through the deployment of Magnetic Induction Tomography, however, this requires the acceleration of high-contrast simulations for 10 MHz carrier frequencies. Generally, these simulations operate with a constrained set of input parameters, such as fixed incident source wave configurations. Although input parameters are comparable across simulation incidences, conventional methods still require full wave simulation.

The motivation of this paper is to report on the construction of SolverEMF2 for a toy problem with low-contrast values at 10 MHz. SolverEMF2 uses the Biconjugate Gradient Stabilized Method (BICGSTAB) to calculate solutions of contrast-source integral equations. This high performance code uses circular convolutions to accelerate multiplication steps via Fast Fourier Transforms (FFT). SolverEMF2 is then used to create a data set for developing Prescient2DL. Prescient2DL can feed back into SolverEMF2 to assist in the provision of solutions to the scattering simulations. Experiments to establish the impact of infusing Prescient2DL into SolverEMF2 are provided with commentary.

A. Problem Specification

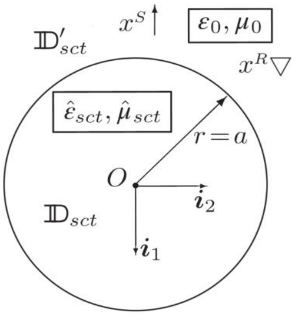

Figure 1. Canonical Problem Diagram

The paper reports on the forward H-polarization vector problem, otherwise known as the Transverse Electric (TE) problem, solving for the electric field strength in a domain with two contrast scatterers, one inside the other. This VEFIE formulation uses the Laplace convention derived in [1] and is solved using a conventional MoM methodology[2]. It is assumed that all wave quantities depend sinusoidally on time with a common angular frequency ω. Since the permeability of biological tissues can be considered roughly equal to that of the background vacuum embedding, no permeability contrast is assumed [2]. The embedding medium has an electromagnetic impedance of $Z_0 = \mu_0 c_0$ and propagation coefficient $\hat{\gamma}_0 = s/c_0$, where μ_0 is the permbeability and c_0 is the wave speed within the embedding. It is assumed that no sources exist within the scatterers. The incident waves are generated by a vertical electric-dipole line source and their components are:

$$\hat{E}_1{}^{inc}(x_T|x_T{}^S) = -\frac{Z_0\hat{M}}{\hat{\gamma}_0}\left(-\hat{\gamma}_0{}^2 + \partial_1\partial_1\right)\hat{G}(x_T - x_T{}^S), \qquad (1)$$

$$\hat{E}_2{}^{inc}(x_T|x_T{}^S) = -\frac{Z_0\hat{M}}{\hat{\gamma}_0}\partial_2\partial_1\hat{G}(x_T - x_T{}^S), \qquad (2)$$

$$Z_0\hat{H}_3{}^{inc}(x_T|x_T{}^S) = \frac{Z_0\hat{M}}{\hat{\gamma}_0}\hat{\gamma}_0\partial_2\hat{G}(x_T - x_T{}^S), \qquad (3)$$

where the 2D Green's function is given by $\hat{G}(x_T) = \frac{1}{2\pi}K_0(\hat{\gamma}_0|x_T|)$. The modified Bessel function of the second kind with second order is denoted by K_0. The electric-dipole moment is denoted by $\hat{M}$. A simplifying assumption is made such that $Z_0\hat{M} = \hat{\gamma}_0$. All other incident components are zero. The contrast is $\hat{\chi}^\varepsilon(x_T) = 1 - \frac{\hat{\varepsilon}_{sct}(x_T)}{\varepsilon_0}$ and the electric contrast source vector is $\hat{w}_k{}^E(x) = \hat{\chi}^\varepsilon(x)\hat{E}_k(x)$ [1]. As the scene assumes that there is invariance in permittivity contrast in the z-direction, the corresponding equation used to provide the basis to solve for the total electric field is as follows:

$$\begin{aligned}\hat{\chi}^\varepsilon\hat{E}_j{}^{inc}(x_T) = \hat{w}_j{}^E(x_T) & \\ & - \hat{\chi}^\varepsilon\big(\hat{\gamma}_0{}^2\delta_{j,k} \\ & - \partial_j\partial_k\big)\int\limits_{x_T{}'\in D_{sct}} \hat{G}(x_T - x_T{}')\hat{w}_k{}^E(x_T{}')\,dA. \quad (4)\end{aligned}$$

An indicator function $\delta_{j,k}$ assumes the property that $\delta_{j,k} = 1$ if j and k are equal, otherwise it is zero. Thus two solutions are required to solve for the electric contrast sources which can then be used to solve for scattered electric field components $\hat{E}_j{}^{sct}(x_T{}^R) = \int_{x_T{}'\in D_{sct}}\left(\hat{\gamma}_0{}^2\delta_{j,k} - \partial_j{}^R\partial_k{}^R\right)\hat{G}(x_T{}^R - x_T{}')\hat{w}_k{}^E(x_T{}')\,dA.$ The permittivity contrasts are frequency independent and assume only a real component with zero conductivity. The diagram, adapted from [1], illustrates the canonical version of the problem. Receivers used to validate the solver against a Bessel-Approach form a ring around the main scatterer lying between the source and the scatterer boundary [4][3].

[1] Date of submission: 2023/08/21. e-mail: anthony.mcelwee2@mail.dcu.ie

[2] This is a more advanced problem than the E-polarization problem described in [3]. The lengthy derivation and explanation of the H-polarization problem are given in Section 3.2.1 of [1].

[3] Validation is illustrated in [4].

II. Existing work in deep learning and electromagnetic scattering problems

As reported in [3] and expanded upon in [5], recent works report successes in the area of applying machine learning to forward electromagnetic scattering problems. Examples are heavily orientated around using physics-informed loss functions [6], [7] or various deep learning architectures that encapsulate elements of the conventional electromagnetic solvers they wish to emulate [8]–[10]. Emulation using deep learning models is almost always the core goal of the works. While other simulation domains have documented hybrid approaches [11]–[13], this aspect of research is lacking in the electromagnetic scattering domain.

This paper benefits from the findings and recommendations from these works focused on emulation to test hypotheses around using deep learning as an assistant to a conventional, integral-method solver. As noted in the introduction, generating solutions to forward electromagnetic scattering problems is a potentially complicated, time-intensive task. In order to generate deep learning training data and to offer state-of-the-art competition to a developed model, this paper builds on the work of [1]. MATLAB code was translated by the student to Python and then adapted to generate solutions in a bid to accelerate the experimental development. The source of the original code emanates from an extremely experienced researcher that is cited recurrently in other references consulted during the investigation of this paper[4].

III. Technical Description

B. Conventional Solver Creation

Equation 4 is defined continuously over the domain, thus giving rise to an infinite number of linear equations with an infinite number of unknown variables for $\hat{w}_j{}^E(x_T)$. The MoM scheme is used to discretize this continuous operator problem. A finite set of basis functions is used to generate a weak form of the continuous operator equation. There exists analytical solutions to the weak formulations of Green Functions but in Equation 4 singularities arise when the position vector and the source vector are equal. Details of the averaging and approximation strategies used to fully express Equation 4 are given in Chapter 1.3 of [1]. From these formulations arises a discrete operator that can be represented using circular convolutions. Through the use of a circulant matrix, the multiplication steps required to solve the MoM linear system of N equations is simplified to a vector of 2*N components. The convolution is computed using FFT which reduces the complexity of the matrix-vector multiplication from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log(N)$.

This MoM discretization, and the exploitation of the circulant properties of the operator functions, leads to the use of Iterative Krylov solvers to find a solution that minimizes a residual error criteria set by the engineer. As noted in [2], and eventually [1], the BICGSTAB solver is favored to solve these MoM electromagnetic scattering problems, with both texts reporting a significant reduction in iterations required to achieve threshold error criteria compared to other Iterative Krylov solvers. Through the use of these mathematical techniques, this high-performance Python code provides a defensible comparison, in terms of inference time, to any deep learning emulator model. The general final deep learning model prediction time is 0.3 seconds with the conventional times for the datasets given in the next section.

C. Dataset Generation Description

Three types of dataset were generated to conduct experimental analysis: major base dataset with two contrast scatterers (DS1); minor single lower-contrast scatterer dataset for testing model generalizability of negative sample cases (DS2); minor single higher-contrast scatterer dataset for testing model generalizability to increased higher-contrast population (DS3). The input parameters for all simulations were the same except for the scatterer contrast values.

The final generation of the scatterer geometry was kept minimal in order to reduce generation time. A variation on the generator used to validate against the Bessel-Approach was adapted to create DS1, DS2 and DS3. In all cases, cells outside the major scatterer area were replaced with the zero contrast value, as illustrated in the figures below. It should be noted that the problem was formulated, via the MoM wrapping, as a discrete-to-discrete approximation of a continuous valued problem, rather than an image-to-image problem. The aim of the model development is to test the impact of using Prescient2DL to assist SolverEMF2, rather than provide emulator generated approximations or visualizations.

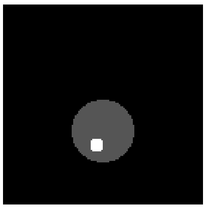

Figure. 2. DS1 geometry sample.

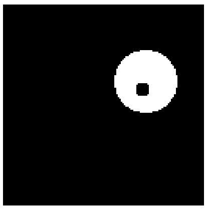

Figure 3. DS2 geometry sample.

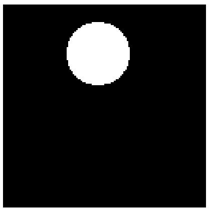

Figure 4. DS3 geometry sample.

DS1 geometry contained one higher-contrast scatterer, $\varepsilon_r = 1.75$ inside a geometrically larger but lower contrast-scatterer $\varepsilon_r = 1.25$. The center point of the lower-contrast scatterer was allowed to be within a distance from the domain origin of its own radius ensuring that it was contained entirely within the domain simulation grid. Both scatterers were of constant fixed size with the smaller scatter populating 5% of the area of the larger scatterer. A seeded random number generator was used to shift the smaller scatterer within a range where at least one cell of higher-contrast scatterer would exist within the boundary of the main scatterer to mimic a positive sample in a biomedical screening scenario.

DS2 has the same geometric rules as DS1 except that the higher-contrast value was set to $\varepsilon_r = 1.0$, thus forming a vacuum void within, or piercing, the larger scatter. This is equivalent to generating negative cases where no secondary tissue exists in the simulation domain.

DS3 has all contrasts set to the higher $\varepsilon_r = 1.75$ value to simulate a total shift in permitivity ε values. This is also a negative sample scenario which tests the model's ability to generalise to larger higher-contrast populations.

With carrier frequency at 10 MHz and the highest permittivity contrast, $\varepsilon_r = 1.75$, the smallest wavelength was 22.7m. The source emmiter is located 170m in the negative x

[4] The final adapted code can be found in [14]. Considerable effort has been made to maintain the original structure of the code as a source of truth so that it can be used more widely in future research efforts, as well as be tied back to the reference text for documentation. Significant gains have been made in the last decade in machine learning due to the open and transparent nature of shared code. The aspiration is that this adaptation can add to development in the electromagnetic scattering domain. The code is fully documented over the breadth of more than one hundred pages in [1] and is the culmination of over 50 articles.

direction. A grid dimension of 128x128 was choosen in order to comply with the FFT requirement that the grid be an integer power of 2, and the typical computer vision approach of using grids divisible by 32. The grid delta was 2m giving rise to a sample per cell of 11. Training a model where the grid dimension is greater than 128 becomes computationally difficult as memory issues arise when the number of layers increases in the deep learning architecture. While references consulted in [3], such as [15], use very small batch sizes to overcome this issue, the dimensions were deemed sufficient to pose as a challenge for the deep learning model.

The material contrast parameters in the medical domain are much more extreme for incident frequencies at 10 MHz [2]. In order to initiate research in the general area, a much lower contrast value was chosen to allow for a large volume of samples to be generated in a shorter time frame.

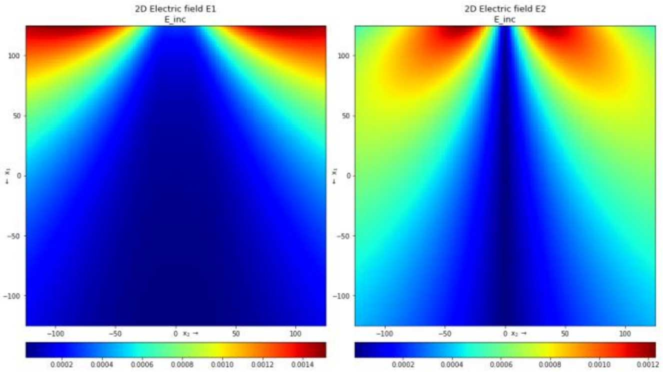

Figure 5. Electric Field incident waves in x and y direction.

All iterative solver computations were carried out on a local laptop CPU i7-11800H @ 2.3 GHz using Python, in particular NumPy and SciPy libraries. DS1 samples took 1 second per sample, while DS2 samples took 0.75 seconds per sample and DS3 samples took 2.2 seconds per sample.

The generated data was saved in NumPy format. All outputs were saved by splitting the real, imaginary and absolute components of the complex fields by channel. The fields were as follows: the real component of the scatterer geometry; the incident E field in the x and y direction; the incident ZH field in the z direction; the two solved scattered E fields in the x and y directions.

In addition, two extra files were saved separately to document the properties of the generated sample: a PNG file illustrating the contained scatter geometry; a separate NumPy file documenting the iteration count, residual error and iteration duration from the iterative solver call-back information. As Equation 4 is solving for the contrast-source, the predicted scattered fields need to be transformed back into the contrast source formulation though the use of the FFT accelerated operator equation at time of inference.

D. Train/Test/Split Approach

The approach taken was to first establish if the model could emulate the conventional solutions. 5000 NumPy arrays were saved in folders of 1000 samples. At the train/validate/test splitting stage, 20% of the 1000 samples were retained for testing (200) and 20% of the training data was retained for validation (160) leaving 640 samples for training.

E. Model Development

Deep learning was carried out on a laptop GeForce RTX 3070 GPU using TensorFlow and Keras Python libraries.

The training inputs were the scatter geometry as a single channel, followed by the real, imaginary and absolute values of the complex incident wave of relevant axis bringing the total input channel count to four. The outputs were the real and imaginary components of the electric field in the axis of interest bringing the output count to two channels. This resulted in a requirement to train two models, one for each axis, in order to provide a full-dimensional, initial guess to the SolverEMF2 workflow. A key step in the pre-processing of the data was that the target arrays were standardized to a range of [0, 1]. This allows the model to train far faster. In order to provide the predictions as inputs in SolverEMF2, the process is reversed at the provision of initial guess time. The information required to perform this standardization can be estimated from a small group of sample solutions making it a robust and simple way to accelerate the training process[5].

1) Model Architecture

The models were fitted using a 54 layer U-net as this type of architecture was core to multiple sources consulted during background research [6], [7], [15]–[17]. 'Elu' activation functions were selected to add non-linearity, as recommended in [15] where experimental evidence was provided to demonstrate that their use gave rise to faster training times in comparison to 'Relu' activation functions. All layers, where possible, had trainable bias terms included in their configurations. Initially a batch normalization layer is applied at the input and each down-sampling block afterwards contains a convolution layer that has stride two. This doubles the channel count and halves the width and height of the input. This is followed by a convolution layer with a single stride to add enhanced complexity to the model. The bottom layer is a bottleneck convolution layer that brings the dimension to 2x2x256 the encoder to the decoder side. On the upscaling decoding side of the U-net, a transposed convolution layer with stride two halves the channel count and doubles the dimensions. Upsampling layers were tried but they led to stronger grid-like lines on the output predictions. Transposed convolution layers also have more trainable parameters which increases the model capacity for handling complexity. This is followed by a convolution layer to increase the model complexity. A seeded Dropout layer with a small parameter value was included after the first stage of the decoder to increase regularisation to the model.

The encoder side is also connected to the decoder side via skip concatenation layers. In order to add non-linearity after the concatenation step, a convolution layer is added on the decoding side for each skip connection. Max Pooling was abandoned due to degradation in initial residual error metrics.

At the penultimate stage of the decoder, two linear convolution layers with kernal size (3, 3) were included in the model. As there is a certain smoothness quality to the predicted fields, these layers are included with the aim of blurring the output, reducing speckling that appeared in earlier predicted plots.

The Adam optimizer was used when fitting the model and the Keras Mean_Squared_Logarithmic_Error function was used as the loss function. The model training times were both roughly 20 minutes which is to be expected as the same model architecture was used in both instances.

The use of augmentations in the training data was avoided since medical applications require pre-designated incident wave directions. Re-orientating the incident fields would not

[5] The description of the process is contained in [4].

increase model generalisability in the path of interest. With the exception of horizontal mirroring, data augmentation would not reduce the required number of generated samples. While it would shrink the possible permutations in the scene configuration space, this would increase the probability of duplicates arising from the symmetry in scenes between the training/test/validation sets over the different folder sessions.

IV. Results

F. Model Training Loss Curves

As depicted in Figure 6, the training loss curve features an extremely abrupt initial decrease followed by a slowly declining plateau. These curves are comparable to loss curves reported in [18], [19]. The loss values are so small that it was required to plot the curves on a logarithmic scale. Both the training and validation curves track each other tightly until the final training session when the training curve starts fluctuating rapidly, albeit with small magnitude. This is an indication that the model has started to overfit the data.

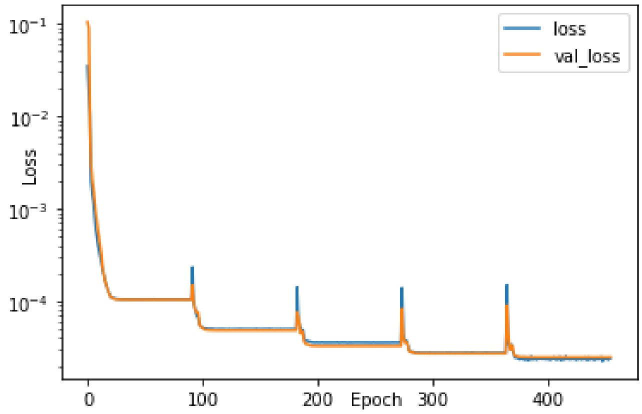

Figure 6. Training history of the E1 model component.

The spikes that occur at the loading of fresh training data indicate there may be possible issues in the model architecture. The difficulty of predicting the y-dimension, E2, is more apparent since the incident wave has greater variation in the y co-ordinate. The E2 component sufferers from greater spikes at each training loading stage than the E1 component.

While it may appear that training could be terminated after the first session, the model is only achieving visually comparable predictions between epoch 300 and 400, as depicted in Figure 6. The final learning rate frequently reduced to the order of magnitude reflected in Table 2.

G. Model Performance

There is a strong visual similarity between the predicted fields and the fully solved fields, as displayed in Figure 7.

Mean Squared Error	E1 Model	E2 Model
Training	2.3042e-05	3.7006e-05
Validation	2.5000e-05	3.7967e-05
Testing	2.6289e-05	4.1606e-05
Final Learning Rate	1.0000e-22	1.0000e-22

Table 1. Final folder model scores and terminal learning rate.

The Mean Squared Error in Table 2 shows that, if treated as an emulator, Prescient2DL achieves strong similarity to the fully solved Iterative Krylov solution. The divergence and rotational components of the fields are being captured by the model and this is evidenced in the plot of the absolute difference in Figure 7.

Table 3 provides descriptive statistics for each test set used to evaluate the impact of Prescient2DL on SolverEMF2. Each set consisted of 100 original samples solved using a naïve initial guess of the incident wave as the scattered field. After training, a second run of SolverEMF2 was used on the same original samples, allowing for direct comparisons across duration of calculation, iteration count and initial error.

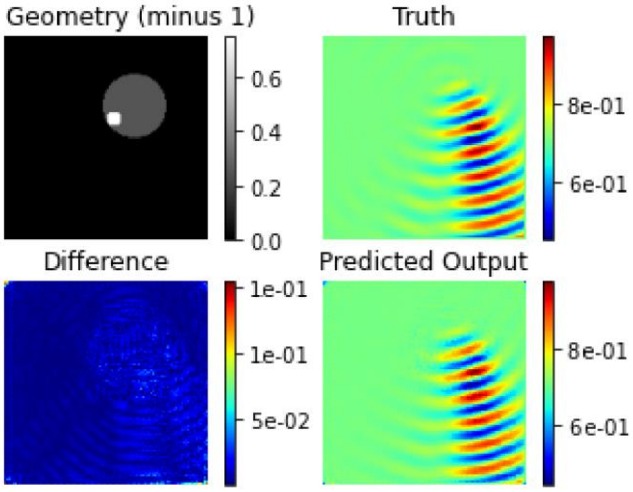

Figure 7. E2 model final prediction on DS1.

Metric	N	Mean	SD
DS1			
Duration_o	100	1.106213	0.055213
Duration_m	100	1.010293	0.093377
Iteration_Count_o	100	22.57	0.655282
Iteration_Count_m	100	22.05	0.479373
Error_Initial_o	100	0.004857	0.002712
Error_Initial_m	100	0.001102	0.000492
DS2			
Duration_o	100	0.772	0.132
Duration_m	100	0.72	0.071
Iteration_Count_o	100	19.57	0.573
Iteration_Count_m	100	19.35	0.52
Error_Initial_o	100	0.003	0.002
Error_Initial_m	100	8.014×10^{-4}	3.702×10^{-4}
DS3			
Duration_o	100	2.218	0.198
Duration_m	100	2.308	0.246
Iteration_Count_o	100	56.65	1.048
Iteration_Count_m	100	56.58	0.955
Error_Initial_o	100	0.03	0.019
Error_Initial_m	100	0.02	0.011

Table 2. Descriptive Statistics of Testing Datasets. "o" corresponds to the naïve guess while "m" gives model-assisted information.

Results indicate that the model generalizes well on DS2. For the generalizability test on DS3, it is evident that the visual resemblance between the predicted fields and the actual fields has broken down, as depicted in Figure 8.

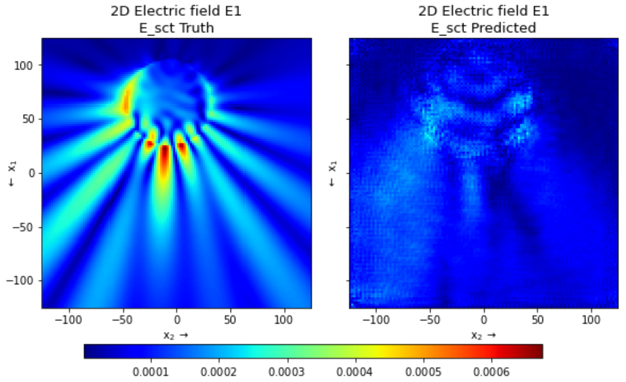

Figure 8. E1 model final prediction on DS3.

Commentary on training issues and a more exhaustive description of the predictive results are available in [20].

V. ANALYSIS[6]

As established in Section IV, Prescient2DL emulates the scenarios of DS1 from both a visual and mean squared error perspective. However, the paper sought to provide experimental data to enable the testing of its impact upon SolverEMF2. Table 3 outlines the statistical results.

	Metric	t	p-values	Mean Difference
DS1	Duration	8.9132305	< .001	0.0959198
DS2		3.394	< .001	0.052
DS3		-3.3	0.001	-0.09
DS1	Iteration Count	7.2478005	< .001	0.52
DS2		4.2	< .001	0.22
DS3		0.572	0.569	0.07
DS1	Error Initial	16.5942404	< .001	0.0037544
DS2		14.493	< .001	0.002
DS3		12.829	< .001	0.01

Table 3. Paired Samples T-Tests comparing naive to model-assisted performance.

H. Research Test 01 – Initial Solution Conveyance t-Test

"Null Hypothesis H_0: The initial error (Residual Norm) in the Krylov Iterative Metrics in SolverEMF2 is the same as for the non-DL assisted conventional solver. Alternative Hypothesis H_A: The initial error (Residual Norm) in the Krylov Iterative Metrics in SolverEMF2 is lower than the non-DL assisted conventional solver" [21].

For all three test sets, Prescient2DL is able to lower the initial error of SolverEMF2 in a statistically significant manner. The null hypothesis is rejected. In DS1, the initial error is lowered to 18% of that provided by the naïve guess. For DS2, Prescient2DL was able to lower the initial error to 27% of the naïve guess showing that there is some generalizability. Although the initial error achieved on DS3 was 66% of the naïve guess, the visual degradation of the guess is apparent in Figure 8.

I. Research Test 02 – Initial Solution Convergence t-Test

"Null Hypothesis H_0: A linear approximation of the slope of the curve for plot Residual Norm versus Iteration Count, labelled as convergence rate, in the Krylov Iterative Metrics for SolverEMF2 is the same as for the non-DL assisted conventional solver. Alternative Hypothesis H_A: A linear approximation of the slope of the curve for plot Residual Norm versus Iteration Count, labelled as convergence rate, in the Krylov Iterative Metrics for SolverEMF2 is the not equal to the non-DL assisted conventional solver" [21].

Again, there is a difference in the mean between the iteration count of naïve and Prescient2DL informed solutions in DS1 and DS2. For DS3, there is no evidence to show a reduction and so the null fails to be rejected. All differences were less than a single iteration on average.

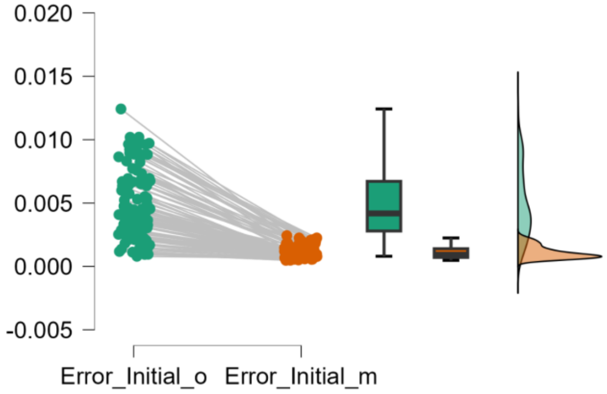

Figure 9. Raincloud of initial error comparison in DS1.

J. Wider Analysis & Impact Requirements

Prescient2DL is able to generalise to DS2 without too much of a degradation in terms of lowering the initial error. DS3 poses larger challenges with much lower relative success compared to DS1 and DS2. Overall, although the initial error in the model-assisted approach is lower and more consistant then the naive approach, as depicted in Figure 9, it is not low enough to impact SolverEMF2. Indeed, the difference in iteration count for DS1is roughly only half of one iteration. The descriptive statistics in Section IV and the statistical analysis indicate that Prescient2DL does not accelerate SolverEMF2 in generating solutions. A subset of the information gathered during a sample run is captured in Table 4. If the current tests and results show low mean squared error but an immaterial impact upon SolverEMF2, how would results that actually achieve MoM solution acceleration manifest in this problem formulation?

Sample 5101	**Naïve Original Run**	**Model-Assisted Run**
Iteration	**Residual Error**	**Residual Error**
0	0.00207481	0.000512025
1	0.000234899	7.02384E-05
2	3.68364E-05	1.58675E-05
3	1.02315E-05	4.2492E-06
4	1.0167E-06	5.663E-07
5	2.997E-07	6.35E-08
6	4.01E-08	1.71E-08

Table 4. Partial Iterative Solver information for Sample 5101

Due to dependance of convergence rates for iterative Krylov solvers on the conditioning of the matrices and the eigenvalue properties of the matrices [22], using Prescient2DL as an initial error lowering tool is the most accessible way to impact SolverEMF2. In this case, Prescient2DL would need to be achieving initial residual errors of 10^{-8} or lower in this toy scenario to diminish the final iteration count by even 25% of the naïve solution iteration count.

[6] A more exhaustive analysis is available in [20].

It may be the case that deep learning is not suitable for achieving this reduced level of error, or that the amount of data required to train a model that can compress and then express the upper-bound estimated 26x10^6 possible permutations in the solution space undermines the gains sought by using deep learning in the first place.

VI. Conclusions

Although a deep learning model has been shown to achieving statistical differences in metric performance compared to naïve approaches, there appears to be nothing materially gained from using the model with regard to accelerating the MoM solver.

Before further experimentation with generalizability and extrapolation is to be carried out, the evidence presented in this paper highlights the need for models to achieve much lower mean squared error results than achieved during this research. In the forward problem, further deep learning model development work should focus on lowering the initial error so that the steps required to achieve iterative solution convergence can be reduced. This paper has also presented evidence that research remains in model development.

If the creation of models that bring the initial residual error to orders of magnitude lower than currently reported is deemed non-viable, after more extensive experimentation, then the attempted application of deep learning in electromagnetic scattering, frequency domain, forward problems could be regarded as frivolous.

In terms of immediate directions for future work arising from this paper, the author has the following suggestions. The use of dense layers [15], residual learning blocks [23] and general adversarial networks [17] may improve training performance for this problem. The inclusion of PINNs or loss functions that make stronger assumptions about the physics underlying the data generating process could help to improve the training behaviour [6]. Combining the information available in each axis to increase the information available to the model may lower the final loss. Co-ordinate systems were siloed for the purposes of this paper and the author expects that inclusion of secondary field information at model inception or at a secondary model stage may increase the final prediction performance. Exploring the domain of deep learning explainability may provide insight into boundary region errors that can be seen in the predictive plots. Deliberately increasing the depth and attempting to overparameterize the model may led to tractability. Theoretical work attempting to increase the mathematical understanding of deep learning models has conjectured that overfitting very complex models does not inevitably lead to poor predictive accuracy [24]. Finally, implementing some of these ideas with the aim of predicting only the absolute component of a field may accelerate the development process.

While positive results are lacking with regard to the goal of the paper, the author points out that the research has been fruitful in providing the electromagnetics community with reproducible data, as well as raising concerns about the model training results. The paper is written with the view that the engineering problem formulation, design approach, data generation process, experimental design and model development are substantial and meaningful in a pre-results context.

References

[1] P. M. van den Berg, *Forward and inverse scattering algorithms based on contrast source integral equations*. Hoboken, NJ: Wiley, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119741602

[2] P. De Tillieux and Y. Goussard, ‘Improving the Computational Cost of Image Reconstruction in Biomedical Magnetic Induction Tomography Using a Volume Integral Equation Approach’, *IEEE Trans. Antennas Propagat.*, vol. 69, no. 1, pp. 366–378, Jan. 2021, doi: 10.1109/TAP.2020.3008618.

[3] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix A: Literature Survey’, DCU, 2023.

[4] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix D: Project Design & Implementation’, DCU, 2023.

[5] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix C: Project Research Log’, DCU, 2023.

[6] 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, doi: 10.1063/5.0071616.

[7] C. Gigli, A. Saba, A. B. Ayoub, and D. Psaltis, ‘Predicting nonlinear optical scattering with physics-driven neural networks’, *APL Photonics*, vol. 8, no. 2, p. 026105, Feb. 2023, doi: 10.1063/5.0119186.

[8] A. P. M. Li, M. Li, and M. Salucci, *Applications of Deep Learning in Electromagnetics: Teaching Maxwell’s Equations to Machines*. Institution of Engineering & Technology, 2023.

[9] S. D. Campbell and D. H. Werner, Eds., *Advances in electromagnetics empowered by artificial intelligence and deep learning*. Hoboken, New Jersey: Wiley-IEEE Press, 2023.

[10] M. Martínez-Ramón, A. Gupta, J. L. Rojo-Álvarez, and C. G. Christodoulou, *Machine learning applications in electromagnetics and antenna array processing*. in Artech House electromagnetics series. Boston London: Artech House, 2021.

[11] R. E. Meethal *et al.*, ‘Finite element method-enhanced neural network for forward and inverse problems’, *Advanced Modeling and Simulation in Engineering Sciences*, vol. 10, no. 1, p. 6, May 2023, doi: 10.1186/s40323-023-00243-1.

[12] A. G. Özbay, A. Hamzehloo, S. Laizet, P. Tzirakis, G. Rizos, and B. Schuller, ‘Poisson CNN: Convolutional neural networks for the solution of the Poisson equation on a Cartesian mesh’, *DCE*, vol. 2, p. e6, 2021, doi: 10.1017/dce.2021.7.

[13] S. Kollmannsberger, D. D’Angella, M. Jokeit, and L. Herrmann, *Deep Learning in Computational Mechanics: an Introductory Course*. in Studies in Computational Intelligence, no. volume 977. Cham: Springer, 2021.

[14] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix F: Source Code Listing’, DCU, 2023.

[15] Q. Ren, Y. Wang, Y. Li, and S. Qi, *Sophisticated Electromagnetic Forward Scattering Solver via Deep Learning*. Singapore: Springer, 2022. doi: 10.1007/978-981-16-6261-4.

[16] C. Brennan and K. McGuinness, ‘Site-specific Deep Learning Path Loss Models based on the Method of Moments’. arXiv, Feb. 02, 2023. doi: 10.48550/arXiv.2302.01052.

[17] Z. Ma, K. Xu, R. Song, C.-F. Wang, and X. Chen, ‘Learning-Based Fast Electromagnetic Scattering Solver Through Generative Adversarial Network’, *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 4, pp. 2194–2208, Apr. 2021, doi: 10.1109/TAP.2020.3026447.

[18] T.-T.-H. Le, H. Kang, and H. Kim, ‘Towards Incompressible Laminar Flow Estimation Based on Interpolated Feature Generation and Deep Learning’, *Sustainability*, vol. 14, no. 19, Art. no. 19, Jan. 2022, doi: 10.3390/su141911996.

[19] Y. Guan, T. Fang, D. Zhang, and C. Jin, ‘Solving Fredholm Integral Equations Using Deep Learning’, *Int. J. Appl. Comput. Math*, vol. 8, no. 2, p. 87, Mar. 2022, doi: 10.1007/s40819-022-01288-3.

[20] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix E: Testing & Results’, DCU, 2023.

[21] A. J. McElwee, ‘Machine Learning Algorithms for EM Wave Scattering Problems, Appendix B: Project Design Plan’, DCU, 2023.

[22] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, 1st ed. Cambridge University Press, 2003. doi: 10.1017/CBO9780511615115.

[23] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Second edition. Beijing [China] ; Sebastopol, CA: O’Reilly Media, Inc, 2019.

[24] P. L. Bartlett, A. Montanari, and A. Rakhlin, ‘Deep learning: a statistical viewpoint’, *Acta Numerica*, vol. 30, pp. 87–201, May 2021, doi: 10.1017/S0962492921000027.