Machine Learning Algorithms for EM Wave Scattering Problems

Anthony James McElwee, *MEng Student, DCU*

***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, H-polarization, electromagnetic scattering problems at the 10 MHz range. The acceleration of conventional solvers at this frequency is of particular interest to the medical community where existing methods face computational disadvantage due to the high contrast nature of the scenes. Recent referenced works report successes in the general area of applying machine learning to electromagnetic scattering problems, however, there is a lack of clarity as to the potency of such models. This paper outlines a statistical experiment to assess the impact of the hybrid methodology, where Prescient2DL contributes to 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 also records two simple tests of generalizability for Prescient2DL where results indicate a degradation in model performance.**

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

I. Introduction

Medical diagnostic tools, such as biological segmentation and classification models, constitute a methodology that can increase the capacity for healthcare practitioners to rapidly and accurately differentiate between benign and malignant biological tissue. Developing such aides requires the generation of large quantities of synthetic data using frequency-domain electromagnetic scattering simulations. The development of such simulations necessitates considerable learning investment [1].

As extolled in [2], significant benefits to patients and medical practitioners could arise through the deployment of Magnetic Induction Tomography (MIT). This requires the acceleration of high-contrast simulation scenarios in the 10 MHz carrier frequency range.

Generally, these simulations operate with a constrained set of input parameters, such as with fixed incident source wave configurations and dielectric material attributes of scatterers. Although input parameters are comparable across simulation incidences, conventional methods require full wave simulation and cannot be estimated with very low or high-frequency approximations. Generating large volumes of such simulations is considered uneconomical.

The motivation of this paper is to report on the construction of a electromagnetic scattering solver workflow, SolverEMF2, for a toy problem with low contrast values for an incident wave frequency of 10 MHz. SolverEMF2 adapts code from [1] to use the Krylov iterative solver Biconjugate Gradient Stabilized Method (BICGSTAB), to calculate the solution to contrast-source integral equations. This high performance code uses circular convolutions to accelerate multiplication steps via Fast Fourier Transforms. SolverEMF2 is used to create a data set for developing a deep learning model called 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

The paper reports on the forward H-polarization 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, receiving a dipole incident wave with E_x, E_y and ZH_z components. This VEFIE formulation uses the Laplace convention derived in [1] and is solved using a conventional MoM methodology[1]. It is assumed that all wave quantities depend sinusoidally on time with a common angular frequency ω. The problem requires finding two electric fields and as a result is a full vector problem. As the task motivation is based around future medical developments, and the permeability of biological tissues can be considered roughly equal that of the background vacuum embedding of the domain, no permeability contrast is assumed [2]. This is a permittivity contrast only problem. 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 componets are:

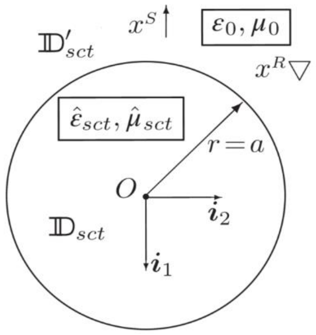

Fig. 1. Canonical Problem Diagram

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

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

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

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.

As the model assumes that there is an invariance in the permittivity contrast in the z-direction, the corresponding equation for the total electric field using contrast source notation as follows:

[1] This is a more advanced problem than the E-polarisation problem described in [3]. The lengthy, full derivation and explanation of the problem are given in section 3.2.1 of [1].

$$\hat{\chi}^{\varepsilon}\hat{E}_j^{\,inc}(x_T) = \hat{w}_j^{\,E}(x_T) - \hat{\chi}^{\varepsilon}\left(\hat{\gamma}_0^{\,2}\delta_{j,k} - \partial_j\partial_k\right)\int_{x_T'\in D_{sct}} \hat{G}(x_T - x_T')\hat{w}_k^{\,E}(x_T')\,dA, \qquad (4)$$

where $\hat{\chi}^{\varepsilon}(x_T) = 1 - \frac{\hat{\varepsilon}_{sct}(x_T)}{\varepsilon_0}$ and the electric contrast source vector is $\hat{w}_k^{\,E}(x) = \hat{\chi}^{\varepsilon}(x)\hat{E}_k(x)$ [1]. 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 the electric field components at the recievers given by $\hat{E}_j^{\,sct}(x_T^{\,R}) = \int_{x_T'\in D_{sct}}(\hat{\gamma}_0^{\,2}\delta_{j,k} - \partial_j^{\,R}\partial_k^{\,R})\hat{G}(x_T^{\,R} - x_T')\hat{w}_k^{\,E}(x_T')\,dA.$ The permittivity contrasts also assume only a real component with zero conductivity and are frequency independent. The diagram, adapted from [1], illustrates the canonical version of the problem. Receivers used to validate the solver against a Bessel Function Approach form a ring around the main scatterer lying between the source and the scatterer boundary [4][2].

II. Existing work in the domain of electromagnetic scattering and machine learning

This is a survey of the state of the art. It should be more than a list of citations of prior work. Give this section a title relevant to your project ("Existing techniques for chronological displacement"). Organize prior work in groups and evaluate them. What are their common features, strengths and weaknesses? This section should be persuading the reader that there is a gaping hole in the research literature, and hint that the technique you are about to describe will fill that hole. The prior art on which you base this section will have already been discussed by you in your Literature Survey. However, you should have greater insight into prior research now, having completed your own project. Do not simply cut and paste text from your literature survey into this section rewrite it so that it is concise enough to meet the length requirements of a research paper and to reflect your improved understanding of your research topic.

End of page 1 (0 pages)

- Look at [2] section A.
- Look at the project log!
- DO THIS WHEN OTHER SECTIONS ARE COMPLETE TO MANAGE SPACE. IF YOU NEED TO BULK OUT THEN DO SO, OTHERWISE NOTE THAT THE PROJECT LOG IN THE APPENDCICES CONTAINS MORE COMMENTRAY AND THE LITERATURE REVIEW REMAINS RELEVANT.
- DENS is useful not just for the meta stuff but because it anchors the use of emulators as substep within larger simulations, they are used as approximating subcomponents [CHECK: is this stated?] whereas this project is trying to go beyond that usage of deep learning.

[2] Validation is illustrated in [4].

III. Technical Description

B. Conventional Solver Creation

As noted in the introduction, generating solutions to forward electromagnetic scattering problems is a potentially complicated, time-intensive task. MATLAB code, provided in conjuction with [1], 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 eminates from an extremely experienced researcher that is cited recurrently in other references consulted during the investigation of the paper[3].

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 gerernate 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 convolution. Through the use of a circulant matrix [CHECK: Tpye up the matrix here if space allowed, see 1.64.], 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 Fast Fourier Transforms (FFT) which reduces the complexity of the matrix-vector multiplication from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log(N))$.

This MoM disctretisation, and the exploitation of the circulant properties of the operator functions, leads to the use of Krylov Iterative solvers to find a solution that minimises a residual error criteria set by the engineer. As noted in [2], and eventually [1], the Biconjugate gradient stabilized method (BICGSTAB) solver is favoured 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 Krylov solvers. Through the use of these mathematical techniques, the Python code is high-performance, thus providing a defensible comparison, in terms of inference time, to any deep learning emulator model. [The general deep learning model predition 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 generalisability of true-negative sample cases (DS2); minor single higher-contrast scatterer dataset for testing model generalisability 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 minimial 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 visualisations.

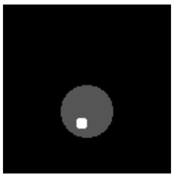

Figure 1. DS1 geometry sample.

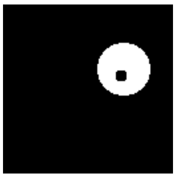

Figure 2. DS2 geometry sample.

Figure 3. DS3 geometry sample.

DS1 geometery contained one higher-contrast scatterer, $\varepsilon_r = 1.75$ inside a geometrically larger lower contrast-scatterer $\varepsilon_r = 1.25$. The centre point of the lower-contrast scatterer was allowed to be within the 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 in order 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 contrast values set to the higher $\varepsilon_r = 1.75$ value to simulate a total shift in permitivity values. This is also a negative sample scenario which tests the ability of the model to generalise to larger higher-contrast populations not seen at training time.

With carrier frequency at 10 MHz and the highest permitivity contrast, $\varepsilon_r = 1.75$, the smallest wavelength was 22.7m. The source emmiter is located 170m in the negative x 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[4].
Although 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 choosen to allow for a large volume of samples to be generated in a shorter time frame.

[3] The final adapted code can be found in [5]. 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 electromagetic 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.

[4] While references consulted in the literature review, such as [6], use very small batch sizes to overcome this issue, the dimensions were deemed sufficent to pose as a challenge for the deep learning model.

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 seconds 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 field 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 seperately 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

In this paper, the approach taken was to first establish if the model could emulate the conventional solutions. The data was 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. The base models were trained on 5000 samples and tested in terms of impact on the SolverEMF2 on three separate test sets, each of 100 samples (DS1, DS2, DS3).

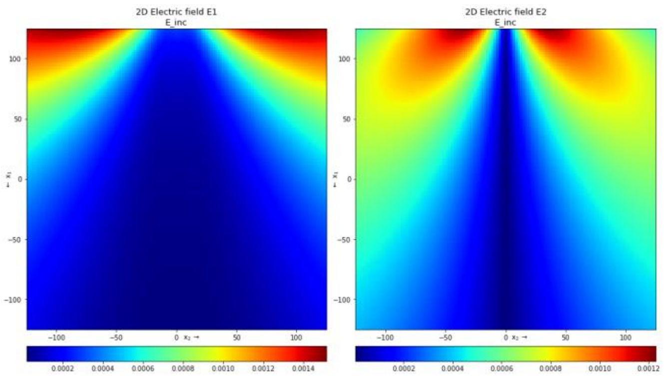

Table 1. Electric Field incident waves in x and y direction.

E. Model Development

All deep learning computations were carried out on a local laptop GeForce RTX 3070 GPU using Tensorflow and Keras Python libraries.

The training inputs were the scatter geometery 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[5]. 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 field, 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 standardised 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 standardisation can be estimated from a small group of sample solutions making it a robust and simple way to accelerate the training process[6].

1) Model Architecture

The models were fitted using a 54 layer U-net type architecture. 'Elu' activation functions were selected to add non-linearity, as recommended in [6] 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 normalisation layer is applied at the input and each downsampling 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 also 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 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 not implemented due to a degradation in the intial 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 the speckling that appears in the predicted versus final solution comparison graphs.

- Maybe expand on the description of Elu if there is space?
- Loss function as an equation

[5] Co-ordinate systems were siloed for the purposes of this paper. The author expects that inclusion of secondary field information at model inception or at a secondary model stage may increase the final prediction performance.

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

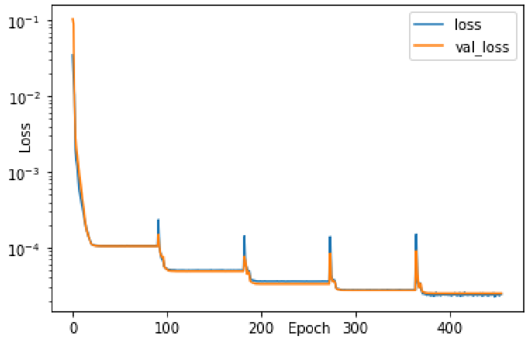

Figure 4. Training history of the E1 component of Prescient2DL.

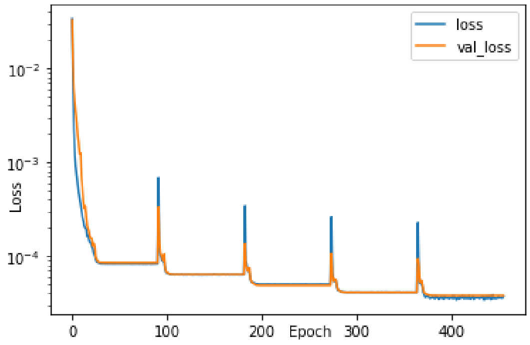

Figure 5. Training history of the E2 component of Prescient2DL.

End of page 3 (2 pages)

IV. Results Obtained[7]

Describe how you know your results are valid. What testing strategies were used? Were enough results obtained? Does your algorithm perform correctly? Does your code implement your algorithm accurately? Does your input data set contain features of the kind the algorithm is supposed to extract?
End of page 4 (1 page)

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

Figure 6. Final folder model scores and terminal learning rate.

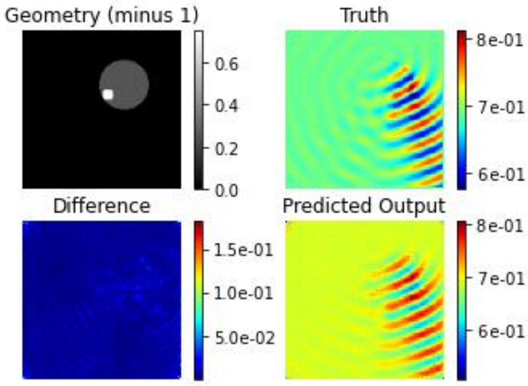

Figure 7. E1 model final prediction on DS1.

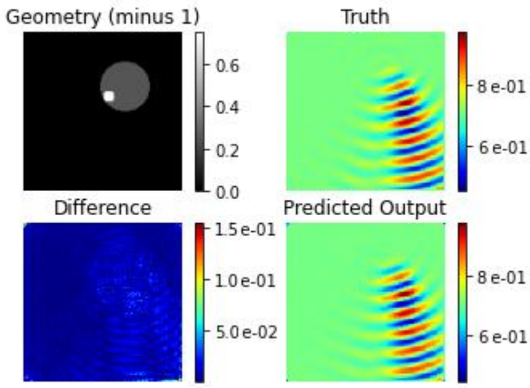

Figure 8. E2 model final prediction on DS1.

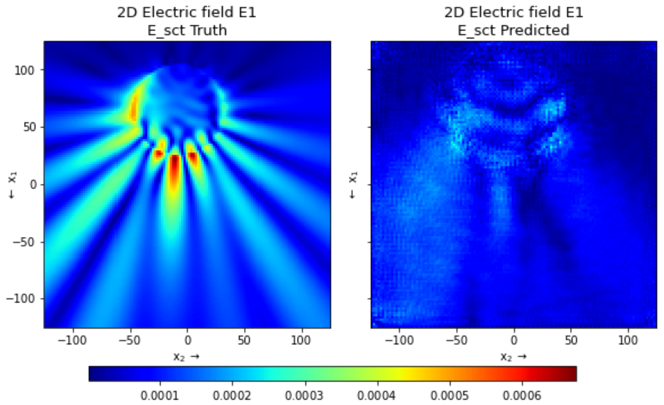

Figure 9. E1 model final prediction on DS3.

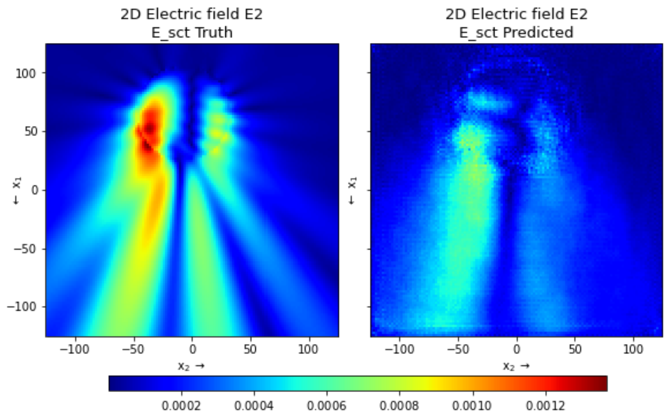

Figure 10. E2 model final prediction on DS3.

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

[7] A more exhaustive description of the results is available in [8].

V. Analysis[8]

Interpret your results here. You've obtained lots of data. What have you learned from it? Does accepting transit traffic overload the router? What do the peaks in the spectral response indicate? Why are there no fluctuations in the EEG data? This section should be ONE PAGE in length. The division of the body of the report into three sections (here named "Technical Description", "Results Obtained" and "Analysis") may be inappropriate for some projects. If you wish to change this structure, you may do so only in consultation with your supervisor and only with his/her written agreement to the revised structure any such revised paper format must have an aggregate length of four pages for the sections equivalent to the above four.

End of page 5 (1 page)

G.Experimental Results

- T-tests for the three metrics from the project plan proposal. There needs to be results for the two architectures tested (final architecture and alternative with max pooling) and there needs to be results for the moved geometry validation set too.
- Wobbley training line in loss curve at the end means the model is starting to overfit

There are two out of distribution tests changing the contrast values. One frops cancer to zero contrast. Slight degradation in performance but visually not too bad. The second ups the main scatter to that of cancer. This takes double the amount of iterations to solve…what is the result? Visually poor but statistically still lower error than naïve guess.

	Metric	t	p	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

Figure 11. Paired Samples T-Tests comparing naive to model-assisted performance.

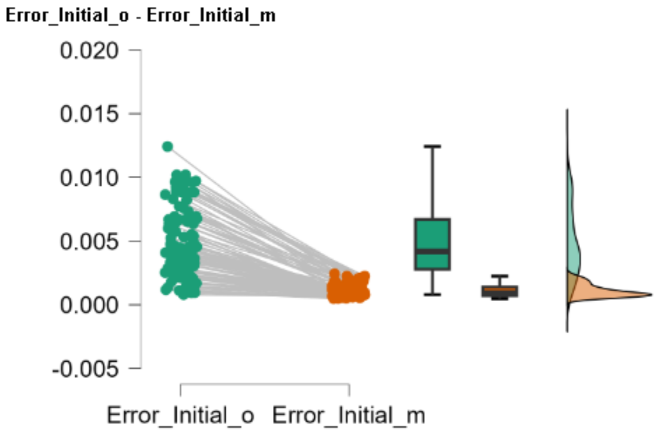

Figure 12. Raincloud plot of initial error comparison. While the initial error in the model-assisted approach is lower and more consistant then the naive approach, it is not low enough to impact SolverEMF2.

[8] A more exhaustive analysis is available in [8].

VI. Conclusions

This is the conclusion. *Here you summarize what has been achieved and learned, and the implications for future research and suggestions for future work that could follow on from your work. This section resembles the introduction in some ways, but remember that by now the reader has read the body of the paper. The introduction was your attempt to encourage them to do so. You can present insights in the conclusions.*

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 2. Iterative Solver information for Sample 5101

- Is this project not a waste of energy and time…no, reference [9]
- Dense layers, as reported in [6], may improve performance since the hybrid infusion relies on deep learning techniques associated with discrete-to-discrete data structures. The architecture deployed in this paper was formulated based on literature that was in the domain of image-to-image data structures.

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, ‘Appendix A: Literature Survey’, Aug. 2023.
[4] A. J. McElwee, ‘Appendix D: Project Design & Implementation’, Aug. 2023.
[5] A. J. McElwee, ‘Appendix F: Source Code Listing’, Aug. 2023.
[6] 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.
[7] 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.
[8] A. J. McElwee, ‘Appendix E: Testing & Results’, Aug. 2023.
[9] F. Chopra, I. Haaland, C. Roth, and A. Stegmann, ‘The Null Result Penalty’, *The Economic Journal*, p. uead060, Aug. 2023, doi: 10.1093/ej/uead060.