Data-Enabled Advancement of Computation in Engineering: A Robust Machine Learning Approach to Accelerating Variational Methods in Electromagnetics and Other Disciplines

Cam Key, *Student Member, IEEE*, and Branislav M. Notaroš, *Fellow, IEEE*

***Abstract*—In this letter, we propose and demonstrate a data-driven machine learning-based approach to accelerate the finite element method, method of moments, finite difference method, and related variational methods while maintaining the attractive properties that have allowed such methods to dominate computational science and engineering fields like computational electromagnetics. We use a neural network to predict a set of macro basis functions for a given problem, using only the solution to an extremely coarse description of the problem as input. We then solve the problem using the predicted macro basis. Unlike some existing methods, ours does not rely on direct prediction of the solution. We show that our macro basis function approach corrects errors in the raw prediction of the network, achieving a far more accurate solution. Results are presented for a class of finite element scattering problems, with error statistics presented from 1000 validation examples and compared to standard and naïve approaches. These results suggest the described macro basis function approach is superior to machine learning approaches that directly predict the solution. Meanwhile our method achieves comparable accuracy to the full solution while requiring only a fraction of the degrees of freedom.**

***Index Terms*—finite element method, method of moments, machine learning, neural networks, macro basis functions, variational methods, computational electromagnetics.**

I. Introduction

VARIATIONAL techniques like finite element method (FEM), method of moments (MoM), and finite difference (FD) method are dominant for solving numerical physics problems in computational electromagnetics (CEM) and computational science/engineering (CSE) due to their flexibility, robustness, and rigorous mathematical underpinnings. The principal shortcoming of these methods is their poor scaling and high computational cost. We introduce a broadly applicable method by which neural networks can be applied to speed up variational methods without sacrificing their desirable characteristics. Rather than predicting solutions to these problems directly, we use neural networks to guess a highly simplified basis on which to solve the problem rigorously using existing techniques.

Previous work seeking to use neural networks to make predictions about the solutions to computational physics or CSE problems has capitalized on the strong predictive power of well-trained neural networks but has not addressed the shortcomings of using such an inherently empirical approach for real-world engineering problems. This has limited the real-world usefulness of such results. Most previous work has focused on predicting quantities derived from a numerical solution given a description of the physical problem, typically material parameters in the computational domain and excitations for the problem [1]–[5]. There has been occasional work that uses a neural network to predict the solution itself, rather than a derived quantity [6]–[7]. In this way, such research has sought to effectively replace variational methods with neural networks as the numerical tool used to solve computational physics problems.

In contrast, some of the biggest breakthroughs and substantial applications of neural networks to perform challenging tasks with the accuracy needed for industry use have used existing, mathematically formal methods guided by the intuitive predictive capability of neural networks to achieve speedup and even improve accuracy [8]–[9]. We believe this is critical to the application of machine learning in most engineering contexts. We have found no existing research that has coupled neural networks with variational methods in a broadly-applicable, robust way. The closest we have found is the use of neural networks to predict bulk material parameters for faster multi-scale FEM simulations in structural mechanics [10]–[15]. We consider this excellent work and in line with the philosophy of using neural networks to guide more-rigorous methods, but unfortunately the method described is specific to structural mechanics problems.

Predicting basis functions directly, rather than trying to predict solutions or derived quantities, we exploit the crucial strength of neural networks: the ability to efficiently and accurately learn low-dimensional representations of complicated, high-dimensional datasets to understand underlying correlations. In the context of variational methods for CSE, this means learning not only the fundamental physical behavior of problems, but also larger emergent trends that define the aggregate behavior of a physical structure under simulation. By using predicted bases to rigorously solve a given problem, we maintain the key strengths of variational

Manuscript first submitted December 8, 2019, resubmitted January 9, 2020. This work was supported in part by the National Science Foundation under grant ECCS-1810492.

Cam Key and Branislav M. Notaroš are with Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: camkey@rams.colostate.edu, notaros@colostate.edu).

methods: rigorous bounding of solution error, accurate error estimation, and well-substantiated methods to improve solution accuracy when solution error is found to be excessive for the given application. These benefits are crucial for any numerical method applied in an engineering context. Meanwhile, the proposed method avoids the downsides traditionally associated with empirical, data-driven predictors like neural networks, namely their black box nature and unpredictability when subject to inputs dissimilar to those used for training.

II. Theory

We consider in general a discretized linear (or linearized) differential or integral equation-based problem with solution S, set of basis functions F, and linear system of form $[A]x = b$, where $f_i \in F$ and x_i denote the i^{th} basis function and associated solution weight, respectively. This system may be Galerkin-weighted but we do not impose this. The weak solution to the problem with N basis functions in this notation is given by

$$\tilde{S} = \sum_{i \in [1..N]} x_i f_i \approx S \tag{1}$$

Construction and solution of the linear system for large problems is computationally time consuming and memory-intensive. With N basis functions, solution of the system has time complexity $O(N^2)$ for iterative methods or $O(N^3)$ for direct methods [16]. Meanwhile, construction of the system, typically dominated by performing the necessary integrations, has complexity $O(N^2)$ for boundary integral methods due to global coupling of the basis functions and $O(N)$ for finite element and finite difference methods due to local coupling.

Convolutional neural networks (CNNs) have seen an explosion in popularity in recent years due to advances in parallel computing power and network architecture that have, together, enabled applicability of CNNs to a broad range of complicated tasks from playing board games [8] to classifying images with record accuracy [17]. For an excellent overview of the theory and concept of modern CNN architectures, see [18]. CNNs take advantage of spatial correlation in data to efficiently learn complicated underlying trends more effectively than classical fully-connected neural networks. If our data have d discrete spatial dimensions, input to a CNN is an array with $d + 1$ dimensions; the extra dimension of the array corresponding to the number of input channels, c. We denote by $n_i, i \in [1..d]$ the size of the input array in the i^{th} spatial dimension. The total number of scalar inputs to a CNN is then

$$N_{inputs} = c * \prod_{i \in [1..d]} n_i \tag{2}$$

The time complexity of evaluating a CNN is $O(N_{inputs})$, a substantial improvement over the $O(N_{inputs}^2)$ complexity of evaluating a fully connected neural network, assuming a typical case where the fully connected network has a similar number of neurons in a hidden layer as the number of inputs [19].

Table I. Asymptotic speedup using CNNs for various cases.

Solver	**MoM**	**FEM**
Direct Solver	$speedup \propto \frac{N}{\gamma^2}$	$speedup \propto \frac{N^2}{\gamma}$
Iterative Solver	$speedup \propto \frac{1}{\gamma^2}$	$speedup \propto \frac{N}{\gamma}$

We propose to use a deep convolutional neural network to predict the solution weights, x, to complicated FEM, MoM, and FD problems given only the solution to a computationally-inexpensive analogue of the problem solved on a reduced basis. Most simply, the reduced basis, $\check{F}$, would constitute a small subset of the complete basis, F. A reduced basis is easy to conceive of for FEM and MoM, especially using higher-order bases. We can simply reduce the number of polynomial basis functions allocated to each element. A reduced basis is less obvious at first for FD, since we typically do not consider the concept of basis functions when working with finite difference techniques, but rather sample points. However, we can consider the sample points used in FD as a weighting of Dirac-delta basis functions centered on the spatial locations of the sample points. In this sense, FD is a special case of FEM given a particular choice of basis and a particular quadrature rule. From this perspective, a reduced basis is easy to conceive of: a coarser grid, the sample points of which are a subset of the original grid.

If a CNN can predict x from $\check{x}$, the solution to the problem discretized using basis $\check{F}$ containing $\check{N} = \gamma N, \gamma \in (0,1]$ basis functions, then the achieved speedup of solving for x is asymptotically proportional to expressions given in Table I.

A difficulty with CNNs and applicability of their results for certain tasks is their black box nature. A trained CNN is a purely empirical model, typically with little theoretical underpinning nor theoretical guarantee on the accuracy of its output. To counteract this to quickly obtain accurate FEM, FD, and MoM solutions, we propose to use x predicted by the network not as the final solution, but rather to generate a set of macro basis functions that can be used to re-solve the problem at comparable accuracy to the approximation using F, but instead using a substantially smaller number of basis functions. We define a macro basis function in general as a linear combination of basis functions from F

$$f_{macro} = \sum_{i \in [1..N]} \alpha_i f_i \tag{3}$$

where α coefficients are specific to a particular macro basis function. We denote the set of macro basis functions F_{macro}.

By this approach, we can guarantee that the solution obtained using the CNN's prediction exactly and rigorously solves a weak formulation of the problem. We also guarantee that the solution satisfies the boundary conditions of the problem by careful choice of the original basis function in F and careful definition of the macro basis functions. We denote by $F_{boundary} \subset F$ the set of basis functions in the original basis that are nonzero wherever a boundary condition is imposed in the original problem. We also denote by $F_{remainder} = F - F_{boundary}$ the remaining basis functions in

the original basis. We then place the additional constraint on any macro basis function that it contains no contribution from basis functions in $F_{boundary}$

$$f_{macro} = \sum_{I_{macro}} \alpha_i f_i \,, \qquad I_{macro} = \{i \in [1..N] \,|\, f_i \notin F_{boundary}\} \tag{4}$$

We then solve the problem with the modified basis $\bar{F} = F_{boundary} \cup F_{macro}$. The macro basis function approach scales as before but with specification that $\gamma = |\bar{F}|/|F|$ where vertical brackets denote set cardinality.

III. Numerical Results and Discussion

We demonstrate here the usefulness of the proposed macro basis function approach for FEM. We randomly generated a dataset of 1000 lossy dielectric slab scattering FEM problems as in [20]. Both slab location and slab material parameters were varied and randomly sampled from a uniform distribution, with slab location varied over 3 wavelengths, slab real relative permittivity varied between 1 and 10, and slab imaginary relative permittivity varied between 0j and -5j. The domain was PML-truncated. F for this test was a set of 6th-order polynomial basis functions as defined in [16]. $\check{F}$ comprised only the linear subset of F, giving $\gamma = 0.33$. A simple feedforward CNN was trained on all 1000 examples to predict x from $\check{x}$. We used a simple four-layer CNN with 3×1 filters and 64 filters per layer. Convolution was performed only in the spatial dimension of the data, with basis functions of different orders encoded as different input or output channels. N_{inputs} for this network was 27, and the network had 162 outputs. For each element, a single macro basis function was constructed as a linear combination of higher-order basis functions with α coefficients equal to predicted complex solution weights in x. To validate, 1000 new problems were generated from the same distribution. For each, the problem was solved using $\bar{F}$ predicted by the network. Fig. 1 compares the solution obtained using $\bar{F}$ to the solution using F and to a 2nd-order solution for a typical example. Note that the 2nd-order basis and $\bar{F}$ have the same number of basis functions.

We see poor agreement between the 2nd-order solution and the 6th-order solution. Meanwhile, despite yielding the same linear system size and structure as the 2nd-order solution, the solution using $\bar{F}$ agrees well with the full 6th-order solution. To further demonstrate the strength of the proposed macro basis function approach, we used the raw output of the neural network (a prediction of the solution weights) to plot a "naïve" predicted solution without re-solving the system. This serves as a benchmark for the somewhat common approach in existing literature to predict a solution directly. Fig. 2 compares this solution with the actual solution and the predicted solution using the proposed method.

Although the naïve predicted solution agrees with the actual solution somewhat better than the 2nd-order solution, we see various inaccuracies in the network's prediction manifest themselves directly in the form of amplitude errors (for instance in the imaginary plot around 2.5 meters) and even substantial errors in solution behavior (around 1.3 meters in the imaginary plot). Because we have taken the solution predicted directly by the neural network at face value as the naïve predicted solution, these errors go uncorrected. Meanwhile, our proposed macro basis function approach compensates for inaccuracies and misconceptions of the network to produce a substantially more accurate solution with the same number of basis functions as the 2nd-order solution.

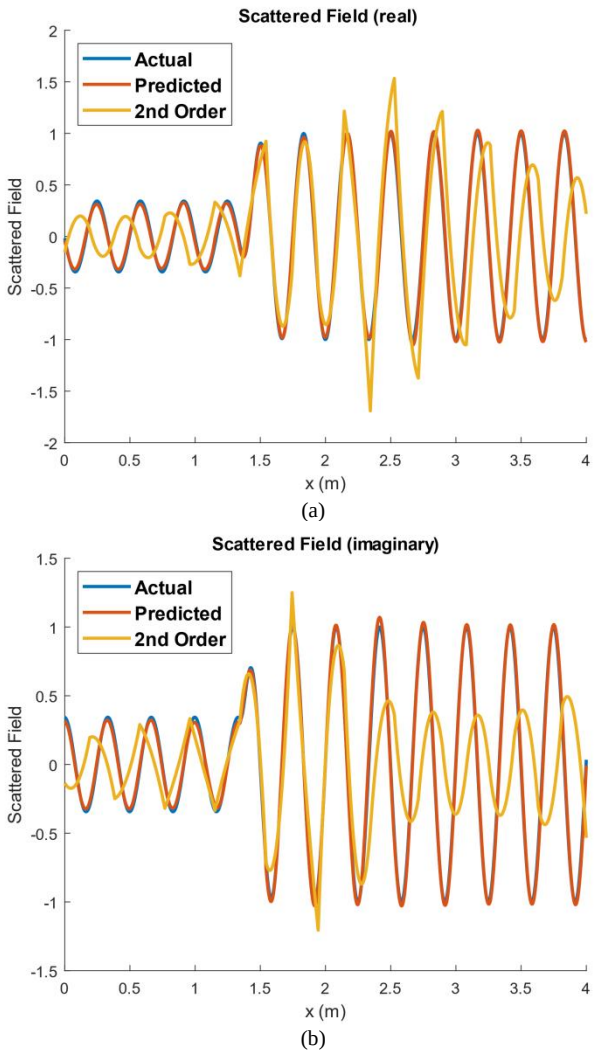

Fig. 1. Scattered field (z-directed) solution comparison between weak solution obtained using predicted macro basis functions, actual solution, and 2nd order weak solution: (a) real component and (b) imaginary component. Predicted solution using the proposed macro basis function approach agrees almost perfectly with the actual solution, despite using only 14% as many basis functions. The 2nd-order solution shown uses the same number of basis functions as the predicted solution but does not agree with the actual solution.

Fig. 3 shows the root mean square (RMS) error with respect to the 6th-order solution for all 1000 validation problems. The validation problem from Figs. 1 and 2 was chosen to fall at the peak of the real predicted RMS error histogram, i.e., an example with typical error. The error at the peak of the histograms for the solutions obtained using the predicted macro basis functions is approximately an order of magnitude less than that for the 2nd order solutions. The proposed method also dominates the naïve predictive approach. In no case does

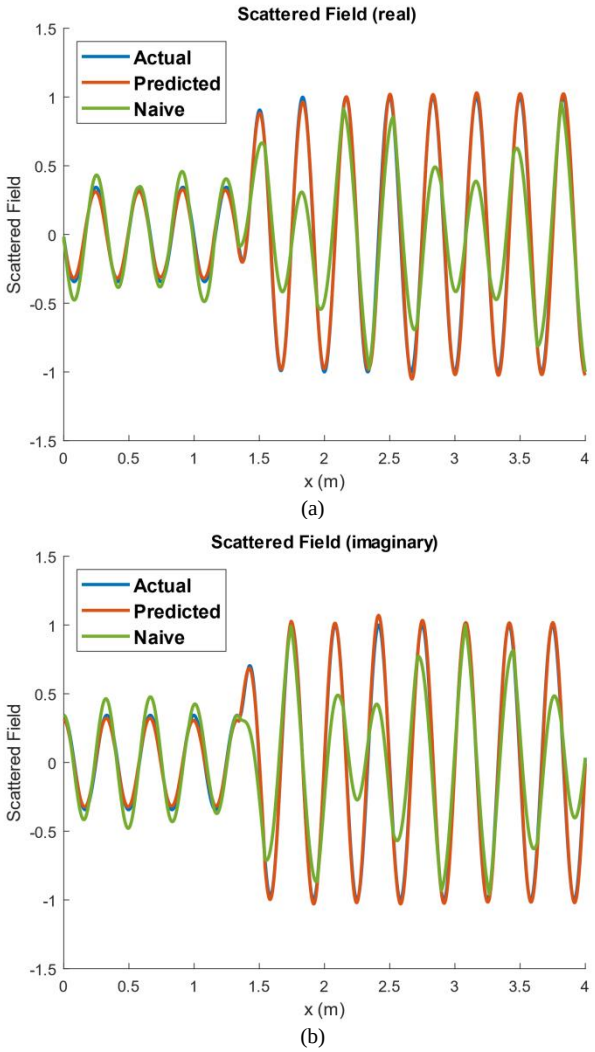

Fig. 2. Scattered field (*z*-directed) solution comparison between weak solution obtained using predicted macro basis functions, actual solution, and naïve predicted solution: (a) real component and (b) imaginary component. Naïve predicted solution is obtained by plotting the solution directly predicted by the network without the macro-basis function approach.

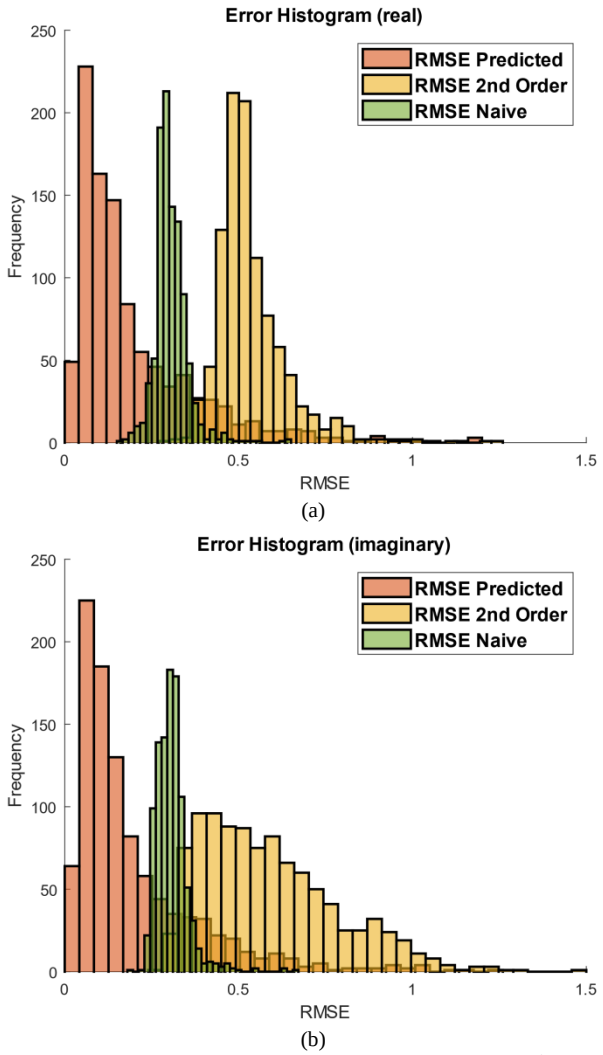

Fig. 3. Real (a) and imaginary (b) RMS error histograms for all 1000 validation problems. Predicted case is for the proposed macro basis function approach. Naïve case gives the error of the solutions directly predicted with the network using no macro basis function approach (the typical, existing approach). 2nd-order case serves as a comparison to the proposed approach. The 2nd-order case and predicted case use the same number of basis functions, but the proposed method yields error an order of magnitude lower.

the naïve approach have error equal to or lower than the peak of the distribution for the proposed method. This demonstrates the potential of the proposed predicted macro basis function approach over both neural network predicted solutions and variational method solutions in isolation.

We also present a direct computation time comparison between the 2nd-order, 6th-order, naïve, and proposed macro basis function methods. Table II gives the time taken by our implementation of each method to solve 1000 randomly generated validation problems. Note that direct time comparisons are highly implementation-dependent, so, although we believe our implementations are efficient, we present Table II with that in mind.

Table II. Direct time comparisons for 1000 random problems.

Method	**2nd Order**	**6th Order**	**Naïve Method**	**Proposed Method**
Time (ms)	41.94	439.7	110.9	152.1

IV. Conclusion

This letter has introduced a robust data-enabled machine learning approach to accelerate CEM and CSE variational methods like FEM, MoM, and FD techniques. Predicting macro basis functions by which a weak formulation can be solved rigorously, the described approach substantially reduces the number of unknowns required to solve a given problem, offering an asymptotic speedup over pure FEM, MoM, or FD solutions while maintaining the rigorousness, accuracy, and broad applicability of these methods. The described method has been demonstrated on a class of FEM problems and rigorously validated on a set of 1000 unseen validation problems. Compared to the naïve approach of predicting the solution directly, our method obtains substantially higher accuracy, its solution typically almost indistinguishable from the true solution. Our method also obtains far higher accuracy than a typical (no predicted macro basis) solution with an equal number of unknowns.

REFERENCES

[1] R. Mishra, "An overview of neural network methods in computational electromagnetics," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 12, no. 1, Dec 2001.

[2] T. Nguyen-Thien and T. Tran-Cong, "Approximation of functions and their derivatives: A neural network implementation with applications," *Applied Mathematical Modelling*, vol. 23, no. 9, pp. 687-704, 1999.

[3] L. Lanzi, C. Bisagni, and S. Ricci, "Neural network systems to reproduce crash behavior of structural components," *Computers & Structures*, vol. 82, no. 1, pp. 93–108, 2004.

[4] J. Ghaboussi, J. H. Garrett, and X. Wu, "Knowledge-based modeling of material behavior with neural networks," *Journal of Engineering Mechanics*, vol. 117, no. 1, pp. 132–153, 1991.

[5] J. D. Martín-Guerrero, M. J. Rupérez-Moreno, F. Martinez-Martínez, D. Lorente-Garrido, A. J. Serrano-López, C. Monserrat, S. Martínez-Sanchis, and M. Martínez-Sober," *IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, 2016, pp. 247-253, 2016.

[6] I. Lagaris, A. Likas, D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987-1000, Sep 1998.

[7] E. Soliman, M. Bakr, N. Nikolova, "Neural networks-method of moments (NN-MoM) for the efficient filling of the coupling matrix," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 6, Jun 2004.

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabisal, ''Mastering the game of go without human knowledge,'' *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[9] R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, S. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, D. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, and A. Senior, "De novo structure prediction with deep-learning based scoring," *Thirteenth Critical Assessment of Techniques for Protein Structure Prediction*, pp. 1-4, Dec 2018.

[10] A. R. Shahani, S. Setayeshi, S. A. Nodamaie, M. A. Asadi, and S. Rezaie, "Prediction of influence parameters on the hot rolling process using finite element method and neural network," *Journal of Materials Processing Technology*, vol. 209, no. 4, pp. 1920–1935, 2009.

[11] X. Wu and J. Ghaboussi, "Neural network-based material modeling," Thesis, University of Illinois, 1991.

[12] W. Ruijter, R. Spallino, L. Warnet, and A. de Boer, "Optimization of composite panels using neural networks and genetic algorithms," *Second MIT Conference on Compurational Fluid and Solid Mechanics*, K. J. Bathe, Ed. Elsevier Science Ltd, pp. 2359–2363, 2003.

[13] A. A. Javadi, T. P. Tan, and M. Zhang. "Neural network for constitutive modelling in finite element analysis", *Computer Assisted Mechanics and Engineering Sciences*, Vol. 10, pp. 523-5299, 2003.

[14] R. I. Levin and N. A. J. Lieven, "Dynamic finite element model updating using neural networks," *Journal of Sound and Vibration*, vol. 210, no. 5, pp. 593–607, 1998.

[15] Y. Hashash, S. Jung, and J. Ghaboussi, "Numerical implementation of a neural network based material model in finite element analysis," *International Journal for numerical methods in engineering*, vol. 59, no. 7, pp. 989–1005, 2004.

[16] B. M. Notaros, "Higher Order Frequency-Domain Computational Electromagnetics," *invited review paper*, Special Issue on Large and Multiscale Computational Electromagnetics, *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, August 2008, pp. 2251-2276.

[17] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85-117, Jan 2015.

[18] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Computer Vision and Pattern Recognition Proceedings*, pp. 770-778, 2016.

[19] K. He and J. Sun, "Convolutional neural networks at constrained time cost," *Computer Vision and Pattern Recognition Proceedings*, pp. 5353-5360, 2015.

[20] C. Key, A. Smull, B. M. Notaroš, D. Estep and T. Butler, "A Posteriori Element-wise Error Quantification for FEM Solvers Using Higher Order Basis Functions," *2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, Boston, MA, 2018, pp. 1319-1320.