A NOVEL HIERARCHICAL MACHINE-LEARNING-BASED METHOD FOR EFFICIENT SOLUTIONS OF ELECTROMAGNETIC SCATTERING PROBLEMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Seçil Eda Doğan

July 2021

A NOVEL HIERARCHICAL MACHINE-LEARNING-BASED METHOD FOR EFFICIENT SOLUTIONS OF ELECTROMAGNETIC SCATTERING PROBLEMS

By Seçil Eda Doğan

July 2021

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Vakur B. Ertürk (Advisor)

Mert Kalfa (Co-Advisor)

Ergin Atalar

Seyit Sencer Koç

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

A NOVEL HIERARCHICAL MACHINE-LEARNING-BASED METHOD FOR EFFICIENT SOLUTIONS OF ELECTROMAGNETIC SCATTERING PROBLEMS

Seçil Eda Doğan
M.S. in Electrical and Electronics Engineering
Advisor: Vakur B. Ertürk
July 2021

In this thesis, we propose a novel hierarchical machine-learning-based method to solve electromagnetic scattering problems from perfectly electrical conductor objects. The proposed method solves the discretized surface integral equation, and uses a tree structure inspired by the multilevel fast multipole algorithm (MLFMA) to provide a hierarchical multilevel scheme with a controllable error. The novelty of the method comes from the evaluation of the matrix-vector-product for the far-zone interactions based on the one-box-buffer scheme. These interactions are computed in a group-wise manner by approximating the box-to-box free space dyadic Green's function for a given geometry with the help of the artificial neural networks. Therefore, once the neural network models to estimate the free-space Green's function are trained for a certain geometry, they become applicable to all arbitrary scatterers with the same electrical size for any excitation. The preliminary results show that the algorithm provides very accurate results and has a comparable complexity with the available state-of-the-art methods, such as MLFMA. Additionally, the proposed method does not exhibit the well-known low-frequency breakdown problem of MLFMA, and its computational efficiency can be greatly improved by parallelization.

Keywords: Integral equations, machine learning.

ÖZET

ELEKTROMANYETİK SAÇILIM PROBLEMLERİNİN VERİMLİ ÇÖZÜMÜ İÇİN ÖZGÜN HİYERARŞİK MAKİNE-ÖĞRENMESİ-TABANLI METOD

Seçil Eda Doğan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Vakur B. Ertürk
Temmuz 2021

Bu tezde, mükemmel elektrik iletkeni nesnelerden elektromanyetik saçılma problemlerini çözmek için yeni bir hiyerarşik makine öğrenimi tabanlı yöntem öneriyoruz. Önerilen yöntem, ayrıklaştırılmış yüzey integrali denklemlerini çözer ve kontrol edilebilir bir hata ile hiyerarşik çok seviyeli bir düzen sağlamak için çok seviyeli hızlı çokkutup yöntemine (ÇSHÇY) benzer bir ağaç yapısı kullanır. Yöntemin özgünlüğü, tek-kutulu-tampon şemasına dayalı uzak bölge etkileşimleri için matris-vektör-çarpımlarının hesaplanmasından gelir. Bu etkileşimler, verilen geometri için grup halinde yapay sinir ağları yardımıyla kutudan kutuya serbest uzay diyadik Green fonksiyonunun yakınsanmasıyla gruplar halinde hesaplanır. Bu nedenle, serbest uzay diyadik Green fonksiyonunu yakınsayacak sinirsel ağ modelleri belirli bir geometri için eğitildikten sonra, aynı elektrik boyutuna sahip tüm saçıcılara herhangi bir uyarım için uygulanabilirler. Ön sonuçlar, algoritmanın isabetli sonuçlar sunduğunu ve ÇSHÇY gibi mevcut en gelişmiş yöntemlerle karşılaştırılabilir bir karmaşıklığa sahip olduğunu göstermektedir. Ek olarak, önerilen yöntem, ÇSHÇY'nın iyi bilinen düşük-frekans bozulması problemini sergilemez ve hesaplama verimliliği paralelleştirme ile iyileştirilebilir.

Anahtar sözcükler: İntegral denklemleri, makine öğrenmesi.

Acknowledgement

Firstly, I would like to express my greatest gratitude to my advisor, Prof. Vakur B. Ertürk, for his guidance and support throughout my graduate studies at Bilkent University. He inspired me with his knowledge and perspectives in my undergraduate years, and he became the reason why I decided to pursue my master's degree in electromagnetics at Bilkent University. He always pushed me to be a better researcher and engineer, starting from my undergraduate studies.

I also would like to thank my co-advisor, Mert Kalfa, for his support and contribution. He helped me find my way out when I got lost, and he simplified the problems related to my research and life.

I also would like to thank the examination committee members, Prof. Ergin Atalar and Prof. Sencer Koç, for showing interest in my studies and accepting to review my thesis.

I want to thank my colleagues and my friends who always believe in me and support me. I also want to thank my mother, Mualla Doğan, my father, Süleyman Doğan, and my sister Selin Seda Timur for their endless support and unconditional love.

I want to thank the Scientific and Technological Research Council of Turkey for supporting my master's degree through the BIDEB scholarship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The scattering problems in electromagnetics (EM) contain an object (scatterer), generally illuminated by a plane wave, in a medium, and we aim to determine the scattered fields (far-fields in general) based on the properties of the medium and the scatterer [1]. One of the most popular methods to solve EM scattering problems are integral-equation-based solvers that mainly include surface integral equations (SIEs) for homogeneous and volume integral equations (VIEs) for inhomogeneous scatterers, respectively [2]. Besides, the high-frequency techniques (such as physical optics (PO), geometrical optics (GO), etc.) under certain assumptions and approximations and the partial-differential-equation-based solvers are also used for certain type scatterers [3]. In typical computational electromagnetics (CEM) practice, the Method of Moments (MoM) is commonly used to convert integral equations (IEs) into a system of linear equations [4, 5]. However, since the memory requirement of MoM is $\mathcal{O}(N^2)$ and its computational complexity is $\mathcal{O}(N^3)$ for direct solution of the resultant matrix equations (and $\mathcal{O}(N^2)$ for each iteration when solved iteratively) for N unknowns, it is not preferable for electrically large problems that contain a large number of unknowns [4, 5]. To enhance the complexity of MoM, Fast Multipole Method (FMM), which is listed as one of the top ten algorithms of the 20th century by the Society of Industrial and Applied Mathematics [6], was introduced to decrease the computational complexity to $\mathcal{O}(N^{3/2})$ while the error is controlled [7, 8]. Then, its multilevel

extension, multilevel fast multipole algorithm (MLFMA), decreases the complexity to $\mathcal{O}(N \log N)$ so that the extremely large electrical problems can be solved efficiently [2, 9].

Machine learning (ML) algorithms have been studied for many years to provide a way of learning a mapping from available data[10, 11, 12]. Nowadays, it is the focus of interest with the increasing amount of available data and the improving computer infrastructures for learning the complex structures to model the complicated problems [10, 12]. It has a wide range of applications such as image classification and/or recognition [13], image super-resolution [14], speech recognition [15, 16], medical diagnosis [17, 18], predictions, recommendations [19], etc. It also has some applications in electromagnetics, i.e., in antenna design (i.e., optimizing the design [20, 21], reducing the coupling [22]), direction-of-arrival estimations [23], scattering problems [24, 25, 26, 27, 28], and inverse scattering problems [29].

In CEM, available algorithms in the literature solve scattering problems with high computational complexity (such as MoM) or provide a solution by making certain assumptions and approximations rather than using the full-wave solutions to improve the computational complexity. Since one of the purposes of ML algorithms is to provide useful approximations and/or future predictions for a given dataset by detecting patterns or regularities [10], estimations obtained from ML techniques can be utilized in CEM to improve the complexity and/or accuracy. MoM is commonly used to provide accurate results with high complexity, and [27] proposes to use artificial neural network (ANN) models to solve the linear system obtained from MoM. MLFMA is one of the commonly used state-of-the-art methods to efficiently solve the scattering problems by utilizing the tree structure and diagonalization. To improve the accuracy of MLFMA, [28] proposes to use a combination of generalized regression neural network and ANN instead of the translation stage of MLFMA. In [25], ANN models are utilized to estimate the errors of the future iterations of the iterative solver, and trimmed-MLFMA (T-MLFMA) is proposed to accelerate the simulations by eliminating some branches of the tree structure strategically.

In this thesis, we propose a novel hierarchical ML-based method, which uses the multilevel tree structure of MLFMA with the controllable error and ML models to approximate box-to-box mapping using the free-space Green's function in its closed-form as a kernel (which may eliminate the limitations due to diagonalization introduced by MLFMA), to solve EM scattering problems accurately and efficiently. The multilevel scheme provides efficient solutions by limiting the complexity of the method and providing an error control scheme. In the tree structure, the boxes are defined as near-zone boxes and far-zone boxes. Calculations for near-zone boxes are completed directly, similar to MLFMA, while the calculations for far-zone boxes are completed in a group-by-group manner with the aid of ML models (fully connected neural network (FCNN) and convolutional neural network (CNN) in this thesis). Each ANN model approximates the free-space dyadic Green's function of the dipoles located at the basis sample points when the observation points are the test sample points. These sample points are defined by dividing the basis and the test cubes (defined in the tree structure), respectively, into smaller cubes to compute the interaction at the far-zone. The neural networks are trained for a given translation direction and a box size beforehand, and the parameters are stored in the memory to be used in the iterative solver. It is significant to note that the parameters are also applicable to all scatterers, which have the far-zone interactions at the level with the box having the same electrical size for any excitation. The proposed method first projects the basis functions (Rao-Wilton-Glisson (RWG) [30] basis functions are used in this thesis) onto dipoles to find the inputs of the neural network based on a far-field mapping, which fits the field pattern of basis functions onto the field pattern of dipoles at the far zone. The error can be controlled at this stage by determining the number of field points at the far-zone and the number of dipoles at the basis box. To use the output of the network, dipoles are projected to basis functions. Note that computational complexity mainly depends on the complexity of the neural network rather than the number of discretization elements used in the problem.

The rest of this thesis is organized as follows: In Chapter 2, preliminaries of electromagnetics starting from Maxwell equations to MoM and MLFMA that are

used to describe and assess the performance metrics of the proposed method, as well as the preliminaries of ML techniques (i.e., basics of ANN and CNN structures and hyperparameters of the neural networks) are explained. In Chapter 3, the proposed novel hierarchical ML-based method to solve EM scattering problems is explained with the implementation details. Chapter 4 presents the preliminary numerical results of the proposed method for the PEC cube and sphere. Chapter 5 concludes the thesis and provides the possible future research directions. In Appendix A, dyadic Green's function that is used in the proposed method is derived. Appendix B contains the tables of the errors of the trained ANN models used in numerical results. Appendix C contains the additional results obtained for the scattering problems of the PEC cube illuminated from different angles. Finally, Appendix D provides the current distributions on the canonical shaped scatterers obtained from different methods.

Throughout this thesis, an $e^{-i\omega t}$ time convention, where $\omega = 2\pi f$ and f is the operating frequency, is used and suppressed. Additionally, $\boldsymbol{a}$, $\boldsymbol{A}$, $\vec{A}$, $\vec{a}$, $\hat{a}$, $\overline{\overline{A}}$ are used to denote a vector, a matrix, a field vector, a position vector, a unit position vector, and a dyad respectively.

Chapter 2

Preliminaries

This chapter aims to provide some introductory information for the novel hierarchical ML-based computational method proposed in this thesis to solve the electric field integral equation (EFIE) for EM scattering problems. Since the proposed method provides the solution for EM problems with the help of ML techniques, the introductory materials will be discussed under two parts. One of them is related to electromagnetic theory, and the other is related to ML algorithms. Regarding the electromagnetic theory, first, a summary of the derivation of EFIE (that uses the free-space Green's function as its kernel) for a simple medium is given starting from Maxwell's equations and making use of the physical equivalence theorem. Then, its solution (i.e., discretization) via MoM and MLFMA is explained briefly. Regarding the ML techniques, the 2-layer ANN structures (FCNN and CNN) and their hyperparameters (i.e., number of epochs, learning rate, etc.) to train the models are explained. The details of the above-mentioned topics can be found in the provided references.

2.1 Electromagnetics Related Preliminaries

This section contains the preliminaries related to electromagnetics, i.e., the basics of the electromagnetic theory, the basics of the physical equivalence theorem, EFIE, MoM and MLFMA.

2.1.1 Basics of Electromagnetic Theory

For a simple (homogeneous, linear, and isotropic) medium, where ϵ and μ correspond to the permittivity and the permeability, respectively, the time-harmonic Maxwell's equations (in differential form) can be written in terms of the field intensities as follows

$$\nabla \times \vec{E}(\vec{r}) = i\omega\mu\vec{H}(\vec{r}) \tag{2.1}$$

$$\nabla \times \vec{H}(\vec{r}) = \vec{J}(\vec{r}) - i\omega\epsilon\vec{E}(\vec{r}) \tag{2.2}$$

$$\nabla \cdot \vec{E}(\vec{r}) = \frac{1}{\epsilon}\rho(\vec{r}) \tag{2.3}$$

$$\nabla \cdot \vec{H}(\vec{r}) = 0. \tag{2.4}$$

In (2.1)–(2.4), $\vec{E}(\vec{r})$ and $\vec{H}(\vec{r})$ are the electric and magnetic field intensities, respectively, $\vec{J}(\vec{r})$ is the electric current density, and $\rho(\vec{r})$ is the electric charge density with $\vec{r}$ representing the position vector. Note that in obtaining (2.1) – (2.4), the constitutive relations, $\vec{D}(\vec{r}) = \epsilon\vec{E}(\vec{r})$ and $\vec{B}(\vec{r}) = \mu\vec{H}(\vec{r})$, where $\vec{D}(\vec{r})$ and $\vec{B}(\vec{r})$ are the electric and magnetic flux densities, respectively, are used. Finally, the relation between the electric current density and the electric charge density is established by the continuity equation, given by

$$\nabla \cdot \vec{J}(\vec{r}) = i\omega\rho(\vec{r}). \tag{2.5}$$

It should be noted that since we are dealing with EM scattering from PEC, only the electric sources are considered (i.e., the fictitious sources are assumed to be zero) [1].

We start deriving the vector wave equation for the electric field, $\vec{E}(\vec{r})$, by first

taking the curl of (2.1) yielding

$$\nabla \times \nabla \times \vec{E}(\vec{r}) = i\omega\mu\nabla \times \vec{H}(\vec{r}). \tag{2.6}$$

Then, combining (2.6) with (2.2) and using the identity $\nabla \times \nabla \times \vec{A} = \nabla(\nabla \cdot \vec{A}) - \nabla^2\vec{A}$, we can obtain

$$\nabla(\nabla \cdot \vec{E}(\vec{r})) - \nabla^2\vec{E}(\vec{r}) = i\omega\mu\left[\vec{J}(\vec{r}) - i\omega\epsilon\vec{E}(\vec{r})\right]. \tag{2.7}$$

Combining (2.7) with (2.3), we can derive the vector wave equation for the electric field as

$$\nabla^2\vec{E}(\vec{r}) + k^2\vec{E}(\vec{r}) = -i\omega\mu\vec{J}(\vec{r}) + \frac{1}{\epsilon}\nabla\rho(\vec{r}), \tag{2.8}$$

where $k = \omega\sqrt{\epsilon\mu}$ is the wave number.

Additionally, a scalar wave equation for scalar potential $\phi(\vec{r})$ and vector wave equation for the vector potential $\vec{A}(\vec{r})$ can be obtained as [1]

$$\nabla^2\phi(\vec{r}) + k^2\phi(\vec{r}) = -\frac{1}{\epsilon}\rho(\vec{r}) \tag{2.9}$$

$$\nabla^2\vec{A}(\vec{r}) + k^2\vec{A}(\vec{r}) = -\mu\vec{J}(\vec{r}). \tag{2.10}$$

Note that in arriving (2.9) and (2.10), we use the Lorentz gauge given by

$$\nabla \cdot \vec{A}(\vec{r}) = i\omega\epsilon\mu\phi(\vec{r}). \tag{2.11}$$

Solutions of (2.9) and (2.10) (i.e., $\phi(\vec{r})$ and $\vec{A}(\vec{r})$) for arbitrary source distributions can be derived with the help of the free-space scalar Green's function, which is given by

$$g(\vec{r}, \vec{r}\,') = \frac{e^{ik|\vec{r}-\vec{r}\,'|}}{4\pi|\vec{r}-\vec{r}\,'|}, \tag{2.12}$$

as

$$\phi(\vec{r}) = \frac{1}{\epsilon}\int g(\vec{r}, \vec{r}\,')\rho(\vec{r}\,')d\vec{r}\,' \tag{2.13}$$

$$\vec{A}(\vec{r}) = \mu\int g(\vec{r}, \vec{r}\,')\vec{J}(\vec{r}\,')d\vec{r}\,'. \tag{2.14}$$

In (2.12)–(2.14), $\vec{r}$ and $\vec{r}\,'$ denote the position vectors of the field and the source points, respectively, and $g(\vec{r}, \vec{r}\,')$ satisfies $(\nabla^2 + k^2)g(\vec{r}, \vec{r}\,') = -\delta(\vec{r}, \vec{r}\,')$. At this point, the electric field can be expressed in terms of $\phi(\vec{r})$ and $\vec{A}(\vec{r})$ as [1]

$$\vec{E}(\vec{r}) = i\omega\vec{A}(\vec{r}) - \nabla\phi(\vec{r}), \tag{2.15}$$

which can be written in terms of $\vec{A}(\vec{r})$ only using the Lorentz gauge, given in (2.11), as

$$\vec{E}(\vec{r}) = i\omega\vec{A}(\vec{r}) - \frac{1}{i\omega\epsilon\mu}\nabla\nabla\cdot\vec{A}(\vec{r}). \tag{2.16}$$

Then, using (2.15) together with (2.13), (2.14), and (2.5), the electric field can be written in terms of the current density as

$$\vec{E}(\vec{r}) = i\omega\mu\int g(\vec{r},\vec{r}')\vec{J}(\vec{r}')d\vec{r}' - \frac{1}{i\omega\epsilon}\int \nabla g(\vec{r},\vec{r}')\nabla'\cdot\vec{J}(\vec{r}')d\vec{r}' \tag{2.17}$$

$$\vec{E}(\vec{r}) = i\omega\mu\int\left(\vec{J}(\vec{r}') + \frac{1}{k^2}\nabla'\cdot\vec{J}(\vec{r}')\nabla\right)g(\vec{r},\vec{r}')d\vec{r}'. \tag{2.18}$$

For scattering problems, we need the electric field expression in the far-zone of the scatterer, where $|\vec{r}| = r >> r' = |\vec{r}'|$ and $r >> \lambda$, where λ is the wavelength. Then, $g(\vec{r},\vec{r}')$ and $\nabla g(\vec{r},\vec{r}')$ can be approximated for the far-zone region as

$$g(\vec{r},\vec{r}') = \frac{e^{ik|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \approx \frac{e^{ikr}}{4\pi r}e^{ik\hat{r}\cdot\vec{r}'}, \tag{2.19}$$

and

$$\nabla g(\vec{r},\vec{r}') = \frac{(\vec{r}-\vec{r}')(ik|\vec{r}-\vec{r}'|-1)e^{ik|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|^3} \approx \hat{r}ik\frac{e^{ikr}}{4\pi r}e^{ik\hat{r}\cdot\vec{r}'}. \tag{2.20}$$

Combining (2.18) with (2.19) and (2.20), we can obtain the electric field for the far-zone region as [2]

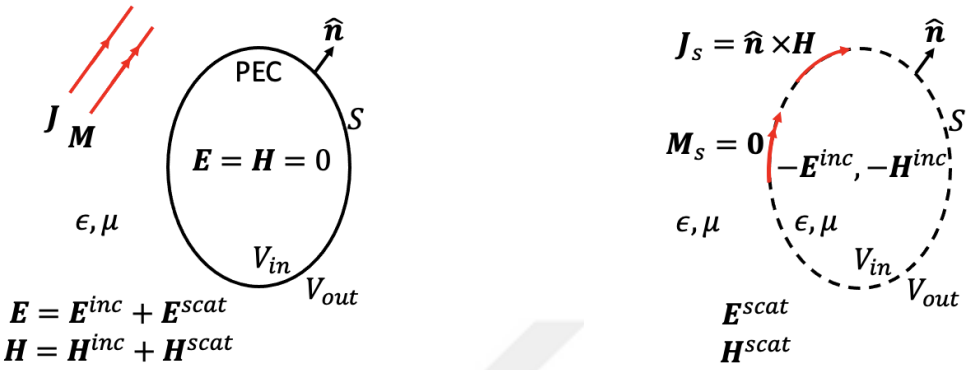
$$\vec{E}(\vec{r}) \approx i\omega\mu\frac{e^{ikr}}{4\pi r}\int\left(\vec{J}(\vec{r}')e^{ik\hat{r}\cdot\vec{r}'} + \hat{r}\hat{r}\cdot\vec{J}(\vec{r}')e^{ik\hat{r}\cdot\vec{r}'}\right)d\vec{r}'. \tag{2.21}$$

(2.21) shows that the distance of the observation point (r) can be separated from the rest of the variables. It also shows that the magnitude of the field decays to zero as the observation point goes to the infinity, i.e., $r \to \infty$, because of $1/r$ dependency. Since the normalized field quantities are the main focus for the far-zone region, far-zone electric field intensity can be written as

$$\vec{E}^{\infty}(\theta,\phi) = \lim_{r\to\infty}\left\{re^{-ikr}\vec{E}(r,\theta,\phi)\right\}. \tag{2.22}$$

2.1.2 Physical Equivalence Theorem and EFIE

Scattering from homogeneous objects can be formulated using the surface equivalence theorem, which states that the actual sources within a region can be replaced



(a) (b)

Figure 2.1: Physical Equivalence: (a) Original problem consisting of a PEC object illuminated by the sources $\vec{J}(\vec{r})$ and $\vec{M}(\vec{r})$. $\vec{J}(\vec{r})$ and $\vec{M}(\vec{r})$ produce $(\vec{E}^{inc}(\vec{r}), \vec{H}^{inc}(\vec{r}))$ in the absence of the PEC object and $(\vec{E}(\vec{r}), \vec{H}(\vec{r}))$ in the presence of the object. (b) Equivalent problem (in the absence of the actual sources) without the PEC object, where the induced equivalent currents radiate in a homogeneous space and create $(\vec{E}^{scat}(\vec{r}), \vec{H}^{scat}(\vec{r}))$ outside the object and $(-\vec{E}^{inc}(\vec{r}), -\vec{H}^{inc}(\vec{r}))$ inside the object.

by the equivalent sources [1]. Based on the uniqueness theorem [1], if the actual sources are taken inside an imaginary bounded region, the fields in a source-free region can be specified by the tangential components of the electric and magnetic fields on the surface bounding the region.

In this thesis, we use the physical equivalence theorem (which is related to the surface equivalence theorem) to derive EFIE that is used to solve the scattered electromagnetic fields from a PEC object. Let $\vec{J}(\vec{r})$ and $\vec{M}(\vec{r})$ (fictitious magnetic current density) create the incident fields, $\vec{E}^{inc}(\vec{r})$ and $\vec{H}^{inc}(\vec{r})$, in a homogeneous region specified by ϵ and μ (in the absence of the PEC object). The original problem is defined by introducing a PEC object inside this region, which results in the scattered fields, $\vec{E}^{scat}(\vec{r})$ and $\vec{H}^{scat}(\vec{r})$, outside the object, and null fields inside the object as illustrated in Fig. 2.1 (a). In order to find the scattered fields, an equivalent problem, shown in Fig. 2.1 (b), is introduced where the PEC object inside the region as well as the original sources are replaced by appropriate equivalent surface currents $\vec{J}_s(\vec{r})$ and $\vec{M}_s(\vec{r})$ on the surface of the PEC object,

S, which satisfies

$$\vec{J}_s(\vec{r}) = \hat{n} \times \left[\vec{H}^{inc}(\vec{r}) + \vec{H}^{scat}(\vec{r})\right] = \hat{n} \times \left[\vec{H}^{scat}(\vec{r}) - (-\vec{H}^{inc}(\vec{r}))\right] \tag{2.23}$$

$$\vec{M}_s(\vec{r}) = -\hat{n} \times \left[\vec{E}^{inc}(\vec{r}) + \vec{E}^{scat}(\vec{r})\right] = -\hat{n} \times \left[\vec{E}^{scat}(\vec{r}) - (-\vec{E}^{inc}(\vec{r}))\right] = 0. \tag{2.24}$$

In (2.23) and (2.24), $\hat{n}$ is the unit outward surface normal of the PEC surface.

(2.24) implies that $\vec{J}_s(\vec{r})$ creates the scattered fields outside and negative of the incident fields inside the eliminated PEC object. Together with the original sources, which generate the incident fields in the absence of the PEC object, we have the total fields, $\vec{E}(\vec{r}) = \vec{E}^{scat}(\vec{r}) + \vec{E}^{inc}(\vec{r})$ and $\vec{H}(\vec{r}) = \vec{H}^{scat}(\vec{r}) + \vec{H}^{inc}(\vec{r})$, outside the PEC object and zero fields inside the PEC object (same as the original problem).

At this point, eliminating the PEC surface and introducing the induced surface current reduce the original problem into a simple radiation problem. However, the induced surface current still depends on both known incident fields and unknown scattered fields. To overcome this issue, we use (2.24) to derive EFIE, which only needs the incident fields to be known on the surface of the PEC surface. Rewriting (2.24) as

$$-\hat{n} \times \vec{E}^{inc}(\vec{r}) = \hat{n} \times \vec{E}^{scat}(\vec{r}), \tag{2.25}$$

where $\vec{E}^{scat}$ is obtained from the induced surface current $\vec{J}_s(\vec{r})$ on the PEC surface and using (2.18) to write $\vec{E}^{scat}$ in terms of $\vec{J}_s(\vec{r})$ and substituting the resulting expression into (2.25), we obtain

$$-\hat{n} \times \vec{E}^{inc}(\vec{r}) = \hat{n} \times i\omega\mu \int \left(\vec{J}_s(\vec{r}') + \frac{1}{k^2}\nabla' \cdot \vec{J}_s(\vec{r}')\nabla\right) g(\vec{r}, \vec{r}')d\vec{r}', \tag{2.26}$$

where $\vec{r}' \in S$. (2.26) is called as the normal-EFIE (N-EFIE). By performing a cross product with $-\hat{n}$ from the left, we obtain the tangential-EFIE (T-EFIE) as

$$\hat{n} \times \hat{n} \times \vec{E}^{inc}(\vec{r}) = -\hat{n} \times \hat{n} \times i\omega\mu \int \left(\vec{J}_s(\vec{r}') + \frac{1}{k^2}\nabla' \cdot \vec{J}_s(\vec{r}')\nabla\right) g(\vec{r}, \vec{r}')d\vec{r}', \tag{2.27}$$

which is used throughout this thesis.

2.1.3 Method of Moments for EFIE

MoM is one of the most commonly used methods to solve integral equations due to the fact that any IE can be considered as a linear operator $\mathcal{L}$ acting on the unknown function $\boldsymbol{f}(\vec{r})$, and the result is the known function $\boldsymbol{g}(\vec{r})$. In a compact form, any IE (and hence, EFIE used in this thesis) can be written as

$$\mathcal{L}\left\{\boldsymbol{f}(\vec{r})\right\} = \boldsymbol{g}(\vec{r}), \tag{2.28}$$

where $\boldsymbol{f}(\vec{r})$, in this thesis, is the induced current density $\vec{J}(\vec{r})$ on the PEC object. MoM procedure starts by expanding the unknown function $\boldsymbol{f}(\vec{r})$ in terms of finite number of linearly independent subdomain basis functions $\boldsymbol{b}_n(\vec{r})$ as

$$\boldsymbol{f}(\vec{r}) \approx \sum_{n=1}^{N} \alpha_n \boldsymbol{b}_n(\vec{r}), \tag{2.29}$$

where α_n, $n = 1, 2, ..., N$, are the unknown coefficents to be found at the end of the MoM procedure. Substituting (2.29) into (2.28) and using the fact that $\mathcal{L}$ is a linear operator, we obtain

$$\sum_{n=1}^{N} \alpha_n \mathcal{L}\left\{\boldsymbol{f}_n(\vec{r})\right\} d\vec{r} = \boldsymbol{g}(\vec{r}), \tag{2.30}$$

which is a single equation with N unknowns. By testing (2.30) with test functions, $\boldsymbol{t}_m(\vec{r})$, $m = 1, 2, ..., N$, (2.30) leads to the following linear system of equations

$$\int \boldsymbol{t}_m(\vec{r}) \cdot \sum_{n=1}^{N} \alpha_n \mathcal{L}\left\{\boldsymbol{f}_n(\vec{r})\right\} d\vec{r} = \int \boldsymbol{t}_m(\vec{r}) \cdot \boldsymbol{g}(\vec{r}), \tag{2.31}$$

where the integral is usually on a surface for SIEs, and hence, EFIE used in the thesis. Interchanging the order of summation and integration, (2.31) can be cast into a matrix form as

$$\boldsymbol{Z}\boldsymbol{i} = \boldsymbol{v}. \tag{2.32}$$

In (2.32), $\boldsymbol{Z}$ is called the impedance matrix, whose elements are given by

$$\boldsymbol{Z}_{m,n} = \int \boldsymbol{t}_m(\vec{r}) \cdot \mathcal{L}\left\{\boldsymbol{b}_n(\vec{r})\right\} d\vec{r}, \tag{2.33}$$

$\boldsymbol{v}$ is the voltage (excitation) vector, whose elements are given by

$$\boldsymbol{v}_m = \int \boldsymbol{t}_m(\vec{r}) \cdot \boldsymbol{g}(\vec{r}) d\vec{r}, \tag{2.34}$$

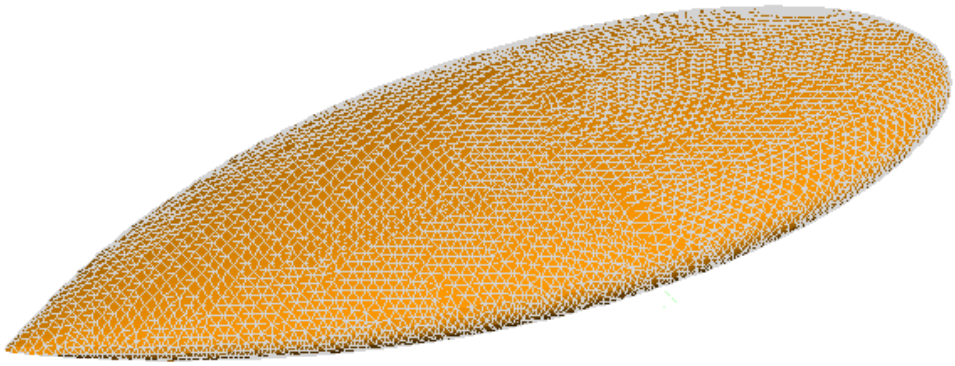


Figure 2.2: Example of a meshed surface. The NASA Almond surface with planar triangles.

and finally $\boldsymbol{i}$ is the current vector that contains the unknown coefficients, α_n.

The implementation of MoM requires meshed geometry. In this thesis, planar triangles are used to mesh the surface of the PEC object. Fig. 2.2 shows the modeling of an object with planar triangular meshes. Then, we use RWG functions [30] as both basis and test functions (Galerkin procedure [2]). Each RWG function is defined on a pair of triangles with a common edge of length l_n as shown in Fig. 2.3, and mathematically expressed as

$$\boldsymbol{b}_n(\vec{r}) = \begin{cases} \frac{l_n}{2A_n^+}(\vec{r} - \vec{r}_n^{\,+}) & , \vec{r} \in S_n^+ \\ \frac{l_n}{2A_n^-}(\vec{r}_n^{\,-} - \vec{r}) & , \vec{r} \in S_n^- \\ 0 & , \text{otherwise,} \end{cases} \tag{2.35}$$

where A_n^+ and A_n^- are the areas of the triangles S_n^+ and S_n^-, respectively, and $\vec{r}_n^{\,+}$ and $\vec{r}_n^{\,-}$ are the position vectors of the free vertices of S_n^+ and S_n^-, respectively. Note that the surface divergence of the n^{th} RWG is proportional to the surface charge density via the equation of continuity (2.5), and is given by

$$\nabla \cdot \boldsymbol{b}_n(\vec{r}) = \begin{cases} \frac{l_n}{A_n^+} & , \vec{r} \in S_n^+ \\ -\frac{l_n}{A_n^-} & , \vec{r} \in S_n^- \\ 0 & , \text{otherwise.} \end{cases} \tag{2.36}$$

It should be noticed at this point that the divergence of the RWG function (and hence the surface charge density) is finite and constant over each triangle, and

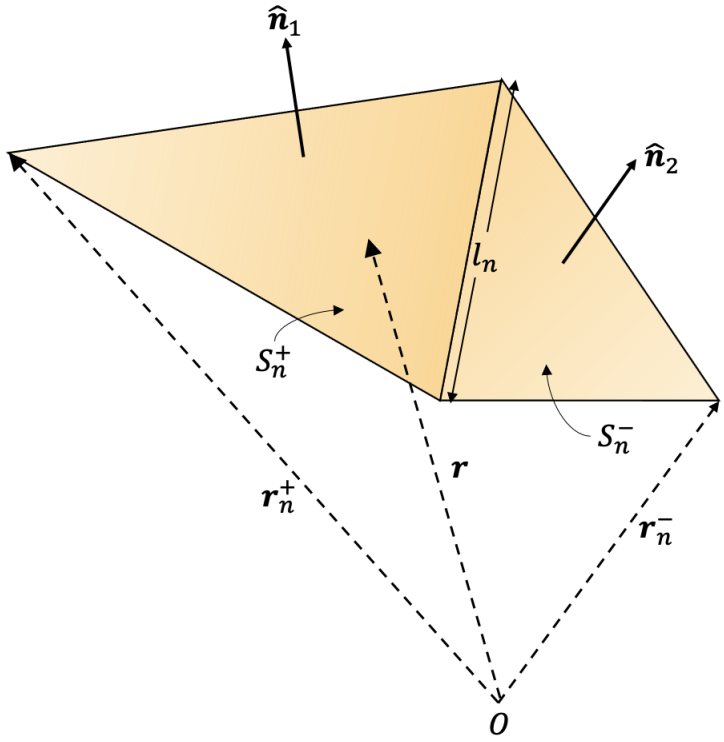


Figure 2.3: Illistration of the n^{th} RWG basis function associated with an interior edge of two triangles.

the total charge density associated with a pair of adjacent triangles is zero. This means that there is no charge accumulation on any edge (RWG basis functions are divergence confirming). Finally, using (2.27), the discretized form of the impedance matrix and the voltage vector entries are given by

$$\begin{aligned} \boldsymbol{Z}_{m,n} = i\omega\mu_0 \int_{S_m} \boldsymbol{t}_m(\vec{r}) \cdot \int_{S_n} \boldsymbol{b}_n(\vec{r}\,')g(\vec{r},\vec{r}\,')d\vec{r}\,'d\vec{r} \\ + \frac{1}{i\omega\epsilon_0} \int_{S_m} \nabla \cdot \boldsymbol{t}_m(\vec{r}) \int_{S_n} \nabla' \cdot \boldsymbol{b}_n(\vec{r}\,')g(\vec{r},\vec{r}\,')d\vec{r}\,'d\vec{r}, \end{aligned} \tag{2.37}$$

and

$$\boldsymbol{v}_m = -\int_{S_m} \boldsymbol{t}_m(\vec{r}) \cdot \vec{E}^{inc}(\vec{r})d\vec{r}, \tag{2.38}$$

respectively, where $\vec{E}^{inc}$ is a uniform plane wave throughout this thesis.

For the PEC objects discretized with N unknowns (not triangles but N RWG

functions), the MoM impedance matrix has N^2 elements that should be calculated separately. Thus, both CPU time and memory complexity to fill the impedance matrix have $\mathcal{O}(N^2)$ complexity. On the other hand, if (2.32) is solved directly (for example, Gaussian elimination), the computational complexity becomes $\mathcal{O}(N^3)$, whereas it is $\mathcal{O}(N^2)$ when (2.32) is solved iteratively for each iteration (cost of the matrix-vector product). Because electrically large problems cannot be solved with such complexities, MLFMA is a state-of-the-art method widely used in computational electromagnetics.

2.1.4 Multilevel Fast Multipole Algorithm

As mentioned in the previous section, IE-based solvers (MoM) result in a dense matrix and require $\mathcal{O}(N^3)$ operations for direct inversion of the resultant matrix equation. Iterative techniques, which are based on the matrix-vector-multiplication (MVM), can result in a computational complexity of $\mathcal{O}(N^2)$ for each iteration. Moreover, the memory requirement is also $\mathcal{O}(N^2)$ for both direct and iterative solutions. Such complexity levels, unfortunately, prohibit to solve electrically large real-life problems consisting of millions of unknowns. FMM, which is based on a fast MVM by computing the interactions between the basis and the test functions in a group-wise manner, reduces both the memory requirement and the computational complexity to $\mathcal{O}(N^{3/2})$ and then to $\mathcal{O}(NlogN)$ via its multilevel version, MLFMA.

The novel hierarchical ML-based method proposed in this thesis borrows certain concepts from MLFMA. Besides, MLFMA, as one of the most popular state-of-the-art method to solve IEs for EM scattering problems, is used for comparison purposes to assess the efficiency and accuracy of our novel method. Consequently, in this section, we review the basic steps of MLFMA.

MLFMA starts with constructing a tree structure. The object under analysis is placed in a cubic box, and a recursive process of dividing the box into eight identical subboxes through bisecting the box edges is performed, and its 2-D version is shown in Fig. 2.4. This hierarchy of boxes is called the tree structure.

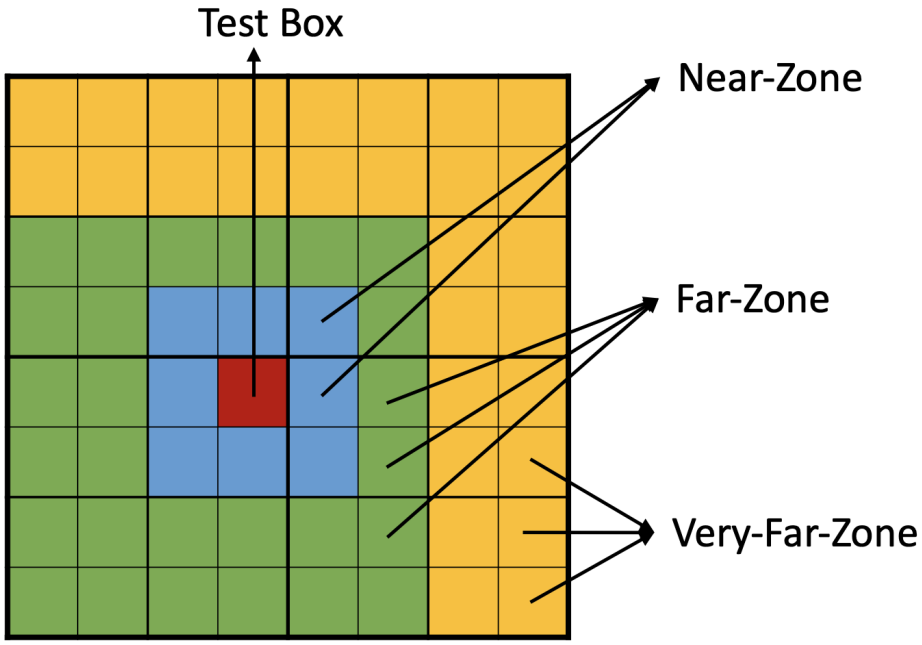


Figure 2.4: Definition of the near, far and very-far zone boxes with respect to the one-box-buffer scheme in a 2-D representation.

This process ends when a stopping criterion is met, such as predetermined level L (usually based on the error [31]) is reached or when the box size is below a certain size (known low-frequency (LF) problem [2]). During this step, all empty boxes at any level are omitted.

Then, using the one-box-buffer scheme [2] and considering the distance between the boxes, a grouping of boxes is performed such that a box may be a near-zone-box, a far-zone-box or a very-far-zone-box with respect to a given box of interest. Based on this grouping, the MVM operation can be expressed as

$$\boldsymbol{Z}\boldsymbol{i} = \boldsymbol{Z}_{NF}\boldsymbol{i} + \boldsymbol{Z}_{FF}\boldsymbol{i} = \boldsymbol{v}. \tag{2.39}$$

In (2.39), $\boldsymbol{Z}_{NF}$ denotes all the interactions among the near-zone boxes that are calculated directly (using MoM). The main contribution comes from the interactions of the basis and the test functions of the near-zone boxes as they are performed element-by-element manner. Far-zone interactions, denoted by $\boldsymbol{Z}_{FF}$, are computed in the group-by-group manner in each and every level resulting in the aforementioned improvement in the complexity. The very-far-zone boxes

have only indirect interactions among each other that only occur at higher levels.

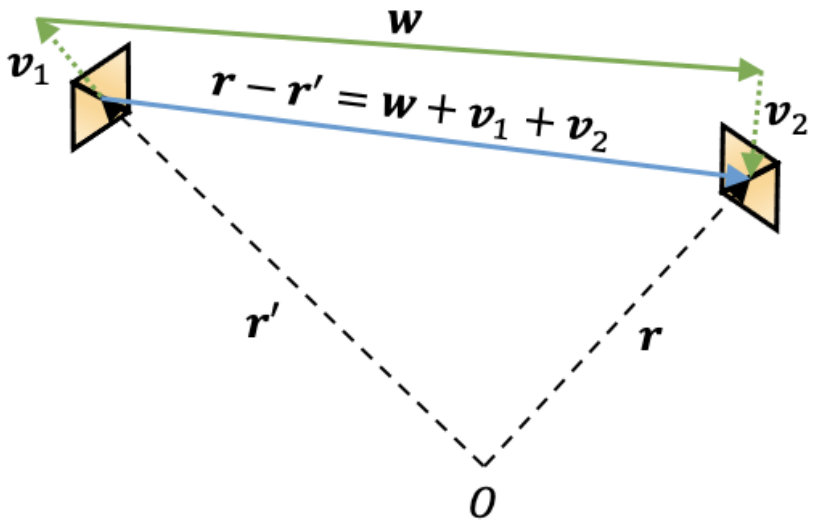


Figure 2.5: Interaction between a basis and a test function.

The efficiency of MLFMA comes from the computations of $\boldsymbol{Z}_{FF}$ elements in each iteration in a group-wise manner that is performed in three separable steps, namely, aggregation, translation, and disaggregation. The separation of these stages is possible due to the factorization and diagonalization of the free-space Green's function using Gegenbauer's addition theorem [2]. Consider an interaction between a basis and a test function, shown in Fig. 2.5, and let $\vec{v} = \vec{v}_1 + \vec{v}_2$ and $\vec{w}$ vectors satisfy $|\vec{w}| = w > v = |\vec{v}|$. Gegenbauer's addition theorem expands the free-space Green's function in terms of spherical harmonics as (factorization)

$$\frac{e^{ik|\vec{w}+\vec{v}|}}{4\pi|\vec{w}+\vec{v}|} = \frac{ik}{4\pi}\sum_{t=0}^{\infty}(-1)^t(2t+1)j_t(kv)h_t^{(1)}(kw)P_t(\hat{w}\cdot\hat{v}), \tag{2.40}$$

where j_t and $h_t^{(1)}$ are the spherical Bessel function and spherical Hankel function of the first kind, respectively, with order t, and P_t is the Legendre polynomial of order t. Then, representing the spherical waves as integrals of the plane wave spectrum as follows

$$j_t(kv)P_t(\hat{w}\cdot\hat{v}) = \frac{1}{4\pi i^t}\int e^{ik\hat{k}\cdot\vec{v}}P_t(\hat{k}\cdot\hat{w})d^2\hat{k} \tag{2.41}$$

with $\hat{k}$ being the radial direction and $d^2\hat{k}$ representing the sampling over the unit sphere, diagonal form of the Green's function (diagonalization) is obtained as

$$\frac{e^{ik|\vec{w}+\vec{v}|}}{4\pi|\vec{w}+\vec{v}|} = \frac{ik}{(4\pi)^2}\sum_{t=0}^{\infty}(i)^t(2t+1)h_t^{(1)}(kw)\int e^{ik\hat{k}\cdot\vec{v}}P_t(\hat{k}\cdot\hat{w})d^2\hat{k} \tag{2.42}$$

$$= \frac{ik}{(4\pi)^2}\int e^{ik\hat{k}\cdot\vec{v}}\sum_{t=0}^{\infty}(i)^t(2t+1)h_t^{(1)}(kw)P_t(\hat{k}\cdot\hat{w})d^2\hat{k}. \tag{2.43}$$

(2.43) can be rewritten as

$$\frac{e^{ik|\vec{w}+\vec{v}|}}{4\pi|\vec{w}+\vec{v}|} = \frac{ik}{(4\pi)^2}\int \beta(\vec{k},\vec{v})\alpha(\vec{k},\vec{w})d^2\hat{k}, \tag{2.44}$$

where

$$\beta(\vec{k},\vec{v}) = e^{i\vec{k}\cdot\vec{v}}, \tag{2.45}$$

and

$$\alpha(\vec{k},\vec{w}) = \sum_{t=0}^{\infty}(i)^t(2t+1)h_t^{(1)}(kw)P_t(\hat{k}\cdot\hat{w}) \tag{2.46}$$

are the shift and the translation operators, respectively.

In MLFMA implementations, the infinite summation in the translation operator, $\alpha(\vec{k},\vec{w})$, is truncated at a finite value τ (truncation number), which is determined by the excess bandwidth formula (EBF) [2] given by

$$\tau = 1.73ka + 2.18(d_0)^{2/3}(ka)^{1/3}, \tag{2.47}$$

where a is the box edge length, and $d_0 = -\log_{10}(\epsilon_d)$ is the desired digits of accuracy with ϵ_d being the desired relative error threshold. It should be noted that EBF is valid for large translation distances and fails at small box sizes. In [31], it is shown that

$$\tau \approx 14.14d_0 - 7.17 \tag{2.48}$$

can be used for electrically small boxes.

In the end, we can write (2.44) in accordance with Fig. 2.5 as

$$\frac{e^{ik|\vec{w}+\vec{v}|}}{4\pi|\vec{w}+\vec{v}|} = \frac{e^{ik|\vec{w}+\vec{v}_1+\vec{v}_2|}}{4\pi|\vec{w}+\vec{v}_1+\vec{v}_2|} \approx \frac{ik}{(4\pi)^2}\int \beta(\vec{k},\vec{v}_1)\alpha_\tau(\vec{k},\vec{w})\beta(\vec{k},\vec{v}_2)d^2\hat{k}. \tag{2.49}$$

In (2.49), $\beta(\vec{k}, \vec{w}_1)$ and $\beta(\vec{k}, \vec{w}_2)$ are simple shift functions that represent the aggregation and the disaggregation operations, and $\alpha_\tau(\vec{k}, \vec{w})$, given by

$$\alpha_\tau(\vec{k}, \vec{w}) = \sum_{t=0}^{\tau} (i)^t (2t+1) h_t^{(1)}(kw) P_t(\hat{k} \cdot \hat{w}) \tag{2.50}$$

is the translation function.

The truncation number directly affects the accuracy of the translation operator and determines the number of sampling points along θ and ϕ [2] for the integration in (2.41)–(2.43). Fig. 2.6 shows the visualization of the tree structure used in multilevel scheme. During the aggregation stage, the radiated fields for each box at every level are computed recursively from the leaf-level (i.e., the deepest level) to the mother-level (the level that has the box which encloses the entire geometry). At different levels, the truncation number may differ due to the change of the box size. Hence, the number of θ and ϕ samples in (2.49) may change as we move from one level to another. Therefore, we may need interpolation when changing the levels. Once the computation of the aggregated fields are computed, translations are computed for boxes with available far-zone interactions. Translated fields are then distributed to lower levels by disaggregation that is performed from the mother-level to leaf-levels (top-to-bottom). Similar to aggregation, anterpolation is needed when changing the levels. Since the Nyquist criterion is utilized for sampling rate, interpolation and anterpolation are performed without error [2].

Note that the improved computational complexity of MLFMA is due to the diagonalization of the free-space Green's function given in (2.42) and (2.43). However, in the diagonal form of the Green's function, the spherical Hankel function, $h_t^{(1)}(kw)$, in the translation operator grows rapidly for small arguments and cannot be compensated with any other terms due to the plane wave expansion. Therefore, when the box size (i.e., box edge length in terms of λ) decreases, w becomes small (in terms of λ) (based on the one-box-buffer scheme), the spherical Hankel function increases exponentially, which may lead to the overflow of the registers for fixed-precision computers [31]. This is the well-known low-frequency breakdown problem of MLFMA, and requires special treatments. One approach

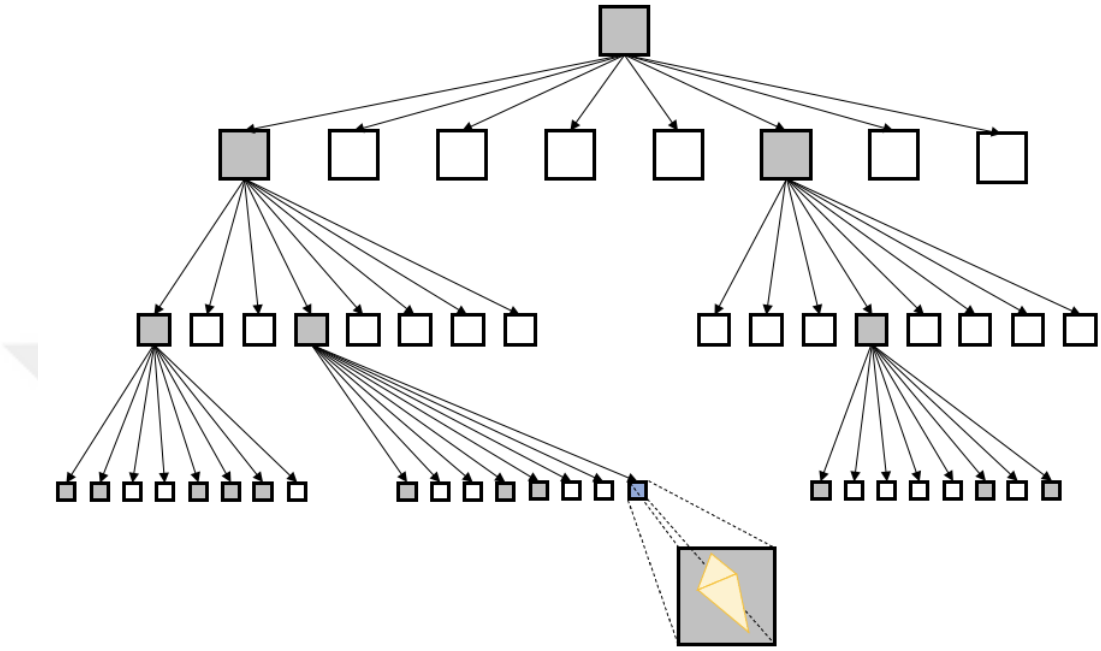


Figure 2.6: Visualization of the tree structure obtained by dividing the non-empty boxes into 8 cubes in 3-D space.

is to use multipoles explicitly without converting them to plane waves [32]. Another approach is to deform the angular integration path in such a way that the contribution of evanescent waves is included for sub-wavelength box-to-box interactions [33]. In both approaches, the complexity increases due to alternative expansion formulations of the free-space Green's function.

2.2 Machine Learning Related Preliminaries

The novel hierarchical ML-based electromagnetic solver introduces a regression problem, which has a real-valued continuous output, and can be modeled with the feedforward neural network models (which will be called networks throughout this thesis), where the information flows from the input to the output (no feedback from the output to the input) [11, 12].

The only structures discussed in this thesis are 2-layer neural network structures,

and the networks which are not addressed in this section may be utilized as a future work (see Chapter 5). First, we will discuss the network structures used in the novel hierarchical ML-based electromagnetic solver, FCNN and CNN. Then, some hyperparameters of the networks will be discussed. Additionally, different network structures can be utilized as future work.

2.2.1 The 2-layer FCNN

FCNN is a computational model that contains many computational units which are connected to each other completely [10]. Fig. 2.7 shows the FCNN structure utilized in this thesis. Each circle corresponds to a computational unit. Yellow, green, and blue units are the input, hidden, and output units, respectively. The arrows show the feed-forward flow from inputs to outputs. Each computational unit has multiple inputs and one output.

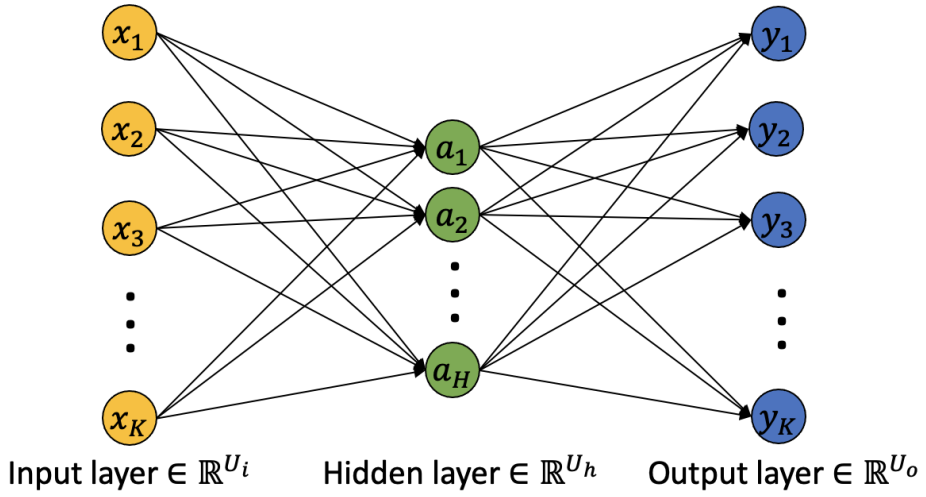


Figure 2.7: The structure of a feed-forward fully connected neural network with one hidden layer.

A network provides a mapping from the inputs to the outputs. To find the mapping, the supervised learning task is used with a training set $D = \{(\boldsymbol{X}, \boldsymbol{Y})\}$ where $\boldsymbol{X} \in \mathbb{R}^{U_i \times N_D}$ is the input and $\boldsymbol{Y} \in \mathbb{R}^{U_o \times N_D}$ is the output with U_i and U_o being the numbers of inputs and outputs, respectively, and N_D is the number of

training data instances [11]. Consider a data instance, $\{(\boldsymbol{x}, \boldsymbol{y})\}$ where $\boldsymbol{x} \in \mathbb{R}^{U_i \times 1}$, $\boldsymbol{y} \in \mathbb{R}^{U_o \times 1}$. The mapping can be described with a function, $\Omega : \mathbb{R}^{U_i \times 1} \rightarrow \mathbb{R}^{U_o \times 1}$, which represents the relation between the input and output vectors as $\boldsymbol{y} = \Omega(\boldsymbol{x})$. ML algorithms aim to find an appropriate Ω function. Typically, the function Ω contains a chain of multiple functions [12], where the Ω function can be described as $\Omega(\boldsymbol{x}) = f^{(2)}(f^{(1)}(\boldsymbol{x}))$ with $f^{(1)}$ being the first layer, and $f^{(2)}$ being the second layer of the network. $f^{(2)}$ is called also called the output layer of the network since it is the final layer, and $f^{(1)}$ is also called the hidden layer since the values are not explicitly defined in the training dataset [12].

The numbers of units at the input, hidden, and output layers are defined as U_i, U_h, U_o, respectively. Throughout this thesis, the number of hidden units is set to be H, and this hyperparameter will be used to optimize the complexity and accuracy of the ML-based electromagnetic solver. In order to decide the number of hidden units, an additional dataset, namely, the validation dataset, can be used. A validation set can be obtained by portioning the training set. In this thesis, 90% of the dataset is used as the training dataset, and the rest is used as the validation dataset. The validation set has a significant rule to avoid overfitting or underfitting [11].

The numbers of inputs and outputs are the same for our method and set to be K ($U_i = U_o = K$). The networks, which contain the same number of inputs and outputs, are called autoassociator [10]. Having more inputs or outputs than the number of hidden units ($K > H$) implies dimensionality reduction, and due to the non-linearity introduced by the FCNN structure, this problem can be considered as a non-linear dimensionality reduction [10]. After the parameter tuning, the first layer acts as an encoder, while the second layer act as a decoder. The output of the hidden layer has sufficient properties of the input signal to estimate the output signal.

In order to express the network shown in Fig. 2.7 mathematically, consider a data instance, $\{(\boldsymbol{x}, \boldsymbol{y})\}$ where $\boldsymbol{x} \in \mathbb{R}^{K \times 1}$, $\boldsymbol{y} \in \mathbb{R}^{K \times 1}$, again. The goal is to find a function, $\Omega : \mathbb{R}^{U_i \times 1} \rightarrow \mathbb{R}^{U_o \times 1}$, that estimates the outputs from the inputs as $\boldsymbol{y} \approx \Omega(\boldsymbol{x}) = f^{(2)}(f^{(1)}(\boldsymbol{x}))$. The output of the first layer (or the hidden layer

for the structure shown in Fig. 2.7), $\boldsymbol{a} = \{a_1, a_2, ..., a_H\}$ where $a_i \in \mathbb{R}$, can be obtained as

$$\boldsymbol{a} = f^{(1)}(\boldsymbol{x}) = g^{(1)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right), \tag{2.51}$$

where $\boldsymbol{W}^{(1)} \in \mathbb{R}^{H\times K}$ is the weight matrix of the first layer with $w_{ij}^{(1)}$ is the weight of the connection between the i^{th} hidden unit and the j^{th} input, $\boldsymbol{b}^{(1)} \in \mathbb{R}^{H\times 1}$ is the bias vector of the first layer (bias term corresponds to the weight coming from the bias unit, +1), and $g^{(1)}$ is the activation function of the first layer.

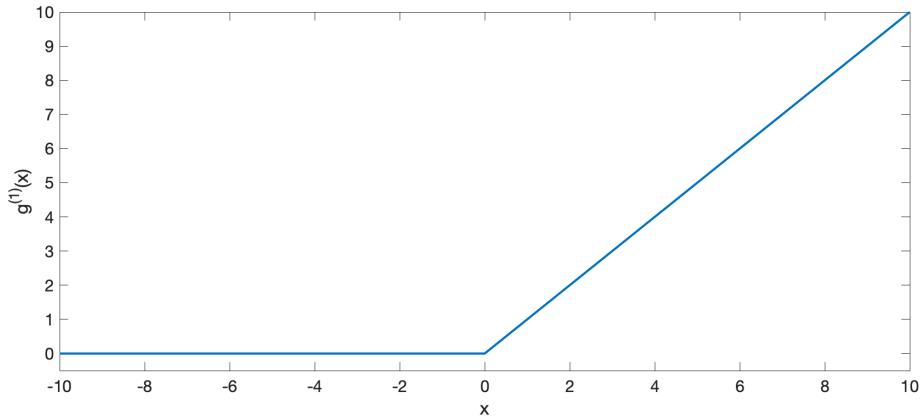


Figure 2.8: Rectified Linear Unit (ReLU) activation function.

The activation function for the hidden layer is commonly chosen as a non-linear one to introduce the non-linearity to the Ω function. In this thesis, the non-linear activation function used in the hidden layer is the Rectified Linear Unit (ReLU) activation function, shown in Figure 2.8. It can be mathematically expressed as

$$g^{(1)}(x) = \max(0, x). \tag{2.52}$$

This function is commonly used due to its linear-like behavior [12] as it is the combination of two linear functions. Therefore, it preserves the linear function behavior to provide large but non-saturated derivatives for the active units [12]. In general, the models containing the ReLU activation function are easy to train with gradient-based methods and converge faster [11]. This activation function must be used carefully since the derivative at zero is problematic, and the output is not zero-centered [11].

We can use the output of the first layer, $\boldsymbol{a}$, to calculate the output of the second layer (or the output layer of the entire network shown in Fig. 2.7), $\boldsymbol{y} = \{y_1, y_2, ..., y_K\}$ where $y_i \in \mathbb{R}$, as

$$\boldsymbol{y} = f^{(2)}(\boldsymbol{a}) = g^{(2)}\left(\boldsymbol{W}^{(2)}\boldsymbol{a} + \boldsymbol{b}^{(2)}\right), \tag{2.53}$$

where $\boldsymbol{W}^{(2)} \in \mathbb{R}^{K \times H}$ is the weight matrix of the second layer with $w_{ij}^{(2)}$ is the weight of the connection between the i^{th} output and the j^{th} hidden unit, $\boldsymbol{b}^{(2)} \in \mathbb{R}^{K \times 1}$ is the bias vector of the second layer, and $g^{(2)}$ is the activation function of the second layer. The activation function of the output layer is commonly chosen as a linear function, defined as

$$g^{(2)}(x) = x. \tag{2.54}$$

This function provides an output, which is a linear combination of the non-linear outputs of the hidden layer [10]. As a result, the function Ω can be obtained as

$$\Omega(\boldsymbol{x}) = g^{(2)}\left(\boldsymbol{W}^{(2)}\left(g^{(1)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)\right) + \boldsymbol{b}^{(2)}\right) \tag{2.55}$$

where $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, $\boldsymbol{b}^{(1)}$, $\boldsymbol{b}^{(2)}$ are the parameters of the 2-layer neural network structures.

2.2.2 The 2-layer CNN

CNN, which contains at least one convolution layer, is a commonly used network for the known grid-like topologies [12]. Convolution layer utilizes convolution, which is a specialized linear operation. Convolution is denoted as $\boldsymbol{S} = \boldsymbol{X} * \boldsymbol{K}$, where $\boldsymbol{X}$ is the input of the convolution, $\boldsymbol{K}$ is the filter (or kernel), and $\boldsymbol{S}$ is the output of the convolution.

Since the novel hierarchical ML-based method works on 3-D space, 3-D CNN structure is utilized. However, we only discuss 2-D CNN for the sake of the visualization, and it can be easily extended to 3-D CNN.

The 2-D convolution operation is defined mathematically as

$$\boldsymbol{S}(i,j) = \boldsymbol{X} * \boldsymbol{K}(i,j) = \sum_l \sum_m \boldsymbol{X}(l,m)\boldsymbol{K}(i-l, j-m). \tag{2.56}$$

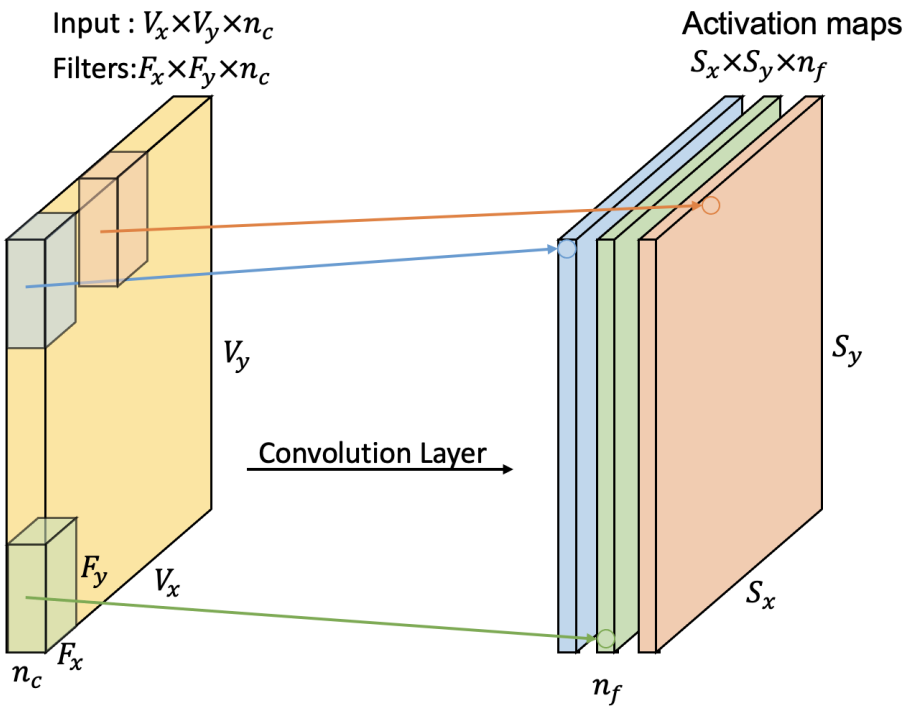


Figure 2.9: 2-D convolution layer.

Fig. 2.9 shows the 2-D convolution layer. The input of the 2-D CNN is $\boldsymbol{X} \in \mathbb{R}^{V_x \times V_y \times c}$, where V_i ($i = x$ or y) is the number of elements along i^{th} direction (the spatial dimensions that the filters move along) and c is the number of channels (also defined as the depth of the input). There may be n_f filters as $\boldsymbol{F} \in \mathbb{R}^{F_x \times F_y \times, c}$, where F_i is the number of elements along i^{th} direction (the spatial dimensions that the filters move along). Note that the numbers of the channels (depth) are the same for both input and the filters. 2-D convolution means that 3-D filters move in 2-D spatial space.

Similar to the FCNNs, a bias term is also added to the output of each filter, and the non-linear activation function is applied (ReLU) after the linear convolution operation. Each filter provides an activation map, and the output of the convolution layer (multiple filters) is called an activation map.

Before any discussion on the size of the output of the convolution layer, there are two terms to be discussed: stride, s, and padding, p. Stride corresponds

to the amount of movement on the input. The default setting is $s = 1$, which means that incrementation along the specified axis is 1. Therefore, the filter moves from the i^{th} element to the $(i+1)^{th}$ element. Padding is adding additional terms to the input, before the convolution process. Valid padding is used for the CNN structure, which corresponds to the case where only valid data is used for convolution (i.e., no padding, $p = 0$).

Finally, the size of the output of the convolution layer, $\boldsymbol{S} \in \mathbb{R}^{S_x \times S_y \times n_f}$, depends on the size of the input, filters, padding and stride as

$$S_i = (V_i - F_i + 2p)/s, \tag{2.57}$$

where $i = x, y$.

$$\boldsymbol{Y}^{(1)} = f^{(1)}(\boldsymbol{X}) = g^{(1)}\left(\boldsymbol{X} * \boldsymbol{F}^{(1)} + \boldsymbol{b}^{(1)}\right) \tag{2.58}$$

where $\boldsymbol{Y}^{(1)}$ is the output of the convolution layer, $\boldsymbol{F}^{(1)}$ is the matrix that contains the weights of the filters, $\boldsymbol{b}^{(1)} \in \mathbb{R}^{n_f \times 1}$ is the bias term, and $g^{(1)}$ is the non-linear activation function for the first layer, which is selected as ReLU. After flattening, reshaping the output of the convolution layer from $o \times o \times o \times n_f$ to $o^3 n_f \times 1$, fully connected layer is introduced with $o^3 n_f$ hidden units. $\boldsymbol{F}^{(1)}$, $\boldsymbol{W}^{(2)}$, $\boldsymbol{b}^{(1)}$, $\boldsymbol{b}^{(2)}$ are the parameters of the 2-layer CNN structure.

2.2.3 Hyperparameters of Neural Network Structures

The training algorithm establishes a correlation between inputs and outputs by tuning the parameters. These parameters are $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, $\boldsymbol{b}^{(1)}$, $\boldsymbol{b}^{(2)}$ for the 2-layer FCNN structure and $\boldsymbol{F}^{(1)}$, $\boldsymbol{W}^{(2)}$, $\boldsymbol{b}^{(1)}$, $\boldsymbol{b}^{(2)}$ for the 2-layer CNN structure. Training is completed with an optimizer to minimize the loss function, which tunes these parameters. In this thesis, Adam optimizer [34], an efficient method for stochastic gradient-based optimization and commonly used for parameter tuning for ML algorithms, is used [12]. It has a hyper-parameter, called learning rate, which corresponds to a step size that controls the amount of change at the weights

during training. It should be selected wisely to find a good convergence on the training dataset. By choosing the learning rate too high or too low, the model may convergence to the sub-optimal solution [12].

Typically, the loss function used in regression problems is the mean absolute error (MAE) or the mean squared error (MSE) [12]. For this regression problem, the loss function is selected as MAE, which is defined as

$$f_{loss} = \frac{1}{N_D} \sum_{n=1}^{N} |\Omega(\boldsymbol{x}) - \boldsymbol{y}|. \tag{2.59}$$

The gradient descent algorithm is an iterative solver and updates the unknown parameters at each iteration. The number of training data instances used at each iteration, which is called batch size, can be selected by the user. It means that, at each iteration, a random part of the training dataset, called mini-batches, which contains batch size data instances, is used to calculate the loss function and update the weights. Therefore, the batch size is basically the number of training data instances used at each iteration. The batch size can be specified from 1 to the number of instances in the training dataset. Typically, the batch size is chosen between 1 and a few hundred, as a power of 2 [12, 35].

Instead of the number of iterations, the number of epochs is commonly used in ML. When all samples in the training dataset are used once, it is called one epoch. The number of iterations per epoch depends on the number of training data instances and the batch size. The number of epochs needed for a good convergence depends on the learning rate and the batch size. The number of epochs also has an optimal point where training less may cause under-fitting, while training more may cause over-fitting [12].

Throughout this thesis, all network trainings are performed in TensorFlow, which is an end-to-end open-source platform for ML [36].

Chapter 3

The Proposed Hierarchical Machine-Learning-Based Method

This chapter presents the proposed hierarchical ML-based method starting with the details of its general implementation, including the tree structure construction, clustering, and near-zone interactions. Then, the computation of the far zone interactions is explained step by step, which constitutes the main contribution of the proposed method. Finally, the complexity analysis of the proposed method is provided and contrasted with the state-of-the-art MLFMA.

3.1 Implementation of the Proposed Method

This section describes the implementation details of the proposed method at a higher level to solve EM scattering problems, and Section 3.2 explains the details of the far-zone interaction calculations.

In this thesis, the proposed hierarchical ML-based method solves EFIE by utilizing RWG basis functions as both basis and test functions, which is discussed in Section 2.1.3. Therefore, it requires a meshed surface of the PEC object with

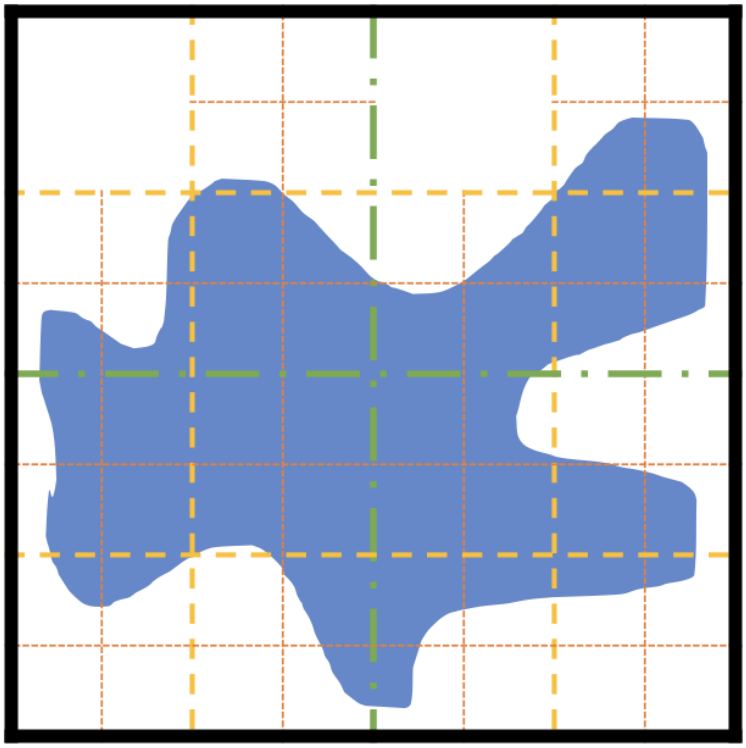


Figure 3.1: 2-D visualization of the tree structure.

planar triangles. Then, the oct-tree structure in three-dimensional (3-D) space is constructed, similar to MLFMA (see Section 2.1.4). Briefly, the geometry to be investigated is placed inside a cube at the first level with an edge length of a_1 (as shown in Fig. 3.1 with black lines). At level 2, the cube is divided into eight cubes with an edge length of $a_2 = a_1/2$ in 3-D space (as shown in Fig. 3.1 with green lines), and all boxes are in the near-zone of other boxes. Therefore, if the tree structure is stopped at that level, all interactions are calculated directly (MoM). The same division is completed at the third level, and the far-zone boxes are introduced with a box size of $a_3 = a_2/2 = a_1/4$ (as shown in Fig. 3.1 with yellow lines). After that level, the boxes are divided into smaller boxes at each level in the same manner (The fourth level is shown in Fig. 3.1 with red lines). For the boxes at the l^{th} level, where $l > 2$, the far-zone boxes are introduced. Note that only non-empty boxes are considered and divided into smaller boxes at each level. The leaf level boxes (boxes at the lowest level) contain the discretization elements (i.e., RWGs). Each RWG basis function is assigned to its parent box based on

the location of the mid-point of its common edge. Then, the boxes are defined as the near-zone, far-zone, and very-far-zone boxes using the one-box-buffer scheme, similar to MLFMA (see Section 2.1.4).

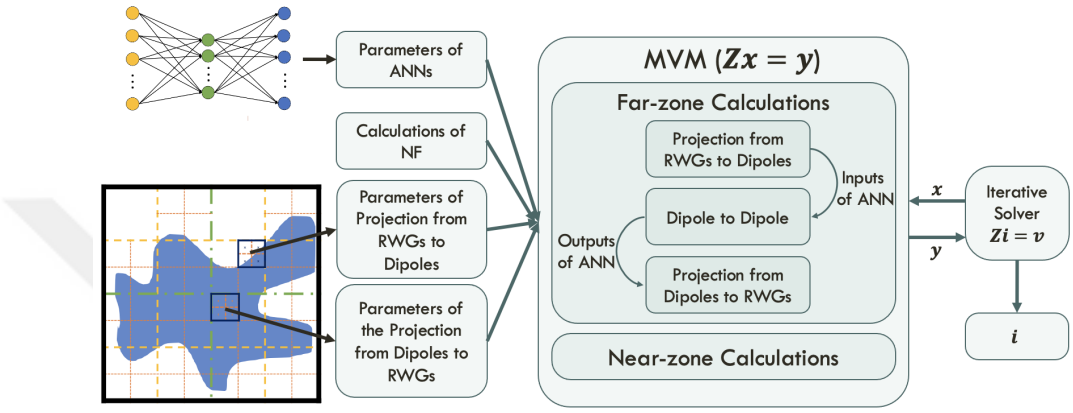


Figure 3.2: Typical simulation environment of the novel hierarchical ML-based method.

Then, the linear system obtained from the discretization of EFIE can be solved by an iterative solver. MVM is computed for a given coefficient vector at each iteration of the iterative solver by attacking near-zone and far-zone interactions separately, as shown in Fig. 3.2. Similar to MLFMA, the proposed method performs the multiplication of impedance matrix and coefficient vector (MVM) at each iteration of the iterative solver. The near-zone interactions, which provide the dominant contribution, are computed directly (element-by-element) and stored in the memory to be used in the MVM at each iteration. For the far-zone calculations, ANN models to approximate the box-to-box interactions are required. Starting from the third level, the boxes are divided into smaller sample cubes, which is discussed in Section 3.2. The number of sample cubes is determined according to the box size and the desired relative error, which is discuss in Section 3.2.2. If the appropriate ANN models have been already trained, they can be used. Otherwise, the training data sets for 16 translation directions at each level are generated as discussed in Section 3.2.1. After the hyperparameters of the ANN model are selected, the ANN models for 16 translation directions are trained. Before the iterative solver, 16 networks are extended to 316 translation directions using the rotational symmetry, and the parameters of the ANN models are stored in the memory, as shown in Fig. 3.2. Finally, the parameters of the

projections discussed in Section 3.2.2 and Section 3.2.4 are calculated and stored in the memory to be used in the iterative solver.

The novelty of the proposed method comes from the far-zone calculations. The far-zone calculations are computed from a basis box to a test box (box-to-box) with the aid of ML models, which approximate the free-space dyadic Green's function, as will be explained in a detailed way in the next section.

3.2 Details of the Far-Zone Calculations

3.2.1 Dyadic Green's Function and Training ML Estimators

Consider a tree structure constructed for a given object. The far zone calculations start by dividing the basis and test boxes into the basis and the test sample cubes, respectively, as shown in Fig. 3.3. The centers of these sample cubes are defined as sample points, which are shown in Fig. 3.3 with 'X' markers. The number of sample cubes inside a given box, N_S, is a design parameter, and its selection is discussed in the next section. In general, increasing the number of sample boxes provides more accurate results but increases the complexity of the method. Fig. 3.3 shows two cubes with different numbers of sampling points, N_S, in 2-D and 3-D spaces.

Consider the b^{th} basis and the t^{th} test boxes at the l^{th} level of the tree structure with the box size (box-edge-length) being a_l and each one containing N_S sample points. The electric field vectors at the test sample points of the test box due to the Hertzian dipoles at the basis sample points of the basis box are calculated using the dyadic Green's function (see Appendix A for its derivation). The electric field at the p^{th} test sample point inside the test box due to the q^{th} basis sample

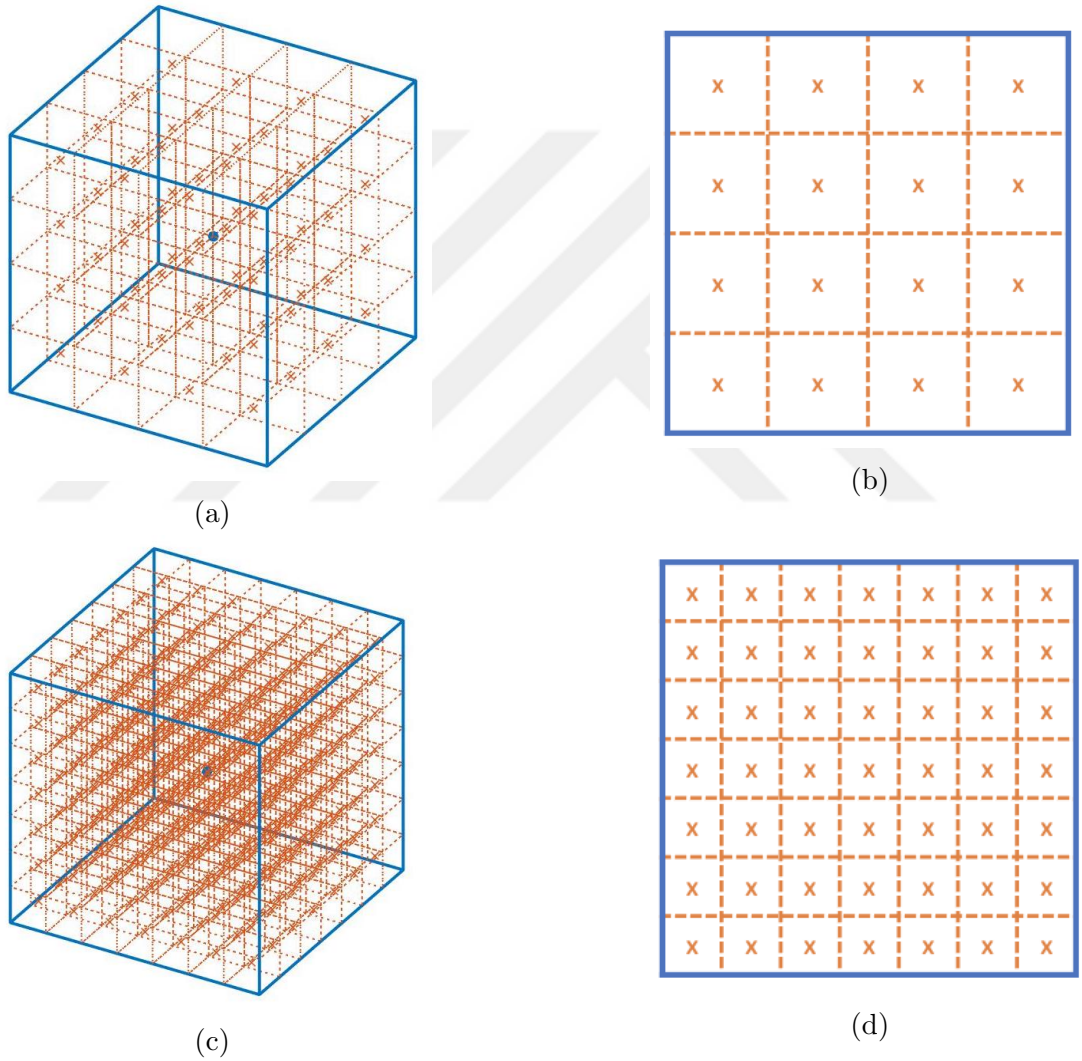


Figure 3.3: Boxes containing different number of sample cubes. (a) 64 sample cubes in 3-D ($N_S = 64$) and (b) is its cross-section in 2-D. (c) 343 sample cubes in 3-D ($N_S = 343$) and (b) is its cross-section in 2-D.

point inside the basis box is given by

$$\overline{\overline{G}}_{t,b,p,q} = \left\{ \left(1 + \frac{i}{kR_{t,b,p,q}} - \frac{1}{k^2 R_{t,b,p,q}^2}\right) \overline{\overline{I}}_{b,q} - \left(1 + \frac{3i}{kR_{t,b,p,q}} - \frac{3}{k^2 R_{t,b,p,q}^2}\right) \hat{R}_{t,b,p,q}\hat{R}_{t,b,p,q} \right\} \frac{e^{-ikR_{t,b,p,q}}}{R_{t,b,p,q}}, \quad (3.1)$$

where $\overline{\overline{I}}_{b,q}$ is the dyadic representation of the coefficients of the dipole excitation current at the q^{th} sample point inside the b^{th} basis box given by $\overline{\overline{I}}_{b,q} = \alpha_{b,q,x}\hat{x}\hat{x} + \alpha_{b,q,y}\hat{y}\hat{y} + \alpha_{b,q,z}\hat{z}\hat{z}$, $R_{t,b,p,q} = |\vec{r}_{t,p} - \vec{r}_{b,q}|$ is the distance between the p^{th} test sample point inside the t^{th} test box with a position vector of $\vec{r}_{t,p}$ and the q^{th} basis sample point inside the b^{th} basis box with a position vector of $\vec{r}_{b,q}$, and finally $\hat{R}_{t,b,p,q}$ is the unit vector along the $\vec{r}_{t,p} - \vec{r}_{b,q}$ direction.

We can superpose the electric fields due to all basis sample points for a given basis box to find the electric field at a test sample point at a given test box, corresponding to box-to-point calculations in the far zone. The overall electric field at the p^{th} test sample point in the t^{th} test box due to all basis sample points in the b^{th} basis box is given by

$$\overline{\overline{G}}_{t,b,p} = \sum_{q=1}^{N_S} \overline{\overline{G}}_{t,b,p,q}. \quad (3.2)$$

Fig. 3.4 shows random dipole excitation vectors (purple arrows) in a basis box (green box) and electric field vectors (with yellow arrows) in a test box (blue box) calculated using (3.1)–(3.2) for a one-box-buffer case with $a_l = 0.25m$ at $f = 300$MHz and $N_S = 64$. Note that the center of the basis box is considered as the origin.

Since the method aims to find box-to-box interactions, $\overline{\overline{G}}_{t,b,p}$ is needed to be calculated for each test sample point. Additionally, since each iteration of the iterative solver provides a different coefficient vector, $\overline{\overline{I}}_{b,q}$ is calculated at each iteration to be used in (3.1)–(3.2) to find the electric field vectors at test sample points. Instead of calculating (3.1)-(3.2) directly, the approximation can be obtained with the aid of ML structures. These ML structures reduce the complexity

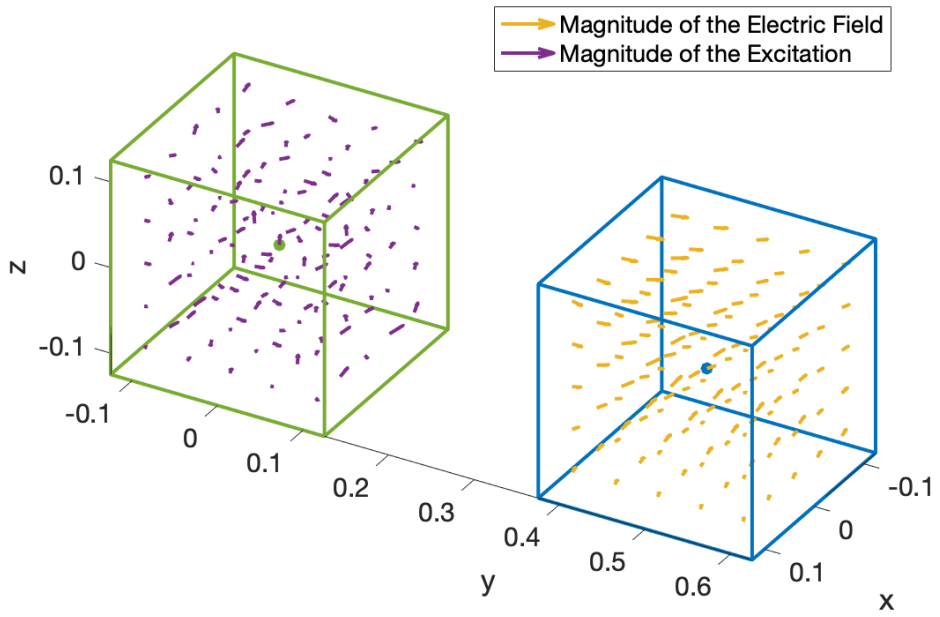


Figure 3.4: Visualization of the box-to-box interactions for the one-box-buffer case along the y direction with $a_l = 0.25m$ at $f = 300$MHz and $N_S = 64$. The green box is the basis box, whose center is the origin, and the blue box is the test box.

by providing an approximation and combining the calculations of $\overline{\overline{G}}_{t,b,p}$ for all test points.

The inputs of the neural networks are the parameters of the Hertzian dipoles. Basically, under the assumption that basis sample points' locations are predefined, the inputs are the x, y, and z components of the current coefficients dipoles (i.e., excitations), which are, in general, complex numbers. However, because the complex numbers are not applicable to real-valued neural networks, each component is also divided into real and imaginary parts. As a result, each dipole is defined by six parameters. Similarly, under the assumption that test sample points' location is also predefined, the neural network outputs are the parameters of the overall electric field at these test points. Hence, each point has six parameters: real and imaginary parts of the x, y, z components of the electric field. Consequently, the numbers of inputs and outputs are each $6N_S$.

Traditional ML techniques are sufficient to approximate the desired superposition of the Green's function since the features are pre-specified. The proposed method utilizes two types of networks discussed in Section 2.2.1 and Section 2.2.2 to approximate the electric fields at the test box due to the dipoles at the given basis box. For the FCNN structure, the numbers of inputs, U_i, and outputs, U_o, are set to be $K = 6N_S$, while the rest stays the same. For CNN structure, the input dimension is defined as $N_{Sk} \times N_{Sk} \times N_{Sk} \times c$ where $N_{Sk} = \sqrt[3]{N_S}$ and $c = 6$, while the rest of the structure stays the same.

To find the parameters of the ML structures, the training dataset is generated considering the fact that the current coefficients of dipoles are 3-D complex vectors. Also, it is assumed that each dipole at each basis sample point is independent of others. For each component of the input, the complex value, $\sqrt{r_j}e^{i\phi_j}$ where $j = x, y, z$, is randomly selected inside the unit circle as $r_j \sim U(0, 1)$ and $\phi_j \sim U(0, 2\pi)$, where $U(a, b)$ represents the uniform distribution on the interval of (a, b). After generating random excitations, the electric fields are calculated using (3.1)–(3.2). Then, the real and imaginary parts of the x, y, z components of both the inputs (i.e., the random excitation vectors) and the outputs (i.e., the electric field vectors) are extracted before the training.

Figure 3.4 shows only one possible combination (one-box buffer case along the y-direction). However, at each level of the tree structure, there are 316 possible far-zone directions [2], which means that 316 networks are required. However, by using the rotational symmetry, the number of needed networks can be reduced to 16. For example, if we have a network, that estimates the one-box-buffer case along the $+z$ direction ($[0, 0, 2]$), it can be used for the one-box-buffer cases along the $\pm x$ ($[\pm 2, 0, 0]$), $\pm y$ ($[0, \pm 2, 0]$), and $-z$ ($[0, 0, -2]$) directions. The list of the unique directions required for a complete description of all translation distances is given in Table 3.1.

Table 3.1: The unique translation directions needed for a complete description of all translation distances.

$[0,0,2]$	$[0,1,2]$	$[0,2,2]$	$[1,1,2]$
$[1,2,2]$	$[2,2,2]$	$[0,0,3]$	$[0,1,3]$
$[0,2,3]$	$[0,3,3]$	$[1,1,3]$	$[1,2,3]$
$[1,3,3]$	$[2,2,3]$	$[2,3,3]$	$[3,3,3]$

3.2.2 Projection from RWGs to Dipoles

The tree structure is constructed, and box-to-box far-zone interactions are performed by approximating the superposition of the free-space dyadic Green's function via ML, as explained in the previous section. However, the object to be investigated is discretized with RWG basis functions. Therefore, we need a projection from RWG basis functions to dipoles. Basically, we aim to find the corresponding complex-valued coefficients of the dipoles at the basis sample points. In this thesis, we find these coefficients by performing a far-field mapping. Briefly, in a given basis box, we find the far-field patterns of the associated RWG functions, then we project the patterns onto the dipoles that best fit this far-field pattern. The details of this projection are as follows.

First, we must emphasize that the aforementioned far-field mapping is performed in the cartesian coordinate system. Next, consider a k-directed ($k = x, y, z$) dipole with a unit magnitude and let $\boldsymbol{A}_{jk} \in \mathbb{C}^{N_F \times N_S}$ be the j-component ($j = x, y, z$) of its far-field pattern. Let $\boldsymbol{b}_{n,k} \in \mathbb{C}^{N_S \times 1}$ be the unknown complex-valued current coefficients of these dipoles due to the n^{th} RWG function, and $\boldsymbol{s}_{n,j} \in \mathbb{C}^{N_F \times 1}$ be the j-component of the far-field pattern of the n^{th} RWG function, where N_F is the number of far-field points, and N_S is the number of basis sample points in a basis cube. The linear system, which is used to find the projection from an RWG function to dipoles, is given by

$$\begin{bmatrix} \boldsymbol{A}_{xx} & \boldsymbol{A}_{xy} & \boldsymbol{A}_{xz} \\ \boldsymbol{A}_{yx} & \boldsymbol{A}_{yy} & \boldsymbol{A}_{yz} \\ \boldsymbol{A}_{zx} & \boldsymbol{A}_{zy} & \boldsymbol{A}_{zz} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_{n,x} \\ \boldsymbol{b}_{n,y} \\ \boldsymbol{b}_{n,z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{n,x} \\ \boldsymbol{s}_{n,y} \\ \boldsymbol{s}_{n,z} \end{bmatrix}. \tag{3.3}$$

In (3.3), our proposed method chooses N_S and N_F to provide an under-determined system ($N_F < N_S$), and a least-squares solution is obtained using the singular value decomposition to find the unknown complex-valued current coefficients of the dipoles, $\boldsymbol{b}_n$. Note that this step (i.e., calculation of $\boldsymbol{b}_n = [\boldsymbol{b}_{n,x}, \boldsymbol{b}_{n,y}, \boldsymbol{b}_{n,z}]^T$) is done before the iterative solver for each RWG by setting its coefficient to 1, and $\boldsymbol{b}_n$s are stored in the memory. At each iteration of the iterative solver, $\boldsymbol{b}_n$ is multiplied with the corresponding coefficient coming from the iterative solver to project the RWG to dipoles at the basis sample points. Then, $\boldsymbol{b}_n$s of the RWGs in the corresponding basis box are added by using the superposition principle. Here, the choice of N_F and N_S are related to each other, and the commonly used MLFMA sampling scheme [8] can be utilized to select the field points since we are dealing with far-zone interactions. Basically, $2L$ points are spaced regularly along the ϕ direction, and L points are sampled by Gauss-Legendre along the θ direction. Hence, $N_F = 2L^2 = 2(\tau+1)^2$ (where τ is the truncation number and $\tau+1$ is the number of the terms in the summation) can be determined according to the desired relative error (see Section 2.1.4) by

$$L = \max(A, B) + 1, \tag{3.4}$$

where

$$A = ka\sqrt{3} + 2.18(d_0)^{2/3}(ka)^{1/3} \tag{3.5}$$

comes from EBF, and

$$B = 14.14d_0 - 7.17 \tag{3.6}$$

comes from [31]. Then, the method determines N_S as the minimum number that satisfies (i) the third power of an integer, N_{Sk}, and (ii) $N_F < N_S$ to have an under-determined system.

Fig. 3.5 shows how the number of sample points varies for different relative errors, and each line corresponds to different box sizes. N_{Sk} changes drastically

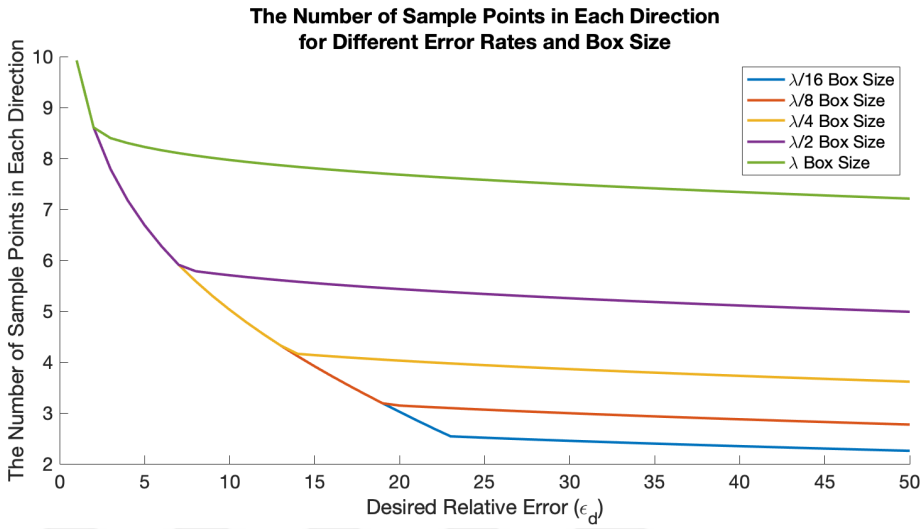


Figure 3.5: The number of sample points in each direction, N_{Sk}, for different relative errors and box sizes.

for small desired error rates and starts to vary slowly after an error level, which changes according to the box size. The number of sample points needed to achieve the desired error level is higher for large boxes, which means that N_S may differ from level to level when the multilevel structure is used.

In a similar fashion, Fig. 3.6 shows the variation of N_{Sk} for different box sizes for the desired relative error. Since the second part of (3.4) is dominant for electrically small boxes, N_{Sk} becomes constant for the desired error rate, which is consistent with [31]. When the box size increases, the first part of (3.4) becomes dominant, and N_{Sk} depends mainly on the box size. One can say that N_S can be roughly approximated according to the desired relative error for small boxes and according to the box size for large boxes. Note that lines are continuous, but the number of sample points is an integer.

Finally, regarding this step of the method (i.e., projection from RWG to dipoles), field mapping is utilized in this thesis since it provides a way to control the error at the projection stage. Alternatively, the projection from RWGs to dipoles can be done based on estimating the total surface current inside the basis sample cube in various ways (a current mapping). These various current mappings may introduce additional spatial information that can significantly improve

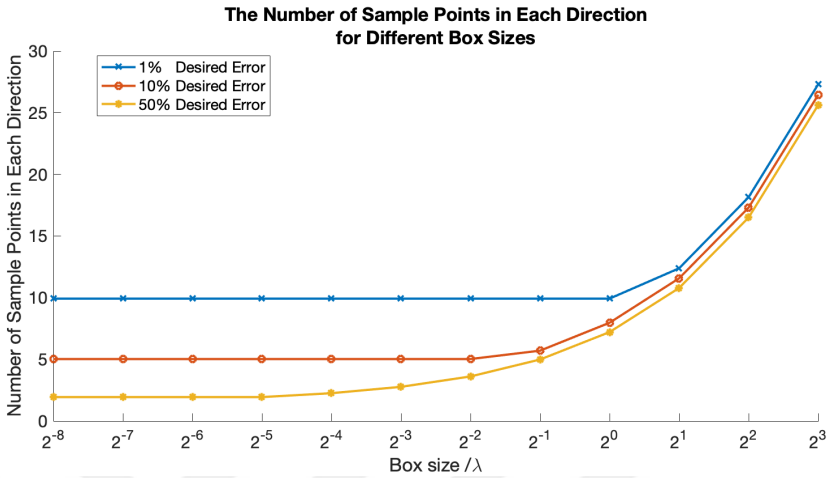


Figure 3.6: The number of sample points in each direction, N_{Sk}, for different relative errors and box sizes.

the efficiency of CNN. However, we lose the capability of controlling the error. On the other hand, since the far-field mapping does not provide any spatial change, the use of CNN does not improve the method drastically (see Chapter 4).

As a final note, the projection used in this thesis provides full matrices to represent the dipoles, which is compatible with the data generated for the training, as discussed in the previous section.

3.2.3 Dipole to Dipole

At this point, we have obtained the current coefficients of dipoles from the projection of RWGs to dipoles (complex-valued 3-D-vectors for each sample point). Next, we extract the real and the imaginary parts of x, y, and z components of these coefficients (as a result, we have $6N_S$ real-values), and they are now the inputs of ANNs. Then, the normalization is applied by dividing the extracted values by the value that has the maximum magnitude since the networks are trained for data with the maximum magnitude of 1 (data is between -1 and 1). The resulting values are fed to ANNs, whose parameters are stored in the memory, to estimate the electric field at the test sample points.

Because of the aforementioned rotational symmetry, the parameters of 316 neural networks are obtained from the 16 trained networks by rotating the vectors at the sample points and rearranging the sample points in the boxes. The parameters of 316 neural networks are stored in the memory to be used in the iterative solver. For each test and basis box combination, the input is fed to the associated neural network.

One of the most significant advantages of ANNs is that it is possible to feed the inputs of multiple levels and multiple directions simultaneously (parallelization), which may improve the complexity of the proposed method.

3.2.4 Projection From Dipoles to RWGs

The final stage of the proposed method is the projection from dipoles to RWG functions, which basically computes the interaction of the electric field on the RWG test functions to find the voltage vector at each iteration of the iterative solver. At the end of the dipole to dipole stage, we obtain the electric field at the test sample points. Now, using these electric fields, we calculate the electric field on the RWG functions as follows:

Consider an RWG function whose location is known. First, we find the 8 nearest test sample points in 3-D (4 points in 2-D) inside the parent box of the RWG. Note that it is possible to have some test sample points closer to the RWG but outside of its parent box. In this case, we still continue with the test sample points only inside its parent box. Fig. 3.7 illustrate two points (namely m and n) on an arbitrary RWG whose nearest sample cubes are the same. By using the electric field vectors of the 8 nearest points, we calculate the electric field as

$$\vec{E}_{RWG}(\vec{r}) = \sum_{k=1}^{8} w_k \vec{E}(\vec{r}_k), \tag{3.7}$$

where $\vec{r}_k$ is the position vector of the k^{th} closest test sample point and w_k is its weight which is used in the linear interpolation.

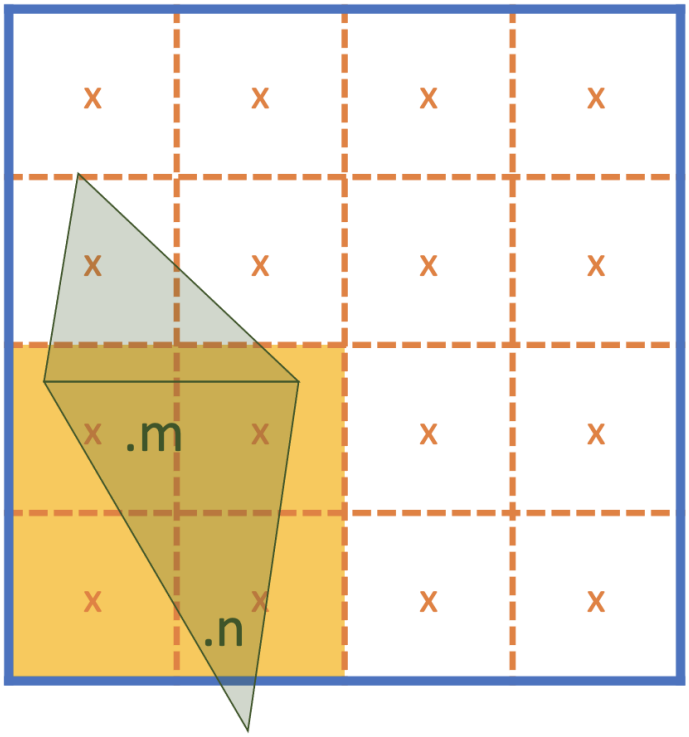


Figure 3.7: Decision of the nearest test sample cubes of an arbitrary RWG basis function.

The closest test sample points and their weights are calculated before the iterative solver and stored in the memory to use at each iteration. Then, the m^{th} element of the voltage vector, which corresponds to the interaction on the m^{th} RWG test function, can be calculated as

$$\int_{S_m} \vec{E}_{RWG}(\vec{r}) \cdot \boldsymbol{t}_m(\vec{r})\, d\vec{r}. \tag{3.8}$$

In this thesis, the integration in (3.8) is approximated with one point integration.

3.3 The Complexity of the Proposed Method

The complexity of the proposed method is mainly determined by the network structure rather than the number of unknowns, N. As discussed in Section 2.2, the number of hidden units is a hyperparameter of ANN, which is determined

according to the desired accuracy and the computational complexity of the neural network. In this section, the complexities of 2-layer FCNN and CNN, used in this thesis, are discussed.

Recall that the method introduces a grid structure $N_{Sk} \times N_{Sk} \times N_{Sk}$ with a total number of sample points being N_{Sk}^3 or N_S and K, which is used in Section 2.2 for the number of inputs of FCNN, is equal to $6N_{Sk}^3$ or $6N_S$. Regarding the computational complexity of the 2-layer FCNN, (2.55) shows the mathematical expression of the structure used in the algorithm. The first layer has KH multiplications and H summations (recall that H denotes the number of hidden units), while the second layer has KH multiplications and K summations. The computational complexity of the activation function of the first layer is proportional to the number of inputs of the function, which is H. Since the activation function of the second layer is linear, it does not introduce additional complexity. Therefore, the computational complexity of the one forward-pass of the 2-layer FCNN is

$$\mathcal{O}(KH). \tag{3.9}$$

In terms of memory complexity, the parameters of one neural network have the complexity of $\mathcal{O}(KH)$, since there are two matrices with KH elements, one vector with H elements, and one vector with K elements.

Regarding the computational complexity of the 2-layer CNN, the grid structure, which the proposed method introduces, is directly used for CNN models, where $c = 6$ in Section 2.2. Each convolution operation has f^3c multiplication. This filter moves along the spatial dimensions, using valid padding. Therefore, the number of convolutions applied is $[(N_{Sk} - f)/s]^3 = o^3$ for each filter. There are n_f filters. Hence, $f^3co^3n_f$ multiplications are needed for the convolution layer. After flattening, there are o^3n_f hidden units and N_{Sk}^3 units at the output layer. As a result, the computational complexity of the one forward-pass of the 2-layer CNN is

$$\mathcal{O}\left((of)^3 cn_f + (oN_{Sk})^3 n_f\right). \tag{3.10}$$

In general, since $f^3c < k^3$, the complexity can be expressed as

$$\mathcal{O}\left((oN_{Sk})^3 n_f\right). \tag{3.11}$$

In terms of memory complexity, the parameters of one neural network have the complexity of $\mathcal{O}(n_f f^3 c + (oN_{Sk})^3 n_f + K)$, since there are n_f matrices with $f^3 c$ elements for the convolution layer, and one matrix with $o^3 n_f N_{Sk}^3$ elements for the second layer, and one vector with K elements as a bias term. However, because $K < n_f f^3 c + (oN_{Sk})^3 n_f$, the memory complexity can be expressed as $\mathcal{O}(n_f f^3 c + (oN_{Sk})^3 n_f)$.

It is important to note that the complexity of projections from RWGs to dipoles and dipoles to RWGs may be dominant if the complexity of the neural network is lower than the complexity of the calculations before and after the neural network. The complexity of the projections depends on the number of basis functions, which is $\mathcal{O}(N)$. However, although increasing the number of basis functions, N, improves the accuracy of the proposed method (due to the fact that projections are computed more accurately), the complexity of the method does not depend on N. This is expected to be one of the major advantages of the method for extremely large scatterers.

Consider MLFMA (see Section 2.1.4), which has the computational complexity of $\mathcal{O}(N \log(N))$. For the case where the complexity of the projections dominates the complexity of the method, the complexities of the proposed method and MLFMA become comparable. On the other hand, for electrically large problems, in which the complexity of the network dominates the overall complexity of the method, the proposed method has adjustable complexity. Hence, the efficiency of the proposed method becomes independent of the number of unknowns introduced by the meshed geometry.

Chapter 4

Numerical Results

Throughout this chapter, the preliminary results for the novel hierarchical ML-based method to solve electromagnetic scattering problems are presented. First, FCNN structures with the different numbers of hidden units for the 3-level tree structures are discussed. Then, preliminary results of CNN structure are provided. Finally, the results for FCNNs are extended to a 4-level tree structure.

Before the details of the numerical results, three error definitions are needed to be discussed. Equation (4.1), (4.2), (4.3) and (4.4) are the definitions of the mean absolute error (MAE), the mean squared error (MSE), root mean squared error (RMSE), and the mean absolute percentage error (MAPE), respectively. Define a reference vector $\boldsymbol{a} = [a_x, a_y, a_z]$ and an estimated vector as $\boldsymbol{b} = [b_x, b_y, b_z]$, where a_x, a_y, a_z, b_x, b_y, and b_z are complex values. Error vector can be defined as $\boldsymbol{e} = [e_x, e_y, e_z]$ where $e_i = |a_i - b_i|$ with $i = x, y, z$ and $|v| = \sqrt{(Real\{v\})^2 + (Imag\{v\})^2}$.

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|\boldsymbol{e}_n| \tag{4.1}$$

$$MSE = \frac{1}{N}\sum_{n=1}^{N}|\boldsymbol{e}_n|^2 \tag{4.2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|\boldsymbol{e}_n|^2} = \sqrt{MSE} \tag{4.3}$$

$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\frac{|\boldsymbol{e}_n|}{|\boldsymbol{a}_n|}. \tag{4.4}$$

Note that (4.1)–(4.4) can be used for real-valued data. During the network trainings, error (4.1)–(4.4) are calculated for real values and are labelled as real, on the other hand if they are used for complex values, they are labelled as complex.

4.1 FCNN Structures

In this section, the preliminary results for FCNN structures with different numbers of hidden units and the 3-level tree structure are discussed for a plane-wave scattering from a PEC sphere and a PEC cube. Scattering from these objects are also solved with different methods to compare the accuracy and the efficiency of the proposed method. Additional to MoM and MLFMA, we introduce the direct Green's function computation (DGFC), which uses the same tree structure as our proposed method. However, in this method the group-by-group interactions are computed directly with the dyadic Green's function in each iteration rather than using ANNs. Although it is expected to be slower than our proposed method, it helps to evaluate the error introduced by the projections from RWGs to dipoles and dipoles to RWGs.

In all calculations, the operating frequency is 300MHz ($\lambda = 1$m). Sizes of the PEC objects are selected as 1m (λ), and the triangular mesh elements have an edge length of 50mm ($\lambda/20$). The meshed surface of the sphere with a diameter

of λ has 2770 triangles and 4155 RWG functions, whereas the meshed surface of the cube with a λ edge length contains 4800 triangles and 7200 RWG functions.

All objects are illuminated by a linearly polarized plane wave with an amplitude of 1V/m. The sphere is illuminated by an x-polarized plane wave ($[1, 0, 0]$) propagating along the $+z$ direction ($[0, 0, 1]$), while the cube is illuminated from different angles. The first, the second, and the third excitations illuminate the cube from its face (Fig. 4.1 (b)), from its edge (Fig. 4.1 (c)) and from its vertex (Fig. 4.1 (d)), respectively. The resulting electric fields at the far-zone are calculated at $\phi = 0$ plane where θ varies between 0 to 360 (see blue lines in Fig. 4.1) for the first and the second illuminations, and at $\phi = 45^\circ$ plane for the third illumination.

For all methods, generalized minimal residual (GMRES) [37] is used as an iterative solver without any preconditioning, with the same tolerance of 10^{-2}, and with a restart value of 100. The maximum number of iterations is set to 1000.

Finally, the far-zone interactions are defined only at the third level with $a_3 = \lambda/4$ for the 3-level tree structure utilized for the λ-sized objects. The number of sample points is determined according to the desired relative error for the projection from RWGs to dipoles using (3.4), and fixed as 20%, which corresponds to $N_{Sk} = 4$ and hence $N_S = 64$. Since the number of sample points (N_S) and the box size (a_l) are known, the training dataset can be generated. The training data contains 100k randomly generated scenarios for this case and 10% of the dataset is used as the validation dataset.

The 2-layer FCNNs with 64 and 128 hidden units with ReLU activation function are utilized. The numbers of parameters of FCNNs are given in Table 4.1. The number of parameters is linearly proportional to the number of hidden units for FCNN structure.

The Adam optimizer [34] is utilized to optimize the loss function (MAE in this thesis). 1000 epochs with $5 * 10^{-5}$ learning rate and 128 batch-size are completed to achieve the convergence. Fig. 4.2 (a), 4.2 (b), and 4.2 (c) show how MAE,

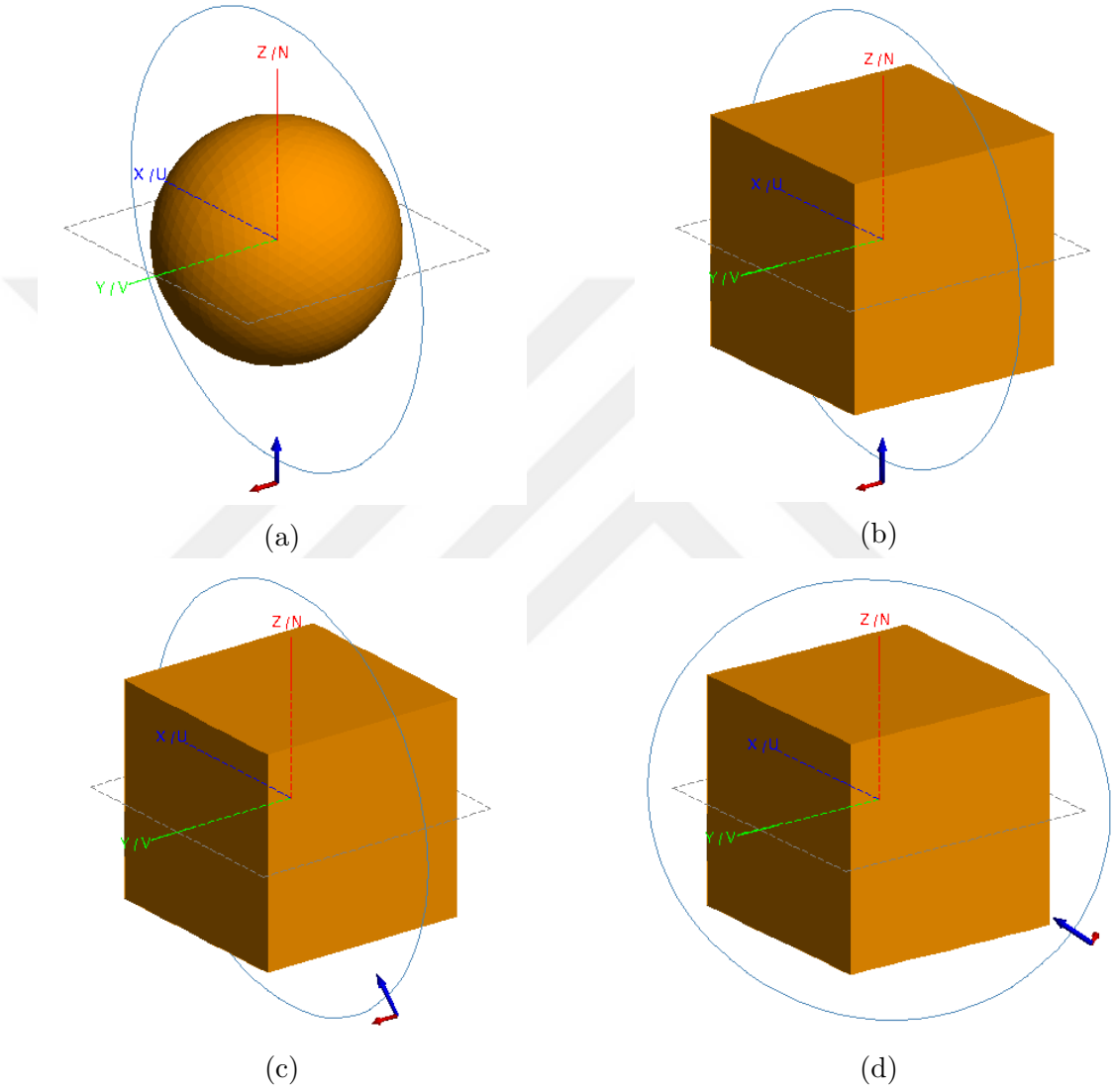
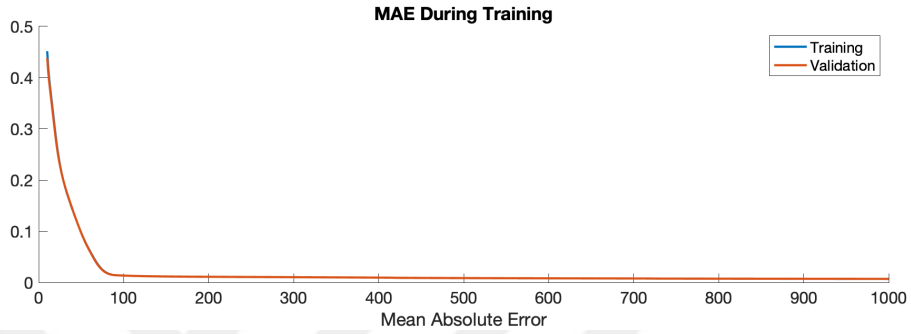
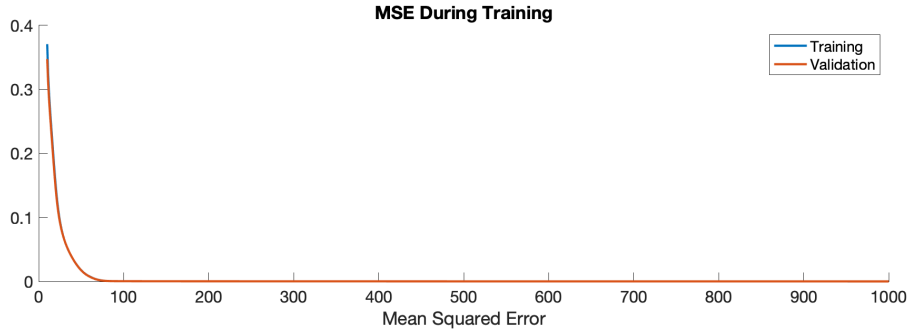


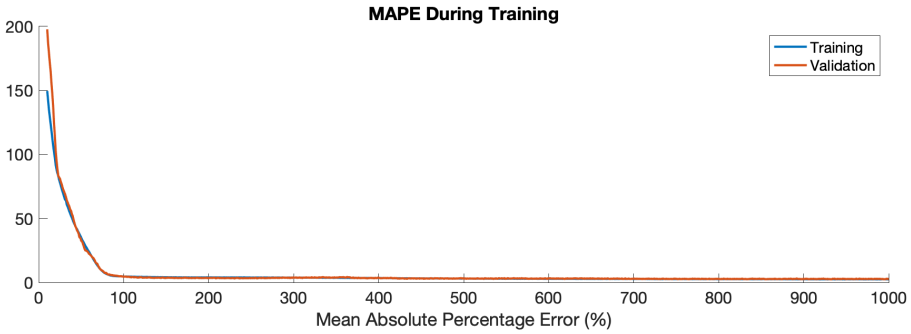
Figure 4.1: Scattering problems solved in this thesis. (a) A PEC sphere illuminated by a plane wave with the propagation vector of $[0, 0, 1]$ and the polarization vector of $[1, 0, 0]$, (b) a PEC cube illuminated from its face by a plane wave with the propagation vector of $[0, 0, 1]$ and the polarization vector of $[1, 0, 0]$, (c) a PEC cube illuminated from its edge by a plane wave with the propagation vector of $\left[\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right]$ and the polarization vector of $[0, 1, 0]$, (d) a PEC cube illuminated from its vertex by a plane wave with the propagation vector of $\left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right]$ and the polarization vector of $\left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0\right]$.



(a)



(b)



(c)

Figure 4.2: Behaviours of MAE, MAPE and MSE on the training and validation datasets during the training of the FCNN with 64 hidden units to estimate the dyadic Green's function for the translation distance of $[0, 0, 2]$, which corresponds to a one-box-buffer case along the $+z$-direction.

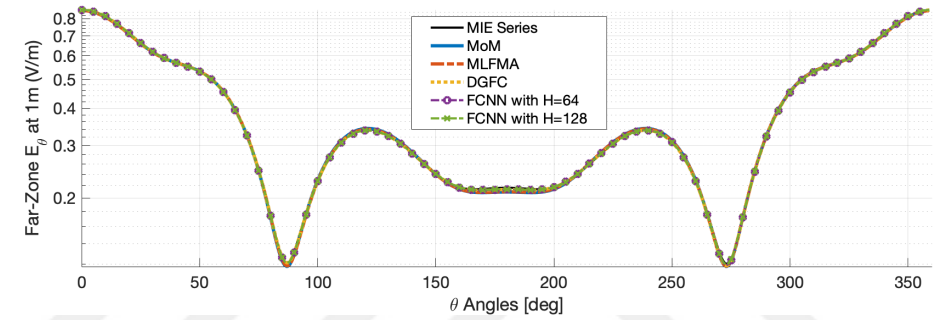
Table 4.1: Number of parameters of FCNN structures with $H = 64$ and $H = 128$ and $a_l = \lambda/4$.

	$H = 64$	$H = 128$
Layer 1	24,640	49,280
Layer 2	24,960	49,536
Total	49,600	98,816

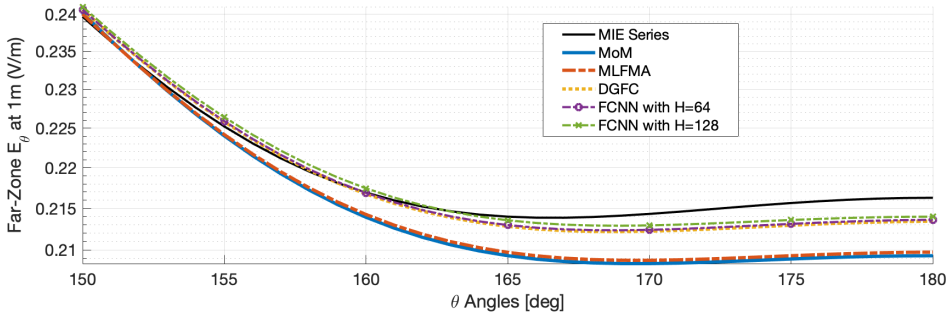
MSE and MAPE behave, respectively, on the training and the validation datasets during the training of the FCNN with 64 hidden units to estimate the dyadic Green's function for the translation distance $[0, 0, 2]$. The first 50 epochs are not included when the figures are prepared to ignore the instant drop at the beginning. The figures clearly show that the convergence is achieved.

Training is completed for 16 networks to obtain all translation distances, and Table B.1 and Table B.2 show the error of the trained FCNN with 64 and 128 hidden units, respectively, on the test sets for different translation directions. The test sets contain $10,000$ instances, which are randomly generated from the same distribution of the training datasets. The highest error is obtained from the network trained for $[0, 0, 2]$, which corresponds to the one-box-buffer case. Then, the error generally decreases for larger translation distances. Since the main contribution of the far-zone interactions are coming from the one-box-buffer case, the error coming from that network may affect the accuracy of the method. MAPE of the one-box-buffer case (which corresponds to the worst trained network) is smaller than 0.5%. Another observation from the tables is the RMS error defined for the real numbers (which corresponds to the case where the real and imaginary part of the complex values are separated) is larger than the RMS error obtained from the complex numbers. Since the commonly used neural networks are not capable of dealing with the complex numbers, errors defined in the real values are used to train the network. However, the error for the complex values, which are commonly used in electromagnetics, is lower than the error for the real values.

Our first example is the scattering from a λ-sized PEC sphere where all details



(a)



(b)

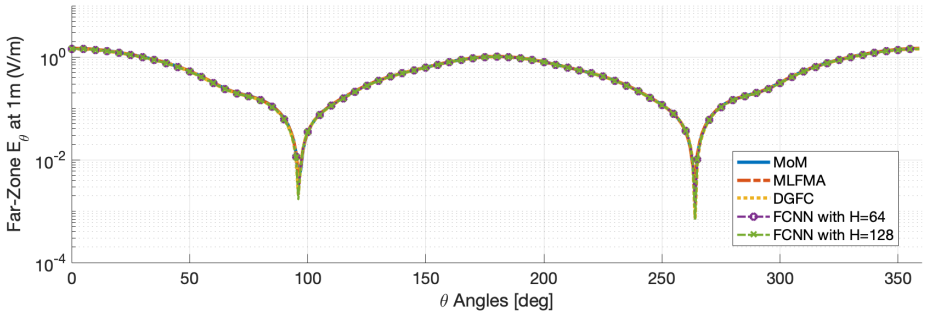
Figure 4.3: Magnitude of $\vec{E}_\theta$ (co-polarization) results where θ varies (a) between 0° and 360°, and (b) between 150° and 180° (backscattering) at the far-zone obtained for the PEC sphere by MoM, MLFMA, DGFC and the novel hierarchical ML-based method with 2-layer FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure.

are given before. The Mie series solution, the analytical solution of the problem, is used as a reference solution in all sphere illuminations. Fig. 4.3 shows the magnitude of $\vec{E}_\theta$ (co-polarization) at the far-zone for different methods. Fig. 4.3 (a) shows a good fit overall, and Fig. 4.3 (b) shows the zoomed version of Fig. 4.3 (a) to see the details around the backscattering. A slight difference between the methods is observed. MoM and MLFMA solutions are closer to each other while DGFC and the novel hierarchical ML-based method are closer to each other. For this particular problem, MoM and MLFMA solutions are slightly more deviated from the Mie series solution. The current distributions obtained from different methods, which are shown in Fig. D.1, are also compatible with the observation that the and hierarchical ML-based method follow each other, while MoM and MLFMA results follow each other. The number of hidden units affect the accuracy of the proposed method but not as dominant as the projections. Therefore, one can say that error introduced at the projections from RWGs to dipoles and dipoles to RWGs dominate the error. Note that the error at the projection from RWGs to dipoles can be controlled. These results are also consistent with MAPE and MAE results provided in Table 4.2.

Table 4.2: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the PEC sphere by MoM, MLFMA, DGFC and FCNN with $N_S = 64$ and $N_S = 128$ for the 3-level tree structure with respect to the Mie series solution.

	MAPE	MAE
MoM	0.7109	0.0018
MLFMA	0.7244	0.0020
DGFC	0.5413	0.0015
FCNN 64	0.5573	0.0016
FCNN 128	0.5238	0.0015

Table 4.3 shows the time comparison for different methods. Near-field calculations take longest for MoM since all interactions are calculated element-by-element. The time spent for near-field interactions is less for other methods since the number of near-field interactions decreases by utilizing the multilevel scheme. The time spent on the iterative solver is the most significant part that needs to



(a)

Figure 4.4: Magnitude of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the cube illuminated from its face by MoM, MLFMA, DGFC and the novel hierarchical ML-based method with 2-layer FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure.

be discussed. The number of iterations increases when DGFC and FCNNs with 64 and 128 hidden units are used, but the time spent per each iteration decreases. As a result, the proposed method's time efficiency is comparable with MLFMA. Note that our codes are not optimized and the parallelization of the method has not been implemented yet. We expect a significant improvement in efficiency once the proposed method is parallelized (i.e., feeding the inputs of the neural networks for different levels and different translation distances simultaneously).

Table 4.3: Time comparison of MoM, MLFMA, DGFC and FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure to solve the scattering problem of the PEC sphere.

	MoM	MLFMA	DGFC	FCNN 64	FCNN 128
Near-Field Calc. (Sec)	49.17	14.97	12.80	9.56	9.53
Iterative Solver (Sec)	105.30	37.09	270.31	26.19	33.33
Number of Iterations	54	57	62	62	63
Time per Iteration (Sec)	1.95	0.65	4.35	0.42	0.52
Overall (Sec)	158.02	56.44	306.33	53.42	61.12

Another canonical shape is the λ-sized PEC cube which is illuminated by three

different excitations as defined before. This section contains only the first illumination and the results for other two illuminations can be found in Appendix C. The magnitude of $\vec{E}_\theta$ (co-polarization) results at the far-zone for the scattering problem of the PEC cube illuminated from its face are shown in Fig. 4.4. Similar to the PEC sphere problem, MoM and MLFMA results are closer to each other, and hierarchical ML-based method and DGFC are closer to each other. The errors for each method with respect to the MoM solution (used as a reference) are given in Table 4.4. Also, see Fig. D.2 to compare the current distributions on the PEC cube when it is illuminated from its face.

Table 4.4: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its face by MoM, MLFMA, DGFC and FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA	1.1587	0.0014
DGFC	2.9472	0.0116
FCNN 64	2.7750	0.0116
FCNN 128	3.0509	0.0118

Note that the cube contains more RWG basis functions than that of the sphere, which affects the time spent on different steps. Table 4.5 shows the time comparison of different methods to solve the scattering problem of the PEC cube illuminated from its face. Near-field calculations take longer time as expected. The number of iterations for different methods are very close, and hence the time per iteration provides a better comparison. The time per iteration is significantly higher for MLFMA compared to the proposed method. Note that the time spent per iteration does not multiply with 2, when we switch from FCNN with $H = 64$ to FCNN with $H = 128$, which shows that time complexity of the projections dominate the time complexity of the method.

Table 4.5: Time comparison of MoM, MLFMA, DGFC and FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure to solve the scattering problem of the PEC cube illuminated from its face.

	MoM	MLFMA	DGFC	FCNN 64	FCNN 128
Near-Field Calc. (Sec)	106.31	31.06	28.51	21.11	20.90
Iterative Solver (Sec)	585.43	203.58	688.40	125.66	138.01
Number of Iterations	135	136	136	137	136
Time per Iteration (Sec)	4.33	1.49	5.06	0.91	1.01
Overall (Sec)	695.23	239.73	752.48	174.56	187.05

4.2 CNN Structure

Since CNN is a commonly used network for the known grid-like topologies, the preliminary results for CNN are also obtained. The same data sets are used for the same geometries provided in the previous section; only the ML structure is changed.

The 2-layer CNN $n_f = 64$ filters where the filter size is $2 \times 2 \times 2 \times 6$, the stride 2, and ReLU activation function is utilized. The number of parameters of the CNN model used in this thesis is given in Table 4.6. Note that the number of parameters at the convolution layer (first layer) is significantly smaller than the number of parameters at the first layer of the FCNN models since convolution layer shares the weights. On the other hand, the number of parameters of the second layer is significantly larger than the FCNN models, since there are 512 hidden units after flattening the output of the convolution layer. For comparison purposes, we set the number of filters to 64 and compare CNN model with the FCNN model, which contains 64 hidden units. This comparison may be done with different methods such as setting the number of total parameters the same, and adding additional fully connected layer at the end of this model. Also, The same training procedure is completed as that of FCNN to obtain all translation distances. Table B.3 shows the error of the trained CNN. The highest MAE is

again obtained from the one-box-buffer case and MAPE is around 1.3% at this case, which shows that the accuracy of the CNN model is around two FCNN models discussed in the previous section.

Table 4.6: Number of parameters of CNN structures with 64 filters and $a_l = \lambda/4$.

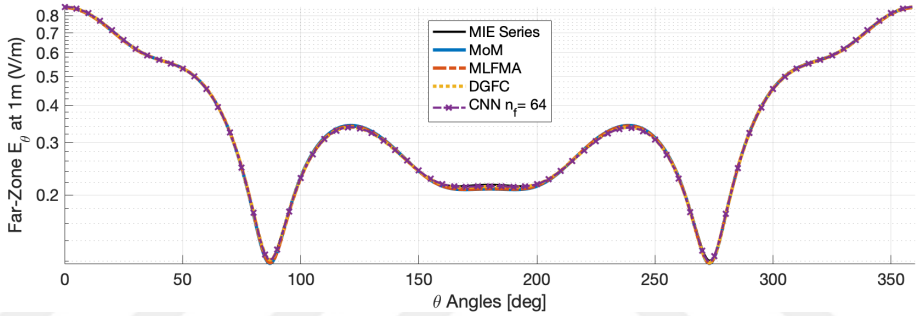
	CNN with 64 filters
Layer 1	3,136
Layer 2	196,992
Total	200,128

Fig. 4.5 shows the magnitude of $\vec{E}_\theta$ (co-polarization) component at the far-zone obtained from different methods for a PEC cube illuminated from its face and a PEC sphere. Similar overlaps are obtained also for the CNN structures. Table 4.7 shows the MAE and MAPE of the far-field results obtained for the PEC sphere with respect to the Mie series solution, and Table 4.8 shows the MAE and MAPE of the far-field results obtained for the cube illuminated from its face, where MoM is used as a reference solution.

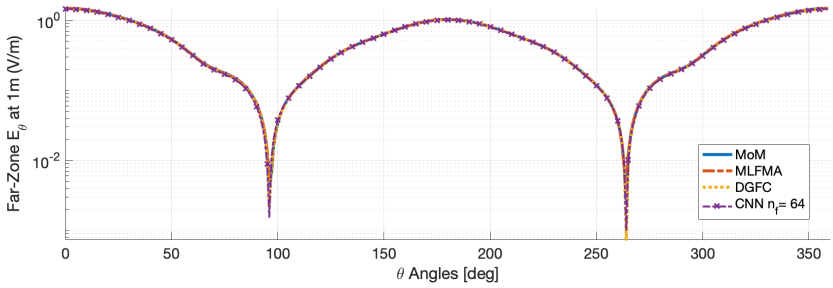
Table 4.7: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the PEC sphere by MoM, MLFMA, DGFC and CNN with $n_f = 64$ for the 3-level tree structure with respect to the Mie series solution.

	MAPE	MAE
MoM	0.7109	0.0018
MLFMA	0.7244	0.0020
DGFC	0.5413	0.0015
CNN	0.5472	0.0016

Regarding the time complexity, since there are more multiplications and more parameters, the elapsed time of CNN is higher than FCNNs and MLFMA. The time per iteration and the number of iterations increase, which means that the number of parameters are needed to be selected wisely to obtain a comparable time complexity with other methods.



(a)



(b)

Figure 4.5: Magnitude of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for (a) the PEC sphere, and (b) the PEC cube illuminated from its face by MLFMA, DGFC and the novel hierarchical ML-based method with 2-layer CNN with $n_f = 64$ for the 3-level tree structure.

Table 4.8: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the cube illuminated from its face by MLFMA, DGFC and CNN with $n_f = 64$ for the 3-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA	1.1587	0.0014
DGFC	2.9472	0.0116
CNN	3.4690	0.0121

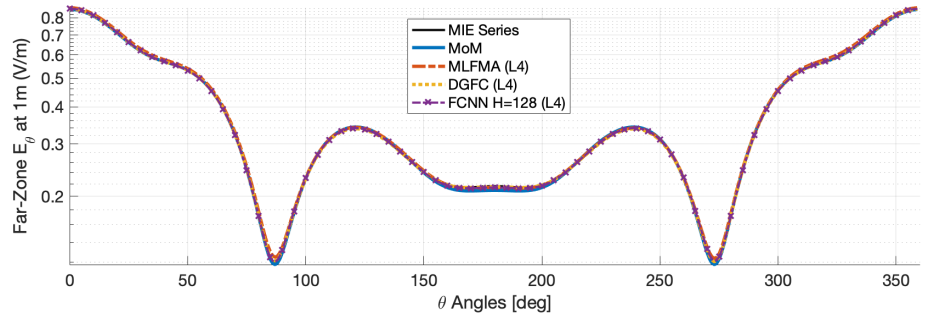
Table 4.9: Time comparison of MLFMA and CNN with $n_f = 64$ for the 3-level tree structure to solve the scattering problem of the PEC sphere (S) and the PEC cube illuminated from its face (C).

	MLFMA(S)	CNN (S)	MLFMA(C)	CNN (C)
Near-Field Calc. (Sec)	14.97	12.90	31.06	29.44
Iterative Solver (Sec)	37.09	66.20	203.58	248.76
Number of Iterations	57	63	136	141
Time per Iteration (Sec)	0.65	1.05	1.49	1.76
Overall (Sec)	56.44	106.52	239.73	320.02

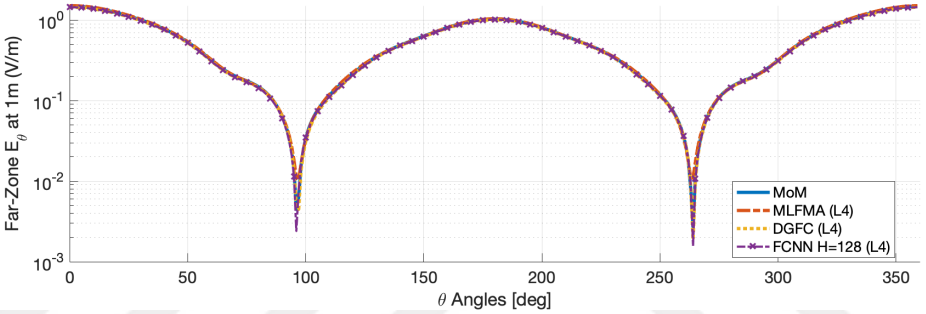
Finally, CNN is used since the grid structure is introduced in the proposed method. However, the projection from RWGs to dipoles also affects the performance of the CNN structure. Projection from RWGs to dipoles used in this thesis provides homogeneously distributed basis sample points, which does not provide the current flow on the surface. The current mapping may increase the performance of the CNN structures.

4.3 FCNN Structure for the 4-Level Tree Structure

Finally, the 3-level tree structure is extended to the 4-level tree structure to see the affect of the multilevel scheme. The same geometries with the same maximum dimension, λ, are used in this section. Since we increase the number of levels of the tree structure, the additional far-zone interactions are introduced at the fourth level with $a_4 = \lambda/8$. The number of sample points are determined according to the same desired relative error for the projection from RWGs to dipoles as that of the 3-level tree structure (which is 20%) using (3.4). It corresponds to $N_{Sk} = 4$ and hence $N_S = 64$. Then, new training data for the box size of $\lambda/8$, which contains 100k data instances, is generated and 10% of the dataset is used as the validation dataset. The 2-layer FCNN that has 128 hidden units with ReLU activation function is utilized. The number of parameters of FCNN is the same as the one introduced in Table 4.1. As a result, for the calculation of the far-zone at the third and fourth levels, FCNNs with $H = 128$ are utilized. The Adam optimizer [10] is utilized to optimize the loss function (MAE in this thesis) with 1000 epochs, $5 * 10^{-5}$ learning rate and 128 batch-size to achieve the convergence. Similar to the previous sections, the training is completed for 16 networks to obtain all translation distances, and Table B.4 shows the error of the trained FCNNs with 128 hidden units for the box size of $\lambda/8$ on the test sets for different translation directions. Moreover, solutions of MLFMA and DGFC are also obtained for the 4-level tree structure. Fig. 4.6 shows the magnitude of $\vec{E}_\theta$ component at the far-zone obtained by different methods with the 4-level tree structure for (a) the PEC sphere, and (b) the PEC cube illuminated from its face. Table 4.10 shows MAE and MAPE of the far-field results obtained for the PEC sphere with respect to the Mie series solution. Similarly, Table 4.11 shows MAE and MAPE of the far-field results obtained for the PEC cube illuminated from its face with respect to the MoM solution. The accuracy of the proposed method increases when the number of levels at the tree structure increases, since the closer interactions are calculated with closer sample points, while the accuracy of MLFMA decreases. Also, see Fig. D.5 and D.6 to compare the current distributions on the PEC



(a)



(b)

Figure 4.6: Magnitude of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for (a) the PEC sphere, and (b) the PEC cube illuminated from its face by MoM, MLFMA, DGFC and the novel hierarchical ML-based method with 2-layer FCNN with $H = 128$ for the 4-level tree structure.

sphere and the PEC cube, respectively.

Regarding the timing, the time per iteration increases from 0.52 to 0.75 for the PEC sphere, and from 1.01 to 1.10 for the PEC cube when the number of levels is changed from 3 to 4 for the proposed method with the same number of hidden units. This increase is expected since the number of networks to be fed increases. But note that, the incrementation is less for the cube, which contains more basis functions. One can say that the complexity coming from the projections dominates the overall complexity of the method for the sphere problem, and the complexity of networks dominates the complexity of the method for the cube problem. This provides promising results for larger problems that contains more unknowns. Also note that, the implementation of the method is parallelizable but parallelization has not been completed yet. After the parallelization of the

Table 4.10: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the sphere by MoM, MLFMA, DGFC and FCNN with $H = 128$ for the 4-level tree structure with respect to the Mie series solution.

	MAPE	MAE
MoM	0.7109	0.0018
MLFMA (4-Level)	1.0859	0.0044
DGFC (4-Level)	0.3931	0.0012
FCNN (4-Level)	0.3442	0.0012

Table 4.11: MAPE and MAE of $\vec{E}_\theta$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its face by MLFMA, DGFC and FCNN with $H = 128$ for the 4-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA (4-Level)	6.8157	0.0151
DGFC (4-Level)	4.0407	0.0148
FCNN (4-Level)	4.2715	0.0142

feeding the networks, the time complexity is expected to improve significantly.

Table 4.12: Time comparison of MoM, MLFMA, DGFC and FCNN with $H = 128$ for the 4-level tree structure to solve the scattering problem of the PEC sphere.

	MoM	MLFMA	DGFC	FCNN 128
Near-Field Calc. (Sec)	49.17	3.25	2.90	3.15
Iterative Solver (Sec)	105.30	16.41	972.99	45.49
Number of Iterations	54	72	64	60
Time per Iteration (Sec)	1.95	0.22	15.20	0.75
Overall (Sec)	158.02	22.60	1006.89	85.26

Table 4.13: The time comparison of MoM, MLFMA, DGFC and FCNN with $H = 128$ for the 4-level tree structure to solve the scattering problem of the PEC cube illuminated from its face.

	MoM	MLFMA	DGFC	FCNN 128
Near-Field Calc. (Sec)	106.31	5.899	5.26	6.65
Iterative Solver (Sec)	585.43	53.237	2389.68	149.68
Number of Iterations	135	149	137	136
Time per Iteration (Sec)	4.33	0.35	17.44	1.10
Overall (Sec)	695.23	62.79	2444.10	193.44

Chapter 5

Conclusion and Future Research Directions

In this thesis, we proposed a novel hierarchical ML-based method to solve electromagnetic scattering problems and showcased the promising potential of the method via preliminary results. The proposed method aims to provide accurate results for EM problems with the aid of ML techniques by utilizing a multilevel tree structure, while having high efficiency and an integrated error control scheme.

Using the tree structure and the one-box buffer scheme, boxes at the tree structure are determined as either near-zone or far-zone boxes depending on the distance between the given box pairs. The near-zone interactions, which provide a significant contribution in the matrix-vector-products within an iterative solution, are calculated directly, while the proposed method provides an efficient approximation of the Green's function to be used in the calculation of the far-zone interactions in a group-wise manner. To be able to utilize ANNs regardless of the given discretization of an arbitrary geometry, unifomly distributed virtual Hertzian dipoles are introduced across the boxes within the tree structure. The networks provide a box-to-box estimation of the Green's function from the dipoles at the basis box to test sample points. For each unique translation direction, a neural network is needed to be trained, and then the network can be used for

rotationally symmetric test and basis box pairs with simple transformations.

To facilitate the mapping from and to the dipoles at the sample points, the proposed method also introduces the projection from RWG basis functions to dipoles as the preprocessing of the neural network inputs and the projection from dipoles to RWG test functions as the postprocessing of the neural network outputs. The projection from RWG basis functions (or any other basis function) to the dipoles is based on a far-field mapping by projecting the far-field patterns of the RWG basis functions onto the far-field pattern of the dipoles. This stage helps to decrease the number of unknowns from the number of basis functions to the number of dipoles and also decreases the dependency of the computational complexity of the proposed method to the number of basis functions. More significantly, the error introduced at this projection is controlled by selecting the number of basis sample points and the field points. Note that because this stage dominates the error of the proposed method, the error of the overall method can be controlled via this stage. Finally, the projection from dipoles to RWG test functions is the step where the fields are interpolated inside the test box, and MVM is calculated.

The networks used in the iterative solver can be trained offline. Therefore, training time and complexity of the neural networks do not contribute to the time and complexity of the method in runtime. More significantly, once the networks to be used in the proposed method are trained, they are available to be used multiple times for different excitations, and applicable to arbitrary shaped scatterers that have the same box sizes within its tree structure.

Preliminary results show that the accuracy and the efficiency of the proposed method are comparable with MLFMA, which is one of the most widely used methods in CEM. Note that the presented results are only preliminary and the efficiency of implementation can be greatly improved. A significant factor in future improvements will be the parallelization of the forward computing of the networks, which is possible since the networks at different levels and different translation directions are entirely independent of each other. Especially using state-of-the-art GPU architectures for the parallelization is a promising first step

in our near-future research directions.

Theoretically, the proposed algorithm is expected to eliminate the low-frequency (LF) breakdown (which is a known problem in MLFMA) since the used ML networks approximate the Green's function directly without any diagonalization (which is the culprit in MLFMA). As another future research direction, the proposed algorithm will be applied to canonical geometries at frequencies that typically exhibit LF breakdown of MLFMA.

In this thesis, we implemented the proposed method in conjunction with the EFIE. However, the proposed method can be extended to other SIEs (i.e., magnetic field IE and combined field IE) and VIEs. Also note that the proposed method is independent from the selection of the basis function, thanks to the introduced mappings to and from the virtual dipoles.

Another important future research direction can be the improvement of ANNs by tuning the hyperparameters such as number of layers, number of hidden layers, number of filters, activation functions, etc. In this thesis, basic network structures are utilized, and their hyperparameters are selected so that we can work with relatively small networks. However, it is important to provide a wide range of hyperparameters to investigate the complexity and the accuracy of the proposed method. Especially, the CNN structure used in this thesis needs improvements to achieve a comparable computational complexity with that of FCNN. Note that CNN structures may provide better accuracy and/or less complexity with the provided current mapping on the scatterer (changing the projection from RWG to dipole). Therefore, another future work can be the investigation of different mappings from RWGs to dipoles, such as current mapping, to compare the results of the proposed method. However, the training dataset needs to be changed to mimic the data provided by different projections since the available datasets are generated under the assumption that the dipoles are independent.

Rather than the ANN structures provided in this thesis, the complex-valued networks and networks that contain transpose CNNs can also be utilized. The complex-valued networks may improve the computational complexity since the

relation between real and imaginary parts of the complex values is lost when they are given as separate inputs to a real-valued network. Transpose CNNs may also improve the complexity by utilizing the grid structure of the neural network outputs.

To sum up, we proposed a novel hierarchical ML-based method to solve EM problems and showed how promising the method is in terms of efficiency and accuracy with many future research directions, including parallelization, hyperparameter investigation, extension to other integral equations, basis functions, and ANN structures.

Bibliography

[1] C. Balanis, *Advanced Engineering Electromagnetics*. Wiley, 2012.

[2] O. Ergül and L. Gürel, *The Multilevel Fast Multipole Algorithm for Solving Large-Scale Computational Electromagnetics Problems*. 2014.

[3] W. Chew, E. Michielssen, J. M. Song, and J. M. Jin, *Fast and Efficient Algorithms in Computational Electromagnetics*. USA: Artech House, Inc., 2001.

[4] W. Gibson, *The Method of Moments in Electromagnetics*. CRC Press, 2007.

[5] R. F. Harrington and J. L. Harrington, *Field Computation by Moment Methods*. USA: Oxford University Press, Inc., 1st ed., 1996.

[6] B. A. Cipra, "The best of the 20th century: Editors name top 10 algorithms," May 2000. [Online].

[7] N. Engheta, W. Murphy, V. Rokhlin, and M. Vassiliou, "The fast multipole method (FMM) for electromagnetic scattering problems," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 6, pp. 634–641, 1992.

[8] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 3, pp. 7–12, 1993.

[9] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 10, pp. 1488–1493, 1997.

[10] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., 2010.

[11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[13] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," vol. 8689, November 2013.

[14] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *ECCV*, 2016.

[15] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," vol. 14, pp. 1532–1543, Jan. 2014.

[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, July 2016.

[17] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology.," *Proceedings of the National Academy of Sciences*, vol. 87, no. 23, pp. 9193–9196, 1990.

[18] O. Mangasarian, N. Street, and W. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, Feb. 1970.

[19] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, p. 81–173, Feb. 2011.

[20] B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell, "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 1, pp. 7–18, 2014.

[21] J. Tak, A. Kantemur, Y. Sharma, and H. Xin, "A 3-D-Printed W-Band Slotted Waveguide Array Antenna Optimized Using Machine Learning," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 11, pp. 2008–2012, 2018.

[22] A. M. Alzahed, S. M. Mikki, and Y. M. M. Antar, "Nonlinear mutual coupling compensation operator design using a novel electromagnetic machine learning paradigm," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 5, pp. 861–865, 2019.

[23] M. Salucci, L. Tenuti, G. Oliveri, and A. Massa, "Efficient prediction of the em response of reflectarray antenna elements by an advanced statistical learning method," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 8, pp. 3995–4007, 2018.

[24] D.-J. Yun, I. I. Jung, H. Jung, H. Kang, W.-Y. Yang, and I. Y. Park, "Improvement in computation time of the finite multipole method by using k-means clustering," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 9, pp. 1814–1817, 2019.

[25] B. Karaosmanoğlu and O. Ergül, "Acceleration of MLFMA Simulations Using Trimmed Tree Structures," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 1, pp. 356–365, 2021.

[26] S. Qi, Y. Wang, Y. Li, X. Wu, Q. Ren, and Y. Ren, "Two-dimensional electromagnetic solver based on deep learning technique," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 5, pp. 83–88, 2020.

[27] H. M. Yao, L. J. Jiang, and Y. W. Qin, "Machine learning based method of moments (ML-MoM)," in *2017 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, pp. 973–974, 2017.

[28] J.-J. Sun, S. Sun, Y. P. Chen, L. Jiang, and J. Hu, "Machine-learning-based hybrid method for the multilevel fast multipole algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 12, pp. 2177–2181, 2020.

[29] I. T. Rekanos, "Inverse scattering of dielectric cylinders by using radial basis function neural networks," *Radio Science*, vol. 36, no. 5, pp. 841–849, 2001.

[30] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, 1982.

[31] M. Kalfa, V. B. Ertürk, and O. Ergül, "Error analysis of mlfma with closed-form expressions," *IEEE Transactions on Antennas and Propagation*, pp. 1–1, 2021.

[32] J. Song and W. Chew, "Multilevel fast multipole algorithm for solving combined field integral equation of electromagnetic scattering," *Microwave and Optical Technology Letters*, vol. 10, pp. 14 – 19, 09 1995.

[33] L. Jiang and W. Chew, "Low-frequency fast inhomogeneous plane-wave algorithm (lf-fipwa)," *Microwave and Optical Technology Letters*, vol. 40, pp. 117 – 122, 01 2004.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[35] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *CoRR*, vol. abs/1206.5533, 2012.

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[37] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.

Appendix A

Dyadic Green's Function

The compact form of the dyadic Green's function can be written in terms of the unit dyad $\overline{\overline{I}}$ as

$$\overline{\overline{G}}(\vec{r},\vec{r}\,') = \left(1 + \frac{1}{k^2}\nabla\nabla\cdot\right) g(\vec{r},\vec{r}\,')\overline{\overline{I}}. \tag{A.1}$$

Using the identity, $\nabla\cdot\left(\psi\overline{\overline{I}}\right) = \nabla\psi\cdot\overline{\overline{I}} = \nabla\psi$, (A.1) becomes

$$\overline{\overline{G}}(\vec{r},\vec{r}\,') = \left(\overline{\overline{I}} + \frac{1}{k^2}\nabla\nabla\right) g(\vec{r},\vec{r}\,'). \tag{A.2}$$

Let $R = |\vec{r} - \vec{r}\,'|$ be the distance between the observation and the source point, then

$$g(\vec{r},\vec{r}\,') = g(R) = \frac{e^{ikR}}{4\pi R}, \tag{A.3}$$

and we can find $\nabla g(R)$ as

$$\nabla g(R) = \frac{d}{dR}g(R)\nabla R = \left(ik - \frac{1}{R}\right) g(R)\nabla R = \left(ik - \frac{1}{R}\right) g(R)\hat{R} \tag{A.4}$$

where

$$\hat{R} = \frac{\vec{r} - \vec{r}\,'}{|\vec{r} - \vec{r}\,'|} \tag{A.5}$$

is the unit vector along the $\vec{R} = (\vec{r} - \vec{r}\,')$ direction. Then, we can find $\nabla\nabla g(R)$ with the help of

$$\nabla\hat{R} = \nabla\left(\frac{\vec{R}}{R}\right) = \frac{\nabla(\vec{R})}{R} + \vec{R}\nabla\left(\frac{1}{R}\right) = \left(\overline{\overline{I}} - \hat{R}\hat{R}\right)\frac{1}{R} \tag{A.6}$$

as follows:

$$\nabla\nabla g(R) = \nabla\left[\left(ik - \frac{1}{R}\right) g(R)\right]\hat{R} + \left(ik - \frac{1}{R}\right) g(R)\nabla\hat{R} \tag{A.7}$$

$$= \left[\frac{1}{R^2} g(R)\hat{R} + \left(ik - \frac{1}{R}\right)^2 g(R)\hat{R}\right]\hat{R} + \left(ik - \frac{1}{R}\right) g(R)\left(\overline{\overline{I}} - \hat{R}\hat{R}\right)\frac{1}{R} \tag{A.8}$$

$$= \left(\frac{3}{R^2} - \frac{3ik}{R} - k^2\right) g(R)\hat{R}\hat{R} + \left(\frac{ik}{R} - \frac{1}{R^2}\right) g(R)\overline{\overline{I}}. \tag{A.9}$$

Combining (A.2) with (A.9), the explicit form of the dyadic Green's function can be written as

$$\overline{\overline{G}}(\vec{r}, \vec{r}\,') = \left[\left(\frac{3}{k^2R^2} - \frac{3i}{kR} - 1\right)\hat{R}\hat{R} + \left(1 + \frac{i}{kR} - \frac{1}{k^2R^2}\right)\overline{\overline{I}}\right] g(R). \tag{A.10}$$

Appendix B

Error Tables for The Networks

Table B.1: MAE, MSE, MAPE of the test sets of FCNN with $H = 64$ for $\lambda/4$-sized box and $N_S = 64$ and MSE and RMSE of the complex valued numbers

Translation Direction	MAE (Real)	MSE (Real)	MAPE (Real)	MSE (Complex)	RMSE (Complex)
$[0, 0, 2]$	0.0070	0.1195e-03	2.2085	0.2716e-06	0.4150e-03
$[0, 1, 2]$	0.0037	0.0327e-03	1.5866	0.0214e-06	0.1294e-03
$[0, 2, 2]$	0.0012	0.0044e-03	1.1390	0.0889e-06	0.0143e-03
$[1, 1, 2]$	0.0022	0.0109e-03	2.3035	0.0033e-06	0.0391e-03
$[1, 2, 2]$	0.0010	0.0019e-03	0.5429	0.0021e-06	0.0057e-03
$[2, 2, 2]$	0.0006	0.0007e-03	0.4795	0.0001e-06	0.0018e-03
$[0, 0, 3]$	0.0011	0.0031e-03	0.6029	0.0013e-06	0.0089e-03
$[0, 1, 3]$	0.0009	0.0037e-03	0.5171	0.7313e-06	0.0091e-03
$[0, 2, 3]$	0.0006	0.0011e-03	0.4900	0.0182e-06	0.0028e-03
$[0, 3, 3]$	0.0005	0.0014e-03	1.0700	0.0597e-06	0.0032e-03
$[1, 1, 3]$	0.0008	0.0014e-03	0.5143	0.0006e-06	0.0036e-03
$[1, 2, 3]$	0.0006	0.0010e-03	0.3901	0.0135e-06	0.0024e-03
$[1, 3, 3]$	0.0005	0.0028e-03	0.3586	0.1661e-06	0.0060e-03
$[2, 2, 3]$	0.0006	0.0026e-03	0.6071	0.3782e-06	0.0058e-03
$[2, 3, 3]$	0.0005	0.0004e-03	0.3431	0.0006e-06	0.0009e-03
$[3, 3, 3]$	0.0004	0.0025e-03	0.4085	0.3314e-06	0.0055e-03

Table B.2: MAE, MSE, MAPE of the test sets of FCNN with $H = 128$ for $\lambda/4$-sized box and $N_S = 64$ and MSE and RMSE of the complex valued numbers

Translation Direction	MAE (Real)	MSE (Real)	MAPE (Real)	MSE (Complex)	RMSE (Complex)
$[0, 0, 2]$	0.0012	0.2897e-05	0.4928	0.0013e-07	0.8885e-05
$[0, 1, 2]$	0.0008	0.1237e-05	0.3445	0.0384e-07	0.3147e-05
$[0, 2, 2]$	0.0006	0.0675e-05	0.3137	0.0001e-07	0.1489e-05
$[1, 1, 2]$	0.0007	0.1104e-05	0.3891	0.0531e-07	0.2503e-05
$[1, 2, 2]$	0.0006	0.0654e-05	0.4399	0.0001e-07	0.1432e-05
$[2, 2, 2]$	0.0006	0.0587e-05	0.7841	0.0036e-07	0.1291e-05
$[0, 0, 3]$	0.0006	0.0682e-05	0.5153	0.0001e-07	0.1469e-05
$[0, 1, 3]$	0.0006	0.0641e-05	0.4137	0.0005e-07	0.1406e-05
$[0, 2, 3]$	0.0006	0.0637e-05	0.4915	0.0066e-07	0.1405e-05
$[0, 3, 3]$	0.0005	0.0815e-05	0.3986	0.1145e-07	0.1858e-05
$[1, 1, 3]$	0.0006	0.0555e-05	0.3460	0.0001e-07	0.1211e-05
$[1, 2, 3]$	0.0006	0.0515e-05	0.5260	0.0001e-07	0.1122e-05
$[1, 3, 3]$	0.0005	0.0469e-05	0.3824	0.0022e-07	0.1031e-05
$[2, 2, 3]$	0.0005	0.0513e-05	0.4763	0.0036e-07	0.1113e-05
$[2, 3, 3]$	0.0005	0.0397e-05	0.3618	0.0029e-07	0.0884e-05
$[3, 3, 3]$	0.0004	0.0872e-05	0.3953	0.3349e-07	0.2007e-05

Table B.3: MAE, MSE, MAPE of the test sets of CNN with $n_f = 128$ for $\lambda/4$-sized box and $N_S = 64$ and MSE and RMSE of the complex valued numbers

Translation Direction	MAE (Real)	MSE (Real)	MAPE (Real)	MSE (Complex)	RMSE (Complex)
$[0, 0, 2]$	0.0041	0.3348e-04	1.2871	0.4416e-07	0.0140
$[0, 1, 2]$	0.0023	0.0992e-04	0.9042	0.0395e-07	0.0076
$[0, 2, 2]$	0.0014	0.0322e-04	0.7574	0.0063e-07	0.0043
$[1, 1, 2]$	0.0018	0.0579e-04	0.7365	0.0254e-07	0.0058
$[1, 2, 2]$	0.0013	0.0270e-04	0.6555	0.0030e-07	0.0040
$[2, 2, 2]$	0.0011	0.0174e-04	0.6474	0.0017e-07	0.0032
$[0, 0, 3]$	0.0013	0.0287e-04	1.1023	0.0036e-07	0.0041
$[0, 1, 3]$	0.0011	0.0203e-04	2.4010	0.0016e-07	0.0035
$[0, 2, 3]$	0.0010	0.0154e-04	0.8112	0.0009e-07	0.0030
$[0, 3, 3]$	0.0009	0.0132e-04	0.6104	0.0069e-07	0.0027
$[1, 1, 3]$	0.0012	0.0226e-04	1.0191	0.0026e-07	0.0036
$[1, 2, 3]$	0.0010	0.0150e-04	0.7497	0.0009e-07	0.0030
$[1, 3, 3]$	0.0010	0.0163e-04	0.7483	0.0861e-07	0.0029
$[2, 2, 3]$	0.0010	0.0169e-04	0.7014	0.0176e-07	0.0031
$[2, 3, 3]$	0.0011	0.0184e-04	0.8691	0.0076e-07	0.0032
$[3, 3, 3]$	0.0009	0.0157e-04	1.0260	0.0324e-07	0.0029

Table B.4: MAE, MSE, MAPE of the test sets of FCNN with $H = 128$ for $\lambda/8$-sized box and $N_S = 64$ and MSE and RMSE of the complex valued numbers

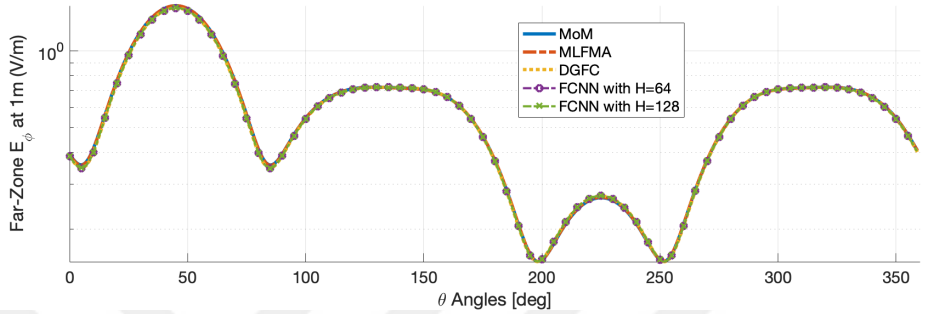
Translation Direction	MAE (Real)	MSE (Real)	MAPE (Real)	MSE (Complex)	RMSE (Complex)
$[0, 0, 2]$	0.0031	0.2330e-04	0.3646	0.0403e-06	0.0115
$[0, 1, 2]$	0.0016	0.0520e-04	0.2978	0.0012e-06	0.0054
$[0, 2, 2]$	0.0011	0.0214e-04	0.2930	0.0139e-06	0.0033
$[1, 1, 2]$	0.0012	0.0413e-04	0.2968	0.3665e-06	0.0037
$[1, 2, 2]$	0.0010	0.0169e-04	0.4437	0.0002e-06	0.0031
$[2, 2, 2]$	0.0009	0.0162e-04	0.3080	0.0045e-06	0.0028
$[0, 0, 3]$	0.0010	0.0171e-04	0.4197	0.0001e-06	0.0031
$[0, 1, 3]$	0.0010	0.0151e-04	0.3639	0.0001e-06	0.0029
$[0, 2, 3]$	0.0009	0.0126e-04	0.2811	0.0001e-06	0.0027
$[0, 3, 3]$	0.0008	0.0285e-04	0.3296	1.1453e-06	0.0025
$[1, 1, 3]$	0.0009	0.0142e-04	0.2314	0.0001e-06	0.0028
$[1, 2, 3]$	0.0009	0.0170e-04	0.2596	0.0111e-06	0.0027
$[1, 3, 3]$	0.0009	0.0247e-04	0.3238	0.0632e-06	0.0026
$[2, 2, 3]$	0.0009	0.0128e-04	0.3226	0.0019e-06	0.0026
$[2, 3, 3]$	0.0008	0.0230e-04	0.3178	0.1491e-06	0.0024
$[3, 3, 3]$	0.0007	0.0105e-04	0.3821	0.0218e-06	0.0022

Appendix C

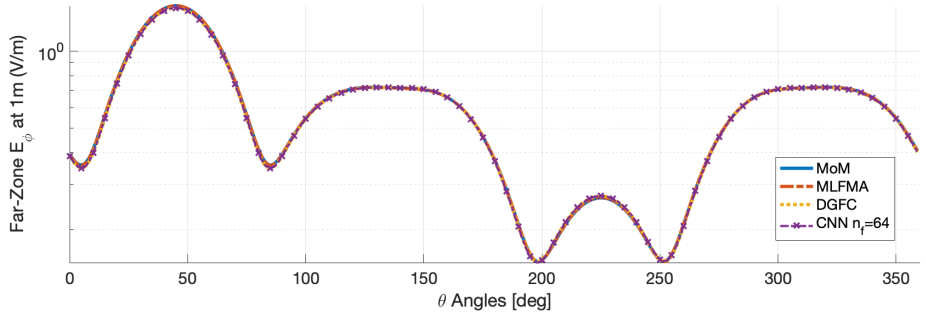
Additional Preliminary Results for the Scattering Problem of a Cube

Table C.1: MAPE and MAE of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its edge by MLFMA, DGFC, FCNN with $H = 64$ and $H = 128$, and CNN with $n_f = 64$ for the 3-level tree structure with respect to the MoM solution.

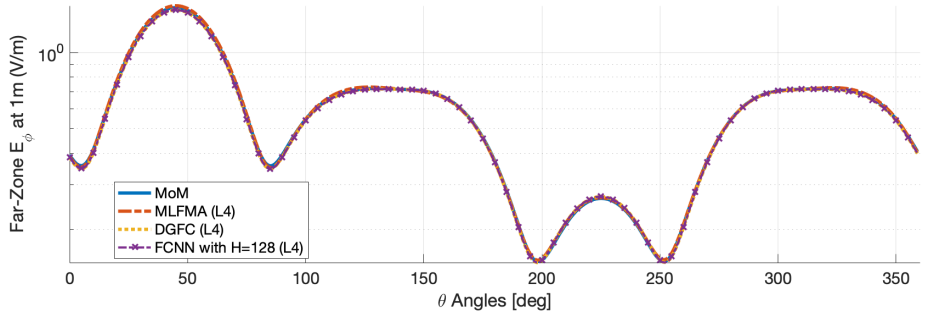
	MAPE	MAE
MLFMA	0.2328	0.0011
DGFC	1.4955	0.0086
FCNN 64	1.4836	0.0085
FCNN 128	1.4803	0.0085
CNN 64	1.5202	0.0086



(a)

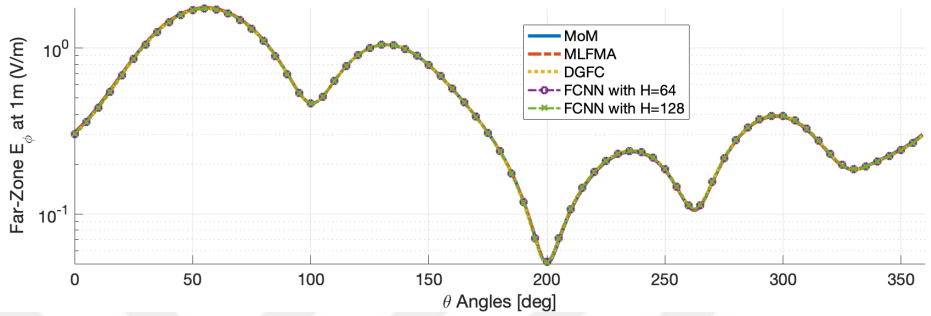


(b)

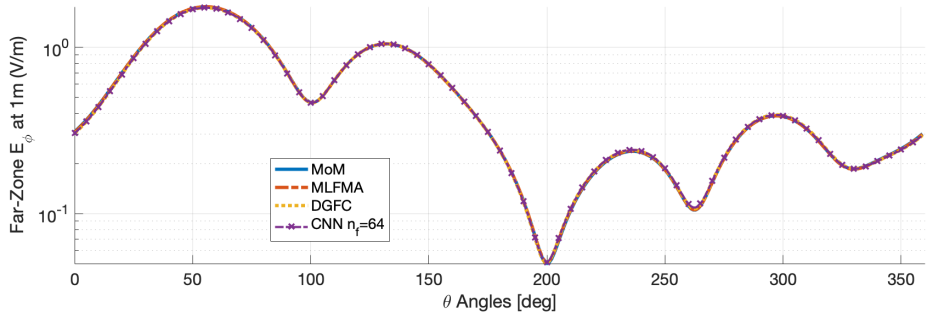


(c)

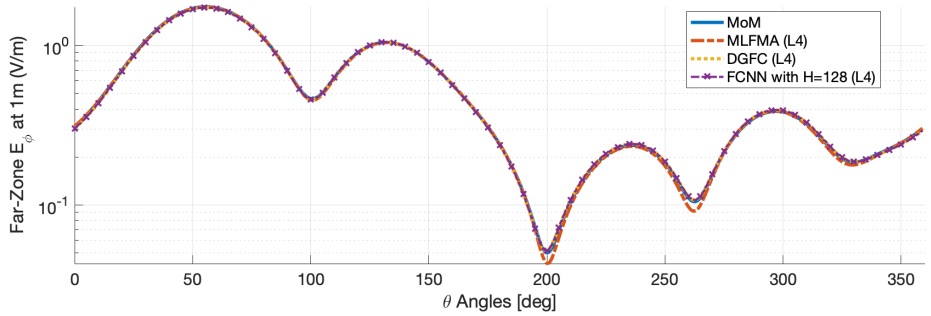
Figure C.1: Magnitude of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its edge by MoM, MLFMA, DGFC and the novel hierarchical ML-based method with (a) 2-layer FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure, (b) 2-layer CNN with $n_f = 64$ for the 3-level tree structure, and (c) 2-layer FCNN with $H = 128$ for the 4-level tree structure.



(a)



(b)



(c)

Figure C.2: Magnitude of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its vertex by MoM, MLFMA, DGFC and the novel hierarchical ML-based method with (a) 2-layer FCNN with $H = 64$ and $H = 128$ for the 3-level tree structure, (b) 2-layer CNN with $n_f = 64$ for the 3-level tree structure, and (c) 2-layer FCNN with $H = 128$ for the 4-level tree structure.

Table C.2: MAPE and MAE of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its edge by MLFMA, DGFC, and FCNN with $H = 128$ for the 4-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA (L4)	2.3605	0.0117
DGFC (L4)	1.6453	0.0095
FCNN 128(L4)	1.6448	0.0095

Table C.3: Time comparison of MoM, MLFMA, DGFC, FCNN with $H = 64$ and $H = 128$, and CNN with $n_f = 64$ for the 3-level tree structure to solve the scattering problem of the PEC cube illuminated from its edge.

	MoM	MLFMA	DGFC	FCNN 64	FCNN 128	CNN
Near Field Calc. (Sec)	104.92	31.83	28.43	21.10	20.95	28.46
Iterative Solver (Sec)	539.50	190.29	629.40	116.72	127.84	215.83
Number of Iterations	126	126	126	127	126	128
Time per Iteration (Sec)	4.28	1.51	4.99	0.91	1.01	1.68
Overall (Sec)	647.85	227.37	692.17	165.60	176.99	284.92

Table C.4: Time comparison of MoM, MLFMA, DGFC, FCNN with $H = 128$ for the 4-level tree structure to solve the scattering problem of the PEC cube illuminated from its edge.

	MoM	MLFMA (L4)	DGFC (L4)	FCNN 128 (L4)
Near Field Calc. (Sec)	104.92	6.39	5.34	5.52
Iterative Solver (Sec)	539.50	53.68	2402.41	108.25
Number of Iterations	126	143	126	126
Time per Iteration (Sec)	4.28	0.37	19.06	0.85
Overall (Sec)	647.85	64.15	2502.38	172.77

Table C.5: MAPE and MAE of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its vertex by MLFMA, DGFC, FCNN with $H = 64$ and $H = 128$, and CNN with $n_f = 64$ for the 3-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA	0.3461	0.0013
DGFC	2.0176	0.0112
FCNN 64	1.7216	0.0085
FCNN 128	1.7271	0.0083
CNN 64	1.6990	0.0084

Table C.6: MAPE and MAE of $\boldsymbol{E}_\phi$ (co-polarization) results at the far-zone obtained for the PEC cube illuminated from its vertex by MLFMA, DGFC, and FCNN with $H = 128$ for the 4-level tree structure with respect to the MoM solution.

	MAPE	MAE
MLFMA (L4)	3.4360	0.0105
DGFC (L4)	1.9581	0.0086
FCNN 128(L4)	1.9769	0.0083

Table C.7: Time comparison of MoM, MLFMA, DGFC, FCNN with $H = 64$ and $H = 128$, and CNN with $n_f = 64$ for the 3-level tree structure to solve the scattering problem of the PEC cube illuminated from its vertex.

	MoM	MLFMA	DGFC	FCNN 64	FCNN 128	CNN
Near Field Calc. (Sec)	105.03	32.89	29.65	20.87	21.08	28.74
Iterative Solver (Sec)	569.51	194.51	660.95	119.28	135.75	239.61
Number of Iterations	131	131	130	131	131	136
Time per Iteration (Sec)	4.34	1.48	5.08	0.91	1.03	1.76
Overall (Sec)	677.89	232.75	728.99	167.82	185.40	310.02

Table C.8: Time comparison of MoM, MLFMA, DGFC, FCNN with $H = 128$ for the 4-level tree structure to solve the scattering problem of the PEC cube illuminated from its vertex.

	MoM	MLFMA (L4)	DGFC (L4)	FCNN 128 (L4)
Near Field Calc. (Sec)	105.03	6.51	5.44	5.55
Iterative Solver (Sec)	569.51	54.76	2540.13	110.63
Number of Iterations	131	148	131	131
Time per Iteration (Sec)	4.34	0.37	19.39	0.84
Overall (Sec)	677.89	65.40	2652.72	174.77

Appendix D

Current Distributions

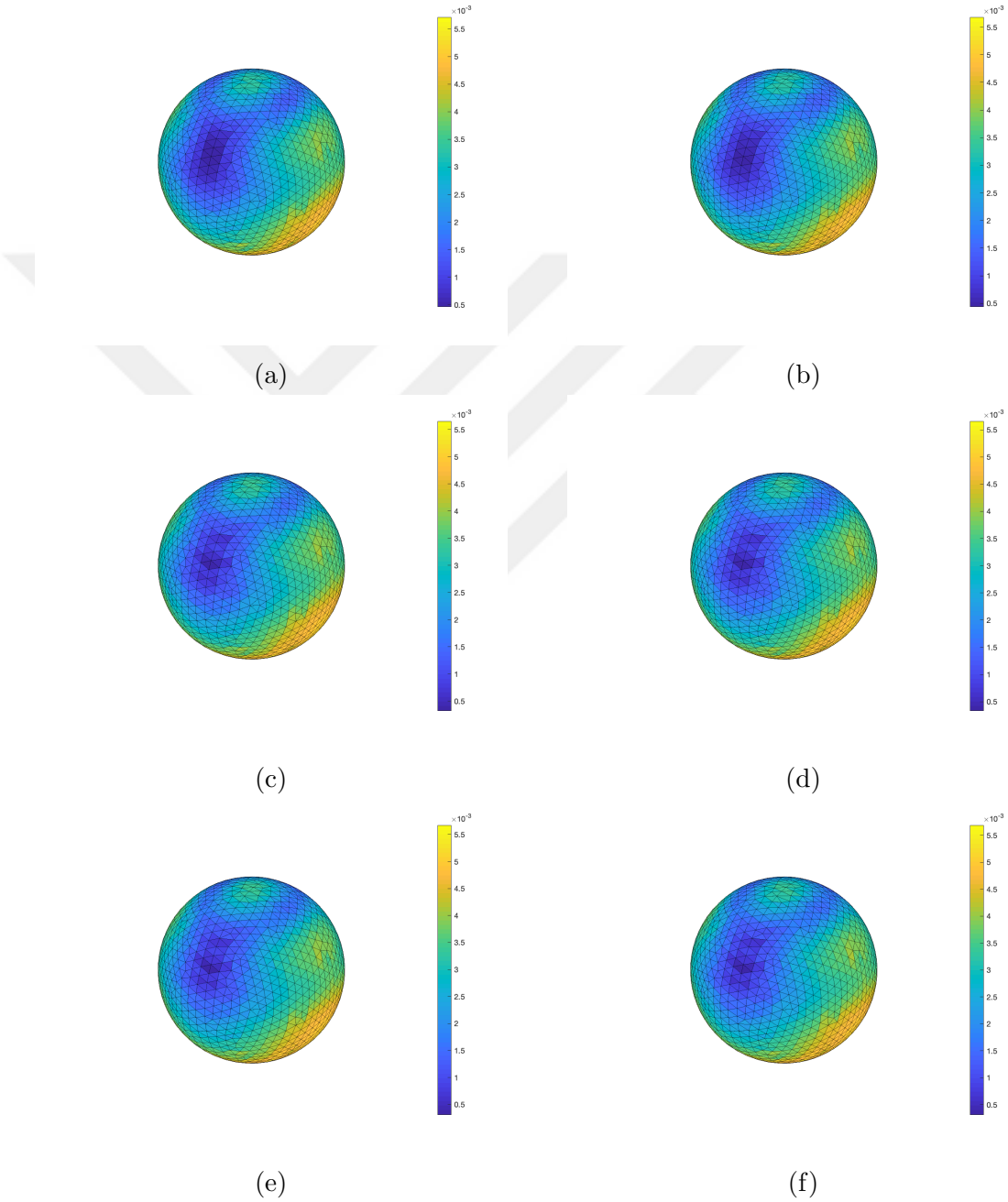


Figure D.1: Current distributions on the PEC sphere obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 64$, (e) hierarchical ML-based method with FCNN with $H = 128$, (f) hierarchical ML-based method with CNN with $n_f = 64$ for the 3-level tree structure.

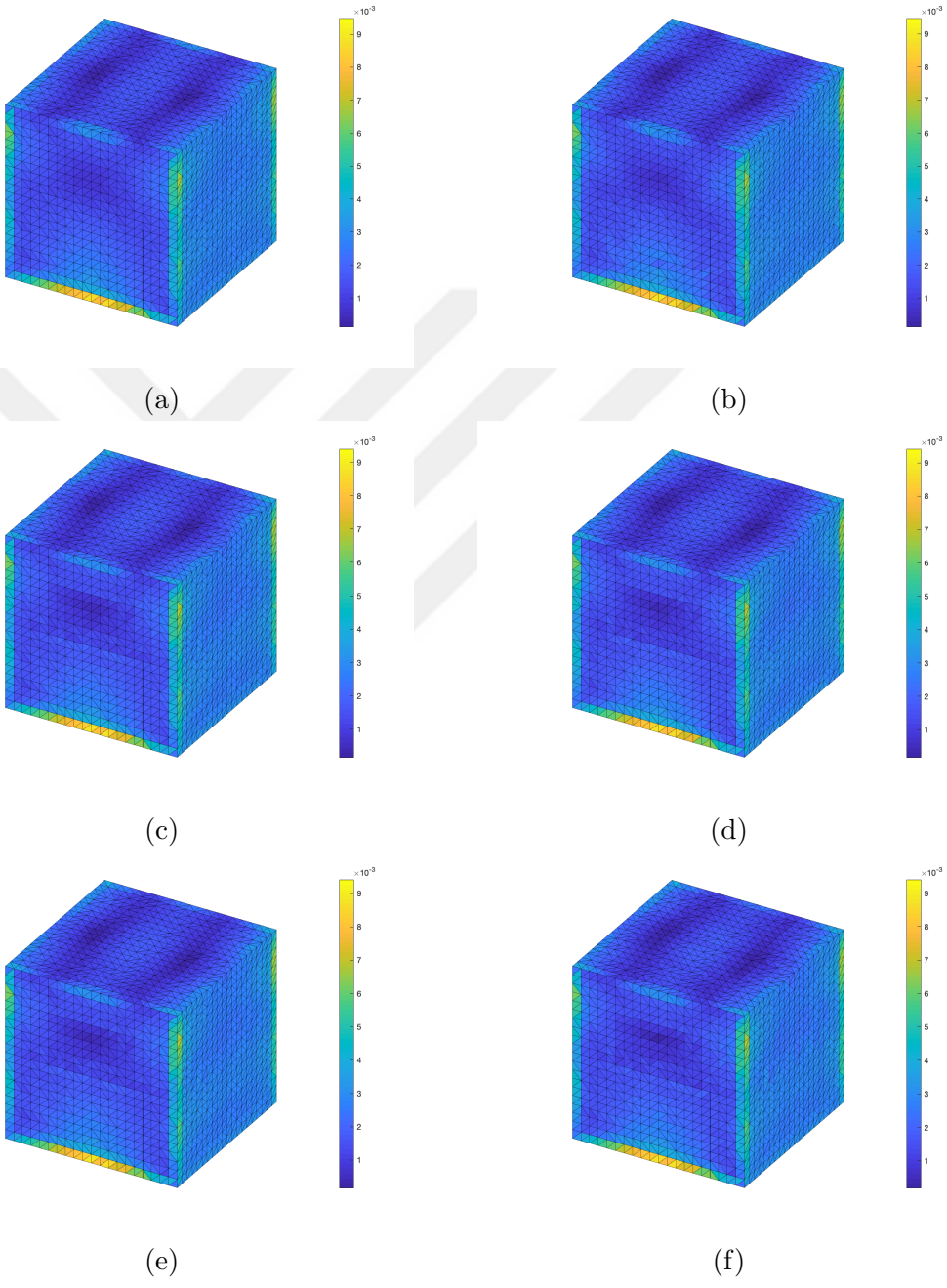


Figure D.2: Current distributions on the PEC cube illuminated from its face obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 64$, (e) hierarchical ML-based method with FCNN with $H = 128$, (f) hierarchical ML-based method with CNN with $n_f = 64$ for the 3-level tree structure.

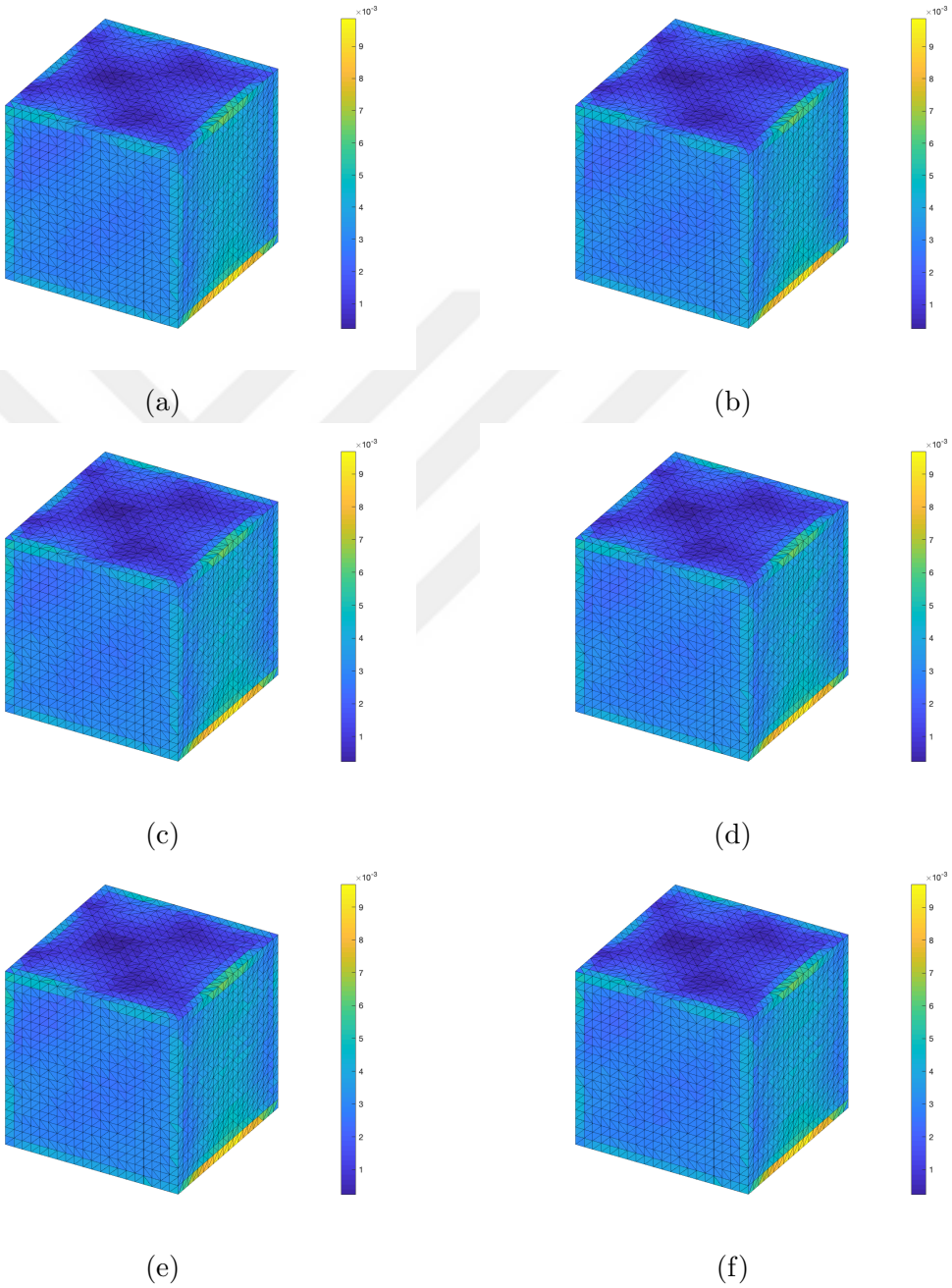


Figure D.3: Current distributions on the PEC cube illuminated from its edge obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 64$, (e) hierarchical ML-based method with FCNN with $H = 128$, (f) hierarchical ML-based method with CNN with $n_f = 64$ for the 3-level tree structure.

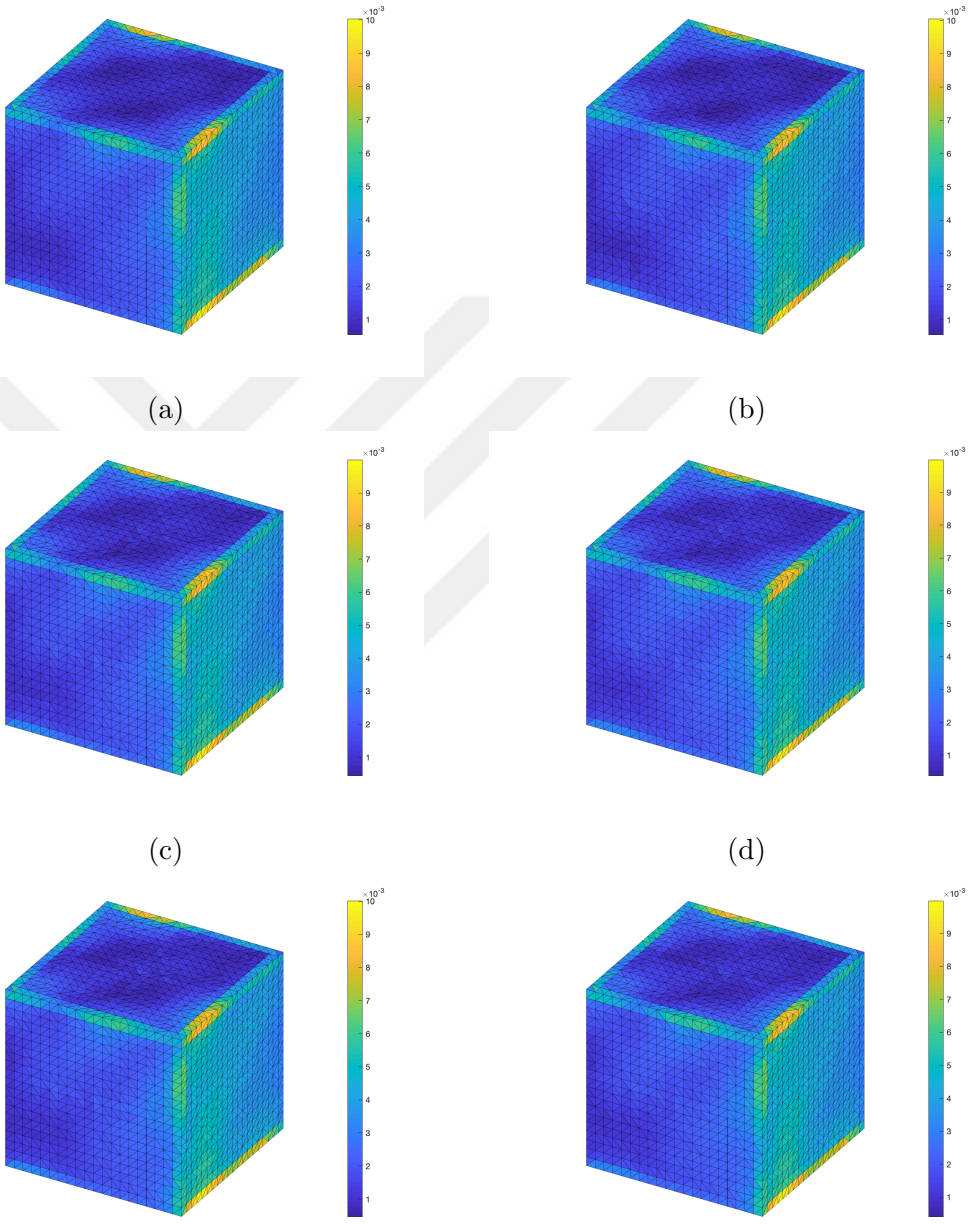


Figure D.4: Current distributions on the PEC cube illuminated from its vertex obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 64$, (e) hierarchical ML-based method with FCNN with $H = 128$, (f) hierarchical ML-based method with CNN with $n_f = 64$ for the 3-level tree structure.

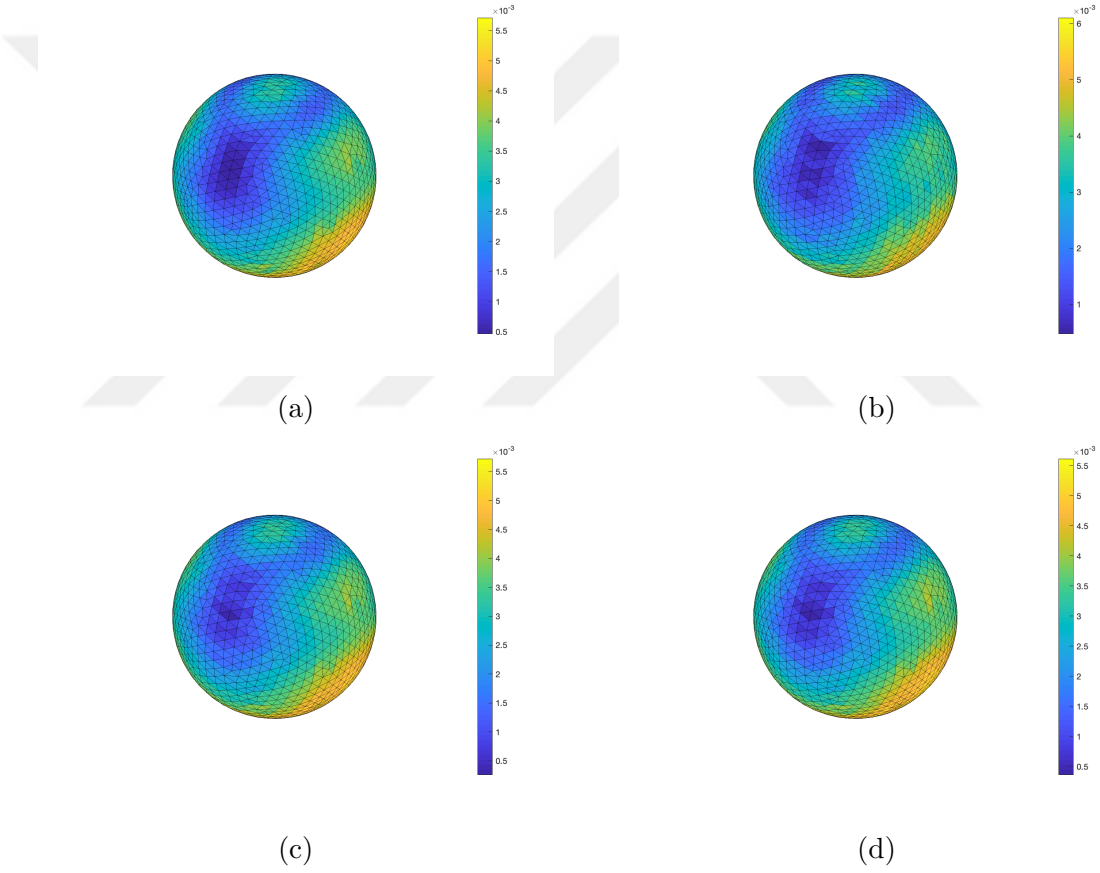


Figure D.5: Current distributions on the PEC sphere obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 128$ for the 4-level tree structure.

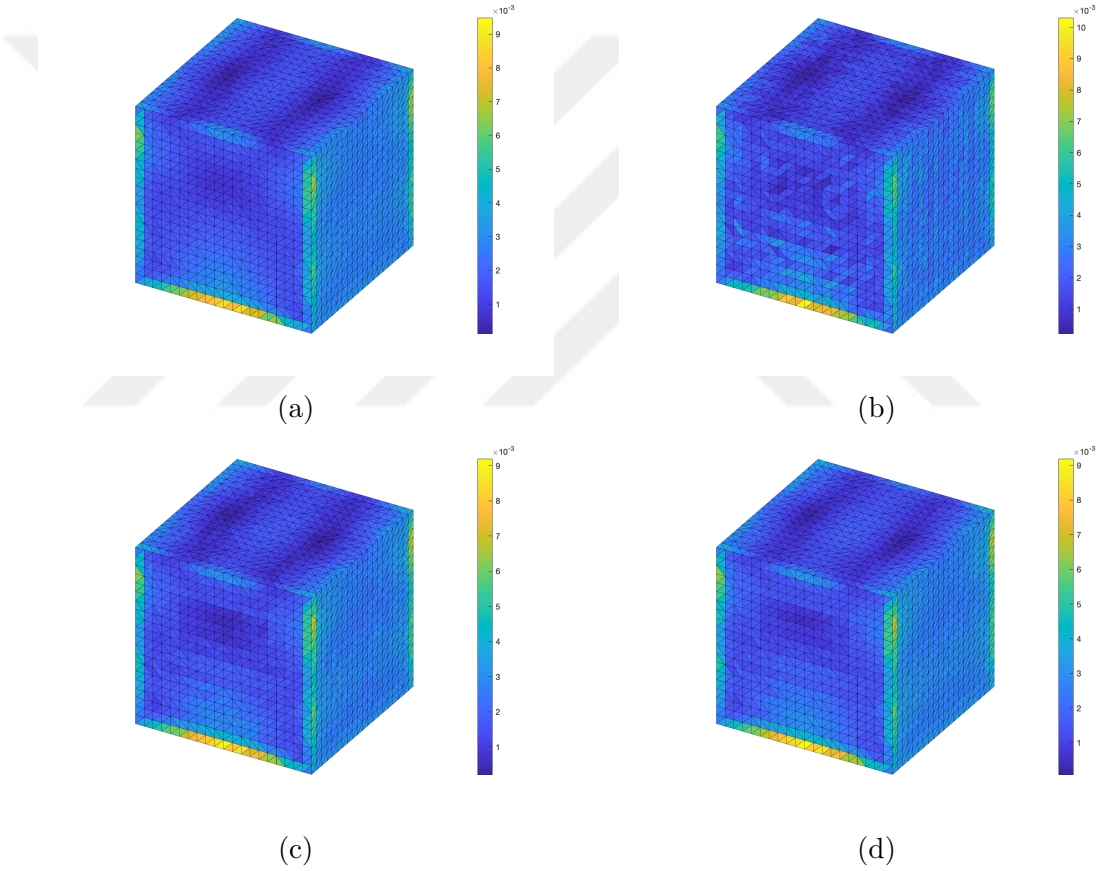


Figure D.6: Current distributions on the PEC cube illuminated from its face obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 128$ for the 4-level tree structure.

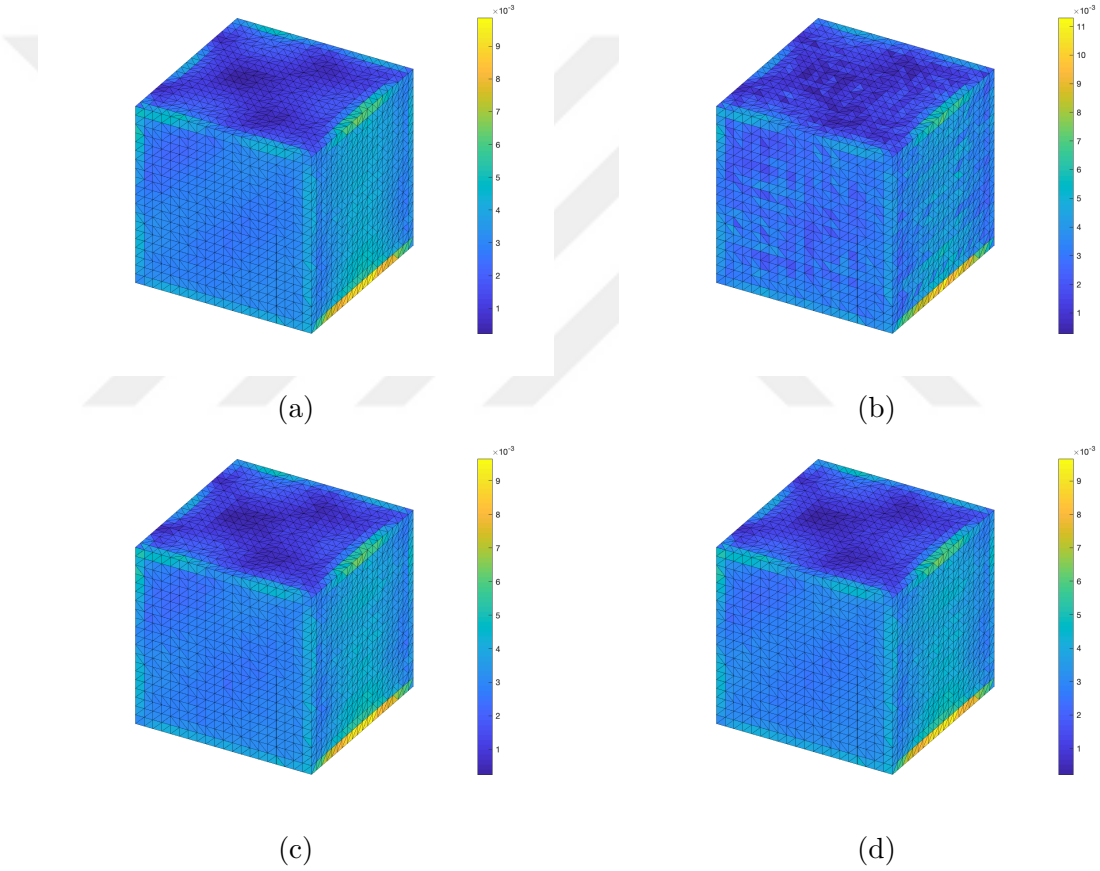


Figure D.7: Current distributions on the PEC cube illuminated from its edge obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 128$ for the 4-level tree structure.

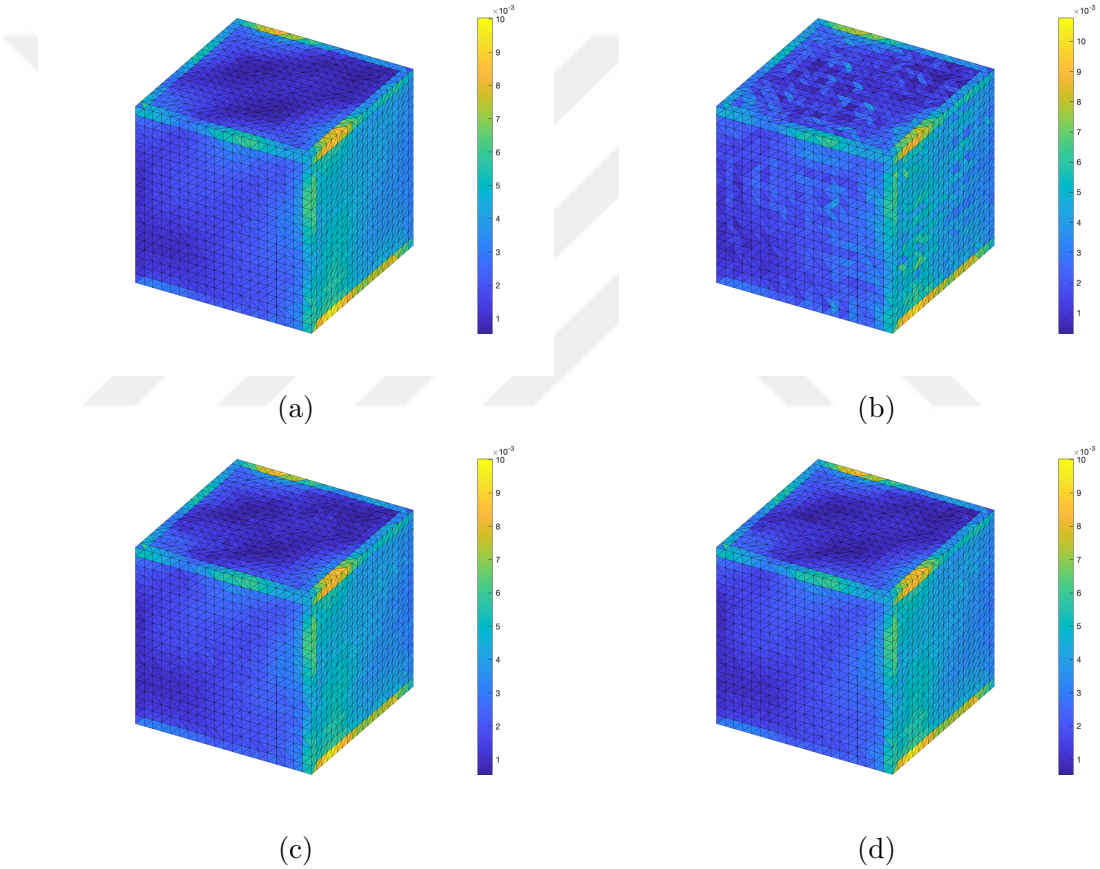


Figure D.8: Current distributions on the PEC cube illuminated from its vertex obtained by (a) MoM, (b) MLFMA, (c) DGFC, (d) hierarchical ML-based method with FCNN with $H = 128$ for the 4-level tree structure.