Reservoir Neuromorphic Computing with Photonic Superconducting Circuits

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Soham Aniruddha Gadre

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Dr. Kaveh Delfanazari

Certificate

This is to certify that this dissertation entitled Reservoir Neuromorphic Computing with Photonic Superconducting Circuits towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Soham Aniruddha Gadre at Indian Institute of Science Education and Research under the supervision of Dr. Kaveh Delfanazari, Associate Professor at the University of Glasgow, College of Science and Engineering , during the academic year 2024-2025.

Dr. Sreejith G.J.

Dr. Kaveh Delfanazari

Committee:

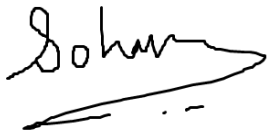
Dr. Kaveh Delfanazari

Dr. Sreejith G. J.

This thesis is dedicated to my mother, father and sister who have always supported and loved me unconditionally

Declaration

I hereby declare that the matter embodied in the report entitled Reservoir Neuromorphic Computing with Photonic Superconducting Circuits are the results of the work carried out by me at the College of Science and Engineering, Indian Institute of Science Education and Research, Pune, under the supervision of Dr. Kaveh Delfanazari and the same has not been submitted elsewhere for any other degree.



Soham Aniruddha Gadre

Roll Number: 20201045

Acknowledgments

Even though the past 10 months have been a challenge on their own, it was truly a joyous journey to overcome them with the help of some truly special people.

First and foremost, I must thank my supervisor Dr. Kaveh Delfanazari for his guidance and confidence in my abilities as a physicist. I would also like to thank Mingqi Zhang, Samane Kalhor for their expertise and collaboration. I must also thank Alex and Uri for the precious friendships we shared that helped me immensely with the challenges of living in a foreign land. Secondly, I want to thank everyone at IISER Pune who over these 5 years have shaped me into the researcher that I am today, especially Dr. Ashish Arora and Dr. Aparna Deshpande. Having them as my role models for an ideal researcher made me progress by leaps and bounds.

And finally, to my parents, my sister and my cousin brother, whom I can never thank enough, I am more grateful than ever. Their unwavering support and faith in me has been more important than they could know.

Abstract

The focus of this thesis is to understand the applicability of Artificial Intelligence mainly, Neural Network models in the prediction of the optical responses of superconducting metamaterials which is inherently a quantum phenomenon. It discusses the viability of several Neural Network architectures in solving this problem and lays out quantitative results comparing the different architectures for the reader to choose based on their approach and parameters such as nature of the task and resources. Superconducting metamaterials like Niobium show the unique phenomenon of plasmon polariton formation, whether they are surface plasmon polaritons or bulk plasmon polaritons, when exposed to radiation in a specific wavelength range. This thesis also focuses on the determination and replication of the mechanisms of plasmon polariton formation in the Niobium metamaterial via simulation environments and paves the way for future Machine Learning work in the ever-changing world of plasmonics.

Contents

Abstract **xi**

1 Introduction **5**

2 Preliminaries **9**

3 Background **13**

3.1 Surface Plasmon Polaritons (SPPs) 13

3.2 Localized Surface Plasmon Polaritons (LSPPs) 18

3.3 Spoof Surface Plasmon Polaritons 21

3.4 Machine Learning 24

4 Surface-Corrugated Niobium nanodevices **37**

4.1 Niobium Nano-device arrays 38

4.2 Deciphering the plasmonic mode type 41

5 Simulation software and data generation **45**

5.1 Learning different simulation software 46

5.2 Inverse Design vs Regular Design Neural Networks 50

5.3 Replicating Experimental Results 50

6 Neural Networks and Results **57**

6.1 Artificial Neural Network Ensemble Model 57

6.2 Transformer Model 59

6.3 Graphical Neural Network (GNN) 60

6.4 Convolutional Neural Network (CNN) 63

6.5 Model Comparison and Quantitative Evaluation 64

7 Discussion **67**

7.1 Analysis of Model Performance 67

7.2 Insights and Key Takeaways 68

8 Future Work **71**

8.1 Prospective work 71

8.2 Conclusion 72

List of Figures

3.1 Illustration of p-polarized electromagnetic waves striking a flat boundary between two materials at an angle of incidence θ_1 from [9] 14

3.2 15

3.3 The dispersion of surface plasmon polaritons on a metallic surface is depicted. The dispersion of surface plasmons and light in the nearby medium is also illustrated. [2] 17

3.4 Illustration of a homogeneous sphere placed into an electrostatic field taken from [10] 19

3.5 The model taken from [11] 22

3.6 General Neural Network Architecture 26

3.7 Message-passing in GNN's from [26] 29

3.8 Message-passing in GNN's from [26] 31

3.9 An example code of Optuna's define-by-run API taken from one of the codes developed during this study 35

4.1 Images taken from [6] with permission 39

4.2 Images taken from [6] with permission 40

4.3 Images taken from [6] with permission 43

5.1 Simulation with PyGDM 47

5.2 Simulated models of the Nb nanodevices 51

5.3 Parametric sweeps to deduce parameters 53

5.4 Simulated extinction spectra for nanoantenna nanodevice 54

5.5 Simulated extinction spectra for SRR nanodevice 54

6.1 Comparison between ANN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1 58

6.2 Comparison between ANN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2 59

6.3 Comparison between Transformer predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1 60

6.4 Comparison between Transformer predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2 60

6.5 Graphical representation of a pointcloud using KNN where K=5 61

6.6 Comparison between GNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1 62

6.7 Comparison between GNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2 62

6.8 Comparison between CNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1 63

6.9 Comparison between CNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2 64

List of Tables

6.1 Model Comparison 65

Chapter 1

Introduction

Plasmonics, also known as nanoplasmonics, involves the generation, detection, and manipulation of optical signals at metal-dielectric interfaces on the nanometer scale. These plasmonic phenomena have catalyzed advancements in diverse optical applications, including nonlinear optics, thermo-photovoltaics, nanoscale heat transfer systems, photothermal therapy, and plasmonic-based solar cells. The unique ability of plasmonic materials to confine light to subwavelength dimensions has fueled extensive research into their properties and applications. The plasmonic materials currently utilized for the aforementioned applications are primarily confined to noble metals like gold and silver, known for their excellent optical characteristics that allow for narrow-band plasmon resonances. Nonetheless, their inadequate thermal stability can be addressed by employing refractory materials in plasmonic applications. Niobium, which is categorized as a refractory material, possesses optical properties akin to gold and additionally demonstrates quantum effects such as superconductivity, making it an extremely promising candidate for both contemporary and upcoming plasmonic systems focused on active plasmonic devices and plasmonic single-photon detectors for quantum computing.

The interaction of light with metal surfaces is primarily influenced by free electrons, which act like a plasma described by a dielectric function of the form $\epsilon = 1 - \frac{\omega_p^2}{\omega}$, becoming negative below the plasma frequency, ω_p, related to bulk longitudinal electron excitations. These oscillations, known as surface plasmons, give rise to various distinct phenomena, including surface-enhanced Raman scattering (SERS). At a fundamental level, a plasmon represents the quantum of these collective electron oscillations. Therefore, a plasmon is essentially the

quantum unit of the collective excitation of free electrons within solids. The frequency of a surface plasmon on a flat metallic surface, ω_{sp}, can be easily determined from the frequency of a bulk plasmon of the metal since it relates to the condition $Re\,\epsilon_m(\omega_{sp}) = -\epsilon_i$, where $\epsilon_i > 0$ denotes the dielectric constant of the dielectric material adjacent to the metal [2].

When radiation of a particular wavelength range (usually in the Vis-NI range) is incident on a metallic surface, the surface plasmon excitation gets coupled to the photon resulting in a hybrid excitation wave called a surface plasmon polariton. Plasmonics leverages these hybrid waves, often through engineered materials called metamaterials. These metamaterials consist of periodic or aperiodic nanostructures with dimensions significantly smaller than the wavelength of light. By tailoring these structures, plasmonic devices can be designed to exhibit customized optical properties for a range of applications. [3].

There are several types of surface plasmon polaritons (SPPs) that arise based on various factors, including the dimensions of the nanostructure, the nature of the surface features, and the periodicity of the structure. Among the most widely studied types of SPPs are localized surface plasmon polaritons (LSPPs) and spoof or designer SPPs. Localized SPPs occur in nanostructures such as metal nanoparticles or nanoholes, where the plasmons are confined within the structure, leading to highly tunable resonance properties. These localized resonances enhance electromagnetic fields at specific points, making them useful for applications such as sensing and nonlinear optics. On the other hand, spoof or designer SPPs are artificially engineered surface modes that mimic conventional plasmonic behavior in materials that do not naturally support plasmonic excitations. By introducing periodic surface patterns or corrugations, these modes allow for the manipulation of light in ways that extend beyond the constraints of traditional plasmonic materials, enabling novel applications in waveguiding and optical communication.

The integration of neural networks into plasmonics has seen the application of diverse architectures to address both forward and inverse design problems. Forward problems involve predicting the optical response of a nanostructure given its physical parameters, while inverse problems focus on determining the structural parameters needed to achieve a desired optical response [4,5]. A typical workflow for applying neural networks in plasmonic research includes:

- **Simulation Environment Setup:** Selecting a computational platform, such as finite-difference time-domain (FDTD) or finite element method (FEM) software, to model

plasmonic interactions.

- **Validation:** Ensuring the accuracy of the simulation results by comparing them with experimental data.
- **Generation of training data:** Generating extensive training datasets through simulations, incorporating variations in structural and material parameters.
- **Feature mapping:** Discretizing the nanostructure's volume and mapping its features to the desired optical responses.
- **Model Optimization:** Fine-tuning neural network architectures to maximize predictive accuracy and computational efficiency.

This thesis adopts a comparative approach to examine the application of neural networks in relating the plasmonic properties of metamaterials to their physical features. A significant focus is placed on evaluating the performance of different neural network architectures for these tasks. Additionally, the thesis investigates various simulation software platforms to identify the most efficient and accurate tools for generating reliable training data. By integrating computational physics with machine learning, this research aims to elucidate the mechanisms and nature of light-matter interactions in artificially designed surface-corrugated niobium metamaterials.

The study relates to niobium nanodevices incorporating nanoantennas and split-ring resonator voids at periodic intervals. It builds upon the previous work of my supervisor and their research team on photonic superconducting circuits. While the experimental data has already been recorded and published, my research takes a purely theoretical and computational approach, employing simulations and machine learning techniques to predict and fine-tune the optical response of these devices based on their structural features [6-8]. By leveraging advanced computational methods, this study seeks to bridge the gap between theoretical modeling and experimental observations, enabling a deeper understanding of how the engineered nanostructures influence plasmonic behavior in niobium-based metamaterials.

The choice of niobium as a material system is particularly compelling due to its dual functionality as a plasmonic material and a superconductor. This duality enables the exploration of novel phenomena, such as the interplay between superconductivity and plasmonics, and their potential applications in quantum information science. Specifically, the surface

corrugation of niobium metamaterials introduces additional complexity to the light-matter interaction, creating opportunities for designing devices with highly tailored optical responses.

Overall, this thesis represents a confluence of materials science, computational physics, and machine learning, aiming to advance the understanding and application of plasmonic metamaterials. By leveraging the unique properties of niobium and the predictive power of neural networks, this work contributes to the development of next-generation plasmonic devices with applications ranging from photonic computing to quantum technologies.

Chapter 2

Preliminaries

Following are some definitions and physical terms used in calculations done throughout the thesis

Definition 2.0.1. *Polarizability α is a measure of how easily a material or particle becomes polarized in response to an external electric field. It quantifies the induced dipole moment* $\mathbf{p}$ *per unit applied electric field E_0 It is defined via the equation:*

$$\mathbf{p} = \epsilon_0 \epsilon_m \boldsymbol{\alpha} E_0 \tag{2.1}$$

where

- $\mathbf{p}$ *is the induced dipole moment*
- ϵ_0 *is the permittivity of free space having value* 8.85×10^{-12} *SI Units*
- ϵ_m *is the dielectric constant of the surrounding medium*
- α *is the polarizability*
- E_0 *is the electric field*

Definition 2.0.2. *The dielectric function $\epsilon(\omega)$ of a free electron gas (also known as the Drude model dielectric function) describes how a material responds to an external oscillating electric field at frequency*

$$\omega$$

. It is defined as:

$$\epsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} \tag{2.2}$$

where

- ω_p *is the plasma frequency, which characterizes the collective oscillations of free electrons*
- γ *is the damping coefficient, representing energy loss due to electron scattering*
- ω *is the angular frequency of the applied fieldis the angular frequency of the applied field*

Definition 2.0.3. *A voxel (short for "volume pixel") is the three-dimensional equivalent of a pixel in a 2D image. It represents a discrete element of a three-dimensional space, typically in a uniform cubic grid. Voxels are often used in computational simulations, medical imaging, and 3D rendering to represent objects and spatial data.*

Voxelization is the process of converting a 3D model, such as a point cloud or a mesh, into a structured voxel grid. This involves dividing the 3D space into smaller cube-like elements, each storing information about the object's presence at that location. The resolution of a voxel grid is determined by:

$$N_x = \frac{L_x}{s},\ N_y = \frac{L_y}{s},\ N_z = \frac{L_z}{s} \tag{2.3}$$

where

- L_x, L_y, L_z *are the dimensions of the bounding box encapsulating the model*
- s *is the user-defined voxel size*
- N_x, N_y, N_z *are the number of voxels along each axis*

For example, if the bounding box dimensions are $15 \times 20 \times 25$ *and the voxel size is set to* $s=0.5$*, then the resolution of the voxel grid will be* $30 \times 40 \times 50$*. A finer voxel size increases resolution, capturing more details of the model, but also significantly increases computational cost. Voxelization is useful in applications requiring a structured representation of 3D objects, such as physics simulations and deep learning models trained on 3D data.*

Definition 2.0.4. *A point cloud is a collection of discrete data points in a 3D coordinate system, representing the shape and structure of an object. Each point in a point cloud is defined by its spatial coordinates (x,y,z) and may also carry additional attributes, such as color, intensity, or surface normals.*

Point clouds are typically obtained from 3D scanning methods like LiDAR, structured light, or photogrammetry. Unlike voxel grids, which impose a fixed structure on the data, point clouds offer a more flexible representation. They provide several advantages, particularly for applications in neural networks:

- **Precision and Detail***: Each point directly represents a position in 3D space, capturing fine details of an object's surface without imposing a predefined grid structure.*
- **Scalability***: Point clouds can be efficiently stored and processed at different densities, making them suitable for large-scale datasets.*
- **Flexibility in Resolution***: Unlike voxel grids, where the resolution is fixed, point clouds can adaptively vary their density, allowing for higher fidelity in regions of interest while maintaining efficiency.*
- **Compatibility in Deep Learning Models***: Many neural network architectures, such as PointNet and Graph Neural Networks, are optimized for point cloud processing, making them a preferred choice for machine learning applications involving 3D data.*

Given the computational cost and memory constraints associated with voxelization, point clouds were chosen as the primary 3D data representation in this study.

Definition 2.0.5. *A mesh is a collection of vertices, edges, and faces that define the shape of a 3D object. The most common type of mesh representation is the triangular mesh, where an object's surface is approximated by a set of interconnected triangles. Each triangle is defined by three vertices in 3D space.*

Meshes are widely used in computer graphics, physics simulations, and 3D printing due to their efficiency in representing complex surfaces. Converting a 3D model into a mesh allows for operations such as:

- **Voxelization***: Converting a mesh into a structured voxel grid for simulations and deep learning applications.*

- **Surface Analysis***: Extracting geometric properties such as curvature and topology.*
- **Rendering and Visualization***: Generating realistic representations of objects in computer graphics.*

Meshes serve as an intermediate representation between raw point clouds and structured voxel grids, offering a balance between precision and computational efficiency.

Chapter 3

Background

This chapter establishes the fundamental plasmonic physics underlying the decay of evanescent fields at metal-dielectric interfaces, focusing on surface plasmon polaritons (SPPs), localized surface plasmon polaritons (LSPPs), and spoof/designer surface plasmon polaritons (SSPPs). By examining these phenomena, we aim to identify the key factors that influence the optical behavior of surface-corrugated niobium nanodevices [9-12]

Additionally, this chapter provides a foundational overview of neural networks, covering their core components, fundamental working principles, and mathematical formulations. It introduces the key types of neural networks employed in this study, detailing their underlying equations and mechanisms to establish a theoretical basis for their application in plasmonic simulations and predictive modeling.

3.1 Surface Plasmon Polaritons (SPPs)

SPPs are electromagnetic waves that travel along the border between a metal and a dielectric material, characterized by the presence of surface charges. To analyze the excitation of these SPPs, we examine a p-polarized wave (TM mode, where the electric field vector lies in the plane of incidence) that strikes a smooth flat interface at an angle of incidence θ_1. The wave entering the dielectric has a photon momentum of $\hbar k_d$, corresponding to a refractive index n_d. According to Snell's law, refraction occurs at the boundary based on the equation given

below: (Derivation from [9])

$$n_d \sin\theta_1 = n_m \sin\theta_2 \tag{3.1}$$

, where n_m is the refractive index of the metal and θ_2 is the angle of refraction as depicted in the figure 3.1.

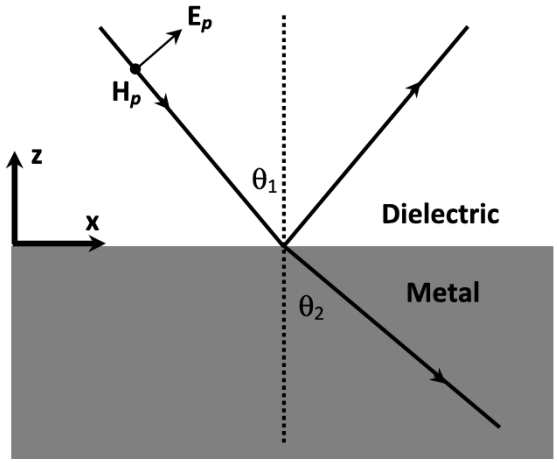


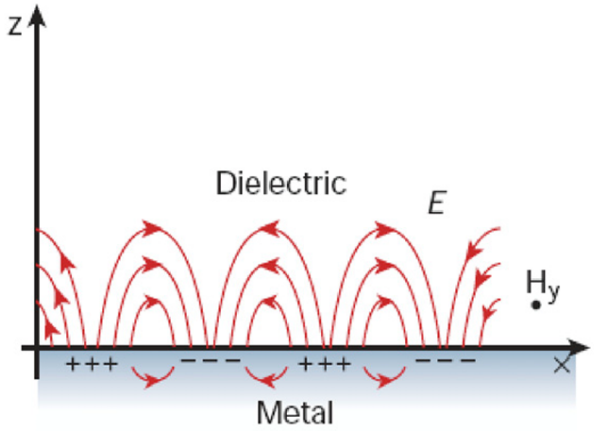
Figure 3.1: Illustration of p-polarized electromagnetic waves striking a flat boundary between two materials at an angle of incidence θ_1 from [9]

Beyond the limiting or critical angle θ_c, the wave cannot propagate in the metal; in this case, the limiting incident angle is called the critical angle θ_c, which is given by

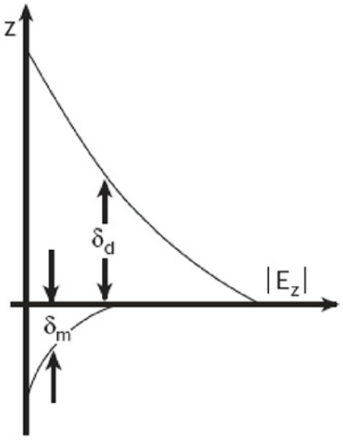
$$\sin\theta_c = \frac{n_m}{n_d} \tag{3.2}$$

A wave striking the surface at an angle greater than θ_c possesses more momentum in the plane of the surface than what the metal can accommodate. When a p-polarized wave hits the interface, the electric field that oscillates induces surface charges at the boundary between the metal and the dielectric, leading to a collective oscillation of these surface charges. Even though the wave is completely reflected at the interface, the oscillating charges generate radiation fields that extend into the metal. These fields decay spatially (evanescent fields) in a direction perpendicular to the interface (see figure 3.2).

Since there is no boundary that is perpendicular to E_x, this component remains constant across the boundary. In contrast, the same does not hold for E_z, the component of E that is normal; the normal component of D, D_z, remains continuous (as there is no free charge),



(a) Illustration of electromagnetic wave and surface charges at the metal-dielectric interface



(b) The local E-field is amplified near the surface and decays as distance perpendicular to surface increases

Figure 3.2

and E_z must change if the permittivity ϵ changes since $D_z = \epsilon_d \epsilon_0 E_{zd} = \epsilon_m \epsilon_0 E_{zm}$. This change in E_z leads to variations in polarization at the interface. From these straightforward considerations, it is clear that while s-polarized incoming radiation typically does not lead to the formation of charges at a flat interface, p-polarized waves will inherently generate time-varying polarization charges at the interface. Assuming the x–y plane serves as the interface and wave propagation occurs in the x direction exclusively, when z>0, one considers

$$E_d = (E_{xd}, 0, E_{zd})\, e^{-k_{zd} z} e^{i(k_x x - \omega t)} \tag{3.3}$$

$$H_d = (0, H_{yd}, 0)\, e^{-k_{zd} z} e^{i(k_x x - \omega t)} \tag{3.4}$$

when z<0, one has

$$E_m = (E_{xm}, 0, E_{zm})\, e^{k_{zm} z} e^{i(k_x x - \omega t)} \tag{3.5}$$

$$H_m = (0, H_{ym}, 0)\, e^{k_{zm} z} e^{i(k_x x - \omega t)} \tag{3.6}$$

By analyzing equations 3.3 and 3.5 with respect to the Maxwell equation $\nabla.E = 0$, the components of the electric field are expressed as

$$E_{zd} = i\frac{k_x}{k_{zd}}E_{xd} \tag{3.7}$$

$$E_{zm} = -i\frac{k_x}{k_{zm}}E_{xm} \tag{3.8}$$

Conversely, we examine equations 3.3 and 3.4 and utilize the Maxwell equation in the following manner:

$$\nabla \times E = -\frac{1}{c}\frac{\partial H}{\partial t} \tag{3.9}$$

Then we have

$$-k_{zd}E_{xd} - ik_xE_{zd} = ikH_{yd} \tag{3.10}$$

where $k = \omega/c$ denotes the wavevector in a vacuum. In a similar manner, by examining equations 3.5, 3.6, and 3.9, we obtain

$$k_{zm}E_{xm} - ik_xE_{zm} = ikH_{ym} \tag{3.11}$$

Regarding equations 3.7 and 3.8, equations 3.10 and 3.11 can respectively be represented as follows:

$$\epsilon_d kE_{xd} = ik_{zd}H_{yd} \tag{3.12}$$

$$\epsilon_m kE_{xm} = -ik_{zm}H_{ym} \tag{3.13}$$

where

$$k_{zd}^2 = k_x^2 - \epsilon_d k^2 \tag{3.14}$$

$$k_{zm}^2 = k_x^2 - \epsilon_m k^2 \tag{3.15}$$

Furthermore, it is noted that the tangential components of the electric field (E) and magnetic field (H) are continuous due to the boundary conditions for electromagnetic fields at z=0. This indicates that we have $E_{xd} = E_{xm}$ and $H_{yd} = H_{ym}$. The relationship between the dielectric constants and the normal components of the wavevectors in the two distinct

media is described by

$$\frac{k_{zd}}{k_{zm}} = -\frac{\epsilon_d}{\epsilon_m} \tag{3.16}$$

In terms of equations 3.14, 3.15 and 3.16, we have

$$k_{spp} = k\sqrt{\frac{\epsilon_d \epsilon_m}{\epsilon_d + \epsilon_m}} \tag{3.17}$$

The dispersion curve for surface plasmon polaritons (SPPs) exhibits nonlinear characteristics, as demonstrated in figure 3.3. At a specific frequency, the momentum of the SPP wave ($\hbar k_{SPP}$) exceeds that of a photon in free space ($\hbar k$), leading to a momentum discrepancy between light and SPP. To resolve this mismatch, it is crucial to couple the light and SPP modes at the interface, such as:

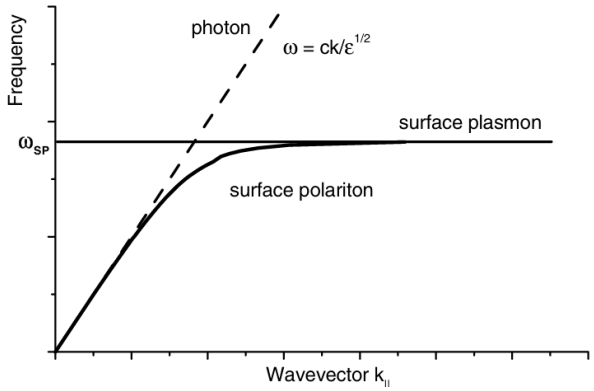


Figure 3.3: The dispersion of surface plasmon polaritons on a metallic surface is depicted. The dispersion of surface plasmons and light in the nearby medium is also illustrated. [2]

$$\epsilon_d + \epsilon_m = 0 \tag{3.18}$$

Collective oscillations can be produced by surface charges, allowing for the excitation of surface plasmon polaritons (SPPs). The metal's dielectric constant, denoted as ϵ_m, follows the free electron model:

$$\epsilon_m = 1 - \frac{w_p^2}{w^2} \tag{3.19}$$

where ω_p represents the bulk plasma frequency. For numerous metals, this frequency falls within the ultraviolet wavelength range, beyond which the material ceases to exhibit metallic

properties. According to equations 3.18 and 3.19, the frequency of the SPP is expressed as

$$\omega_{spp} = \frac{\omega_p}{\sqrt{1+\epsilon_d}} \tag{3.20}$$

It has been noted that the frequency of the SPP is lower than that of the bulk plasma frequency. As previously mentioned, the excited SPP waves feature evanescent fields that penetrate into both the dielectric and the metal. These fields decrease in strength spatially in a direction that is perpendicular to the interface (see figure 3.1), and the decay behavior of the SPP wave in the dielectric and the metal aligns with equations 3.3 and 3.5. In this context, the penetration depths within the dielectric and the metal, referred to as δ_d and δ_m, are defined respectively as $\delta_d = \frac{1}{k_{zd}}$ and $\delta_m = \frac{1}{k_{zm}}$, where the electric fields reach $\frac{1}{e}$ of their peak values. According to equations 3.13 and 3.16, the penetration depth δ_d of the SPP wave into the dielectric is expressed as

$$\delta_d = \frac{1}{k}\left|\frac{\epsilon_d + \epsilon_m}{-\epsilon_d^2}\right|^{\frac{1}{2}} \tag{3.21}$$

$$\delta_m = \frac{1}{k}\left|\frac{\epsilon_d + \epsilon_m}{-\epsilon_m^2}\right|^{\frac{1}{2}} \tag{3.22}$$

This concludes the fundamental electrodynamic formulation for SPP excitation at a metal-dielectric interface. To fully understand the key factors influencing the optical behavior of surface-corrugated niobium nanodevices, it is essential to examine the two primary types of SPPs—localized surface plasmon polaritons (LSPPs) and spoof/designer surface plasmon polaritons (SSPPs).

3.2 Localized Surface Plasmon Polaritons (LSPPs)

Localized surface plasmons are stationary excitations of the conduction electrons within metallic nanostructures that interact with the electromagnetic field. The curvature of the particle's surface applies an effective restoring force on the driven electrons, enabling a

resonance to occur, which results in field enhancement both within the particle and in the near-field region outside of it. This phenomenon is known as localized surface plasmon resonance (LSPR).

To grasp LSPR, the resonance criteria are established by examining how metal nanoparticles interact with an electromagnetic wave. The behavior of a particle with size d in relation to the electromagnetic field can be assessed using the straightforward quasi-static approximation, as long as d << λ i.e. the particle is much smaller than the wavelength of light in the surrounding medium. The geometry and setup considered here is the most simple geometry however upon considering more complex cases it is seen that this simple geometry describes more complex cases very effectively [10]. First, consider a homogenous, isotropic sphere of radius a positioned at the origin within a steady, uniform electric field E = $E_0\hat{z}$ (figure 3.4) (Derivation from [10])

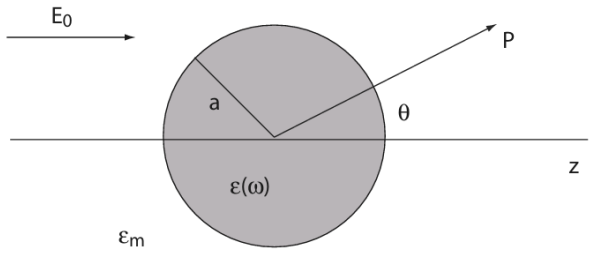


Figure 3.4: Illustration of a homogeneous sphere placed into an electrostatic field taken from [10]

In the electrostatic approach, the solution is described by the Laplace equation $\nabla^2\phi = 0$, allowing the electric field to be expressed as $E = -\nabla\phi$. The problem's inherent azimuthal symmetry suggests that the solution will follow a particular structure;

$$\phi(r,\theta) = \Sigma_{l=0}^{\infty}[A_l r^l + B_l r^{-(l+1)}]P_l(\cos\theta) \tag{3.23}$$

where $P_l(\cos\theta)$ represents the Legendre Polynomials of order l, with θ being the angle formed between the position vector r at point P and the z-axis. To ensure that the potentials do not diverge at the origin, the expressions for the potentials ϕ_{in} within the sphere and ϕ_{out} beyond the sphere can be expressed as

$$\phi_{in}(r,\theta) = \Sigma_{l=0}^{\infty}A_l r^l P_l(\cos\theta) \tag{3.24}$$

$$\phi_{out}(r,\theta) = \Sigma_{l=0}^{\infty}[B_l r^l + C_l r^{-(l+1)}]P_l(\cos\theta) \tag{3.25}$$

The coefficients A_l, B_l, and C_l can now be calculated based on the boundary conditions at r $\to \infty$ and at the surface of the sphere at r = a. The condition that $\phi_{out} \to -E_0 z = -E_0 r\cos\theta$ as r $\to \infty$ implies that $B_l = -E_0$ and $B_l = 0$ for l $\neq 1$. The boundary conditions at r = a determine the remaining coefficients A_l and C_l. The requirement for the equality of the tangential components of the electric field necessitates that

$$-\frac{1}{a}\frac{\partial\phi_{in}}{\partial\theta}\bigg|_{r=a} = -\frac{1}{a}\frac{\partial\phi_{out}}{\partial\theta}\bigg|_{r=a} \tag{3.26}$$

and the equality of the normal components of the displacement field

$$-\epsilon_0\epsilon\frac{\partial\phi_{in}}{\partial r}\bigg|_{r=a} = \epsilon_0\epsilon_m\frac{\partial\phi_{out}}{\partial r}\bigg|_{r=a} \tag{3.27}$$

Applying these boundary conditions results in $A_l = C_l = 0$ for $l \neq 1$, and through the computation of the other coefficients, A_l and C_l for the potentials are determined to [Jackson, 1999]

$$\phi_{in} = -\frac{3\epsilon_m}{\epsilon + 2\epsilon_m}E_0 r\cos\theta \tag{3.28}$$

$$\phi_{out} = -E_0 r\cos\theta + \frac{\epsilon - \epsilon_m}{\epsilon + 2\epsilon_m}E_0 a^3\frac{\cos\theta}{r^2} \tag{3.29}$$

The interpretation of Equation 3.29 is that ϕ_{out} represents the combination of the external field and the field generated by a dipole situated at the center of the particle. To express ϕ_{out} differently, we can incorporate the dipole moment $\mathbf{p}$ as

$$\phi_{out} = -E_0 r\cos\theta + \frac{\mathbf{p}.\mathbf{r}}{4\pi\epsilon_0\epsilon_m r^3} \tag{3.30}$$

$$\mathbf{p} = 4\pi\epsilon_0\epsilon_m a^3\frac{\epsilon - \epsilon_m}{\epsilon + 2\epsilon_m}E_0 \tag{3.31}$$

Hence the applied field induces a dipole moment inside the sphere of magnitude proportional to $|E_0|$. Now using the definition of polarizability α, we have

$$\alpha = 4\pi a^3\frac{\epsilon - \epsilon_m}{\epsilon + 2\epsilon_m} \tag{3.32}$$

The complex polarizability of a small sphere with a diameter below the wavelength is presented

in Equation 3.32 within the electrostatic approximation. Analyzing how polarizability and its phase vary with frequency ω leads to the clear conclusion that there is a resonant enhancement of polarizability when the condition $|\epsilon + 2\epsilon_m|$ reaches a minimum, which for the scenario of small or gradually changing $\mathrm{Im}[\epsilon]$ around the resonance translates to

$$Re[\epsilon(\omega)] = -2\epsilon_m \tag{3.33}$$

This relationship is known as the Fröhlich condition, with the corresponding mode (in an oscillating field) referred to as the dipole surface plasmon or localized surface plasmon. It is important to emphasize that the fundamental physics of localized surface plasmon resonance in a sub-wavelength metallic nanostructure is accurately characterized by this specific case. LSPP resonances significantly influence the behavior of surface plasmon polaritons (SPP) on rough surfaces when their frequency is close to that of the SPP. LSPPs can decay into surface polaritons, which can also be excited by SPP.

In the case of voids in the bulk of a metal, as is the case for the Niobium nanodevices considered in this thesis, LSP frequencies can be found by solving the respective Dirichlet problem. The LSP frequencies of a particle and a void of the same shape are related to each other as

$$\omega^2_{particle} + \omega^2_{void} = \omega^2_p \tag{3.34}$$

Consequently, the resonances linked to surface plasmons concentrated on voids in metals can be inferred from established localized surface plasmons on metal particles.

3.3 Spoof Surface Plasmon Polaritons

Recent technological progress and notable sub-wavelength effects have prompted the investigation of surface plasmon polaritons (SPPs) for various applications, such as super-resolution imaging, integrated optical circuits on chips, biosensing, compact sensors, data storage, photovoltaic systems, graphene-related technologies, and photolithography [12-16]. There is an increasing interest in utilizing the distinctive characteristics of SPPs at lower frequencies to explore numerous integrated circuits and devices within the microwave and terahertz frequency ranges. However, SPP waves are unable to propagate at these frequencies, as metals behave like perfect electric conductors in the terahertz and microwave domains. To

realize SPPs at these lower frequencies, plasmonic metamaterials have been proposed [11, 17-23], which are made up of engineered surfaces that contain holes and grooves on metal surfaces, also known as designer or spoof SPPs.

Materials exhibiting distinct dielectric properties from the plasma form (as shown in equation 3.19) can also host electromagnetic surface modes. While a perfect conductor is typically an exception, it can still be made to support surface modes through the creation of an array of holes in its surface. The subsequent calculations will demonstrate the presence of these surface modes. Let's consider a basic model of a perfect conductor surface that has been pierced by a pattern of holes. For the sake of simplicity, we will assume that the holes have a square cross-section of $a \times a$ and that they are arranged in a square array of side length d (see figure 3.5). (Derivation from [11])

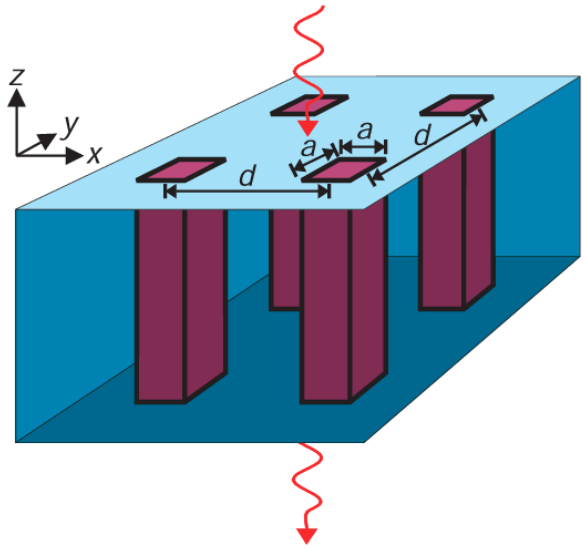


Figure 3.5: The model taken from [11]

The electric and magnetic fields are nonexistent within the conductor, but within the cavities, the electric field exhibits a specific configuration.

$$\mathbf{E} = E_0[0, 1, 0] \sin \frac{\pi x}{a} e^{ik_z z - i\omega t} \quad 0 < x < a, 0 < y < a \tag{3.35}$$

where

$$k_z = i\sqrt{\frac{\pi^2}{a^2 - \epsilon_h \mu_h k_0^2}} \tag{3.36}$$

where ϵ_h and μ_h represent the permittivity and permeability, respectively, of any material

that might occupy the holes. Incoming radiation from outside does not respond to the specifics of the holes, which it cannot distinguish, and instead perceives an average response characterized by ϵ_z, $\epsilon_x = \epsilon_y$, and μ_z, $\mu_x = \mu_y$, for an anisotropic effective homogeneous medium. It is noted that the waveguide mode's dispersion remains unaffected by k_x or k_y in either of the two potential polarizations, and thus $\epsilon_z = \mu_z = \infty$ Assume that the effective homogenous fields in the medium are

$$\mathbf{E}' = E_0'[0, 1, 0]e^{ik_x x + ik_z z - i\omega t} \tag{3.37}$$

It is necessary for k_z to remain identical to that in the waveguide, while k_x is determined by the direction of incidence. In order for this effective field to correspond with the incident and reflected fields outside the surface, equations 3.35 and 3.37 must yield equivalent average fields at the surface. Therefore, following the matching process with the incident and reflected waves,

$$\overline{E_y} = E_0 \frac{a}{d^2} \int_0^a \sin \frac{\pi x}{a}\, dx = E_0 \frac{2a^2}{\pi d^2} = E_0' \tag{3.38}$$

An argument must be made that the instantaneous flow of energy acroos the surface, $(E \times H)_z$, must same inside and outside the surface, both for the real and effective media,

$$(E \times H)_z = \frac{-k_z E_0^2}{\omega \mu_h \mu_0} \frac{a}{d^2} \int_0^a \sin \frac{\pi x}{a}^2\, dx = \frac{-k_z E_0^2}{\omega \mu_h \mu_0} \frac{a^2}{2d^2} = (E' \times H')_z = \frac{-k_z E_0'^2}{\omega \mu_0 \mu_x} \tag{3.39}$$

The intricate details of the fields surrounding the holes correspond with the highly evanescent fields present in the vacuum. Since these fields remain confined to the surface and possess minimal energy, they are often disregarded. By replacing $E_0'^2$, we arrive at the conclusion:

$$\mu_x = \mu_y = \frac{2d^2 \mu_h}{a^2} \left[\frac{2a^2}{\pi d^2} \right]^2 = \frac{8a^2 \mu_h}{\pi^2 d^2} \tag{3.40}$$

Also, note that we know

$$k_z = k_0 \sqrt{\epsilon_y \mu_x} = i \sqrt{\pi^2/a^2 - \epsilon_h \mu_h k_0^2} \tag{3.41}$$

and if c_0 is the velocity of light in vacuo

$$\epsilon_y = \epsilon_x = \frac{1}{\mu_x}\left(\epsilon_h\mu_h - \frac{\pi^2}{a^2k_0^2}\right) = \frac{\pi^2 d^2\epsilon_h}{8a^2}\left(1 - \frac{\pi^2 c_0^2}{a^2\omega^2\epsilon_h\mu_h}\right) \tag{3.42}$$

which is the canonical plasmon form with a plasma frequency of

$$\omega_{pl} = \frac{\pi c_0}{a\sqrt{\epsilon_h\mu_h}} \tag{3.43}$$

Notice that this is a universal finding, and this effective medium suggests a bound surface state. This holds true as long as the distance between the voids is significantly less than the wavelength. Thus, the existence of voids, regardless of their size, leads to a surface plasmon polariton-like bound state.

By analyzing the underlying electrodynamic principles, we aim to determine which plasmonic modes play a dominant role in shaping the optical response of the Niobium nanodevices.

The following section provides a brief introduction to Machine Learning, outlining its fundamental concepts and mathematical framework. Additionally, it presents an overview of the different types of Neural Networks employed in this study, detailing their architectures, working principles, and relevance to modeling plasmonic interactions.

3.4 Machine Learning

ML makes use of statistical techniques and computational algorithms to analyze data, make predictions and model complicated functions without relying heavily on explicit programming instructions. It instead allows algorithms to learn purely from past data or experience. The algorithm autonomously learns the most effective approach to solving the problem by trying to optimize some objective function, representing the task.

ML techniques come under three broad categories [24]:

- **Supervised Learning**: These algorithms focus on learning from data that has been

labeled, where each input is linked to a specific "correct" output label. They examine patterns within the labeled data to learn how to predict or classify new, previously unseen data instances.

- **Unsupervised Learning**: These algorithms operate on unlabeled data, aimed at trying to discover classifications and structures within the data. Without explicit guidance, the algorithms aim to identify groupings among data points, enabling tasks such as clustering or dimensionality reduction

- **Reinforcement Learning**: Reinforcement learning algorithms strive to equip an agent with the ability to make a series of decisions within an interactive setting. The environment gives agents feedback in the form of rewards or penalties as they explore it. By engaging in trial and error, these algorithms develop optimal strategies to enhance cumulative rewards over time.

3.4.1 Neural Networks

A neural network is a computational model that draws inspiration from the organization and functioning of the neural networks found in the human brain. It is made up of connected units known as neurons, which are arranged in layers. These neurons accept inputs, carry out calculations, and generate output signals.

In a typical neural network, there are three main types of layers [25]:

- **Input Layer**: Obtains the initial dataset, which may consist of images, text, or numerical values. Every neuron in the input layer corresponds to a characteristic or property of the input data.

- **Hidden Layer**: These serve as intermediate layers situated between the input and output layers. Every neuron in a hidden layer receives inputs from the neurons in the prior layer, executes a calculation through weighted connections, and transmits the outcome to the neurons in the subsequent layer.

- **Output Layer**: The output layer generates the ultimate result of the neural network, which may be a classification label, a regression value, or another type of prediction.

The count of neurons in the output layer is determined by the specific task that the neural network is intended to accomplish.

Neural networks incorporate activation functions to add non-linearities to the model. In the absence of activation functions, the network would function like a linear regression model, capable only of learning linear relationships between inputs and outputs. A variety of activation functions are utilized in neural networks, each possessing distinct properties and benefits. Among the most common activation functions are sigmoid, tanh, ReLU (Rectified Linear Unit), Leaky ReLU, and softmax. These functions exhibit various shapes and behaviors, enabling them to capture diverse types of non-linearities present in the data.

Neural networks initially learn from training data to identify approximate patterns and relationships within the input data, transforming them into functions. Throughout the training phase, the network modifies its weights and biases based on the pairs of inputs and outputs provided in a training dataset. This modification process, commonly known as backpropagation, seeks to reduce the discrepancy between the network's predictions and the actual outcomes. After training, a neural network can apply its learned knowledge to make predictions on new, previously unseen data. Neural networks have shown impressive success across numerous applications, such as image and speech recognition, natural language processing, medical diagnostics, and autonomous driving, among others [26].

A general representation of a Neural Network Architecture is given in Figure 3.6

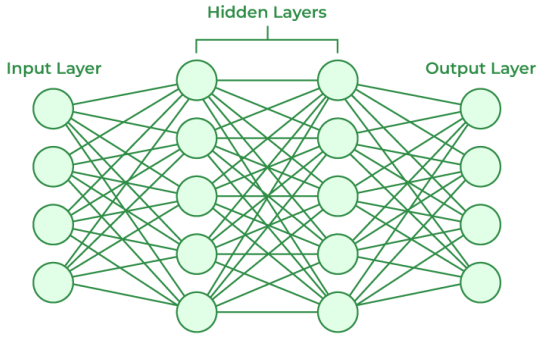


Figure 3.6: General Neural Network Architecture

3.4.2 Simple Feed-Forward Networks

The fundamental neural network architecture can be explained through a sequence of functional changes by formulating M linear combinations of the input variables x_1,...,x_D arranged as follows: (taken from [24])

$$a_j = \Sigma_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{3.44}$$

where j=1,...,M and the superscripts denote the respective parameters in the network's first layer. The parameters $w_{ji}^{(1)}$ are referred to as weights, while the parameters $w_{j0}^{(1)}$ are referred to as biases. The values a_j are termed activations. They are processed through a differentiable, nonlinear activation function $h()$, resulting in $z_j = h(a_j)$. These values represent the outputs of the basis function:

$$y(x, w) = f\left(\Sigma_{j=1}^{M} w_j \phi_j(x)\right) \tag{3.45}$$

According to equation 3.45, the nonlinear h() functions are typically selected to be either sigmoidal or the tanh function. These outcomes are then combined linearly to produce the activations of the output units:

$$a_k = \Sigma_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{3.46}$$

where k=1,...,K and K represents the total number of outputs. This transformation constitutes the second layer of the network, and likewise, $w_{k0}^{(2)}$ refers to the bias parameters. Ultimately, the activations of the output units are processed using a suitable activation function to produce the set of outputs y_k.

To expedite training in the following iterations of a simple feed-forward neural network, it is essential to discover an effective method for calculating the gradient of the error function E(w). This can be accomplished by utilizing a local message-passing approach where information is transmitted alternately in the forward and backward directions throughout the network, referred to as Error Backpropagation or simply Backprop. The steps involved in the backpropagation process can be outlined as follows:

1. Input a vector x_n into the network and propagate it forward through the network to

determine the activations of all hidden and output units.

2. Evaluate the errors $\delta_k \equiv \frac{\partial E_n}{\partial a_j}$ for all the output units

3. Backpropogate the $\delta' s$ using the backpropogation formula as:

$$\delta_j = h'(a_j)\Sigma_k w_{kj}\delta_k \tag{3.47}$$

to calculate δ_j for every hidden unit within the network

3.4.3 Graphical Neural Networks

Graph Neural Networks (GNNs) represent a category of deep learning models tailored for analyzing data organized in graph structures. Unlike traditional neural networks that handle grid-like datasets such as images or sequences, GNNs bring deep learning capabilities to non-Euclidean spaces, making them especially valuable for issues where the connections between entities are significant.

A graph G = (V,E) comprises a collection of nodes (or vertices) V, and edges E, which establish pairwise connections among the nodes. Each node $v \in V$ may have corresponding feature vectors h_v, and the edges e_{uv} may also possess attributes that detail the nature of interactions between the nodes. GNNs capitalize on these graph architectures to derive complex, node-specific representations by disseminating information through the edges.

Graph Neural Networks have become widely adopted in numerous scientific and industrial fields due to their proficiency in modeling intricate relationships within data. In computational physics, GNNs have been utilized to investigate molecular interactions and forecast material properties by learning from atomic graphs [28]. In the realm of bioinformatics, they aid in predicting protein-protein interactions and facilitating drug discovery by capturing structural and functional relationships within biological networks [29]. GNNs have also found applications in computer vision, particularly for generating scene graphs [30].

A significant characteristic of GNNs is their capacity to execute message passing, where every node refines its representation by sharing information with its neighboring nodes. This iterative procedure enables GNNs to grasp both local and global graph patterns, which makes them especially effective for applications such as node classification, link prediction, and

graph regression. Different architectures like Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and Message Passing Neural Networks (MPNNs) have been created to improve the learning process through various aggregation and update strategies.

In each iteration of message passing within a GNN, a hidden embedding $h_u^{(k)}$ for each node $u \in V$ is modified based on information collected from the neighborhood of node u, denoted as N(u) (figure 3.7) (Derivation from [26])

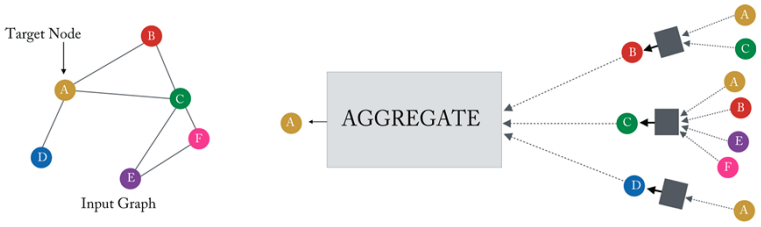


Figure 3.7: Message-passing in GNN's from [26]

This message-passing update can be expressed as follows:

$$h_u^{k+1} = UPDATE^{(k)}(h_u^k, AGGREGATE^{(k)}([h_v^k, \forall v \in N(u)]) = UPDATE^{(k)}(h_u^k, m_{N(u)}^k) \tag{3.48}$$

where UPDATE and AGGREGATE are arbitrary differentiable functions (i.e., neural networks) and $m_{N(u)}$ is the 'message' that is aggregated from u's graph neighbourhood N(u).

This is a fairly abstract framework of GNNs as a series of message-passing iterations using UPDATE and AGGREGATE functions. The following describes the most basic GNN framework.

The basic GNN message passing is defined as

$$h_u^k = \sigma\left(W_{self}^k h_u^{k-1} + W_{neigh}^k \Sigma_{v \in N(u)} h_v^{k-1} + b^k\right) \tag{3.49}$$

where $W_{self}^k, W_{neigh}^k \in \mathbb{R}^{d^k \times d^{k-1}}$ represent trainable parameter matrices, and σ indicates an element-wise nonlinearity (such as a tanh or ReLU function). The message-passing process in the fundamental GNN framework is similar to that of a conventional multi-layer perceptron (MLP), as it depends on linear operations followed by a single element-wise nonlinearity.

We can equivalently define the basic GNN through the UPDATE and AGGREGATE

functions;

$$m_{N(u)} = \Sigma_{v \in N(u)} h_v \tag{3.50}$$

$$UPDATE(h_u.m_{N(u)}) = \sigma(W_{self} h_u + W neigh m_{N(u)}) \tag{3.51}$$

where we use

$$m_{N(u)} = AGGREGATE^k([h_v^k, \forall v \in N(u)]) \tag{3.52}$$

3.4.4 Transfomers

The Transformer model, introduced by [31], revolutionized deep learning by replacing recurrent architectures with a self-attention mechanism, enabling highly parallelizable and context-aware representations. Unlike traditional sequence models, Transformers process entire input sequences simultaneously, making them highly efficient for tasks requiring long-range dependencies. The key innovation lies in the self-attention mechanism, which allows the model to dynamically weigh the importance of different input tokens, capturing complex relationships without relying on sequential processing. Transformers have since become the foundation of state-of-the-art architectures in natural language processing (NLP), computer vision, and scientific computing.

The versatility of Transformer models extends across multiple domains. In natural language processing, models like BERT [32], have achieved breakthroughs in tasks such as text generation, machine translation, and question-answering. Transformers have also been applied in drug discovery and bioinformatics, where models such as AlphaFold [33], leverage attention mechanisms to predict protein folding structures with unprecedented accuracy.

The Transformer architecture referenced in [31] is summarized here briefly. The foundational Transformer architecture consists of six stacked layers. The output generated by layer l serves as the input for layer l+1 until the ultimate prediction is achieved. On the left side, there exists a six-layer encoder stack, while on the right side, there is a six-layer decoder stack:

In this process, the encoder transforms an input sequence of symbol representations $(x_1, ...x_n)$ into a sequence of continuous representations $\mathbf{z} = (z_1, ...z_n)$. Based on $\mathbf{z}$, the decoder produces an output sequence $(y_1, ..., y_m)$ of symbols, generating one element at a time.

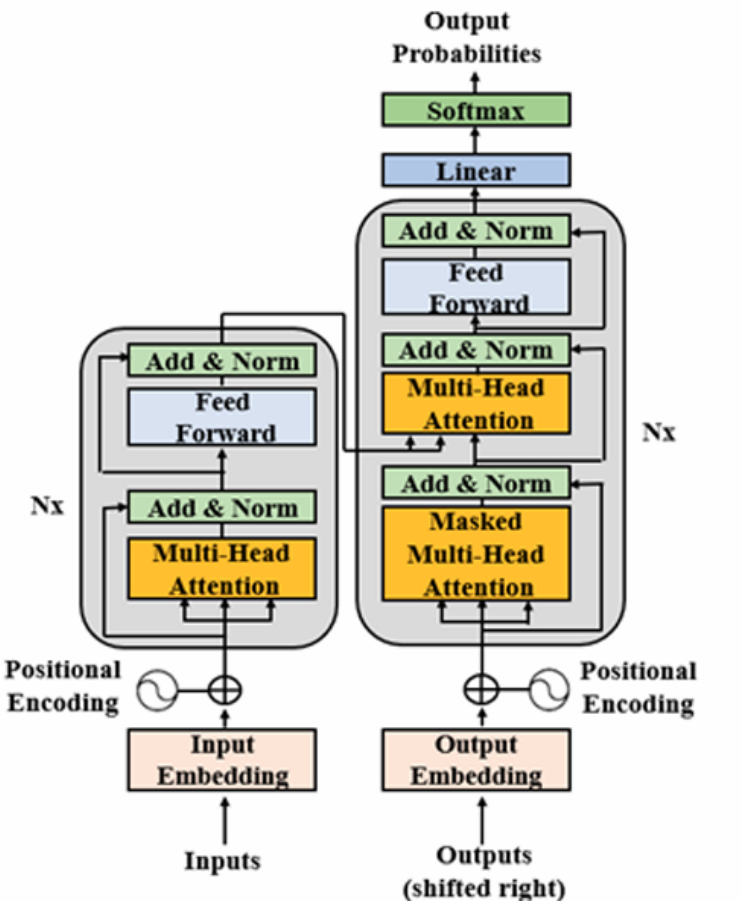


Figure 3.8: Message-passing in GNN's from [26]

1. **Encoder stacks**: The encoder consists of a series of N = 6 identical layers. Each of these layers features two sub-layers. The first sub-layer employs a multi-head self-attention mechanism, while the second consists of a straightforward, positionwise fully connected feed-forward network. The output of each sub-layer is calculated as $LayerNorm(x + SubLayer(x))$, where $SubLayer(x)$ denotes the function performed by that specific sub-layer.

2. **Decoder Stacks**: The decoder consists of a series of N = 6 identical layers. Besides the two sub-layers present in each encoder layer, the decoder includes an additional third sub-layer that carries out multi-head attention over the encoder stack's output.

An attention mechanism can be described as converting a query alongside a set of key-value pairs into a result, with the query, keys, values, and result all being represented as vectors. The result is computed as the weighted sum of the values, with the weight for each value

coming from a compatibility function that evaluates the relationship between the query and the corresponding key.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.53}$$

where Q,K and V are the queries, keys and values matrices.

There are several other components to the Transformer model described above including Position-wise FFN's, Multi-head attention, Embeddings etc but the main components are as given above.

3.4.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing grid-like data, such as images and structured spatial data. Inspired by the human visual system, CNNs utilize convolutional layers that apply learnable filters to detect spatial hierarchies of patterns, ranging from simple edges to complex textures. Unlike fully connected networks, CNNs exploit spatial locality through weight sharing, significantly reducing the number of parameters and improving computational efficiency. By leveraging key operations such as convolution, pooling, and non-linear activations, CNNs extract robust feature representations, making them highly effective for tasks involving spatial relationships [34].

CNNs have been instrumental in computer vision, powering applications such as image classification [35], object detection [36]. In medical imaging, CNNs assist in disease diagnosis by analyzing MRI and CT scans for anomalies such as tumors [37].

A convolution operation is generally defined as an operation on two functions of a real valued argument (Derivation from [25]):

$$s(t) = \int x(a)w(t-a)\,da \tag{3.54}$$

In convolution terminology, the first argument (here x) is called as the input, while the second argument (here w) is called the kernel and the output is sometimes called the feature map. If

x and w are defined only on integer t, then its known as a discrete convolution:

$$s(t) = (x * w)(t) = \Sigma_{a=-\infty}^{\infty} x(a)w(t-a) \tag{3.55}$$

Typically, the input consists of a multi-dimensional data array, while the kernel is also a multi-dimensional array of parameters that the learning algorithm adjusts. Furthermore, convolutions frequently operate across multiple axes simultaneously. For instance, when a two-dimensional image I serves as the input, a two-dimensional kernel K is generally required as well:

$$S(i,j) = (I * K)(i,j) = \Sigma_m \Sigma_n I(m,n)K(i-m, j-n) \tag{3.56}$$

Convolution is commutative, hence equivalently:

$$S(i,j) = (K * I)(i,j) = \Sigma_m \Sigma_n I(i-m, j-n)K(m,n) \tag{3.57}$$

A standard convolution layer comprises three key stages. In the initial stage, the layer conducts multiple convolutions simultaneously to generate a collection of linear activations. During the second stage, each of these linear activations is processed through a nonlinear activation function, such as the rectified linear activation function, often referred to as the detector stage. In the final stage, a pooling function is applied to further adjust the layer's output.

A pooling function substitutes the output of the network at a specific location with a summary statistic derived from the surrounding outputs. For instance, the max pooling operation identifies the highest output within a defined rectangular area. In all instances, pooling contributes to making the representation roughly invariant to minor shifts in the input.

This outlines the essential background for all the neural network types employed in this research. A significant optimization aspect shared among all the neural networks in this study was Optuna.

3.4.6 Optuna

Hyperparameter tuning is a critical step in optimizing the performance of machine learning models, particularly deep learning architectures. Optuna is an open-source, automatic hyperparameter optimization framework designed to efficiently search for the best set of hyperparameters through various optimization strategies. Unlike traditional grid search or random search methods, Optuna leverages sequential model-based optimization (SMBO) and techniques like Tree-structured Parzen Estimator (TPE), Bayesian Optimization, and pruning mechanisms to dynamically explore the hyperparameter space while reducing computational costs [38]. By providing a flexible and user-friendly interface, Optuna enables researchers to define objective functions, search spaces, and sampling strategies, making it a powerful tool for optimizing deep learning models.

In this research, Optuna was employed to optimize the hyperparameters of the above described neural network architectures. Given the complexity of these models and their sensitivity to hyperparameters such as learning rate, batch size, number of layers, and activation functions, a systematic optimization approach was essential to achieve optimal predictive performance. Traditional methods like grid search are computationally expensive and inefficient for high-dimensional search spaces, whereas Optuna's adaptive sampling and pruning techniques significantly reduced training time while ensuring convergence to the best-performing hyperparameter configurations. This automated tuning process was instrumental in enhancing model accuracy and generalization while minimizing overfitting. The use of Optuna thus played a pivotal role in refining the predictive capabilities of the neural networks used in this study.

Optuna uses a define-by-run API [38](an example of which is given in Figure 3.9)

In a define-by-run API, users aren't required to completely outline every detail of the optimization strategy upfront. Optuna treats hyperparameter optimization as the task of minimizing or maximizing an objective function that accepts a set of hyperparameters and returns a validation score. Each optimization process in Optuna is known as a study, while each evaluation of the objective function is referred to as a trial. Through interactions with the trial object, Optuna progressively develops the objective function. The trial object's methods create the search spaces dynamically during the execution of the objective function.

This is the basic functionality tool that was used throughout this study and almost all the

```python
def objective(trial):
    #Hyperparameter choice
    n_hidden1 = trial.suggest_int("n_hidden1", 32, 128, step=16)
    n_hidden2 = trial.suggest_int("n_hidden2", 16, 64, step=16)
    dropout = trial.suggest_float("dropout", 0.2, 0.5)
    learning_rate = trial.suggest_float("lr", 1e-4, 1e-2, log=True)
    weight_decay = trial.suggest_float("weight_decay", 1e-5, 1e-3, log=True)

    #Initialize model,optimizer,and loss function
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model = GNN(n_hidden1=n_hidden1, n_hidden2=n_hidden2, dropout=dropout).to(device)
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate, weight_decay=weight_decay)
    criterion = torch.nn.MSELoss()

    #Use DataLoader for batching
    loader = DataLoader(data_list, batch_size=4, shuffle=True)

    #Train the model for a few epochs
    model.train()
    for epoch in range(20):  #Train for 20 epochs during optimization
        epoch_loss = 0
        for batch in loader:
            batch = batch.to(device)
            optimizer.zero_grad()
            out = model(batch)
            loss = criterion(out, batch.y)
            loss.backward()
            optimizer.step()
            epoch_loss += loss.item()

    #Validation loss (use train data for simplicity, or split a validation set)
    val_loss = epoch_loss / len(loader)
    return val_loss

# === 6. Run Optuna Study ===
study = optuna.create_study(direction="minimize")
study.optimize(objective, n_trials=15)  #15 trials for optimization
```

Figure 3.9: An example code of Optuna's define-by-run API taken from one of the codes developed during this study

results (predictions from the NN's) obtained are with the help of this optimization framework.

Chapter 4

Surface-Corrugated Niobium nanodevices

This chapter focuses on niobium-based nanodevices and their role in predicting extinction spectra based on surface corrugations and their resulting electromagnetic properties. The study aims to analyze how engineered surface structures influence plasmonic behavior, thereby enabling precise tuning of optical responses for various applications.

In particular, this chapter introduces chip-integrated, visible-light communication-band modulators based on niobium (Nb) plasmonic nanoantenna arrays and split-ring resonators (SRRs). These nanostructures support distinct types of plasmonic excitations, which play a fundamental role in shaping the optical characteristics of the devices. By leveraging periodic and aperiodic corrugations, these devices exhibit enhanced light-matter interactions, making them promising candidates for applications in photonic computing, quantum technologies, and beyond [6-8].

The experimental fabrication and real-world physical designs of these nanodevices were carried out by my supervisor and their research team. My contribution to this study is primarily theoretical and computational, employing numerical simulations and machine learning techniques to analyze the optical response of these structures. Utilizing the well-established background on plasmons and their various types in chapter 3, along with experimental findings from the research team, I aim to decipher the specific plasmonic modes excited in these niobium nanodevices in this chapter. This involves determining whether localized surface

plasmon polaritons (LSPPs), spoof surface plasmon polaritons (SSPPs), or conventional SPPs dominate the optical response based on structural and material properties.

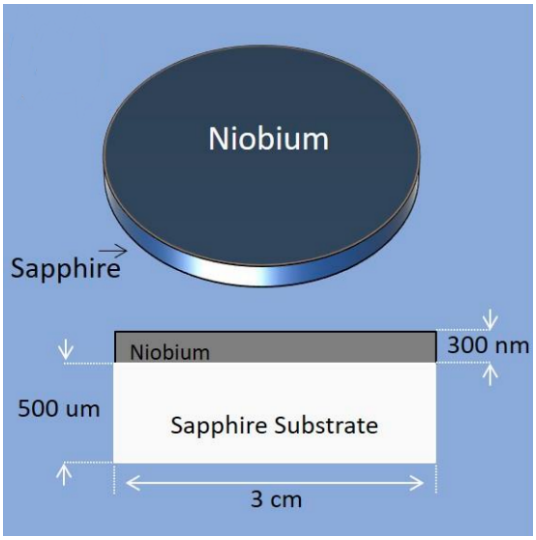
Additionally, a significant aspect of this work is evaluating the predictive capabilities of different neural network architectures in modeling extinction spectra. By incorporating physical and electromagnetic features of the nanodevices as input parameters, various deep learning models, including convolutional neural networks (CNNs), graph neural networks (GNNs), and transformer-based architectures, are assessed for their ability to accurately predict plasmonic responses in the next chapters. This computational approach not only provides insights into the fundamental physics governing plasmonic interactions in niobium-based metamaterials but also demonstrates the potential of machine learning in accelerating nanophotonic device design and optimization.

4.1 Niobium Nano-device arrays

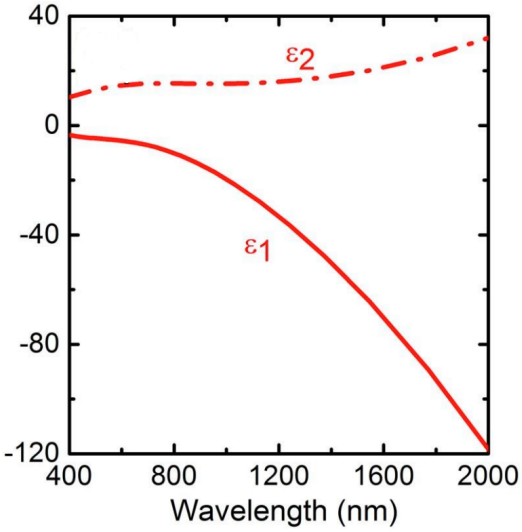
The electrodynamic properties of niobium (Nb) and aluminum (Al) have been investigated through both theoretical and experimental approaches, categorizing them as metallic superconductors. Metallic superconductors serve as essential components for low-temperature communication circuits and superconducting quantum technologies. Most research conducted on this technology has focused on the RF, microwave, and terahertz (THz) frequency ranges, operating under the assumption that photons with energies exceeding several meV—varying with the material—disrupt the superconducting energy gap, preventing the formation of Cooper pairs and consequently diminishing superconductivity.

Niobium-based devices exhibit intriguing optoelectronic characteristics when exposed to visible and near-infrared light. The fabrication of these devices involved the use of subwavelength nanocircuit arrays applied to the surface of Niobium thin films. A Niobium thin film with a thickness of 300 nm was deposited onto a Sapphire substrate measuring 30 mm in diameter and 0.5 mm thick.

Variably angled spectroscopic ellipsometry measurements were conducted across a wavelength range of 200 nm to 2000 nm at room temperature to investigate the optical properties of unstructured niobium and to facilitate the simulation of the photoresponse of niobium nanodevices in COMSOL (as detailed in chapter 5).



(a) Illustration of the Nb chip



(b) Complex permittivity of the 300nm thickness Nb chip

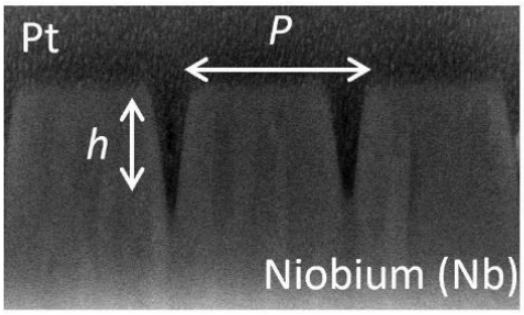
Figure 4.1: Images taken from [6] with permission

In Figure 4.1, the plasmonic characteristics of Nb and the observed negative permittivity across UV to NIR frequency ranges are presented, along with the experimental setup of the Nb chip. The superconducting temperature T_c for the unstructured Nb chips was validated to be approximately 9K via DC transport measurements.

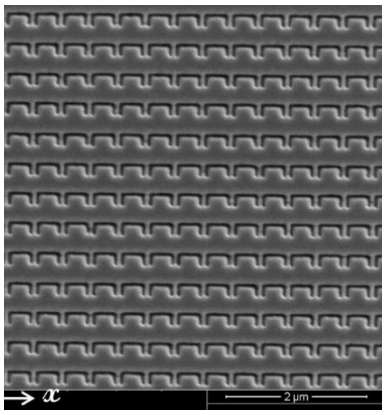
The nano-antenna arrays were fabricated using focused ion beam milling (FIB) with the FEI Helios NanoLab 600. Five chips were created, each featuring an array of Nb subwavelength components with varying geometries, and each array occupies an overall area of 100×100 μm^2.

Four of the chips were designed with a nano-antenna (slit) configuration, while the fifth chip was created in a Split Ring Resonator design, as illustrated in Figures 4.2.

The nano-antenna arrays exhibit plasmonic colors when exposed to white light. The plasmonic colors of the optical nano-antenna arrays vary according to the length of each meta-atom and the polarization of the incident light. The extinction spectra $A(\lambda) = 1 - R(\lambda)$ (with transmission at around 10^{-8} being negligible) are observed at the visible light communication wavelengths, revealing that the resonances shift to longer wavelengths (redshift) as the nano-antenna length increases. For the P_y light polarization, the extinction spectra demonstrate significantly higher resonance compared to the P_x polarization. Moreover, it was observed



(a) Nanoantenna array slits (Pt layer was deposited to protect it during sectioning)



(b) SRR slits

Figure 4.2: Images taken from [6] with permission

that the light modulation in the Nb nanodevices was achieved by adjusting the geometrical parameters of the meta-atoms.

The modulation depth is defined as

$$MD = \frac{A_{Nb\,nanoantenna\,array} - A_{unpatterned\,Nb\,film}}{A_{Nb\,nanoantenna\,array}} \tag{4.1}$$

where $A_{unpatterned\,Nb\,film}$ and $A_{Nb\,nanoantenna\,array}$ are the maximum extinctions of the unpatterned film and nanoantenna arrays with lengths varying from L = 200nm to 350nm. An apparent increase in MD is observed for increasing length of nanoantenna arrays and reaches a maximum value of 60% for L = 350nm at $\lambda = 716$nm as shown. For P_x polarization, no resonant peaks are observed as well as no resonant peaks in unstructured Nb film for either polarization indicating the importance of the surface-corrugations in light-matter interactions.

The fifth chip exhibiting the SRR geometry displays a notable shift in plasmonic color akin to the nanoantenna array when subjected to P_y polarization, contrary to observations under P_x polarization. All five chips underwent measurements in an optical cryostat using a microspectrophotometer at room temperature. The nano-split ring resonator arrays exhibit dipole-like plasmon modes localized primarily around the side arms of the SRR, while the nano-antenna arrays are primarily influenced by vertical dipoles. The nano-split ring resonator array chip demonstrates a highly sensitive photoresponse to the polarization of incident light, achieving nearly 80% absorptance of visible light across a wide wavelength

range from λ = 400nm to 800nm, with a peak absorptance exceeding 90% at a resonance wavelength of λ=650nm. This makes it an excellent candidate for integration with Nb-based superconducting quantum hardware and elements of visible light communication systems, such as polymer optical fiber (POF), due to their durability against bending and stretching.

The optical responses of all 5 devices are shown in Figures 4.3.

4.2 Deciphering the plasmonic mode type

Based on the all the background in chapter 3 as well as the experimental results in the previous section, certain things can be deduced systematically quite elegantly, such as the type of plasmonic modes in the Nb nanodevices.

The three types of possible plasmonic modes are SPP's, LSPP's and SSPP's. The following subsections will explain whether a plasmonic mode is possible in the nanodevices.

4.2.1 SPPs in nanodevices

One of the main things about surface plasmon polaritons is that they require wavevector matching between the incident light wavevector and the SPP wavevector. This happens because the wavevector of light in free space is smaller than k_{SPP}. Hence to match them, coupling mechanism such as prisms or diffraction gratings are required.

The coupling efficiency depends quite highly on the angle of incidence (θ), which affects $k_x = k_0 \sin\theta$. Only at a particular angle of incidence, the wavevectors can be matched and hence lead to surface plasmon polariton resonance as was seen in the extinction spectra.

However, all the above experiments was done at a 0° angle of incidence and no coupling mechanism were used. Even though, the incident light wavelength range matched the SPR required range of Vis-NI range as well as the p-polarization requirement which was also satisfied, due to the above contradiction, SPPs are extremely unlikely to be the main cause of the resonance and plasmonic modes observed in the device.

On the other hand, certain plasmonic modes are found to decay into SPPs in the bulk of

the metal which in turn leads to hybridized plasmon states, hence making SPPs a secondary cause of any observed resonances, the main cause for the resonances is yet to be found.

4.2.2 SSPPs in nanodevices

As explained in chapter 3, spoof surface plasmon polaritons are engineered that mimic the behavior of traditional SPPs. The engineering of SSPPs is primarily due to the periodic or quasi-periodic or aperiodic surface structures made by the user in their nanodevice. The periodicity of the voids is what accounts for their observed behavior.

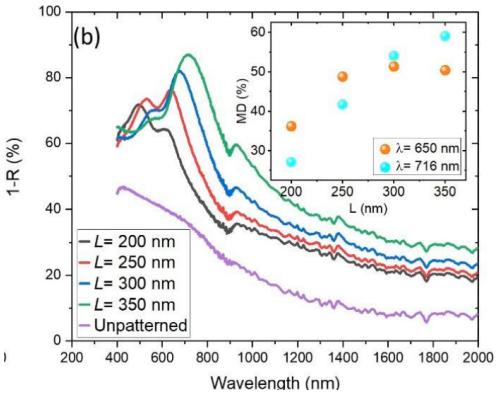
Additionally, SSPPs exhibit electromagnetic field confinement near the structured surface. Hence, the dipole-like behavior as observed in the paper might be akin to this phenomenon of SSPPs. Even though the evident stark similarity of the SSPPs to the experimental observations as well as their sole purpose being mimicking the behavior of SPPs in specially engineered metamaterials might be enough to hypothesize their formation in the Nb nanodevices, unfortunately the hypothesis will be incorrect.

This is because, SSPPs typically respond to light in the centimeters to millimeter wavelength range i.e., microwave to THz range which is far from the experimentally used Vis-NI range. Furthermore, the periodicity of the structures is strictly required to be << than the wavelength of the incident radiation. The periodicity of the nanoantennas and the SRRs is in the range of 300-800nm which is comparable or higher than the wavelength of the incident radiation.

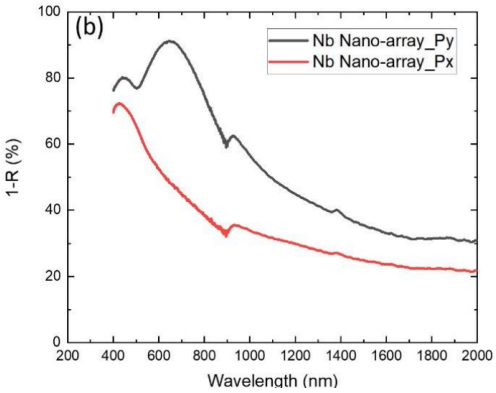
Both these facts make it decidedly clear that SSPPs are not the main cause of the observed resonances in the nanodevices.

4.2.3 LSPPs in nanodevices

LSPPs are confined oscillations of conduction electrons localized to nanoscale structures, such as nanoparticles or nanoantennas. Contrary to their name, localized surface plasmon polaritons can also occur in voids and deformities of particular sizes satisfying particular conditions.



(a) Extinction spectra of nano-antenna arrays for different lengths under P_y polarization



(b) Extinction spectra of nano-split ring resonator array

Figure 4.3: Images taken from [6] with permission

LSPPs are significantly dependent on the polarization of the incident light, meaning the intensity and resonant wavelength of the plasmon excitation will vary depending on the polarization direction relative to the shape and orientation of the nanostructure involved; essentially, the LSPP is most efficiently excited when the polarization aligns with the axis of the nanoparticle along which the electrons can oscillate most freely. Additionally, LSPPs show resonance in the Vis-NI range of radiation as well as do not depend on the angle of incidence.

Most significantly, LSPPs show dipole-like behavior in the electric and magnetic-field distributions at their resonant wavelengths. The electric field localization is especially high near sharp corners or edges as is observed in the SRR or nanoantennas geometry.

This is compelling evidence for the fact that the main cause of the resonance in extinction spectra observed in the Nb nanodevices is due to localized surface plasmon polaritons. There could also be different secondary factors that are affecting this as discussed previously, but the main cause is LSPPs.

Chapter 5

Simulation software and data generation

This chapter explores the simulation software utilized to replicate experimental results within controlled environments. The accurate modeling of nanophotonic systems, particularly surface-corrugated Niobium nanodevices, requires robust tools capable of handling the complexity of electrodynamic calculations at the nanoscale. Several platforms were explored, including PyGDM, Tidy3D, and COMSOL Multiphysics, each with unique methodologies and strengths.

PyGDM leverages the Green's Dyadic Method (GDM) to simulate optical responses by modeling dipole interactions, while Tidy3D employs the Finite-Difference Time-Domain (FDTD) approach for time-resolved electromagnetic analysis. Despite their strengths, these tools presented challenges such as computational inefficiency or lack of flexibility for complex geometries. Ultimately, COMSOL Multiphysics, which uses the Finite Element Method (FEM), emerged as the most reliable and versatile tool due to its ability to handle intricate designs and diverse physical environments. This chapter also highlights the learning process involved, including overcoming the steep learning curves of these platforms to effectively simulate and analyze nanostructures.

To ensure the accuracy of simulations, results were validated against experimental data provided by the research team. Extinction spectra and electromagnetic field distributions were replicated for specific geometries under varying polarizations. Metrics like Mean Squared Error (MSE) and the Kendall Tau Correlation Coefficient demonstrated a high degree of

agreement, confirming the reliability of the simulation data.

Finally, this chapter emphasizes the role of simulation in generating robust training datasets for neural networks. By accurately modeling the interaction between physical features and optical properties, the simulation results form the foundation for machine learning models, enabling reliable predictions of extinction spectra based on geometric and electromagnetic features.

5.1 Learning different simulation software

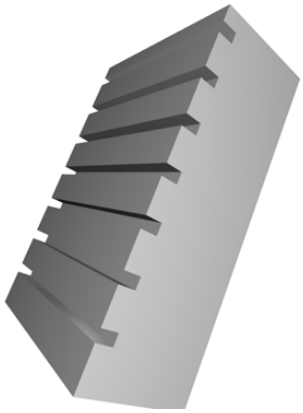
Simulating complex electrodynamic interactions at the nanometer scale requires a robust and reliable computational tool. Given the need for accurate predictions of optical behavior in surface-corrugated Niobium nanodevices, several simulation software packages were explored. Each of these software solutions employs different numerical techniques, impacting their efficiency, usability, and accuracy.

5.1.1 PyGDM: Green Dyadic Method (GDM)

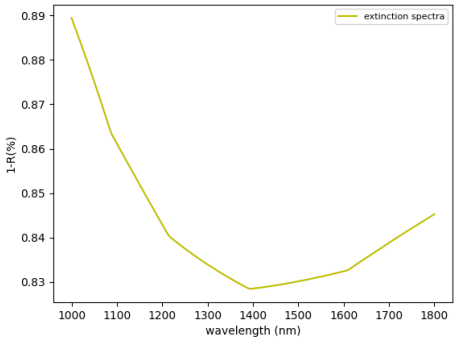
PyGDM is a fully Python-based simulation package that utilizes the Green Dyadic Method to solve Maxwell's equations. This method is particularly useful for computing the optical response of nanostructures. The advantage of PyGDM lies in its flexibility and complete integration with Python, allowing easy customization and automation. However, its limitations include:

- **High computational cost**: Simulating even simple nanostructures requires extensive computational resources.
- **Code complexity**: Implementing custom geometries requires significant coding effort.
- **Debugging overhead**: Error handling in PyGDM can be tedious, making debugging a time-consuming process.

A sample simulation in PyGDM involved a self-designed CAD aluminum model with no substrate, where p-polarized light was incident from the -z direction. The extinction spectra



(a) CAD Model with aluminum properties



(b) Simulated extinction spectra

Figure 5.1: Simulation with PyGDM

for this model were generated, but even this relatively simple simulation required multiple hours due to debugging and computational inefficiencies. The model along with the simulated extinction spectra are given in Figure 5.1

The general equation behind the GDM principle is the iterative field calculation which is the total electric field solved iteratively for all the dipoles in the system [39]:

$$E_{total,i} = E_{inc,i} + \Sigma_{j \neq i} G(r_i, r_j) p_j \tag{5.1}$$

where $G(r_i, r_j)$ is the Green's function describing the interaction between dipoles.

5.1.2 Tidy3D: Finite-Difference Time-Domain (FDTD) Method

Tidy3D is an effective simulation tool that employs the Finite-Difference Time-Domain (FDTD) method, a numerical approach commonly utilized in computational electromagnetics. It features an intuitive interface with a tailored GUI and scripting capabilities based on Python, making it user-friendly for both novices and experienced users.

The FDTD method solves Maxwell's equations in the time domain by discretizing both space and time [40]. It calculates the evolution of electric (E) and magnetic (H) fields

iteratively using the following core equations:

$$\frac{\partial E}{\partial t} + \frac{\sigma}{\epsilon}E = \frac{1}{\epsilon}\nabla \times H \tag{5.2}$$

$$\frac{\partial H}{\partial t} = -\frac{1}{\mu}\nabla \times E \tag{5.3}$$

These core equations are converted into their discretized iterative form as:

$$\frac{E_x^{n+1} - E_x^n}{\Delta t} = \frac{1}{\epsilon}\left(\frac{H_z^{n+1/2} - H_z^{n-1/2}}{\Delta y} - \frac{H_y^{n+1/2} - H_y^{n-1/2}}{\Delta z}\right) \tag{5.4}$$

$$\frac{H_x^{n+1/2} - H_x^{n-1/2}}{\Delta t} = \frac{1}{\mu}\left(\frac{E_y^n - E_y^{n-1}}{\Delta z} - \frac{E_z^n - E_z^{n-1}}{\Delta y}\right) \tag{5.5}$$

Tidy3D applies these equations on a discretized grid, allowing for time-stepped simulations of complex optical phenomena. Its strengths include:

- **Efficient handling of large − scale problems**: FDTD simulations are well-suited for optical nanostructures.
- **User − friendly**: The graphical user interface simplifies model setup.
- **Versatile meshing**: Adaptive meshing enhances computational efficiency.

However, despite its advantages, the computational overhead and meshing constraints made it less suitable for rapid iteration, especially compared to COMSOL.

5.1.3 COMSOL Multiphysics: The final chosen platform

After thorough evaluation, COMSOL Multiphysics emerged as the primary simulation platform for this study due to its exceptional versatility and accuracy. COMSOL utilizes the Finite Element Method (FEM) to solve complex electrodynamic problems, making it well-suited for analyzing intricate nanostructures and their optical responses.

The FEM is based on discretizing a continuous domain into smaller, simpler elements and solving partial differential equations (PDEs) over these elements. In the context of

electrodynamics, the Maxwell equations form the core of FEM modeling, where the electric field E and magnetic field H are related through:

$$\nabla \times E = -\mu \frac{\partial H}{\partial t} \tag{5.6}$$

$$\nabla \times H = \epsilon \frac{\partial E}{\partial t} + J \tag{5.7}$$

FEM breaks down these equations into smaller, element-specific algebraic equations using basis functions, which approximate E and H. The variational form used by FEM is derived from Maxwell's equations, expressed as:

$$\int_{\Omega} (\nabla \times E) \cdot (\nabla \times E')\, d\Omega = \omega^2 \int_{\Omega} \epsilon E \cdot E'\, d\Omega \tag{5.8}$$

where Ω is the domain, and E' represents the test function.

Key advantages of COMSOL Multiphysics include:

- **Graphical User Interface**: A user-friendly interface that simplifies the construction, visualization, and management of simulation models.
- **Multiphysics Integration**: Enables seamless coupling of electromagnetic simulations with other physical phenomena, such as thermal or structural mechanics, enhancing the scope of analysis.
- **Customizability**: Supports extensive parameter sweeps, design optimizations, and automation through integration with Python and MATLAB, offering unparalleled flexibility for advanced studies.

Although mastering COMSOL required considerable time and effort, the software's intuitive interface and robust capabilities ultimately enabled precise and efficient modeling. Its superior speed, accuracy, and ability to handle complex geometries and physical interactions made it the ideal choice for this research, particularly when generating training data for neural network models.

5.2 Inverse Design vs Regular Design Neural Networks

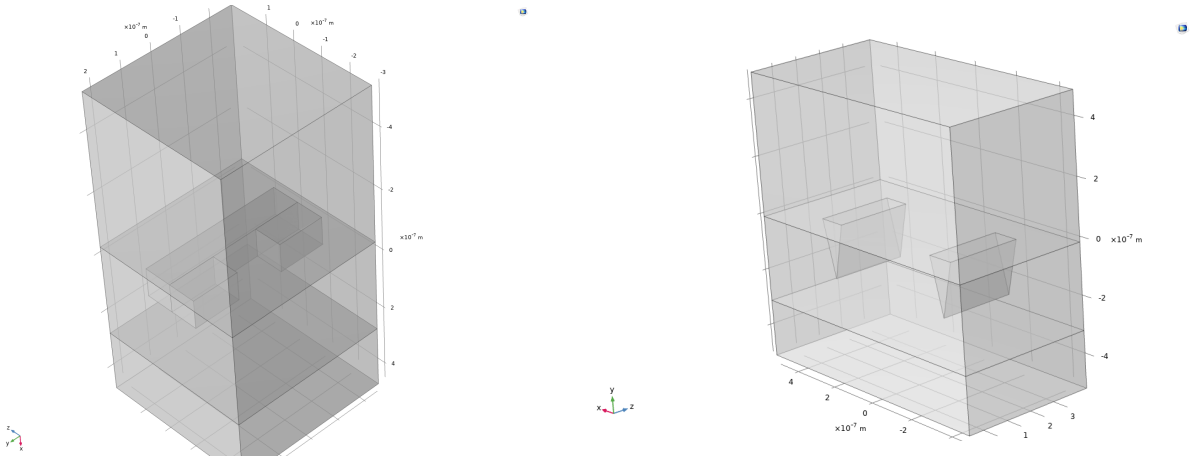
After achieving proficiency in COMSOL, two potential approaches for integrating neural networks with simulation results were explored:

1. **Inverse Design Neural Network**:
    - Input: Extinction spectra or vector components of the electric/magnetic fields.
    - Output: Voxel grid coordinates, which, when arranged, reconstruct the 3D model of the device.
    - Challenges: Requires and extensive training dataset. Achieving high accuracy is difficult due to the complexity of the inverse problem.
2. **Regular Design Neural Network**:
    - Input: Several physical and electromagnetic features of the nanodevice, including voxel grid coordinates.
    - Output: Extinction spectra of the nanodevice
    - Challenges: Easier to implement and train compared to inverse design. Accuracy improves more efficiently with increased training data.

After an extensive literature survey [4,5,42-46] the **Regular Design** approach was chosen due to its higher feasibility and faster convergence.

5.3 Replicating Experimental Results

To create the training dataset required for training the different Neural Networks explored in this study, the training dataset and the method of preparing it must be proven reliable and accurate. This implies validating the simulation environment by comparing it with the experimental data. Thus the Niobium devices were created in the COMSOL simulation environment as shown in Figure 5.2. The devices were tested in identical conditions as mentioned in the paper [6-8], for the same wavelength range and for 2 different polarizations. The simulated and experimental results were then compared both visually and statistically.



(a) Model of SRR

(b) Model of Nanoantennas

Figure 5.2: Simulated models of the Nb nanodevices

5.3.1 Validation Process

While COMSOL Multiphysics offers powerful capabilities for simulating nano-optical systems, certain challenges arise when replicating real-world nanodevice setups. In many cases, the precise geometrical or material parameters of the devices are not explicitly provided in the literature, particularly when confidentiality is involved. This necessitated the deduction of unknown parameters through indirect methods.

For example, when working with nanoantennas, the exact dimensions and configurations were clearly specified in the referenced papers, making the replication process relatively straightforward, though still labor-intensive. These designs could be directly modeled in COMSOL with minimal trial and error, as the given specifications were accurate enough to yield results closely matching the experimental data.

However, the scenario was significantly more complex when simulating the Split Ring Resonator (SRR) nanodevice. In this case, none of the required geometric parameters—such as arm width, depth, and length—were provided due to confidentiality restrictions. The only reference was a scanning electron microscope (SEM) image of the device, which offered an approximate visualization of the geometry but lacked the precise numerical details needed for accurate simulations.

To bridge this gap, the initial dimensions of the SRR were estimated based on the SEM

image. While this provided a starting point, obtaining the high level of accuracy required to match the simulated extinction spectra with the experimental data demanded a more systematic approach. This is where the parametric sweep functionality in COMSOL became indispensable.

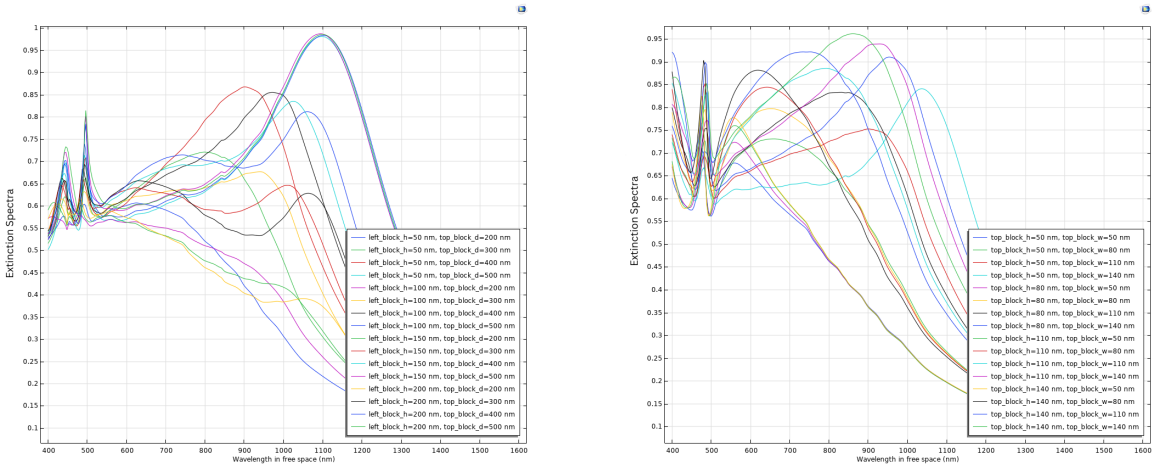
To bridge this gap, the initial dimensions of the SRR were estimated based on the SEM image. While this provided a starting point, obtaining the high level of accuracy required to match the simulated extinction spectra with the experimental data demanded a more systematic approach. This is where the parametric sweep functionality in COMSOL became indispensable.

Parametric sweeps allow the user to iteratively perform FEM-based electrodynamic simulations over a range of predefined parameter values, either in discrete steps or continuously. In the case of the SRR, extensive parametric sweeps were conducted to fine-tune critical dimensions such as arm width, gap size, and overall depth. By iteratively adjusting these parameters and analyzing the resulting extinction spectra, a set of dimensions that produced the closest match to the experimental data was identified.

In addition to standard parametric sweeps, nested sweeps were employed for greater efficiency and precision. Nested sweeps allow multiple parameters to be varied simultaneously within a hierarchical structure, significantly reducing computational time. For example, two nested sweeps were performed to determine the top block height, width, and depth of the SRR structure. By linking these parameters together in a nested sweep configuration, the software could explore a multi-dimensional parameter space more systematically, ensuring interdependencies between parameters were accounted for. Figure 5.3 demonstrates the results of these nested sweeps, showcasing how simultaneous variations in the block dimensions converged on values that minimized the difference between simulated and experimental extinction spectra.

This nested approach proved especially useful in cases where parameters were strongly correlated. For instance, changes in the block height directly influenced the required block width to maintain a consistent optical response. By exploring these interdependencies, nested sweeps provided a faster and more accurate method for deducing optimal dimensions compared to independent sweeps performed for each parameter.

In summary, parametric and nested sweeps proved to be invaluable tools for deducing

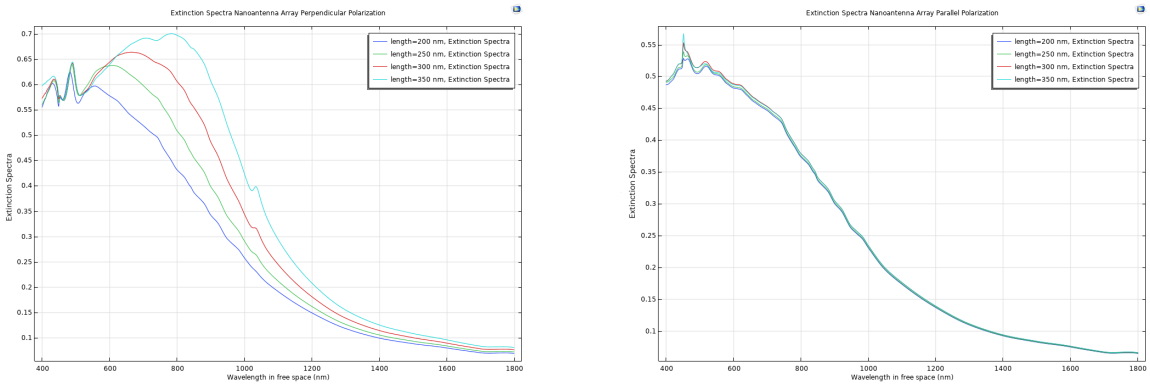


(a) Nested sweep for left block height and depth (b) Nested sweep for top block height and width

Figure 5.3: Parametric sweeps to deduce parameters

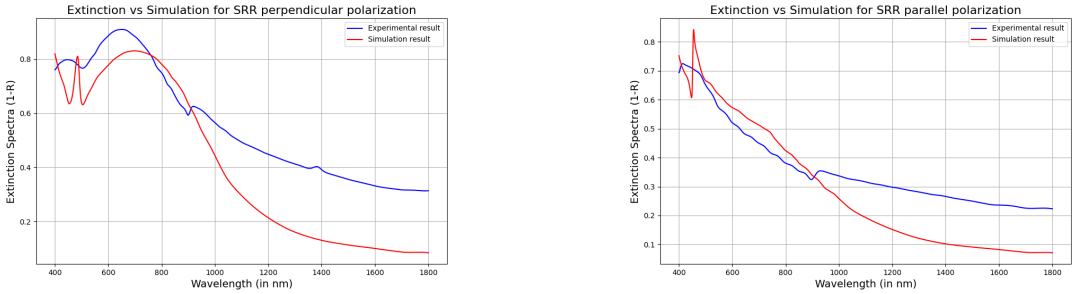
unknown parameters in this study. By leveraging the computational power and automation capabilities of COMSOL, accurate representations of complex nanostructures like the SRR could be achieved, even when direct measurements were unavailable. This method not only underscored the adaptability of COMSOL as a simulation platform but also highlighted the importance of iterative and hierarchical approaches when dealing with incomplete or ambiguous data.

From Figure 5.3a, a general conclusion can be drawn that the combination of values producing the most accurate extinction spectra (in reference to the know SRR extinction spectra given previously) possibly lies within the ranges of [50,100] nm for the left block height and [400,500] nm for the top block depth. Similarly, Figure 5.3b indicates that the optimal range for the top block height and width is likely between [80,110] nm and [50,110] nm, respectively. To refine these results, additional nested sweeps were configured within these specific parameter ranges. By simultaneously varying the left block height and top block depth (Figure 5.3a) or the top block height and width (Figure 5.3b) in nested sweeps, the parameter space was explored more efficiently, accounting for interdependencies between the variables. This approach allowed for a further narrowing of these ranges to a single discrete value for each parameter, ensuring the simulated extinction spectra closely matched the experimental data.



(a) Perpendicular Polarization spectra (b) Parallel Polarization spectra

Figure 5.4: Simulated extinction spectra for nanoantenna nanodevice



(a) Perpendicular Polarization spectra (b) Parallel Polarization spectra

Figure 5.5: Simulated extinction spectra for SRR nanodevice

5.3.2 Statistical Comparison

The parameters for the nanodevices were accurately determined and the extinction spectra were replicated as accurately as possible with quantitative and qualitative comparison metrics as given below.

To ensure the reliability of the simulation environment, the results from COMSOL simulations were validated against recorded experimental data using statistical metrics. The comparison covered both the nanoantennas and Split Ring Resonators (SRR) under perpendicular and parallel polarizations. The results are summarized below:

Nanoantennas:

1. Perpendicular Polarization:

- **200nm Slit Length**: MSE = 0.0132, Kendall Tau Correlation Coefficient = 0.969
- **350nm Slit Length**: MSE = 0.0339, Kendall Tau Correlation Coefficient = 0.936

2. Parallel Polarization:
    - **200nm Slit Length**: MSE = 0.00218, Kendall Tau Correlation Coefficient = 0.971
    - **350nm Slit Length**: MSE = 0.00324, Kendall Tau Correlation Coefficient = 0.966

Split Ring Resonator (SRR):

1. **Perpendicular Polarization**: MSE = 0.033, Kendall Tau Correlation Coefficient = 0.921
2. **Parallel Polarization**: MSE = 0.013, Kendall Tau Correlation Coefficient = 0.981

The Kendall Tau correlation coefficients were chosen over Pearson coefficients for their robustness with smaller datasets and their ability to capture non-linear relationships. Across all configurations, the high correlation coefficients (ranging from 0.921 to 0.981) and low MSE values indicate strong agreement between the simulated and experimental results. These metrics confirm the reliability of the COMSOL simulation framework in generating high-fidelity datasets for neural network training.

5.3.3 Discrepancies in Accuracy

While the validation results are highly encouraging, some discrepancies between simulated and experimental spectra persist, which can be attributed to the following factors:

- **Simulation Environment Limitations**: Simulated environments, while highly accurate, inherently simplify real-world experimental conditions. This introduces small deviations due to computational approximations in solving Maxwell's equations and setting boundary conditions. For instance, the periodicity boundary conditions imposed on either sides of the model imply that the model is being replicated infinitely in either

direction which is obviously not the case in the real-world scenario but is the closest approximation to it.

- **Nanodevice Fabrication Errors**: The Focused Ion Beam (FIB) process used for fabricating the nanodevices is prone to inaccuracies. For instance, during the cutting of slits on the niobium surface, the angles created were not perfectly 90°, leading to deviations in the geometry and affecting the resulting optical response.

- **Parameter Estimation Challenges**: For the SRR device, the exact dimensions were unavailable due to confidentiality restrictions, and only an SEM image was provided. Approximate values were deduced from the image and refined using nested parametric sweeps. While these sweeps narrowed the parameter ranges, the final values remained approximations and may not agree with the actual device dimensions.

In conclusion, the COMSOL simulation framework has demonstrated exceptional reliability and precision, making it a robust tool for generating training datasets. However, the minor discrepancies observed highlight the inherent challenges in reconciling theoretical simulations with real-world experimental data, especially when fabrication inaccuracies and parameter estimation approximations are involved.

Now that the COMSOL environment has been validated it was used to generate the training dataset for the different Neural Networks involved and their predictions are compared in the following chapters.

Chapter 6

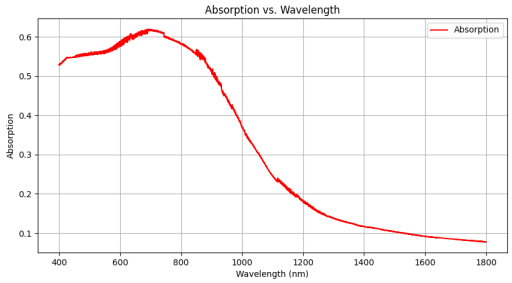
Neural Networks and Results

This chapter discusses all the main results obtained in this study. The four different Neural Networks were trained on two training datasets based on the nanoantennas and SRR nanodevices. Optuna was run on each of the network types for several hundred iterations to obtain the best possible hyperparameters for each Neural Network, as discussed in Chapter 3. Their prediction capabilities are explored in each section, along with possible limiting factors. A final table summarizes the accuracy of each Neural Network in comparison to their simulation counterparts.

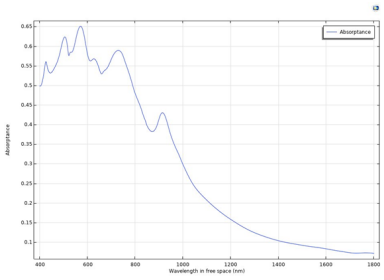
The extinction spectra was simulated for 2 randomly chosen Split Ring Resonator designs with feature values of width = 145nm, period = 860nm, height = 325nm, depth = 175nm; and width = 84nm, period = 938nm, height = 325nm, depth = 108nm under parallel polarization incident light. Additionally, the transmittance spectra for all these simulated nanodevices was in the range of 10^{-9} hence for calculation purposes, will be ignored and the extinction spectra is taken to be equivalent to the absorption for the remainder of this thesis

6.1 Artificial Neural Network Ensemble Model

The Artificial Neural Network (ANN) used in this study was designed as a fully connected feedforward network, capable of predicting the absorptance and reflectance of 3D nanograting-based SRR structures. Additionally, a "Wisdom of Many" protocol was employed. This



(a) ANN Prediction



(b) Simulation Result

Figure 6.1: Comparison between ANN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1

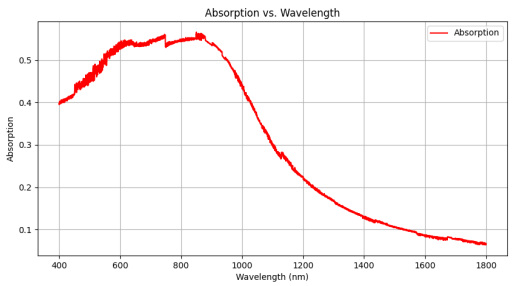
approach involved training multiple neural networks initialized with different random seeds. Predictions from individual networks were averaged to produce more robust and accurate outputs.

Key benefits observed with the ensemble approach included reduced variance and improved generalization. However, computational constraints limited extensive comparisons with high-resolution simulations. The input data consisted of nine features, including the x, y, z spatial coordinates of the point cloud, physical parameters such as the width and depth of the metasurface deformations, and electrodynamic parameters like the dielectric function of superconducting Niobium. The outputs comprised two target variables: absorptance and reflectance.

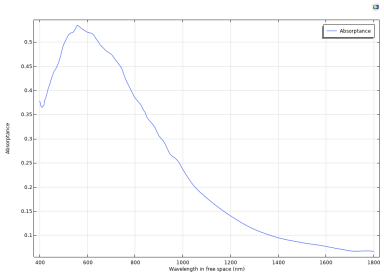
The Optuna framework optimized the hyperparameters, such as the learning rate, the number of neurons in each hidden layer, and the dropout rate for regularization. The best-performing configuration for a two-hidden-layer ANN included:

- Learning rate: 0.00034
- Number of neurons in the first hidden layer: 256
- Dropout rate: 0.2
- Number of neurons in the second hidden layer: 96

Figures 6.1 and 6.2 illustrate the results and demonstrate the accuracy of the ANN ensemble model.



(a) ANN Prediction



(b) Simulation Result

Figure 6.2: Comparison between ANN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2

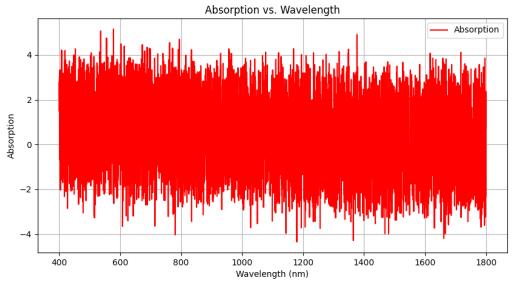
6.2 Transformer Model

Transformers, originally introduced for natural language processing, are celebrated for their ability to capture long-range dependencies and contextual information through their self-attention mechanism. In this study, the Transformer architecture was adapted to predict absorptance and reflectance spectra. However, the results achieved with the Transformer model were underwhelming compared to other architectures, raising questions about its suitability for this specific task.

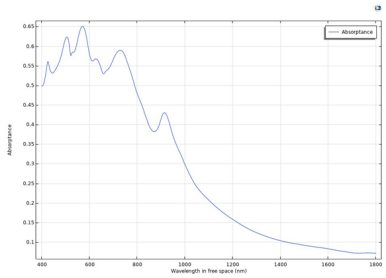
One notable issue was that the extinction spectra predictions occasionally fell outside the physically meaningful range of [0,1], extending to approximately [-4.5, 4.5]. Several factors may have contributed to this outcome. First, Transformers are primarily designed for sequence modeling and may not fully exploit the spatial relationships inherent in point cloud datasets. Second, the relatively modest dataset size limited the model's ability to generalize effectively, as Transformers typically excel with extremely large datasets. Third, the quadratic scaling of self-attention with input size imposed significant computational burdens, necessitating compromises in model complexity during optimization. Finally, the lack of explicit constraints on the output layer exacerbated the issue of physically unrealistic predictions.

The best Transformer model configuration, as determined by Optuna, included:

- Number of attention heads: 4
- Key dimension: 64

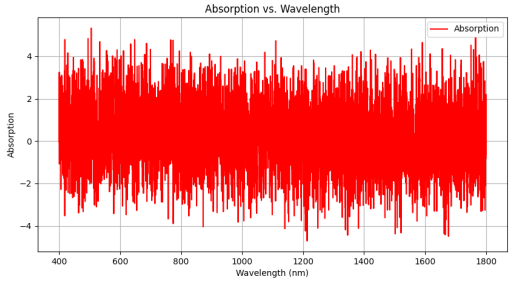


(a) Transformer Prediction

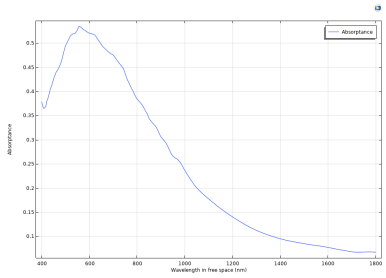


(b) Simulation Result

Figure 6.3: Comparison between Transformer predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1



(a) Transformer Prediction



(b) Simulation Result

Figure 6.4: Comparison between Transformer predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2

- Feedforward layer dimension: 192
- Dropout rate: 0.1
- Learning rate: 0.00019

Figures 6.3 and 6.4 present the performance of the Transformer model.

6.3 Graphical Neural Network (GNN)

Graph Neural Networks (GNNs) are highly effective for learning from graph-like structures, such as point clouds. In this study, point clouds were transformed into graph representations, with each point treated as a node and edges established using the k-nearest neighbors

algorithm (k=5). This approach captured the local spatial relationships among points, which are critical for predicting extinction spectra.

The architecture of the GNN comprised two layers of graph convolution, each followed by batch normalization and a ReLU activation function. These layers incrementally extracted hierarchical features from the graph's topology. A concluding fully connected layer converted the output to the desired absorptance values. To mitigate overfitting, a flexible dropout layer was integrated before the fully connected layer. Hyperparameters, including the hidden dimensions of the graph convolution layers, the dropout rate, and the learning rate, were fine-tuned using Optuna.

Despite its theoretical advantages, the GNN's performance was mixed. The predictions showed moderate agreement with the ground truth, potentially due to the complex interactions between physical features and spectra, which may not be fully captured by local graph structures alone. Figure 6.5 provides a visualization of the graph constructed from a point cloud (K=5 KNN representation used), highlighting how the model integrates spatial and feature-based information.

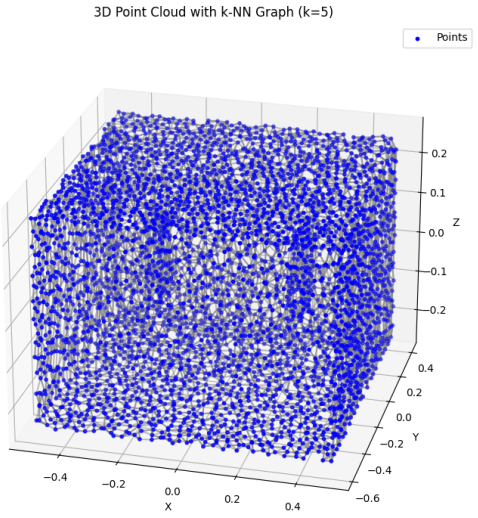
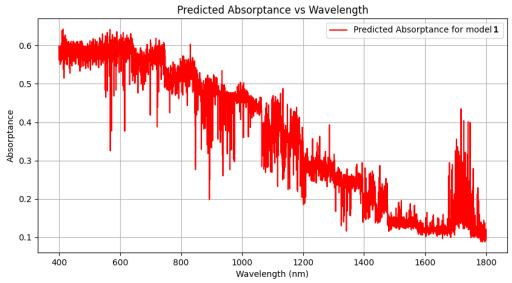
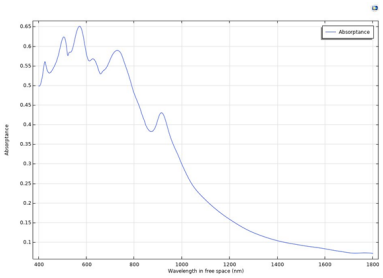


Figure 6.5: Graphical representation of a pointcloud using KNN where K=5

The best-performing configuration for a two-hidden-layer GNN included:

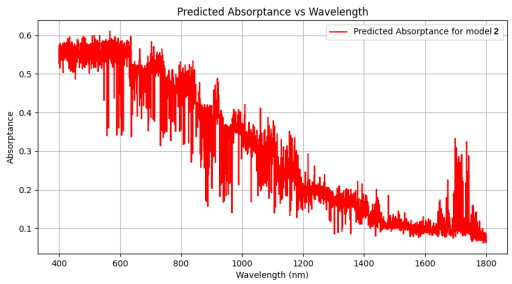


(a) GNN Prediction

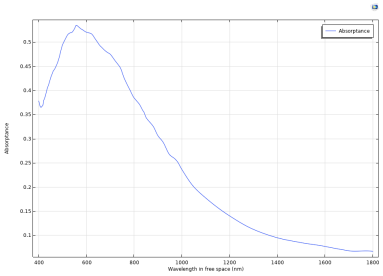


(b) Simulation Result

Figure 6.6: Comparison between GNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1



(a) GNN Prediction

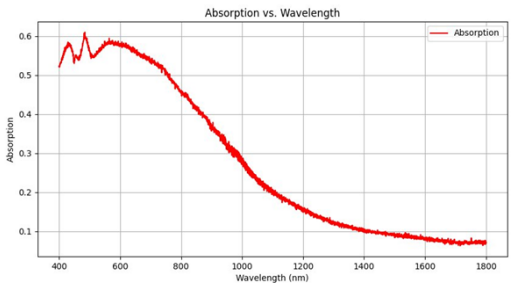


(b) Simulation Result

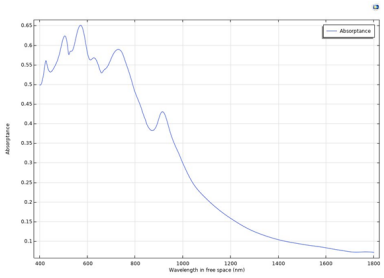
Figure 6.7: Comparison between GNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2

- Learning rate: 0.00259
- Number of neurons in the first hidden layer: 96
- Dropout rate: 0.2949
- Number of neurons in the second hidden layer: 32
- Weight decay: 0.0006

Figures 6.6 and 6.7 illustrate the GNN's results and predictions.



(a) CNN Prediction



(b) Simulation Result

Figure 6.8: Comparison between CNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 1

6.4 Convolutional Neural Network (CNN)

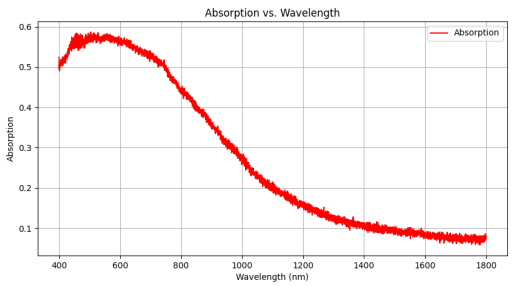
Convolutional Neural Networks (CNNs) are widely recognized for their effectiveness in extracting spatial and feature hierarchies from structured data. In this study, the CNN model was applied to tabular data representing point clouds, treating each point as an input with nine features, including spatial coordinates and physical parameters.

The CNN architecture consisted of multiple 1D convolutional layers that progressively captured local feature patterns along the feature dimension. These were followed by fully connected layers that mapped the extracted patterns to the target absorptance values. Regularization techniques, such as dropout and batch normalization, were incorporated to enhance generalization and stabilize training.

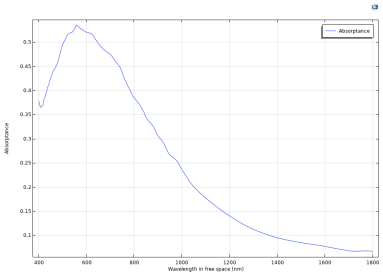
The CNN exhibited strong performance, leveraging its ability to extract localized feature interactions efficiently. However, its reliance on sequential patterns meant it could not fully utilize the spatial relationships encoded in the point clouds. This limitation was partially mitigated by including spatial coordinates as features. Nevertheless, the CNN's relative simplicity and computational efficiency make it a viable option for tasks requiring quick predictions.

The best-performing configuration for a two-hidden-layer CNN included:

- Number of filters: 64
- Kernel size: 7



(a) CNN Prediction



(b) Simulation Result

Figure 6.9: Comparison between CNN predicted extinction spectra and simulation spectra for the Split Ring Resonator design 2

- Number of neurons in densely connected layers: 128
- Dropout rate: 0.1
- Learning rate: 0.00047

Figures 6.8 and 6.9 provide a qualitative comparison of the CNN's performance.

6.5 Model Comparison and Quantitative Evaluation

To assess the effectiveness of the implemented models—ANN, Transformer, GNN, and CNN—a quantitative comparison was performed based on three metrics:

- **Validation Loss**: The best validation loss observed during hyperparameter optimization for each model.
- **Mean Squared Error**: The error between the predicted and actual absorptance values across the entire spectrum.
- **Pearson's Correlation Coefficient**: A statistical measure of the linear association between predicted and actual absorptance values, evaluating the models' predictive accuracy.

Table 6.1: Model Comparison

Model Comparison					
Model	Validation Loss during Optimization	MSE for Design 1	Pearson's Correlation Coefficient for Design 1	MSE for Design 2	Pearson's Correlation Coefficient for Design 2
ANN	0.00018	0.01101	0.911000	0.00288	0.97781
Transformer	1.04053	2.94519	0.12720	3.94772	0.19448
GNN	0.00460	0.01964	0.92676	0.00331	0.96668
CNN	0.00498	0.00069	0.99641	0.00168	0.99371

Table 6.1 summarizes the performance of each model based on these metrics, providing insights into their relative strengths and weaknesses.

In conclusion, the chapter illustrates the main results and comparisons between the predictions of all Neural Networks and the inferences that can be drawn as well as future work in this area are discussed in the following chapters.

Chapter 7

Discussion

This chapter pivots on the analysis of the findings of Chapter 6 and presents conclusions that can be drawn from them. By examining the trends observed in model performance, this discussion aims to highlight the strengths and limitations of each approach, draw conclusions about their suitability for different applications, and suggest potential improvements for future work. Additionally, it also suggests readers potential Neural Network types for certain purposes as well as combinations wherever required.

7.1 Analysis of Model Performance

The ANN ensemble model exhibited strong generalization performance, leveraging the "Wisdom of Many" protocol to reduce variance and improve stability. The averaging of multiple neural networks mitigated the impact of random initialization effects, leading to robust predictions. However, despite these advantages, the computational cost of training multiple instances of the ANN made it less efficient compared to single-network architectures. Additionally, while the model achieved relatively high accuracy, its reliance on fully connected layers meant it lacked specialized mechanisms for capturing spatial relationships within the point cloud data.

Despite the transformative impact of Transformer models in various domains, their application to point cloud-based spectral prediction yielded mixed results. The self-attention

mechanism, while powerful, did not seem to adequately capture spatial dependencies within the dataset. The observed issue of physically unrealistic predictions, with spectra values occasionally falling outside the expected range, suggests that additional constraints or normalization techniques may be required.

The GNN model offered a structured approach to handling point cloud data, leveraging k-nearest neighbors (k=5) to define meaningful spatial relationships. By treating each point as a node within a graph, the model preserved local connectivity information, leading to improved spatial awareness in predictions. However, the GNN's performance did not significantly surpass that of the ANN or CNN, suggesting that the local graph representation alone may not be sufficient to capture the complex interactions governing absorptance and reflectance.

The CNN model demonstrated strong performance, balancing computational efficiency and accuracy. Its ability to extract localized feature patterns made it particularly well-suited for processing tabular representations of point clouds. However, its primary limitation lay in its sequential approach, which, while effective for structured data, did not fully leverage the spatial properties of the dataset.

7.2 Insights and Key Takeaways

7.2.1 Importance of Feature Engineering

One critical observation from this study is that feature representation plays a fundamental role in model performance. The inclusion of spatial coordinates, physical parameters, and electrodynamic properties significantly influenced predictive accuracy. Future efforts could explore additional derived features, such as higher-order spatial relationships or frequency-based transformations, to enhance model learning.

7.2.2 Computational Trade-offs

Each model presented a trade-off between computational complexity and predictive power. The ANN ensemble, while robust, was computationally expensive. The Transformer model

faced scalability limitations due to self-attention's quadratic complexity. The GNN, despite its ability to model spatial dependencies, required significant preprocessing. The CNN, on the other hand, provided a good balance of accuracy and efficiency, making it a practical choice for real-time or resource-constrained applications.The dataset's relatively modest size likely contributed to some of the generalization challenges observed, particularly in the Transformer model. Neural networks typically benefit from large amounts of training data, and while techniques such as data augmentation or synthetic data generation could improve generalization, this remains an area for further research.

7.2.3 Model Generalization and Data Limitations

The dataset's relatively modest size likely contributed to some of the generalization challenges observed, particularly in the Transformer model. Neural networks typically benefit from large amounts of training data, and while techniques such as data augmentation or synthetic data generation could improve generalization, this remains an area for further research.

Chapter 8

Future Work

The final chapter is based on the future work possible in this study and final conclusions.

8.1 Prospective work

8.1.1 Hybrid Architectures

A promising avenue for future research involves combining the strengths of multiple architectures. For instance, integrating a GNN with a CNN could enable efficient feature extraction while preserving spatial dependencies. Similarly, augmenting Transformers with graph-based positional encoding may improve their ability to process point cloud data more effectively.

8.1.2 Improved Hyperparameter Optimization

While Optuna provided a systematic approach to hyperparameter tuning, exploring more advanced search strategies, such as Bayesian optimization with early stopping, could yield better performance. Additionally, fine-tuning architectures with task-specific loss functions might further enhance predictive accuracy.

8.1.3 Expanded Dataset and Transfer Learning

Acquiring a larger dataset or employing transfer learning techniques could significantly improve model robustness. Pretraining on related physics-based datasets before fine-tuning on the specific problem domain could help mitigate data scarcity issues.

8.2 Conclusion

In summary, this study demonstrated the type of plasmon polariton interactions in the metamaterial system as well as the fact that while all four neural network architectures exhibited potential for predicting absorptance and reflectance spectra, their effectiveness varied based on computational efficiency, the ability to capture spatial relationships, and generalization capabilities. The ANN ensemble model provided robust predictions but was computationally intensive. The Transformer model struggled with dataset constraints and spatial relationships. The GNN leveraged graph-based structures but required further refinement to match fully connected models. The CNN delivered strong performance with a balance between accuracy and efficiency.

Moving forward, hybrid models, enhanced hyperparameter tuning, and larger datasets present promising avenues for further research. Ultimately, selecting the most suitable neural network depends on the specific trade-offs between accuracy, interpretability, and computational feasibility, ensuring that future applications can leverage machine learning to its full potential in nanophotonic device modeling.

Bibliography

[1] Shahin Bagheri, Nikolai Strohfeldt, Monika Ubl, Audrey Berrier, Michael Merker, Gunther Richter, Michael Siegel, and Harald Giessen "Niobium as Alternative Material for Refractory and Active Plasmonics" *ACS Photonics*, 2018, 5 (8), 3298-3304

[2] Anatoly V Zayats and Igor I Smolyaninov, Shahin "Near-field photonics: surface plasmon polaritons and localized surface plasmons" *J. Opt. A: Pure Appl. Opt.* 2003, S16

[3] Wen Xuan Tang, Hao Chi Zhang, Hui Feng Ma, Wei Xiang Jiang, Tie Jun Cui "Concept, Theory, Design, and Applications of Spoof Surface Plasmon Polaritons at Microwave Frequencies" *Advanced Optical Materials*, 2019, 7, 1800421

[4] Peter R. Wiecha and Otto L. Muskensi "Deep Learning Meets Nanophotonics: A Generalized Accurate Predictor for Near Fields and Far Fields of Arbitrary 3D Nanostructures" *Nano Letters*, 2020, 20 (1), 329-338

[5] Blanchard-Dionne, Andre-Pierre and Martin, Olivier, "Teaching optics to a machine learning network" *Optics Letters*, 2020, 45

[6] Delfanazari, Kaveh and Muskens, Otto. "On-chip visible light communication-band metasurface modulators with niobium plasmonic nano-antenna arrays" *IEEE Explore*, 2021

[7] K. Delfanazari and O. L. Muskens, "Visible to Near-infrared Chip-integrated Tunable Optical Modulators Based on Niobium Plasmonic Nano-antenna and Nano-circuit Metasurface Arrays" *PIERS*, Hangzhou, China, 2022, pp. 924-929

[8] Delfanazari, Kaveh and Muskens, Otto. "Light-matter interactions in chip-integrated niobium nano-circuit arrays at optical fibre communication frequencies" *IEEE Explore*, 2021

[9] Junxi Zhang et al., "Surface plasmon polaritons: physics and applications" *J. Phys. D: Appl. Phys.* 2012, 45 113001

[10] Stephen A Maier, "Plasmonics: Fundamentals and Applications" 2007

[11] J. B. Pendry et al., "Mimicking Surface Plasmons with Structured Surfaces" *Science* , 305,847-848(2004)

[12] Liu H, Wang B, Ke L, Deng J, Chum CC, Teo SL, Shen L, Maier SA, Teng J, "High aspect subdiffraction-limit photolithography via a silver superlens" *Nano Letters* 2012, 12(3):1549–1554

[13] Wang B, Zhang X, Yuan X, Teng J, "Optical coupling of surface plasmons between graphene sheets" *Applied Physics Letters* 2012, 100(131111)

[14] Atwater H, Polman A, Kosten E, Callahan D, Spinelli P, Eisler C, Escarra M, Warmann E, Flowers C, "Nanophotonic design principles for ultrahigh efficiency photovoltaics" *AIP Conference Proceedings* 2013, 1519:17–21

[15] Schmidt MA,LeiDY,Wondraczek L, Nazabal V, Maier SA, "Hybrid nanoparticle-microcavity-based plasmonic nanosensors with improved detection resolution and extended remote-sensing ability" *Nature Communications* 2013

[16] Jaiswal, R.K., Pandit, N. and Pathak, N.P., "Design, Analysis, and Characterization of Designer Surface Plasmon Polariton-Based Dual-Band Antenna" *Plasmonics* 2018, 13, 1209–1218

[17] Hibbins AP, Evans BR, Sambles JR, "Experimental verifica tion of designer surface Plasmons" *Sci Rep* 2005, 308(5722):670–672

[18] Rusina A, Durach M, Stockman MI, "Theory of spoof plas mons in real metals" *Appl Phys A Mater Sci Proces* 2010, 100(2):375 378

[19] GaoX, HuiShiJ, ShenX, FengMaH, XiAngJiangW, LiL, CuiTJ. " Ultrathin dual-band surface plasmonic polariton waveguide and frequency splitter in microwave frequencies" *Appl Phys Lett* 2013, 102(15):9–13

[20] Ourir, Abdelwaheb and Maurel, Agnès and Félix, Simon and Mercier, Jean-François and Fink, Mathias, "Manipulating light at subwavelength scale by exploiting defect-guided spoof plasmon modes" *Physical Review B* 2017, 96

[21] Schnitzer, Ory, "Spoof surface plasmons guided by narrow grooves" *Physical Review B* 2017

[22] Wood, John and Tomlinson, L. and Hess, Ortwin and Maier, S. and Fernández-Domínguez, Antonio. "Spoof plasmon polaritons in slanted geometries", *Physical Review B.* 2012, 85.

[23] Xiong, Yuzan and Christy, Andrew and Yan, Zixin and Pishehvar, Amin and Mahdi, Muntasir and Wu, Junming and Cahoon, James and Yang, Binbin and Hamilton, Michael and Zhang, Xufeng and Zhang, Wei., "Hybrid Magnonics with Localized Spoof Surface Plasmon Polaritons" *Phys Rev Applied.* 2024, 22(3)

[24] Christopher M. Bishop, "Pattern Recognition and Machine Learning", 2009

[25] Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", 2015

[26] William L. Hamilton, "Graph Representation Learning", 2020

[27] S. Schmidgall, J. Achterberg, T. Miconi, L. Kirsch, R. Ziaei, S. P. Ha jiseyedrazi, and J. Eshraghian, "Brain-inspired learning in artificial neu ral networks: a review," *APL Machine Learning* 2023

[28] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E., "Neural Message Passing for Quantum Chemistry" *Proceedings of the 34th International Conference on Machine Learning* , 2017

[29] Fout, A., Byrd, J., Shariat, B., and Ben-Hur, A., "Protein Interface Prediction using Graph Convolutional Networks" *Advances in Neural Information Processing Systems* , 2017

[30] Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D., "Graph R-CNN for Scene Graph Generation" *European Conference on Computer Vision* , 2018

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., "Attention Is All You Need" *Advances in Neural Information Processing Systems* , 2017

[32] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* , 2019

[33] Jumper, J., Evans, R., Pritzel, A., Green, T., et al., "Highly accurate protein structure prediction with AlphaFold" *Nature* , 2021, 596(7873), 583-589.

[34] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition" *Proceedings of the IEEE* , 1998, 86(11), 2278-2324

[35] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks" *Advances in Neural Information Processing Systems* , 2012

[36] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" *Advances in Neural Information Processing Systems* , 2015

[37] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., et al., "A Survey on Deep Learning in Medical Image Analysis" *Medical Image Analysis* , 2017, 42, 60-88

[38] Akiba, Takuya and Sano, Shotaro and Yanase, Toshihiko and Ohta, Takeru and Koyama, Masanori., "Optuna: A Next-generation Hyperparameter Optimization Framework" *25th ACM SIGKDD International Conference* , 2019, 2623-2631

[39] Martin, Olivier J. F., and Christian Girard, "Using a Green's tensor approach to calculate the optical properties of heterogeneous nanostructures" *Physical Review B* ,1997, 1382–1392.

[40] Yee, Kane S, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media." *IEEE Transactions on Antennas and Propagation* , 1966, 302–307.

[41] Jin, Jian-Ming, "The Finite Element Method in Electromagnetics", John Wiley and Sons, 2014.

[42] Cheng Wang, Ming Cheng, Ferdous Sohel, Mohammed Bennamoun, Jonathan Li, "NormalNet: A voxel-based CNN for 3D object classification and retrieval", *Neurocomputing* , Volume 323, 2019, Pages 139-147

[43] Molesky, S., Lin, Z., Piggott, A.Y. et al., "Inverse design in nanophotonics", *Nature Photon* , 2018, 12, 659–670

[44] Sambit Ghadai, Aditya Balu, Soumik Sarkar, Adarsh Krishnamurthy, "Learning localized features in 3D CAD models for manufacturability analysis of drilled holes", *Computed Aided Geometric Design* , Volume 62, 2018, Pages 263-275

[45] Malkiel, Itzik and Mrejen, Michael and Nagler, Achiya and Arieli, Uri and Wolf, Lior and Suchowski, Haim., "Plasmonic nanostructure design and characterization via Deep Learning", *Light: Science and Applications* , 2018, 7

[46] Wiecha, Peter and Arbouet, Arnaud and Christian, Girard and Muskens, Otto., "Deep learning in nano-photonics: inverse design and beyond", *Photonics Research* , 2021, 9, B182-B200