National College of Ireland

Predicting Optical Property in One-Dimensional Axionic Photonic Crystals by Deep Learning Approach

MSc Research Project
MSc in Data Analytics

Anny Faria
Student ID: x24124770

School of Computing
National College of Ireland

Supervisor: Dr. David Hamill

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Anny Faria
Student ID:	x24124770
Programme:	MSc in Data Analytics
Year:	2026
Module:	MSc Research Project
Supervisor:	Dr. David Hamill
Submission Due Date:	28/01/2026
Project Title:	Predicting Optical Property in One-Dimensional Axionic Photonic Crystals by Deep Learning Approach
Word Count:	8436
Page Count:	36

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anny Caroline de Araújo Faria
Date:	28th January 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐✓

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting Optical Property in One-Dimensional Axionic Photonic Crystals by Deep Learning Approach

Anny Faria
x24124770

Abstract

Axionic photonic crystals (APCs) are a generalized class of topological photonic materials characterized by physical parameters ϵ, μ, and θ. Controlling their transmission spectra is crucial for developing future axionic photonic devices. The central problem is that traditional computational methods (like TMM) are slow and require substantial memory. Furthermore, the literature lacked a deep learning prediction focused specifically on the axionic crystal domain, and it was unproven if MLPs could accurately predict phenomena characterized by abrupt oscillations, such as their transmission spectra. This study employs a Multi-Layer Perceptron (MLP) to predict the transmission spectra of 1D APCs, solving the traditional method's cost limitations. The study was conducted in three experiments with increasing input complexity (X, R, δ). Datasets were generated via TMM and an MLP was trained to map inputs to the output spectrum $T(\overline{\omega})$, with performance evaluated primarily by Euclidean Distance (ED). The results show the MLP is highly efficient on calculation time. The model achieved its best accuracy in the one-parameter experiment, with a mean ED = 0.0365. However, in the three experiments the results indicate the MLP works well in low-complexity regimes but struggles to model abrupt oscillations that occur in high-non-linearity parameter regimes near $\delta = 1.0$, where performance was poor and unstable. This study validates the MLP as an effective method to accelerate 1DAPC analysis and provides the foundation for all future research into the inverse design of these complex axionic devices.

1 Introduction

1.1 Background of the Study

It is well established in the literature that solid-state physical phenomena can occur not only with massive particles but also with photons. Photonic crystals (PCs) have become a major focus of study in recent years (Dhanabalan et al. 2023). According to their geometrical symmetry, PCs can be categorized into one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) structures. They are systems whose refractive index changes periodically in space, enabling control of the propagation of electromagnetic waves.

A special type of photonic crystal is called a one-dimensional axionic photonic crystal (1DAPCs), which are periodic structures formed by alternating layers A and B that control the propagation of light in a single direction (angle of light incidence = 0). Each

layer has intrinsic physical and geometric properties, known as structural parameters, which influence how light propagates through the material. The physical properties are electrical permittivity $\epsilon_{B,A}$, magnetic permeability, $\mu_{B,A}$ and axionic term $\theta_{B,A}$. The geometric property is the size of the layer d_A and d_B. When light propagates in this crystal, at the interface between layers A and B, a portion of the light is reflected and another portion is transmitted. The transmitted portion continues to propagate through the layer, while the reflected portion is redirected backward. This process repeats at each interface between layers A and B until the light finishes propagating through the structure. This light behaviour produces an optical spectrum that is relevant for photonic technological applications, such as optical filters and photonic devices based on axions. For such a system, the transfer matrix method (TMM) is a traditional method used to investigate the propagation of light within one-dimensional photonic materials. This method is composed of the transmission (when the waves cross any interface) and propagation (when the waves propagate inside a given layer) matrices, as well as the deterministic sequences investigated. However, despite their potential applications, the computational analysis of these systems is challenging.

1.2 Research Problem

A major issue in the photonics field is that, although many studies have contributed significantly to the development of photonic technologies, most investigations involving these crystals have been conducted theoretically using traditional computational methods such as the Transfer Matrix Method (TMM), Plane Wave Expansion (PWE), and Finite-Difference Time-Domain (FDTD). However, these approaches are often time-consuming and require substantial computational memory.

1.3 Research Questions and Objectives

In recent decades, with advances in artificial intelligence (AI), particularly in the development of neural network models, many studies in the field of physics have emerged using neural networks (NNs) like Multilayer Perceptron (MLP), Deep Neural Network (DNN) and Convolutional Neural Networks (CNNs) as an alternative of low cost and huge potential, significantly overcoming limitations found in traditional methods. In this context, very recent works involving the application of neural networks in PCs have emerged in many areas of the photonics research field, like inverse design of photonic crystals (Long et al., 2019, 2024; Li et al., 2025; Liu & Yu, 2019) especially deep neural networks (DNNs) combined with optimization methods (Li et al., 2025; Zhan et al., 2022), aiming to solve the limitation of designing optical systems and materials based on traditional methods and human intuition.

This study focuses on three central research questions:

Research question 1: How accurately can a neural network predict the transmission spectra of circularly polarized light in one-dimensional axionic photonic crystals?

Research question 2: How does the neural networks performance compare with the Transfer Matrix Method in terms of accuracy and computational cost?

Research question 3: How sensitive is the models generalization to variations in physical and geometrical parameters (X, R, δ)?

To address this question, we propose an extension of the method proposed by Liu & Yu (2019), adapting it to the investigation of the highly oscillatory spectrum of transmission of circularly polarized wave propagation in a one-dimensional axionic photonic crystal where only one neural network model called Multi-Layer Perceptron (MLP) is trained, and their performances are tested and compared with the Transfer Matrix Method (TMM). The investigation has been carried out considering three cases, which are, respectively, one-, two-, and three-parameter predictions similarly to Liu & Yu (2019). The present work is hopeful to provide an effective method for the spectrum analysis in 1DAPCs and to lay a foundation for inverse design of 1DAPCs.

1.4 Significance of the Research

This research addresses the limitation of current computational methods TMM used in the analysis of one-dimensional axionic photonic crystals (1DAPCs), which are time consuming and computationally costly.

By employing a Multi-Layer Perceptron (MLP), this study aims to validate a low-cost and highly efficient alternative for accurately predicting the transmission spectrum of circularly polarized light. The methodology applied in this research will not only accelerate the simulation and optimization processes but, also will lay the groundwork for the inverse design of 1DAPCs, thereby accelerating the development of novel optical filters and axion based photonic devices.

1.5 Thesis Organisation

The structure of this thesis is organized as follows: In Sec. 2 we present a critical analysis of the literature review. Its divided into four parts that cover knowledge of the classical photonic crystal theory, time photonic crystals, and the recent advances in methodologies using deep learning in the photonics field. It concludes by identifying the research niche and outlining the expected contributions of this study. In Sec. 3 we present the research methods and specifications.

In Sec. 4 we present in detail the implementation of the proposed solution of the research question and research resources. In Sec. 5 we present a comprehensive analysis of the results, and the main findings of the study are presented. In Sec. 6 we summarize the main results obtained in this work.

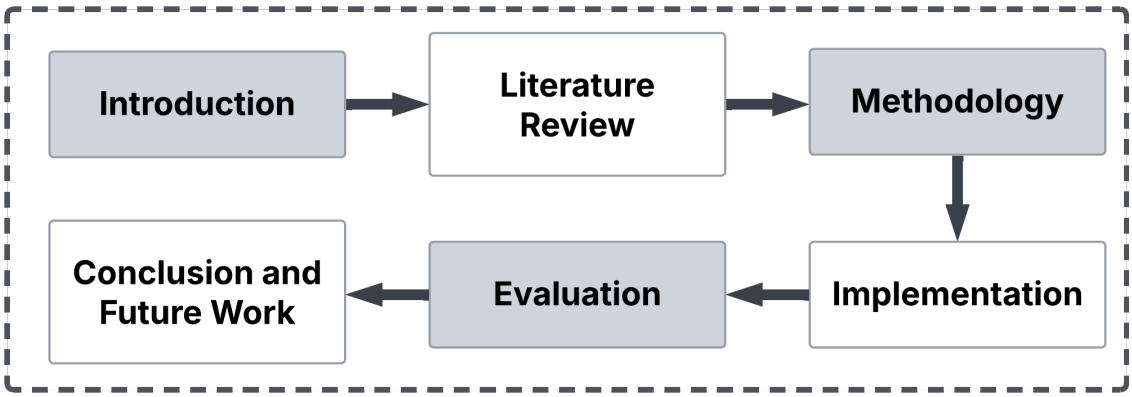

Figure 1: Proposed roadmap for the thesis

2 Literature Review

2.1 Theoretical Framework

The use of deep learning to predict optical spectrum properties like transmission is a new trend and an innovative step toward overcoming traditional methods, which usually require extensive numerical simulations that are costly, time-consuming, and computational memory. Recently, machine learning and deep learning (ML/DL) have been applied in many areas of photonics to reduce simulation costs, accelerate design, and predict complex properties. In this section, we present an overview of the fundamental concepts for understanding the use of traditional methods in photonic crystals from different perspectives within this research field, as well as a critical overview of recent research papers in deep learning, which helps us to analyse the strengths, weaknesses, or limitations in this field.

This section has been divided into the following subsections: In Subsec. 2.2.1 and 2.2.2 we address classical photonic crystal methods, outlining basic concepts of photonic crystals (PCs) and classical methods used in the study of the propagation of light in PCs and critically analyzing traditional methods. In Subsec. 2.2.3 and, 2.2.4, and 2.2.5 we presents a very recent contribution in photonics using a deep learning approach in different application domains, showing the strengths, weaknesses, or limitations in this field.

2.2 Review of Related Studies

2.2.1 Conventional Methods in Optical Property Analysis

The fundamental concepts of photonic crystals (PCs) and the classical methods used to study the light propagation in these materials are well documented in two important books: Joannopoulos et al. (2008) and Markoš & Soukoulis (2008). The first describe how periodic dielectric structures manipulate electromagnetic waves, producing photonic bandgaps (PBGs), and dispersion phenomena through solutions of Maxwells equations in periodic media. They employ the plane-wave expansion (PWE) method in an one-dimensional periodic multilayer system, presenting numerical calculations to compute bandgap structures. On the other hand, Markoš & Soukoulis (2008) use the transfer matrix method (TMM) to analyze the propagation of electrons and electromagnetic waves through one-dimensional periodic multilayer systems. They explicitly use the transfer matrix formalism to study problems related to the transmission of electrons and electromagnetic waves, particularly in photonic crystals and left-handed materials, and also dispersion relation and band structure conditions from the trace of the transfer matrix. Their approach demonstrates how the different formalisms can predict allowed and forbidden bands for different wave types, highlighting TMM as a versatile alternative to plane-wave expansion methods. Both of these approaches form a solid foundation for analysing wave phenomena in periodic multilayer systems.

Although the book of Joannopoulos Joannopoulos et al. (2008) emphasizes the precision of plane-wave expansion (PWE) for numerical calculations of photonic optical properties and band structures, the method becomes computationally intensive as the system size or material complexity increases, requiring a specific procedure called truncation of Fourier harmonics. Similarly, Markoš & Soukoulis (2008) show that the transfer matrix method (TMM) is versatile compared to plane-wave expansion, but the approach still

requires a transfer matrix calculation for each frequency and parameter set to compute transmission and bandgaps, requiring computational costs. The next subsection examines how the TMM has been applied in an axionic photonic crystal to compute simulations to analyse optical properties.

2.2.2 Axionic Photonic Crystals

With the advent of topological photonic materials, several studies have utilized the transfer matrix method (TMM) to investigate electromagnetic waves in one-dimensional axionic systems, where the presence of magnetoelectric interactions changes the electromagnetic response. Granada & Rojas (2017) present a 4×4 transfer matrix method (TMM) applied to a one-dimensional topological photonic insulator. They present analytical expressions that describe how the topological parameter axion, such as θ, induces changes in the reflection coefficients for incident TE and TM waves, as well as the polarization rotation of the reflected and transmitted waves. On the other hand, Araújo et al. (2021) apply a conventional 2×2 TMM to one-dimensional axionic photonic crystals (1DAPCs), analyzing the propagation of circularly polarized light within this material. Their analysis reveals that the first photonic band gap (PBG) is governed by the dielectric contrast of $R = \epsilon_B/\epsilon_A$, the thickness ratio $X = d_B/d_A$, and the axionic contribution $\delta = \pi^2(\theta_A - \theta_B)^2/\alpha^2$. They also show that there θ is a mechanism for controlling the frequency interval that propagates in the axionic material.

However, both studies show the precision of the transfer matrix method (TMM) in capturing the influence of axionic parameters θ on the transmittance, reflectance, and band structures. It uses numerical methods to compute simulations, setting varied parameters and presenting curves for each configuration, which become computationally intensive. The next subsection examines how machine learning has been applied to solve the computational cost, forming the basis for an important path to analyse optical phenomena and band structures.

2.2.3 Deep Learning for Optical Property Prediction

Recently, many studies have emerged using deep learning approaches as practical computational tools to exploit dispersion relation, optical spectrums, and the inverse design approach, showing excellent performance in reducing significantly computationally costly, time-consuming and memory-intensive providing an alternate avenue to speed simulations. The following discusses recent papers for many applications of neural networks in photonics.

2.2.4 Neural Network for Prediction

Neural networks can significantly reduce computational cost. Liu & Yu (2019) proposed a study in which they trained deep backpropagation neural networks (DBP-NNs) and radial basis function neural networks (RBF-NNs) to predict dispersion relations in 1D phononic crystals. The training data came from the transfer matrix method (TMM). They show that RBF-NNs achieve faster training and high accuracy for single-structural parameters, while DBP-NNs offer more stability for structural multi-parameter parameters. Both models drastically reduce computational time once trained, providing accurate predictions properties. Also Jabin & Fok (2022) shows feed-forward multilayer perceptron in another system called photonic crystal fibers (PCFs). They address a more complex

forward prediction problem, proposing a feed-forward multilayer perceptron (MLP) to simultaneously predict 12 distinct optical parameters of photonic crystal fibers (PCFs) from 6 input parameters. Their study directly agrees with Liu & Yu on the main benefit, reporting a reduction of over 99.9% in computation time.

The strength of both authors Liu & Yu (2019) and Jabin & Fok (2022) lies in demonstrating that multilayer perceptron (MLP) can accurately predict physical properties from structural parameters and reduce computation time. However, a notable limitation is the restricted scope of the optical phenomena investigated, which does not extend to the analysis of transmission or reflection spectra. More critically, it remains an open question whether the proposed MLP architecture possesses sufficient generalization capacity to accurately predict phenomena characterized by abrupt oscillations. Also, None of these studies addressed the specific physical domain of axionic photonic crystals, which introduce unique physical parameters such as the term θ.

To conclude, their work directly informs our study, providing a precedent for the use of NNs as a fast forward predictor. Our work applies this same philosophy to a different and more complex physical domain, axionic crystals, and focuses on a different but equally important outcome, the transmission spectrum, testing their ability to generalize to complex dynamics that were not the focus of previous studies.

2.2.5 Inverse Design Approaches

Neural networks have been integrated into inverse design frameworks for photonic systems, although axionic effects remain unexplored. Long et al. (2019) propose a study for topological photonic crystals. They develop a tandem neural network architecture combining a pre-trained forward predictor with an inverse generator to design one-dimensional dielectric PCs with specific topological invariants (Zak phases). This approach solves the common mapping challenge in inverse design problems, ensuring that the obtained solutions correspond to physically feasible structures, the validity of which was confirmed through simulations using the transfer matrix method (TMM). Also, Long et al. (2024) extends the inverse design approaches to photonic time crystals (PTCs) through a tandem neural network pipeline (forward and inverse), capable of mapping vectors of topological properties including the relative phases (ϕ) associated with Berry phases in momentum gaps into temporal layer configurations. This strategy mitigates the challenge typical of inverse mappings and enables the generation of physical structures, validated via TMM simulations.

These studies on inverse design are critical, as they reveal the true importance of a robust model. The methodologies of Long et al. (2019) and Long et al. (2024) are not alternatives to forward prediction, they are dependent on it. In addition, these studies collectively demonstrate that deep learning in the inverse design of photonic systems can accelerate and scale the design process. A notable limitation is that this approach primarily focuses on dielectric 1DPCs, and does not address inverse design of a complex physical phenomena such as axionic terms.

To conclude, the present study is focused on creating this foundational building block. By developing an MLP to accurately predict the transmission spectrum in 1DAPCs, our work provides the essential, missing tool required to motivate all future research into the inverse design of these photonic devices based on axion.

3 Methodology

The methodology is an essential part of any research, it ensures the effectiveness of the proposed research and contributes to the successful achievement of the research objectives. In this study, the methodology is described in a series of steps, as illustrated in figure 3. These steps provide a structured process that can be followed to replicate the results produced by our proposed research.

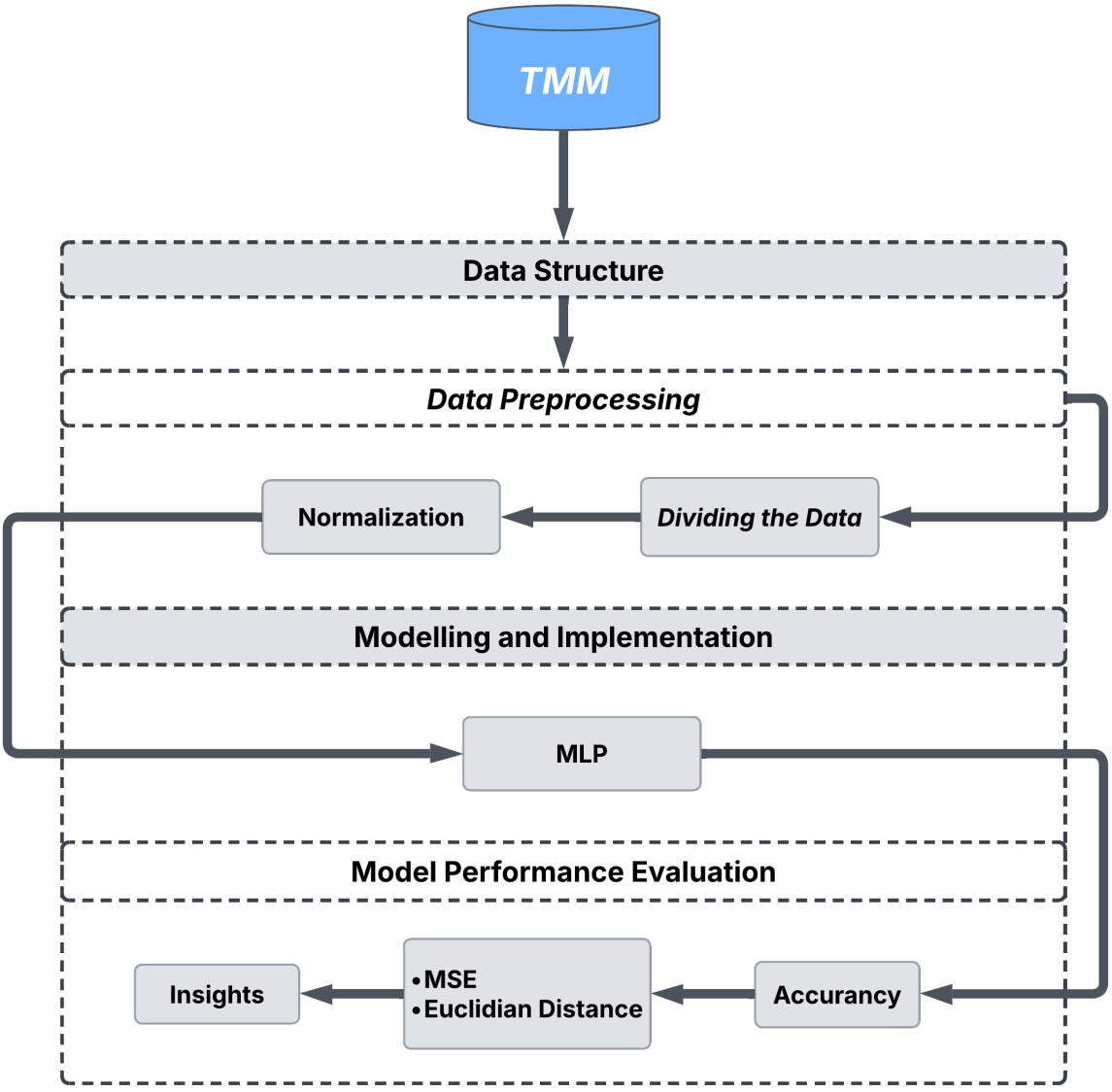

Figure 2: Research Methodology for the Study

3.1 Generating Data with TMM

Data acquisition is the first and most important step before starting an investigation. In this section, we summarize the same mathematical method used in the paper Araújo et al. (2025) regarding the mathematical approach of the transfer matrix method (TMM) to generate numerical transmission spectra of light for 1D axionic photonic crystals (1DAPC) composed of an infinite periodic arrangement of two layers named A and B with dieletric

permittivities ϵ_A and ϵ_B, magnetic permeabilities μ_A and μ_B, topological angles θ_A and θ_B, and thicknesses d_A and d_B, respectively, as described in Fig. 4. We choose a coordinate system such that the topological photonic crystal is perpendicular to the z-axis Araújo et al. (2021).

The transfer-matrix method (TMM) is a powerful analytical technique for the study of any kind of wave propagating through any multilayered media. One of the advantages of applying TMM is to simplify the algebra that can otherwise be quite involved. Within the framework of TMM and considering electromagnetic waves, there are two kinds of matrices: (i) the transmission matrix, which connects the fields across an interface, and (ii) the propagation matrix, which describes the fields propagating inside a layer Markoš & Soukoulis (2008) and Araújo et al. (2025).

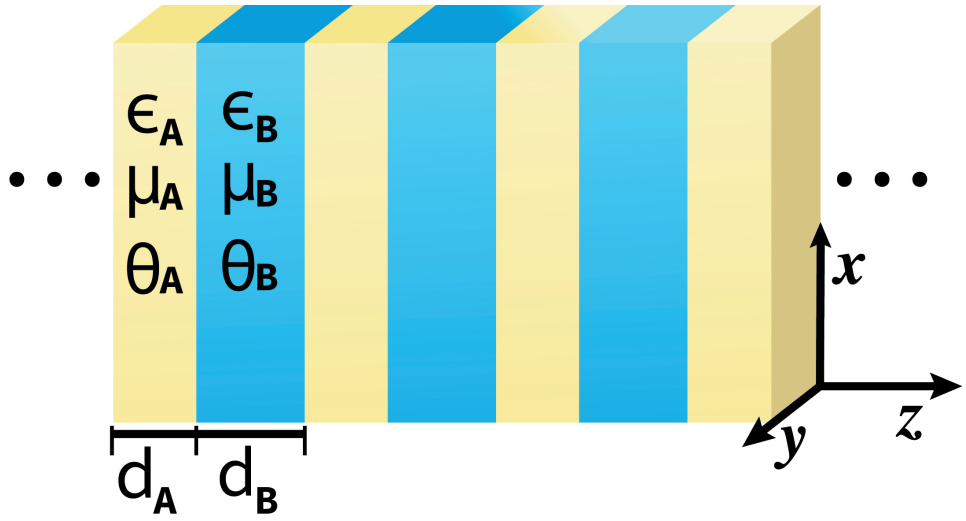

Figure 3: Design of 1D axionic photonic crystal. Source: Araújo et al. (2021)

Here, we assume that the electrical permittivity $\epsilon(\vec{r})$, the magnetic permeability $\mu(\vec{r})$, and the topological angles $\theta(\vec{r})$ are real, isotropic, non-dispersive and periodic, with translational vector $\vec{R} = D\hat{z}$, where $D = d_A + d_B$ is the unit cell size Araújo et al. (2021), that is,

$$\epsilon(\vec{r}) = \epsilon(\vec{r} + \vec{R}) \Rightarrow \epsilon(z) = \epsilon(z + D), \tag{1}$$

$$\mu(\vec{r}) = \mu(\vec{r} + \vec{R}) \Rightarrow \mu(z) = \mu(z + D), \tag{2}$$

$$\theta(\vec{r}) = \theta(\vec{r} + \vec{R}) \Rightarrow \theta(z) = \theta(z + D). \tag{3}$$

For the normal incidence case, i.e., when the light wave incides perpendicularly to the xy-plane, one can propose as a solution for the wave equation the superposition of two propagating waves with opposite directions in z-axis Araújo et al. (2021). After some algebra, we obtain that the total electric and magnetic fields in a medium j $(= A, B)$ can be given as

$$\vec{E}_{j,\lambda} = \left[E_{j,\lambda} e^{ik_j z}\hat{v}_\lambda + \bar{E}_{j,\lambda} e^{-ik_j z}\hat{v}_\lambda\right], \tag{4}$$

$$\vec{H}_{j,\lambda} = \left[g_{j,\lambda} E_{j,\lambda} e^{ik_j z}\hat{v}_\lambda + \bar{g}_{j,\lambda}\bar{E}_{j,\lambda} e^{-ik_j z}\hat{v}_\lambda\right]. \tag{5}$$

where the real wave-vector is $\vec{k_j} = (0, 0, k_j)$, where $k_j = n_j\omega/c$ and $n_j = \sqrt{\epsilon_j \mu_j}$ is the refraction index of the medium. Here, $\lambda = \pm$ and is related to the right $(+)$ and left $(-)$

circularly polarized waves, and the $\hat{v}_\lambda = \hat{v}_\pm$ is given by

$$\hat{v}_\pm = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i \end{pmatrix} \tag{6}$$

In addition, we have defined $g_{j,\lambda}$ ($\bar{g}_{j,\lambda}$ is the conjugate complex)

$$g_{j,\lambda} = \frac{\alpha}{\mu_0 c \pi} \theta_j - \lambda \frac{i}{\omega \mu_0 \mu_j} k_j. \tag{7}$$

We need to know how the electromagnetic waves behave at the boundaries Araújo et al. (2021). By starting from the interface, the boundary conditions are given by

$$\vec{E}_{A,\lambda} = \vec{E}_{B,\lambda}, \tag{8}$$

$$\vec{H}_{A,\lambda} = \vec{H}_{B,\lambda}. \tag{9}$$

Applying the above boundary conditions at $z = 0$, and using Eqs. (4)-(5), we have (one can omit the index λ without loss of generality) Araújo et al. (2021).

$$E_A + \bar{E}_A = E_B + \bar{E}_B, \tag{10}$$

$$g_A E_A + \bar{g}_A \bar{E}_A = g_B E_B + \bar{g}_B \bar{E}_B, \tag{11}$$

where the coefficients E_B and $\bar{E}_B$ can be related to the E_A and $\bar{E}_A$ ones by a 2×2 interface matrix from medium A to B Araújo et al. (2021).

$$\begin{pmatrix} E_B \\ \bar{E}_B \end{pmatrix} = M_{AB} \begin{pmatrix} E_A \\ \bar{E}_A \end{pmatrix}, \tag{12}$$

where

$$M_{AB} = \frac{1}{2i\mathfrak{Im}\left[g_B\right]} \begin{pmatrix} g_A - \bar{g}_B & \bar{g}_A - \bar{g}_B \\ -g_A + g_B & -\bar{g}_A + \bar{g}_B \end{pmatrix} \tag{13}$$

From Eqs. (10)-(11), one can obtain 2×2 interface matrix M_{BA} that relates the coefficients from the medium B to the medium A in the following way Araújo et al. (2021).

$$\begin{pmatrix} E_A \\ \bar{E}_A \end{pmatrix} = M_{BA} \begin{pmatrix} E_B \\ \bar{E}_B \end{pmatrix}, \tag{14}$$

where

$$M_{BA} = \frac{1}{2i\mathfrak{Im}\left[g_A\right]} \begin{pmatrix} g_B - \bar{g}_A & \bar{g}_B - \bar{g}_A \\ -g_B + g_A & -\bar{g}_B + \bar{g}_A \end{pmatrix}. \tag{15}$$

A general expression for the transmission matrix when a electromagnetic wave that crosses an interface from a medium m to another one n can be written as follows Araújo et al. (2021),

$$M_{mn} = \frac{1}{2i\mathfrak{Im}\left[g_n\right]} \begin{pmatrix} g_m - \bar{g}_n & \bar{g}_m - \bar{g}_n \\ -g_m + g_n & -\bar{g}_m + \bar{g}_n \end{pmatrix}, \tag{16}$$

where $\mathfrak{Im}[g_n]$ corresponds to the imaginary part of the g term for the medium n, and that is defined in Eq. (7).

Now, for a wave that propagates inside a layer with thickness d_j and has wave-vector k_j, the 2×2 propagation matrix is given by Araújo et al. (2021),

$$M_j = \begin{pmatrix} e^{ik_j d_j} & 0 \\ 0 & e^{-ik_j d_j} \end{pmatrix}. \tag{17}$$

For more details please see Ref. Markoš & Soukoulis (2008).

When dieletrics are arranged in a periodical way to form a photonic crystals, eletromagnetic waves are modulated by Bragg scatterings, showing the photonic band structure, where the photonic bandgaps emerge Araújo et al. (2021). Consider the l-th unit cell $[A|B]$ of an one-dimensional topological photonic insulator. The electric fields inside A and B slabs are named as $\vec{E}_{A,l} = (E_{A,l}, \bar{E}_{A,l})$ and $\vec{E}_{B,l} = (E_{B,l}, \bar{E}_{B,l})$. In order to obtain the transfer-matrix for this unit cell, one must relate the electric fields in $(l+1)$-th unit cell to the l-th one Araújo et al. (2021).

For an electromagnetic wave that propagates inside a slab A and crosses the interface from A to B, both in the same j-th unit cell, we have

$$\begin{pmatrix} E_{B,l} \\ \bar{E}_{B,l} \end{pmatrix} = M_A M_{AB} \begin{pmatrix} E_{A,l} \\ \bar{E}_{A,l} \end{pmatrix}. \tag{18}$$

In order to the same electromagnetic wave can propagate itself through the layer B and crosses the interface from B, that belongs to the l-th unit cell, to A, which is now in the $(l+1)$-th unit cell, the transfer-matrix is given by Araújo et al. (2021),

$$\begin{pmatrix} E_{A,l+1} \\ \bar{E}_{A,l+1} \end{pmatrix} = M_B M_{BA} \begin{pmatrix} E_{B,l} \\ \bar{E}_{B,l} \end{pmatrix}. \tag{19}$$

By replacing Eq. (18) in Eq. (19), we have

$$\begin{pmatrix} E_{A,l+1} \\ \bar{E}_{A,l+1} \end{pmatrix} = M \begin{pmatrix} E_{A,l} \\ \bar{E}_{A,l} \end{pmatrix} \tag{20}$$

where

$$M = M_B M_{BA} M_A M_{AB} \tag{21}$$

is a 2×2 transfer-matrix for the whole unit cell $[A|B]$ Araújo et al. (2021).

It is known from the literature that the coefficients of transmission $\mathbf{T}$ is obtained from the elements of the transfer-matrix M, which are given by Araújo et al. (2025),

$$\mathbf{T} = \frac{1}{M_{11}} \tag{22}$$

Using the Transfer Matrix Method (TMM), we generated datasets based on three experimental scenarios, each representing a different parameter configuration of the system under study. For the experiment we considered X and R are unchanged and the range of δ is from 0.0 to 1.0, for the experiment, we considered X is unchanged and the ranges of R and δ are respectively from 0.00 to 1.0 and from 0.25 to 3.00, and for the third experiment we considered, where the ranges of X, R and δ are respectively from 0.00 to 3.00, from 0.25 to 3.00 and from 0.00 to 1.00. The number of the total file generated in each experiment is shown in Tab. 3.1:

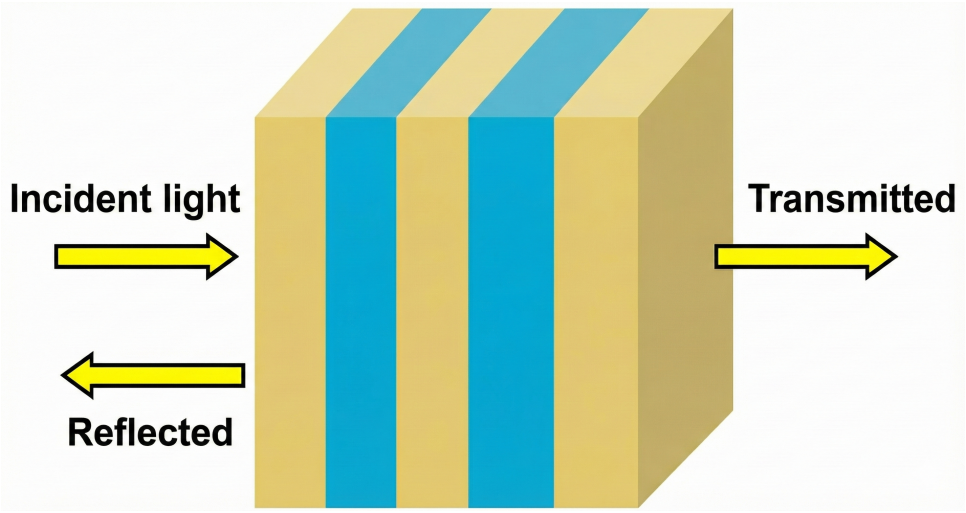

Figure 4: To better illustrate the transmission and reflection of light within the material.

Variable parameters	Number of files
δ	5001
R, δ	5656
X, R, δ	7812

Table 1: Total number of files generated in three experiments.

3.2 Data Structure

The data structure is an essential part of data analysis because it refers to the way datasets are formatted. In this research, the files obtained from the transfer matrix method simulation were organized in tabular format. The *data type* is structured and is continuous. Within each file, each line has four columns, which are the respective values of the reduced frequency $\bar{\omega}$, transmission T, geometric parameter, X and physical parameters of the system R and δ. In summary, for the three experiments the data is organized as shown in Table 2

First column	Second column	Third column	Fourth column	Fifth column
$\bar{\omega}$ (float64)	T (float64)	X (float64)	R (float64)	δ (float64)

Table 2: Summary of content data structure. Data type: float64.

3.3 Data Preprocessing

In this research, preprocessing does not involve data cleaning, data transformation, feature engineering, etc. Instead, it involves the initial organization of the simulated results into a format suitable for machine learning. This step included standardized formatting of the results in text files `.txt` containing columns corresponding to reduced frequency, $\bar{\omega}$, the input parameters (X, R and δ) and the corresponding simulation output (T).

3.3.1 Dividing the Data

For each experiment, the files are generated and randomly divided into three folders: the training set to train the model, the validation set to be used during the training to set the hyperparameters, and the testing set to be used only once at the end to evaluate the final performance of the chosen model. Also, the datasets stored in those folders are completely different from each other. Therefore, to ensure that the files were different in each folder, the library `import random` and `import shuttil` were used. Since the split is randomized, running the code multiple times would generate different partitions of the data. For consistency and reproducibility, the code was executed only once, and the resulting file allocations for train, validation, and test were recorded in `.txt` files and saved for later use. Automatic division into training, validation, and test sets is considered as shown in Table. 3.

Dataset	**Split**
Training	75.0%
Validation	12.5%
Testing	12.5%

Table 3: Proportional distribution of files in the three experiment.

This partitioning was implemented in the TMM code itself, ensuring consistency from the moment the data were generated. Thus, preprocessing ensured that the data that emerged from the TMM simulation were already structured, partitioned, and ready for application in neural network models.

In this context, neural network models operates on two sets of main matrices: the input matrix $\mathbf{x}$ and the output matrix $\mathbf{y}$. For each experiment $\mathbf{x}$ and $\mathbf{y}$ are defined as

Experiment	**Input parameters (x)**	**Output (y)**
1	δ	$T(\bar{\omega})$
2	R, δ	$T(\bar{\omega})$
3	X, R, δ	$T(\bar{\omega})$

Table 4: Summary definition input and output of each experiment

where $\mathbf{X}$, $\mathbf{R}$, δ are the geometric and physical input parameters mapped to the output $T(\bar{\omega})$ which is a non-linear curve.

3.3.2 Normalization of Data

To increase the credibility of the MLP model, the normalization technique was used for the data processing. In general, the normalization step is applied to both the input (X, R and δ) and target vectors $T(\bar{\omega})$ in the dataset Razi et al. (2013). The purpose of this concept is to scale values in order to match input neurons range. It is better to think of the MLP as having a pre-processing block that appears between the input and the first layer of the network and a post-processing block that appears between the last layer of the network and the output, as shown in Figure 5 Eze (2019). In this research, the most common pre-processing and post-processing function, was used to normalize input and output data for the range of $[-1, 1]$ Razi et al.,(2013).

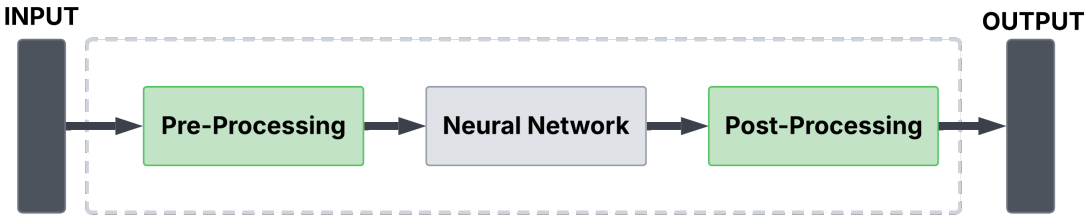

Figure 5: A schematic normalization process of a neural network. Readapted from Razi et al.,(2013)

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x} - \mathbf{x}_{\text{min}}^{\text{train}}}{\mathbf{x}_{\text{max}}^{\text{train}} - \mathbf{x}_{\text{min}}^{\text{train}}} \tag{23}$$

where $\mathbf{x}$ is the input parameter vector. The minimum and maximum values $\mathbf{x}*\text{min}^{\text{train}}, \mathbf{x}*\text{max}^{\text{train}}$ are computed from the training set. The post-processing is

$$\mathbf{y}_{\text{norm}} = \frac{\mathbf{y} - \mathbf{y}_{\text{min}}^{\text{train}}}{\mathbf{y}_{\text{max}}^{\text{train}} - \mathbf{y}_{\text{min}}^{\text{train}}} \tag{24}$$

where $\mathbf{y}$ is the output vector corresponding to the transmission curve $T(\bar{\omega})$. The global limit ${\mathbf{y}_{\text{min}}}^{\text{train}}$, $\mathbf{y}_{\text{max}}^{\text{train}}$ are computed over the entire output matrix of the training set.

3.4 Neural Network Models

Once the data have been generated, structured, and preprocessed, the next crucial step involves selecting the model that best fits the desired predictive performance. In this research, the objective of using neural network models is to learn the mapping function to predict optical properties, such as transmission spectra in three cases: (i) One-parameter prediction (δ), (ii) Two-parameter prediction (R, δ) and (iii) Three-parameter prediction (X, R, δ). Therefore, a Multi-Layer Perceptron (MLP) is selected for its ability to model complex, nonlinear relationships, characteristic of optical spectra and are well established in the literature as benchmark model (Verma et al.,2023; Zhu et al., 2021).

(i) Experiment 1 - One-parameter prediction ($\mathbf{x} = \delta$):

$$f(\delta) \rightarrow T(\bar{\omega})$$

(ii) Experiment 2 - Two-parameter prediction ($\mathbf{x} = R, \delta$):

$$f(R, \delta) \rightarrow T(\bar{\omega})$$

(iii) Experiment 3 - Three-parameter prediction ($\mathbf{x} = X, R, \delta$):

$$f(X, R, \delta) \rightarrow T(\bar{\omega})$$

where $\mathbf{X}, \mathbf{R}, \delta$ are the geometric and physical input parameters mapped to the output $T(\bar{\omega})$.

The following subsections show the fundamental concepts of the neural network used in this research.

3.4.1 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) is a supervised model and is one of the classic deep learning models Goodfellow et al.(2016). In this research, the Multi-Layer Perceptron is trained with backpropagation. The network architecture consists of a collection of "neurons" arranged in layers called the input layer, hidden layer, and output layer that are connected in a way that allows communication between them.

Input layers: the input layer consists of two neurons corresponding to the input *features* $\mathbf{x} \in \mathbb{R}^d$.

Hidden layers: networks have more than one hidden layer. Each layer is composed of multiple neurons, whose function is to extract features from the data. To do this, each neuron individually performs a two-step transformation on its input, such as linear transformation and non-linear activation as shown in figure 6.

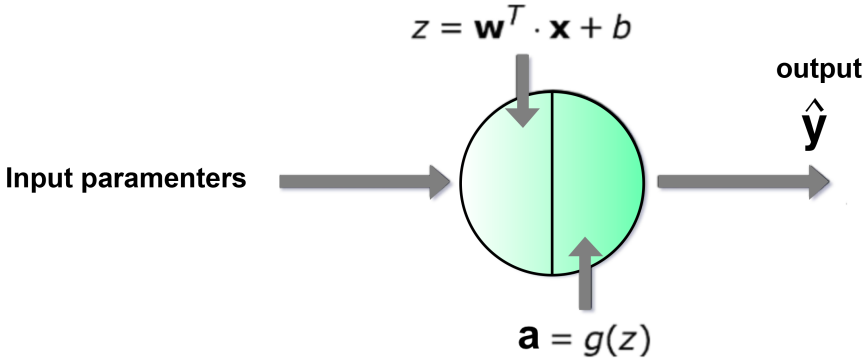

Figure 6: To better illustrate the linear transformation and non-linear activation in a single neuron [2].

(i) *Linear Transformation:* In the linear transformation, the hidden layer computes a pre-activation vector, $\mathbf{z}_{(l)}$, through a matrix multiplication between the activation vector of the previous layer, like $\mathbf{a}_{(l-1)}$, and the weight matrix of the current layer, $\mathbf{w}_{(l)}$, followed by the addition of the bias vector $\mathbf{b}_{(l)}$ the compact form of this operation is as follows:

$$\mathbf{z}_{(l)} = \mathbf{w}_{(l)}^T \mathbf{a}_{(l-1)} + \mathbf{b}_{(l)}. \tag{25}$$

(ii) *Non-Linear Activation:* After the linear transformation, the pre-activation vector $\mathbf{z}_{(l)}$ is then passed through a non-linear activation function g. The element of non-linearity allows us greater flexibility and the creation of complex functions during the learning process. In addition, activation functions are one of the essential

[2]Davi, V.C., 2025. *A deep dive into the mathematics behind neural networks.* Medium. Available at: link [Accessed 7 Dec. 2025].

elements of a neural network. Without them, our neural network would become a combination of linear functions, remaining just a linear function in itself. Also, the activation function also has a significant impact on learning speed, which is one of the main criteria for its selection. For hidden layers, the Rectified Linear Unit (ReLU) function is used:

$$\mathbf{a}_{(l)} = g(\mathbf{z}_{(l)}) = \text{ReLU}(\mathbf{z}_{(l)}) = \max(0, \mathbf{z}_{(l)}). \tag{26}$$

The result of $\mathbf{a}_{(l)} = g(\mathbf{z}_{(l)})$, is the output vector of the hidden layer, which serves as input to the next layer.

Output Layer: The output layer is the final layer of the network, and its purpose is to transform the result from the last hidden layer into the format of the desired prediction. In this study, it applies the same linear transformation described in Eq. (25). However, unlike hidden layers, which use nonlinear activation functions, the output layer employs a "sigmoid" activation function $g(z) = z$ because we want the values returned from the model to be continuous. Mathematically, if the network has l hidden layers, the final prediction $\mathbf{y}$ is the following:

$$\hat{\mathbf{y}} = \sum_i \mathbf{w}^T_{(l+1)} \mathbf{a}_{(l)} + \mathbf{b}_{(l+1)}. \tag{27}$$

The essential step of a neural network is the training process, which is the process that explains how the network learns from errors. For the network to learn, it first needs to quantify how wrong its predictions are. The main source of information on the progress of the model learning process is the cost function. The cost function compares the prediction with the actual value and calculates how incorrect the network is. There are several cost functions in the literature, each suitable for different types of problems. The learning process is the change of values in the parameters w and b where the cost function is minimized.

Cost Function: For this research, we want to predict a continuous value. Therefore, the Mean Squared Error (MSE) was used:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{28}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of samples. This mathematical function calculates the average of the squared differences between predicted and actual values.

Optimization: During the training process, we want to adjust the parameters weights and biases in a way that the cost function L_{MSE} is as small as possible, ideally getting it as close to zero as possible. Minimization of the cost function is a process of optimization that is performed using the gradient descent method, which shows us the direction of greatest growth of the cost function. In each iteration of training, the values of the partial derivatives of the cost function with respect to each weight and bias of the network are calculated using the chain rule of calculus. This mean, if we cause a small variation in weight, $\mathbf{w}_{(l)}$ this automatically cause a small variation in the pre-activation function $\mathbf{z}_{(l)}$, which causes automatically a small variation in $\mathbf{a}_{(l)} = g(\mathbf{z}_{(l)})$, which directly influences of the cost function. This variation then propagates forward through the network, ultimately influencing the cost function as a chain reaction.

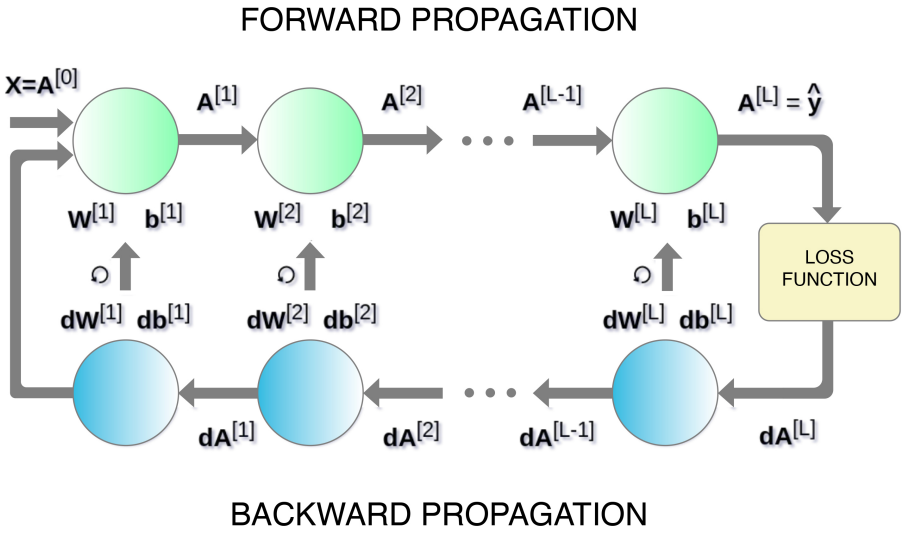

Figure 7: To better illustrate the process of forward and backward propagation in an entire architecture [2].

The gradient of the cost function L_{MSE} with respect to a weight $w_{ji}^{[l]}$ in a layer l with pre-activation $z_j^{[l]}$ and activation $a_j^{[l]}$ is decomposed as:

$$\frac{\partial L_{MSE}}{\partial w_{ji}^{[l]}} = \frac{\partial L_{MSE}}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial w_{ji}^{[l]}} \tag{29}$$

$$= \frac{\partial L_{MSE}}{\partial a_j^{[l]}} \frac{\partial a_j^{[l]}}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial w_{ji}^{[l]}}. \tag{30}$$

Similarly, for the bias $b_j^{[l]}$,

$$\frac{\partial L_{MSE}}{\partial b_j^{[l]}} = \frac{\partial L_{MSE}}{\partial z_j^{[l]}} \frac{\partial z_j^{[l]}}{\partial b_j^{[l]}}. \tag{31}$$

The partial derivatives Eqs. (29-31) of the cost function with respect to $w_{ji}^{[l]}$ and $b_j^{[l]}$ parameters indicate how much and in which direction they should be adjusted.

Finally, to decrease L_{MSE}, we must move the parameters in the opposite direction to the gradient Goodfellow et al.(2016). This needs to be updated iteratively until $w_{ji}^{[l]}$ and $b_j^{[l]}$ reach a global optimal minimum. In this way, the mathematical concept behind the updated parameters to reach the global minimum comes from the first-order Taylor expansion of the cost function,

$$\mathbf{w}^{[l]} = \mathbf{w}^{[l]} - \alpha \frac{\partial L}{\partial \mathbf{w}^{[l]}} \tag{32}$$

$$\mathbf{b}^{[l]} = \mathbf{b}^{[l]} - \alpha \frac{\partial L}{\partial \mathbf{b}^{[l]}} \tag{33}$$

where α represents the learning rate which means the size of the step taken in the opposite direction to the gradient. This has to be updated iteratively until "w" and "b" reach a global minimum.

Adaptive Moment Estimation (Adam): Modern libraries like TensorFlow/Keras does not explicitly compute a Taylor expansion at each iteration. Instead, it directly applies the parameter update rule that was theoretically derived from this expansion. Basically, it initializes two vectors, m and v. A time step counter t is also initialized to zero. For each iteration t, Adam computes the gradient at t [3]. Then, it updates the first-moment vector m, which stores the moving average of the gradients.

$$m_t^{[l]} = \beta_1 m_{t-1}^{[l]} + (1-\beta_1)\frac{\partial L}{\partial \mathbf{w}_t^{[l]}}, \tag{34}$$

Then, similarly, the second-moment vector v is updated [3].

$$v_t^{[l]} = \beta_2 v_{t-1}^{[l]} + (1-\beta_2)\left(\frac{\partial L}{\partial \mathbf{w}_t^{[l]}}\right)^2 \tag{35}$$

The same is applied to biases:

$$m_t^{b[l]} = \beta_1 m_{t-1}^{b[l]} + (1-\beta_1)\frac{\partial L}{\partial \mathbf{b}_t^{[l]}}, \tag{36}$$

$$v_t^{b[l]} = \beta_2 v_{t-1}^{b[l]} + (1-\beta_2)\left(\frac{\partial L}{\partial \mathbf{b}_t^{[l]}}\right)^2 \tag{37}$$

where $\beta_1 = 0.9$ is the exponential decay rate for the 1st moment estimates for default, $\beta_2 = 0.999$ is the exponential decay rate for the 2nd moment estimates for default. Because these moving averages start at zero, they are biased toward zero in early steps and Adam corrects this with:

The update rules for Adam optimizer is:

$$\mathbf{w}_{t+1}^{[l]} = \mathbf{w}_t^{[l]} - \alpha \frac{\hat{m}_t^{[l]}}{\sqrt{\hat{v}_t^{[l]}} + \varepsilon}. \tag{38}$$

The same is applied to biases:

$$\mathbf{b}_{t+1}^{[l]} = \mathbf{b}_t^{[l]} - \alpha \frac{\hat{m}_t^{b[l]}}{\sqrt{\hat{v}_t^{b[l]}} + \varepsilon}. \tag{39}$$

where ε is a small constant for numerical stability for default the value is $1e-7$. In this research, inicially, an initial learning rate is set. This value of 0.001 is a standard, balanced starting point, neither so high as to cause instability nor so low as to make training too slow. A higher learning rate could lead to faster convergence but risks overshooting the minimum, while a lower learning rate ensures more stable convergence but at the risk of getting stuck in local minima or taking too long to converge [3]. The small constant ε is added to prevent any issues with division by zero, which is especially important when the second moment estimate v^t is very small [3]. While Adam is generally robust to noisy data, extreme outliers or highly noisy datasets might impact its performance, but its not our

[3]Cristian Leo, 2024. *The Math behind Adam Optimizer*. Towards Data Science. Available at: link [Accessed 7 Dec. 2025].

case, once don't have noisy. On the other hands, the Adam optimizer nearly always works better due to its ability to adapts the learning rate for each parameter [3]. The benefits of that is faster convergence, better handling of sparse data, and more stable training by allowing each parameter to be updated at a different pace Goodfellow et al.(2016) and Kingma & Ba(2014).

Controlling Overfitting: To improve generalization and to fight overfitting, the following techniques were employed: `Early Stopping` for reducing the size of each parameter dimension are regularization. This is widely used because it is simple to understand and implement and has been reported to be superior to regularization methods in many cases Prechelt (2002). Also the `ReduceLROnPlateau` is used, and its purpose is to track the models performance and reduce the learning rate when there is no improvement for a certain quantity number of epochs. The intuition is that the model approached a sub-optimal solution with current learning rate and oscillate around the global minimum. Reducing the learning rate would enable the model to take smaller learning steps to the optimal solution of the cost function [4].

3.4.2 Evaluation

The evaluation metric is an important part of the training process because it measures how effectively the model achieves its prediction goals. In this research, we performed two different metrics. The first is the training metric, which uses the mean squared error (MSE) equation to evaluate the model performance of MLP. Also, to evaluate how well the model generalizes on the training and validation sets. The second is the evaluation metric, which uses the Euclidean distance (ED) equation to assess prediction accuracy by measuring the distance between the transmission predicted and target (T) . Also, the architectures used are evaluated by compareing the Euclidean mean error on the validation sets. The smaller the Euclidean distance is, the higher the prediction accuracy is Liu & Yu (2019). The mathematical ED is defined as

$$ED_i = \sqrt{\sum_{i=1}^{n}(y_{i,j} - \hat{y}_{i,j})^2} \tag{40}$$

The mean of Euclidean distance is

$$\text{MED} = \frac{1}{N}\sum_{i=1}^{N}\left(\sqrt{\sum_{j=1}^{M}(\mathbf{y}_{i,j} - \hat{\mathbf{y}}_{i,j})^2}\right) \tag{41}$$

The Euclidean Distance (ED) is the L_2 norm of the error between the predicted curve and the real curve. In the context of curve regression, ED is an intuitive and robust measure of how "far" the predicted curve is from the real one. An ED value close to zero means that the MLP is capable of reproducing the physical characteristics of the curve, such as peaks and valleys, with high fidelity.

[4] 2021. *Why models often benefit from reducing the learning rate during training.* StackOverflow. Available at: link [Accessed 7 Dec. 2025].

3.5 Ethical Considerations

In this research, there is no direct human or animal involvement. Ethical considerations focus on transparency, making the methodology and trained models available for verification and reproducibility, and ensuring deep learning codes and datasets will be published in an open repository to ensure accessibility.

4 Implementation

The implementation and reproducibility is a fundamental pillar of scientific investigation. To ensure that the results of this study are reproducible, this section presents the set of hardware, software, and tools used in all stages of the investigation, from data simulation to the training and evaluation of MLP models.

4.1 Hardware Configuration

The computing platform used in our work is a laptop whose configuration is shown in Table 4.1.

Classification	**Name**
CPU	Google Colab
Memory	8 GB
OS	Sequoia Version 15.6 (24G84)

Table 5: Computer configuration.

4.2 Software and Libraries

The training of the neural network models was carried out on the cloud computing platform *Google Colaboratory (Colab)*. The use of this platform allowed access to CPUs. The RAM memory available in the Colab environment was sufficient to manage the datasets and the developed models. The following Table 4.2 details the main tools and libraries employed to produce all experimental outputs and results.

Category	**Tools**
Database Management	Google Drive
IDEs	Google Colab and Visual Studio Code (configured for Jupyter Notebook)
Programming Language	Python (3.12.5)

Table 6: Software used in the proposed research.

All the programs are written in Python programming language because it is open-source and free, easy to learn due to its intuitive syntax, and has an extensive standard library and a wide variety of documented and comprehensive libraries for machine learning. The MLP are developed in `Tensorflow/Keras` for being easy to use. The function of `time.clock()` is used to calculate the running time of programs Liu & Yu (2019). The following Table 4.2 shows the principal libraries in used in each stage process.

Modules	Libraries
Data Generation	Numpy, Pandas, time.clock(), random, shutil
Modelling and Evaluation	Sklearn, TensorFlow/Keras, time.clock(), Matplotlib

Table 7: Libraries used in each module of the proposed research.

4.3 Model Training

The training for the final model was performed using the $TensorFlow/Keras$ library. The model training was performed on the Google Colaboratory. The final MLP model has been trained for approximately 150 (experiment 1) and 100 (experiment 2, 3) epochs using the one cycle learning rate method.

4.3.1 MLP architectures and Hyperparameters

The design of the network architecture and the selection of hyperparameters are crucial to ensure the model's capacity to generalize. In Experiments 1, 2, and 3, deep MLPs were used because the input dimensionality is low. A Deep architecture enables the network to capture oscillations in the input and output mapping, generating a high-dimensional output of 1000 points.

The batch size (BS) determines how many samples are processed before updating the models weights. Choosing an appropriate BS provides a stable and accurate estimate of the gradient, which is especially important for complex regression tasks.

Hidden layers use the ReLU activation function, introducing non-linearity and enabling the network to model the complex shape of the transmission curve $T(\bar{\omega})$. The output layer uses a sigmoid activation to map the predictions because of the oscillatory nature.

The Adam optimizer was selected for its fast and robust convergence. The initial learning rates were chosen to allow rapid progress toward the global minimum, while learning rate schedulers further refine training by reducing the learning rate when progress stalls. Early stopping prevents overfitting by halting training once validation loss stops improving, restoring the best weights.

The ReduceLROnPlateau is a fine-tuning technique for the learning rate. It was chosen to monitor a metric, and if the metric does not improve for a defined number of epochs called `patience`, the learning rate is multiplied by 0.5, effectively halving it. Once the model is near the minimum, reducing the learning rate allows it to take smaller and more precise steps, preventing oscillations and converging to an optimal point in the weight space, which is necessary to achieve a very low Euclidean distance and, consequently, good accuracy.

The following sections summarize the architectures and hyperparameters used in each experiment. Unless otherwise specified, the networks use ReLU activations in hidden layers, Sigmoid activation in the output layer, Adam optimizer, L2 regularization, ReduceLROnPlateau scheduler, and Early Stopping.

Experiment 1 - One-parameter prediction (δ):
For one-parameter prediction, the following six architectures of the MLP are compared:

Model	Hidden Layers	Neurons per Layer
MLP_1-1	2	[192, 96]
MLP_1-2	3	[256, 128, 64]
MLP_1-3	4	[256, 128, 128, 64]
MLP_1-4	6	[384, 384, 192, 192, 96, 96]
MLP_1-5	3	[512, 256, 128]
MLP_1-6	4	[512, 256, 256, 128]

Table 8: Summary of tested MLP architectures (1 Inputs: δ).

Category	Values
Input Dimension	1 (δ)
Output Dimension	1000 (T curve)
Batch Size	128
Epochs	150
Activation Functions	`ReLU()` (hidden layers), `Sigmoid` (output)
Optimizer	`Adam`(learning_rate=5×10^{-4})
Regularization L2	5×10^{-6}
Early Stopping	Monitor: val_loss, Patience: 5, Restore Best Weights: True
Learning Rate Scheduler	`ReduceLROnPlateau`(factor=0.5, patience=8, min_lr=1×10^{-7})
Seed	42 (Python, NumPy, TensorFlow)
Determinism	Enabled (`tf.config.experimental.enable_op_determinism()`)

Table 9: MLP Hyperparameters (1 Input: δ)

Experiment 2 - Two-parameter prediction (R, δ):
For two-parameter prediction, the following six architectures of the MLP are compared:

Model	Hidden Layers	Neurons per Layer
MLP_2-1	2	[192, 96]
MLP_2-2	3	[256, 128, 64]
MLP_2-3	4	[1024, 1024, 384, 384]
MLP_2-4	6	[1024, 1024, 384, 384, 192, 192]
MLP_2-5	3	[512, 256, 128]
MLP_2-6	4	[512, 256, 256, 128]

Table 10: Summary of tested MLP architectures (2 Inputs: R, δ).

Category	Values
Input Dimension	2 (R, δ)
Output Dimension	1000 (T curve)
Batch Size	128
Epochs	100
Activation Functions	`ReLU()` (hidden layers), `Sigmoid` (output)
Optimizer	`Adam`(learning_rate=5×10^{-4})
Regularization L2	1×10^{-6}
Early Stopping	Monitor: val_loss, Patience: 5, Restore Best Weights: True
Learning Rate Scheduler	`ReduceLROnPlateau`(factor=0.5, patience=8, min_lr=1×10^{-7})
Seed	42 (Python, NumPy, TensorFlow)
Determinism	Enabled (`tf.config.experimental.enable_op_determinism()`)

Table 11: MLP Hyperparameters (2 Inputs: R, δ).

Experiment 3 - Three-parameter prediction (X, R, δ):
For three-parameter prediction, the following four architectures of the MLP are compared:

Model	Hidden Layers	Neurons per Layer
MLP_3-1	2	[192, 96]
MLP_3-2	3	[256, 128, 64]
MLP_3-3	4	[1024, 1024, 384, 384]
MLP_3-4	6	[1024, 1024, 384, 384, 192, 192]
MLP_3-5	6	[2048, 2048, 1024, 1024, 512, 512]
MLP_3-6	6	[2048, 1024, 512, 256, 128, 64]

Table 12: Summary of tested MLP architectures (3 Inputs: R, X, δ).

Category	Values
Input Dimension	3 (R, X, δ)
Output Dimension	1000 (T curve)
Batch Size	128
Epochs	200
Activation Functions	`ReLU()` (hidden layers), `Sigmoid` (output)
Optimizer	`Adam`(learning_rate=2×10^{-4})
Regularization L2	1×10^{-6}
Early Stopping	Monitor: val_loss, Patience: 40, Restore Best Weights: True
Learning Rate Scheduler	`ReduceLROnPlateau`(factor=0.5, patience=15, min_lr=1×10^{-7})
Seed	42 (Python, NumPy, TensorFlow)
Determinism	Enabled (`tf.config.experimental.enable_op_determinism()`)

Table 13: MLP Hyperparameters (3 Inputs: R, X, δ).

5 Evaluation

5.1 Experiment 1: One-parameter prediction (δ)

The comparison of the architectures tested for the prediction of two parameters δ is based on the Mean Euclidean Distance (ED) calculated on the validation sets, as detailed in the Fig. 8. This graph was generated with the purpose of showing which type of architecture best learned the complexity of the oscillatory curves of an axionic system in an MLP model with a low-dimensional input and a high-dimensional output.

5.1.1 MLPs Architectures Comparison

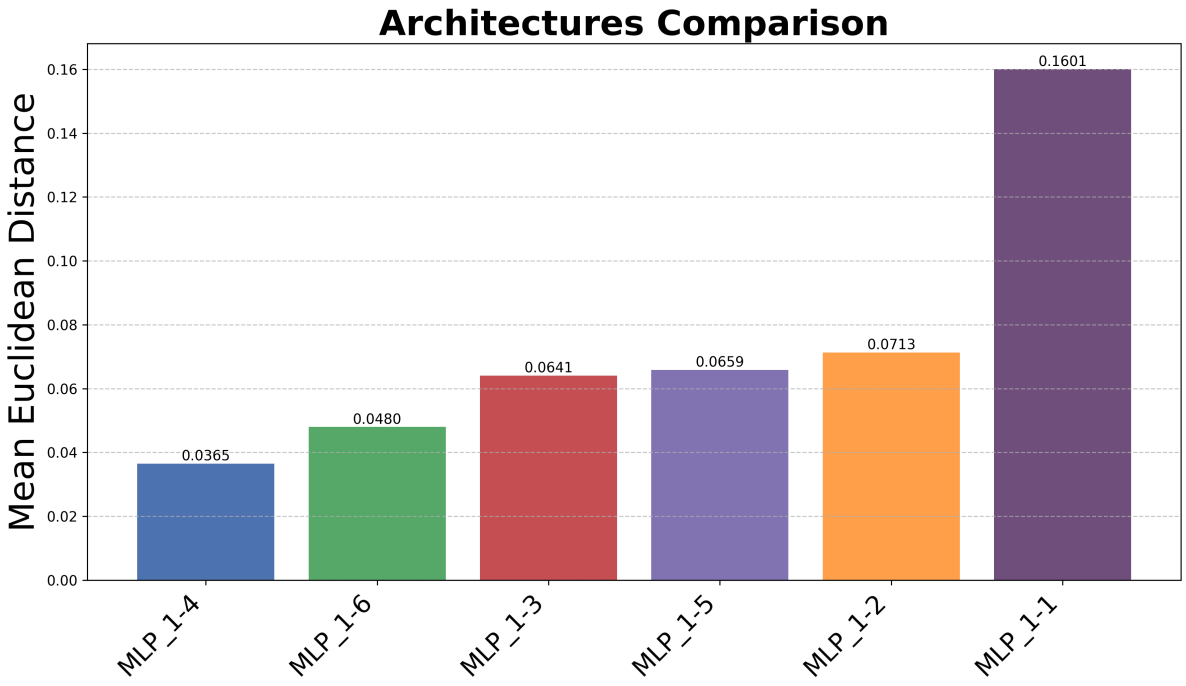

Figure 8: Comparisons of the mean errors of the corresponding validation sets for different MLPs architectures.

The MLP_1-4 model highlighted, achieving the lowest mean error (ED) of 0.0365 and thus being classified as the best-performing architecture for this experiment. In contrast, the MLP_1-1 model showed the highest mean error, with an ED of 0.1601. The intermediate architectures achieved the following error results, in ascending order: MLP_1-6 with 0.0480, MLP_1-3 with 0.0641, MLP_1-5 with 0.0659, and MLP_1-2 with 0.0713. The MLP_1-4 committed less error in the training stage, and it is the best model choice to make transmission predictions.

5.1.2 Transmissions Predictions

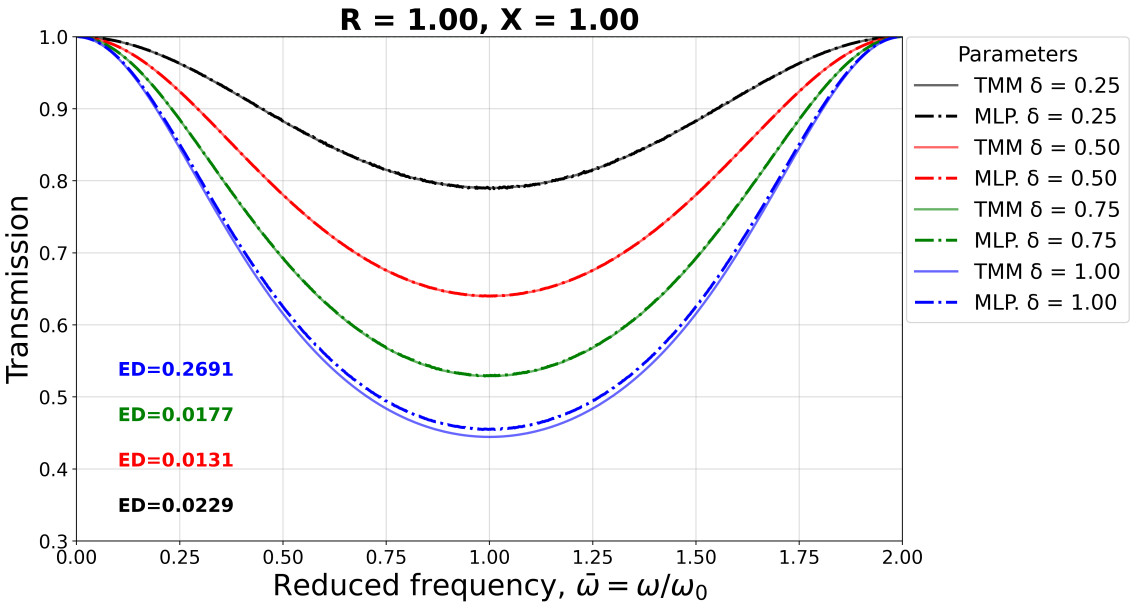

Figure 9: Hidden layer = 6, Neurons = 1344 and Epoch = 200 shows the comparison between the predicted transmission spectra (MLP) with respect to the original transmission spectra (TMM).

Figure 9 shows the transmission predictions of the best model (MLP) as a function of the reduced frequency $\bar{\omega} = \omega/\omega_0$ compared to the reference method (TMM) for different combinations of the parameters $\delta = 0.25, 0.50, 0.75, 1.00$ and fixed values of ratio thickness $X = 1$ and ratio electric permittivity $R = 1$. The solid curves represent the results on transmission obtained using the reference method Transfer Matrix Method (TMM), while the dashed curves represent the transmission predictions of the Multilayer Perceptron (MLP) model.

It can be observed that increasing the value of δ modifies the amplitude of the transmission spectra. The analysis reveals that, for $\delta = 0.25$, the $ED = 0.2691$. Then, for $\delta = 0.50$ and $\delta = 0.75$, a significant reduction in the errors was observed, with $ED = 0.0177$ and $ED = 0.0131$ showing that the physical behavior calculated by the MLP_1-4 model accurately reproduces the traditional TMM method. This is an indication that the MLP_1-4 model has correctly learned the nonlinear relationship between the input variable δ and the transmission response $T(\bar{\omega})$. On the other hand, the error increases for $\delta = 1.00$ with $ED = 0.0229$.

From the models perspective, the explanation for a smooth curve $\delta = 0.25$ with low amplitude having an error only slightly higher than the values for $\delta = 0.50$ and $\delta 0.75$ is related to the gradient used to minimize the prediction error. A near-zero error gradient provides a very weak correction signal for the neural network. This suggests that the learning algorithm *Gradient Descent* does not find a clear "reference point" and may stop making adjustments even if the ideal accuracy has not been reached, because the cost error changes very little at each step.

On the other hand, the transmission curve for the value of $\delta = 1.00$ presents a deeper valley. At the edges where $\tilde{\omega} \approx 0.25$ and $\tilde{\omega} \approx 1.75$, it is noticeable that the MLP begins to deviate from the exact shape of the curve generated by the TMM. The error becomes

even more visible when $\tilde{\omega} \approx 1.00$, reaching its peak at $\tilde{\omega} = 1.00$, the region that represents the greatest difference between the curve predicted by the MLP and the curve generated by the TMM. This may indicate that the activation functions of the MLP model are in a region where the derivative is very small, and the neuron can no longer adjust its weights effectively, suggesting that this regime is the most difficult for the model. This suggests a saturation of the Sigmoid activation function in the last layer, where the model consistently fails to replicate values close to 1 or 0, while performing well in the middle of the interval, as occurs with its $\delta = 0.50$ and $\delta = 0.75$.

It can be seen that the predicted transmission for $\delta = 0.25, 0.50, 0.75$ are in good agreement with the TMM target values. However, the predicted transmission for $\delta = 1.00$ has a difference of 0.26 in its curves between MLP predicted and TMM target values. In general, the MLP exhibit good performances for predicting the transmission of APCs with different δ values.

5.1.3 Model Convergence

The Fig. 10 is essential to identifying model generalization capability, and detecting issues like overfitting or underfitting.

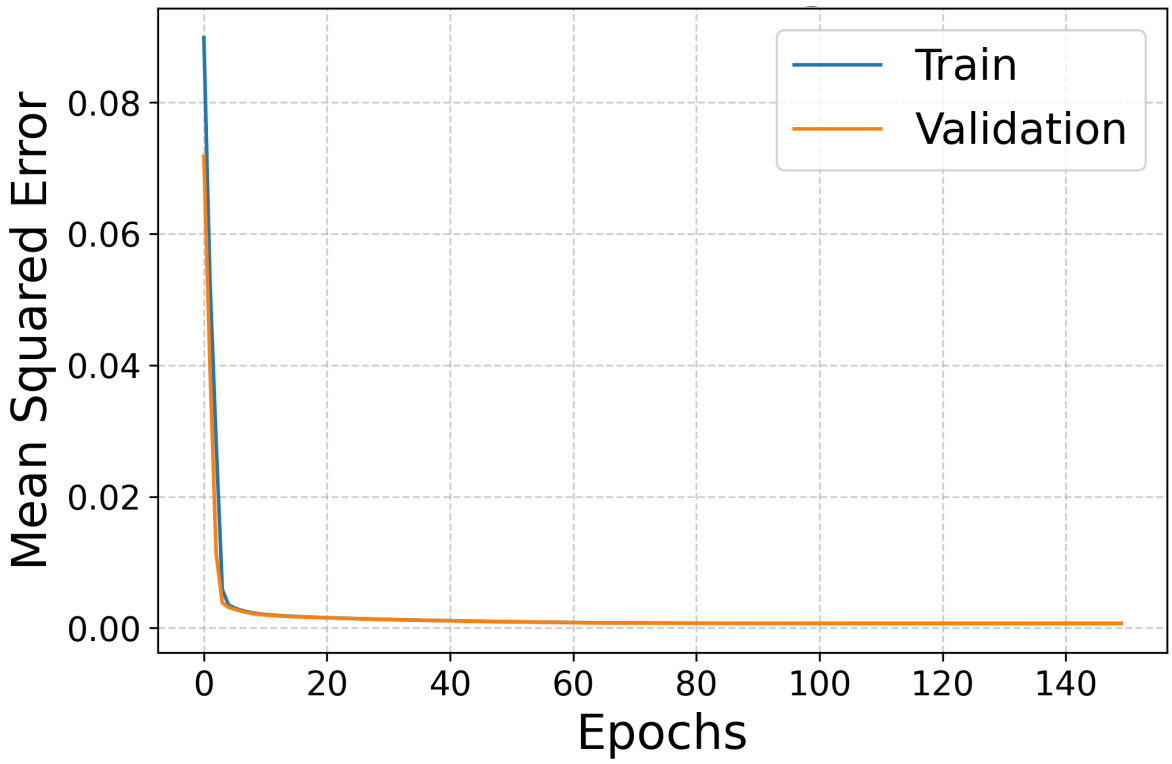

Figure 10: Variation of mean squared error (MSE) vs. the number of epochs for training and validation dataset.

Figure 10 shows variation of the of the Mean Squared Error (MSE) versus number of epochs for training and validation datasets during the learning process of the "MLP_1–4" model. The results correspond to the model architecture configuration with 6 hidden layers, 1344 neurons, and a total of 150 training epochs.

The x-axis represents the number of training epochs, which corresponds to the iterations over the entire training dataset. The y-axis shows the Mean Squared Error (MSE),

which quantifies the average squared difference between the MLP predicted and TMM values.

The rapid decrease of the values of MSE for both the training and validation during the initial epochs approximately up to epoch 25, the close alignment between the training and validation curves with no evident signs of overfitting or underfitting and the low and stable MSE values suggest that the model is well set and generalizes well to unseen data, the data are well normalized, and the optimizer efficiently adjusted the network weights causing fast convergence, reaching an almost constant value for epochs greater than approximately 75 suggesting effective learning in the early stages.

5.2 Experiment 2: Two-parameter prediction (R, δ)

The comparison of the architectures tested for the prediction of two parameters δ is based on the Mean Euclidean Distance (ED) calculated on the validation sets, as detailed in the Fig. 11. The comparison of the architectures tested for the prediction of two parameters (R, δ) is based on the Mean Euclidean Distance (ED) calculated in the validation sets, as detailed in Figure 11. The MLP_2-4 model highlighted, achieving the lowest mean error (ED) of 0.0914 and thus being classified as the best performing architecture for this experiment. In contrast, the MLP_2-2 model showed the highest mean error, with an ED of 0.4475. The intermediate architectures achieved the following error results, in ascending order: MLP_2-6 with 0.1391, MLP_2-3 with 0.1658, MLP_2-1 with 0.2243, and MLP_2-5 with 0.4443.

5.2.1 MLPs Archithectures Comparison

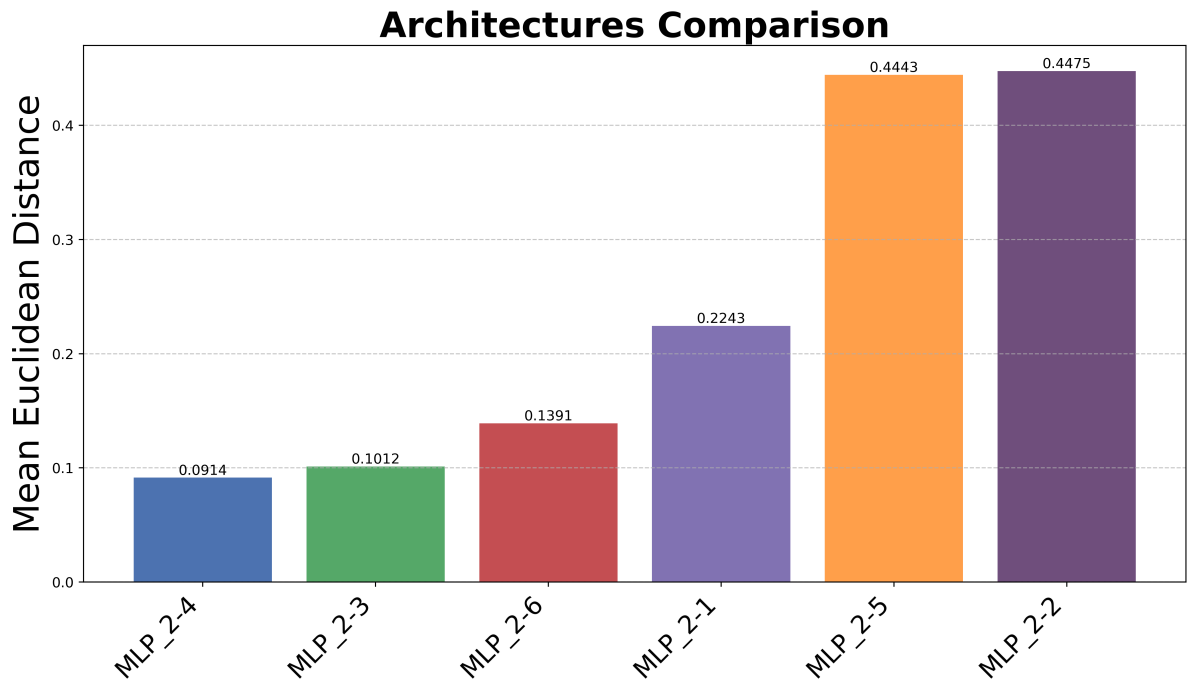

Figure 11: Comparisons of the mean errors of the corresponding validation sets for different MLPs architectures.

5.2.2 Transmission Predictions

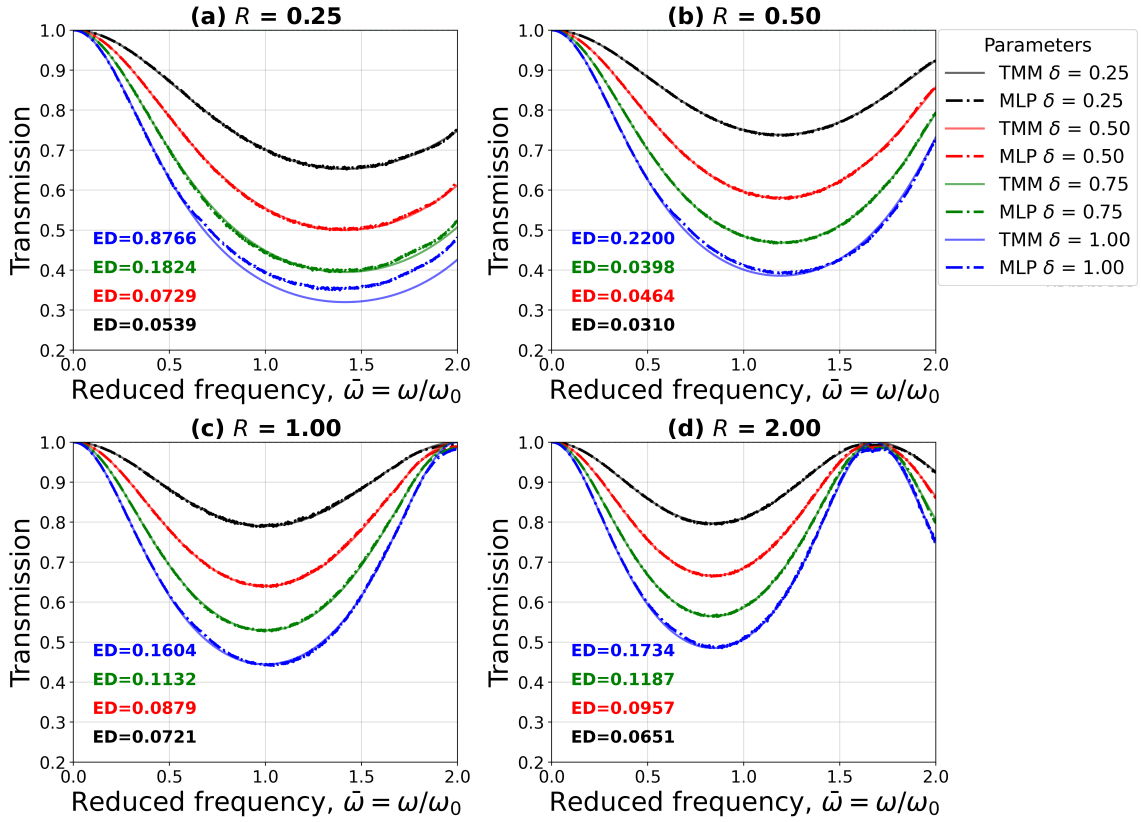

Figure 12: Hidden layer = 6, Neurons = 3200 and Epoch = 100 shows the comparison between the predicted transmission spectra (MLP) with respect to the original transmission spectra (TMM). For $X = 1$ with (a) $R = 0.25$, (b) $R = 0.50$, (c) $R = 1.00$ and (d) $R = 2.00$.

Figure 12 shows the transmission predictions of the best model MLP_2-4 compared with the reference method (TMM) for different combinations of parameters R and δ. The analysis reveals that, for $R = 0.25$ in Figure 12(a), the Euclidean Distances (ED) ranged from $ED = 0.0539$ for $\delta = 1.00$ to $ED = 0.8766$ for $\delta = 0.25$. Next, for $R = 0.50$ in Figure 12(b), a significant reduction in errors was observed, with ED values ranging from $ED = 0.0310$ for $\delta = 0.25$ to $ED = 0.2200$ for $\delta = 1.00$. For $R = 1.00$ in Figure 12(c), the ED values remained within the range of $ED = 0.0721$ for $\delta = 0.25$ to $ED = 0.1604$ for $\delta = 1.00$. Finally, at $R = 2.00$ in Figure 12(d), the ED values varied from $ED = 0.0651$ for $\delta = 0.25$ to $ED = 0.1734$ for $\delta = 1.00$.

Table 14 shows the accuracy ED corresponding to the mean error between the two curves for a specific pair of parameters (R and δ).

R	$\delta = 0.25$	$\delta = 0.50$	$\delta = 0.75$	$\delta = 1.00$
0.25	0.0539	0.0729	0.1824	0.8766
0.50	0.0310	0.0464	0.0398	0.2200
1.00	0.0721	0.0879	0.1132	0.1604
2.00	0.0651	0.0957	0.1187	0.1734

Table 14: Summary of the Euclidian Distance error values obtained from the MLP vs. TMM comparison.

After analyse the accurancy of the model, we should ask how does the performance of the MLP model vary as δ changes, considering all possible variations of R? To answer this question, for each value of δ, we collected all the error values (ED) corresponding to that δ, considering all variations of R. From this set of EDs, we calculated the mean and standard deviation, thus obtaining a summarized measure of the models performance for that specific δ. This procedure was repeated for each value of δ, allowing a clear assessment of how the models error varies with δ.

$$\overline{ED}_\delta = \frac{1}{4}\sum_R ED(R,\delta) \tag{42}$$

$$\sigma_\delta = \sqrt{\frac{1}{4}\sum_R \left(ED(R,\delta) - \overline{ED}_\delta\right)^2} \tag{43}$$

As the axionic photonic system depends on δ, it is important to analyze how the performance of the MLP changes systematically as δ increases. Table 15 numerically illustrates the behavior of the MLP_2-4 model.

δ	$\overline{ED}_\delta$	σ_δ	MLP Performance
0.25	0.0555	0.0156	Best performance and most stable
0.50	0.0757	0.0188	Good performance
0.75	0.1135	0.0505	Fair
1.00	0.3576	0.3005	Poor and highly unstable

Table 15: Overall error metrics of the MLP for each value of δ.

We can observe from Table 15 that the discretization of δ is uniform, but accordigly to the Fig. 12 the transmission spectrum changes very rapidly near $\delta = 1.0$. As δ increases, the axionic term introduces more irregularities in the transmission spectrum, leading to sharper oscillations with narrow peaks and valleys. The *Gradient Descent* does not find a clear "reference point" and may stop making adjustments.

5.2.3 Model Convergence

Figure 13 illustrates the variation of the Mean Squared Error (MSE) over the training epochs for both the training and validation datasets. The MSE curves for training and validation show a rapid decrease during the initial epochs, indicating an efficient learning process. Additionally, the curves remain closely aligned and tend to stabilize at low values, suggesting a good generalization capability and showing no evident signs of *overfitting* or *underfitting*.

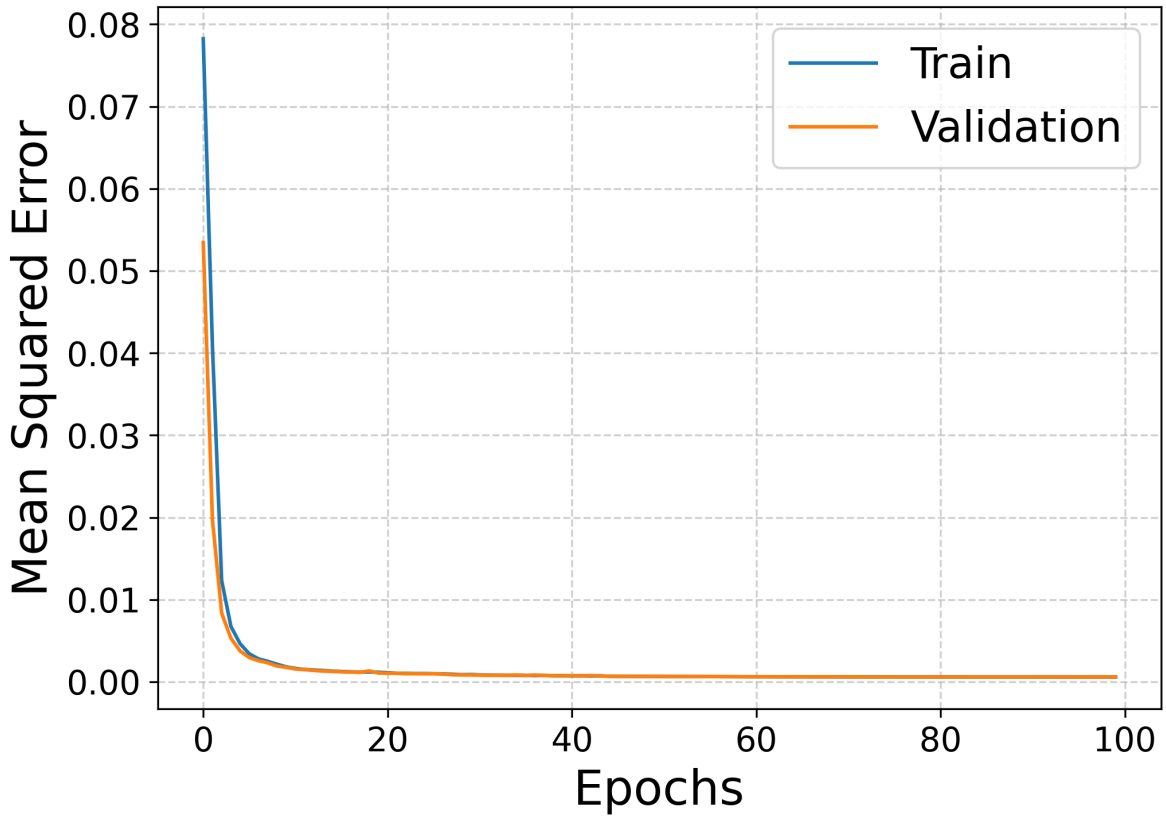

Figure 13: Variation of mean squared error (MSE) vs. the number of epochs for training and validation dataset.

5.3 Experiment 3: Three-parameter prediction (X, R, δ)

The comparison of the architectures tested for the prediction of two parameters δ is based on the Mean Euclidean Distance (ED) calculated on the validation sets, as detailed in the Fig. 14. The comparison of the architectures tested for the prediction of three parameters (X, R, δ) is based on the Mean Euclidean Distance (ED) calculated in the validation sets, as detailed in Figure 14. The MLP_3-4 model highlighted, achieving the lowest mean error (ED) of 0.2109 and thus being classified as the best performing architecture for this experiment. In contrast, the MLP_3-3 model showed the highest mean error, with an ED of 0.3285. The intermediate architectures achieved the following error results, in ascending order: MLP_3-5 with 0.03528, MLP_3-6 with 0.4231, MLP_3-2 with 0.6883, and MLP_3-1 with 1.3073.

5.3.1 Architectures Comparison

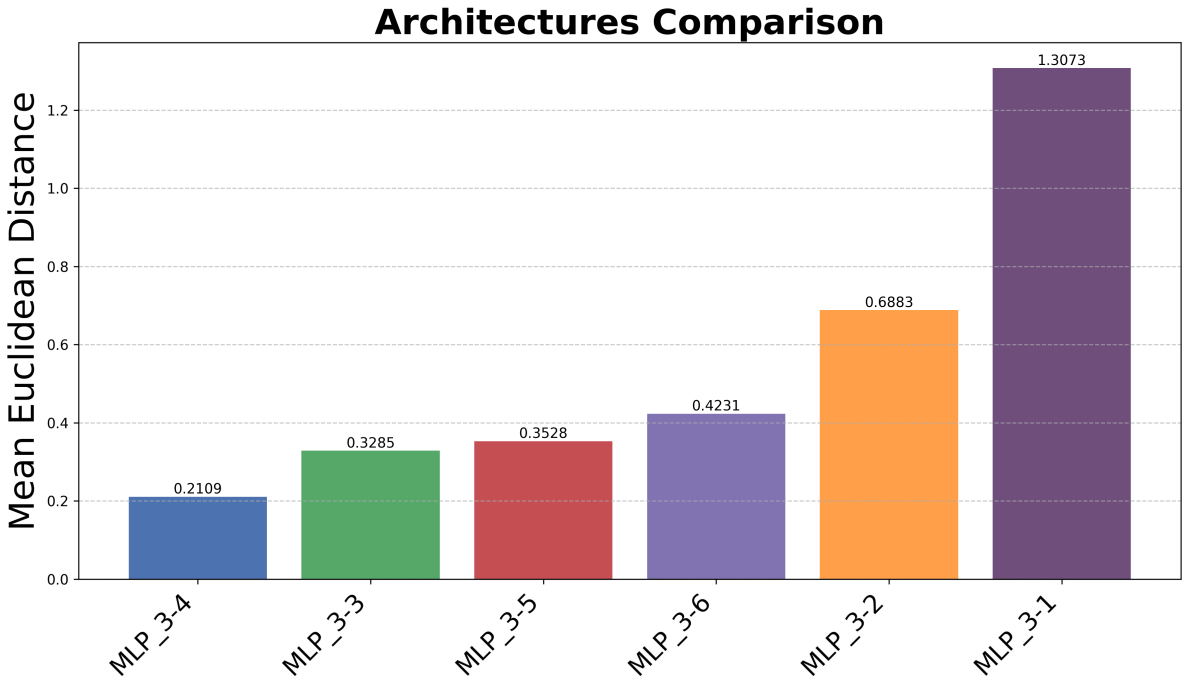

Figure 14: Comparisons of the mean errors of the corresponding validation sets for different MLPs architectures.

5.3.2 Transmission Predictions

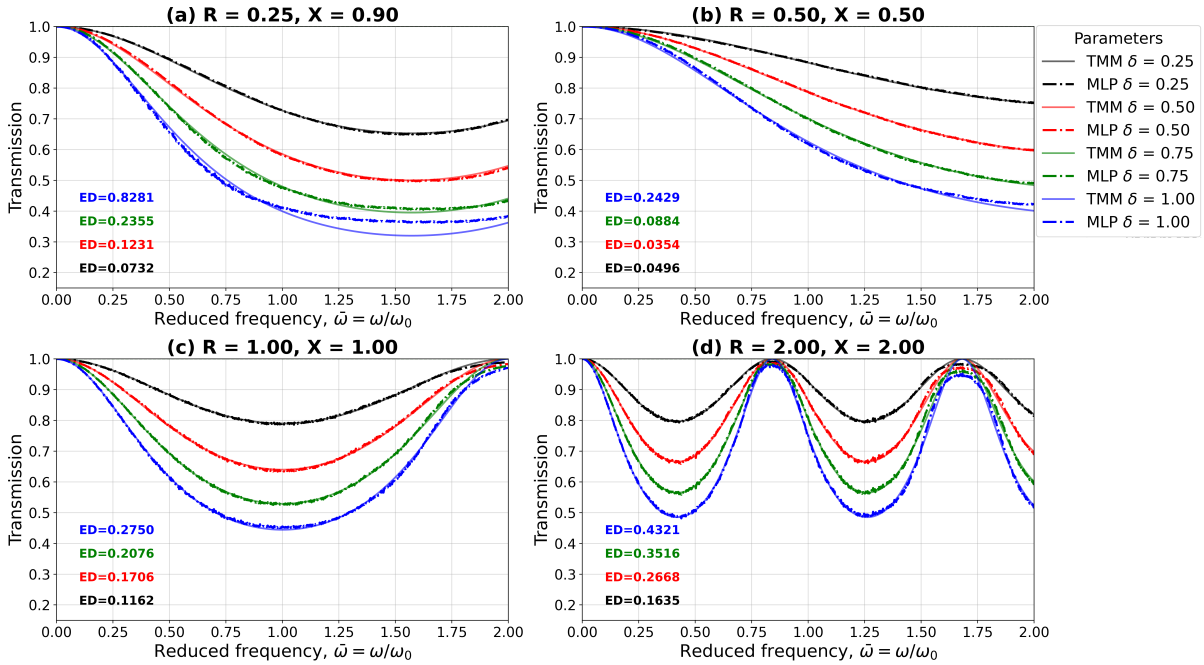

Figure 15: Hidden layer = 6, Neurons = 3200 and Epoch = 200 shows the comparison between the predicted transmission spectra (MLP) with respect to the original transmission spectra (TMM). For (a) $R = 0.25$, $X = 0.90$, (b) $R = X = 0.50$, (c) $R = X = 1.00$ and (d) $R = X = 2.00$.

Figure 15 shows the transmission predictions of the best model MLP_3-4 compared with the reference method (TMM) for different combinations of parameters R, X and δ. The analysis reveals that, for $R = 0.25$ and $X = 0.90$ in Figure 15(a), the Euclidean Distances (ED) ranged from $ED = 0.0732$ for $\delta = 0.25$ to $ED = 0.8281$ for $\delta = 1.00$. Next, for $R = 0.50$ and $X = 0.50$ in Figure 15(b), the errors were generally low, with the minimum error being $ED = 0.0354$ (at $\delta = 0.50$) and the maximum $ED = 0.2429$ (at $\delta = 1.00$). For $R = 1.00$ and $X = 1.00$ in Figure 15(c), the ED values ranged from $ED = 0.2750$ for $\delta = 0.25$ down to $ED = 0.1162$ for $\delta = 1.00$. Finally, at $R = 2.00$ and $X = 2.00$ in Figure 15(d), the ED values varied from $ED = 0.4321$ for $\delta = 0.25$ down to $ED = 0.1635$ for $\delta = 1.00$.

Table 16 shows the specific ED accuracy for each combination of parameters (R, X) and δ.

(R, X)	$\delta = 0.25$	$\delta = 0.50$	$\delta = 0.75$	$\delta = 1.00$
(0.25, 0.90)	0.0732	0.1231	0.2355	0.8281
(0.50, 0.50)	0.0496	0.0354	0.0884	0.2429
(1.00, 1.00)	0.2750	0.2076	0.1706	0.1162
(2.00, 2.00)	0.4321	0.3516	0.2668	0.1635

Table 16: Summary of the Euclidian Distance error values obtained from the MLP vs. TMM comparison for **MLP_3-4**.

After analysing the accurancy of the model, we should ask how does the performance of the MLP model vary as δ changes, considering the 4 pairs of (R, X) tested. To answer

this question, for each value of δ, we collected all the error values (ED) corresponding to that δ from Table 16. From this set of 4 EDs, we calculated the mean and standard deviation, thus obtaining a summarized measure of the models performance for that specific δ.

$$\overline{ED}_\delta = \frac{1}{4}\sum_{i=1}^{4} ED_i(\delta) \tag{44}$$

$$\sigma_\delta = \sqrt{\frac{1}{4}\sum_{i=1}^{4}\left(ED_i(\delta) - \overline{ED}_\delta\right)^2} \tag{45}$$

As the axionic photonic system depends on δ, it is important to analyze how the performance of the MLP changes systematically as δ increases. Table 17 numerically illustrates the behavior of the **MLP_3-4** model based on the data from Table 16.

δ	$\overline{ED}_\delta$	σ_δ	MLP_3-4 Performance
0.25	0.2075	0.1585	Fair performance, but unstable
0.50	0.1794	0.1132	Best performance and relatively stable
0.75	0.1903	0.0718	Good performance and most stable
1.00	0.3377	0.2831	Poorest performance and highly unstable

Table 17: Overall error metrics of the **MLP_3-4** model for each value of δ, calculated from Table 16.

5.3.3 Model Convergence

Figure 16 illustrates the variation of the Mean Squared Error (MSE) over the training epochs for both the training and validation datasets. The MSE curves for training and validation show a rapid decrease during the initial epochs approx. 25, indicating an efficient learning process. Additionally, the curves remain closely aligned and tend to stabilize at low values, suggesting a good generalization capability. The validation loss exhibits several small, sharp spikes around epochs 100, 110, and 115. The training loss remains smooth. This suggests minor, temporary instability and shows no evident signs of *overfitting* or *underfitting*.

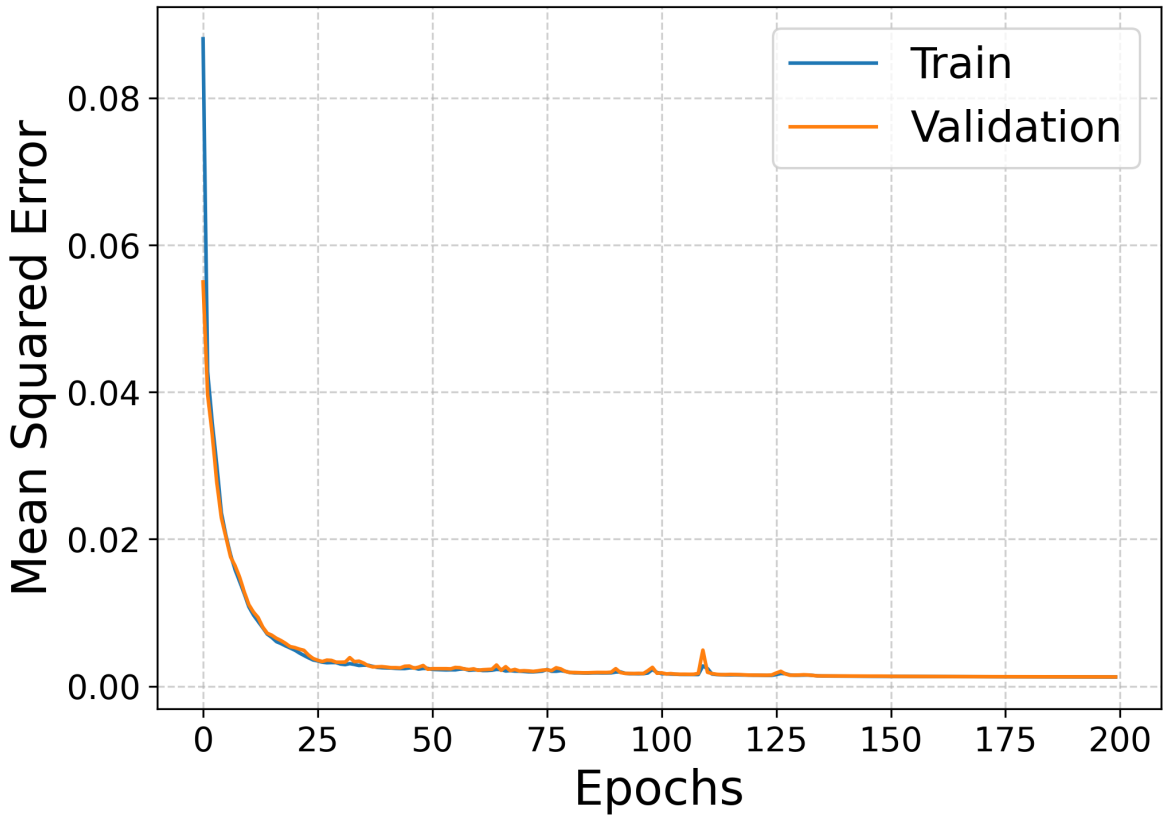

Figure 16: Variation of mean squared error (MSE) vs. the number of epochs for training and validation dataset.

5.4 Discussion

Parameters	Method	Model training time (s)	Calculation time (s)	Mean of ED
δ	MLP_1-4	73.06s	0.00012954s	0.036494
R, δ	MLP_2-4	283.30s	0.00019654s	0.091408
R, X, δ	MLP_3-4	3085.06s	0.00047988s	0.205738
	TMM	–	0.105	–

Table 18: Comparison between the MLP and transfer matrix method. Model training time refers to the time training a neural network needs; Calculation time refers to the average time taken by the trained neural networks or the TMM to calculate the transmission of a APC; Mean of ED is the mean of the accuracies of the corresponding testing set Liu & Yu (2019).

Table 5.4 presents a performance comparison among neural network models (MLP) and the traditional transfer matrix method (TMM). The analysis focuses on three main metrics, such as training time, calculation time, and accuracy measured by mean euclidean distance. It can be noticed that the time required by MLPs is extremely short, and the prediction accuracy is remarkable, similarly to Liu & Yu (2019). For one-parameter prediction, the δ are superior to the two-parameter (R, δ) and three-parameter (X, R, δ) on training time, prediction accuracy, and simplicity of the model. For one-parameter

prediction, the MLP has a smaller mean error, similarly to Liu & Yu (2019). Also one-parameter prediction, the MLP_1-4 is a better choice because of its high accuracy and stability. In terms of calculation time, the TMM (0.105s) is significantly slower than a trained MLP.

6 Conclusion and Future Work

This research addressed the limitations of current computational methods used by TMM in the analysis of one-dimensional axionic photonic crystals (1DAPCs) and the use of neural networks such as the Multi-Layer Perceptron (MLP) to investigate the accuracy and computational time consumption in predicting the transmission optical spectrum, focusing on evaluating how the model generalizes for one, two, and three system parameters. Key findings indicate that the MLP has positive implications for time efficiency, achieving calculation times as low as 10^{-4} s compared to 10^{-1} s for TMM, representing a drastic reduction in processing time. Additionally, the research underscores significant accuracy in low-complexity scenarios, particularly in the one-parameter experiment where the model MLP_1-4 achieved an accuracy of ED = 0.0365. The model also revealed sensibility within the model's generalization capabilities in high-complexity regimes, specifically when the axionic term (δ) approaches 1.0, leading to increased error rates and instability due to abrupt oscillations in the transmission spectra. The implications of these findings underline the urgent need for researchers to integrate machine learning into the simulation and optimization of photonic devices. Incorporating these neural network models could mitigate computational bottlenecks and promote the development of photonic technologies. Moreover, the study contributes to existing literature by highlighting the feasibility of using MLPs as a surrogate for TMM, prompting stakeholders to consider deep learning as an integral tool for the rapid analysis of optical properties. This research acknowledges limitations such as the models difficulty in accurately predicting phenomena characterized by high non-linearity and high spectral peaks, particularly in the three-parameter experiments where the mean ED becomes the worst for transmission regions around 0.5. Additionally, the analysis utilized forward prediction perspectives, thus not exploring the complexities of inverse design. Future research directions should focus on refining neural network architectures to reduce errors in high-oscillation regimes and provide more precise predictions for complex parameter combinations. Investigations of inverse design and optimization would also enrich the current understanding, enabling the study of crystal structures based on desired optical outputs. Further studies examining the effectiveness of different deep learning models beyond MLPs could significantly contribute to the optimization of axionic photonic devices. In conclusion, MLP is highly efficient on calculation time. The model achieved its best accuracy in the one-parameter experiment, with a mean ED = 0.0365. However, in the three experiments the results indicate the MLP generalizes well in low-complexity regimes but struggles to model abrupt oscillations that occur in high-non-linearity parameter regimes (near $\delta = 1.0$), where performance was poor and unstable. The three experiments showed that the computational challenges posed by traditional methods in the analysis of axionic photonic crystals can be overcome by neural networks, ensuring efficiency in optical simulations and the development of future axionic photonic devices.

References

Araújo, A. C., Costa, C. H., Azevedo, S., Viswanathan, G. M. & Bezerra, C. G. (2025), 'Optical properties of 1D photonic time quasicrystals', *Journal of the Optical Society of America B* **42**, 949–963. Available at: https://doi.org/10.1364/JOSAB.542001.

Araújo, A., Azevedo, S., Furtado, C., Chaves, A., Costa, C. & Bezerra, C. (2021), 'Transfer-matrix method of circular polarization light in an axionic photonic insulator', *Physical Review A* **104**(5), 053532. Available at: https://doi.org/10.1103/PhysRevA.104.053532.

Dhanabalan, S., Thirumurugan, A., Raju, R., Kamaraj, S. & Thirumaran, S., eds (2023), *Photonic Crystal and Its Applications for Next Generation Systems*, Springer, Singapore. Available at: https://doi.org/10.1007/978-981-99-2548-3.

Eze, E. C. (2019), Wind farm power output prediction based on machine learning recurrent neural networks, PhD thesis, University of Sussex. Available at: https://hdl.handle.net/10779/uos.23473442.v1 (accessed 27 January 2026).

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, MA. Available at: http://www.deeplearningbook.org.

Granada, E. & Rojas, D. (2017), 'Transfer matrix method in systems with topological insulators', *Journal of Physics: Conference Series* **850**(1), 012024. Available at: https://doi.org/10.1088/1742-6596/850/1/012024.

Jabin, M. & Fok, M. (2022), 'Prediction of 12 photonic crystal fiber optical properties using MLP in deep learning', *IEEE Photonics Technology Letters* **34**(7), 391–394. Available at: https://doi.org/10.1109/LPT.2022.3157266.

Joannopoulos, J., Johnson, S., Winn, J. & Meade, R. (2008), *Photonic Crystals: Molding the Flow of Light*, 2nd edn, Princeton University Press, Princeton, NJ.

Kingma, D. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* . Available at: https://doi.org/10.48550/arXiv.1412.6980.

Li, J., Qian, M., Yin, J., Lin, W., Zhang, Z. & Liu, S. (2025), 'Topology design of soft phononic crystals for tunable band gaps: a deep learning approach', *Materials* **18**(2), 377. Available at: https://doi.org/10.3390/ma18020377.

Liu, C. & Yu, G. (2019), 'Predicting the dispersion relations of one-dimensional phononic crystals by neural networks', *Scientific Reports* **9**, 15322. Available at: https://doi.org/10.1038/s41598-019-51662-3.

Long, Y., Ren, J., Li, Y. & Chen, H. (2019), 'Inverse design of photonic topological state via machine learning', *Applied Physics Letters* **114**(18), 181105. Available at: https://doi.org/10.1063/1.5094838.

Long, Y., Zou, L., Yu, L., Hu, H., Xiong, J. & Zhang, B. (2024), 'Inverse design of topological photonic time crystals via deep learning', *Optical Materials Express* **14**(8), 2032–2039. Available at: https://doi.org/10.1364/OME.525396.

Markoš, P. & Soukoulis, C. (2008), *Wave Propagation: From Electrons to Photonic Crystals and Left-Handed Materials*, Princeton University Press, Princeton, NJ.

Prechelt, L. (2002), *Early stopping but when?*, Springer, Berlin. Available at: https://doi.org/10.1007/3-540-49430-8_3.

Razi, M., Arz, A. & Naderi, A. (2013), 'Annular pressure loss while drilling prediction with artificial neural network modeling', *European Journal of Scientific Research* **95**(2), 272–288. Available at: https://www.researchgate.net/publication/239735549.

Verma, S., Sharma, A. & Gupta, R. (2023), 'A comprehensive deep learning method for empirical spectral prediction and its quantitative validation of nano-structured dimers', *Scientific Reports* **13**(1), 1129. Available at: https://doi.org/10.1038/s41598-023-28076-3.

Zhan, T., Liu, Q., Sun, Y., Qiu, L., Wen, T. & Zhang, R. (2022), 'A general machine learning-based approach for inverse design of one-dimensional photonic crystals toward targeted visible light reflection spectrum', *Optics Communications* **510**, 127920. Available at: https://doi.org/10.1016/j.optcom.2022.127920.

Zhu, S., Wang, J. & Zhang, Y. (2021), 'RBF neural network-based frequency band prediction for future frequency hopping communications', *Wireless Communications and Mobile Computing* **2021**, 5570685. Available at: https://doi.org/10.1155/2021/5570685.