

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Physics-informed machine learning
for electromagnetism

VIKTOR AADLAND LILJA

Department of Physics and Astronomy
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2026

Physics-informed machine learning for electromagnetism
VIKTOR AADLAND LILJA

© Viktor Aadland Lilja, 2026

Department of Physics and Astronomy
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone +46 (0)31 772 10 00

Cover: Maxwell teaches physics to a machine.

Chalmers digitaltryck
Göteborg, Sweden 2026

Physics-informed machine learning for electromagnetism

VIKTOR AADLAND LILJA
Department of Physics and Astronomy
Chalmers University of Technology

Abstract

Electromagnetism underpins a wide range of modern technologies, from electric machines and antennas to optical components and photonic circuits. To design these devices, accurate and efficient modeling of electromagnetic phenomena is essential. Traditional numerical techniques such as finite-element and finite-difference methods have achieved great success over many decades, but their high computational cost becomes a bottleneck as electromagnetic devices grow in complexity.

In recent years, machine learning has emerged as a powerful tool for accelerating electromagnetic modeling by replacing costly numerical simulations with fast surrogate models. Such models can enable rapid exploration of large design spaces and thereby support the development of higher-performing devices. However, purely data-driven models often require impractical amounts of training data and may fail to respect fundamental physical laws. Physics-informed machine learning addresses these limitations by embedding prior knowledge about the governing physics into the model architecture or learning process.

In this thesis, I explore new methods for physics-informed machine learning in electromagnetism, with the aim of improving data efficiency, reliability, and interpretability of neural network-based surrogate models. The thesis centers on two complementary methods. The first exploits the resonant structure underlying electromagnetic scattering spectra, while the second utilizes symmetries in Maxwell's equations. By developing widely applicable tools to improve surrogate models, the work contributes to advancing electromagnetic analysis and design.

Keywords: machine learning, electromagnetism, scattering, neural networks, inverse design, quasinormal modes, symmetry

LIST OF APPENDED PAPERS

- I A General Framework for Knowledge Integration in Machine Learning for Electromagnetic Scattering Using Quasinormal Modes**
Viktor A. Lilja, Albin J. Svärdsby, Timo Gahlmann, and Philippe Tassin
Laser & Photonics Reviews **20**, e02769 (2026)
- II Symmetry-Informed Deep Learning for Electromagnetic Scattering**
Viktor A. Lilja and Philippe Tassin
Submitted for publication, preprint at arXiv:2607.12810

The author's contribution to the papers:

- I Co-conceived the idea, implemented the code, performed the numerical analysis, wrote the manuscript, and contributed to revisions of the manuscript.
- II Conceived the idea, implemented the code, performed the numerical analysis, and wrote the manuscript.

Contents

1	Introduction	1
1.1	Thesis overview	2
2	Neural network surrogate models	5
2.1	Motivation: Inverse design	5
2.2	Neural networks	7
2.3	When are data-driven models useful?	9
2.4	Physics-informed machine learning	10
3	Theory of electromagnetism	13
3.1	Electromagnetic fields in linear local media	13
3.2	Fields as vectors	14
3.3	Guided modes and the scattering matrix	15
3.4	Numerical solvers	17
4	Resonances as an inductive bias	19
4.1	Resonance-based physics layers	19
4.2	Normal and quasinormal modes	21
4.3	QNM expansion of the scattering matrix	22
4.4	The QNM-Net	25
5	Symmetry as an inductive bias	27
5.1	Example problem	28
5.2	Introduction to group theory	29
5.3	Equivariant neural networks	31
5.4	Implementing equivariant layers	32
5.5	Implications of equivariance	34
6	Summary of papers	35
7	Conclusions and outlook	37
	Acknowledgments	39
	Bibliography	41
	Papers I–II	47

Introduction

Electromagnetic devices underpin nearly every aspect of our modern lives. Electricity powers the alarm clock that wakes us up in the morning. We go to work on bikes, trains, and cars powered by electric motors. Many of us work in front of computer screens that emit electromagnetic waves of light, displaying information sent over the internet via wireless electromagnetic links and optical fibers. Microwaves heat the food we brought for lunch, and in the evening when we get home, light-emitting diodes light up our homes. Even as we read a book before going to bed, our glasses bend electromagnetic waves to correct our vision. Electromagnetism is, quite literally, everything we see.

As our ability to model electromagnetic phenomena has improved, so has the complexity of the devices we are able to design. At first, design was guided by empirical observations and analytical models written with pen on paper. With the advent of computer simulations, we became able to explore geometries that would have been too complex to model analytically. Design was no longer restricted to what could be calculated by hand, but limited only by our imagination. Today, we have begun to go beyond even the limits of imagination through *inverse design*, where algorithms automatically search the space of possible designs until they find one that meets our specified criteria. The limitation is now the computational resources we have available to run these algorithms, and the efficiency with which we use them.

Machine learning can make inverse design more efficient by speeding up electromagnetic simulations. Traditional inverse design algorithms wander the design landscape in the dark, able to inspect their surroundings only by running expensive numerical simulations. In this context, machine learning can provide a map in the form of a *surrogate model*: an approximation of the simulations that lets the algorithm evaluate candidate designs much faster, allowing it to explore the design space more efficiently. However, training a surrogate model requires surveying the design landscape to generate training data, a process that is itself costly. Naive machine learning algorithms

require mapping out every corner of the design space, which is so expensive that it may negate their purpose.

Physics-informed machine learning can alleviate the training data requirement of surrogate models by utilizing prior knowledge of the underlying physics. The idea is that by embedding known physical laws in the model, fewer aspects of the problem need to be learned from data alone. In other words, physics-informed machine learning is a way to transfer the hard-earned knowledge of scientists and engineers to inverse design algorithms, to make use of the best of both humans and machines. By developing new methods in physics-informed machine learning for electromagnetism, we can elevate the capabilities of inverse design, ultimately leading to better electromagnetic devices that improve our everyday lives.

1.1 Thesis overview

In this thesis, I develop new methods for physics-informed machine learning applied to passive linear electromagnetic devices. An electromagnetic device here refers to a distribution of material in space, interacting with its environment through a set of ports, as illustrated in Figure 1.1(a). The geometry of the device is parameterized by a set of design parameters $\mathbf{d}$, and the response of the device is encapsulated by its scattering parameters S_{ij} , relating the amplitudes of incoming and outgoing waves through $\mathbf{s}^- = S\mathbf{s}^+$. The focus of the thesis is surrogate models that predict scattering parameters from design parameters, replacing the complex electromagnetic interactions inside the device as shown in Figure 1.1(b).

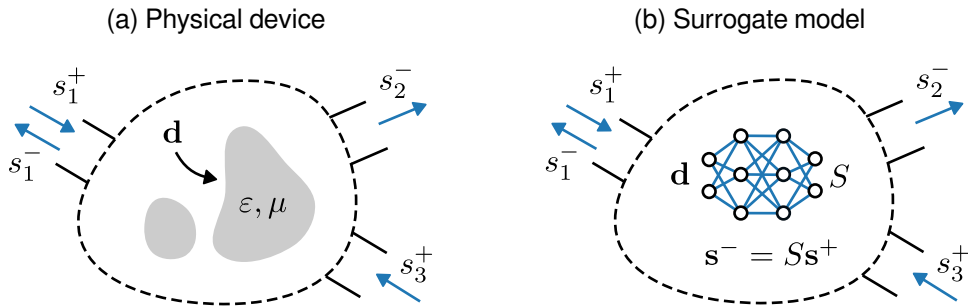


Figure 1.1: (a) The prototypical electromagnetic device studied in this thesis. The device consists of material distributed in space as defined by its design parameters $\mathbf{d}$ and interacts with the outside world through a set of ports. (b) A surrogate model replaces the physical device with a neural network trained to map the design parameters $\mathbf{d}$ to scattering parameters S , which encapsulate the relation between the incoming and outgoing port amplitudes.

A wide range of structures that are frequently studied in electromagnetic inverse design can be described this way. Figure 1.2 illustrates three examples that are relevant to the appended papers: photonic crystals, diffraction gratings, and metasurfaces.

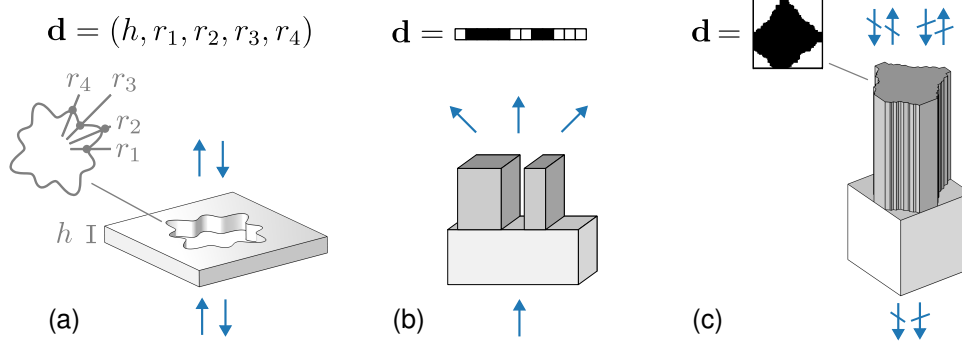


Figure 1.2: Examples of devices studied in the appended papers. (a) Photonic-crystal slab (**Paper I**). (b) Diffraction grating (**Paper II**). (c) Freeform metasurface (**Paper I**, [1]). All three geometries are periodic structures, of which only a single unit cell is illustrated.

Outline

In essence, the goal of the thesis is to identify structure in the physics in Figure 1.1(a) that can be built into the surrogate model in Figure 1.1(b) to improve its performance. The work is driven by two central research questions:

- How can prior knowledge of the physics be integrated in data-driven surrogate models for electromagnetic modeling and inverse design?
- To what extent does integrating prior knowledge influence the computational cost, interpretability, accuracy, and generalizability of the models?

The thesis is divided into six main chapters following this introduction. The next two chapters serve as a background. Chapter 2 introduces neural networks in the context of electromagnetic inverse design, while Chapter 3 summarizes the relevant theory of electromagnetism. The two chapters after that present complementary approaches to incorporate the physics of electromagnetism into neural networks. Chapter 4 is related to **Paper I** and provides a method to exploit the resonant structure of the scattering spectrum of electromagnetic devices. Chapter 5 is related to **Paper II** and develops a method for embedding symmetries of Maxwell's equations into the network architecture. Chapter 6 summarizes the appended papers. Finally, Chapter 7 concludes the thesis and proposes directions for future work.

Neural network surrogate models

Before delving into the physics of electromagnetism, we review the fundamental concepts of data-driven surrogate models for inverse design. This will serve as a motivation for the rest of the thesis and help us understand what aspects of surrogate models are most important to improve. At this stage, we do not need to know about the physics; the electromagnetic device can be considered a black box mapping design parameters $\mathbf{d}$ to scattering parameters S , that we want to approximate with a neural network.

The idea of using a neural network to model electromagnetic responses and speed up inverse design is not new. The concept was explored in the microwave and RF domain already in 1990 [2–4]. A recent wave of interest centered on nanophotonics began around 2017 [5–7], supported by improved computational hardware and developments in generative deep learning. The field has grown rapidly since then [8], and surrogate models have now been applied to a wide range of devices [9–13].

In this chapter, we start from the basics. Section 2.1 provides a background to inverse design and the problem that surrogate models solve. Section 2.2 introduces neural networks, the most popular type of data-driven surrogate model. Section 2.3 provides a critical assessment of when data-driven models provide benefits over numerical simulations, motivating the development of physics-informed models. Lastly, Section 2.4 gives a general introduction to physics-informed machine learning, providing context to the new methods introduced later in the thesis.

2.1 Motivation: Inverse design

The goal of inverse design is, given a set of desired properties $\mathbf{p}^*$, to find a device design $\mathbf{d}^*$ that exhibits those properties. In our case the properties $\mathbf{p}$ are usually scattering parameters or quantities that can be derived from them. The term *inverse design* refers

to the notion that finding $\mathbf{d}^*$ conceptually corresponds to evaluating the inverse f^{-1} of a simulation $f : \mathbf{d} \mapsto \mathbf{p}$ which calculates the properties of a given design.

While the forward problem of evaluating $\mathbf{p} = f(\mathbf{d})$ always has a unique solution, the inverse problem is fundamentally ill-posed, since there may be either zero or multiple designs for any given properties. A common workaround is to reformulate the inverse design as an optimization problem [14–17]. First, an objective function $F(\mathbf{p}, \mathbf{p}^*)$ quantifying how well the properties $\mathbf{p}$ match the desired properties $\mathbf{p}^*$ is defined. Then, the problem is restated to finding a design that maximizes the objective

$$\mathbf{d}^* = \operatorname{argmax}_{\mathbf{d}} F(f(\mathbf{d}), \mathbf{p}^*).$$

This way, if it is impossible to find a perfect solution, $\mathbf{d}^*$ will at least be “as good as possible” in the sense that the objective is maximized. Finding the global maximum of the objective is still computationally hard, since it generally requires searching the full space of designs with a costly forward-evaluation at every step. A gradient-based optimization method is often used in practice, resulting in a locally optimal design.

In this context, a surrogate model is a computationally cheap substitute $\hat{f}$ for the more expensive true forward model f [7, 16, 18]. The expensive model is in our case electromagnetic simulation, and the surrogate model is usually a neural network trained on a finite number of samples from f . Whereas the simulation f solves for the electromagnetic field everywhere in space, the surrogate $\hat{f}$ provides a shortcut to the answer, allowing for efficient optimization. The concept is illustrated in Figure 2.1.

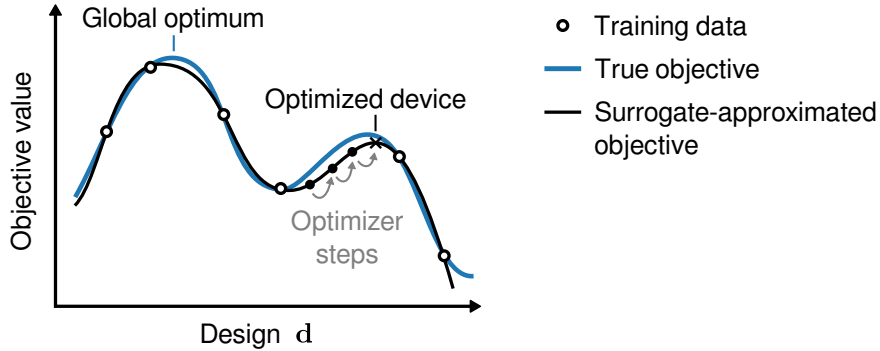


Figure 2.1: Conceptual illustration of surrogate-accelerated iterative inverse design. The final device is found by iterative optimization of a surrogate-approximated objective function. The true objective is only known for a few randomly chosen designs that make up the training data for the surrogate model. The optimized device differs from the global optimum due to surrogate model error and the optimization algorithm getting stuck in local optima.

It is important to note that a design found using the surrogate model will generally not be truly optimal since $\hat{f}$ is an approximation of f . However, a small error is often acceptable, for example if using the surrogate model allows for a more extensive search

that would not have been possible otherwise, or if there are other sources of error that are more significant, such as fabrication imperfections. Even when a small error is unacceptable, the design found by the surrogate model can at least be useful as a starting guess for further refinement using the true forward model.

2.2 Neural networks

A common type of surrogate model for electromagnetic scattering parameter prediction are neural networks [10, 18–25]. The study of neural networks with many layers is often referred to as deep learning [26].

In essence, a neural network is a parameterized function with a very large number of parameters. Inspired by biological neural networks, they consist of many connected artificial neurons, also referred to as nodes. Each neuron has a number of inputs x_i and produces an output y of the form

$$y = \sigma \left(b + \sum_i w_i x_i \right),$$

where the bias b and weights w_i are internal parameters of the neuron, and $\sigma(x)$ is a predefined non-linear activation function. Figure 2.2(a) illustrates one such neuron.

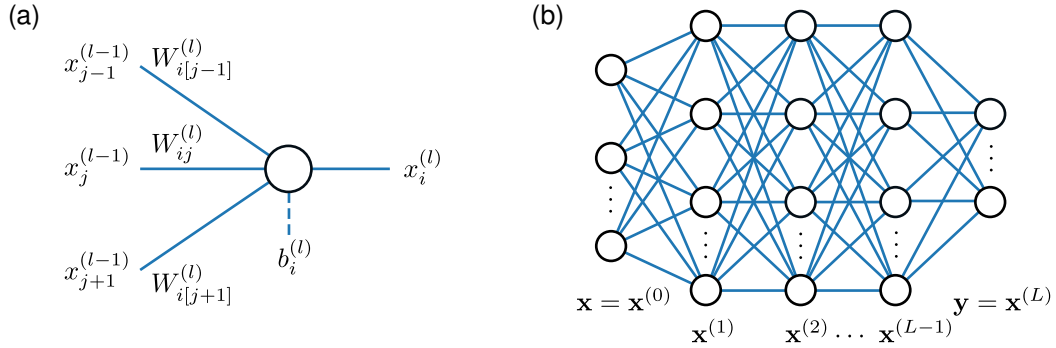


Figure 2.2: A fully-connected feed-forward neural network, also referred to as a multilayer perceptron. (a) Illustration of one neuron of the network, showing how the inputs to the left are combined using the weights and bias to create the output on the right. (b) Illustration of the architecture of the network, showing how the neurons are connected.

The way the neurons are connected is called the architecture of the neural network. A simple and common architecture is the multilayer perceptron [26, 27], which organizes the neurons into layers where the inputs to the neurons in one layer are the outputs of all neurons in the previous layer, as shown in Figure 2.2. Letting $\mathbf{x}^{(l)}$ denote a vector with all activations in layer $l \in \{1, 2, \dots, L\}$, this can be written as

$$\mathbf{x}^{(l)} = \sigma^{(l)}(\mathbf{b}^{(l)} + W^{(l)}\mathbf{x}^{(l-1)}),$$

where $\mathbf{b}^{(l)}$ is the bias vector of the layer, $W^{(l)}$ is the weight matrix, and $\sigma^{(l)}$ is an element-wise activation. The output of the final layer $\hat{\mathbf{y}} = \mathbf{x}^{(L)}$ is a complicated function of the inputs to the first layer $\mathbf{x} = \mathbf{x}^{(0)}$. We will denote this function by $\hat{\mathbf{y}} = \hat{f}_\theta(\mathbf{x})$, where θ is the set of all weights $W^{(l)}$ and biases $\mathbf{b}^{(l)}$.

The parameters θ are determined through training. In our case the purpose of training is to make the neural network $\hat{f}_\theta$ approximate a true function f . Suppose we have generated a dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_d}$ by sampling random inputs $\mathbf{x}_i$ and evaluating the corresponding outputs $\mathbf{y}_i = f(\mathbf{x}_i)$. The neural network would be a good estimate of f if the difference between $\hat{\mathbf{y}} = \hat{f}_\theta(\mathbf{x})$ and $\mathbf{y} = f(\mathbf{x})$ were small for all $\mathbf{x}$. We therefore define a loss function L that quantifies the average difference between $\hat{\mathbf{y}}$ and $\mathbf{y}$ across the dataset, and try to make L as small as possible by tuning the parameters θ . A common loss function is the mean-squared error

$$L(\theta) = \frac{1}{N_d} \sum_{i=1}^{N_d} \|\hat{f}_\theta(\mathbf{x}_i) - f(\mathbf{x}_i)\|^2,$$

but many alternatives exist. In practice, L is typically minimized through a gradient-based optimization algorithm [26, 28]. The gradient of L with respect to θ can be calculated for arbitrary network architectures using *automatic differentiation*, a differentiation algorithm relying on repeated applications of the chain rule [29].

Because the dataset can only contain a limited subset of the design space, it is always important to check how well the trained surrogate model generalizes to new designs. The dataset is therefore typically split into three parts: a training set, a validation set, and a test set. The training set is used to update the parameters of the network, while the validation set is used to judge its generalization to new data. If the validation performance of the network is not sufficient, it can be improved by tuning hyperparameters, i.e., parameters related to the training procedure or architecture of the network. The test set is reserved for a final check since the validation performance is no longer representative of new data after the hyperparameters have been tuned for it [26].

One of the major advantages of neural networks is that they are extremely flexible. This notion is formalized through so-called universal approximation theorems, which prove that neural network architectures in certain limits can approximate any continuous function to arbitrary precision. For example, multilayer perceptrons with non-polynomial activation functions are universal if either the width [30] or depth [31] is sufficiently large. This means neural networks "always work" as surrogate models, in the sense that they can in principle capture arbitrarily complicated relations between design parameters and scattering parameters as long as they are large enough.

2.3 When are data-driven models useful?

The purpose of a surrogate model is to make inverse design cheaper, but generating the surrogate model is in itself costly, since it requires training data. This raises the question: Is there really any benefit to using a surrogate model for inverse design? We will try to answer this question using a simple model for the computational cost.

Suppose that a single simulation has a computational cost C_{sim} (measured, for example, in core-hours), and that evaluating the gradient of the simulation costs another C_{sim} (for example using the adjoint method, [15, 32]). Designing N_{devices} devices using a gradient-based optimization procedure with N_{steps} steps using the true forward model would cost in total

$$C_{\text{direct}} = 2N_{\text{devices}}N_{\text{steps}}C_{\text{sim}}.$$

Using a surrogate model, the inverse design cost is due to a combination of generating the training data, training the model, and then running optimization with the trained model. If it takes N_{train} samples to train to acceptable accuracy, the cost of training is C_{train} , and the cost of a forward pass is $C_{\text{surrogate}}$, then the total cost will be

$$C_{\text{surrogate}} = N_{\text{train}}C_{\text{sim}} + C_{\text{train}} + 2N_{\text{devices}}N_{\text{steps}}C_{\text{surrogate}}.$$

In most cases $N_{\text{train}}C_{\text{sim}} \gg C_{\text{train}}$, so the training time can be neglected. The relation between the first and last term depend on N_{devices} , so we have to keep both.

Figure 2.3 shows the computational cost of direct and surrogate-accelerated inverse design normalized to C_{sim} as a function of N_{devices} . The plot assumes $N_{\text{steps}} = 10^2$, $N_{\text{train}} = 10^4$, $C_{\text{surrogate}} = 10^{-4}C_{\text{sim}}$, which are typical orders of magnitude for the systems studied in papers I and II.

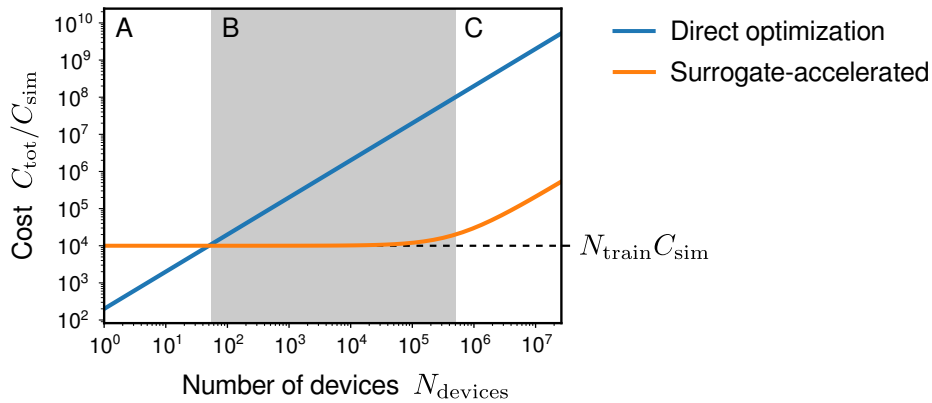


Figure 2.3: Computational cost versus number of designed devices for direct and surrogate-accelerated inverse design.

The plot reveals three regimes, indicated by A, B, and C. In regime A, where the number of designed devices is small, it is more efficient to run direct inverse design. At

$$N_{\text{devices}} = \frac{N_{\text{train}}}{2N_{\text{steps}}} \approx 50$$

there is a crossover to regime B, where it is more beneficial to use the surrogate model and the dominant cost is due to training data generation. The benefit of using the surrogate grows until

$$N_{\text{devices}} = \frac{N_{\text{train}}C_{\text{sim}}}{2N_{\text{steps}}C_{\text{surrogate}}} \approx 500\,000,$$

at which point we enter regime C, where the cost of accelerated inverse design is dominated by the optimization cost. In this regime, surrogate-accelerated optimization approaches a factor of $C_{\text{sim}}/C_{\text{surrogate}} = 10^4$ faster than direct optimization.

This analysis makes it clear that surrogate models are most useful when they are reused to design many devices. Good applications include designing devices consisting of many smaller subunits, such as metaatoms on a metasurface [1, 16], and statistical explorations of the design space [33], such as fabrication-robust inverse design or mapping out Pareto fronts. In applications where only a single optimization run is necessary, the cost of dataset generation dominates and direct optimization is likely to be the most computationally efficient choice. It should, however, be noted that dataset generation is often easily parallelized, whereas direct optimization typically requires simulations to be run sequentially. A surrogate-accelerated approach can therefore potentially reduce wall-clock time even when its computational cost is larger.

The analysis also tells us how to make surrogate models more useful. Firstly, the main limitation of surrogate models in regimes A and B is the amount of data needed to train them. We should therefore strive to make surrogate models that are as data-efficient as possible. Secondly, the benefit of using the surrogate model relies entirely on it being much faster than the simulations. We must therefore make sure that as we improve data-efficiency, the models remain fast. This is in contrast to other areas of machine learning, for example large language models, where training data is abundant and extreme inference costs can be justified by improved accuracy.

2.4 Physics-informed machine learning

One way to improve data-efficiency is to make use of prior knowledge about the physics in addition to the training data. Intuitively, utilizing prior knowledge means that not all information has to be learned from scratch, making a smaller amount of data sufficient to achieve acceptable accuracy. This is demonstrated in Figure 2.4, which shows model error as a function of the number of training samples for two example problems studied

in papers I and II. The improvement in data-efficiency can be quantified by comparing the amount of training data that is necessary to achieve a certain accuracy. Plotted this way, it is also clear that improving data-efficiency is equivalent to making a model more accurate; at a fixed number of training samples, a more data-efficient model has smaller error.

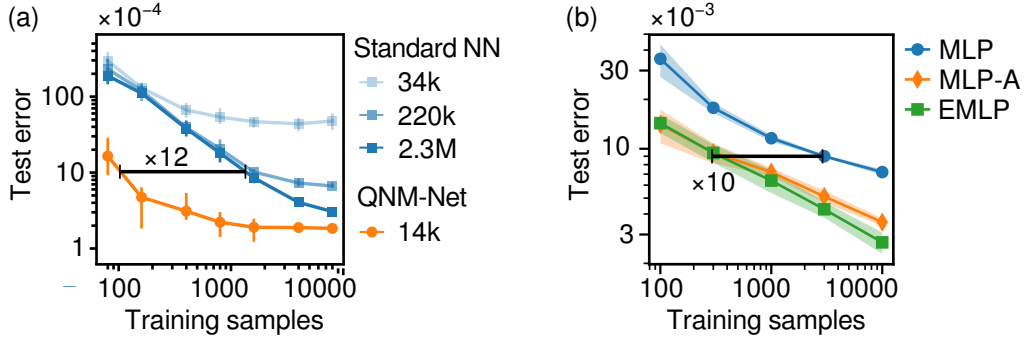


Figure 2.4: Figures reproduced from papers I and II showing increased data-efficiency of physics-informed surrogate models (orange and green) compared to standard surrogate models (blue) on benchmark problems. (a) From **Paper I**, demonstrating a twelve-fold improvement in data efficiency by utilizing knowledge about resonances. (b) From **Paper II**, demonstrating a ten-factor improvement in data efficiency by utilizing knowledge about symmetries.

Karniadakis *et al.* [34] classify methods in physics-informed machine learning into three main categories. *Observational biases* build knowledge into the training data, for example through data augmentation. *Learning biases* impose constraints softly by incorporating prior knowledge in the training process, for example by enforcing conservation laws through additional terms in the loss function. Finally, *inductive biases* build knowledge into the architecture of the neural network, reducing the space of learnable functions to those that satisfy underlying physical principles. The latter can be a particularly strong method when the architecture is constructed so that the physical principles are obeyed exactly. This is the primary type of bias considered in this thesis.

The downside of tailoring the neural network architecture to the physics of a specific problem is that it requires effort on the part of the model implementer. It could be argued that the time required to implement a physics-informed model would be better spent just generating more training data. In papers I and II, we address this problem by striving to make our methods *adaptable*. In other words, we propose general frameworks for constructing system-specific models. Using these frameworks, it is possible to tailor the inductive biases of the model to the properties of a specific device, without having to invent a new network architecture every time.

Theory of electromagnetism

Motivated by the discussion of the previous chapter, we now set out to understand the electromagnetic interactions inside the scattering device at a fundamental level, with the goal of identifying general patterns that can be exploited as inductive biases in physics-informed machine learning. Our starting point is Maxwell's equations, originally written down by James Clerk Maxwell in 1864, simplified to the form we know today by Rudolf Hertz and Oliver Heaviside [35], and used as the theoretical basis for all of classical electromagnetism to this day.

Section 3.1 derives the frequency-domain form of Maxwell's equations in linear and local materials, which is the relevant formulation for scattering problems with dielectric and metallic materials. Section 3.2 introduces a more compact notation of Maxwell's equations and some useful mathematical tools. Section 3.3 rigorously defines the scattering matrix, starting from the concept of guided modes. Finally, Section 3.4 provides a brief background on the numerical solvers of Maxwell's equations used here to generate training data.

3.1 Electromagnetic fields in linear local media

The fundamental quantities in electromagnetism are the electric field $\mathbf{E}$ and magnetic field $\mathbf{B}$. Their time evolution in vacuum is given by the *microscopic* Maxwell's equations

$$\begin{aligned}\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0}, & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, \\ \nabla \cdot \mathbf{B} &= 0, & \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t},\end{aligned}$$

where $\mathbf{J}$ and ρ are the current density and charge density, and ε_0 and μ_0 are the vacuum permittivity and permeability. The microscopic equations can in principle model fields

in a material by accounting for the motion of all the charged particles that make up the material [36]. However, this level of detail is often not necessary, making it more convenient to introduce an induced polarization field $\mathbf{P}$ and an induced magnetization field $\mathbf{M}$ with microscopic details averaged out. Defining the auxiliary fields $\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$ and $\mathbf{H} = \mathbf{B}/\mu_0 - \mathbf{M}$ leads to the *macroscopic* Maxwell's equations

$$\begin{aligned}\nabla \cdot \mathbf{D} &= \rho_f, & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, \\ \nabla \cdot \mathbf{B} &= 0, & \nabla \times \mathbf{H} &= \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t},\end{aligned}$$

where $\mathbf{J}_f$ and ρ_f now only contain free currents and charges (not due to polarization or magnetization). The macroscopic equations are undetermined until the *constitutive equations*, describing the material response, have been specified. The constitutive equations typically provide the induced polarization $\mathbf{P}$ and magnetization $\mathbf{M}$ as functions of $\mathbf{E}$ and $\mathbf{H}$.

Metals and dielectrics are usually well-approximated by a linear and local response. Linearity implies that driving the fields at a frequency ω results in a response at the same frequency ω . Hence, we can make the ansatz that all fields vary in time as $e^{i\omega t}$, and use complex amplitudes of the fields to represent relative phase. Linearity and locality together imply that $\mathbf{P}(\omega) = \varepsilon_0 \chi_e(\omega) \mathbf{E}(\omega)$ and $\mathbf{M}(\omega) = \chi_m(\omega) \mathbf{H}(\omega)$ where χ_e and χ_m are the electric and magnetic linear response function (susceptibility). Here we will neglect the frequency dependence for simplicity, which is valid when the material response is much faster than the frequency of the field, or if we are only interested in the response in a small frequency window where $\varepsilon(\omega)$ and $\mu(\omega)$ are approximately constant. Inserting these assumptions into the macroscopic Maxwell's equations yields the form of the equations that will be used throughout the thesis:

$$\begin{aligned}\nabla \cdot (\varepsilon \mathbf{E}) &= \rho_f, & \nabla \times \mathbf{E} &= -i\omega \mu \mathbf{H}, \\ \nabla \cdot (\mu \mathbf{H}) &= 0, & \nabla \times \mathbf{H} &= i\omega \varepsilon \mathbf{E} + \mathbf{J}_f,\end{aligned}$$

where $\varepsilon = \varepsilon_0(1 + \chi_e)$ and $\mu = \mu_0(1 + \chi_m)$ are functions of space. We will sometimes omit the divergence equations since they follow from the curl equations together with charge conservation $\nabla \cdot \mathbf{J}_f + i\omega \rho_f = 0$.

3.2 Fields as vectors

For convenience, we introduce a shorthand notation from the theory of quasinormal modes [37–39] where the six components of the electric and magnetic fields are collected into a single vector $\phi = (\mathbf{E}, \mathbf{H})$. With this notation, the source-free macroscopic equations

$$\begin{pmatrix} 0 & -i\varepsilon^{-1}\nabla \times \\ i\mu^{-1}\nabla \times & 0 \end{pmatrix} \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix} = \omega \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix}$$

can be expressed neatly as

$$H\phi = \omega\phi.$$

If we define a volume V enclosing the electromagnetic device (dashed lines in Figure 1.1), the set of all possible fields inside V forms a vector space. It will be useful to equip this space with an product

$$\langle \phi_A, \phi_B \rangle = \frac{1}{4} \int_V (\varepsilon \mathbf{E}_A^* \cdot \mathbf{E}_B + \mu \mathbf{H}_A^* \cdot \mathbf{H}_B) dV. \quad (3.1)$$

We also define a surface product

$$[\phi_A, \phi_B] = \frac{1}{4} \oint_{\partial V} (\mathbf{E}_A^* \times \mathbf{H}_B + \mathbf{E}_B \times \mathbf{H}_A^*) \cdot d\mathbf{S} \quad (3.2)$$

on the space of possible electromagnetic fields on the boundary ∂V of V . Many other choices of products are possible, but this particular choice has a nice physical interpretation in terms of the energy of the field. The time averaged energy density of the electromagnetic field is $(\varepsilon |\mathbf{E}|^2 + \mu |\mathbf{H}|^2)/4$, so the time averaged total energy of ϕ in V can be written compactly as $\langle \phi, \phi \rangle$. Similarly, the time averaged Poynting flux is $\mathbf{S} = \text{Re}(\mathbf{E} \times \mathbf{H}^*)/2$, so the time averaged power radiated out of V is $[\phi, \phi]$ [36, 40]. The products are related by the identity

$$\langle \phi_A, H\phi_B \rangle - \langle H\phi_A, \phi_B \rangle = i[\phi_A, \phi_B],$$

which holds for any two fields ϕ_A and ϕ_B , assuming ε and μ are real. This follows directly from the vector identity $\mathbf{A} \cdot (\nabla \times \mathbf{B}) = \mathbf{B} \cdot (\nabla \times \mathbf{A}) - \nabla \cdot (\mathbf{A} \times \mathbf{B})$ and the divergence theorem. This identity will be important in Chapter 4.

3.3 Guided modes and the scattering matrix

To properly understand the scattering matrix description of an electromagnetic device introduced in Chapter 1, we must first define the concept of a port. Assume that the device is connected to its environment through waveguides, i.e., regions of space where the material distribution is constant along one direction. Introduce a local coordinate system where this direction is aligned with the z -axis, where $z = 0$ on ∂V and z increases in the direction pointing away from the device. In the waveguide regions, solutions to Maxwell's equation can take the form

$$\mathbf{E}_m(x, y, z) = \mathbf{e}_m(x, y)e^{-i\beta_m z}, \quad (3.3)$$

$$\mathbf{H}_m(x, y, z) = \mathbf{h}_m(x, y)e^{-i\beta_m z}, \quad (3.4)$$

where m is an index enumerating the modes [40, 41]. For simplicity, we will only consider modes with real β (propagating modes). For lossless isotropic materials ($\varepsilon = \varepsilon^*$

and $\mu = \mu^*$), it can be easily verified that if $(\mathbf{E}, \mathbf{H})$ is a valid solution to Maxwell's equations, so is its time reversal $(\mathbf{E}^*, -\mathbf{H}^*)$ [36]. This implies that modes come in pairs where each mode $(\mathbf{e}_m, \mathbf{h}_m, \beta_m)$ has a time-reversed partner $(\mathbf{e}_m^*, -\mathbf{h}_m^*, -\beta_m)$ propagating in the opposite direction. Incoming modes correspond to modes propagating toward the device (superscript $+$, negative β), and outgoing modes correspond to modes propagating away from the device (superscript $-$, positive β).

A useful property of port modes with real β is that they are orthogonal with respect to the surface inner product defined in Equation (3.2). To prove this, start from the identity

$$\nabla \cdot (\mathbf{E}_A^* \times \mathbf{H}_B + \mathbf{E}_B \times \mathbf{H}_A^*) = 0$$

which can be shown to hold for any two fields ϕ_A and ϕ_B that solve the source-free Maxwell's equations for the same lossless material distribution at the same real frequency. Inserting two modes m and n in yields

$$-i(\beta_m - \beta_n)\hat{z} \cdot (\mathbf{E}_m^* \times \mathbf{H}_n + \mathbf{E}_n \times \mathbf{H}_m^*) = \nabla_t \cdot (\mathbf{E}_m^* \times \mathbf{H}_n + \mathbf{E}_n \times \mathbf{H}_m^*)$$

where $\nabla = \nabla_t + \hat{z}\partial_z$. Integrating over a surface A in the $z = 0$ plane and applying the divergence theorem we get

$$\begin{aligned} -i(\beta_m - \beta_n) \int_A (\mathbf{E}_m^* \times \mathbf{H}_n + \mathbf{E}_n \times \mathbf{H}_m^*) \cdot \hat{z} dA &= \int_A \nabla_t \cdot (\mathbf{E}_m^* \times \mathbf{H}_n + \mathbf{E}_n \times \mathbf{H}_m^*) dA \\ &= \oint_{\partial A} (\mathbf{E}_m^* \times \mathbf{H}_n + \mathbf{E}_n \times \mathbf{H}_m^*) \cdot \hat{n} dl, \end{aligned}$$

where $\hat{n}$ is a unit vector normal to the curve ∂A and pointing outward. If the modes are sufficiently confined to the waveguides, the contribution from the boundary ∂A is negligible, and the right hand side of this equation is zero. Using the shorthand notation for the surface integral, the resulting equation reads

$$(\beta_m - \beta_n)[\phi_m, \phi_n] = 0.$$

This can only be satisfied if $[\phi_m, \phi_n] = 0$ if $\beta_m \neq \beta_n$, proving that non-degenerate modes are orthogonal. For $m = n$ the inner product corresponds to the power carried by the mode, which is negative for incoming modes and positive for outgoing modes. Thus, if the modes are normalized to carry unit power, we have $[\phi_m^\pm, \phi_n^\pm] = \mp \delta_{mn}$.

To define the scattering matrix, assume that the total electromagnetic field ϕ on the boundary ∂V can be expanded as

$$\phi = \sum_{i=1}^{N^-} s_i^- \phi_i^- - \sum_{j=1}^{N^+} s_j^+ \phi_j^+,$$

where ϕ_i^- and ϕ_j^+ denote a finite set of N^- outgoing modes and N^+ incoming modes. Note that if we only want propagating modes to be included, the boundary must be

placed sufficiently far away that all evanescent modes have decayed to negligible amplitude. The minus sign is included so that $s_i^- = [\phi_i^-, \phi]$ and $s_i^+ = [\phi_i^+, \phi]$ for power-normalized modes. As long as the problem is well posed such that every excitation has a single well-defined response, if the incoming amplitudes s_j^+ are fixed, there will be a unique solution for the outgoing amplitudes s_i^- that satisfies Maxwell's equations inside the device. This defines a map from the incoming amplitudes $\mathbf{s}^+ = (s_1^+, \dots, s_{N^+}^+)$ to the outgoing amplitudes $\mathbf{s}^- = (s_1^-, \dots, s_{N^-}^-)$. For devices made of linear materials this must be a linear map, so it can be expressed as a $N^- \times N^+$ matrix S such that

$$\mathbf{s}^- = S\mathbf{s}^+.$$

We let this be the defining property of the scattering matrix S , and refer to the elements S_{ij} of S as the scattering parameters [41–43].

3.4 Numerical solvers

Maxwell's equations usually cannot be solved analytically, forcing us to resort to numerical simulations. A *full-wave solver* is a general simulation tool that solves the complete Maxwell's equations without any simplifying assumptions except numerical discretization. In the context of surrogate modeling, full-wave solvers are useful to generate datasets and to validate the performance of designed devices. Although the methods developed in this thesis are agnostic to how the full-wave solver is implemented, a basic understanding is useful to interpret the results and as a point of comparison.

The full-wave solver used in papers I and II is based on the finite-element method, as implemented in the commercial simulation software COMSOL Multiphysics. In short, the finite-element method is a way of converting partial differential equations (PDEs) into finite-dimensional problems. It is derived through the following procedure [44]:

- Convert the PDE to the *weak form* by demanding it to be satisfied only as the overlap integral with a set of test functions, rather than pointwise.
- Include boundary conditions in the weak form by applying integration by parts to the overlap integral.
- Make the problem finite-dimensional by expanding the field as a finite number of basis functions. In the Galerkin method, the basis functions are equal to the test functions.

The basis functions are constructed by subdividing the geometry into a mesh of many smaller regions (the "finite elements" of the finite-element method) where the field is represented by a small number of predefined shape functions. The resulting problem is a sparse system of equations with a number of degrees of freedom roughly proportional

to the number of mesh elements [44]. Making the mesh finer improves the accuracy of the solution, but increases the computational cost. Figure 3.1 shows the steps of simulating a device using the finite-element method.

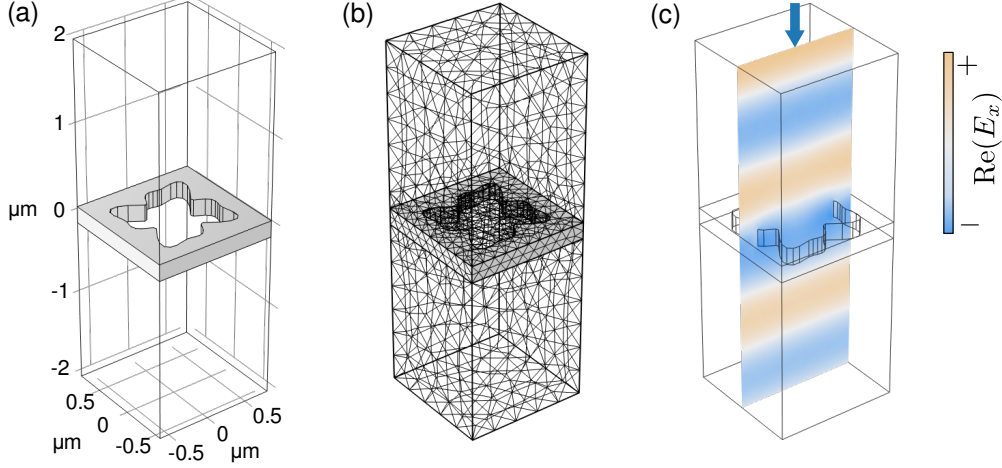


Figure 3.1: Illustration of a full-wave simulation of a photonic crystal slab from **Paper I**. (a) Geometry of the device. Port boundary conditions are used on the top and bottom, and periodic boundary conditions are used on the remaining four sides. (b) Meshed geometry. (c) Simulated electric field when the device is excited by a port on the top surface, indicated by an arrow.

It is important to distinguish between frequency-domain simulations and eigenmode simulations. In frequency-domain simulations, the structure is excited by incoming ports and the frequency ω is a fixed value. This provides a direct way to calculate the scattering parameters of a device. In eigenmode simulations, the boundary conditions have no incoming field and the frequency ω is solved for. Maxwell's equations then take the form of an eigenvalue problem with no driving and ω as the eigenvalue. Eigenmode simulations are a way to calculate the resonances of the system, as we will see in the next chapter.

Resonances as an inductive bias

Having established the basic theoretical framework, this chapter describes the first of two methods to embed physics knowledge into surrogate models. This method encodes structure in the frequency dependence of the scattering parameters $S(\omega)$, also known as the scattering spectrum. Every physicist will be familiar with the fact that spectra often contain peaks, and that these peaks are related to resonances. The core idea of this chapter is that it is easier to predict the resonances first, and then calculate the spectrum from the resonances, than it is to predict the scattering spectrum directly.

Section 4.1 introduces the general idea and reviews prior work on the topic. Section 4.2 formalizes the concept of resonances as *quasinormal modes* of the electromagnetic device. Section 4.3 provides the derivation of the quasinormal mode expansion of the scattering matrix, connecting resonances to scattering response. Finally, Section 4.4 describes the QNM-Net, a neural network architecture introduced in **Paper I** that uses the quasinormal mode expansion as a built-in physics model.

4.1 Resonance-based physics layers

The general concept of using resonances as an intermediate representation when predicting scattering spectra is illustrated in Figure 4.1. The dominant and most straightforward approach to predicting scattering spectra is to let a neural network output $S(\omega_i)$ at a predefined range of frequencies ω_i [1, 7, 9–11, 16, 20, 45–61]. Resonance-based models instead let the network predict the parameters of an algebraic expression for $S(\omega)$ (a “physics layer”), which is then evaluated to yield the scattering response $S(\omega)$. Note that the inputs and outputs of the model as a whole are still designs and scattering parameters. This means that the model can be trained on scattering parameters as usual; the physics parameters do not have to be explicit in the training data.

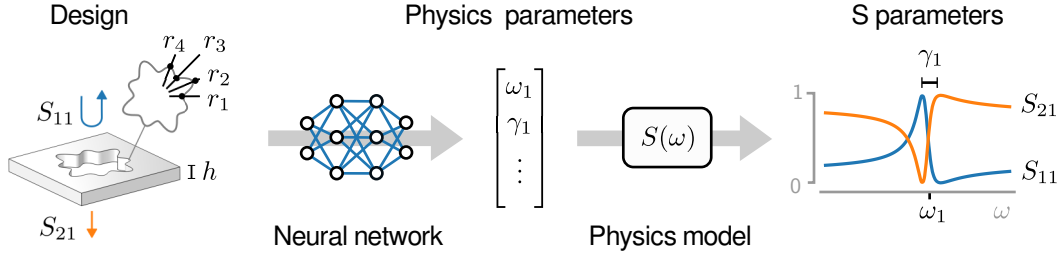


Figure 4.1: Conceptual illustration of a resonance-based physics model, predicting the parameters of resonances instead of scattering parameters, then reconstructing the scattering parameters using an analytical physics model.

Embedding a physics model in the network has several advantages. Firstly, if the physics parameters vary more smoothly with the design than the scattering parameters, they are likely easier to interpolate, and thus likely require less data to learn. Secondly, the surrogate model will automatically satisfy any constraints imposed on the physics model, making predicted scattering spectra consistent with the laws of physics. Lastly, the physics parameters automatically become accessible as a latent representation after training. This improves the interpretability of the network [11] and enables inverse design with the physics parameters as the target properties [62], as we will show in Section 4.4.

A key step in implementing surrogate models operating according to this principle is the choice of physics model. Several choices have been explored in earlier work. Blanchard-Dionne expressed the transmission amplitude of a metagrating as a heuristic sum of Lorentzians [48]. Khatib proposed to use an effective slab model with resonant effective parameters to describe the response of a dielectric metasurface [52]. Xu developed a custom physics model based on a combination of the transfer matrix method and temporal coupled mode theory [62] for tunable metasurfaces. A similar concept utilizing a circuit-based model for metallic metaatoms where resonances arise from LC-circuit oscillations was explored by Yan [45].

Although these previous approaches were successful for modeling their respective devices, they all rely on heuristic or problem-specific physics models. This limits the reusability of the methods, since it is unclear when each physics model is applicable and when it would be overly restrictive. Furthermore, it is not clear how the resonances of the models correspond to the resonances of the device, or how to build additional knowledge about the resonances into the model. **Paper I** resolves these issues by using a physics model derived from first principles: *the quasinormal mode expansion of the scattering matrix*. The rest of this chapter is devoted to deriving this model and exploring the consequences of incorporating it into the neural network.

4.2 Normal and quasinormal modes

The first step in deriving the quasinormal mode expansion is to understand quasinormal modes (often abbreviated as QNMs) and how they correspond to resonances of the electromagnetic device. A good starting point is to consider normal modes, which are eigenmodes of a closed system. In Section 3.2, we saw that for any two fields ϕ_A and ϕ_B ,

$$\langle \phi_A, H\phi_B \rangle - \langle H\phi_A, \phi_B \rangle = i[\phi_A, \phi_B], \quad (4.1)$$

where H is the Maxwell differential operator for lossless materials. For a closed system with the boundary sufficiently far away, the boundary overlap integral on the right-hand side is zero. The resulting equation is the definition of H being a Hermitian (self-adjoint) operator. Therefore, eigenmodes of H for a closed system are orthogonal with real eigenvalues ω , corresponding to unitary time evolution $e^{i\omega t}$. These are the familiar eigenmodes of electromagnetic cavities, illustrated in Figure 4.2(a).

Quasinormal modes are an extension of normal modes to open systems [37, 63]. If the boundary contribution is not zero, H is not Hermitian, and thus its eigenmodes are not orthogonal and can have complex eigenvalues $\tilde{\omega} = \omega + i\gamma$. In the time domain, a complex eigenfrequency corresponds to a decay in time $e^{i\tilde{\omega}t} = e^{i\omega t}e^{-\gamma t}$. Inserting a solution to Maxwell's equations $H\phi = \tilde{\omega}\phi$ into the identity (4.1) yields

$$\gamma = \frac{[\phi, \phi]}{2\langle \phi, \phi \rangle} = \frac{1}{2} \frac{\text{Radiated power}}{\text{Stored energy}},$$

showing that the complex part of the eigenvalue corresponds to the rate at which energy is leaking out of the system. For lossless materials, leakage must be due to radiation through the boundary, as illustrated in Figure 4.2(b). The factor of two is due to the stored energy being proportional to the square of the field amplitude; if the amplitude decays as $e^{-\gamma t}$, the energy decays as $e^{-2\gamma t}$.

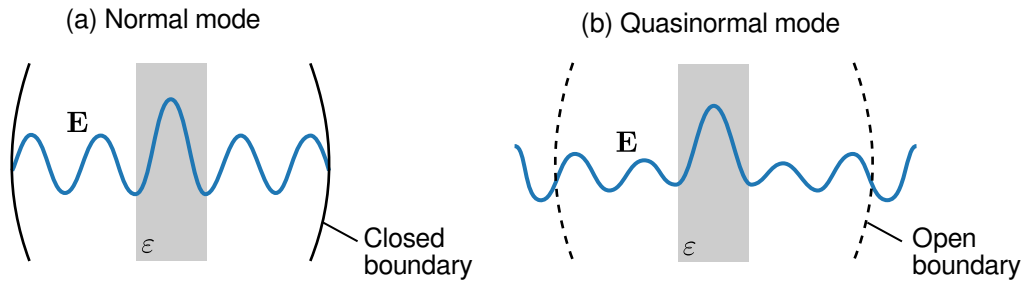


Figure 4.2: Comparison of normal and quasinormal modes of an electromagnetic resonator.

4.3 QNM expansion of the scattering matrix

We now seek an expression for the scattering spectrum $S(\omega)$ in terms of the quasinormal modes. Several derivations of the expansion have been proposed, relying on different assumptions and arriving at slightly different formulations [38, 39, 63–66]. Many start from a scattered field formulation of Maxwell’s equations and assume that the quasinormal modes form a complete basis, such that the scattered field can be written as a sum of (possibly infinitely many) quasinormal modes [38, 39, 63–65]. The sum is then truncated a posteriori and parameters are tuned through fitting to enforce physics constraints [64, 65]. Here we take a new approach, starting from the reduced set of modes and picking out their contribution to the total field through projection. The final result matches that of Reference [65], which was used in **Paper I**.

We begin by recalling that the total electromagnetic field ϕ solves Maxwell’s equations

$$H\phi = \omega\phi \quad (4.2)$$

at a fixed real frequency ω with boundary conditions such that on ∂V the field can be expanded as

$$\phi = \sum_{n=1}^{N^-} s_n^- \phi_n^- - \sum_{n=1}^{N^+} s_n^+ \phi_n^+$$

where ϕ_n^+ are incoming port modes and ϕ_n^- are outgoing port modes. The incoming amplitudes s_i^+ are fixed values defining the driving of the field, while the outgoing amplitudes s_n^- are free parameters of the field determined by solving Maxwell’s equations. From this total field, we want to pick out the contribution from M linearly independent quasinormal modes $\tilde{\phi}_m$ indexed by $m = 1, \dots, M$, each corresponding to an eigenmode of Maxwell’s equations

$$H\tilde{\phi}_m = \tilde{\omega}_m \tilde{\phi}_m \quad (4.3)$$

with no driving and outgoing radiation boundary conditions. Assume that the quasinormal mode field on the boundary lies in the span of the outgoing port modes, i.e.,

$$\tilde{\phi}_m = \sum_{n=1}^{N^-} D_{nm} \phi_n^-,$$

where D_{nm} is the amplitude of the m th quasinormal mode on the n th port. We approximate D_{nm} as constant even if it is in general frequency dependent, since $\tilde{\phi}_m$ is constant while the port modes ϕ_n^- may vary with frequency ω .

Subsequently, we separate the total field into two parts, one that lies in the span of the quasinormal modes and one that lies in the complement space, i.e.,

$$\phi = \phi_c + \sum_{m=1}^M a_m \tilde{\phi}_m, \quad (4.4)$$

with the complement field ϕ_c by definition satisfying $\langle \tilde{\phi}_m, \phi_c \rangle = 0$ for all m . Since no quasinormal modes have incoming amplitudes, ϕ_c must have the same incoming amplitudes as ϕ . We define the $N^- \times N^+$ background scattering matrix $C(\omega)$ as the linear map that relates the incoming amplitudes to the outgoing amplitudes of the complementary field such that

$$\phi_c = \sum_{n=1}^{N^-} \sum_{l=1}^{N^+} C_{nl}(\omega) s_l^+ \phi_n^- - \sum_{n=1}^{N^+} s_n^+ \phi_n^+$$

on ∂V . In other words, $C(\omega)$ is the scattering matrix after the quasinormal modes have been projected out of the response.

Next, we make use of the surface inner product to pick out the port amplitudes of the fields. If the port modes are normalized such that $[\phi_i^\pm, \phi_j^\pm] = \mp \delta_{ij}$ and $[\phi_i^\pm, \phi_j^\mp] = 0$, we have

$$[\phi_n^\pm, \phi] = s_n^\pm, \quad [\phi_n^-, \tilde{\phi}_m] = D_{nm}, \quad \text{and} \quad [\phi_n^-, \phi_c] = [C(\omega) \mathbf{s}^+]_n.$$

Applying $[\phi_l^-, \cdot]$ to both sides of Equation (4.4) shows that

$$\mathbf{s}^- = C(\omega) \mathbf{s}^+ + D \mathbf{a}, \quad (4.5)$$

where $\mathbf{s}^\pm = (s_1^\pm, \dots, s_{N^\pm}^\pm)^T$ and $\mathbf{a} = (a_1, \dots, a_M)^T$ are column vectors of amplitudes.

All that remains now is to find the quasinormal mode amplitudes $\mathbf{a}$ in terms of the driving amplitudes $\mathbf{s}^+$. Inserting the separated field (4.4) into Maxwell's equations (4.2), using the definition of the quasinormal modes (4.3), and rearranging leads to

$$\sum_{m=1}^M a_m (\omega - \tilde{\omega}_m) \tilde{\phi}_m = (H - \omega) \phi_c.$$

Applying $\langle \tilde{\phi}_l, \cdot \rangle$ to both sides and making use of Equation (4.1) on the right-hand side then yields

$$\sum_{m=1}^M \langle \tilde{\phi}_l, \tilde{\phi}_m \rangle (\omega - \tilde{\omega}_m) a_m = (\tilde{\omega}_l^* - \omega) \langle \tilde{\phi}_l, \phi_c \rangle + i[\tilde{\phi}_l, \phi_c] = i[D^\dagger C(\omega) \mathbf{s}^+]_l, \quad (4.6)$$

where $\langle \tilde{\phi}_l, \phi_c \rangle = 0$ by definition of ϕ_c , and we used that

$$[\tilde{\phi}_l, \phi_c] = \sum_{n=1}^{N^-} D_{nl}^* [\phi_n^-, \phi_c] = \sum_{n=1}^{N^-} (D^\dagger)_{ln} [C(\omega) \mathbf{s}^+]_n = [D^\dagger C(\omega) \mathbf{s}^+]_l.$$

To simplify the left-hand side of Equation (4.6), note that inserting two quasinormal modes $\tilde{\phi}_l$ and $\tilde{\phi}_m$ in (4.1) results in

$$(\tilde{\omega}_m - \tilde{\omega}_l^*) \langle \tilde{\phi}_l, \tilde{\phi}_m \rangle = i[\tilde{\phi}_l, \tilde{\phi}_m] = i \mathbf{d}_l^\dagger \mathbf{d}_m,$$

where $\mathbf{d}_m$ is a column vector containing the N^- port amplitudes in the m th column of D , corresponding to the m th quasinormal mode. Defining the $M \times M$ overlap matrix Q with elements

$$Q_{lm} := \langle \tilde{\phi}_l, \tilde{\phi}_m \rangle = i \frac{\mathbf{d}_l^\dagger \mathbf{d}_m}{\tilde{\omega}_m - \tilde{\omega}_l^*},$$

as well as the diagonal frequency matrices $\Omega = \omega I$ and $\tilde{\Omega} = \text{diag}(\tilde{\omega}_1, \dots, \tilde{\omega}_M)$, Equation (4.6) can be written neatly as

$$Q(\Omega - \tilde{\Omega})\mathbf{a} = iD^\dagger C(\omega)\mathbf{s}^+.$$

Inserting this in Equation (4.5) leads to $\mathbf{s}^- = [C(\omega) + iD(\Omega - \tilde{\Omega})^{-1}Q^{-1}D^\dagger C(\omega)]\mathbf{s}^+$, from which we can finally identify the scattering matrix

$$S(\omega) = C(\omega) + iD(\Omega - \tilde{\Omega})^{-1}Q^{-1}D^\dagger C(\omega).$$

In summary, this provides an expression for the frequency-dependent scattering matrix $S(\omega)$ in terms of a background $C(\omega)$ and a resonant term from the quasinormal modes, where each quasinormal mode is characterized by its resonance frequency $\tilde{\omega}_m$ and its port amplitudes $\mathbf{d}_m$. An illustrative example is provided in Figure 4.3, showing how the quasinormal mode expansion can explain the Fano resonance of a photonic crystal slab.

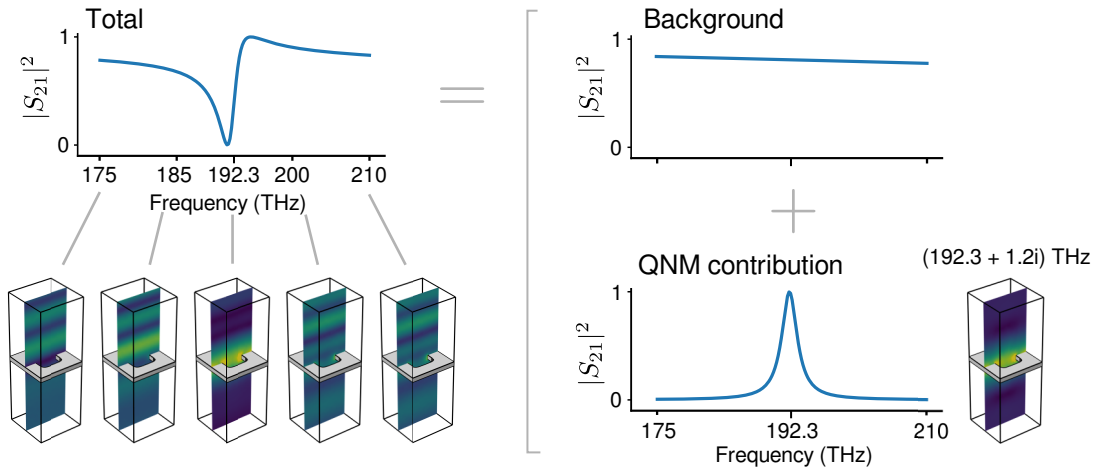


Figure 4.3: Demonstration of the quasinormal mode expansion. The scattering response of a photonic crystal slab (top left) can be written as a coherent sum of a slowly varying background (top right) and a sharp resonant contribution (bottom right). Plotting the field intensity while sweeping the excitation frequency (bottom left) shows that the field is concentrated in the slab at the resonance frequency 192.3 THz. An eigenmode simulation (bottom far right) confirms the existence of a quasinormal mode with matching field distribution at this frequency. The colormap at resonance is rescaled for visibility.

The background matrix $C(\omega)$ accounts for non-resonant scattering, i.e., what remains after the selected quasinormal modes have been projected out. We can expect it to vary smoothly if all sharp spectral features have been accounted for. Loss due to material absorption can be included perturbatively [64, 65], and the phase shift due to the propagation distance between the device and the ports is sometimes accounted for explicitly [65].

4.4 The QNM-Net

In **Paper I**, we propose a physics-informed neural network architecture of the type illustrated in Figure 4.1 with the quasinormal mode expansion as the built-in physics model. We call this architecture the QNM-Net. The QNM-Net is made up of several submodels connected according to the schematic in Figure 4.4. The internal structure of the submodels is intentionally left unspecified to improve adaptability, as described in Section 2.4. By choosing submodels on a system-by-system basis, it is possible to encode problem-specific prior knowledge into the model. This knowledge can be, for example, energy conservation for lossless materials (zero non-radiative loss in the mode models), spatial symmetries of the device (enforcing symmetries in the port amplitudes and background model), or efficient schemes to encode the design (through the choice of feature extractor).

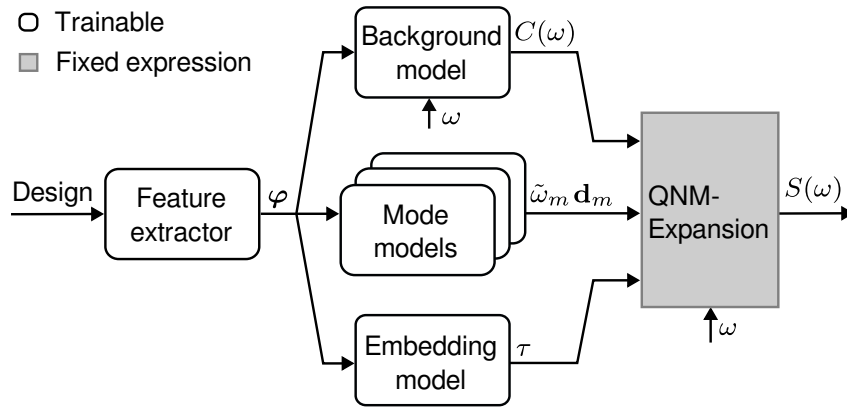


Figure 4.4: Overview of the QNM-Net architecture, adapted from **Paper I**. White boxes are submodels with trainable parameters. The gray box is an analytical physics model with no trainable parameters. Arrows indicate the relations between inputs and outputs of each submodel.

Besides the improved data efficiency shown in Section 2.4, the QNM-Net and other surrogate models using resonances [52] have the benefit that the physics parameters are accessible as an intermediate representation in the network, making predicted spectra explainable in terms of the underlying resonances. This is in contrast to standard neural

networks that directly map design parameters to scattering parameters, which are black boxes in the sense that there is no interpretable justification for the predicted spectra. It should be noted that the accuracy of the learned quasinormal modes must be verified by full-wave eigenmode simulations since there is no formal guarantee that training on spectra makes the internal mode representation converge to the correct physics. However, when the learned modes are correct, the mode representation can be useful to provide insights into the operational principle of a device inverse designed using the network, or to run inverse design with the objective formulated in terms of the modes. We demonstrate the latter in Figure 4.5, showing five photonic crystal slabs with linearly spaced quasinormal mode frequencies, inverse-designed using a trained QNM-Net.

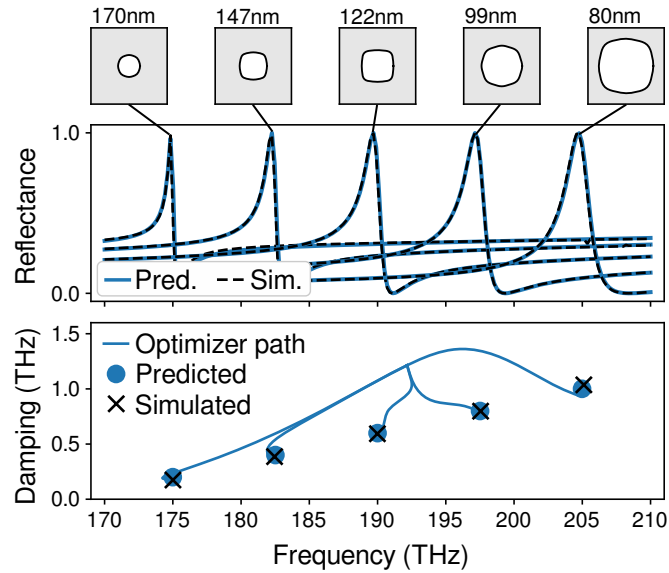


Figure 4.5: Demonstration of inverse design of photonic crystal slabs with learned quasinormal mode frequencies as design targets, adapted from **Paper I**. The geometries in the top row were optimized for linearly spaced complex eigenfrequencies, as shown in the lower panel. The blue dots and lines are QNM-Net predicted quasinormal mode frequencies, while the black crosses are from full-wave eigenmode simulations of the final designs. QNM-Net predictions and full-wave simulations of the photonic crystal reflectivity spectra are shown in the middle panel. Numbers above the geometries are thicknesses of the photonic-crystal slabs.

Symmetry as an inductive bias

In the previous chapter, we introduced a method to build knowledge about resonances into neural networks. In this chapter, we instead utilize knowledge about symmetries. The basic idea is that applying transformations to one design yields new designs with new scattering parameters that can be inferred without additional simulations. In the language of group theory, we say that the map from designs to scattering parameters is *equivariant* with respect to a group of transformations. Although it may be obvious to us that the new designs are all the same device, just viewed from different perspectives, this is not automatically known by a neural network. Consequently, it is knowledge that we can utilize to make the neural network better.

The field studying how to make neural networks respect symmetries is called equivariant deep learning. The canonical example of an equivariant neural network is the convolutional neural network, developed to exploit translation-equivariance in image analysis [67]. Equivariant architectures for other data geometries are studied in geometric deep learning [68, 69], and equivariant neural networks have been applied for a wide range of tasks, including protein structure prediction [70], molecular simulations [71], microscopy [72], biological imaging [73], and particle physics [74].

In this chapter, based on the theory developed in **Paper II**, we apply equivariant deep learning to surrogate models for scattering parameter prediction. Section 5.1 introduces a toy problem that will be used as an example throughout the chapter. Section 5.2 reviews the theoretical foundations of group theory. Section 5.3 describes how equivariance is applied as a constraint to neural networks. Section 5.4 describes how to implement equivariant neural networks in practice. Finally, Section 5.5 discusses the implications of using equivariant neural networks in the context of electromagnetic scattering.

5.1 Example problem

Consider a waveguide splitter consisting of a metal box with three ports and a dielectric pillar, as illustrated in Figure 5.1. The position of the pillar is parameterized by a single design parameter $\mathbf{d} = d$. There is a single input port, labeled 1, and two output ports, labeled 2 and 3. The scattering response is described by the scattering matrix

$$S = \begin{pmatrix} S_{21} \\ S_{31} \end{pmatrix}.$$

This problem has a reflection symmetry, in the sense that reflecting the geometry results in a new valid geometry. We will refer to the design and scattering parameters of the new geometry as $\hat{\mathbf{d}}$ and $\hat{S}$. Reflecting the design corresponds to changing the position of the pillar from d to $-d$, so we must have $\hat{\mathbf{d}} = -d$.

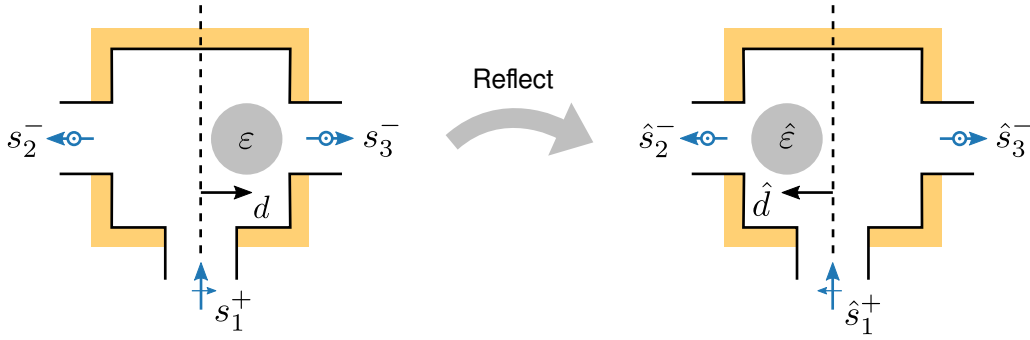


Figure 5.1: Geometry of the example problem.

Suppose we have simulated design d and know its scattering parameters S . Can we infer $\hat{S}$ without running a new simulation? The answer is yes, as long as we also keep track of what happens to the electromagnetic field when we apply the transformation. The phase of the incoming field is flipped, so the new incoming amplitude must be $\hat{s}_1^+ = -s_1^+$. The left and right outgoing fields are swapped, so the new outgoing amplitudes must be $\hat{s}_2^- = s_3^-$ and $\hat{s}_3^- = s_2^-$. The new scattering parameters by definition relate the new incoming and outgoing amplitudes, so they must be $\hat{S}_{21} = \hat{s}_2^- / \hat{s}_1^+ = -S_{31}$ and $\hat{S}_{31} = \hat{s}_3^- / \hat{s}_1^+ = -S_{21}$. In summary, the new scattering matrix is therefore

$$\hat{S} = \begin{pmatrix} -S_{31} \\ -S_{21} \end{pmatrix}.$$

For more complex devices and transformations it may not be as easy to infer the new scattering parameters. As an aid in identifying the transformation rules for new systems, **Paper II** provides a general step-by-step procedure to calculate the transformed

scattering parameters that works for any transformation of the material parameters and electromagnetic fields such that the transformed quantities still obey Maxwell's equations. This also includes more exotic transformations like curvilinear coordinate transformations, time reversal, and swapping the electric and magnetic quantities. The main restriction is that the transformed geometry must lie in the parameterized design space, and the transformed field must lie in the span of the original ports.

5.2 Introduction to group theory

To prepare for implementing equivariant neural networks, we first introduce some concepts from group theory, the mathematical framework used to study symmetries.

Groups

A *group* G is a set of group elements g together with a *group law* $G \times G \rightarrow G$ (a way to "multiply" group elements) such that

- $(g_1 g_2) g_3 = g_1 (g_2 g_3)$ for all $g_1, g_2, g_3 \in G$ (associativity);
- G contains an identity element, i.e., a $e \in G$ such that $eg = g$ for all $g \in G$;
- for any $g \in G$, G contains an inverse element of g , i.e., a $g^{-1} \in G$ such that $gg^{-1} = e$ and $g^{-1}g = e$.

In the example problem, there is only reflection symmetry, so the relevant group is $C_2 = \{e, m\}$, where the e is the identity element and m is associated with reflection. Reflecting twice does nothing, so the group law must be $m^2 = e$.

Actions

An intuitive way to think about group elements is that they are transformations that can *act* on various objects. This is formalized by the concept of *group actions*. For a group G and set X , a (left) group action is a map $\triangleright : G \times X \rightarrow X$ satisfying

- $e \triangleright x = x$ for all $x \in X$;
- $g_1 \triangleright (g_2 \triangleright x) = (g_1 g_2) \triangleright x$ for all $g_1, g_2 \in G$ and all $x \in X$.

In our example, the transformation rules for $\mathbf{d}$ and S can be expressed as actions of the group. The action of reflection on the design is

$$m \triangleright \mathbf{d} = \hat{\mathbf{d}} = -\mathbf{d},$$

and the action of reflection on the scattering parameters is

$$m \triangleright \begin{pmatrix} S_{21} \\ S_{31} \end{pmatrix} = \begin{pmatrix} \hat{S}_{21} \\ \hat{S}_{31} \end{pmatrix} = \begin{pmatrix} -S_{31} \\ -S_{21} \end{pmatrix}.$$

The action of the identity element is always to do nothing. Note that we overload the notation $\triangleright$ to mean different things depending on which object it is applied to.

Equivariance

We are now ready to formally define equivariance. Given two sets X and Y with corresponding actions of a group G , a function $f : X \rightarrow Y$ is said to be equivariant if $f(g \triangleright x) = g \triangleright f(x)$ for all $x \in X$ and all $g \in G$. In other words, a function is equivariant if it commutes with the action of the group. The map from design parameters to scattering parameters $F : \mathbf{d} \mapsto S$ is equivariant, because $F(g \triangleright \mathbf{d}) = g \triangleright S = g \triangleright F(\mathbf{d})$, i.e., the scattering parameters of the transformed design are the transformed scattering parameters. Figure 5.2 illustrates this relation in the form of a commutation diagram.

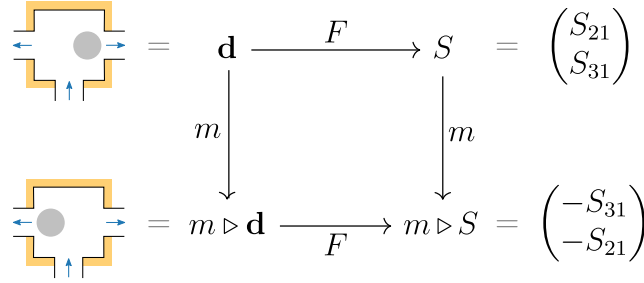


Figure 5.2: Commutation diagram for the example problem.

Representations

The final ingredient we need to implement equivariant neural networks is a way to represent group actions numerically. The tool we use for this is *representations*. Given a group G and a vector space V , a representation is a map that assigns a linear transformation of V to every element in G such that $\rho(g_1 g_2) = \rho(g_1) \rho(g_2)$ for all $g_1, g_2 \in G$. If the elements of V are column vectors, representations are a way to assign square matrices to every group element.

In the example, the actions on both $\mathbf{d}$ and S are linear, so they can be expressed using representations. The action on the design parameters can be written as $m \triangleright \mathbf{d} = \rho_{\mathbf{d}}(m) \mathbf{d}$ with the 1×1 representation

$$\rho_{\mathbf{d}}(m) = (-1).$$

If we flatten the scattering matrix S into a column vector $\mathbf{S}$ (in this case it was already a column, so $S = \mathbf{S}$), the action on the scattering parameters can be written as $m \triangleright \mathbf{S} = \rho_{\mathbf{S}}(m) \mathbf{S}$ with the 2×2 representation

$$\rho_{\mathbf{S}}(m) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Representations of the identity element $\rho_{\mathbf{d}}(e)$ and $\rho_{\mathbf{S}}(e)$ are identity matrices, so we don't have to bother writing them out.

5.3 Equivariant neural networks

The idea behind equivariant deep learning is to apply equivariance as a constraint to a neural network. As we saw in Section 2.2, feed-forward networks are functions $f : \mathbf{x}_0 \mapsto \mathbf{x}_L$ composed of several simpler functions $f_l : \mathbf{x}_{l-1} \mapsto \mathbf{x}_l$ (the “layers” of the network). To obey equivariance, the layers have to be carefully constructed and combined such that an action on the input leads to the corresponding action on the output. One way to ensure this is to assign a group action to every intermediate activation in the network and demand that each layer is equivariant independently. This can be depicted as a chained commutation diagram

$$\begin{array}{ccccccc}
 \mathbf{x}^{(0)} & \xrightarrow{f_1} & \mathbf{x}^{(1)} & \xrightarrow{f_2} & \dots & \xrightarrow{f_L} & \mathbf{x}^{(L)} \\
 \downarrow g & & \downarrow g & & & & \downarrow g \\
 \rho_0(g)\mathbf{x}^{(0)} & \xrightarrow{f_1} & \rho_1(g)\mathbf{x}^{(1)} & \xrightarrow{f_2} & \dots & \xrightarrow{f_L} & \rho_L(g)\mathbf{x}^{(L)}.
 \end{array}$$

The representations of the input ρ_0 and output ρ_L are defined by the task, but the representations of the hidden layers $\rho_1 \dots \rho_{L-1}$ are free to choose. Choosing unnecessarily large hidden representations leads to large activation tensors that make inference computationally expensive. On the other hand, choosing representations that are too restrictive for the desired function can over-constrain the network. A common choice is therefore to use copies of the *regular representation* for the hidden layers, which is known to be universal [75]. The regular representation exists for every group, and for finite groups of size n it maps group elements to $n \times n$ permutation matrices.

Suppose that we want to construct an equivariant neural network for the example problem with a single hidden layer of two neurons, as shown in Figure 5.3(a). For simplicity, we do not use any bias and treat the scattering parameters as if they are real numbers. The representation of the input must be ρ_d and the representation of the output must be ρ_s . A natural choice for the hidden layer would be the regular representation, which for $\{e, m\}$ is

$$\rho_{\text{reg}}(m) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The resulting commutation diagram for the network is shown in Figure 5.3(b). If we wanted four neurons in the hidden layer, we could have used a block-diagonal 4×4 representation consisting of two 2×2 blocks, where each block is the regular representation.

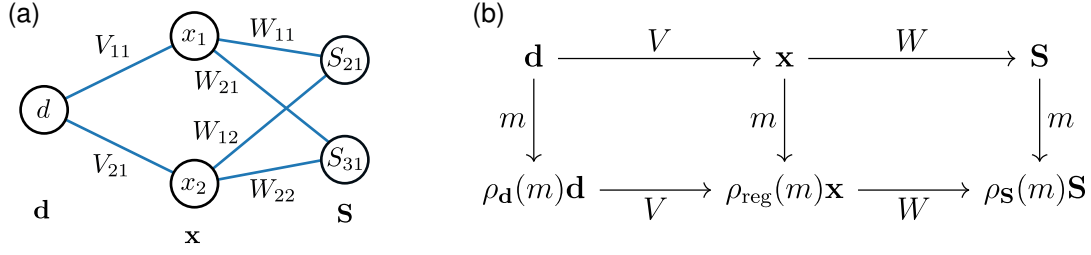


Figure 5.3: A simple neural network for the example problem. (a) Architecture of the network. (b) Desired commutation diagram for the activations.

5.4 Implementing equivariant layers

There are several software libraries available for implementing equivariant neural networks [76, 77]. These libraries implement equivariant counterparts of basic network components such as linear layers, nonlinearities, and convolutions, which can be combined to implement equivariant versions of most neural network architectures. Using these libraries, making a neural network equivariant is as simple as choosing the desired representation of each layer. Nevertheless, deriving the equivariance constraint by hand is instructive to gain intuition for how these libraries work. Here we will derive how to implement equivariant linear (fully connected) layers.

Consider a linear layer $\mathbf{x}_B = W\mathbf{x}_A$ that maps the input $\mathbf{x}_A$ to the output $\mathbf{x}_B$ through a weight matrix W (neglecting bias and non-linearity for simplicity). If $\mathbf{x}_A$ should transform with representation $\rho_A(g)$ and $\mathbf{x}_B$ with representation $\rho_B(g)$, equivariance demands that

$$W\rho_A(g)\mathbf{x}_A = \rho_B(g)\mathbf{x}_B = \rho_B(g)W\mathbf{x}_A,$$

for all $\mathbf{x}_A$ and all $g \in G$. This implies that the weight matrix must satisfy the constraint

$$W = \rho_B(g)W\rho_A(g^{-1}). \quad (5.1)$$

To build equivariant networks with linear layers, we must find a method to construct the W satisfying this constraint that works for any choice of ρ_A and ρ_B . One way is to use that any finite-dimensional unitary representation can be reduced to a block-diagonal form

$$\rho(g) = Qr(g)Q^{-1} = Q \begin{pmatrix} \psi_1(g) & & 0 \\ & \ddots & \\ 0 & & \psi_n(g) \end{pmatrix} Q^{-1},$$

where Q is a change of basis and r is a block-diagonal representation where the blocks ψ_i are irreducible representations [77]. Inserting this in Equation (5.1) and rearranging yields

$$[Q_B^{-1}WQ_A] = r_B(g)[Q_B^{-1}WQ_A]r_A(g^{-1}).$$

Writing the weight matrix as $W = Q_B W' Q_A^{-1}$ and considering the elements of W' as the free parameters of the layer, we find that W' must satisfy the constraint

$$W' = r_B(g) W' r_A(g^{-1}). \quad (5.2)$$

Importantly, since $r_A(g)$ and $r_B(g)$ are block-diagonal, this is now a block-wise constraint on W' , where the constraint on each block only involves two irreducible representations. Thus, it is enough to solve the constraint for any pair of irreducible representations, which is feasible to do on a case-by-case basis for each new group.

Let's apply this method to the example network in Figure 5.3(b). The weights W mapping $\mathbf{x}$ to $\mathbf{S}$ should satisfy

$$W = \rho_S(m) W \rho_{\text{reg}}(m^{-1}).$$

The input and output representations can be reduced to block-diagonal forms

$$\rho_{\text{reg}}(m) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}}_{Q_{\text{reg}}} \underbrace{\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}}_{r_{\text{reg}}(g)} \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}}_{Q_{\text{reg}}^{-1}}$$

and

$$\rho_S(m) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}}_{Q_S} \underbrace{\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}}_{r_S(m)} \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}}_{Q_S^{-1}}$$

In this case it happens that r_A and r_B are diagonal rather than block diagonal because the only irreducible representations of the group $\{e, m\}$ are (1) and (-1) . Now defining $W = Q_S W' Q_{\text{reg}}^{-1}$ and using Equation (5.2), we get

$$\begin{pmatrix} W'_{11} & W'_{12} \\ W'_{21} & W'_{22} \end{pmatrix} = \underbrace{\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}}_{r_S(m)} \begin{pmatrix} W'_{11} & W'_{12} \\ W'_{21} & W'_{22} \end{pmatrix} \underbrace{\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}}_{r_{\text{reg}}(m^{-1})} = \begin{pmatrix} W'_{11} & -W'_{12} \\ -W'_{21} & W'_{22} \end{pmatrix}$$

This constraint can be solved block-wise (in this case element-wise since the blocks are 1×1), yielding $W'_{12} = W'_{21} = 0$. The constrained weight matrix can be constructed from the remaining parameters as

$$\begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}}_{Q_S} \begin{pmatrix} W'_{11} & 0 \\ 0 & W'_{22} \end{pmatrix} \underbrace{\frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}}_{Q_{\text{reg}}^{-1}}.$$

A similar computation shows that the weight matrix for the first linear layer, mapping d to $\mathbf{x}$ can be parameterized as

$$\begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} V'_{11} \\ 0 \end{pmatrix}$$

Implementing the network in code, we could store the primed weights as free parameters, and compute the unprimed weights before each forward pass during training. After training, the unprimed weights only have to be calculated once.

5.5 Implications of equivariance

What do we gain by enforcing equivariance on surrogate models? In Section 2.4, we have already seen that it can improve data efficiency. One way to understand why is to consider that if an equivariant network is trained on one design, by construction it must be equally accurate for all transformed versions of that design. Therefore, enforcing equivariance should be at least as good as adding all transformed versions of every design to the dataset (data augmentation). Ideally, if transformed samples are just as good as new samples, data efficiency would increase by a factor equal to the size of the group. This is validated numerically in **Paper II**.

Another benefit is that equivariant neural networks make predictions that are consistent across the symmetry group. Figure 5.4 demonstrates this by studying the scattering response of a diffraction grating. The symmetry of the problem dictates that translating the grating rotates the complex phase of parameter S_{-1} , making it trace out a circle in the complex plane. This is true for exactly equivariant neural networks, but not for standard neural networks, even when trained with data augmentation.

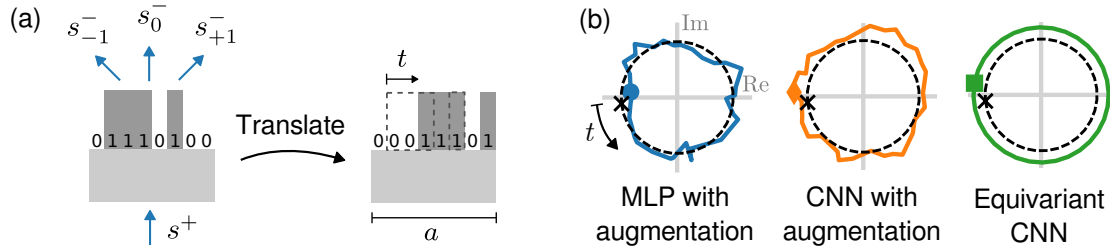


Figure 5.4: Demonstration of consistency across the symmetry group for a diffraction grating benchmark problem, adapted from **Paper II**. Translating the grating as shown in (a) corresponds to adding a complex phase to scattering parameter S_{-1} , making it trace out a circle in the complex plane. As shown in (b), this is respected for the full-wave solver (black dashed line) and the equivariant CNN, but not for standard network architectures (MLP and CNN).

Lastly, equivariant neural networks are more parameter efficient. We saw this in the previous section, where enforcing reflection equivariance on the example problem reduced the number of free parameters from six to three. In principle, having fewer independent parameters makes it possible to reduce the amount of memory required to store the network. Note, however, that the computational cost of inference is not automatically reduced if the equivariant layer is evaluated using the full unprimed weight matrix.

Summary of papers

Paper I

A General Framework for Knowledge Integration in Machine Learning for Electromagnetic Scattering Using Quasinormal Modes

Paper I demonstrates the quasinormal-mode-based neural network concept described in Chapter 4. The main contribution of the paper in relation to earlier work on resonance-based neural networks is that it introduces the quasinormal mode formalism as a rigorous and general framework, improving on the heuristic or problem-specific models used previously. Using scattering by photonic crystal slabs as an example, we demonstrate that the formal theoretical basis of the quasinormal mode expansion allows for strict problem-specific constraints to be imposed, leading to an order-of-magnitude improvement in data efficiency compared to standard neural networks. We also apply the framework to freeform metasurfaces with a large design space, low symmetry, several overlapping resonances, and several scattering ports, to demonstrate the generality of the method.

Paper II

Symmetry-Informed Deep Learning for Electromagnetic Scattering

Paper II demonstrates the symmetry-informed machine learning concept described in Chapter 5. We develop the necessary theory to implement data augmentation and equivariant neural networks for a wide class of transformations of Maxwell's equations, including curvilinear coordinate transformations, time reversal, and duality. We

then demonstrate the concept on two example systems: photonic crystals with discrete symmetry and diffraction gratings with continuous translation symmetry. For both systems, we find that utilizing equivariance leads to at least an order-of-magnitude improvement in data efficiency. For the diffraction gratings, the continuous translation symmetry implies that a convolutional neural network should be used. However, standard convolutional neural networks often break exact equivariance through pooling, zero padding, and fully connected readout layers. We therefore show how to construct exactly equivariant networks, and in the process derive a novel readout layer that maps feature fields to scattering parameters.

Conclusions and outlook

This thesis has demonstrated that utilizing prior knowledge about electromagnetic scattering is a fruitful way to improve data efficiency, interpretability, and consistency of neural network surrogate models. Starting from first principles, we formulated two complementary methods of incorporating the physics of electromagnetic scattering into neural networks. The first method was based on the quasinormal mode expansion of the scattering matrix and utilized the resonant structure underlying the scattering spectrum. The second method utilized the equivariance of Maxwell's equations to constrain the weights of the neural network, enforcing consistency with the symmetries of the problem. Demonstrating the methods on a range of problems from photonics, we found that both lead to significantly improved data efficiency, alleviating the main computational bottleneck in machine-learning-accelerated inverse design. Furthermore, we saw that physics-informed models enjoyed several additional benefits, such as improved interpretability and guarantees about consistency with physical laws.

Outlook

Many challenges and opportunities remain at the intersection between machine learning and electromagnetism. Here, I point out three important directions for further development.

The first is tighter integration between data-driven and direct numerical methods. In this thesis, the primary focus has been on integrating simple physical principles into fundamentally data-driven models. As mentioned in Section 2.1, a limitation of this approach is that surrogate models incur an approximation error, causing designed devices to deviate from the optimum of the objective function. It would therefore be desirable to have a mechanism for applying cheap numerical corrections to surrogate model predictions to ensure their accuracy. Solving some aspects of the problems numerically

and learning only the most time-consuming steps could also improve the generalization of surrogate models, further improving their data efficiency and reusability. Paths to achieving this may be to shift the focus from surrogate models predicting the final solution to predicting preconditioners, initial guesses, iteration steps, or reduced basis functions that are then used by an exact solver. Ideally, such hybrid approaches would provide a simulation speed comparable to neural networks while retaining the accuracy and dependability of full-wave simulations.

The second important direction is bringing surrogate models into the workflows of device designers. As discussed in Section 2.3, surrogate models are costly to develop but cheap to use. We should therefore strive to reuse surrogate models as much as possible. One way to facilitate reuse would be to integrate pretrained models into commonly used electromagnetic simulation software. Another possibility in the context of integrated photonics is to provide surrogate models in photonic design kits (PDKs), which would enable efficient circuit-level modeling and inverse design. This will require rigorous benchmarking and uncertainty quantification methods to ensure that surrogate model predictions can be trusted by designers.

The third and perhaps most exciting direction is to find novel ways to take advantage of the capabilities unlocked by fast simulations. Today, surrogate models are mostly used in the same way as full-wave solvers were used previously, simulating the same kinds of devices but faster. However, having access to a simulator that is several orders of magnitude more efficient enables things that we may not have even considered before, such as restarting optimization thousands of times, simulating several designs and excitations at each optimization step, or co-optimizing tens or hundreds of connected devices. This enables more exhaustive searches, optimization of full systems, and design for complex multifunctional objectives. Taking full advantage of the possibilities enabled by these methods has the potential to fundamentally transform electromagnetic modeling and inverse design.

Acknowledgments

First of all, I would like to thank my supervisor Philippe for his guidance and feedback throughout the work on this thesis. Second, I thank my examiner Andreas, my co-supervisor Witlef, and all members of our group and the quantum optomechanics team for valuable discussions and advice. I am also sincerely grateful for my friends and colleagues, both in the division and in Metapix, for creating a fun, inspiring, and collaborative work environment. Finally, I would like to thank my family for their support.

Bibliography

- [1] T. Gahlmann and P. Tassin, *Deep Neural Networks for the Prediction of the Optical Properties and the Free-Form Inverse Design of Metamaterials*, Phys. Rev. B **106**, 085408 (2022). doi: 10.1103/PhysRevB.106.085408.
- [2] J. Hwang, C. Chan, and II. Marks, R.J., *Frequency Selective Surface Design Based on Iterative Inversion of Neural Networks*, in 1990 IJCNN Int. Jt. Conf. Neural Netw., 39, 1990. doi: 10.1109/IJCNN.1990.137541.
- [3] D. Davis, C. Chan, and J. Hwang, *Frequency Selective Surface Design Using Neural Networks Inversion Based on Parametrized Representations*, in Antennas Propag. Soc. Symp. 1991 Dig., 200, 1991. doi:10.1109/APS.1991.174807.
- [4] G. Creech, B. Paul, C. Lesniak, T. Jenkins, R. Lee, and M. Calcaterra, *Artificial Neural Networks for Accurate Microwave CAD Applications*, in 1996 IEEE MTT- Int. Microw. Symp. Dig., vol. 2, 733, 1996. doi:10.1109/MWSYM.1996.511043.
- [5] I. Malkiel, A. Nagler, M. Mrejen, U. Arieli, L. Wolf, and H. Suchowski, *Deep Learning for Design and Retrieval of Nano-photonic Structures*, 2017, arXiv:1702.07949.
- [6] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, *Nanophotonic Particle Simulation and Inverse Design Using Artificial Neural Networks*, Sci. Adv. **4**, eaar4206 (2018). doi:10.1126/sciadv.aar4206.
- [7] C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, *Deep Learning for Accelerated All-Dielectric Metasurface Design*, Opt. Express **27**, 27523 (2019). doi:10.1364/OE.27.027523.
- [8] Q. Wang, Y. Zhang, A. Burguete-Lopez, S. Rodionov, and A. Fratalocchi, *The Transformational Dive of Nanophotonics Inverse Design from Deep Learning to Artificial General Intelligence*, APL Photonics **9**, 100902 (2024). doi:10.1063/5.0226592.
- [9] S. So, T. Badloe, J. Noh, J. Bravo-Abad, and J. Rho, *Deep Learning Enabled Inverse Design in Nanophotonics*, Nanophotonics **9**, 1041 (2020). doi:10.1515/nanoph-2019-0474.
- [10] P. R. Wiecha, A. Arbouet, C. Girard, and O. L. Muskens, *Deep Learning in Nano-Photonics: Inverse Design and Beyond*, Photonic Res. **9**, B182 (2021). doi:10.1364/PRJ.415960.
- [11] O. Khatib, S. Ren, J. Malof, and W. J. Padilla, *Deep Learning the Electromagnetic Properties of Metamaterials—A Comprehensive Review*, Adv. Funct. Mater. **31**, 2101748 (2021). doi: 10.1002/adfm.202101748.
- [12] D. Midtvedt, V. Mylnikov, A. Stilgoe, M. Käll, H. Rubinsztein-Dunlop, and G. Volpe, *Deep Learning in Light–Matter Interactions*, Nanophotonics **11**, 3189 (2022). doi:10.1515/nanoph-2022-0197.

- [13] Y. Dong, S. An, H. Jiang, B. Zheng, H. Tang, Y. Huang, H. Zhao, and H. Zhang, *Advanced Deep Learning Approaches in Metasurface Modeling and Design: A Review*, Progress in Quantum Electronics **99**, 100554 (2025). doi:10.1016/j.pquantelec.2025.100554.
- [14] J. S. Jensen and O. Sigmund, *Topology Optimization for Nano-Photonics*, Laser Photonics Rev. **5**, 308 (2011). doi:10.1002/lpor.201000014.
- [15] L. Su, D. Vercruysse, J. Skarda, N. V. Sapra, J. A. Petykiewicz, and J. Vučković, *Nanophotonic Inverse Design with SPINS: Software Architecture and Practical Considerations*, Appl. Phys. Rev. **7**, 011407 (2020). doi:10.1063/1.5131263.
- [16] Y. Deng, S. Ren, K. Fan, J. M. Malof, and W. J. Padilla, *Neural-Adjoint Method for the Inverse Design of All-Dielectric Metasurfaces*, Opt. Express **29**, 7526 (2021). doi:10.1364/OE.419138.
- [17] J. Park, S. Kim, D. W. Nam, H. Chung, C. Y. Park, and M. S. Jang, *Free-Form Optimization of Nanophotonic Devices: From Classical Methods to Deep Learning*, Nanophotonics **11**, 1809 (2022). doi:10.1515/nanoph-2021-0713.
- [18] A. Khairah-Walieh, D. Langevin, P. Bennet, O. Teytaud, A. Moreau, and P. R. Wiecha, *A Newcomer's Guide to Deep Learning for Inverse Design in Nano-Photonics*, Nanophotonics **12**, 4387 (2023). doi:10.1515/nanoph-2023-0527.
- [19] R. S. Hegde, *Deep Learning: A New Tool for Photonic Nanostructure Design*, Nanoscale Adv. **2**, 1007 (2020). doi:10.1039/C9NA00656G.
- [20] J. Jiang, M. Chen, and J. A. Fan, *Deep Neural Networks for the Evaluation and Design of Photonic Devices*, Nat. Rev. Mater. **6**, 679 (2021). doi:10.1038/s41578-020-00260-1.
- [21] Y. Deng, J. Dong, S. Ren, O. Khatib, M. Soltani, V. Tarokh, W. J. Padilla, and J. M. Malof, *Benchmarking Data-driven Surrogate Simulators for Artificial Electromagnetic Materials*, in *35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [22] Z. Liu, D. Zhu, L. Raju, and W. Cai, *Tackling Photonic Inverse Design with Machine Learning*, Adv. Sci. **8**, 2002923 (2021). doi:10.1002/advs.202002923.
- [23] M. K. Chen, X. Liu, Y. Sun, and D. P. Tsai, *Artificial Intelligence in Meta-optics*, Chem. Rev. **122**, 15356 (2022). doi:10.1021/acs.chemrev.2c00012.
- [24] V. Iyer, B. A. Wilson, Y. Chen, A. V. Kildishev, V. M. Shalaev, and A. Boltasseva, *Deep Learning in Photonic Device Development: Nuances and Opportunities*, npj Nanophoton. **3**, 5 (2026). doi:10.1038/s44310-025-00097-y.
- [25] R. Pestourie, *Fast Approximate Solvers: Metamaterials Design in Electromagnetism.*, IEEE Antennas Propag. Mag. **68**, 35 (2026). doi:10.1109/MAP.2025.3588058.
- [26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (Cambridge, Massachusetts: The MIT Press, 2016). ISBN 978-0-262-03561-3.
- [27] F. Rosenblatt, *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.*, Psychological Review **65**, 386 (1958). doi:10.1037/h0042519.
- [28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*, Nature **323**, 533 (1986). doi:10.1038/323533a0.
- [29] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, *Automatic Differentiation in Machine Learning: A Survey*, 2015, arXiv:1502.05767.

-
- [30] K.-I. Funahashi, *On the Approximate Realization of Continuous Mappings by Neural Networks*, Neural Networks 2, 183 (1989). doi:10.1016/0893-6080(89)90003-8.
 - [31] P. Kidger and T. Lyons, *Universal Approximation with Deep Narrow Networks*, 2020, arXiv:1905.08539.
 - [32] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, *Adjoint Shape Optimization Applied to Electromagnetic Design*, Opt. Express 21, 21693 (2013). doi:10.1364/OE.21.021693.
 - [33] R. P. Jenkins, S. D. Campbell, and D. H. Werner, *Establishing Exhaustive Metasurface Robustness against Fabrication Uncertainties through Deep Learning*, Nanophotonics 10, 4497 (2021). doi:10.1515/nanoph-2021-0428.
 - [34] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Physics-Informed Machine Learning*, Nat. Rev. Phys. 3, 422 (2021). doi:10.1038/s42254-021-00314-5.
 - [35] M. Mitolo and R. Araneo, *A Brief History of Maxwell's Equations [History]*, IEEE Ind. Appl. Mag. 25, 8 (2019). doi:10.1109/MIAS.2019.2898096.
 - [36] J. D. Jackson, *Classical Electrodynamics* (Hoboken: Wiley, 2021). ISBN 978-1-119-77077-0.
 - [37] P. T. Kristensen, K. Herrmann, F. Intravaia, and K. Busch, *Modeling Electromagnetic Resonators Using Quasinormal Modes*, Adv. Opt. Photonics 12, 612 (2020). doi:10.1364/AOP.377940.
 - [38] T. Weiss and E. A. Muljarov, *How to Calculate the Pole Expansion of the Optical Scattering Matrix from the Resonant States*, Phys. Rev. B 98, 085433 (2018). doi:10.1103/PhysRevB.98.085433.
 - [39] H. Zhang and O. D. Miller, *Quasinormal Coupled Mode Theory*, 2020, arXiv:2010.08650.
 - [40] D. M. Pozar, *Microwave Engineering: Theory and Techniques* (Hoboken, NJ: Wiley, 2021). ISBN 978-1-119-77061-9.
 - [41] C. Vassallo, *Optical Waveguide Concepts* (Amsterdam: Elsevier, 1991). ISBN 978-0-444-88684-2.
 - [42] G. K. Svendsen, M. W. Haakestad, and J. Skaar, *Reciprocity and the Scattering Matrix of Waveguide Modes*, Phys. Rev. A 87, 013838 (2013). doi:10.1103/PhysRevA.87.013838.
 - [43] COMSOL, *COMSOL 6.4 RF Modules User's Guide, S-Parameter Calculations*, 2025.
 - [44] COMSOL, *Multiphysics Cyclopedia: The Finite Element Method (FEM)*, 2017.
 - [45] Y. Yan, F. Li, J. Shen, M. Zhuang, Y. Gao, W. Chen, Y. Li, Z. Wu, Z. Dong, and J. Zhu, *Highly Intelligent Forward Design of Metamaterials Empowered by Circuit-Physics-Driven Deep Learning*, Laser Photonics Rev. 19, 2400724 (2024). doi:10.1002/lpor.202400724.
 - [46] L. Xu, M. Rahmani, Y. Ma, D. A. Smirnova, K. Z. Kamali, F. Deng, Y. K. Chiang, L. Huang, H. Zhang, S. Gould *et al.*, *Enhanced Light-Matter Interactions in Dielectric Nanostructures via Machine-Learning Approach*, Adv. Photonics 2, 026003 (2020). doi:10.1117/1.AP.2.2.026003.
 - [47] Y. Jing, H. Chu, B. Huang, J. Luo, W. Wang, and Y. Lai, *A Deep Neural Network for General Scattering Matrix*, Nanophotonics 12, 2583 (2023). doi:10.1515/nanoph-2022-0770.
 - [48] A.-P. Blanchard-Dionne and O. J. F. Martin, *Teaching Optics to a Machine Learning Network*, Opt. Lett. 45, 2922 (2020). doi:10.1364/OL.390600.

- [49] T. Zhang, C. Y. Kee, Y. S. Ang, E. Li, and L. K. Ang, *Symmetry Enhanced Network Architecture Search for Complex Metasurface Design*, IEEE Access **10**, 73533 (2022). doi:10.1109/ACCESS.2022.3190419.
- [50] S. So, J. Mun, and J. Rho, *Simultaneous Inverse Design of Materials and Structures via Deep Learning: Demonstration of Dipole Resonance Engineering Using Core–Shell Nanoparticles*, ACS Appl. Mater. Interfaces **11**, 24264 (2019). doi:10.1021/acsami.9b05857.
- [51] X. Ma, Y. Ma, P. Cunha, Q. Liu, K. Kudtarkar, D. Xu, J. Wang, Y. Chen, Z. J. Wong, M. Liu et al., *Strategical Deep Learning for Photonic Bound States in the Continuum*, Laser Photonics Rev. **16**, 2100658 (2022). doi:10.1002/lpor.202100658.
- [52] O. Khatib, S. Ren, J. Malof, and W. J. Padilla, *Learning the Physics of All-Dielectric Metamaterials with Deep Lorentz Neural Networks*, Adv. Opt. Mater. **10**, 2200097 (2022). doi:10.1002/adom.202200097.
- [53] W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, *Probabilistic Representation and Inverse Design of Metamaterials Based on a Deep Generative Model with Semi-Supervised Learning Strategy*, Adv. Mater. **31**, 1901111 (2019). doi:10.1002/adma.201901111.
- [54] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, *Generative Model for the Inverse Design of Metasurfaces*, Nano Lett. **18**, 6570 (2018). doi:10.1021/acs.nanolett.8b03171.
- [55] J. Xu, P. Xu, Z. Yang, F. Liu, L. Xu, J. Lou, B. Fang, and X. Jing, *Freeform Metasurface Design with a Conditional Generative Adversarial Network*, Appl. Phys. A **130**, 530 (2024). doi:10.1007/s00339-024-07694-2.
- [56] C. Yeung, B. Pham, R. Tsai, K. T. Fountaine, and A. P. Raman, *DeepAdjoint: An All-in-One Photonic Inverse Design Framework Integrating Data-Driven Machine Learning with Optimization Algorithms*, ACS Photonics **10**, 884 (2023). doi:10.1021/acsp Photonics.2c00968.
- [57] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, *Deep Neural Network Inverse Design of Integrated Photonic Power Splitters*, Sci. Rep. **9**, 1368 (2019). doi:10.1038/s41598-018-37952-2.
- [58] Y. Qian, B. Ni, Z. Feng, H. Ni, X. Zhou, L. Yang, and J. Chang, *Deep Learning for the Design of Random Coding Metasurfaces*, Plasmonics **18**, 1941 (2023). doi:10.1007/s11468-023-01919-5.
- [59] D. Liu, Y. Tan, E. Khoram, and Z. Yu, *Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures*, ACS Photonics **5**, 1365 (2018). doi:10.1021/acsp Photonics.7b01377.
- [60] G. You, C. Qian, S. Tan, E. Li, and H. Chen, *Driving Deep-Learning-Based Metasurface Design with Kramers-Kronig Relations*, Phys. Rev. Appl. **22**, L041002 (2024). doi:10.1103/PhysRevApplied.22.L041002.
- [61] Y. Gao, W. Chen, F. Li, M. Zhuang, Y. Yan, J. Wang, X. Wang, Z. Dong, W. Ma, and J. Zhu, *Meta-Attention Deep Learning for Smart Development of Metasurface Sensors*, Adv. Sci. **11**, 2405750 (2024). doi:10.1002/advs.202405750.
- [62] Y. Xu, J.-Q. Yang, K. Fan, S. Wang, J. Wu, C. Zhang, D.-C. Zhan, W. J. Padilla, B. Jin, J. Chen et al., *Physics-Informed Inverse Design of Programmable Metasurfaces*, Adv. Sci. **11**, 2406878 (2024). doi:10.1002/advs.202406878.

- [63] P. Lalanne, W. Yan, K. Vynck, C. Sauvan, and J.-P. Hugonin, *Light Interaction with Photonic and Plasmonic Resonances*, Laser Photonics Rev. **12**, 1700113 (2018). doi:10.1002/lpor.201700113.
- [64] F. Alpeggiani, N. Parappurath, E. Verhagen, and L. Kuipers, *Quasinormal-Mode Expansion of the Scattering Matrix*, Phys. Rev. X **7**, 021035 (2017). doi:10.1103/PhysRevX.7.021035.
- [65] M. Benzaouia, J. D. Joannopoulos, S. G. Johnson, and A. Karalis, *Quasi-Normal Mode Theory of the Scattering Matrix, Enforcing Fundamental Constraints for Truncated Expansions*, Phys. Rev. Res. **3**, 033228 (2021). doi:10.1103/PhysRevResearch.3.033228.
- [66] C. Sauvan, T. Wu, R. Zarouf, E. A. Muljarov, and P. Lalanne, *Normalization, Orthogonality, and Completeness of Quasinormal Modes of Open Systems: The Case of Electromagnetism*, Opt. Express **30**, 6846 (2022). doi:10.1364/OE.443656.
- [67] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-Based Learning Applied to Document Recognition*, Proc. IEEE **86**, 2278 (1998). doi:10.1109/5.726791.
- [68] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, 2021, arXiv:2104.13478.
- [69] J. E. Gerken, J. Aronsson, O. Carlsson, H. Linander, F. Ohlsson, C. Petersson, and D. Persson, *Geometric Deep Learning and Equivariant Neural Networks*, Artif. Intell. Rev. **56**, 14605 (2023). doi:10.1007/s10462-023-10502-7.
- [70] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko *et al.*, *Highly Accurate Protein Structure Prediction with AlphaFold*, Nature **596**, 583 (2021). doi:10.1038/s41586-021-03819-2.
- [71] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, *E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials*, Nat. Commun. **13**, 2453 (2022). doi:10.1038/s41467-022-29939-5.
- [72] B. Midtvedt, J. Pineda, F. Skärberg, E. Olsén, H. Bachimanchi, E. Wesén, E. K. Esbjörner, E. Selander, F. Höök, D. Midtvedt *et al.*, *Single-Shot Self-Supervised Object Detection in Microscopy*, Nat. Commun. **13**, 7492 (2022). doi:10.1038/s41467-022-35004-y.
- [73] M. Winkels and T. S. Cohen, *Pulmonary Nodule Detection in CT Scans with Equivariant CNNs*, Medical Image Analysis **55**, 15 (2019). doi:10.1016/j.media.2019.03.010.
- [74] A. Bogatskiy, B. Anderson, J. T. Offermann, M. Roussi, D. W. Miller, and R. Kondor, *Lorentz Group Equivariant Neural Network for Particle Physics*, in *International Conference on Machine Learning*, 2020. doi:10.48550/arXiv.2006.04780.
- [75] S. Ravanbakhsh, *Universal Equivariant Multilayer Perceptrons*, in *Proc. 37th Int. Conf. Mach. Learn.*, 7996, PMLR, 2020.
- [76] M. Finzi, M. Welling, and A. G. Wilson, *A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups*, 2021, arXiv:2104.09459.
- [77] G. Cesa, L. Lang, and M. Weiler, *A Program to Build E(N)-Equivariant Steerable CNNs*, in *International Conference on Learning Representations*, 2022.

