

Supervised Machine Learning for Space Antenna Performance Prediction

By Niels Skovgaard Jensen

Master Thesis



Supervised Machine Learning for Space Antenna Performance Prediction

Master Thesis

July, 2022

By

Niels Skovgaard Jensen

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Cover photo: Vibeke Hempler, 2012

Published by: DTU, Department of Electrical Engineering, HC Ørsteds Plads, Building 348, 2800 Kgs. Lyngby Denmark
www.elektro.dtu.dk

ISSN: [0000-0000] (electronic version)

ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)

ISBN: [000-00-0000-000-0] (printed version)

Approval

This thesis has been prepared over five months at the Institute of Electrical Engineering, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

Niels Skovgaard Jensen - s174010


.....
Niels Skovgaard Jensen

Signature

14/07/2022

.....
Date

Abstract

Numerical simulation techniques are essential to modern antenna engineering. As engineering problems are increasing in scale and complexity, the demand for fast and accurate computations is paramount. Machine learning models of these numerical simulations are a way to alleviate some of the computational burdens. This is done using machine learning methods to make statistical predictions within a space of possible antenna designs, from which a set of samples have already been simulated. These types of models are usually referred to as surrogate models. Microwave and antenna surrogate modeling is a mature field; still, knowledge about predicting antenna far-field radiation patterns with machine learning models is in its infancy.

This thesis explores supervised machine learning models for predicting far-field antenna radiation patterns from antenna parameters. Neural network and Gaussian process models are investigated for use as direct prediction models. Latent space models such as principal component analysis, autoencoders, and variational autoencoders are also used to explore the behavior of the high-dimensional radiation pattern datasets. Semi-supervised models that combine the direct prediction surrogates and latent space models into a single larger model are also proposed.

We show that both neural network and Gaussian process models can effectively be trained to predict the far-field in some of the investigated antenna cases, whereas other of the studied cases seem to be more complex.

We also show that the investigated latent space methods are highly effective dimensionality reduction tools when used on the sampled far-field datasets. The reduced dimension spaces found seem to contain even lower-dimensional dataset manifolds with a substantial connection to the parameterizations of the antennas. Furthermore, all investigated latent space methods seem generative and can create unseen dataset samples by interpolation of the discovered latent spaces. Gaussian processes are also demonstrated to take advantage of the lower dimensionality latent spaces for field prediction.

These results indicate that both Gaussian process regression and neural networks can be effective far-field antenna machine learning models and that both PCA and autoencoders can be applied effectively as dimensionality reduction techniques to sampled far-field datasets. These insights could help generate surrogate antenna models, speeding up numerical simulators and leading to more well-optimized antenna designs.

Acknowledgements

Line Katrine Harder Clemmensen, Associate Professor at DTU, Supervisor

Tue Herlau, Assistant Professor at DTU, Co-Supervisor

Simon Aertssen, Machine Learning Engineer at TICRA Fond, Company-Supervisor

Oscar Borries, Head of Team, Mathematics & AI at TICRA Fond, Company-Supervisor

I would also like to thank the rest of the team at TICRA for receiving me, for creating such a pleasant working environment, and for providing access to TICRA Tools. In no other company can you have an hour-long impromptu talk about spherical sampling strategies, with everyone having an opinion; I enjoy that very much.

Finally, I would like to say a special thanks to Diana for her unending patience and understanding of my human limitations under pressure and for being my partner in sabotage.

Contents

Preface	ii
Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Formal Problem Statement and Research Questions	3
1.2 Thesis Structure	4
2 Data sets and Design of Experiments	7
2.1 Datasets Field Expressions	7
2.2 Data structures and Frameworks	8
2.3 Sampling Methods	9
2.4 Error Metrics	10
2.5 Reflector Antenna (RFLCT)	11
2.6 Patch Antenna (PATCH)	13
2.7 Circular Horn With Dielectric Lense (CHA)	16
2.8 ML-Array (MLA)	18
3 Literature Review	23
3.1 General Surrogate Modelling	23
3.2 Microwave Surrogate Modelling	24
3.3 Antenna Surrogate Modelling	25
3.4 Far-field modeling	26
4 Methods	29
4.1 Neural Networks	29
4.2 Gaussian Process Regression	33
4.3 Latent Space Methods	36
4.4 Latent Space Regression Models	39
5 Experiments and Results	43
5.1 Latent Space Representations	43
5.2 Direct Parameter to Field Prediction	53
5.3 Model Sensitivity to Dataset Sizes	60
5.4 Latent Space Regression	61
6 Discussion and Conclusion	67
6.1 Latent Space Models	67
6.2 Direct Surrogate Performance	68
6.3 Novelty of This Thesis	69
6.4 Implications	69
6.5 Conclusion	69
6.6 Future Work	70
Bibliography	73
A Background Theory	79
A.1 Electromagnetic Fields	79

B Phasor Notation	86
C Fields from a hertzian dipole	87
D Single Point of the Far-Field Sphere Parameter Sweeps	88
D.1 PATCH	88
D.2 CHA	89
D.3 RFLCT	89
E Physics Informed Neural Networks in Far-Field Surrogates	91
F VAE Reconstruction Examples	93

List of Figures

1.1 Far Field Example	2
1.2 Parameter to Field Relation	2
1.3	3
2.1 Holdout Validation Scheme	9
2.2 Reflector antenna model from TICRA Tools	12
2.3 RFLCT Dataset Example	12
2.4 Histogram of RFLCT dataset	12
2.5 Single point on far-field sphere diameter sweep	13
2.6 Patch antenna model	14
2.7 Histogram of PATCH dataset	14
2.8 PATCH Dataset Example	15
2.9 Patch Single Point on Far-Field Sphere Parameter Sweep	16
2.10 Circular Horn Antenna Model	16
2.11 2D Histogram of CHA dataset	17
2.12 Example of co and cross-polar magnitude in CHA dataset	17
2.13 Effect of relative permittivity on a single point of the far-field sphere.	18
2.14 MLA Array Setup	18
2.15 ML Array Closeup	19
2.16 Histogram of MLA Dataset	20
2.17 MLA dataset magnitude example	20
4.1 Mesh Grid Neural Network Representation	30
4.2 Continuous space neural network representation	31
4.3 Pyramidal Neural Network Architecture	31
4.4 Autoencoder architecture	37
4.5 Variational Autoencoder (VAE) Architecture	38
4.6 Architecture of latent space regression model	40
4.7 AE-FC backpropagation paths	40
5.1 PCA Reconstruction loss for all datasets, solid is training set and striped is validation loss	44
5.2 RFLT dataset PCA latent space with parameter coloring	44
5.3 RFLCT Dataset 3D PCA plot	45
5.4 PCA of PATCH dataset with parameter coloring	45
5.5 First two principal components of the CHA dataset with parameter coloring.	46
5.6 3D PCA Projection of CHA dataset	46
5.7 MLA Dataset PCA Components	47
5.10 Autoencoder architecture	48
5.8 CHA cross-component latent space reconstruction	49
5.9 CHA cross-component latent space reconstruction	50
5.11 Autoencoder training	51
5.13 2D Autoencoder latent space CHA dataset	51
5.12 CHA dataset far-field reconstruction from 2D autoencoder latent space	52
5.15 Reconstruction performance of the 5 VAE models also shown in fig. 5.14. Reconstruction examples can be see in appendix F.	53
5.16 PDNN Patch Prediction	56

5.17	PDNN CHA Prediction	56
5.18	PDNN training progression	57
5.19	Examples of loss landscapes for the different kernel types	57
5.24	GP Dataset Size Sensitivity	61
5.26	PCA and GP-PCA reconstruction loss as function of latent space size for the PATCH dataset	62
5.27	GP-PCA Performance CHA	63
5.28	GP-PCA Performance MLA	63
5.29	Showcase of GP Prediction in varying number of latent dimensions for the CHA dataset.	64
5.30	Prediction by GP model into Iantet space	65
D.1	Single point on the far-field sphere value as function of coaxial x placement in PATCH dataset	88
D.2	Single point on the far-field sphere value as function of coaxial Y placement in PATCH dataset	88
D.3	Single point on the far-field sphere value as function of loss tangent in CHA dataset	89
D.4	Single point on the far-field sphere value as function of focal length in RFLCT dataset	89
D.5	Single point on the far-field sphere value as function of offset in RFLCT dataset	90
F.1	2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-5}$	93
F.2	2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-4}$	93
F.3	2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-3}$	93
F.4	2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-2}$	94
F.5	2D VAE reconstruction at $\beta_{KL} = 10$	94

List of Tables

2.1 Datasets Summary	21
3.1 Comparison of methods in recent microwave surrogate literature	27
5.1 VAE CNN architecture	53
5.2 KNN-regression baselines	55
5.3 PDNN regression performance	56
5.4 Direct Prediction Performance of SOGP	58
5.5 Validation Error Comparison of Surrogates to Baselines	60

Acronyms

Abbreviation	Description
GP	Gaussian Process, mostly in the context of GP regression
SOGP	Single Output Gaussian Process, GP model that does not correlate between output dimensions
MOGP	Multi output Gaussian process, with correlation between output dimensions, also called Co-Kriging.
NN	Neural Networks
CNN	Convolutional Neural Networks
FC-NN	Fully Connected Neural Networks
PCA	Prinicpal Component Analysis
PLF	Power Loss Factor
AE	Autoencoder
VAE	Variational Autoencoder
ReLU	Rectified Linear Unit
KL-Divergence	Kullbach-Liebler Divergence
PINNs	Physics Informed Neural Networks
RFLCT	Reflector dataset, see section 2.5
PATCH	Patch antenna dataset, see section 2.6
CHA	Circular Horn Antenna dataset, see section 2.7
MLA	ML-Array dataset, see section 2.8

Notation

The mathematics described in this thesis includes electromagnetics, machine learning, statistics, and probability theory. These fields have sizeable notational traditions, inevitably leading to a certain amount of notational overlap. We have tried to maintain as much of the original notation of all fields, though some notation is changed slightly, usually with suffixes, from standard practices.

Bold capitalized symbols denote vector fields; uncapitalized bold symbols indicate vectors. Superscript asterisk denotes complex conjugates. Subscript asterisks denote validation sets. An arrow either means basis vector or training set data. We will also follow the electrical engineering convention of j being the imaginary unit.

Notation	Meaning
x	Scalar
\boldsymbol{x}	Vector
$\hat{\boldsymbol{x}}$	Basis vector
\boldsymbol{X}	Vector field
$\overline{\boldsymbol{X}}$	Matrix
$\boldsymbol{x}[i]$	i'th sample of discretized \boldsymbol{x}

Symbol	Description
x, y, z	Cartesian coordinates
r, θ, ϕ	Spherical coordinates
$\hat{a}_x, \hat{a}_y, \hat{a}_z$	Cartesian basis vectors
$\hat{a}_r, \hat{a}_\theta, \hat{a}_\phi$	Spherical basis vectors
$\hat{\rho}_{co}, \hat{\rho}_{cross}$	Co and Cross polar basis vectors
j	Imaginary unit
\mathcal{E}	Electric field intensity
\mathcal{D}	Electric flux density
\mathcal{B}	Magnetic field intensity
\mathcal{H}	Magnetic flux density
E	Time harmonic electric field intensity
D	Time harmonic electric flux
B	Time harmonic magnetic flux
H	Time harmonic magnetic field intensity
F	Radially normalised electric far field
U	Radiation intensity
D	Antenna directivity
G	Antenna gain
μ	Magnetic permeability
ϵ	Electric permittivity
f	Frequency
ω	Angular frequency $\omega = 2\pi f$
β	Wave number $\beta = \omega^2 \epsilon \mu$
Γ	Reflection Coefficient, often $S_{11} = \Gamma$
$\Gamma(\cdot)$	Gamma function
e_d	Dielectric loss factor
e_c	Conduction loss factor
\bar{S}	Scattering matrix
Ω_d	Parameter design space
χ	Parameter Design Vector
Z	Latent Space
z	Sample vector in the latent space Z
R_f	Response from a fine model such as a simulation tool
R_c	Response from a coarse model such as a surrogate model which approximates a fine model
P	Parameter mapping
\mathcal{L}	Loss function
$\theta_{NN}, \phi_{NN}, \varphi_{NN}$	Neural Network parameters
H	Shannon Entropy
\mathcal{N}	Normal Distribution
\mathcal{D}_{KL}	Kullback Liebler divergence
\mathcal{GP}	Gaussian process
ELBO	Evidence Lower Bound
$\delta_{i,j}$	Kronecker Delta Function
$k(\cdot, \cdot)$	Kernel function
\bar{C}_N	\mathcal{GP} covariance matrix
ℓ	RBF or Matérn kernel length scale
p	Periodicity factor of a periodic kernel
ν	Matérn kernel smoothness factor

1 Introduction

Antennas are at the heart of modern-day communications technologies. The IEEE definition of an antenna is defined as: "The part of a transmitting or receiving system that is designed to radiate or to receive electromagnetic waves." Since every wireless technology needs a method to convert energy from wired applications into wireless signals, every wireless technology needs antennas [1].

As communications have become increasingly important in the operation of modern society, increasingly strict requirements for antenna systems are needed. Examples such as satellite communication, 5G, and IoT require more advanced antenna systems that push the boundaries of electromagnetics to optimize shapes, sizes, and material properties to find designs that comply with these systems' requirements [2–4]. An example could be geostationary satellite communications, where exact knowledge about the antenna's radiation pattern is needed to ensure that the correct signal strength is present in the area of service.

Because of the intrinsically tricky problem of finding closed-form or approximate expressions for anything but the most straightforward antenna setups, modern-day antenna modeling is primarily a field that relies entirely on electromagnetic simulation methods such as finite element analysis or finite difference methods. These mesh-grid integral equation solvers are extremely precise, and modern antenna modeling software is a very mature technology that is proven to lead to antenna designs that could never have been invented without numerical simulations[5, sec. 1.5].

Some of the most common objectives that antenna engineers face are optimization of antenna characteristics, and statistical analysis of performance variations of an antenna [6]. These goals are usually obtained by varying a parameterized model of antenna dimensions and material properties, which are then simulated and evaluated. Both optimization and variational analysis require a large set of simulations. This means that even if single simulations are both computationally cheap and accurate, many practical applications require many simulations, resulting in a significant computational expense.

When evaluating antennas' performance, the far-field radiation pattern¹ is often the most important characteristic. Along with the positioning of the antenna, it determines how much power can be received and transmitted by the antenna at different angles of incidence. This far-field is a complex function over a sphere that describes the intensity and phase of the radiated field in all directions away from the antenna. In this thesis we denote this function as $\mathbf{F}(\theta, \phi)$.

The antenna is defined by a set of design parameters called the *antenna parameterization*. Often an extensive amount of design parameters are needed to define an antenna fully, and in this thesis, we will only be varying a couple of parameters defined in the *parameterization vector*, χ . Both \mathbf{F} and χ will be defined more formally later in the text. In fig. 1.1, an example of a far-field of one of the investigated datasets can be seen in a rectangular plot. Here the magnitude of \mathbf{F} is plotted as a function of θ , and several ϕ -cuts are displayed. A simple illustration of the physical interpretation of this plot can be seen in fig. 1.3a.

¹Often referred to as the 'Far-field' or 'radiation pattern,' all these expressions have exactly the same meaning in this thesis.

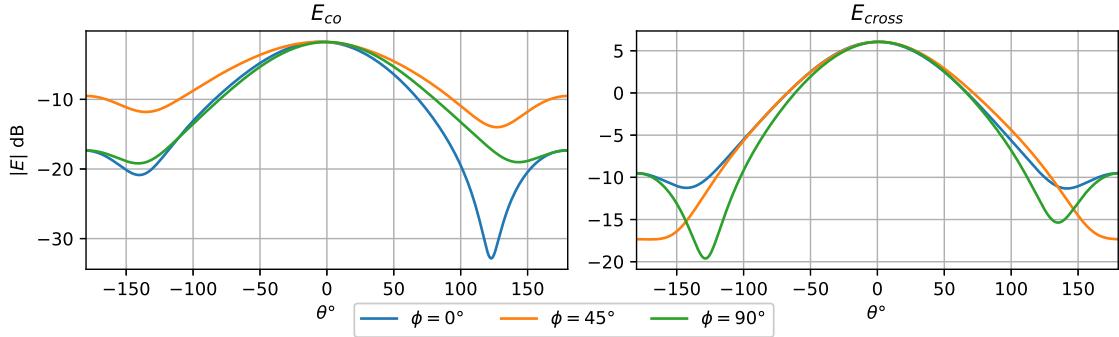


Figure 1.1: Patch Antenna far-field magnitude example for illustration purposes. Here 3 cuts of the far-field sphere have been made at $\phi = 0, 45$ and 90 degrees. The x-axis is then the magnitude of the field in power decibels at different values of θ .

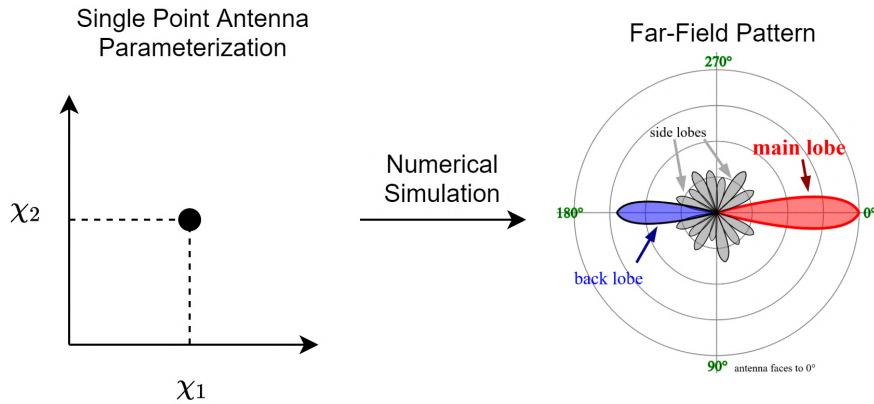


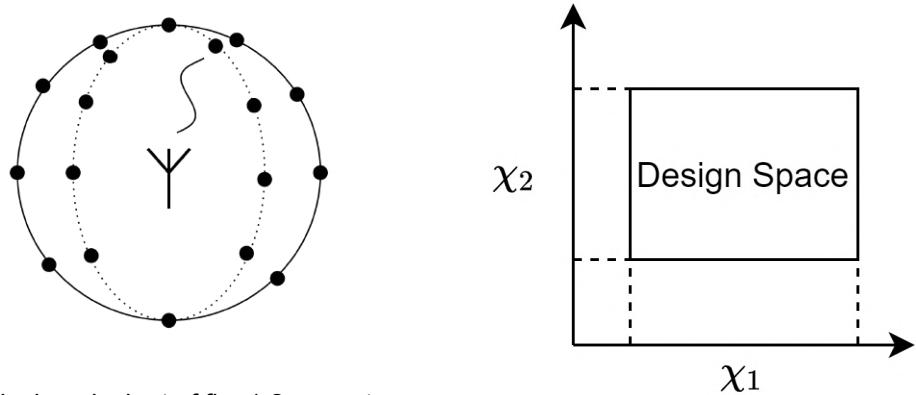
Figure 1.2: Numerical simulations provide a projection of an antenna parameterization, χ , into a spherical far-field function F . Far-field drawing used with CC-license from [7]

As seen in fig. 1.2, an example of this parameterization vector, χ leads to an example of a far-field, $F(\theta, \phi)$. This is simply equivalent to a unique antenna having its own far-field pattern. Furthermore, for optimization and variational analysis, the space of potential antenna parameterizations being investigated are often referred to as the *design space*, see fig. 1.3b, or *the parameter space*. We will use both terms interchangeably.

A restricted design space is widespread in antenna design since most parameterizations only have a very limited set of feasible values. As an example, one could imagine stringent dimensional requirements on a satellite or in a phone, and strict material properties limited by the feasibility of manufacturing and material physics[5].

Because of the large number of evaluations needed to characterize the design space and find locally optimal antenna designs, much work has been put into optimization strategies that reduce the number of simulations required for a converged result. One of these methods is *surrogate modeling*, also called metamodels or model emulators. Surrogate models approximate some precise and difficult to evaluate models, such as a simulation or measurement, by making statistical inferences between the parameterizations, χ that define the antenna, and the far-field F , that is produced by the model.

In this thesis, we will investigate methods for generating far-field antenna surrogate models and use dimensionality reduction tools for understanding the very high dimensional



(a) Practical equivalent of fig. 1.2; an antenna with a parameterization χ is radiation outwards in a sphere. This radiation intensity, $F(\theta, \phi)$ is then measured at a set of sample point defined by the spherical coordinates.

(b) Design space, Ω_D , is a bounded space of parameterizations where the antenna is evaluated.

Figure 1.3

far-field dataset patterns. All the used datasets come from the TICRA software suite, and all antennas are real antenna case studies that have been proposed by TICRA.

1.1 Formal Problem Statement and Research Questions

The investigated problem in this thesis is; given a dataset of antenna far-fields and antenna parameterizations, how well can machine learning methods learn the relations between the design space of the parameterizations and its given space of far-field radiation patterns?

Formally, the problem being investigated can be described as follows.

Given an antenna parameterization χ , the far-field \mathbf{F} of an antenna can be represented by a complex vector field on a sphere with two orthogonal components:

$$\mathbf{F}(\theta, \phi; \chi) \in \mathbb{C}^2 \text{ for } \theta \in [0, \pi], \phi \in [0, 2\pi], \chi \in \Omega_D \quad (1.1)$$

Where θ and ϕ are spatial coordinates on the sphere and χ is a vector representing a physical parameterization of an antenna structure within a design space of N real design parameters, $\Omega_D \in \mathbb{R}^N$.

In this project, we can see the far-field \mathbf{F} as a response from a fine model $R_f(\chi)$, which has been obtained through numerical solvers, in our case, provided by the TICRA software suite.

$$R_f(\chi) = \mathbf{F}(\chi | \theta, \phi) \quad (1.2)$$

The goal is then to investigate to what extent modern machine learning methods can find a coarse model response $R_c(\chi)$

$$R_c(\chi) \simeq R_f(\chi) \quad (1.3)$$

The main questions that the thesis will try to provide answers to are:

1. To what extent can machine learning models be trained to be effective antenna surrogates, and can unsupervised methods aid in surrogate effectiveness?
2. What are the possibilities for visual interpretation through dimensionality reduction with machine learning tools?
3. Can linear latent space models be used for effective dimensionality reduction of sampled electromagnetic far fields, and can non-linear models significantly outperform linear models at equal latent space dimensions?
4. How many evaluations of the fine model R_f , i.e., how much data is needed to train a model within a particular design space Ω_D ?

1.2 Thesis Structure

Chapter 2, **Datasets and Design of Experiments**, will describe the case studies and their respective datasets from an antenna-theoretical perspective to try to give the reader a sense of what the different types of modeled antennas look like and what analytically known relations, if any, the chosen parameterization should have to the far-field.

Chapter 3, **Literature Review**, will do a literature study of the field of surrogate modeling, focusing on methods used in microwave engineering and antenna modeling and discussing more recent developments in surrogate modeling. Then we will use this perspective to outline how this thesis adopts and builds on top of these methods.

Chapter 4, **Methods**, will outline the methods chosen in this thesis for creating far-field surrogate models and reduced dimensionality representations in more detail.

Chapter 5, **Experiments and Results**, will go over the results from creating surrogate models based on the datasets described in chapter 2 with the methods described in chapter 4.

Chapter 6, **Discussion and Conclusion**, will discuss the findings shown in chapter 5 and relate them to the findings of other surrogate literature. We will also discuss the outcome of this thesis project, which novel findings have been obtained, and what future possibilities in far-field antenna surrogate modeling might look like.

Furthermore, appendix A, **Background Theory**, outlines the relevant electromagnetic theory and establish the connection between current distributions on antennas and the far fields that are modeled throughout this thesis. This provides a more rigorous connection between the parameterizations and the far field expressions and outlines some of the antenna-terminology which will be used in the thesis.

1.2.1 Reading Guide

Appendix A, **Background Theory**, was initially written as a theoretical introduction to the rest of the thesis, but it has been moved to be an appendix for readability purposes. It is important to note that multiple relations in this section are important to understand the physics of the modeled problems more thoroughly. Therefore, we will refer to the sections and equations described in this appendix throughout the main text. The interested reader might want to read the section in full, else a short skim of the relations and sections should be sufficient to understand the rest of the text when the appendix is referred to.

Chapter 2 will first describe how the datasets are structured and how they are sampled from the far-field. Then it will describe all four case studies in detail, and therefore the

reader might want to skim these later sections of the chapter and return to them as needed when reading the results in chapter 5.

There are brief summary sections at the end of each chapter, which can be utilized for getting an overview of the structure and main points of each section.

Project Source Code

All data, source code, and serialized trained models are released open-source on GitHub². The main part of this github repository is the installable 'SSAPP' (Supervised machine learning for Space Antenna Performance Prediction) package, which lays the foundation of all of the analysis conducted in the throughout this thesis. Full reproducibility of all results should be possible by installing the SSAPP package, and its requirements and then running appropriate Jupyter notebooks and analysis scripts.

²<https://github.com/SkoogyDan/Supervised-Machine-Learning-for-Space-Antenna-Performance-Prediction>.

2 Data sets and Design of Experiments

This chapter outlines the different antenna cases that have been investigated during this thesis. It outlines the physical simulated setup of each case-study including its parameterization and will briefly discuss the antenna-theory expectations of the relationship between the field and the design parameters, if they exist in closed form. It will also explain the chosen field representation and sampling methods.

Dataset Naming

Since most of the datasets have very long and descriptive names that are not easily shortened without loosing descriptiveness; dataset acronyms are used to denote the different datasets. The datasets acronyms will always be denoted fully capitalized. In this section we will use the full names to establish the acronyms and in later sections only the acronyms will be used.

2.1 Datasets Field Expressions

This section will provide a formal mathematical form of how the datasets modeled in this thesis are related to the theory shown appendix A, where more electromagnetically rigorous version of this section can also be found.

The far-field radiation pattern is a complex vector field on a sphere with two orthonormal components. Written in a polar coordinates, it takes the form:

$$\mathbf{F}(\theta, \phi) = \hat{\rho}_{co} E_{co} e^{j\varphi_{co}} + \hat{\rho}_{cross} E_{cross} e^{j\varphi_{cross}} \quad (2.1)$$

Where E_{co} and E_{cross} are scalar magnitudes the co and cross-polar parts of the electric field, φ_{co} and φ_{cross} are their respective phases and $\hat{\rho}_{co}$ and $\hat{\rho}_{cross}$ are the co and cross polarization orthonormal basis vectors. E_{co} , E_{cross} , φ_{co} and φ_{cross} are functions of θ and ϕ .

The co and cross component can be seen as the desired and undesired part of the polarization of the field, respectively. In simplified terms the co-polar field will contribute to the transmitted and received power, whereas cross-polar field will be lost. More details about these two components can be found in appendix A.1.3.

When the continuous function eq. (2.1) is discretized over the sphere it is defined it takes the form:

$$\mathbf{F}_s[\theta_i, \phi_j] = \hat{\rho}_{co} E_{co}[\theta_i, \phi_j] e^{j\varphi_{co}[\theta_i, \phi_j]} + \hat{\rho}_{cross} E_{cross}[\theta_i, \phi_j] e^{j\varphi_{cross}[\theta_i, \phi_j]} \quad (2.2)$$

Where we have adapted sampling notation from signal processing like in [8], where $f_s[\cdot]$ indicates the sampled version of a continuous function $f(\cdot)$.

In eq. (2.2) θ_i, ϕ_j are the scalar values of the θ, ϕ vectors, with $i \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, M\}$. In this thesis we therefore have sampled far fields of dimensionality $\mathbf{F}_s[\theta, \phi] \in \mathbb{C}^{2 \times N \times M}$, though for most of the machine learning methods we have chosen to represent it with its euclidean counterpart $\mathbf{F}_s[\theta, \phi] \in \mathbb{R}^{4 \times N \times M}$. In all fields that are modelled in this thesis, $N = 361$ and $M = 3$. Since usually a large part of the field variation is limited to θ .¹

¹Many examples of far fields from different antennas can be seen in [5], here they also follow the antenna theoretic customs of limit ϕ -samples, since not much variation happens here.

A single example of a discretized $\mathbf{F}_s \in \mathbb{R}^{4 \times N \times M}$, will be referred to as a *far-field sample* since it is a single far-field coming from a single parameterization. We will use the term *sample on the far-field sphere* if we are referring to a single value like $\mathbf{F}_s[\theta_i, \phi_j] \in \mathbb{R}^4$.

Since these fields are power normalized to the input power, we can also relate them to the antenna's gain², as defined in eq. (A.29), through.

$$G_{co}(\theta, \phi) = |E_{co}|^2 \quad (2.3)$$

$$G_{cross}(\theta, \phi) = |E_{cross}|^2 \quad (2.4)$$

Which will mainly be shown in power decibels, i.e., on the form:

$$G_{co,dB}(\theta, \phi) = 20 \log_{10} (|E_{co}|) \quad (2.5)$$

$$G_{cross,dB}(\theta, \phi) = 20 \log_{10} (|E_{cross}|) \quad (2.6)$$

Which will mostly be referred to on a form like $|E|$ dB.

Therefore when the electric field is shown in decibel, it is accounting for an expression like eq. (A.29), with losses included.³

When plots that display the direct real and imaginary components of the co and cross-polar values of \mathbf{F} , we will display them as $\text{Re}(E_{co})$, $\text{Im}(E_{co})$, $\text{Re}(E_{cross})$ and $\text{Im}(E_{cross})$, respectively.

2.2 Data structures and Frameworks

All the datasets are implemented as the PyTorch dataset type, which allows PyTorch dataloaders to handle presentation the datasets to the models during training and evaluation. All datasets are randomized, such that training-schemes do not have a particular order and that training, validation and test datasets are as representative as possible.

For neural network models a holdout method has been used where the datasets are split into training, validation and tests sets. The validation set is evaluated during training together with the training set, but the validation set is never used for training the models, only for estimating if the model is overfitting. A test set is then kept for post-training validation to ensure that the model has not been stopped in a local validation-optimal minimum that does not generalize. The a train/validation split has been conducted beforehand, so all models train on exactly the same dataset and validation/test splits are then made of the remaining dataset as needed. For models other than neural network models, a simpler test/validation only scheme has been used.

Cross-validation has also been considered, but dataset sizes and model training time has made it impractical to utilize⁴. Furthermore, the validation/test scheme has not indicated

²A metric for how much power is radiated in a certain direction, which also takes into account multiple loss mechanisms

³With the use of decibels, the unfamiliar reader should be careful in their interpretation of the magnitude plots in dB and their relation to other metrics like mean squared error. Of course, a relationship does exist, but it will also be common for a significant amount of the 'real' error to be so close in log-space to the actual value that it will be hard to differentiate the two visually, especially in high-gain regions of the gain-pattern. Practically, thinking of 6dB as a power doubling, 12 as a quadrupling, and 20dB as an order of magnitude scaling can be helpful.

⁴Model training time has varied between datasets, but for the larger datasets training time of 10s of minutes to hours has been observed.

that local validation overfitting has happened to any of the finally trained models, and therefore extensive cross-validation has been considered unnecessary.

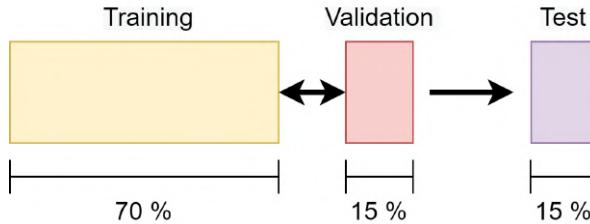


Figure 2.1: Holdout validation scheme used for Neural network

2.3 Sampling Methods

In the process of generating the data from the fine model, R_f as described in chapter 1, a discretization scheme, i.e a sampling method, has to be chosen in both parameter space, Ω_d , and in the far-field F .

Parameter space is euclidean with $\Omega_d \in R^D$ where D is the number of parameters in the parameterization. Whereas the far-field, F , is a two-component complex vector field on the surface of a sphere.

Therefore in the following section, we will discuss the used sampling methods and more advanced sampling methods in both parameter space and for the far-field since the choice of sampling methods has a significant impact on the structure of the datasets and therefore, also on the surrogate model performance.

2.3.1 Spherical Far-Field Sampling Methods

As discovered by Carl Friedrich Gauss in 1827 as his Theorema egregium, there is a fundamental issue when trying to project a spherical surface onto a planar surface; spheres and planes are not isometric. That is, there does not exist a distance preserving transformation between the two, not even locally on a sphere. For machine learning methods where we are representing the field by a 2-dimensional matrix of sampling points in θ and ϕ , there is no 2D matrix representation that can preserve distance and therefore no transformation that can preserve sampling density.

The implication is that a sampling density bias will always be present in the dataset for a given spherical sampling strategy. Therefore, it is vital to be aware of this bias and understand which bias types will be present for different sampling methods.

Turned θ -Cuts

Inspired by antenna theoretic customs, the θ -cut method makes circular cuts at a given constant value of ϕ and then samples equidistantly in θ . This is usually a very valid method for most antenna setups since the largest and most interesting variation happens as a function of θ , when the antenna boresight is aligned with $\theta = 0, \phi = 0$. In most antenna-theoretical literature only a couple or even as little as one θ -cut is shown for a measured antenna, and this is often deemed sufficient to understand the antenna's radiation pattern, examples of this can be seen in [5].

Another approach is to sample in such a way that all points on the sphere are equally distant from each other, a spherically equidistant sampling method is outlined in [9] and should be able to solve the problem of uneven sampling density across the far-field sphere. The related issue would then be how to represent the dataset nicely in a tensor, since it

would lead to large areas of sparsity in the tensor, in the same manner that some map-projections lead to having non-rectangular planar projections.

From an electromagnetic point of view, [10] proposes a spherical sampling method that is based on the dimensions of the electromagnetic structure under analysis. Thus it also provides a method for determining the needed sampling density before far-field calculations are made.

These two methods point to the fact that there are well-known methods that might be less density biased and also that might ensure no oversampling is done. This comes at the cost of going away from a non-sparse tensor representation. Therefore we have chosen during this project to use the ℓ -cut method with a fixed spherical sampling scheme, as to be able to compare models with the same feature dimensionality and to be able to represent the sampled fields nicely in a regular non-sparse tensor structure.

2.3.2 Design Parameter Sampling Methods

Inherently when we want to represent the desired design parameter space Ω_d , we also need to choose a sampling density and sampling method that can make random distributions representative of the chosen design space.

Linear Spaced Sampling is the naïve approach to parameter sampling. Here we choose the bounds of design space and sample equidistantly with the desired sampling density. This sampling strategy is an easily implementable method but has the issue of being entirely arbitrary. I.e., it has the properties of a uniform distribution but has a significant implicit bias in its structure. Therefore random sampling methods are often preferred.

Latin Hypercube Sampling [11] (LHS) is a sampling method that attempts to obtain a more realistic randomly sampled distribution within a the design space. Latin hypercube sampling is performed by making a significantly higher density grid, of which only points that do not share rows and columns are accepted. It is equivalent to placing rooks on a chess board that cannot attack each other.

LHS, and more advanced derivative sampling algorithms, are widespread in surrogate modeling [12, 13], since they generated significantly better quasi-random distributions without a significant extra computational complexity. This thesis uses the SciPy implementation of latin hypercube sampling [14].

2.4 Error Metrics

It is inevitably difficult to make good visualisations of highly dimensional datasets and a error metric that can capture the intuition of how well a high dimensional sample and prediction compare with each other is therefore very important. We have the following error metrics in this thesis:

In order to account for the relative magnitude differences of the antenna setups, *Relative Root Mean Square Error*, is the most widely used metric in this thesis.

$$\text{relRMSE}(y, \hat{y}) = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{\sum \hat{y}_i^2}} \quad (2.7)$$

Mean Squared Error has been used for some of the neural network training, since it achieves the same error rate with as relRMSE, with a scaling factor that can be accounted for in dataset-to-dataset cases. Ensuring that the denominator calculation of eq. (2.7)

does not have to be done at every training iteration⁵.

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \quad (2.8)$$

On acceptable error levels

Giving a single number of some error metric that is 'acceptable' is an impossible task, multiple metrics need to be used and the fields visually inspected to see if they are at an acceptable error level, and each practical case will have widely different requirements, even when using a relative error such as relRMSE. That being said, we will outline some rough guidelines to give the reader some intuitive sense of how good different levels of relRMSE are, in the case of antenna modeling. These are purely anecdotal, and should not be taken as any kind of absolute guidelines.

- $10^0 < \text{relRMSE}$ – No Real Shape is captured, could be achieved by any prediction that always predicts zero.
- $10^{-1} < \text{relRMSE} < 10^0$ – Rough shape of field is captured in dB-magnitude plot. In linear plots this can already visually look good.
- $10^{-2} < \text{relRMSE} < 10^{-1}$ – General shape is obtained in dB-plots, and probably most of the main lobe is captured precisely, can be acceptable in lower demand cases.
- $10^{-3} < \text{relRMSE} < 10^{-2}$ – High precision, better than many antenna simulation/measurement comparison papers.
- $\text{relRMSE} < 10^{-3}$ – Extremely good, has captured almost all detail above measurable dB scales.

These are rough anecdotal guidelines and are in no way meant to be absolute. Be are instead intended to give an intuitive indication of the goodness of the trained models for the reader.

Another point is to be aware of the extremely low energy levels in some of the magnitude plots. For some dataset cases, magnitude dB values down to -300dB can be observed. This is simply a numerical result from the simulators, which could never be recreated in the real world. Thus the reader has to be careful in their analysis of the logarithmic magnitude comparisons.

In the following sections, we will outline the different case studies that this thesis revolves around. Firstly we will give a short introduction of the case, then we will show examples of the dataset, and finally, we will explain, from an antenna-theoretical point of view, the different known relations some of the antenna parameters have to the modeled far-fields.

2.5 Reflector Antenna (RFLCT)

A parabolic reflector antenna, colloquially known as a dish antenna, are antennas that focus the beam of a feed antenna using the geometrical properties of parabolas. This makes them able to have high gain while being cheap and easy to manufacture and are therefore widely used commercially. Advanced reflector antennas are also widely used on spacecraft, making up most of the antennas used for geostationary communications satellites[15]. The reflector under investigation can be seen in 2.2.

⁵In practice this seems to speed up neural net training by a 3-4% in our experience.

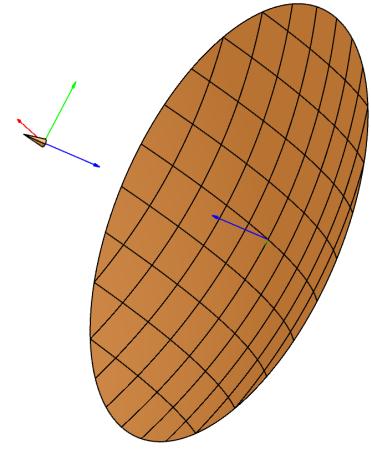


Figure 2.2: Reflector antenna model from TICRA Tools

As seen in fig. 2.3, the reflectors are high gain with large variations in θ . The same can be seen of the θ histogram in fig. 2.4, where a 2D histogram of the entire dataset is shown in linear scale. Here we can also readily see the just of directive the antenna is.

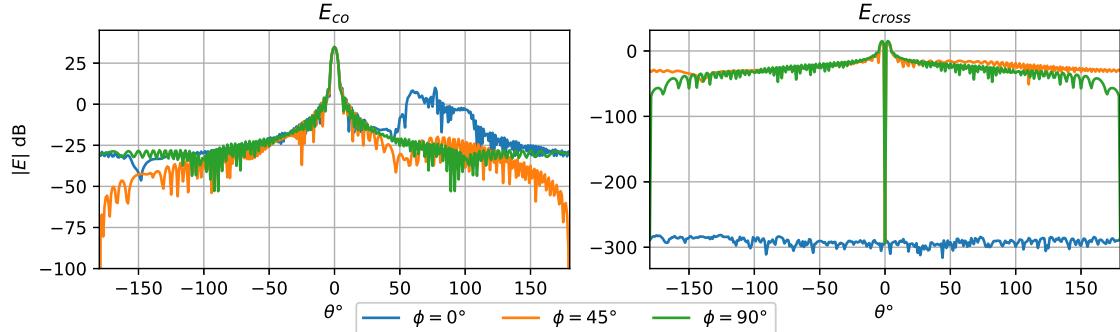


Figure 2.3: RFLCT Dataset Example

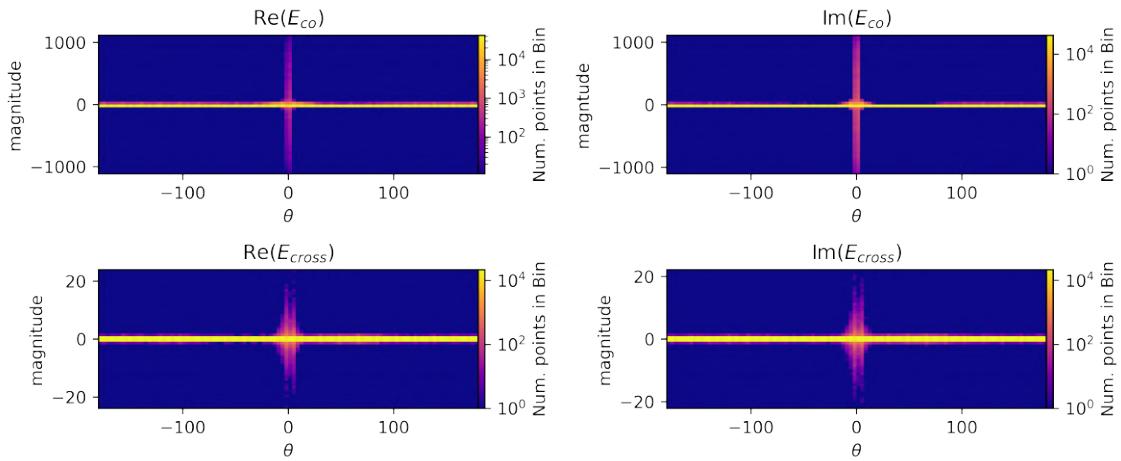


Figure 2.4: 2D Histogram of RFLCT dataset, all ϕ -cuts contribute to the histogram

2.5.1 Design Parameter Description

The *Reflector Diameter* is the physical size of the parabolic reflector. In the dataset the size is normalised by the wavelength, λ , of the excitation frequency.

A lot of mathematical background exists on parabolic reflectors [5, Sec. 15.4.1], though most of the expressions are quite hard to generalize in a fashion that leads to intuition about the dataset at hand. For that goal, a more empirical rule of thumb can be used.

$$D_0 = \frac{4\pi A_{em}}{\lambda^2} \quad (2.9)$$

Where D_0 is the maximum directivity of the antenna, A_{em} is the size of the antenna, and λ is the wavelength. This is a significantly oversimplified relationship. However, it is also expected that a considerable part of the variance of the dataset will be governed by a relationship very close to eq. (2.9) and be a function of the reflector diameter. In fig. 2.5 a plot of the co and cross polar fields at single point of the far-field sphere as a function of diameter can be seen.

The two other parameters are *Focal Length*, which is a measure of the distance between the horn and the reflector, and *Offset*, which offsets the horn from the middle of the reflector. There are no good, generally applicable analytical relations for these two parameters, but it is expected that they will impact the 'side lobes' more than the main beam pattern of the antenna. Their relation to a single sample point can be seen in appendix D.3.

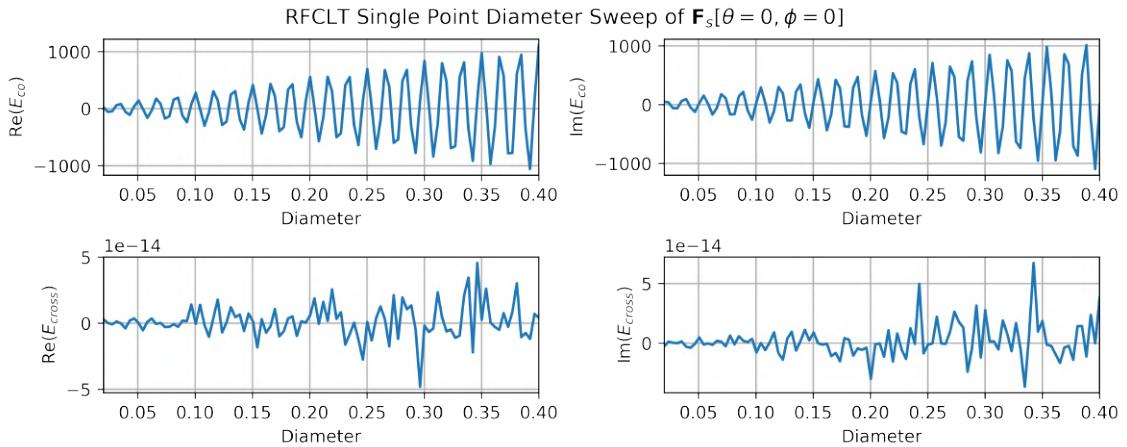


Figure 2.5: Single point on the sampled far-field sphere as a function of reflector diameter. Rest of parameter-relation plots can be found in appendix D.3. Data is explicitly generated for this plot, so parameter sampling density is not identical to RFLCT.

2.6 Patch Antenna (PATCH)

Patch antennas are probably the most manufactured antenna type in existence. In practice, they are manufactured directly on printed circuit boards, making them extremely easy and cheap to manufacture and integrate into any electronic device[5].

They are also widely used in large arrays for a plethora of advanced devices such as military radars, Doppler radars, high gain satellite communications, and 5G networks. An example of the antenna can be seen in the fig. 2.6.

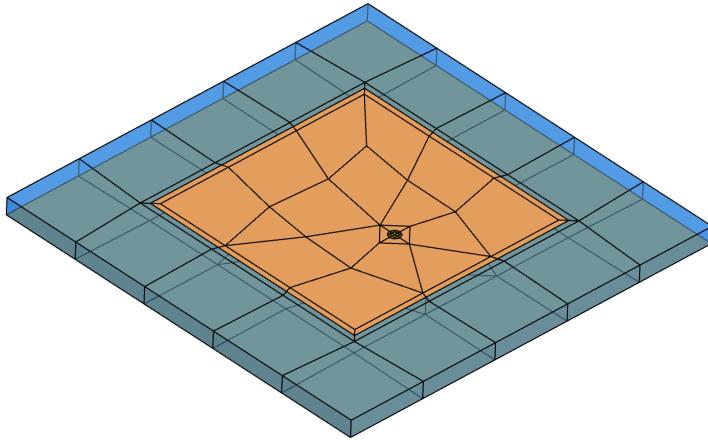
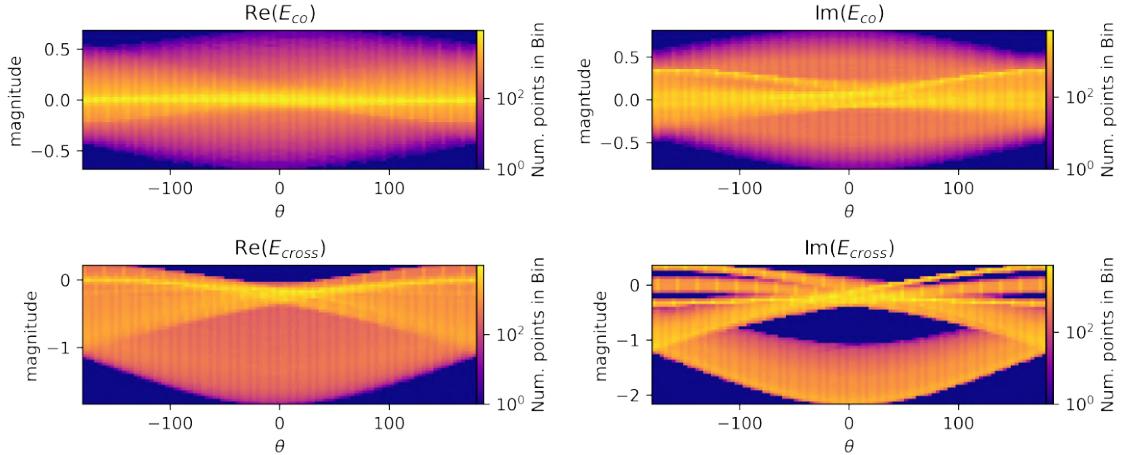


Figure 2.6: Patch antenna on dielectric substrate from TICRA Tools, with MOM mesh overlay

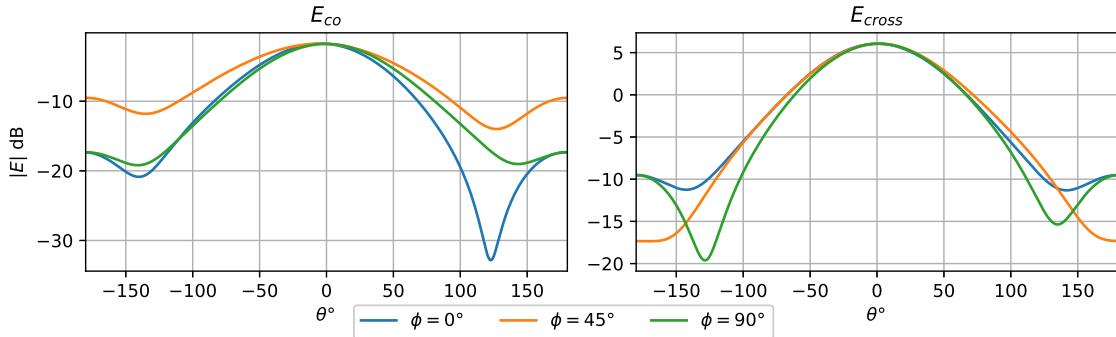
A Patch antenna is generally a low gain antenna with a single lobe and very well-understood radiation characteristics. In fig. 2.7, a 2D histogram of the entire antenna dataset is shown.

Figure 2.7: 2D Histogram of PATCH dataset, all ϕ -cuts contribute to the histogram



In fig. 2.8, we see an example of the patch antenna radiation pattern. It is often relatively low gain and does not have much variation in ϕ . Though significant field differences can be observed within the dataset as a function of how well the particular antenna parameterization radiates as a function of the design parameters.

Figure 2.8: PATCH Dataset magnitude example.



2.6.1 Design Parameter Description

The patch antenna is parameterized with three parameters: the X and Y position of the coaxial feeding point and the relative permittivity of the underlying substrate.

The coaxial feeding point is a small wire that is put in from the bottom of the antenna to excite the surface of the patch electrically. This feeding point is critical since it affects both the impedance⁶ and the antenna's polarization.

The *Coaxial x-placement* is known to induce different electromagnetic modes across the patch antenna, and using these degenerative modes are a very common method of designing the polarization of patch antennas[16]. Therefore the x-placement will have a known relationship with how much power is in the co and cross-polarization of the radiation patterns.

The *Coaxial y-placement* is also a well-known design parameter and is used to control the antenna's impedance to match the source's impedance, so all energy is radiated instead of being reflected back through the coaxial line. Therefore this parameter will generally have a strong relationship to the overall power radiated. [5, eq. 14.18b]

The *Substrate Relative Permittivity* is a material property of the *dielectric substrate* that the copper patch is placed on top of. This will affect the resonance frequency of the antenna, and since we are only modeling a single frequency, this will look like an impedance change, changing the amount of radiated power [5, eq 14.14-14.19]. In fig. 2.9, the effect on a single point of the far-field sphere can be seen as a function of the relative substrate permittivity. Similar plots of the rest of the parameters can be seen in appendix D.1.

⁶Having non correct impedance will result in reflections, lowering the overall radiated power of the antenna.

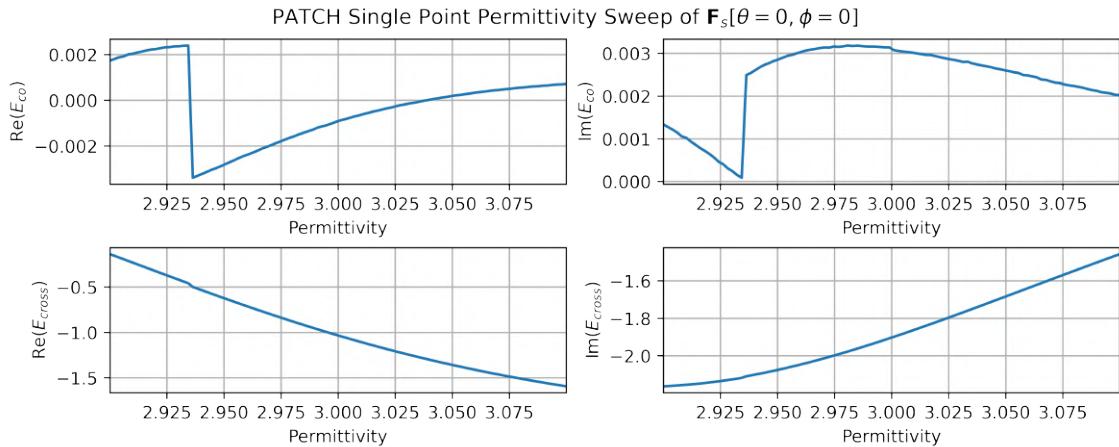


Figure 2.9: Patch single point on far-field sphere parameter sweep

2.7 Circular Horn With Dielectric Lense (CHA)

A horn antenna, fig. 2.10 is a type of *waveguide* antenna, where the antenna is excited at a narrow part by a circular waveguide, and the horn then slowly allows for the guided wave to expand and into free space and propagate. Precisely in the same way that an acoustic horn can direct sound waves. In this antenna scenario, a dielectric lens is put on top of the waveguide; this lens has specific dielectric properties that will diffract the electromagnetic wave.



Figure 2.10: Horn antenna model with dielectric dome, picture from TICRA Tools

The circular horn antenna is also a high gain antenna, which has a minimal ϕ variation because of its rotational symmetry. It is also highly co-polar, as seen in fig. 2.12, which might effectively reduce the dimensionality of prediction since almost half of the sample points will have values very close to zero.

Figure 2.11: 2D Histogram of CHA dataset, all ϕ -cuts contribute

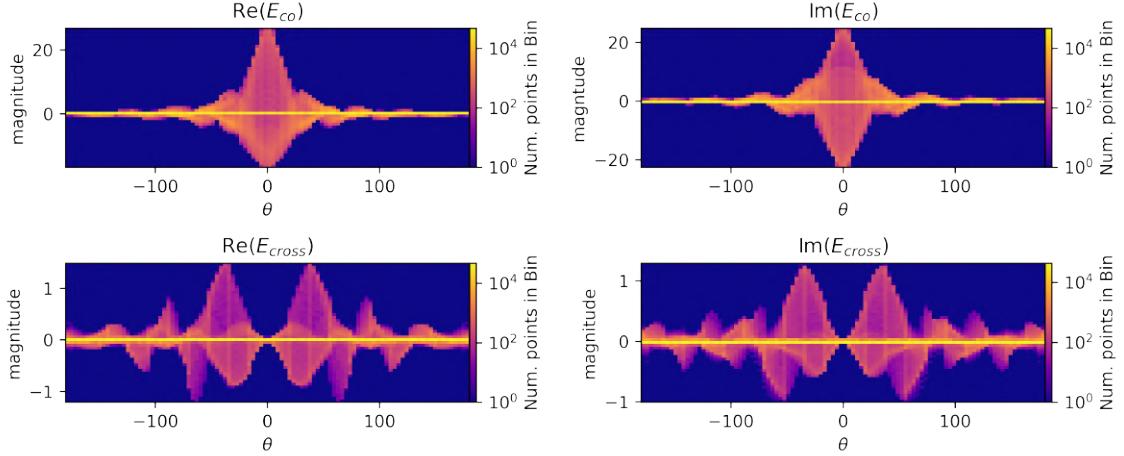
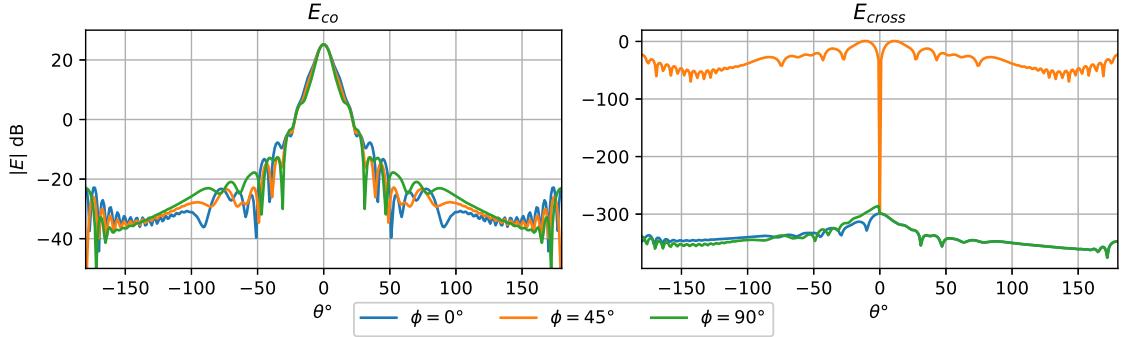


Figure 2.12: Example of co and cross-polar magnitude in CHA dataset



2.7.1 Design Parameter Description

In the circular horn case, we are changing the dielectric properties of the dielectric lens. The lens has two defining properties: relative permittivity and the loss tangent. The *relative permittivity*, ϵ_r is a factor multiplied on the free-space permittivity ϵ_0 such that for a given dielectric with permittivity ϵ the relative permittivity is $\epsilon = \epsilon_0 \epsilon_r$. The permittivity defines the relations between the electric field and electric flux density, see appendix A, which changes the propagation speed and boundary conditions at the interface between the lens and free space. This leads to diffractive and reflective behavior, changing the far-field pattern. The *loss tangent*, usually denoted $\tan \delta$, is a loss factor of a dielectric material. I.e., it is a constant that relates to how much power is lost by the electromagnetic field when propagating through the lens.

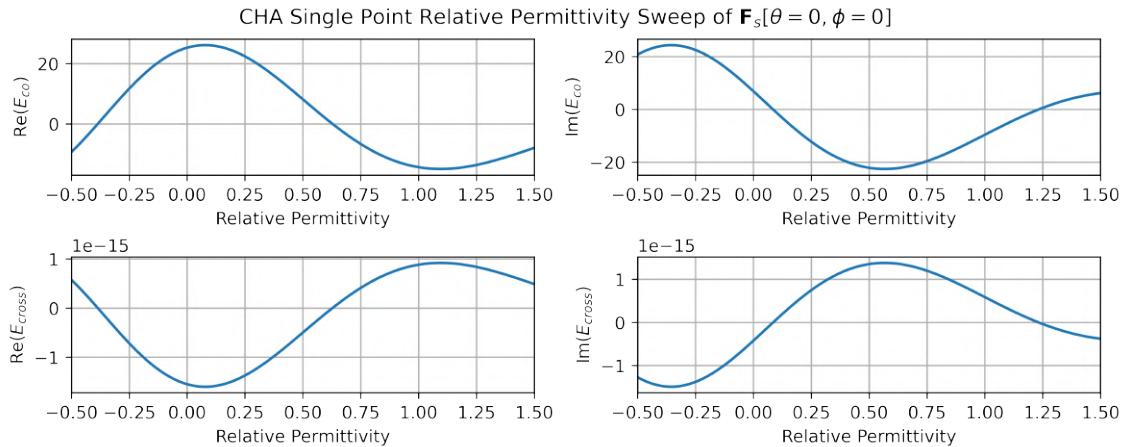


Figure 2.13: Effect of relative permittivity on a single point of the far-field sphere.

2.8 ML-Array (MLA)

The ML-Array antenna, seen in fig. 2.14, is a more complicated reflector style antenna. The ML-Array has five feed-antennas positioned at an oblique angle to the reflector. These five antennas then radiate separately unto the reflector and interfere, creating the overall observed radiation pattern in the far-field of the full antenna. In this case study, we only vary the phase and amplitude of the fields coming from the five antennas in the array seen in fig. 2.15.

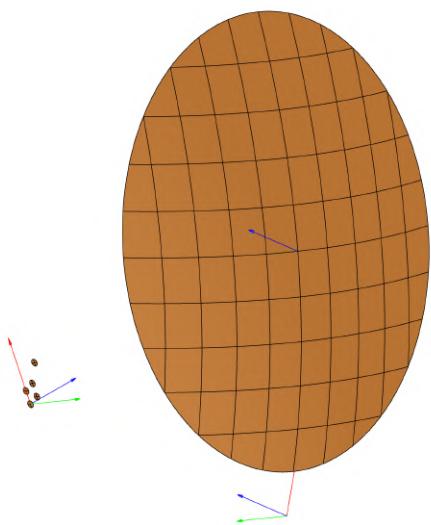


Figure 2.14: ML-Array with 5-element feed array radiating unto a parabolic reflector

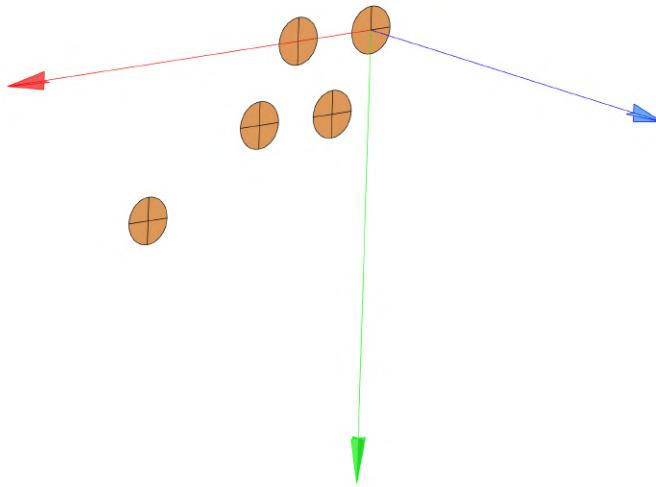


Figure 2.15: Closeup view of the 5-feed array that radiates unto the reflector

If multiple antennas are correctly positioned and are given the correct excitation, their far fields will add up constructively in specific directions, making their combined far-field more directive. This is the concept behind *antenna arrays*.

This can be utilized for both making higher gain antennas and configurable antennas that can be controlled electrically. By varying the amplitude and phase of the individual antenna excitations, the 'main lobe' of the antenna can be steered without mechanical parts.

Assuming identical elements in the antenna, with a single element gain of G_{SE} , the gain of an antenna array is then by.

$$G_{array}(\theta, \phi) = AF(\theta, \phi)G_{SE}(\theta, \phi) \quad (2.10)$$

Where AF is the *array factor*[5].

$$AF(\theta, \phi) = \sum_{M=1}^{N-1} A_m e^{j\varphi_m} e^{j(\beta \hat{r} \cdot r'_m)} \quad (2.11)$$

Where r'_m is the position of the m 'th antenna, $\hat{r} = \hat{a}_x \sin \theta \cos \phi + \hat{a}_y \sin \theta \sin \phi + \hat{a}_z \cos \theta$ is the unit vector pointing in the direction (θ, ϕ) , and A_m and φ_m are, respectively, the amplitude and phase excitation of the m 'th antenna. Thus is is quite simply a factor of constructive and destructive interference far away from the antenna.

The array factor assumes that the antennas do not interact and can often not be used in real-life scenarios. In this antenna dataset are therefore investigating the antennas in an 5-element array with a parameterization that only contains the excitation coefficients, A_m and φ_m .

Figure 2.16: 2D Histogram of MLA dataset, all θ -cuts contribute

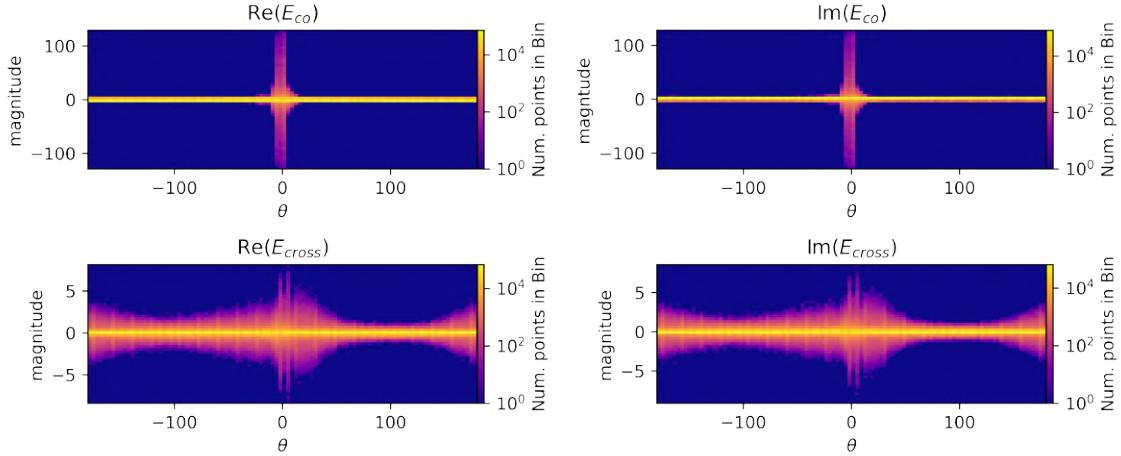
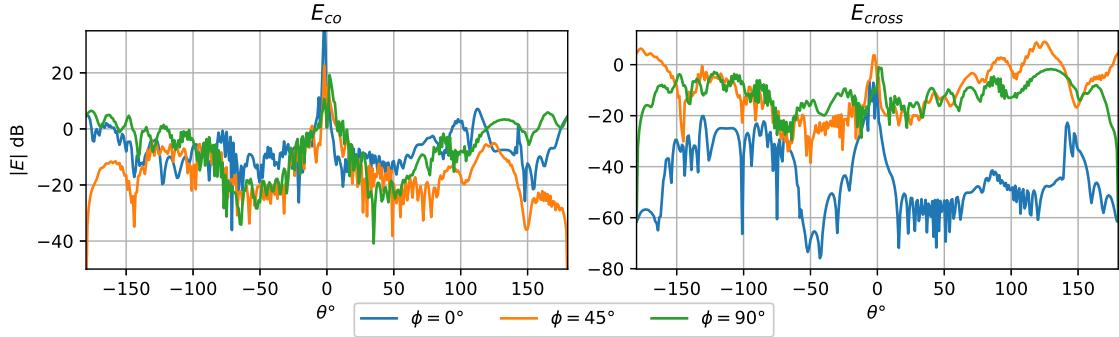


Figure 2.17: MLA dataset magnitude example



2.8.1 Design Parameter Description

The MLA dataset contains ten parameters which are all *excitation parameters*. Thus the parameterization does not control any physical dimensions or material properties, in contrast with all the other datasets, but instead expresses the amplitude and phase of the excitation signal applied to all five antennas that make out the array.

Each parameterization is thus made up of 5 normalised amplitudes $\in \{-30\text{dB}, 0\text{dB}\}$ and 5 phases between 0-360 degrees.

$$\chi_i = [A_1, A_2, A_3, A_4, A_5, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5]^T \quad (2.12)$$

Summary

In this section, we have described the mathematical form of the modeled electromagnetic far-field patterns. Then we discussed sampling methods in the far-field sphere and inside the parameter design space.

Furthermore, we have described the four datasets generated to train surrogate antenna models. RFLCT: is a reflector antenna with varying diameter, focal length, and offset. PATCH: a patch antenna dataset where the parameters control the coaxial feed point and dielectric properties of its substrate. CHA: a circular horn antenna with a dielectric microwave lens, where the dielectric lens's relative permittivity and loss tangent is altered.

MLA: which is based on a 5-element reflector-antenna-array setup with varying excitation parameters for each of the five antenna feeds.

Dataset	Num. of Param.	Num. Samples	Sample Shape			Notes
			θ	ϕ	Num. Comp.	
RFLCT	3	3600	361	3	4	LH
PATCH	3	3375	361	3	4	LS
CHA	2	4000	361	3	4	LH
MLA	10	10000	361	3	4	LH

Table 2.1: Summary of datasets used for training surrogate models. Number of components refer to the 4 real values that are used to represent the two orthogonal complex vectors. LH: Latin Hypercube, LS: Linear Spaced Sampling

3 Literature Review

This chapter will outline the field of surrogate modeling through a literary study. The goal is to give an overview of the general field of surrogate modeling such that the reader will be acquainted with standard methods and approaches usually employed in surrogate modeling. This will then provide the basis for the modeling and architecture choices made during this project and outline how this work aligns and how it differentiates itself from the rest of the antenna surrogate modeling literature.

Firstly we will create an overview of the general surrogate modeling definitions used in the literature, and then we will go into techniques used in surrogate modeling. We will then outline some of the recent results from general microwave surrogate models, which share many methods of antenna surrogates. Then we will focus on purely far-field prediction surrogates. Finally, cutting-edge surrogate modeling techniques such as physics-informed neural networks are discussed, and future applications for far-field surrogate models are outlined.

3.1 General Surrogate Modelling

Surrogate modeling, also called metamodeling, is a field within modeling and optimization used in many engineering disciplines to speed up repeated modeling and obtain knowledge about design domains. As also outlined in chapter 1, the idea is to model a ‘fine’ system with a response R_f . This fine system is defined as having a high precision in its response, but the response might be challenging to obtain or enormous if it had to be stored. Examples of fine models could be direct system measurements or high-precision numerical simulations.

The goal of any surrogate modeling is to obtain a *coarse response* R_c from the created surrogate model that is not as precise as the fine model but which, in some essential criteria, makes the model more usable than the fine model. Thus, the goal is to obtain a given fine model input χ within a design space Ω_D .

$$R_c(\chi) \approx R_f(\chi), \quad \chi \in \Omega_D \quad (3.1)$$

Thus surrogate modeling is fundamentally a sub-field of regression machine learning where the goal is not necessarily to obtain predictive models for unknown relations but to obtain surrogate models with the same predictive powers as an already known model.

One significant difference between this thesis’s objective and most microwave surrogate literature’s usual objectives is access to data. This thesis assumes large quantities of precomputed data are accessible to the model and are more interested in what precision is achievable within the given dataset. In contrast, many surrogate models are used for live optimization, where data is sparse and the evaluation time of the fine model, i.e., the access to data, is the limiting factor.

There are generally two schools of thought when generating surrogate models: data-driven and physics-driven modeling[12]. This thesis will entirely focus on data-driven models, but we will also quickly outline the concept of physics-driven models for completeness.

3.1.1 Physics driven models

Physics-driven models base themselves on some already known approximation with an analytically known model. This model could be a schematic or an equation modeling some system behavior [12]. Thus for physics-driven models, a coarse model R_c is given a priori, and the goal is then to obtain some alteration of the model that makes it fit the criteria in eq. (3.1).

A very common way of refining the physical models is the concept of *space mapping*[17]. In space mapping an a priori coarse model $R_c(\chi)$ is altered by mapping the parameterization through a mapping $P(\chi)$, such that $R_c(P(\chi)) \approx R_f(\chi)$

The idea of physics-driven modeling in microwave surrogates is, therefore, to use a physically derived model, such as a schematic with a coarse response R_c , and then to correct it towards the intended fine model response R_f through some modification, such as a response mapping, or alteration of the physically derived model[18]. This is widely used for microwave circuits and simple antenna models since schematic approximations are a mature approximation method. For more advanced antenna models, often the coarse model is simply a lower density mesh simulation, which is then space-mapped to the response of a higher mesh-density simulation [19].

3.1.2 Data-driven models

Data-driven models are surrogate models where modern statistical tools and machine learning is used to infer relations about the behavior of the response of the fine model R_f such that a coarse model R_c is obtained[6]. Thus, it is significantly related to the usual machine learning objective of having a set of inputs and outputs and then minimizing some criteria such that the coarse model can obtain predictive powers, making it a supervised regression task. The goal of a data-driven model is to obtain:

$$\operatorname{argmin} |R_f(\chi) - R_c(\chi)| \quad \text{for } \chi \in \Omega_D \quad (3.2)$$

Here the choice of model, training method, and dataset design is, of course, paramount to the success of the method. The modern methods usually employed in data-driven surrogate literature include radial basis functions, gaussian process regression, support vector regression, polynomial chaos expansions, and neural networks [12].

3.2 Microwave Surrogate Modelling

Microwave engineering sets itself apart from other electrical engineering disciplines because of the highly complicated relationships that arise in high-frequency electronics. Most assumptions made for low-frequency electrical engineering design break down, and a large set of frequency-dependent parasitic effects make even the most simple circuits have either narrow operating assumptions or no closed-form expressions. Therefore numerical modeling is the go-to standard in modern microwave engineering.

The most common design task for microwave engineers utilizing numerical simulations are parametric optimization, where the values of selected design variables are varied to improve the system performance and statistical design (e.g., Monte Carlo analysis and uncertainty quantification) [12]. This has created a significant incentive to make faster numerical models that can execute quickly such that microwave components, and entire microwave schematics, can be rapidly evaluated and have quick design iterations.

Since microwave engineering had its roots before modern computational methods, many analytical models have been built up using linear approximations of nonlinear relationships or electrical schematic models. These models are often quite precise within certain

assumptions and give good intuition. Still, in modern practical designs, the models are often frequently shifted or altered due to parasitic effects¹.

However, these imprecise models still allow physics-driven surrogate techniques to be widely used in microwave surrogate modeling[19–21], and antenna models have also been optimized by using simplified antenna models or equivalent circuit models as a priori. Antenna circuit models are only good for developing *S*-parameter predictive surrogate models, such as in [22] and far-field models have also been obtained using combinations of infinitesimal Hertzian dipoles [23], just like the one derived in appendix C.

3.3 Antenna Surrogate Modelling

Antenna engineering is a sub-field of microwave engineering that is significantly driven by numerical simulations, as evident by a large number of modeling tools such as ADS, HFFS, CST, and Ticra Tools. In many cases, numerical antenna modeling and optimization are the only practical method for obtaining reliable designs for high-requirement modern antenna solutions.

Most of the literature on surrogate antenna modeling defines the model response from the so-called scattering parameters. Scattering parameters are measurements of reflections and transmissions inside a 'black box' microwave circuit such as an antenna. Therefore, by measuring the amount of energy reflected back into the feed from the antenna when an excitation voltage is applied, one can approximate the amount of energy that is actually radiated and, therefore how good the antenna is. An excellent and detailed description of scattering parameters can be found in [24]. This is sufficient when making surrogates in a design space where the far-field radiation is relatively well known and where the real difficulty is to obtain a model that is broadband and radiates at the correct center frequency, which is often the design goal for lower-requirement antenna systems.

In [25] A 'pyramidal' neural network predicts the achieved gain and reflection coefficient S_{11} of three different printed circuit board antennas. These antennas are optimized using the generated surrogate models and achieve a relative RMSE performance of 2.2 percent. They also show that the neural network outperforms Gaussian processes and ensemble learning methods, especially at small dataset sizes. The neural net architecture is parameterized to achieve a low dimensional hyperparameter tuning; this hyperparameter parameterization is then tuned using the k-folds method using Bayesian optimization to find the optimal hyperparameters. The final architecture is then chosen and trained on the whole training dataset.

They also take advantage of a method to constrain the design space into a region of useful designs, effectively making the model a 'Pareto-front' surrogate model that only has to learn the 'useful' design-manifold instead of the entire design space.

In [19], low fidelity, low-mesh simulations are used as a coarse response surrogate, where space mapping is then applied to optimize the low-mesh simulation response to be in line with a high mesh simulation.

Especially models that are way out of regular design consideration can benefit from surrogate modelling. The S_{11} -prediction surrogate of a deforming antenna is done in [26, 27], based on multiphysics simulations, with polynomial chaos expansions being used for generating the surrogate.

¹"a parasitic effect is an unwanted electrical characteristic that is possessed by an electrical component. An example would be a resistor that has parasitic capacitance.

3.4 Far-field modeling

A particular case of microwave modeling and antenna modeling is antenna far-field prediction. Here the response of a fine and coarse model is an electromagnetic far-field on the form outlined in appendix A.1.5

$$R_f = \mathbf{F}(\theta, \phi) \quad (3.3)$$

This thesis aims to create far-field surrogates partially because it is quite an unexplored area since most antenna surrogate literature only predicts reflections at the input terminal of the antenna. Some interest in far-field surrogate models has come from the desire to obtain good, fast, and low-memory models of the radiation pattern of array beamforming, which is an enabling technology for cutting-edge communications technologies such as advanced satellite swarms, and 5G. This section will outline some of the main papers that have done far-field surrogate modeling and the methods they have applied.

3.4.1 Array Beamforming

As noted, a large part of the field-prediction literature comes from fast and accurate field predictions needed for next-generation communications networks such as 5G and other mm-Wave technologies. Since at these frequencies, large arrays of antennas need to be able to quickly find and identify connection targets through beamforming, a technology where the excitation of large antenna arrays is altered to change the directionality of the antenna beam towards the intended recipient. This requires the antenna system to infer far-fields from antenna excitations and, inversely, antenna excitations from desired far-fields in real-time and with high precision. Here, machine learning methods' low and stable computational cost has significant advantages over traditional beamforming schemes, often driven by iterative optimization that can be quite costly in high-frequency beam-switching[28].

In[29], a feed-forward neural net was trained to predict far-field patterns from excitation states of a 1×8 patch antenna array. They used a five-layer feed-forward neural network to predict directly from antenna excitation to field pattern. In parallel, they also trained a network for the inverse problem of predicting excitation states from field data. They used a single θ -cut with 360 mesh points and 16 design parameters with phase and magnitude for each patch antenna. They achieved a root mean squared error of their field pattern of around $2 \cdot 10^{-2}$.

Antenna surrogates are also used for planning communication schemes. In [30] A convolutional neural network is used to predict a set of parameters that define the beamforming scheme used in different cases of user connection. They showed that a highly effective neural network could be trained, which is precise and significantly faster than conventional iterative solvers. In [31] a convolutional neural network is used to predict the beam pattern for correct communication between two antenna systems in urban environments.

A convolutional network method for estimating the required phase-excitation of patch antennas in an antenna array to synthesize a given pattern is proposed in [32]. They use purely supervised learning and compare far-fields from the numerical solver ANSYS HFSS. They used a dataset of 205,000 far-field samples with a train/validation split of 0.78. They used both datasets and batch normalization and leaky ReLu activation.

Some surrogate models have also been used to obtain field expressions for individual elements of models that have been combined into a larger array structure. In [33] the reflection coefficient of reflectarray elements that make up a reflectarray antenna are modeled

². Here they have used a two-layer feed-forward neural network to predict both the co and cross-polar reflection coefficient of the elements as a function of the incidence angle, the geometry, and the frequency of excitation. They achieve decent results when looking at the obtained far field but do not provide any error metric that can be easily compared.

Another study of reflectarray elements modeling is done in [34, 35], here they use support vector regression models to model the reflection coefficients and achieve very precise results with a relRMSE of around 1% when modeling the far-field from the entire reflectarray antenna.

Paper	Method	Case	Notes
[29]	FC-NN	Field Beamforming	ReLU, Mesh-grid
[26]	PCE	Deformed Antenna FFP	
[27]	PCE	Deforming Antennas S_{11} prediction	Adaptive
[32]	CNN	Field Beamforming	LReLU, grid-phase prediction
[31]	CNN	Beam Pattern Inference	
[30]	CNN	Beamforming Schemes	AF: Sigmoid, TL
[33]	FC-NN	Reflectarray element modelling	Cont.-Space, AF: Tanh
[34, 35]	SVR	Reflectarray Reflection coefficient prediction	
[25]	FC-NN	S_{11} and gain prediction	LReLU, mesh-grid, f -dep.

Table 3.1: Comparison of methods in recent microwave surrogate literature. NN: Neural Network, AF: Activation Function, FC: Fully Connected NN, CNN: Convolutional Neural Network, PCE: Polynomial Chaos Expansion, SVR: Support Vector Regression, ReLu: Rectified Linear Activation Unit, LReLU: LeakyReLU, TL: Transfer Learning, FFP: Far Field Prediction. f -dep.: Frequency dependant problem

3.4.2 Inverse Radiation Problems

Even though obtaining field solutions from parameterized antennas is the usual way, and therefore considered the forward problem. The opposite approach is also very interesting, considering the ability to make antenna design directly from the chosen field configuration instead of searching a design space for a suitable field pattern. In [36] a solution to the inverse loop radiation problem is proposed using artificial neural networks doing regression from simulated electric field intensity of a single ϕ -cut to predict the diameter of a loop antenna. The chosen configuration was a 4-layer 'even' feed-forward network with a theta sample for every input neuron and a single output neuron predicting the diameter. They showed very high accuracy, which is promising, but it is also one of the simplest antennas known. In [37], a deep, fully connected deep neural network is used to infer the required antenna excitation of a patch array directly from a mesh-grid radiation pattern. Though they only used the magnitude values of the far-field and a single θ -cut. They showed that the trained model was in good agreement with the simulation models.

²reflectarray antenna are arrays of printed copper patches that are radiated unto by a source, in a similar fashion to a parabolic reflector. An equivalent phase shift can then be constructed by altering the individual patch dimensions, making them effectively behave like parabolic antennas while being able to have flat shapes. Therefore modeling the individual patches is essential.

Summary

In this section we have attempted to broadly outline the field of modern surrogate modelling, focusing on microwave and specifically antenna surrogate modelling.

For the specific case of far-field surrogate modelling from antenna parameterization, only a few examples can be found in the literature, and most of these are concerned with predicting the far fields of antenna arrays from excitation parameters due their proposed use in 5G beamforming schemes. For this purpose convolutional and fully connected nets are often used. Gaussian processes and polynomial chaos expansions are more common in the general microwave surrogate literature.

A more recent development of in surrogate models, are physics informed neural networks (PINNs). Since PINNs have ended up being out of scope of the rest of the work, we have chosen to move this section. For the interested reader, the section can be found in appendix E.

4 Methods

In this chapter, we will go over the methods that have been applied to train far-field surrogates and methods used for dimensionality reduction. We will not go into large detail about specific well-known methods but will include brief method descriptions when important to the results, or when necessary for completeness.

Firstly a description and discussion on neural network architectures will be done, then gaussian processes regressions and their application in this thesis will be introduced. We will then turn our attention to the investigated latent space methods, such as PCA, autoencoders, and variational autoencoders, and give a general overview of them as methods and their expected contributions to the results. Finally, we will discuss the combination of the prediction and latent space models.

4.1 Neural Networks

In this section we will discuss the methods, architectures, training schemes of all the neural networks that have been implemented in this project. The Python deep learning framework, PyTorch[38], has been the basis of all the neural networks. SciKit-Learn implementations of fully connected deep neural nets where also used, but quite early in the project GPU acceleration was utilized for training the neural networks, which SciKit-Learn does not support. All the networks where trained on clusters of the DTU High Power Computer using NVIDIA Tesla V100 graphics cards.

4.1.1 Neural Net Far-Field Representation

One of the most critical questions for neural surrogate modeling is the architecture of the output. There are two main questions at hand. Firstly how to express the complex value of both the co and cross-polar component and secondly on whether the far-field prediction should be conducted as a mesh-grid tensor or through the introduction of spatial values as input parameter, making the output a continuous spherical function.

Regarding the complex representation, one method is to express the complex values as in Euclidean space such that $\mathbf{F} \in \mathbb{C}^2 \in \mathbb{R}^4$, and then express each sample on the far field sphere by the combination of four output neurons. Another method is to use the built-in PyTorch complex tensor. This package deals with loss and backpropagation by using the complex Wirtinger derivative[39]. According to PyTorch, this package should be more optimized for complex representations, though it was not compatible with all operations that wanted to be used during this project and is still in a beta feature, so the standard euclidean representation was chosen. A final method is letting the neural net represent the complex vector field in polar form with the magnitude and phase of each component. This seems arbitrary, especially since it will force half of the output neurons to take on a phase relationship, which will be intrinsically periodic. Unless this is explicitly introduced by running the output through trigonometric functions, it will make the individual dimensional regressions problems more detached. Therefore, we have chosen to use a real and complex output neuron for each point's co and cross-polar value on the far field sphere.

Now that a method for expressing each sample point in \mathbb{C}^2 has been established, the next question is how to properly express the combined sphere of all NM amounts of \mathbb{C}^2 samples. The following two methods have been considered:

In fig. 4.1 the mesh-grid approach is seen, here we model each point on the sampled far-field sphere through the output of four neurons, thereby representing the field as a mesh-grid of sampled field-points on a sphere. The advantage of this is that we have neurons that are fixed to only represent a single point on the far field sphere, which one could imagine would lead to less variance needed in the individual outputs. The downside is the large number of outputs needed, and the fact that they do not have any implicit way of interpolating between them.¹

In fig. 4.2 the continuous space approach is seen. Here we use ϕ and θ as design parameters, along with the rest of χ . This has the advantage that the neural net will have to learn a continuous representation of the prediction space since θ and ϕ can take any values within a sphere. Effectively making the prediction and training invariant to far-field sampling density, and non-full sphere sampling and is able to make predictions with arbitrary sampling density. This is the usual approach that is taken in physics-informed neural networks, where a set of test points, called collocation points, can easily be generated and evaluated with the physics-informed loss function.

The disadvantage of the continuous space representation is that it will be considerably more inhibited by the curse of dimensionality at the input through the expansion of input parameters per epoch from χ to $[\chi, \theta, \phi]^T$,

Both approaches have been used in antenna surrogates [25, 33], but as of this writing, there are no good structured comparisons of these two architectures in the SciML literature that we have been able to locate. We have chosen to use the mesh-grid approach because of the worry of the input side curse of dimensionality that would seem to make the training significantly slower and possible performance as well, compared with the mesh-grid approach.

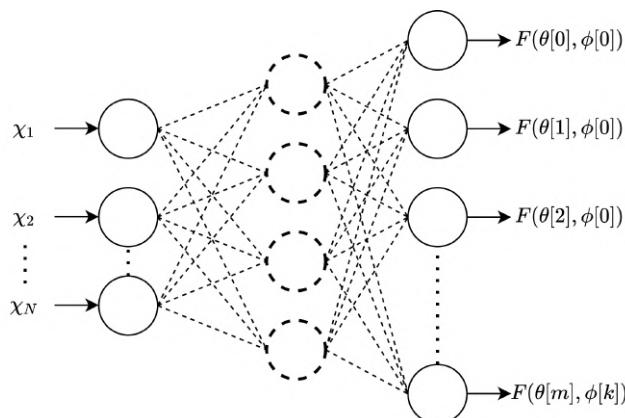


Figure 4.1: Mesh grid neural net representations, each output neuron represents four neurons to predict both real and imaginary parts of the co and cross-polar component of the field at each sample point.

¹It is possible to interpolate post-prediction if the mesh-grid is fine enough to catch all 'lower-frequency' details of the far-field pattern. In the same manner the Nyquist–Shannon sampling theorem exists for up-sampling of time-series signals.

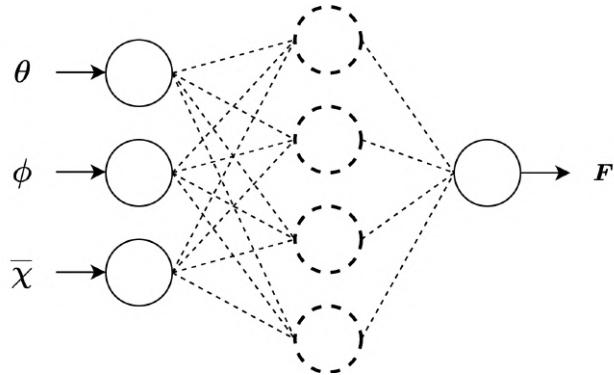


Figure 4.2: Continuous space neural net representation, predicting directly into both complex components

4.1.2 Pyramidal Deep Neural Networks

The recent paper [25, Mar. 2022], formulated a systematic approach to making fully connected 'pyramidal' neural networks (PDNN) from an antenna parameterization to a response prediction. The idea behind the pyramidal structure is to make the network architecture able to distill the parameter-field relations by first expanding the low-dimensional parameterization to a very high dimensional space, and then distilling it down to the high dimensional output. The PDNN architecture is defined entirely from four parameters, which also makes hyperparameter tuning simpler:

- φ_K – Neurons in the last layer
- s_c – Scaling factor
- k – Number of layers
- α – Leakage ratio of the leaky rectified linear unit (ReLU) activation function.

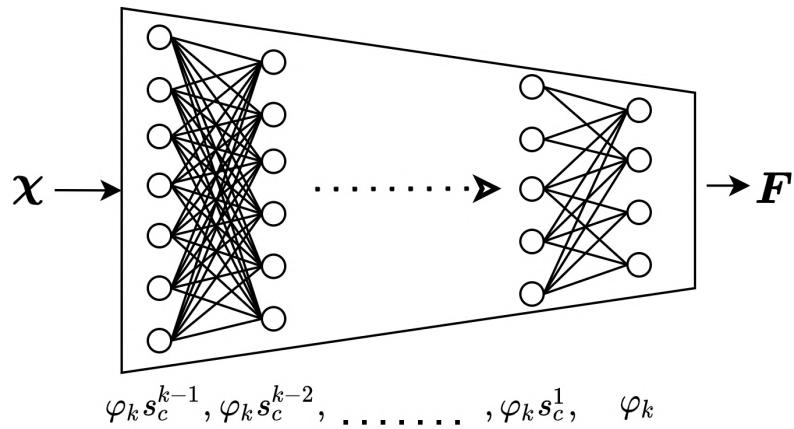


Figure 4.3: Pyramidal neural network architecture for direct parameter, to field prediction. $\chi \in \Omega_D \in \mathbb{R}^D$ and the sampled far field $F \in \mathbb{R}^{NM}$ with $D < NM < \varphi_k s_c^{k-1}$, where N is the number of θ samples and M is the number of ϕ samples.

Such that the number of neurons in i 'th layer can be described as:

$$\varphi_i = \lceil \varphi_k s_c^{k-i} \rceil \quad \text{for } i = 1, 2, \dots, k \quad (4.1)$$

With each layer having the Leaky ReLU activation function:

$$\sigma_\alpha(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \quad (4.2)$$

Which has the advantages over the regular ReLU activation function of having no zero-gradient areas, that usually can lead to 'dead' neurons, and also has the ability to change the 'amount' of non-linearity of the activation function. As shall be seen in the results, chapter 5, relatively high values of α were found to be advantageous in the obtained surrogate models.

This has been the basis for all direct parameter-to-field prediction neural networks.

4.1.3 Convolutional Neural Networks

Image analysis has improved significantly in the last decade through deep convolutional neural network architectures, which are much better at handling the high dimensional problems associated with modern RGB-colored pictures. They are especially good at dimensionality reduction problems where spatial features are present that can be represented within the convolutional kernels[40].

Fundamentally convolutional neural networks are sparsely connected neural networks with a connection pattern that is learned by the convolutional kernels. Usually, image analysis problems are formulated as *channels*, *height* \times *width* problems, where the height and width are the number of pixels in the picture and the channels are the number of colors, usually 3 for RGB colors. When oriented in a tensor format, the far-field data in this thesis is very similar to RGB pictures since we have samples in θ and ϕ , where each sample has two co-polar and two cross-polar values. Also, since the far-field patterns are continuous functions, strong spatial correlation must be present in the datasets. Therefore the appropriation of CNN methods for unsupervised methods such as autoencoders has seemed like an obvious choice.

Convolutional neural networks have been used for both variational and regular autoencoder architectures. Generally, stride has been used for dimensionality reduction instead of max-pooling, and a channel-doubling-per-layer scheme inspired by deep image recognition convolutional nets like the VGG-x nets[41], has been adopted.

4.1.4 Smoothing loss functions

The derivation and application of PINN loss functions, see appendix E, for antenna surrogates, are outside of the scope of this thesis. However, some regularization factors have been used in training neural network surrogate models to obtain solutions more aligned with known far-field patterns.

One of these is a simple regularization for forcing smooth network outputs, so for output from the network of \mathbf{F}' following loss factor is added:

$$\mathcal{L}_{diff} = \beta \int_0^{2\pi} \int_0^\pi \left| \frac{\partial \mathbf{F}'}{\partial \theta} \right| d\theta d\phi \quad (4.3)$$

Where we have only penalized θ variance because most of the erratic behavior has been observed here. In practice section 4.1.4 has been calculated in discretized form as:

$$\mathcal{L}_{diff} = \beta \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \|\mathbf{F}[\theta_i, \phi_j] - \mathbf{F}[\theta_{i+1}, \phi_j]\|_1 \quad (4.4)$$

For high β this will, of course, make the network output a constant, though when β is tuned correctly it has been used to lessen some of the erratic behavior in the output that is not penalized highly by mean-squared proportional loss.

Optimizers

All models have been trained using the minibatched ADAM gradient descent as implemented in PyTorch[42], since it is generally considered to be more robust than standard stochastic gradient descent and is designed for machine learning problems with large datasets and/or high-dimensional parameter spaces[43].

4.1.5 Regularization Strategies

A slight amount of L2-Weight Decay has been used on all the models. Generally it has also been observed that small batch sizes have a good regularization effect on some of the datasets. Early stopping has been used based on the lowest validation error by progressively saving the model with the lowest validation error. Models were then verified to have the correct validation performance by verifying it with their test performance. Though in practice this never resulted in a model getting discarded. No, more aggressive, regularization strategies were chosen. Since for most cases there was not any real significant divergence between validation and training error.

Common layers like batch normalization, and dropout were not used, since they did not improve performance when tested, and since for most of the converging models, overfitting was not a huge issue.

Hyperparameter Tuning

The hyperparameters of the neural network models have been tuned through a Bayesian optimization scheme using the Weights And Biases sweep-agent API [44, 45]. Here the tunable parameters were chosen to be the four parameters defining the PDNN network as described in section 4.1.2, plus batch-size and base learning rate. It was found that many of the networks seem to find better minima at the end of training if an aggressive learning-rate scheduler was applied. Therefore all optimizers had their learning rate reduced by order of magnitude, one or two times in the last quarter of training. This method has been used to avoid decaying the learning rate too early, which might happen with methods such as exponential learning rate decay. Thus, long training with a single learning rate is achieved, and the converged minima is optimized with low learning rates.

Manual hyperparameter tuning was then often conducted based on the best candidates found. All runs, with hyperparameters, losses, architectures and more can be found on the used Weights and Biases profile²

4.2 Gaussian Process Regression

Gaussian process regression, also called Kriging models or Wiener–Kolmogorov prediction, is a regression method to build upon the idea that collections of random variables have joint gaussian distributions [46]. Instead of predicting scalar values, gaussian process regressions predicts probability distributions. A practical feature of these Gaussian processes, just like Gaussian distributions, is that they can be entirely defined from a mean function $m(\mathbf{x})$ and covariance or kernel function $k(\mathbf{x}, \mathbf{x}')$.

²https://wandb.ai/skoogy_dan/projects

The advantage of predicting gaussian distributions is that the mean of these Gaussian distributions can be interpreted as the prediction of the gaussian process, and the variance of the distribution can be interpreted as the model uncertainty. Thus we implicitly have both a prediction and uncertainty quantification built into the model, which is all defined only using the kernel and the prior dataset.

For a regression problem we have a set of observed values $y_n \in \mathbf{y}$ with a noise variable ϵ_n .

$$t_n = y_n + \epsilon_n \quad (4.5)$$

Usually this noise process is assumed Gaussian, i.e $p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1})$, making the multivariate gaussian:

$$p(\mathbf{t}|\mathbf{y}) = \mathcal{N}(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}) \quad (4.6)$$

The marginal distribution of \mathbf{t} can then be shown to be [47]

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{y})p(\mathbf{y})d\mathbf{y} = \mathcal{N}(\mathbf{t}|0, \bar{\mathbf{C}}) \quad (4.7)$$

From the kernel function this *covariance matrix*, $\bar{\mathbf{C}}$ can be constructed from the datasets predictor values $x, x' \in \mathbf{x}$

$$C(x, x') = k(x, x') + \beta^{-1}\delta_{nm} \quad (4.8)$$

Then for an unseen prediction value, x_{N+1} , the marginal distribution for t_{N+1} will be

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1}|0, \bar{\mathbf{C}}_{N+1}) \quad (4.9)$$

With the new covariance matrix

$$C_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \quad (4.10)$$

Thus $p(\mathbf{t}_{N+1}|\mathbf{y})$ is a gaussian distribution with mean and covariance function

$$m(x_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t} \quad (4.11)$$

$$\sigma^2(x_{N+1}) = k(x_{N+1}, x_{N+1}) - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \quad (4.12)$$

Where \mathbf{k} is a vector made from the kernel values $k(x_n, x_{N+1})$ for $n = 1, \dots, N$

Which are the two main relations used for predicting the expected Gaussian distribution.

4.2.1 Kernel Selection

One of the main difficulties of using Gaussian process regression is to choose the kernel function $k(x, x')$ that defines the covariance matrix $\bar{\mathbf{C}}_N$.

A detailed description of many useful kernels can be found in [46, Chap. 4] and many of the practical kernel selection methods used have been adapted from [48, Chap. 2].

We have chosen to focus on some simple kernel configurations build upon two of the standard kernels.

Radial Basis Function (RBF) kernel

$$k(x, x') = \sigma_0^2 \exp\left(-\frac{d(x, x')^2}{2\ell^2}\right) \quad (4.13)$$

Where $d(\cdot, \cdot)$ is the euclidean distance between two observations. and ℓ is the *length scale* of the kernel and σ_0^2 is the variance scale constant.

The Matérn Kernel

$$k(x, x') = \frac{\sigma_0^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{\ell} d(x, x') \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{\ell} d(x, x') \right) \quad (4.14)$$

Where $K_\nu(\cdot)$ is the modified Bessel function of the second kind and $\Gamma(\cdot)$ is the gamma function. The Matérn kernel is a generalization of the RBF kernel eq. (4.13) with an extra smoothness hyperparameter ν . At $\nu \rightarrow \infty$ the Matérn kernel converges towards the RBF kernel.

The Periodic Kernel (Exponential Sine Squared)

$$k(x_i, x_j) = \exp \left(-\frac{2 \sin^2(\pi d(x_i, x_j)/p)}{l^2} \right) \quad (4.15)$$

The periodic, or exponential-sine-squared kernel has also been considered, since it seems like lots of the parameter-field relations has a high amount of periodicity, as can be seen in plot from chapter 2 like fig. 2.13 and fig. 2.5. Though all attempts at applying the periodic kernel led to numerically unstable results, even with higher values of β^{-1} as in eq. (4.8), which should help ensure that the covariance matrix is constructed as a positive definite matrix[49].

For the implementations of kernels and Gaussian processes regression itself, scikit-learn[50] has been used for single-output GPs and GPyTorch [51] have been used for attempts are multi-output GPs.

The most common way to tune the hyperparameters in the kernel functions is through gradient descent maximization of the log-likelihood function $\ln p(\mathbf{y}|\boldsymbol{\theta})$ [47].

$$\ln p(\mathbf{y}|\boldsymbol{\theta}) = -\frac{1}{2} \ln |\mathbf{C}_N| - \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \mathbf{t} - \frac{N}{2} \ln(2\pi) \quad (4.16)$$

The concrete implementation in scikit-learn led to extreme overfitting for all tried kernels since there is no easy way to include a validation evaluation of eq. (4.16) in scikit-learn. So instead, hyperparameter grid searches with holdout validation were used as an optimization strategy since the computation time was not prohibitive in the used datasets.

4.2.2 Multi-Output Gaussian Process Regression

Multi-output Gaussian process regression (MOGP), often called co-kriging in surrogate literature, is a quite non-trivial problem and is a current research topic. [52, 53]

Ideally, a MOGP should be able to infer relations between the outputs and relate them through a combined covariance matrix. This requires an intelligent scheme for covariance matrix construction since a naive implementation would need a large covariance matrix to whose inversion is a computationally intractable task for large and high dimensional datasets³.

³For a dataset of size N , the matrix inversion in eq. (4.11) and eq. (4.12) is the limiting time complexity of $\mathcal{O}(N^3)$, thus for a SOGP predicting into D dimensions the time complexity is $\mathcal{O}(N^3 D)$. However, for a naïve MOGP the time complexity is $\mathcal{O}(N^3 D^3)$

We have chosen to use an output-output independent method, often called a Single Output Gaussian Process (SOGP), where every output dimension is modeled as an individual regression problem from parameter space. This implicitly assumes that the outputs are independent, which is not strictly correct but seems to give good predictive powers still, as shall be seen in chapter 5.

Though MOGPs are outside of the scope of this project, their application and practicality could be significantly increased if efficient dimensionality reduction methods could be applied to sampled far-field datasets, which leads us to the next section.

4.3 Latent Space Methods

For highly sampled datasets such as densely sampled far-field patterns, it is possible that the dimensionality inside of the dataset is artificially high and that the real relation to be understood can be expressed in a significantly lower dimensionality such that this lower dimensionality *subspace* contains the variance of the dataset. Furthermore, since the method generating the data variance comes from the relation between the far-field and a sampled continuous parameter space, one could imagine that inside this latent space, the samples will be projected unto a low dimensional manifold by some nonlinear projection of the design space. In other words, samples of our far-field datasets might be expressed as samples on a low dimensional manifold by some projection onto our high dimensionality space. This would make sampled far-field datasets a confirming example of the manifold hypothesis [54]

This is not a very well-studied phenomenon in surrogate antenna modeling. Therefore a considerable part of the work in this project has been conducted in order to try to investigate if the far-field datasets really can be expressed in small latent spaces, if they contain latent manifolds, and to empirically study what properties we can expect in the latent spaces and latent manifolds.

Another critical aspect is not only to understand the latent manifold but also to determine a projection back to the high-dimensional far-field. i.e., not only to obtain an encoder that projects the datasets unto a manifold but also to have a decoder that can project manifold samples into a sampled far-field pattern. This decoder should ultimately be able to create representative far-fields that have never been observed in the dataset, thus making the decoder *generative*. Therefore most manifold learning algorithms, such as TSNE or Spectral Embedding, which do not have a practical inverse transformation, have been discarded as options.

Lastly, obtaining a projection between design space Ω_d into the latent space, Z , would create a semi-supervised end-to-end learning method that would allow for a significant increase in ML interpretability and hopefully could outperform direct methods by splitting the task into more learnable sub-tasks.

Thus if we have a far field \mathbf{F} , latent space Z , and an antenna parameterization $\chi \in \Omega_d$ then we would like to obtain a encoder \mathcal{E}_{θ_H} , decoder \mathcal{D}_{ϕ_H} and parameter projection \mathcal{P}_{φ_H} with parameters $\theta_H, \phi_H, \varphi_H$ respectively.

$$\mathcal{E}_{\theta_H} : \mathbf{F} \rightarrow Z \quad (4.17)$$

$$\mathcal{D}_{\phi_H} : Z \rightarrow \mathbf{F} \quad (4.18)$$

$$\mathcal{P}_{\varphi_H} : \Omega_D \rightarrow Z \quad (4.19)$$

With the goal of obtaining a generated field reconstruction \mathbf{F}' from parameters with

$$\mathbf{F}' = \mathcal{D}(\mathcal{E}(\mathbf{F})) \quad (4.20)$$

By a minimization of the error between the original and reconstructed field.

$$\underset{\theta_H, \phi_H}{\operatorname{argmin}} \|\mathbf{F} - \mathbf{F}'\| \quad (4.21)$$

Furthermore, by then obtaining some parameter projection between design space and latent space.

$$\mathbf{F}' = \mathcal{D}(\mathcal{P}(\chi)) \quad (4.22)$$

We might be able to obtain a semi-supervised model that not only can express samples of the far-field datasets in lower dimensionality but also that can then understand the relationship between this latent space and the design space

In the following text, we will outline the models used and show how they have been applied during this project.

4.3.1 Principal Component Analysis

Principal Component analysis is a linear projection of a dataset unto an orthonormal subspace that expresses the maximum variance possible of the dataset within that subspace, ordering the subspace dimensions by the amount of variance contained within each of them[47].

Principal component analysis is widely used, and because of its linear decomposition, it will provide the baseline for our latent space models and our understanding of the latent spaces of the sampled far-field datasets.

In this project, we have used principal component analysis to obtain latent spaces of the far-field patterns $\mathbf{F}_s[\theta, \phi]$. We have used an SVD-based SciKit-Learn implementation of principal component analysis for all PCA-decompositions in this thesis[55].

4.3.2 Autoencoders

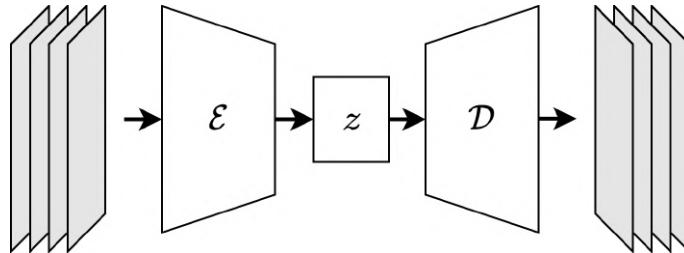


Figure 4.4: Autoencoder architecture. The input is fed through a deep neural trapezoidal neural network, \mathcal{E} , down into the latent bottleneck, z , from which the decoder, \mathcal{D} then is trained to reconstruct the input.

Originally introduced in [56], autoencoders is an unsupervised neural network model that makes latent space representations of datasets by introducing a 'latent bottleneck,' which ensures that the network is forced to represent the dataset in a significantly smaller dimensionality than the original dataset.

There are two parts of an autoencoder, the encoder $\mathcal{E}_{\phi_{NN}}$ mapping.

$$\mathcal{E}_{\phi_{NN}} : \mathbb{R}^k \rightarrow \mathbb{R}^d \text{ for } k > d \quad (4.23)$$

And a decoder mapping $\mathcal{D}_{\theta_{NN}}$ on the form.

$$\mathcal{D}_{\theta_{NN}} : \mathbb{R}^d \rightarrow \mathbb{R}^k \text{ for } k > d \quad (4.24)$$

Where d then becomes the latent dimension size and ϕ_{NN} and θ_{NN} are the encoder and decoder network parameters. The autoencoder is then created by cascading the encoder and decoder and optimizing the goal.

$$\operatorname{argmin}_{\theta, \phi} \|x - \mathcal{D}_\theta(\mathcal{E}_\phi(x))\| \quad (4.25)$$

By minimizing a loss function, often MSE, with the reconstruction $x' = \mathcal{D}_\theta(\mathcal{E}_\phi(x))$

$$\mathcal{L}(x, x') = \|x - x'\|_2^2 \quad (4.26)$$

And using backpropagation and gradient descent.

Even though regular autoencoders can make a latent mapping that allows for high decoding precision, it is not necessarily the case that the latent representation it learns will be continuous or behave as a well-known distribution. These are two features that can be extremely useful. A continuous latent space can make the autoencoder perform significantly better in generative tasks where it has to produce responses not seen in the training set but which are captured by the dataset's domain. Moreover, knowing which distribution the latent space will be expressed in can help make the latent representation more ordered.

Generally, we cannot say that a trained autoencoder is *generative*. I.e., it does not necessarily converge upon a latent space solution that allows for generating new examples of the dataset from latent space interpolation. These problems can be solved by a neural network architecture called the variational autoencoder (VAE).

Variational Autoencoders

In this section we will introduce the basic idea behind variational autoencoders, the purpose is not to give a full-mathematical background, which is out of the scope of this thesis, but to give a short introduction and some intuition about variational autoencoders differs from regular autoencoders.

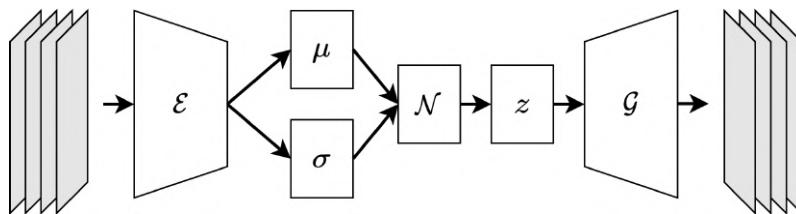


Figure 4.5: Variational Autoencoder (VAE) Architecture

From information theory, the Shannon entropy of a random variable is defined as:

$$H = - \sum_{i=1}^N p(x_i) \log p(x_i) \quad (4.27)$$

This can be interpreted as a measure that "quantifies how much information we gain, on average, following a measurement of a particular quantity" [57, sec 15, p. 158].

We can then imagine a metric that can calculate the difference in information entropy between two chosen distributions; this is usually called the *Kullback–Leibler divergence*.

$$\mathcal{D}_{KL}(p\|q) = \sum_{i=1}^N p(x_i) \log \frac{p(x_i)}{q(x_i)} \quad (4.28)$$

Simplified the KL-divergence allows us to compare the similarity between two random distributions.

This can be utilised together with an architecture similar to an autoencoder where the encoder, \mathcal{E} , is used for projection data samples into a variance and mean value, which then gets sampled from a prior distribution.

For a variational autoencoder, we thus have an encoder \mathcal{E} , which provides a mean and variance estimate, and a generator \mathcal{G} which converts samples from the latent space distribution into field examples, see fig. 4.5. Practically, this is done with the so-called 'reparameterization trick,' which is adapted from[58].

The loss function for the variational autoencoder is therefore both a combination of the KL-divergence and the reconstruction loss.

$$\mathcal{L} = \|\mathbf{F} - \mathcal{G}(\mathcal{E}(\mathbf{F}))\| + \beta_{KL}\mathcal{D}_{KL} \quad (4.29)$$

Where β_{KL} is a weighting factor of the KL–Divergence. As will be seen in the future sections, no implicit distribution from PCA analysis was found in the dataset that would justify a particular prior. So we have used the most common one, the unit-gaussian distribution. Therefore the KL-Divergence should penalize the latent space samples they do not distribute themselves as a unit gaussian.

$$z \sim \mathcal{N}(0, \bar{\mathbf{I}}) \quad (4.30)$$

4.4 Latent Space Regression Models

One might be able to obtain a latent space model with predictive abilities from parameter to latent space representations, thus making a *latent space regression*. Hypothetically this might be a make since it will divide the task of finding a parameter to field mapping into two more specialized tasks that might individually be easier.

The disadvantages of a semi-supervised model like in fig. 4.6, include that it complicates the models significantly, and therefore finding an appropriate tuned model might be a bigger challenge; furthermore, it requires both the supervised and unsupervised models to be extremely precise, since prediction error will be determined by the error cascading through the two models, this cascation is likely quite unpredictable and in some cases might be nonlinear.

For a semi-supervised model like this to generalize, again the unsupervised model needs to be generative. Decoding abilities and generativeness is the difference between learning the given parameterizations χ and learning the design space Ω_d , i.e does the latent model generalise. Mathematically there it is not ensured that linear projections such as PCA, or non-linear latent space models such as autoencoders are generative. This is why models such as variational autoencoders, which specifically penalise non-continuous latent spaces, are often used for generative tasks.

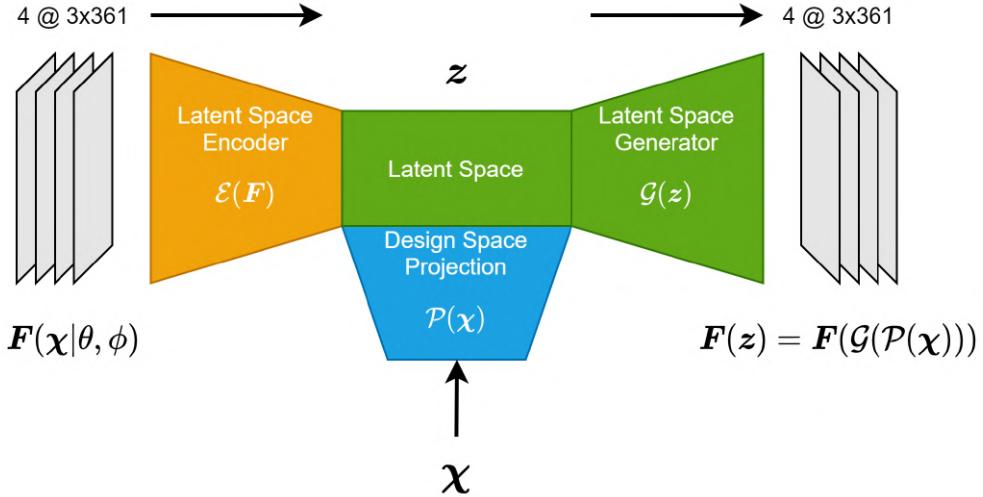


Figure 4.6: Architecture of latent space regression model

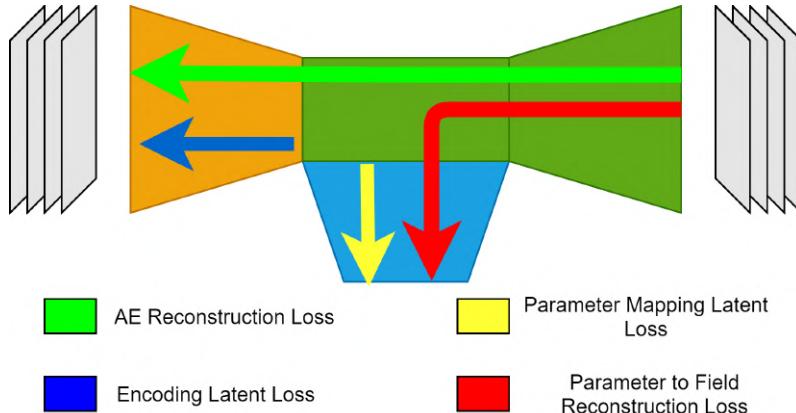


Figure 4.7: Possible backpropagation paths in semi-supervised AE-FC latent space architecture.

4.4.1 PCA Based Models

For PCA-based latent space regression, the a PCA has first been fit to a training partition of the dataset, then the training dataset has been transformed into the latent space, which then has been used as a regression target for a GP model with the antenna parameterizations as the predictor value.

4.4.2 Autoencoder-GP models

Where the linear latent decomposition of the PCA is not expected to make the latent space more learnable, non-linear latent methods such as autoencoders might obtain less complex relations to the parameterizations. Therefore using Gaussian processes to predict into the latent space with the decoder and then predicting into the far-field might improve the overall prediction performance. This is, of course, dependent on obtaining significantly good autoencoders that significantly outperform the direct prediction models.

4.4.3 Autoencoder-FC Based Models

Another considered model has been to make an end-to-end FC-AE model, where an autoencoder was trained in an unsupervised fashion on the far-field pattern to find a latent space representation, and then a parameter-to-latent FC network that is trained to do

regression inside of the obtained latent space.

The idea is that there are multiple loss mechanism that could be back propagated, and therefore obtain a solution that would both cater to having a good reconstruction error, and be easily learnable by the parameter to latent space FC-NN. In fig. 4.7, we see the potential paths for backpropagation model of this type. Firstly we still have the AE reconstruction loss that can be backpropagated through both the encoder and generator. But the idea is that we can also use the 'encoding latent loss' to together with the parameter map latent loss so that

4.4.4 Summary

In this section, we have outlined the different methods behind the dimensionality reduction techniques and the surrogate prediction models investigated in this thesis. Pyramidal deep neural networks (PDNN) and Gaussian process regression have been chosen as the direct prediction contenders. The PDNN networks have a mesh-grid output based on the sample density of the investigated datasets, and the Gaussian process models predict in an output-output manner, where every sample of the far-field sphere is seen as an independent regression problem.

Principal component analysis, autoencoders, and variational autoencoders have been outlined as dimensionality reduction techniques. Here we have focused on the importance of generativity as a goal for the latent space models, such that they cannot only reconstruct seen far-field examples but also generalize the latent space and generate new data samples.

Finally, some novel architectures have been outlined, combining the latent space models and the prediction models.

5 Experiments and Results

This chapter will outline some of the significant results obtained during this project. We will not include all relevant results in the main text but will consist of the most exciting results and then add the rest of the results in the appendix, referring to them in the main text when relevant.

Firstly we will test the datasets on a set of baseline models to get a reference for the difficulty of the problem of generating surrogate models from the dataset. Then we will display the results of various linear and nonlinear dimensionality reduction techniques applied to the different datasets as described in chapter 4. We will start with principal components analysis, then compare this with autoencoders and variational autoencoder architectures. We will then show the best direct prediction neural networks based on the PDNN architecture outline in section 4.1.2 and outline surrogates based on Gaussian process regression. Then we will show some of the experiments done on gaussian process and PCA sensitivity to datasets sizes. Finally, we will combine the latent space analysis with the surrogate models and outline the results from training the surrogate models inside the lower-dimensional latent spaces.

5.1 Latent Space Representations

In this section, we will show the results from applying the unsupervised methods outlined in chapter 4, to the four far-field datasets described in chapter 2.

We have only applied these methods to the far-fields, F , since the distributions of χ inside the design space are designed according to our sampling methods, and their distribution within the dimensions of the design space is therefore known a priori. Many of the plots in this section are colored by their parameterizations, χ . Therefore any pattern that arises in the coloring is reconstructed by the unsupervised method and is not a direct product of the parameters themselves. Any observed pattern is only a product of how parameterizations affect the high-dimensional far-field patterns and how the latent space methods can reconstruct those patterns.

5.1.1 Principle Component Analysis

In fig. 5.1, the reconstruction loss of the four antenna datasets can be seen as functions of the number of principal components from which the far-fields have been reconstructed. The dataset has been split with holdout validation, and both reconstruction accuracies seem to be almost perfectly identical. We can see that even though the datasets are high-dimensional, at 4332 dimensions, a large reconstruction precision is achieved within 10s or even in some cases only a few dimensions.

In fig. 5.2, the two principal components of RFLCT-PCA are shown three times, with each of the three parameters of the parameterization used to color it. We see that the focal length and the offset do not seem to display any pattern in the latent space. In contrast, the reflector diameter has a fascinating radial relationship in the latent space.

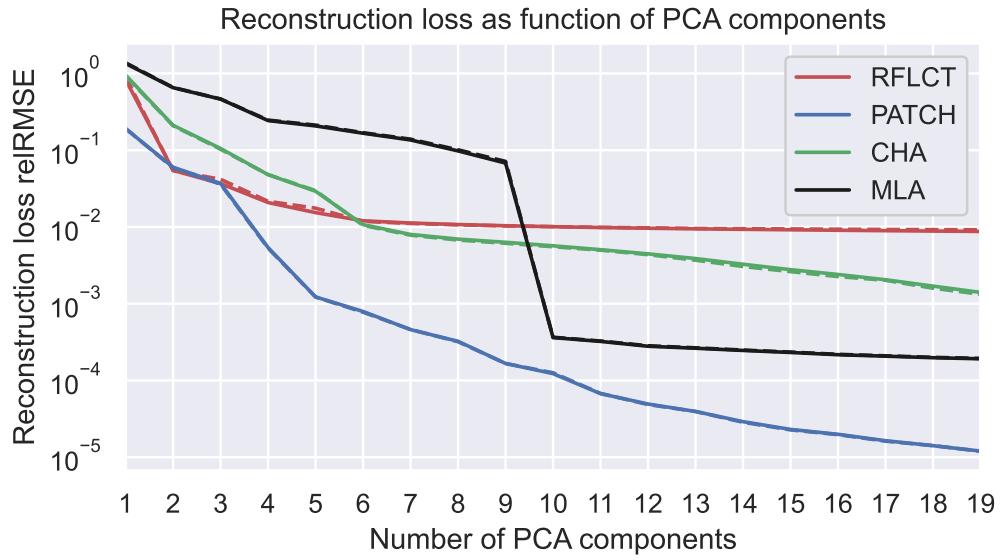


Figure 5.1: PCA Reconstruction loss for all datasets, solid is training set and striped is validation loss

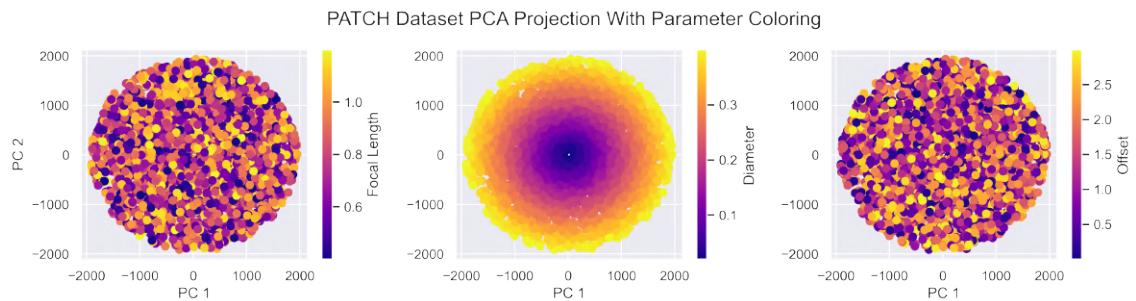


Figure 5.2: RFLT dataset PCA latent space with parameter coloring

In fig. 5.3, a 3D version of fig. 5.2 is seen. Here the rest of the latent space reveals itself as a more complex manifold. The point cloud is projected unto the sides of the plot, and

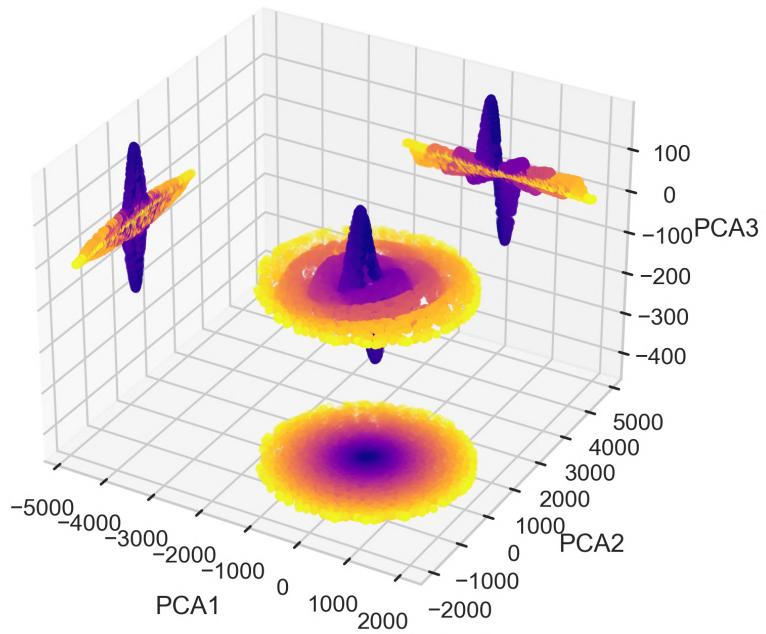


Figure 5.3: RFLCT Dataset 3D PCA plot, with 2D projections unto the sides of the plotting box. Coloring based on the reflector diameter as also seen in fig. 5.2

The PCA in fig. 5.4 shows the first two components of the PATCH dataset colored by the three parameters. It is interesting that the coaxial x-placement and substrate permittivity seem to account for one PCA dimension, whereas the coaxial-y placement seems to be correlated with a sort of radial expansions in the latent space.

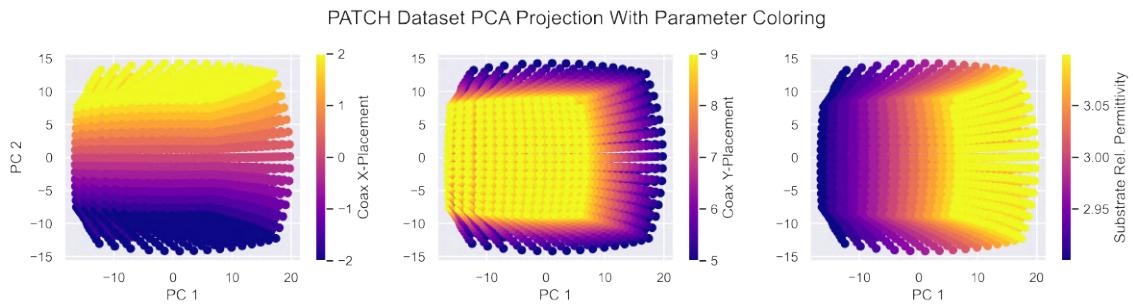


Figure 5.4: PCA of PATCH dataset with parameter coloring

fig. 5.5 shows the PCA-Parameter coloring of the CHA dataset. Here we see that the relative permittivity of the lens has some sort of angular relationship inside of the first principal components.

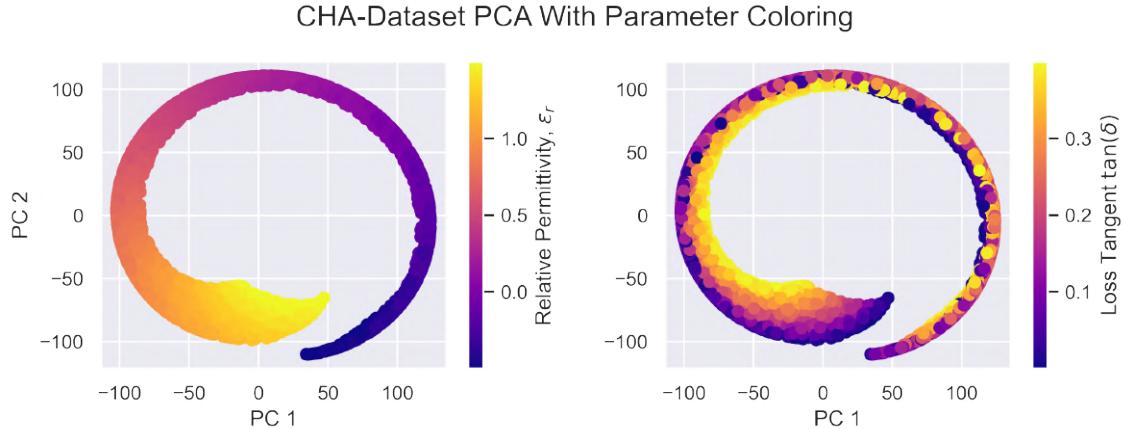


Figure 5.5: First two principal components of the CHA dataset with parameter coloring.

The relationship from fig. 5.5 is further displayed in fig. 5.19. Here we see the relative permittivity still keeps its angular relationship, but that much of the loss tangent is expressed in the third PCA. It is interesting to note that it looks like a two-dimensional manifold inside the three-dimensional PCA latent space.

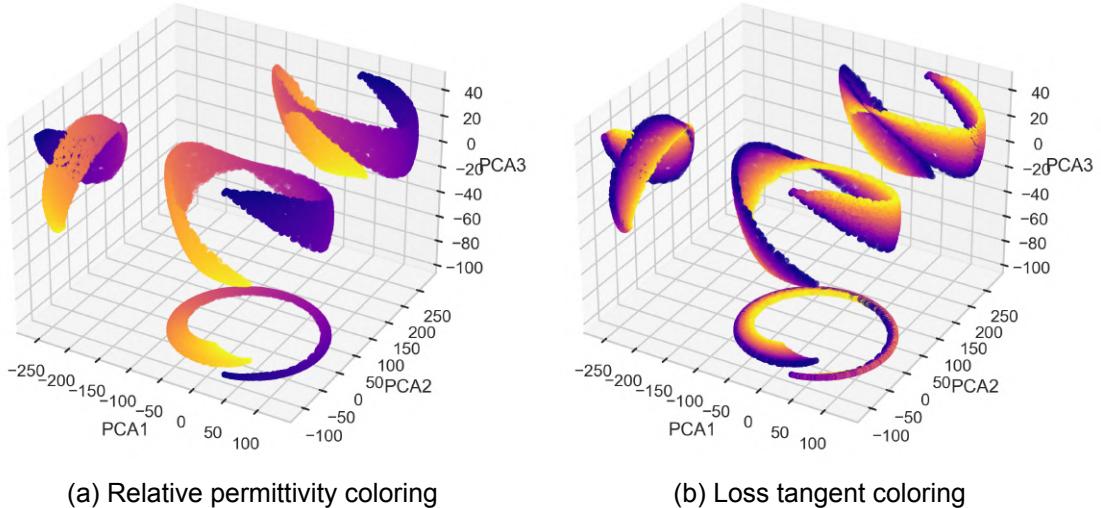


Figure 5.6: 3D PCA Projection of CHA dataset with parameter coloring and 2D projection of inner cloud unto the sides of the plot.

In fig. 5.7, we see the coloring of 10 chosen PCAs of the MLA dataset. Here we see that each amplitude, A_1, \dots, A_5 , and phase excitation, $\varphi_1, \dots, \varphi_5$, are captured in 5 different pairs of principal components. This also supports the results of fig. 5.1, where we see that the MLA dataset obtained as very good reconstruction performance in a very abrupt way at the 10th PCA components.

5.1.2 Reconstruction from the latent space

Now that we have seen that a large part of the variance of the high-dimensional far-fields can be contained in relatively few components and that these latent spaces have a structure that is closely related to the parameterizations, we will study how these latent spaces

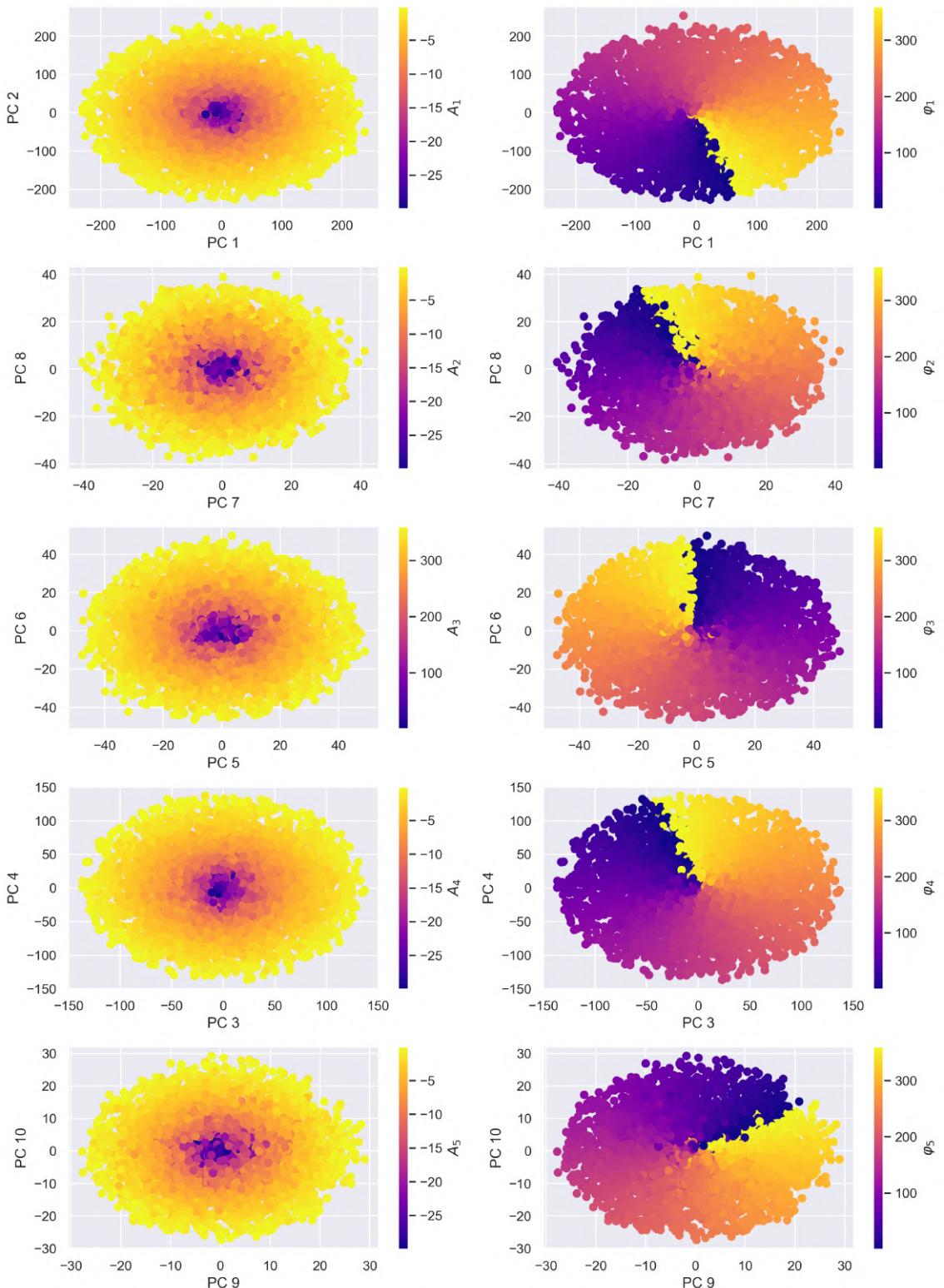


Figure 5.7: Selected PCA components of the MLA Dataset. No other PCA-Pair colorings seem to have a structure in the latent space.

relate themselves to their corresponding fields when doing the inverse transformation from the latent space into field space.

In fig. 5.8, we see the reconstruction of different points of the CHA-dataset latent space obtained by PCA back into the far-field. Since the manifold that is captured has large areas where no data is present, it is interesting to note that the fields which are extrapolated still seem to have the general shape of the fields in the CHA dataset.

The cross-polar component of the reconstructed field is seen in fig. 5.9. This also makes it apparent how much information about the far-field has been captured by the PCA.

5.1.3 Auto-encoders

The fundamental limitation of the principal component analysis when compressing datasets into lower-dimensional spaces is that it is a purely linear transformation. From the previous section it has become apparent that PCA can account for a large amount of the variance and that the datasets distribute themselves in clouds of data whose interpolation can create new examples of the far fields that have not been seen. This could indicate that the PCA finds a manifold of lower dimensions, which it has to contain in a high dimension. And that this manifold might be unwrapped into its intrinsic dimension if a sufficient nonlinear projection is to be found.

Therefore we have also investigated autoencoder architectures to see if we can obtain a better reconstruction loss at a lower dimensionality and if this is because the autoencoders are able to find the nonlinear mapping that can unwrap the latent manifolds into their intrinsic manifold.

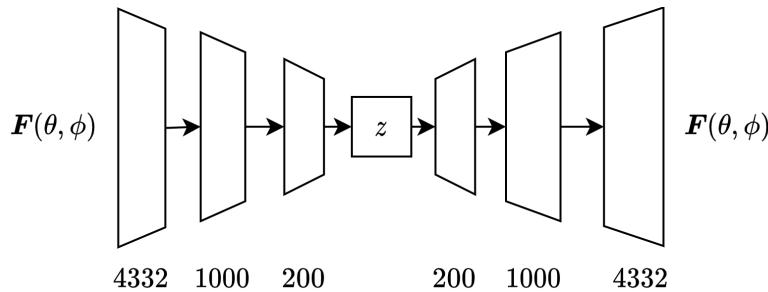


Figure 5.10: Specific autoencoder architecture that is used for the results in this section; the latent space size is varied and will be specified for each result. The numbers are the number of fully connected neurons in each layer.

In this project we have both trained convolutional autoencoders and regular autoencoders, but it has turned out that fully connected autoencoders have outperformed the convolutional ones. This might be because we have not been significantly constrained on training performance on the GPUs, and therefore fully connected autoencoders have been easier to tune the architecture. The chosen architecture we are displaying the results of can be seen in fig. 5.10.

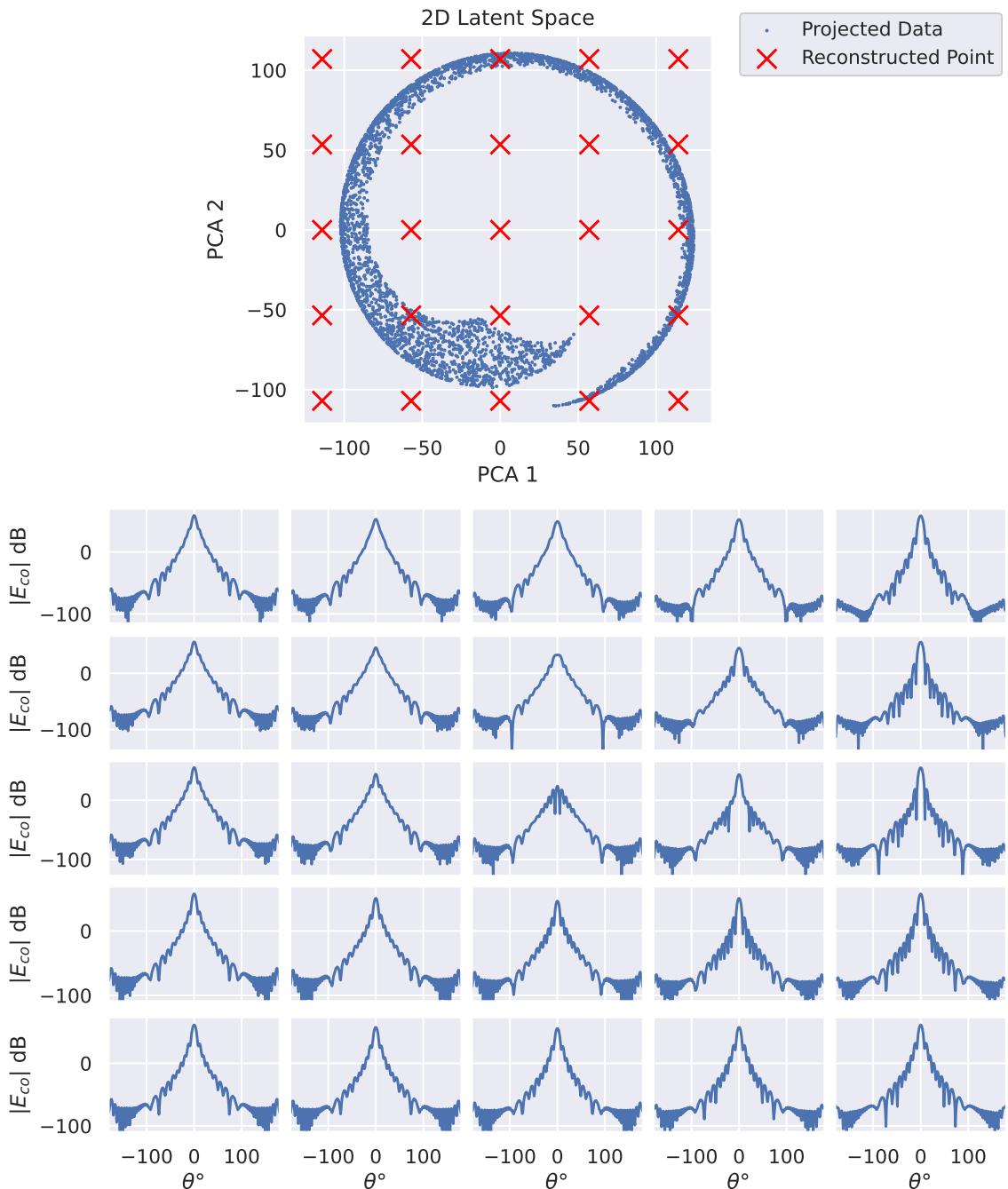


Figure 5.8: CHA cross-component latent space reconstruction, only $\phi = 0$ component plotted for eligibility since all cuts are very similar. Each point in the red grid of crosses corresponds to the same point in the grid of far-field plots below.

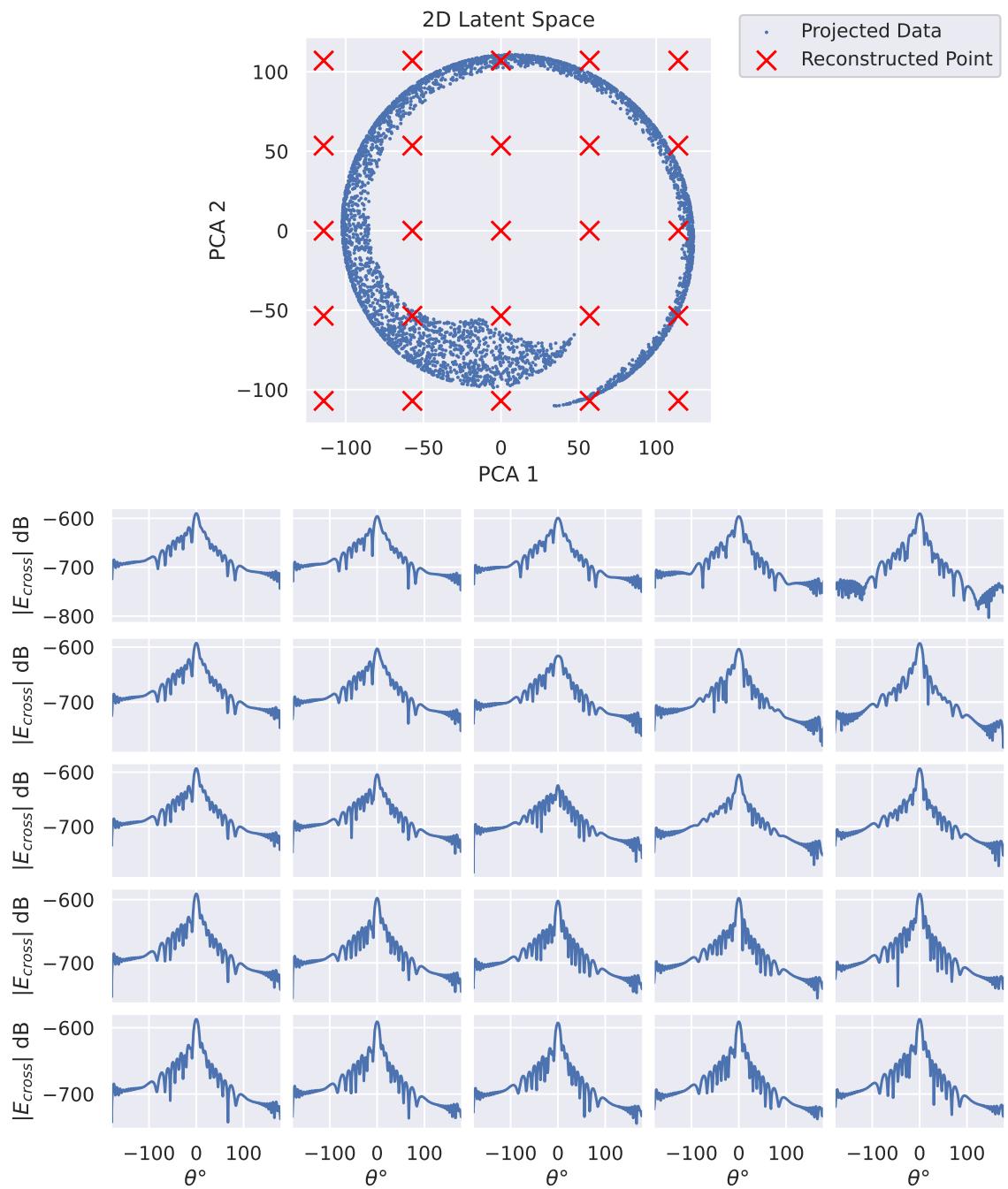


Figure 5.9: CHA cross-component latent space reconstruction, only $\phi = 0$ component plotted for eligibility since all cuts are very similar.

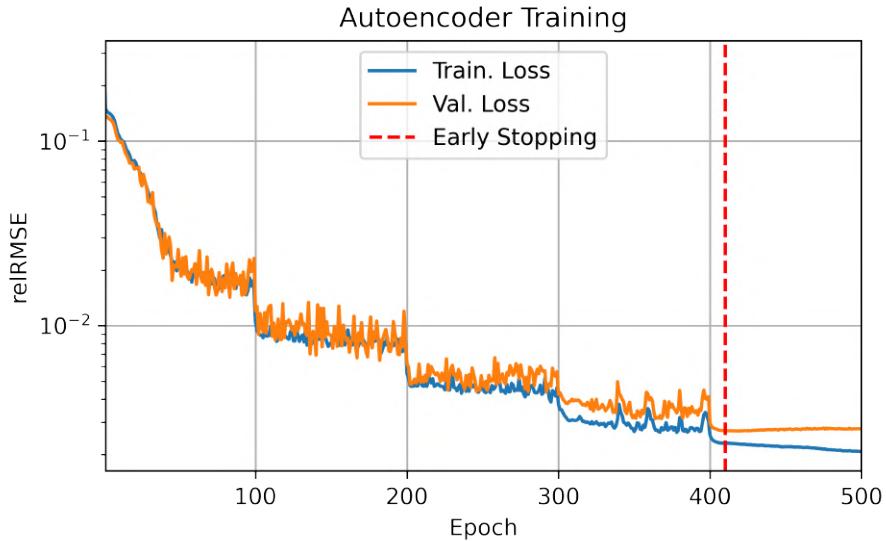


Figure 5.11: Autoencoder training progression. Here we chose to reduce the learning rate by 1/2 at every 100 epoch, and early stopping was used to minimize overfitting

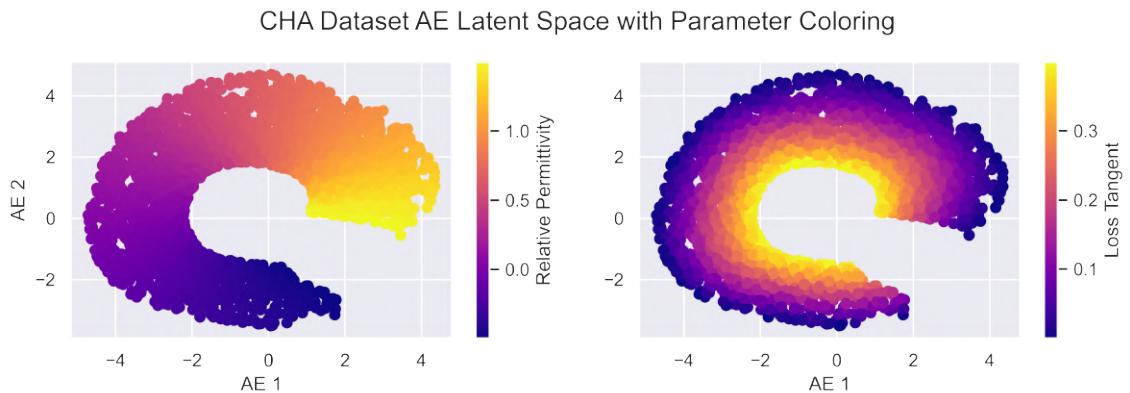


Figure 5.13: 2D Autoencoder latent space of CHA dataset, in comparison with fig. 5.19. reconstruction relRMSE of this 2D autoencoder is $2.10 \cdot 10^{-3}$. I.e approximately same performance as a 19-component PCA of the CHA dataset, as seen in fig. 5.1.

In fig. 5.13, we see the same manifold as in fig. 5.19, but flattened out by the autoencoder. Notice the relationship with the loss tangent where the, now radial, relationship is ‘untwisted’.

5.1.4 Variational Autoencoder

In this section we will show results from using a convolutional variational autoencoder to generate a latent space and to reconstruct samples from the latent space. The variational autoencoder is a 6 layer network with two convolutional layers followed by a fully connected layer for the encoder, this is then mirrored for the decoder. The specific architecture is outlined in table 5.1.

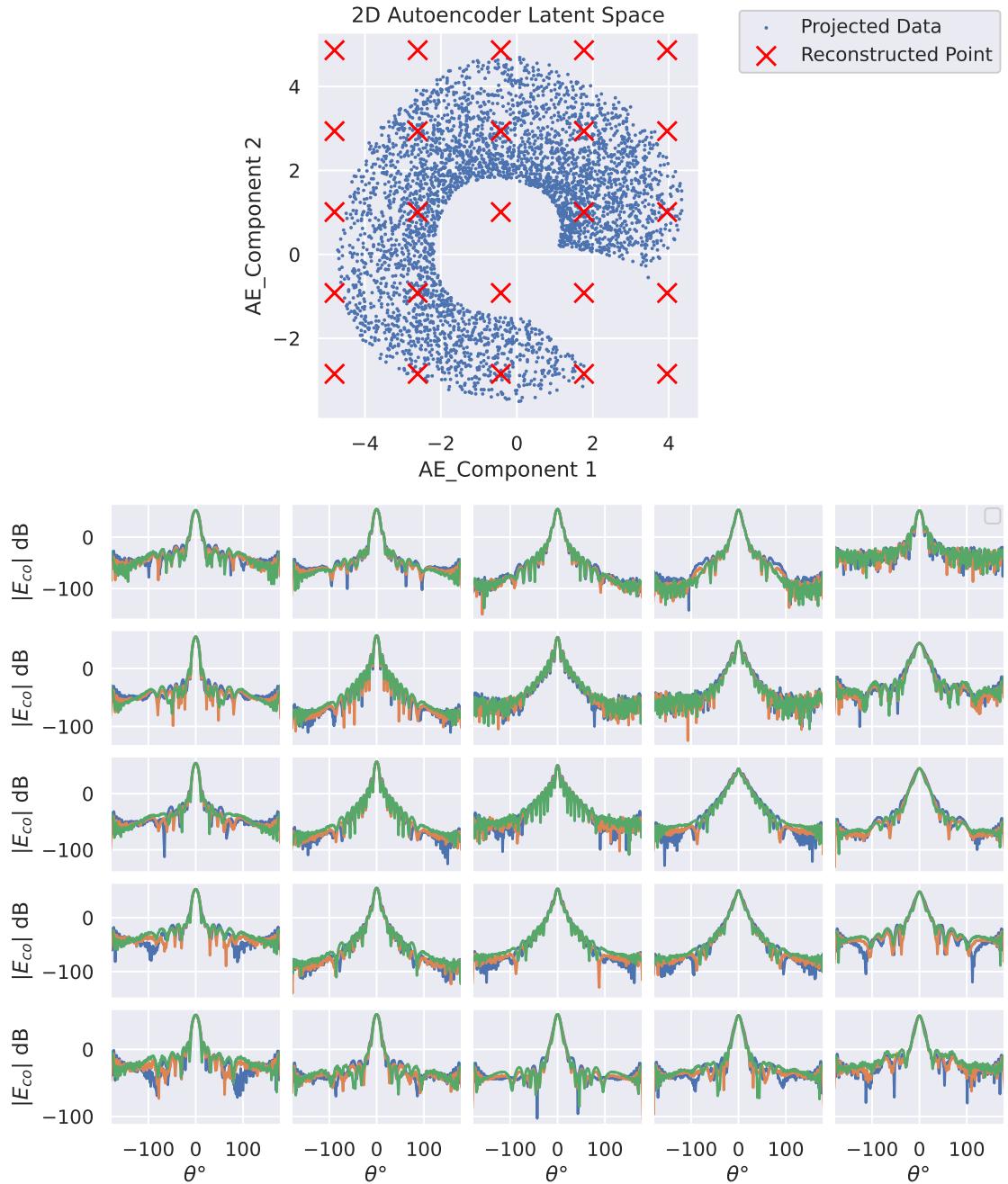


Figure 5.12: CHA dataset far-field reconstruction from 2D autoencoder latent space, as seen a significantly higher variety of far fields is represented in the 2D Space compared with the PCA 2D space of fig. 5.8

Layer	Channels	kernel size		stride		padding	
		H	W	H	W	H	W
1	32	3	11	1	2	1	10
2	128	3	11	2	2	0	0

Table 5.1: Architecture of convolutional layers used for VAE architecture, a mirrored deconvolution design is used for the decoder. Channel number where increased from a doubling scheme, because of the relative shallowness and its positive effects on performance.

During the investigation of using variational autoencoders for generating latent representations, the main question was if the generative abilities of the could be found by introducing some KL-divergence into the loss function. Figure 5.14 shows a comparison of 5 the latent spaces of 5 identical variational autoencoders, with varying amount of KL-divergence loss-factor introduced, as shown in section 4.3.2. As we can at low β_{KL} seems to yield results quite similar to the regular autoencoder in section 5.1.3, though for higher β_{KL} we see the latent space become more compressed generally. But also that a certain set of the points seems to expand further away from the center of the distribution. Figure 5.14e shows that the variational autoencoder converges on a unit-gaussian at sufficiently high β_{KL} .

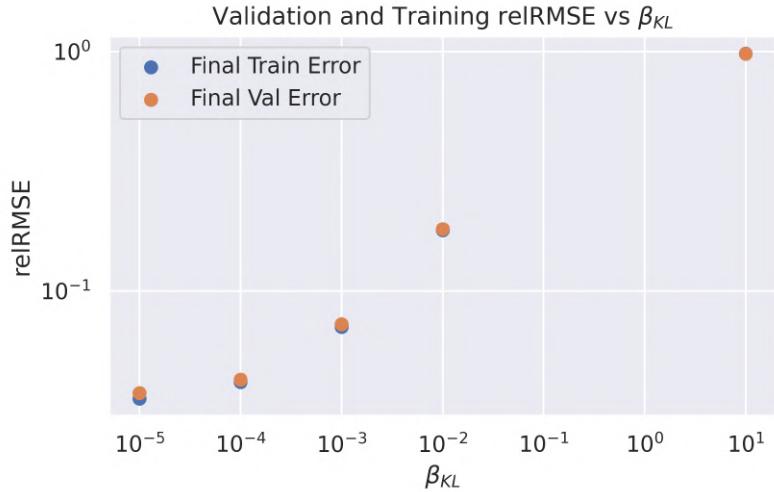


Figure 5.15: Reconstruction performance of the 5 VAE models also shown in fig. 5.14. Reconstruction examples can be see in appendix F.

Figure 5.15 shows the reconstruction performance of the five variational autoencoder models. Here we see clearly that not only is the variational autoencoder generative, it losses performance with higher values of β_{kl} . Thus it does not seem like there is much advantage to trying to force the latent space to take on a normal distribution. Examples of reconstructions can be found in appendix F.

5.2 Direct Parameter to Field Prediction

In this section, we will go over the direct parameter-to-field prediction surrogates trained during this project. Firstly we will go over the pyramidal deep neural networks architecture,

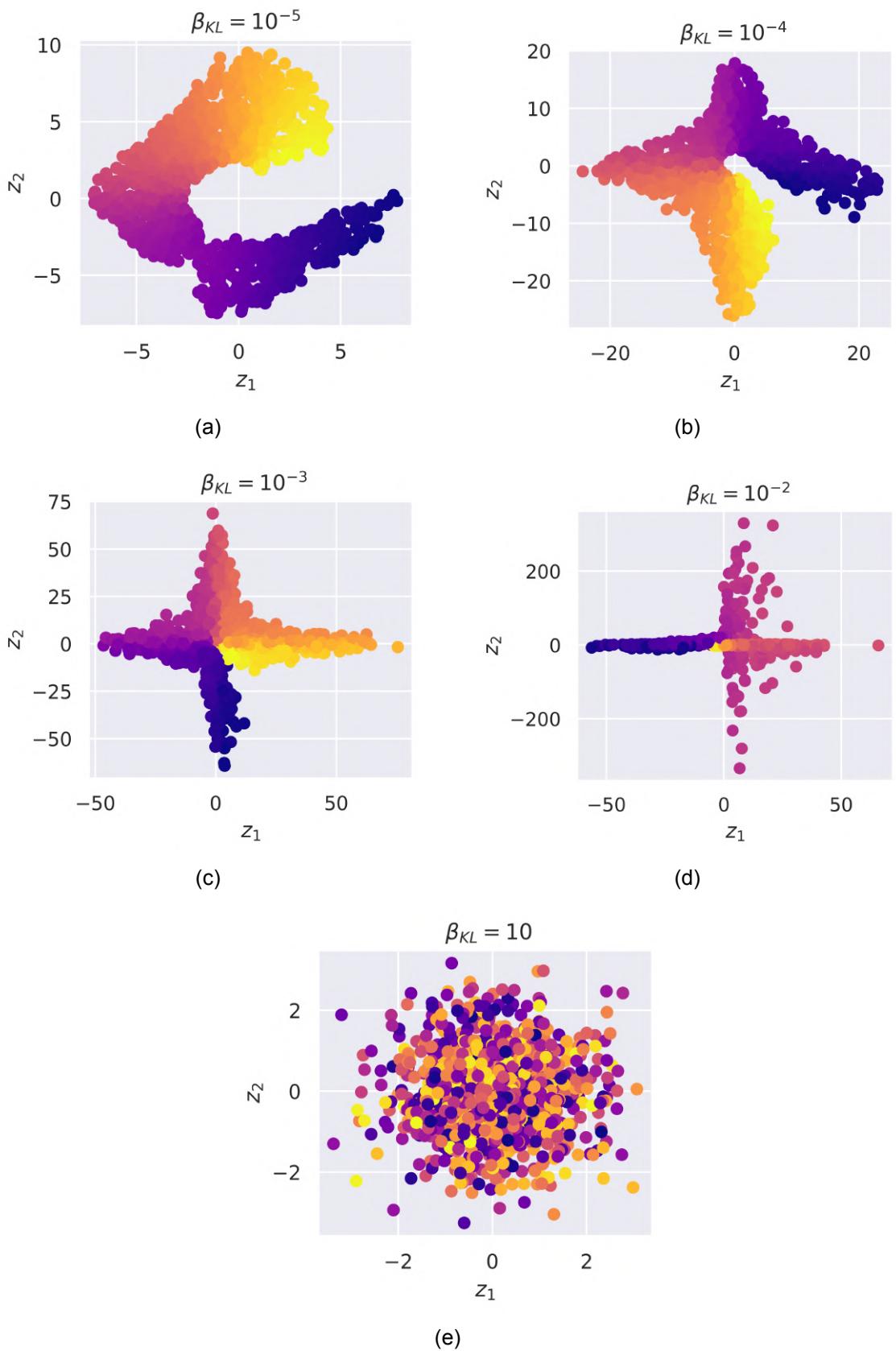


Figure 5.14: Latent space samples from a variational autoencoder trained on the CHA dataset with varying levels of KL-Divergence weighting. Colored by relative permittivity.

as outlined in section 4.1.2. Here we will discuss the hyperparameter tuning of the PDNN models and show their performance on the four different datasets. Then we will show the Gaussian process surrogates models in the same manner.

5.2.1 Simple Interpolation Baselines

Since no former models have been run on the datasets, and since there are no real baseline datasets that we can use to compare our developed models with the rest of the literature, we will try to create a 'simple' model baseline that should give some intuition of the complexity of the datasets. We have chosen to use the K-nearest-neighbor model since it is a good representation of the sparsity of the sampled design space. Since if the design space is significantly oversampled, very high precision K-nearest neighbor regression should be possible. We conducted it with one neighbor KNN regression, the equivalent of a linear interpolating between samples, and also made a number-of-neighbors-sweep to see if we could find a more optimal number of neighbors that contribute to the KNN. Finally, we have included a mean value dummy regression method that predicts the mean of the training in every regression dimension.

In table 5.2 we see the result of this baseline. The RFLCT and MLA datasets seem to have very high baseline error rates. They are both very high gain, and more variance is expected in their datasets. Interestingly the CHA dataset, while also being high gain, is the most 'interpolable' dataset of the four. It is also the dataset with the fewest parameters, and the curse of dimensionality will be significantly less burdensome than higher parameter number datasets.

Dataset	Mean Reg.	$K = 1$ KNN		Opt. KNN Reg.	
	Val. relRMSE	Val. relRMSE	K Neigh.	Train relRMSE	Val. relRMSE
RFLCT	$1.49 \cdot 10^1$	$1.50 \cdot 10^0$	1	0	$1.50 \cdot 10^0$
PATCH	$4.67 \cdot 10^0$	$6.89 \cdot 10^{-2}$	1	0	$6.89 \cdot 10^{-2}$
CHA	$3.21 \cdot 10^{-1}$	$2.41 \cdot 10^{-2}$	5	$1.58 \cdot 10^{-2}$	$1.81 \cdot 10^{-2}$
MLA	$9.10 \cdot 10^1$	$9.44 \cdot 10^{-1}$	2	$5.24 \cdot 10^{-1}$	$9.39 \cdot 10^{-1}$

Table 5.2: KNN-regression baselines.

5.2.2 Pyramidal Deep Neural Networks

This section will show some of the best-obtained models from using the pyramidal deep neural network architecture outlined in section 4.1.2.

All models that have been chosen have been optimized through a Bayesian optimizer. Due to the long training time¹, only 10s of hyperparameter tuning iterations have been conducted at most.

Most trained PDNN surrogate models seem to get performance increases from small batch sizes. To both take advantage of the larger model, that is possible to train with sizeable parallelizable batch sizes and to obtain the performance gain of the small batch sizes, a compromise between the size of the network and the training batch size has tried to be achieved.

Direct Prediction Performance

In table 5.3 we see the table of the best-achieved prediction accuracy for the fully connected PDNNs. For the PATCH and CHA datasets, extremely good accuracies are achieved.

¹10s of minutes to hours

Though for the MLA and RFLCT datasets an significantly lower accuracy was achieved. Especially the MLA dataset was very prone to overfitting.

Dataset	Architecture				Error in reIRMSE		
	φ_k	k	s_c	α	Train	Validation	Test
RFLCT	2000	4	1.5	0	$0.999 \cdot 10^{-1}$	$9.94 \cdot 10^{-1}$	$1.02 \cdot 10^0$
PATCH	1000	6	1.25	0.1	$5.11 \cdot 10^{-4}$	$1.41 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$
CHA	1500	4	1.5	0.01	$6.01 \cdot 10^{-3}$	$1.01 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$
MLA	2000	4	1.4	0.08	$2.11 \cdot 10^{-1}$	$3.98 \cdot 10^{-1}$	$3.80 \cdot 10^{-1}$

Table 5.3: Performance of best obtained direct neural network regressors using PDNN-architecture and Bayesian hyperparameter optimization. Training time in 10s of minutes. A model that would successfully converge on the RFLCT dataset was never found.

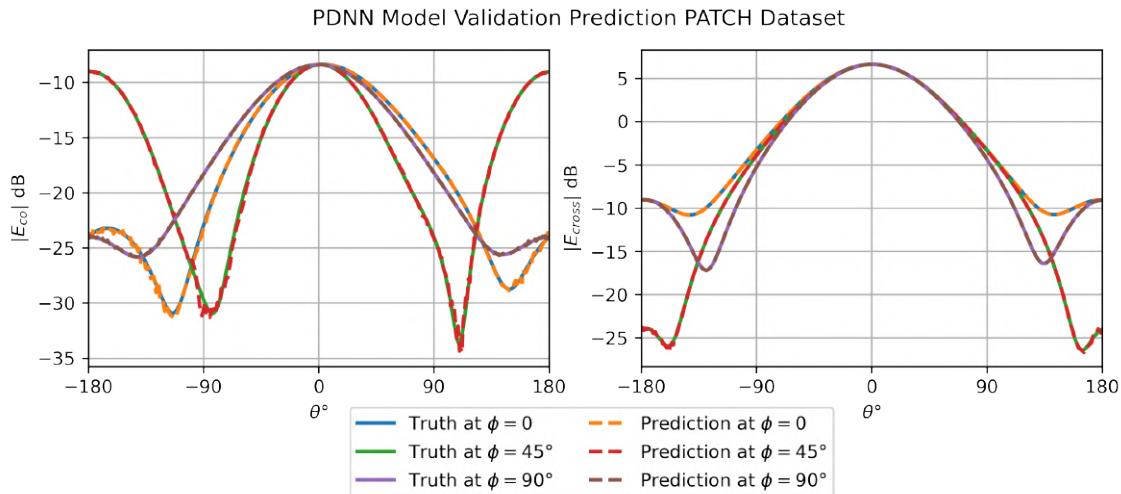


Figure 5.16: PDNN validation prediction for PATCH dataset, $\text{reIRMSE} = 3.33 \cdot 10^{-3}$

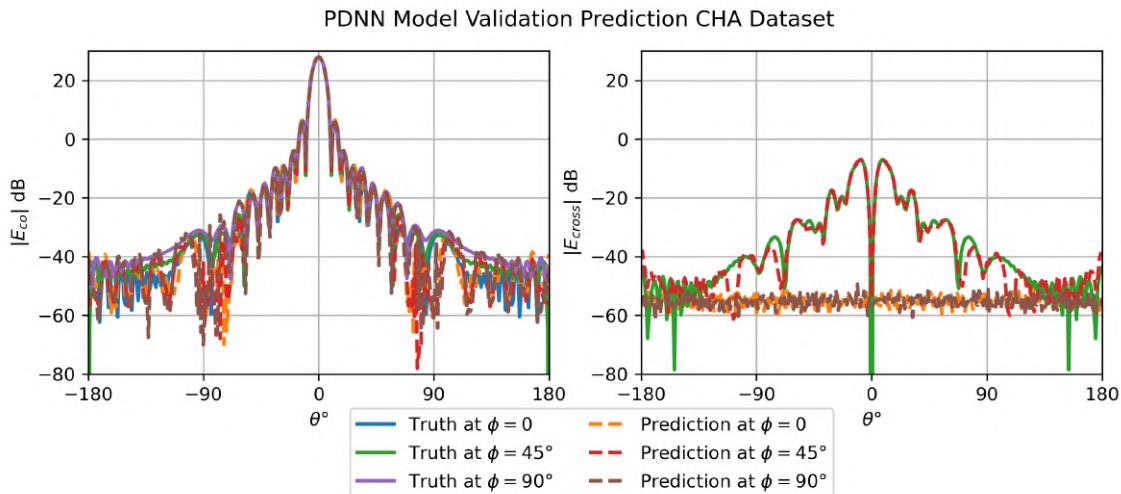


Figure 5.17: PDNN CHA Prediction

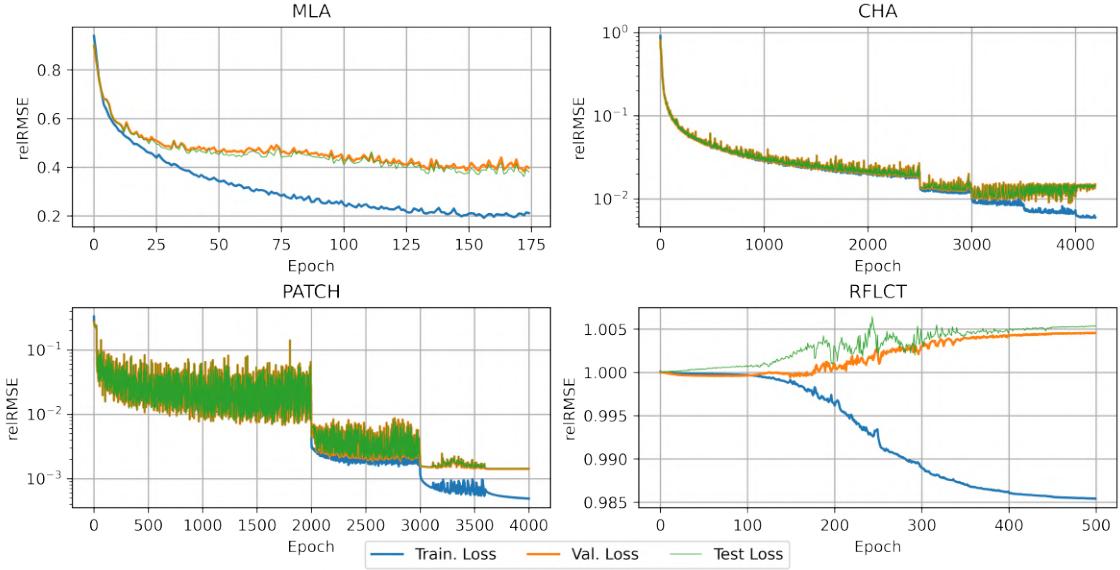


Figure 5.18: PDNN training progression

5.2.3 Gaussian Processes

In this section we will outline the trained gaussian process direct prediction models. Firstly we will give some examples of the loss landscapes that where found during the hyper-parameter tuning, then we will outline the achieved performance for each of the datasets and compare them.

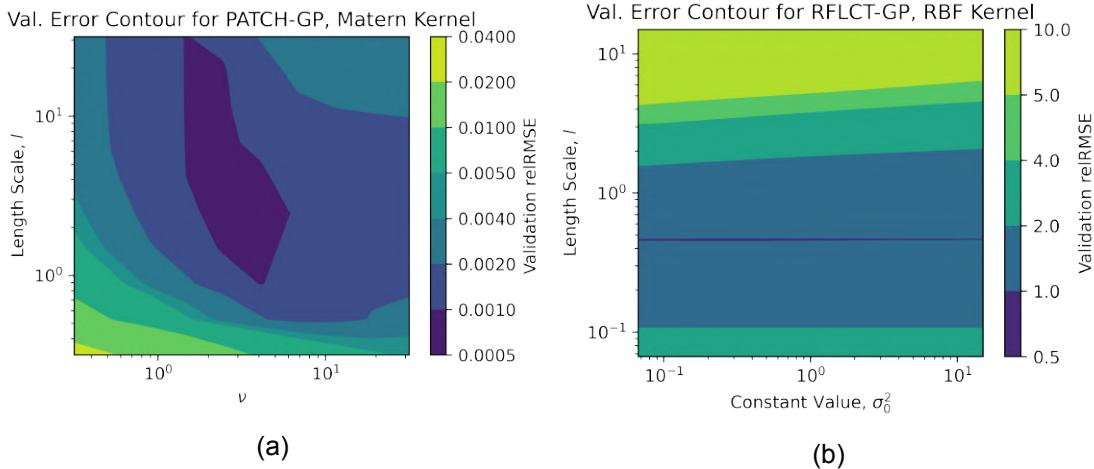


Figure 5.19: Examples of loss landscapes for the different kernel types. Done from 10×10 grid of evaluations in the space.

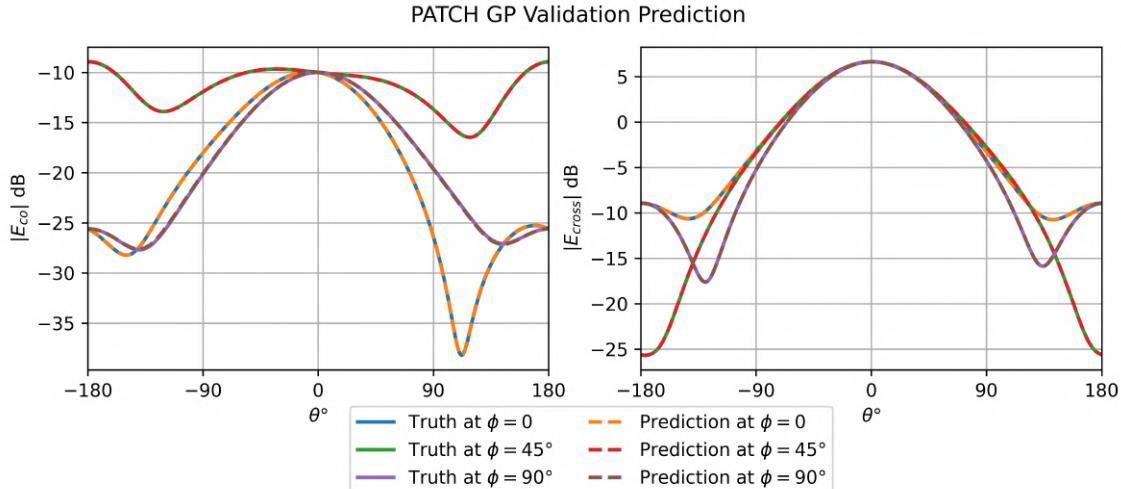


Figure 5.20: PATCH validation example of the GP model prediction

5.2.4 Direct Prediction Performance

Dataset	Kernel	Hyperparameters				Error relRMSE	
		ν	l	p	σ^2	Train	Validation
RFLCT	RBF	—	0.88	—	1	$1.006 \cdot 10^0$	$1.012 \cdot 10^0$
PATCH	Matérn	7.5	5	—	0.5	$1.44 \cdot 10^{-3}$	$1.55 \cdot 10^{-3}$
CHA	RBF	—	0.335	—	10	$5.37 \cdot 10^{-4}$	$1.44 \cdot 10^{-3}$
MLA	RBF	—	40	—	100	$1.69 \cdot 10^{-1}$	$2.20 \cdot 10^{-1}$

Table 5.4: Direct prediction performance of single output Gaussian processes (SOGP) models. Hyperparameters were chosen based on a compromise between best validation error and overfitting. Matérn was tuned first, and if a very high ν value was needed not to overfit, the RBF kernel was used instead. The implementation of the periodic kernel seemed to have numerical instabilities, we attempted to solve the numerical instabilities by introducing some Gaussian noise through the β^{-1} value, but this led to high error rates. All datasets were standardized.

A thing to note is that for most of the best-found hyperparameters, there was no generalizability in the sense that the datasets all needed significantly different hyperparameters to ensure low validation error. For example, the CHA datasets' best-found hyperparameters lead to extreme overfitting on the MLA and RFLCT datasets with $9.985 \cdot 10^{-12}$ and $4.170 \cdot 10^{-11}$ respective training error with $1.48 \cdot 10^2$ and 1.5 relRMSE validation error.

In fig. 5.20, we see the performance of the GP model in predicting a random validation example of the PATCH dataset. With a relRMSE validation error of $1.55 \cdot 10^{-3}$, it is on visual inspection indistinguishable from the true field, even down to -30dB. The same is the case in 5.23, where even with a significantly more complicated radiation pattern, the validation error is better at $1.40 \cdot 10^{-3}$. Both these models also outperform the KNN regression baselines by more than an order of magnitude.

In fig. 5.21, a validation prediction of the RFLCT dataset is shown. Here we see the general shape is captured consistently, but that a lot of the details seems to be lost. With such a high gain radiation pattern, it also has to be considered that a large part of the

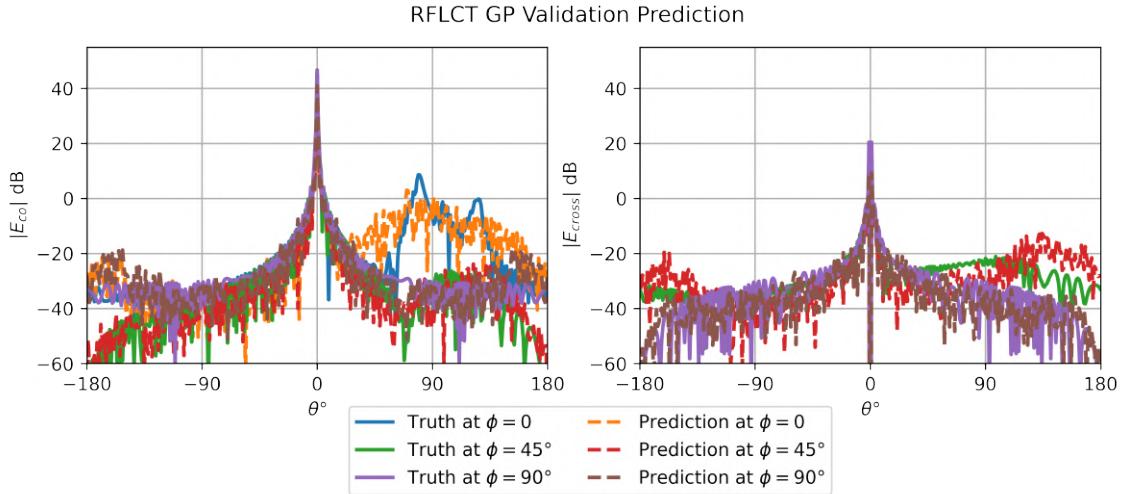


Figure 5.21: RFLCT validation example of the GP model prediction

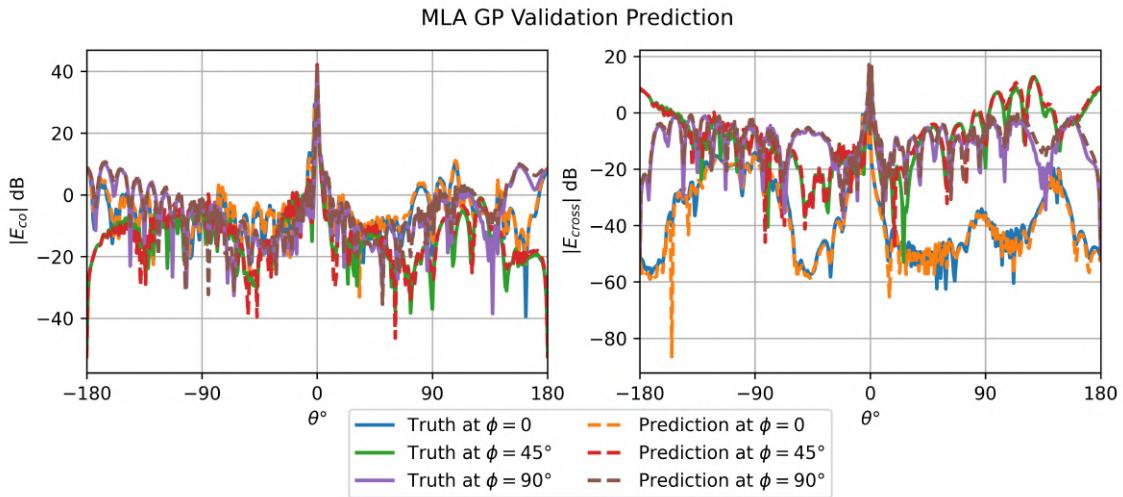


Figure 5.22: Validation example of GP model prediction in the MLA dataset

error is probably captured in not predicting the main beam properly. With a reIRMSE loss of ≈ 1 this also seems to be the case. The MLA dataset in fig. 5.22 most of the low-dB variance seem to be captured, but with an dataset average of reIRMSE of $2.20 \cdot 10^{-1}$, a substantial part of the error is probably captured in main beam pattern.

Direct Prediction Summary

We have seen that GP and PDNN models successfully outperform the interpolation baseline in all cases. The PDNN performs the best in the CHA, PATCH, and RFLCT datasets, but only by small margins in all cases. The GP model seems to outperform the PDNN for the CHA datasets, where the PDNN was also very prone to overfitting. A summary of the results, and a comparison with the baseline, can be seen in table 5.5.

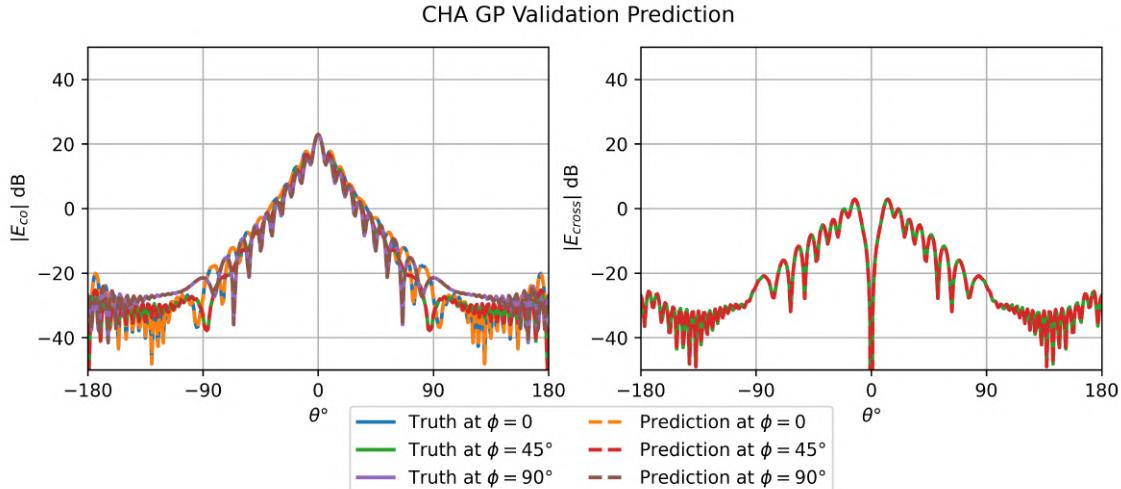


Figure 5.23: CHA validation example of the GP model prediction

Dataset	$N = 1$ KNN	PDNN	GP Regression
	Val. relRMSE	Val. relRMSE	Val. relRMSE
RFLCT	$1.50 \cdot 10^0$	$9.99 \cdot 10^{-1}$	$1.012 \cdot 10^0$
PATCH	$6.89 \cdot 10^{-2}$	$1.41 \cdot 10^{-3}$	$1.55 \cdot 10^{-3}$
CHA	$2.41 \cdot 10^{-2}$	$1.01 \cdot 10^{-2}$	$1.40 \cdot 10^{-3}$
MLA	$9.44 \cdot 10^{-1}$	$3.98 \cdot 10^{-1}$	$2.20 \cdot 10^{-1}$

Table 5.5: Validation error comparison of PDNN and GP models with the KNN interpolation baseline

5.3 Model Sensitivity to Dataset Sizes

We have shown that both Gaussian processes and neural networks can be trained to be surrogate models of the datasets under investigation and that both PCA and autoencoders can be trained effectively to be generative latent space models. With these two results in mind, the practical question arises; how do the model accuracies compare with regard to data starvation. I.e., how much can we reduce the number of training samples of the datasets before the surrogates become inaccurate and the latent space models lose generativity? An empirical study of this relation will be the subject of the following section.

The performance of a GP model as a function of its dataset size is outlined in fig. 5.24. We see that the PATCH GP model training and validation error converge quickly and would seem to reach acceptable error levels at only a couple of hundred samples and that it does not gain significantly more precision after a dataset size of 1000 samples. The CHA dataset is a different story; even though high precision is eventually achieved, it still seems to benefit from more data. This contrasts with the fact that CHA only has two parameters, and the design space is volumetrically more densely sampled, whereas PATCH has three parameters. One note to consider on fig. 5.24 is that it is the same hyperparameters for all dataset sizes and that more precision might be achieved at smaller dataset sizes if a hyperparameter tuning is conducted at the specific dataset size.

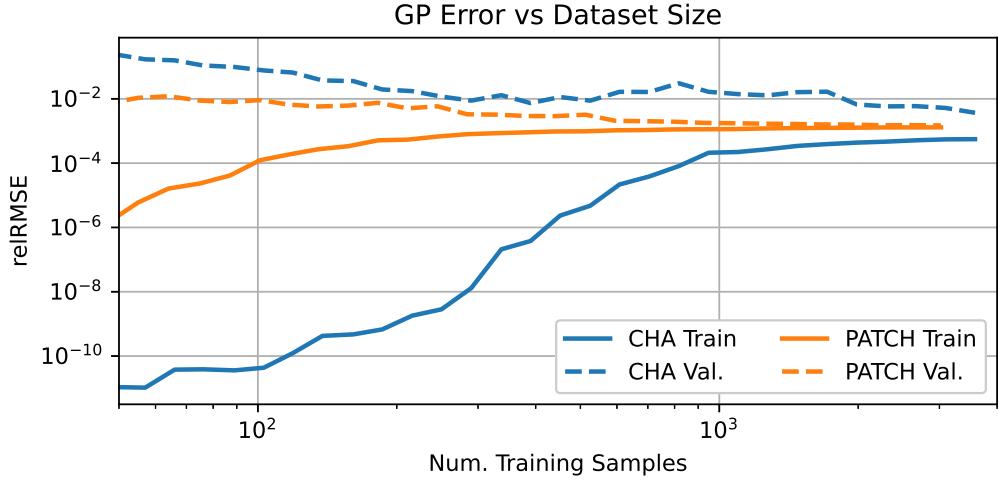


Figure 5.24: Loss of Gaussian process regressors from table 5.4 as function of training dataset size. The dataset was shuffled at every iteration. Training size was selected and validation was done with remaining dataset in a holdout manner.

The dataset-size sensitivity of the PCA generalizability seems to be quickly converging towards the overall performance that can be achieved within the dataset, as seen in fig. 5.25. Here all datasets are transformed into a 10-dimensional PCA latent space and then inversely transformed back into the full far-field dimensionality. We can see that even at only a few 100 samples of the training sets, the convergence of training and validation errors can be seen.

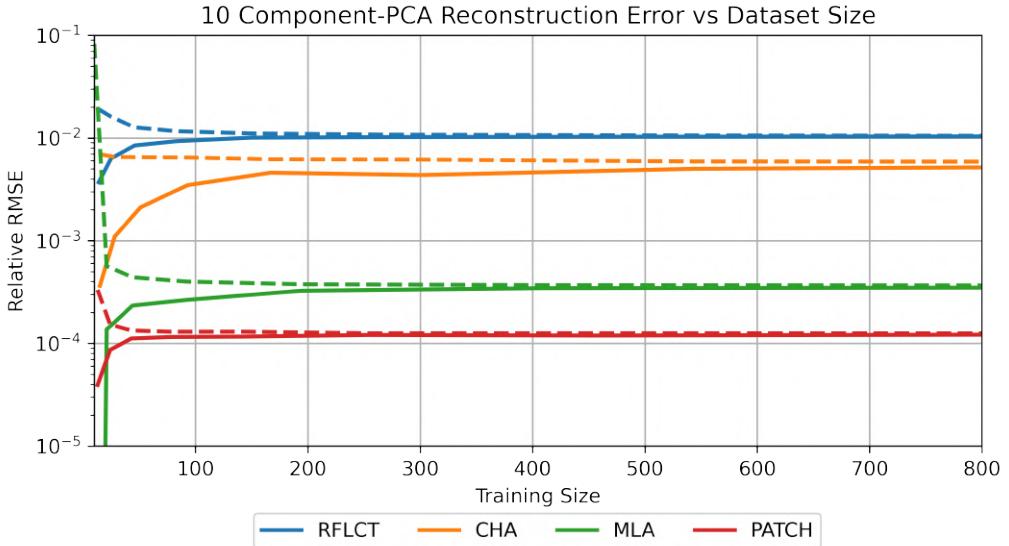


Figure 5.25: Training and validation reconstruction error of 10D PCA, solid is training loss and striped is validation error.

5.4 Latent Space Regression

The results in 5.1 show that the sampled far-fields suffer from severe over-dimensionality. In this section, we will see if these previously shown dimensionality reduction techniques, combined with the prediction techniques of chapter 5 yield similar results for the linear-

dimensionality reduction, such as PCAs, and if we can use the nonlinear autoencoders to ‘boost’ the regression models, by training the unsupervised methods to make the latent space more learnable than the sampled far-field.

5.4.1 PCA-GP

In this section, we will outline the different PCA-GP models that have been made to understand if GP regression can take advantage of latent spaces found by the principal component analysis.

In the following 3 plots, figs. 5.26 to 5.28, we see the performance of a GP-regression model predicting into the latent spaces of the PCA models. These predictions are then reconstructed from the latent space by the inverse PCA transformation, and their reconstruction error is then determined. We compare their performance with ‘baseline’ models, which are the same GP-regression models but are given the entirety of the far-field dimensionality instead of the reduced dimensionality of the latent space.

In these plots we can see that none of the models outperform their GP-baseline. But that many of them converge on an identical performance at a very low dimensionality relative to the dimensionality of the dataset.

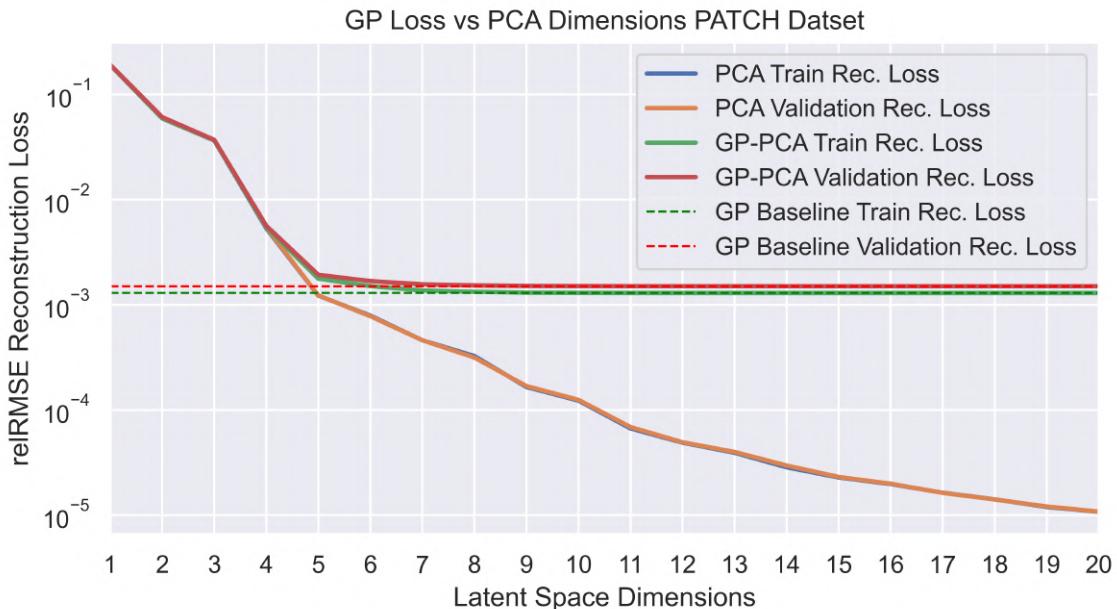


Figure 5.26: PCA and GP-PCA reconstruction loss as function of latent space size for the PATCH dataset

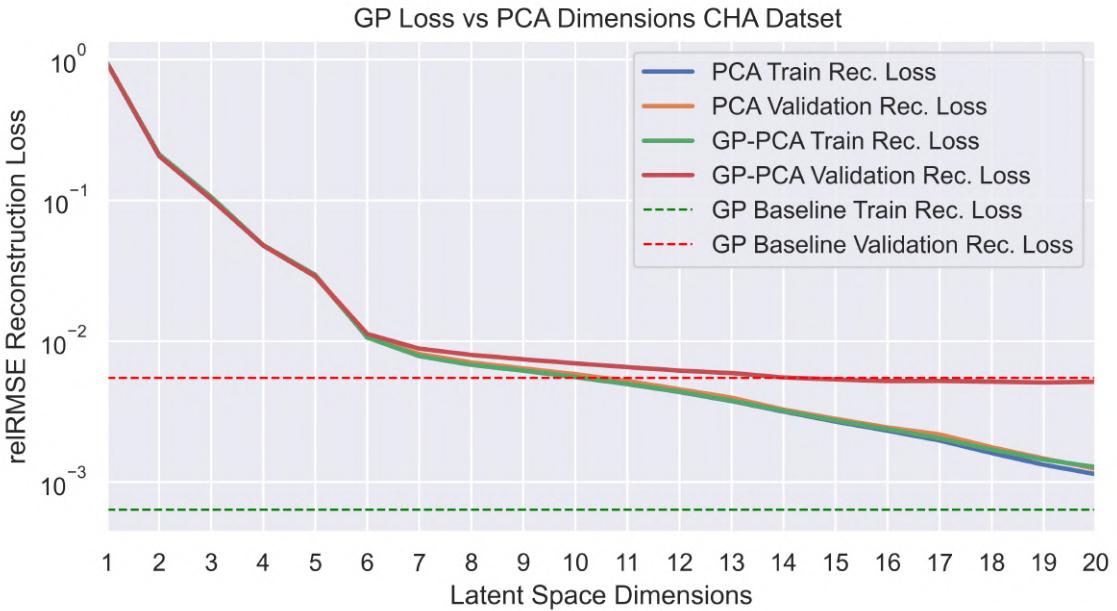


Figure 5.27: PCA and GP-PCA reconstruction loss as function of latent space size for the CHA dataset

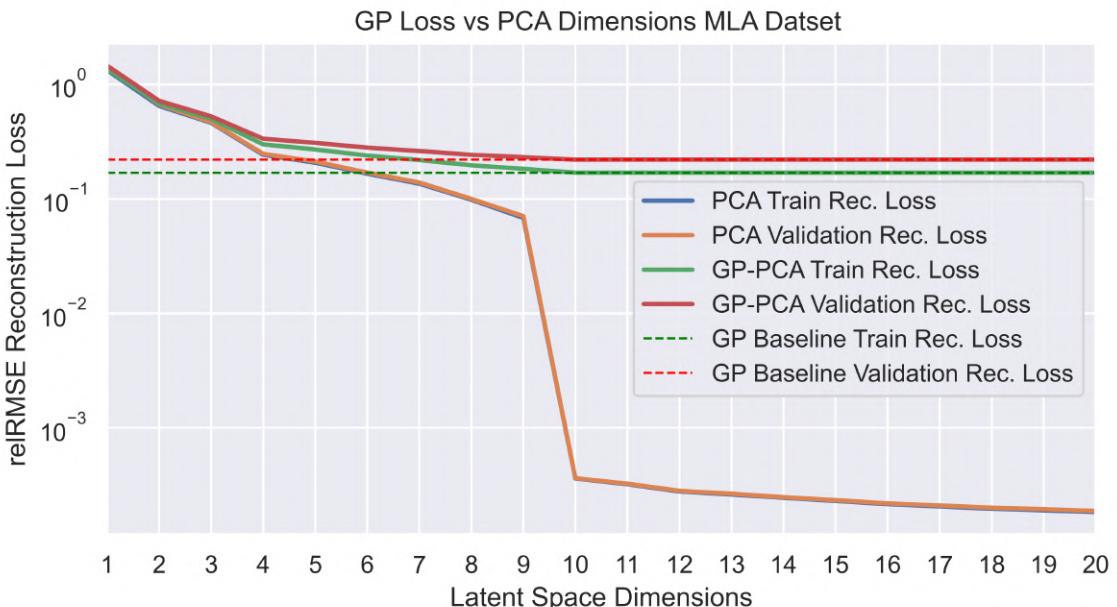


Figure 5.28: PCA and GP-PCA reconstruction loss as function of latent space size for the MLA dataset

Figure 5.29, a single θ -cut of the GP-PCA prediction model in fig. 5.27. Here we see a visual indication of the reconstruction loss and that the models still perform very well, even at low dimensionality. At 10 and 20 dimensions, only the last low-magnitude behavior is accounted for, which is also around 140 dB (7 orders of magnitude) below the peak of the radiation pattern.

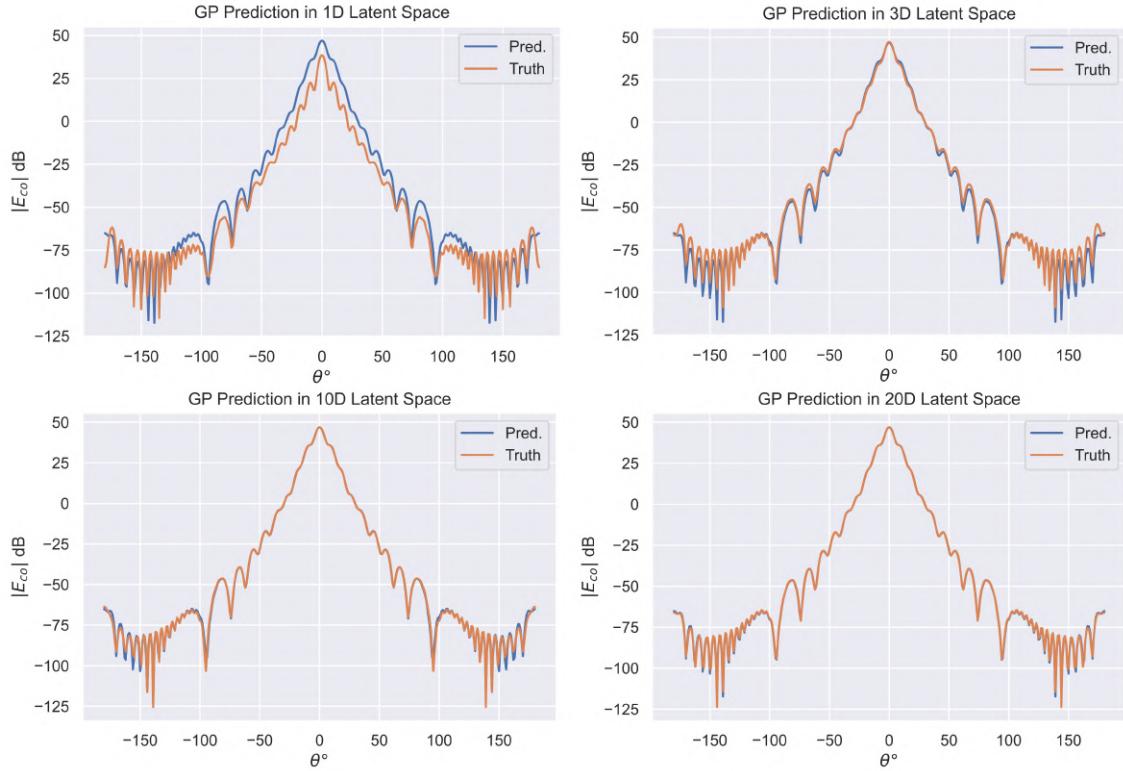


Figure 5.29: Showcase of GP Prediction in varying number of latent dimensions for the CHA dataset.

5.4.2 AE-GP

In fig. 5.30, we see the application of a GP regression model that is predicting into the latent space from the design space. I.e becoming the projection $\mathcal{P} : \Omega_D \rightarrow Z$. Here we see that the Gaussian process can learn the autoencoder's latent space, thereby constructing a semi-supervised model. The reconstruction performance is significantly higher than for the PCA-GP model at the same latent space dimension. However, since significantly more work and larger models have been used for direct GP and PDNN models, the autoencoder's original performance limits the final prediction performance of these semi-supervised methods.

Summary

In this section, we have shown some of the most interesting results obtained during this thesis project. We have made KNN regression baselines for all the datasets under investigation. Pyramidal deep neural networks and Gaussian processes regression models have also been trained. Both models seem to outperform the KNN baseline in all dataset cases. The Gaussian process models significantly outperformed the neural network models in the CHA and MLA case, whereas the PDNN models slightly outperformed the GPs in the PATCH case.

The RFLCT dataset seems to be a complex problem, with the PDNN and GP regression models obtaining similar errors at a relRMSE of around 1.

Multiple latent space models have been investigated. PCA has shown to find very low dimensional representations of the datasets that still have good reconstruction performance. The CHA, PATCH, and MLA case can reconstruct the far-fields with a relRMSE below 10^{-3} at only 20 latent dimensions down from 4332 dimensions. The distribution

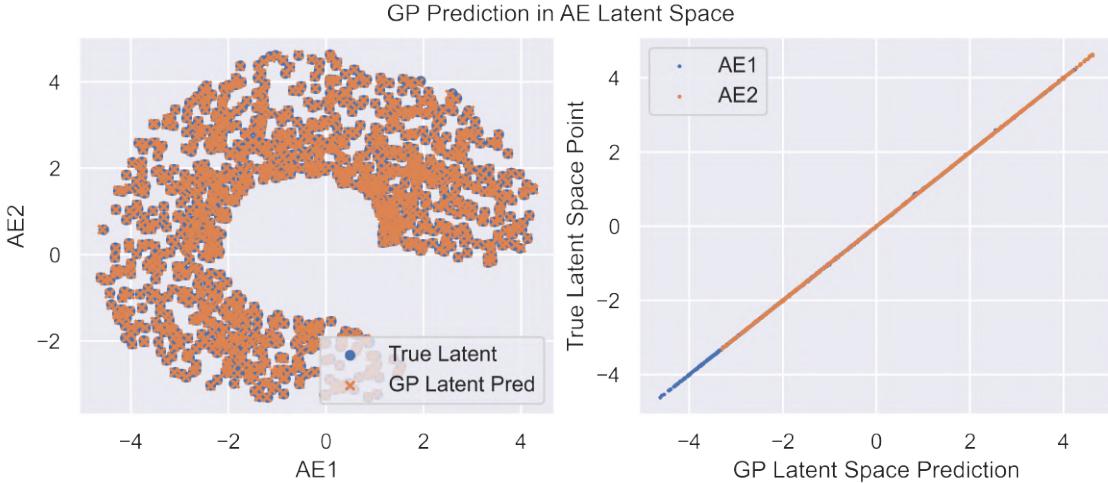


Figure 5.30: Prediction by GP model into the latent space of a 2D AE model trained on the CHA dataset. The displayed predictions are the validation set. The GP used an RBF kernel with $\ell = 0.37$ and $\sigma^2 = 0.05$. Reconstruction reIRMSE of pure AE was $2.10 \cdot 10^{-3}$, latent space GP reIRMSE = $2.64 \cdot 10^{-3}$, and the total loss between from parameter to field was $3.15 \cdot 10^{-3}$.

of far-field samples that are projected down into the latent space seems to behave as low dimensional manifolds, which might be expressed in an even lower dimensional latent space with the proper nonlinear decomposition. The results of the autoencoder show that the CHA dataset, which shows itself as a two-dimensional manifold in the PCA latent space, can be expressed in only two latent dimensions and still have a reconstruction reIRMSE of only $2.1 \cdot 10^{-3}$, making it outperform a 19-component PCA.

Variational autoencoders have been used to try to create gaussian distributed latent spaces from the far-fields. This seems only to have made the reconstruction error worse and a clear inverse relationship between the KL-divergence weighting factor and the reconstruction precision seems to be present in the implemented models.

The dataset sensitivity of gaussian processes regression and PCA has been investigated. The gaussian process seems to be able to take advantage of lower number of samples effectively. The PCAs are also not only able to make training reconstructions effectively at lower number of samples, but converge to a very similar performance as the full dataset at only a couple of hundred examples of training data, further testatmenting their usefulness in capturing the field relations.

Gaussian process regression has also shown to be effective at utilizing the latent space of PCA decompositions, converging on the full-dimensional prediction error at three orders of magnitude fewer dimensions. Gaussian process regression also seems very good at learning a mapping between the antenna parameters and the autoencoder's latent space. Making it able to extend the unsupervised autoencoder model to a a semi-supervised model that is also able to make parameter-to-field predictions.

6 Discussion and Conclusion

In this thesis, we set out to better understand how to find good surrogate models that can appropriately describe the far-field radiation pattern of a given antenna given its parameterization within a known design space. We wanted to know how unsupervised latent space methods can describe sampled far-field datasets and to understand how dataset sizes constrain both these supervised and unsupervised models. We conclude that the trained surrogates significantly outperform the 'dummy' baselines for all datasets and that some of the models achieve extremely high accuracy. Latent space models such as PCA and autoencoders have shown to be very effective at dimensionality reduction of sampled far-fields, effectively reducing their dimensionality by up to 3 orders of magnitude while preserving a usable amount of information in the reconstructed fields. Variational autoencoder methods were investigated, but these investigations did not find any advantages over regular autoencoders. Gaussian processes and PCA seemed to have plenty of data, reaching similar accuracies at significantly smaller dataset sizes than where available. Finally, PCA's linear decompositions of latent spaces have shown to be just as learnable for GP models as the full dimensional fields. In this chapter, we will discuss these points further and discuss the implication of the findings.

6.1 Latent Space Models

A deep relationship between the latent spaces and the parameterizations of the antennas is present in all datasets. Only the RFLCT dataset has two parameters that do not show any pattern in the latent space. It seems that all the datasets, in reality, can be represented by latent manifolds, which have a significantly smaller dimensionality than the original dataset. These manifolds also seem to be continuous to an extent where methods that are not specifically trained to be generative, such as PCA or autoencoders, have generative properties.

When one considers that the electromagnetic solvers make some non-linear projection between a low dimensional parameter space and a high dimensional far-field space, it is not surprising that we end up seeing this resulting manifold in the datasets. However, it is surprising that they does not take on higher dimensions and their structure is so readily obtainable by PCA decomposition.

We have also shown that regular autoencoders can 'unwrap' these manifolds into their intrinsic dimension as done in 5.1.3. Where we successfully unwrapped the CHA dataset manifold found by PCA into a two-dimensional representation that obtained the same reconstruction accuracy as a 19-dimensional PCA. We believe this to be further evidence pointing to the fact that the tested dataset seems to lie in a latent manifold. Only the CHA dataset has been tested by the autoencoder, but since the PCAs seem to express the PATCH and RFLCT as 2D surfaces and the MLA as a ten-dimensional hypersphere, there is reason to believe that appropriate autoencoders perform well in these cases as well. Therefore, all the datasets seem to be examples of the manifold hypothesis.

Variational autoencoders were able to recreate decent examples of far-fields. Though at the current implementation and use case, they only seem to prohibit the learning of the latent manifold. Often the advantage of variational autoencoders is that they can make the latent space representation more 'compact' by ensuring that the entire dataset takes on some a priori distribution. This is the feature that makes variational autoencoders generative instead of simply being able to decode the obtained latent space. In our case,

there were no problems obtaining latent generative models. Even a linear subspace such as PCA has shown to be highly generative in all datasets in fig. 5.1 since its validation loss is almost identical to the training loss for all latent space dimensions shown. We think, therefore, that in the current use case, the variational autoencoder decreased the reconstruction performance through the secondary KL-divergence goal, which is also seen clearly in section 5.1.4. Simply making it a less reconstruction-accurate version of the regular autoencoder.

6.2 Direct Surrogate Performance

Both neural network and gaussian process models have outperformed the baseline KNN regression model by more than an order of magnitude for the PATCH and CHA dataset. The PDNN networks outperformed the GP models in the PATCH case, but only by a relatively small margin. Gaussian processes have significantly outperformed the PDNN in the CHA and MLA datasets.

This is surprising, since from fundamentals neural networks should have a clear advantage over the SOGP models, because of their ability draw statistical power between output dimensions[47]. Though the answer might simply be that training neural networks can be a difficult task, both because of their model flexibility, but also because of the long training time. Perhaps neural networks would outperform gaussian processes in larger datasets.

None of the models seemed to be able to obtain any real predictive performance in the RFLCT dataset. We suspect this might be due to undersampling on the far-field sphere. Since the RFLCT dataset has very quickly varying fields on the far-field sphere, the actual underlying regression problem of predicting each far-field sample as a function of antenna parameters might become extremely non-linear if the sphere is undersampled. A similar problem might be the case for the MLA dataset, and a more careful analysis of the far-field sphere sampling density should be investigated as the first step in achieving better prediction performance.

The effectiveness of linear and non-linear dimensionality reduction techniques could make output-output dependant MOGP models more feasible since their time complexity does not allow for high-dimensional regression problems. But MOGPs might be possible when predicting into smaller latent spaces, similar to what we have shown in our semi-supervised models.

Since the PCA and autoencoders do not seem to be affected by the sampling density of the far-field sphere in the same manner that the PDNN and GP models do, they seem less far-field sample density sensitive. Maybe because they fundamentally abstract the prediction problem of antenna parameter to a single point on the far field sphere away and obtain a more general understanding of the field. Since the latent methods can recreate and generate the entire dataset, which we have shown through their validation error, semi-supervised models similar to the ones we have proposed might be significantly more robust to far-field sample density.

Investigations in using fully connected nets for predicting into the autoencoder latent space have also been conducted, but at the time of writing, no real interesting results have come of this. Since the autoencoder performs significantly better at reconstructing the CHA far-fields than the PDNN does at direct prediction, it might be plausible that a semi-supervised AE-FC model can outperform the PDNN.

6.3 Novelty of This Thesis

At the time of writing, we are unaware of any other published literature that has used PCA analysis on sampled antenna far-field patterns. Surprisingly, simple linear decompositions can express a large part of the variance of sampled far-field patterns in relatively small latent spaces. Furthermore, the fact that these latent space geometries seem continuous and express themselves as even lower-dimensional manifolds is a novel finding. Using PCA analysis in surrogate models is not novel, though, and has been done in microwave surrogate-modeling before, like in [59]. Implicitly far-field PCA has also been done in [60], but not directly on sampled far-fields of the type used in this thesis.

Using autoencoders on far-field samples in the same manner as they are used on images and using variational autoencoders on far-field samples has never been done in the literature to our knowledge. The semi-supervised autoencoder models of section 4.4 are therefore also novel in their architecture.

6.4 Implications

The understanding that sampled far-fields can be expressed in very low dimensional PCA-decompositions is surprising. Many electromagnetic expansions, such as the spherical wave expansions[61], exist that can express far-fields in a lower number of spherical waves. These are generalized expansions with a theoretical background ensuring the satisfaction of the Maxwell equations. However, the PCA seems significantly more effective at dimensionality reduction because of its dataset-specific decompositions. The effective dimensionality reduction of far-fields could be used to compress groups of far-field patterns from the same design space into significantly lower memory without losing much information. Additionally, PCA could make output-output correlated Gaussian process models, as discussed in section 4.2.2, feasible since they would not be as limited by the dataset dimensionality. Furthermore, the understanding that these PCA decompositions are generative within the design spaces and require significantly less data than has been provided in this project, and therefore than was expected, accounts for their usability and how good they are at generalizing.

With the successful training of multiple far-field surrogate models, these models could be used for optimization or simply model evaluations within their trained design spaces. This could be used to speed up the optimization convergence of antennas inside the design spaces.

The success of single output Gaussian processes as far-field surrogates is impressive, considering that they perform better than large neural networks in many cases. This might also indicate the remaining potential of neural networks as surrogates, as they should be able to draw statistical power between outputs and from a fundamentals perspective, therefore should be able to outperform SOGP models.

6.5 Conclusion

Neural networks and Gaussian processes can be trained to be very accurate far-field prediction models, on the CHA and PATCH datasets. The RFLCT and MLA datasets seemed to be harder problems; this seems to be a combination of under-sampling in the far-field sphere making each prediction dimensionality more non-linear, the fact that both antennas are higher gain and have more radical relationships to the parameters, and in the MLA case, the curse of dimensionality because of the large number of antenna parameters.

Latent space methods have shown to be very efficient at dimensionality reduction of the sampled far-field datasets, reducing the dimensionality of the datasets by up to 3 orders of magnitude while still being able to reconstruct the fields satisfactorily. The far-field seems to be projected onto a latent manifold that is significantly related to the parameterizations. These far-field latent projections seem to form lower-dimensional manifolds within the latent PCA spaces. Empirically autoencoders seem to be able to reduce the dimensionality of the datasets down to the intrinsic dimensionality of these manifolds since they can learn a mapping that can 'flatten' them.

Both PCA and GP models perform quite well at smaller dataset sizes. It is especially surprising that not only do the PCAs act generatively on the complete datasets but reach generative properties at only a couple of hundred far-field examples for all datasets

The effectiveness of PCA can also be used in conjunction with the Gaussian process models, where we have shown that the GP achieve similar performance in 10s of dimensions of the latent space, which is then reconstructed into the full far-field, as the GP models do when they predict directly into the higher dimensional space. This could make output-output dependant Gaussian process models significantly more feasible. Furthermore, Gaussian process models can also predict from design space into the latent space of an 2D autoencoder, getting nearly identical performance compared with the pure reconstruction loss of the autoencoder itself and the same performance as a 19-component PCA-GP.

6.6 Future Work

Fully connected pyramidal networks work well as surrogates but probably have a lot of wasteful parameters, as is often seen in wide, fully connected networks of the same type. More cutting-edge machine learning methods often favor very deep networks, using methods such as the ResNet[62] method to get over the *degradation* problem as depth increases. It would be very interesting if a significant performance increase at the same learnable parameter count can be obtained by making thinner and deeper PDNN networks using residual networks or similar techniques.

Since some of the very high gain antenna patterns, like the RFLCT and MLA datasets, seem insufficiently sampled in far-field. Investigations into versions of the same dataset with higher far-field sampling density would be interesting to test this hypothesis.

Mixing different antenna datasets with the same sampled far-field dimensionality and investigating their latent space could give insights into how well the latent spaces generalize. Would they distribute themselves as individual clouds, and if so, would interpolation between these clouds give anything meaningful? The use of variational autoencoders might be advantageous in these multi-design-space datasets.

This idea also lends itself to the potential of knowledge transfers between design spaces. Since most far-fields have similar structures, one could imagine there is potential to boost the training of new design spaces with previously obtained surrogates.

As outlined in section section 2.3, there is always implicit bias when sampling a spherical function. In this thesis, we have only outlined a set of concerns regarding this and used a simple θ -cut based method. Further investigations should be made into the relationship between the far-field sample densities and model performance. As noted in chapter 3, there exist theories for electromagnetically justified sampling densities, but just like how neural networks can gain performance from oversampling of time-series beyond Nyquist sampling, NN models might also perform better with over-sampled far-fields. This higher

sampling density could also lead to more justification for using convolutional nets instead of fully connected networks.

Deriving generally applicable physics-based loss functions for physics-informed neural networks, described in appendix E, methods could also help the models converge faster and improve interpolation and extrapolation, such that less data is needed to train the surrogate models. This would generally require continuous space neural network representations, which might, due to the curse of dimensionality, make the regression problems more difficult. Therefore operator-based neural network models such as neural operator networks [63] could have the potential to be able to solve the difficulty of direct continuous regression by learning continuous operators instead.

Bibliography

- [1] IEEE. "IEEE Standard for Definitions of Terms for Antennas". In: (2014). DOI: 10.1109/IEEESTD.2014.6758443.
- [2] By G Christos Christodoulou et al. "Reconfigurable Antennas for Wireless and Space Applications". In: (2012). DOI: 10.1109/JPROC.2012.2188249.
- [3] Jeffrey G Andrews et al. "What Will 5G Be?" In: *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS* 32.6 (2014). DOI: 10.1109/JSAC.2014.2328098. URL: http://www.ieee.org/publications_standards/publications/rights/index.html.
- [4] Jayavardhana Gubbi et al. "Future Generation Computer Systems Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future Generation Computer Systems* 29 (2013), pp. 1645–1660. DOI: 10.1016/j.future.2013.01.010. URL: <http://dx.doi.org/10.1016/j.future.2013.01.010>.
- [5] Constantine A. Balanis. *Antenna Theory: Analysis and Design*. 4th. Wiley, 2016. ISBN: 978-1-118-642060.
- [6] Slawomir Koziel and Anna Pietrenko-Dabrowska. "Knowledge-based performance-driven modeling of antenna structures". In: *Knowledge-Based Systems* 237 (Feb. 2022). ISSN: 09507051. DOI: 10.1016/J.KNOSYS.2021.107698.
- [7] Timothy Truckle. *Far-Field Pattern CC License*. URL: shorturl.at/kp368.
- [8] John Proakis and Dimitris Manolakis. *Digital Signal Processing*. 4th ed. Vol. 1. Pearson, 2006.
- [9] Markus Deserno. "How to generate equidistributed points on the surface of a sphere". In: *Max-Planck-Institut für Polymerforschung*, (2004).
- [10] Seppo Jarvenpaa and Pasi Yla-Oijala. "A Global Interpolator With Low Sample Rate for Multilevel Fast Multipole Algorithm". In: *IEEE Transactions on Antennas and Propagation* 61.3 (Mar. 2013), pp. 1291–1300. ISSN: 0018-926X. DOI: 10.1109/TAP.2012.2231927. URL: <http://ieeexplore.ieee.org/document/6374227/>.
- [11] M. D. McKay, R. J. Beckman, and W. J. Conover. "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code". In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 15372723. DOI: 10.1080/00401706.1979.10489755.
- [12] Slawomir Koziel and Anna Pietrenko-Dabrowska. *Performance-Driven Surrogate Modeling of High-Frequency Structures*. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-38925-3. DOI: 10.1007/978-3-030-38926-0. URL: <http://link.springer.com/10.1007/978-3-030-38926-0>.
- [13] Slawomir Koziel and Adrian Bekasiewicz. "Inverse surrogate models for fast geometry scaling of miniaturized dual-band couplers". In: *2016 11th European Microwave Integrated Circuits Conference (EuMIC)*. IEEE, Oct. 2016, pp. 293–296. ISBN: 978-2-87487-044-6. DOI: 10.1109/EuMIC.2016.7777548. URL: <http://ieeexplore.ieee.org/document/7777548/>.
- [14] *Latin Hypercube Sampling — SciPy v1.8.0 Manual*. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.qmc.LatinHypercube.html#id1>.
- [15] Peter Fortescue, Graham Swinerd, and John Stark. *Spacecraft Systems Engineering*. 2011. ISBN: 9780470750124.
- [16] Misao Haneishi and Shinichiro Yoshida. "A design method of circularly polarized rectangular microstrip antenna by one-point feed". In: *Electronics and Communications in Japan (Part I: Communications)* 64.4 (Apr. 1981), pp. 46–54. ISSN: 87566621. DOI: 10.1002/ecja.4410640407.

- [17] John W. Bandler et al. "Space mapping: The state of the art". In: *IEEE Transactions on Microwave Theory and Techniques* 52.1 II (Jan. 2004), pp. 337–361. ISSN: 00189480. DOI: 10.1109/TMTT.2003.820904.
- [18] M.H. Bakr et al. "Space mapping optimization of microwave circuits exploiting surrogate models". In: *2000 IEEE MTT-S International Microwave Symposium Digest (Cat. No.00CH37017)*. Vol. 3. IEEE, 2000, pp. 1785–1788. ISBN: 0-7803-5687-X. DOI: 10.1109/MWSYM.2000.862325. URL: <http://ieeexplore.ieee.org/document/862325/>.
- [19] Slawomir Koziel and Stanislav Ogurtsov. "Rapid optimization of dielectric resonator antennas using surrogate models". In: *LAPC 2011 - 2011 Loughborough Antennas and Propagation Conference* (2011). DOI: 10.1109/LAPC.2011.6114020.
- [20] S. Koziel. "Robust optimization of microwave structures using co-simulation-based surrogate models". In: *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, July 2011, pp. 2924–2927. ISBN: 978-1-4244-9563-4. DOI: 10.1109/APS.2011.5997140. URL: <http://ieeexplore.ieee.org/document/5997140/>.
- [21] Slawomir Koziel and Stanislav Ogurtsov. "Fast design of microstrip antenna arrays exploiting surrogate models". In: *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*. IEEE, Apr. 2014, pp. 3429–3431. ISBN: 978-8-8907-0184-9. DOI: 10.1109/EuCAP.2014.6902566. URL: <http://ieeexplore.ieee.org/document/6902566/>.
- [22] Brandt Klopper and Dirk I. L. de Villiers. "Space mapping optimization of aperture-coupled patch antennas using circuit models". In: *2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, June 2016, pp. 289–290. ISBN: 978-1-5090-2886-3. DOI: 10.1109/APS.2016.7695853. URL: <http://ieeexplore.ieee.org/document/7695853/>.
- [23] Igor A. Baratta et al. "Infinitesimal Dipole Model Using Space Mapping Optimization for Antenna Placement". In: *IEEE Antennas and Wireless Propagation Letters* 17.1 (Jan. 2018), pp. 17–20. ISSN: 1536-1225. DOI: 10.1109/LAWP.2017.2771721. URL: <http://ieeexplore.ieee.org/document/8100884/>.
- [24] David M. Pozar. *Microwave Engineering*. 4th ed. Wiley, 2012. ISBN: 978-0-470-63155-3.
- [25] Slawomir Koziel et al. "Accurate Modeling of Antenna Structures by Means of Domain Confinement and Pyramidal Deep Neural Networks". In: *IEEE Transactions on Antennas and Propagation* 70.3 (Mar. 2022), pp. 2174–2188. ISSN: 0018-926X. DOI: 10.1109/TAP.2021.3111299. URL: <https://ieeexplore.ieee.org/document/9538974/>.
- [26] Jinxin Du and Christophe Roblin. "Statistical Modeling of Disturbed Antennas Based on the Polynomial Chaos Expansion". In: *IEEE Antennas and Wireless Propagation Letters* 16 (2017), pp. 1843–1846. ISSN: 1536-1225. DOI: 10.1109/LAWP.2016.2609739.
- [27] Jinxin Du and Christophe Roblin. "Stochastic Surrogate Models of Deformable Antennas Based on Vector Spherical Harmonics and Polynomial Chaos Expansions: Application to Textile Antennas". In: *IEEE Transactions on Antennas and Propagation* 66.7 (July 2018), pp. 3610–3622. ISSN: 0018-926X. DOI: 10.1109/TAP.2018.2829820.
- [28] Wonil Roh et al. "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results". In: *IEEE Communications Magazine* 52.2 (Feb. 2014), pp. 106–113. ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6736750. URL: <http://ieeexplore.ieee.org/document/6736750/>.

- [29] Zhao Zhou et al. “Training of Deep Neural Networks in Electromagnetic Problems: a Case Study of Antenna Array Pattern Synthesis”. In: *2021 IEEE MTT-S International Wireless Symposium (IWS)*. IEEE, May 2021, pp. 1–3. ISBN: 978-1-6654-3527-7. DOI: 10.1109/IWS52775.2021.9499638. URL: <https://ieeexplore.ieee.org/document/9499638/>.
- [30] Wenchao Xia et al. “Model-Driven Beamforming Neural Networks”. In: *IEEE Wireless Communications* 27.1 (Jan. 2020), pp. 68–75. DOI: 10.1109/MWC.001.1900239. URL: <http://arxiv.org/abs/2001.05277%20http://dx.doi.org/10.1109/MWC.001.1900239>.
- [31] Michele Polese, Francesco Restuccia, and Tommaso Melodia. “DeepBeam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks; DeepBeam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks”. In: (). DOI: 10.1145/3466772.3467035. URL: <https://doi.org/10.1145/3466772.3467035>.
- [32] Ricardo Lovato and Xun Gong. “Phased Antenna Array Beamforming using Convolutional Neural Networks”. In: *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*. IEEE, July 2019, pp. 1247–1248. ISBN: 978-1-7281-0692-2. DOI: 10.1109/APUSNCURSINRSM.2019.8888573. URL: <https://ieeexplore.ieee.org/document/8888573/>.
- [33] Angelo Freni, Marco Mussetta, and Paola Pirinoli. “Neural Network Characterization of Reflectarray Antennas”. In: *International Journal of Antennas and Propagation* 2012 (2012), pp. 1–10. ISSN: 1687-5869. DOI: 10.1155/2012/541354. URL: <http://www.hindawi.com/journals/ijap/2012/541354/>.
- [34] Daniel R. Prado et al. “General Framework for the Efficient Optimization of Reflectarray Antennas for Contoured Beam Space Applications”. In: *IEEE Access* 6 (2018), pp. 72295–72310. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2882271. URL: <https://ieeexplore.ieee.org/document/8540382/>.
- [35] Daniel R. Prado, Jesús A. López-Fernández, and Manuel Arrebola. “Systematic Study of the Influence of the Angle of Incidence Discretization in Reflectarray Analysis to Improve Support Vector Regression Surrogate Models”. In: *Electronics* 9.12 (Dec. 2020), p. 2105. ISSN: 2079-9292. DOI: 10.3390/electronics9122105.
- [36] Theodoros N Kapetanakis et al. “Neural Network Modeling for the Solution of the Inverse Loop Antenna Radiation Problem”. In: *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION* 66.11 (2018), p. 6283. DOI: 10.1109/TAP.2018.2869136. URL: http://www.ieee.org/publications_standards/publications/rights/index.html.
- [37] Jae Hee Kim and Sang Won Choi. “A Deep Learning-Based Approach for Radiation Pattern Synthesis of an Array Antenna”. In: *IEEE Access* 8 (2020), pp. 226059–226063. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3045464. URL: <https://ieeexplore.ieee.org/document/9296770/>.
- [38] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: (2019).
- [39] *Complex Numbers — PyTorch 1.12 documentation*. URL: https://pytorch.org/docs/stable/complex_numbers.html.
- [40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539. URL: <http://www.nature.com/articles/nature14539>.
- [41] Karen Simonyan and Andrew Zisserman. *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. Tech. rep. 2015. URL: <http://www.robots.ox.ac.uk/>.

- [42] Adam — PyTorch 1.11.0 documentation. URL: <https://pytorch.org/docs/stable/generated/torch.optim.Adam.html>.
- [43] Diederik P Kingma and Jimmy Lei Ba. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. Tech. rep.
- [44] Hyperparameter Tuning - Documentation. URL: <https://docs.wandb.ai/guides/sweeps>.
- [45] W&B Hyperparameter Sweep Engine GitHub. URL: https://github.com/wandb/sweeps/blob/master/src/sweeps/bayes_search.py.
- [46] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Dec. 2005, p. 272. ISBN: 9780262256834. DOI: 10.7551/MITPRESS/3206.001.0001.
- [47] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2009.
- [48] David Duvenaud. “Automatic Model Construction with Gaussian Processes”. PhD thesis. University of Cambridge, 2014.
- [49] Learn Scikit. `sklearn.gaussian_process.GaussianProcessRegressor` — scikit-learn 1.1.1 documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html.
- [50] 1.7. Gaussian Processes — scikit-learn 1.1.1 documentation. URL: https://scikit-learn.org/stable/modules/gaussian_process.html.
- [51] Jacob R Gardner et al. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. Tech. rep. URL: <https://gpytorch.ai..>
- [52] Haitao Liu, Jianfei Cai, and Yew Soon Ong. “Remarks on multi-output Gaussian process regression”. In: *Knowledge-Based Systems* 144 (Mar. 2018), pp. 102–121. ISSN: 0950-7051. DOI: 10.1016/J.KNOSYS.2017.12.034.
- [53] Bo Wang and Tao Chen. “Gaussian process regression with multiple response variables”. In: *Chemometrics and Intelligent Laboratory Systems* 142 (Mar. 2015), pp. 159–165. ISSN: 0169-7439. DOI: 10.1016/J.CHEMOLAB.2015.01.016.
- [54] Lawrence Cayton. “Algorithms for manifold learning”. In: (2005).
- [55] scikit-learn. PCA — 1.1.1 documentation. 2022. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>.
- [56] Mark A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 37.2 (Feb. 1991), pp. 233–243. ISSN: 1547-5905. DOI: 10.1002/AIC.690370209. URL: <https://onlinelibrary-wiley-com.proxy.findit.cvt.dk/doi/full/10.1002/aic.690370209%20https://onlinelibrary-wiley-com.proxy.findit.cvt.dk/doi/abs/10.1002/aic.690370209%20https://aiche-onlinelibrary-wiley-com.proxy.findit.cvt.dk/doi/10.1002/aic.690370209>.
- [57] Stephen Blundell and Katherine Blundell. *Concepts in Thermal Physics*. Second Edition. Oxford, 2017. ISBN: 978-0-19-956209-1.
- [58] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: (Dec. 2013). URL: <http://arxiv.org/abs/1312.6114>.
- [59] Slawomir Koziel, Anna Pietrenko-Dabrowska \$, and John W Bandler. *Computationally Efficient Performance-Driven Surrogate Modeling of Microwave Components Using Principal Component Analysis; Computationally Efficient Performance-Driven Surrogate Modeling of Microwave Components Using Principal Component Analysis*. 2020. ISBN: 9781728168159.
- [60] Min Liang et al. “Reconfigurable Array Design to Realize Principal Component Analysis (PCA)-Based Microwave Compressive Sensing Imaging System; Reconfigurable Array Design to Realize Principal Component Analysis (PCA)-Based Microwave Compressive Sensing Imaging System”. In: *IEEE ANTENNAS AND WIRES*

- LESS PROPAGATION LETTERS* 14 (2015). DOI: 10.1109/LAWP.2014.2386356. URL: http://www.ieee.org/publications_standards/publications/rights/index.html.
- [61] J.E. Hansen et al. *Spherical Near-Field Antenna Measurements*. Ed. by J.E. Hansen. 1988. ISBN: 0 86341 110.
 - [62] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: (Dec. 2015).
 - [63] Lu Lu, Pengzhan Jin, and George Em Karniadakis. “DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators”. In: (Oct. 2019). DOI: 10.1038/s42256-021-00302-5. URL: <http://arxiv.org/abs/1910.03193%20http://dx.doi.org/10.1038/s42256-021-00302-5>.
 - [64] Constantin A. Balanis. *Advanced Engineering Electromagnetics*. 2nd. Wiley, 2012. ISBN: 978-0-470-58948-9.
 - [65] David Cheng. *Field and Wave Electromagnetics*. 2nd. Pearson, 1989.
 - [66] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations”. In: (Nov. 2017). URL: <http://arxiv.org/abs/1711.10561>.
 - [67] *A Gentle Introduction to torch.autograd — PyTorch Tutorials 1.11.0+cu102 documentation*. URL: https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html.
 - [68] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. ISSN: 00219991. DOI: 10.1016/j.jcp.2018.10.045. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125>.
 - [69] Lu Lu et al. “DeepXDE: A Deep Learning Library for Solving Differential Equations”. In: *SIAM Review* 63.1 (Jan. 2021), pp. 208–228. ISSN: 0036-1445. DOI: 10.1137/19M1274067. URL: <https://pubs.siam.org/doi/10.1137/19M1274067>.
 - [70] Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. “Locally adaptive activation functions with slope recovery term for deep and physics-informed neural networks”. In: (Sept. 2019). DOI: 10.1098/rspa.2020.0334. URL: <http://arxiv.org/abs/1909.12228%20http://dx.doi.org/10.1098/rspa.2020.0334>.
 - [71] Yeonjong Shin, Jerome Darbon, and George Em Karniadakis. “On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs”. In: (Apr. 2020). DOI: 10.4208/cicp.OA-2020-0193. URL: <http://arxiv.org/abs/2004.01806%20http://dx.doi.org/10.4208/cicp.OA-2020-0193>.
 - [72] Arbaaz Khan and David A. Lowther. “Physics Informed Neural Networks for Electromagnetic Analysis”. In: *IEEE Transactions on Magnetics* (2022), pp. 1–1. ISSN: 0018-9464. DOI: 10.1109/TMAG.2022.3161814. URL: <https://ieeexplore.ieee.org/document/9740189/>.
 - [73] Ruohan Gong and Zuqi Tang. “Further investigation of convolutional neural networks applied in computational electromagnetism under physics-informed consideration”. In: *IET Electric Power Applications* 16.6 (June 2022), pp. 653–674. ISSN: 1751-8660. DOI: 10.1049/elp2.12183. URL: <https://onlinelibrary.wiley.com/doi/10.1049/elp2.12183>.

A Background Theory

This section will go over the well-established theory and prerequisites to understand the rest of the thesis from an electromagnetic standpoint. Firstly, we will introduce the basics of vector-based electromagnetic theory, deriving the relevant solutions to Maxwell's equations. We will discuss the far-field approximation, which is applied to all datasets modeled in this thesis and we will go over some of the notation and values of merit often encountered in antenna theory, which is also used here. Then finally, we will formally define the general expressions for the sampled far-field, utilizing the theory introduced in the previous sections.

A.1 Electromagnetic Fields

Surrogate machine learning models are designed to emulate the behavior of physical systems. In this thesis, we are interested in emulating the predicted behavior of antenna systems. Therefore, the fundamental aspect of electromagnetics and antenna theory is necessary to understand what we are modeling and how to interpret the responses from numerical simulations and predictions from the surrogate models.

This section will establish the fundamental quantities of electromagnetics, derive the free space wave equation, and briefly use it to show common forms of solutions used in antenna modeling. Then we will go over the electromagnetic field concepts used in this thesis and, in detail, establish the mathematical structure of the antenna data modeled throughout the thesis, building on the previously introduced theory.

The goal of this chapter is to make the reader acquainted with the duality between mapping current distributions to far-fields $\mathbf{J} \rightarrow \mathbf{F}$, which is what numerical solvers do, and mapping antenna design spaces to far fields, $\Omega_D \rightarrow \mathbf{F}$, which is the goal of this thesis. But also to introduce the mathematical forms sampled electromagnetic far-fields take to better understand them in future sections.

A.1.1 The Maxwell Equations, Helmholtz Equations, and Their Solutions

The four governing postulates of electromagnetics are the Maxwell equations; in their most general form, they are

$$\nabla \times \mathcal{E} = -\mathcal{M} - \frac{\partial \mathcal{B}}{\partial t} \quad (\text{A.1})$$

$$\nabla \times \mathcal{H} = \mathcal{J} + \frac{\partial \mathcal{D}}{\partial t} \quad (\text{A.2})$$

$$\nabla \cdot \mathcal{D} = \varrho_e \quad (\text{A.3})$$

$$\nabla \cdot \mathcal{B} = \varrho_m \quad (\text{A.4})$$

Where \mathcal{E} and \mathcal{H} are the electric and magnetic field intensities, \mathcal{D} , \mathcal{B} are the electric and magnetic flux densities, \mathcal{J} and \mathcal{M} are the electric and magnetic current densities and ϱ_e and ϱ_m are the electric and magnetic charge densities.

For strictly time harmonic fields in homogeneous, linear and isotropic media, and with a time factor of $e^{j\omega t}$, these can be expressed in complex phasor notation¹ as:

¹Phasor notation is briefly summarized in appendix B.

$$\nabla \times \mathbf{E} = -\mathbf{M} - j\omega\mu\mathbf{H} \quad (\text{A.5})$$

$$\nabla \times \mathbf{H} = \mathbf{J} + j\omega\epsilon\mathbf{E} \quad (\text{A.6})$$

$$\nabla \cdot \mathbf{E} = \frac{\rho_e}{\epsilon} \quad (\text{A.7})$$

$$\nabla \cdot \mathbf{H} = \frac{\rho_m}{\mu} \quad (\text{A.8})$$

These equations describe all macroscopic electromagnetic phenomena and relate electric fields, magnetic fields, currents, and charges into one unified theory of *electromagnetics*.

As can be seen in eq. (A.6), a time harmonic current distribution (such as an antenna), \mathbf{J} , induces a time harmonic magnetic field, \mathbf{H} , which also has to have an equivalent electric field, \mathbf{E} as seen in eq. (A.5). These current-induced electric and magnetic fields exist outside this current distribution, and in this free space ($\mathbf{J} = \mathbf{M} = \rho_e = \rho_m = 0$) combining eq. (A.5) with eq. (A.8) and eq. (A.6) with eq. (A.7) we can decouple the \mathbf{E} and \mathbf{H} fields by combining them into a higher order differential equation, the Helmholtz wave equation.

$$\nabla^2 \mathbf{E} = -\beta^2 \mathbf{E} \quad (\text{A.9})$$

$$\nabla^2 \mathbf{H} = -\beta^2 \mathbf{H} \quad (\text{A.10})$$

Where the wavenumber $\beta = \omega^2\epsilon_0\mu_0$. Thus the fields, \mathbf{E} and \mathbf{H} , must be solutions to this partial differential vector field equation. Decomposing eq. (A.9), we can write it as three scalar differential equations.

$$\nabla^2 E_x(x, y, z) + \beta^2 E_x(x, y, z) = 0 \quad (\text{A.11a})$$

$$\nabla^2 E_y(x, y, z) + \beta^2 E_y(x, y, z) = 0 \quad (\text{A.11b})$$

$$\nabla^2 E_z(x, y, z) + \beta^2 E_z(x, y, z) = 0 \quad (\text{A.11c})$$

Solutions to these equations can be found by a 'separation of variables' and guessing a solution for eq. (A.11a) on the form:

$$E_x(x, y, z) = f(x)g(y)h(z) \quad (\text{A.12})$$

We can combine eq. (A.11a) and eq. (A.12) into

$$\frac{1}{f(x)} \frac{d^2 f(x)}{dx^2} + \frac{1}{g(y)} \frac{d^2 g(y)}{dy^2} + \frac{1}{h(z)} \frac{d^2 h(z)}{dz^2} = -\beta^2 \quad (\text{A.13})$$

Here each term is a function of only a single independent variable, and thus they must all equate their separate wave number.

$$\begin{aligned} \frac{d^2 f(x)}{dx^2} &= -\beta_x^2 f(x) \\ \frac{d^2 g(y)}{dy^2} &= -\beta_y^2 g(y) \\ \frac{d^2 h(z)}{dz^2} &= -\beta_z^2 h(z) \end{aligned} \quad (\text{A.14})$$

Which then must uphold

$$\beta_x^2 + \beta_y^2 + \beta_z^2 = \beta^2 \quad (\text{A.15})$$

Where each of the variable solutions can take multiple forms, though we will only focus on one of the most common solutions; the traveling wave solution.²

$$f(x) = A_x e^{-j\beta_x x} + B_x e^{j\beta_x x} \quad (\text{A.16a})$$

$$g(y) = A_y e^{-j\beta_y y} + B_y e^{j\beta_y y} \quad (\text{A.16b})$$

$$h(z) = A_z e^{-j\beta_z z} + B_z e^{j\beta_z z} \quad (\text{A.16c})$$

Where A and B are complex constants to be found through the boundary conditions of the antenna and $\beta_x, \beta_y, \beta_z$ are the wavenumbers. These solutions can be shown to be travelling waves, with $e^{-j\beta_x x}$ indicating a $(+x)$ -propagating wave and $e^{j\beta_x x}$ indication a $(-x)$ -propagating wave. [65]

In this case, we have shown the solution based on the x-component of the electric field. Still, from symmetry, it can easily be seen that the same methods and solutions can readily be applied to the other components, and that the magnetic Helmholtz equation (A.10) has similar solutions.

Here we have shown the solution for the Cartesian coordinate system, though all fields from finite current sources are radially radiating, and therefore propagate outward in a sphere. The spherical wave solutions are much more cumbersome, and as we shall see in appendix A.1.2, we can approximate these fields on a form very similar to eq. (A.16).

In conclusion, we have shown that a time-harmonic current density, J , will lead to a time-harmonic electric and magnetic fields E , and H , we have then shown that these fields can behave as traveling waves, which are written on the form eq. (A.16).

A.1.2 Far Fields

Since all fields modeled in this thesis are far-fields, we will spend a little time understanding the assumptions that lead to the far-field approximation.

Even the simplest current configuration has quite complex, non-orthogonal electromagnetic fields³, though their fields become local plane waves on the sphere at large radii compared with the size of the radiating structure. Usually, the far-field is defined through the distance from the antenna, r , by the Rayleigh distance [5, p. 36].

$$r > \frac{2D^2}{\lambda} \quad (\text{A.17})$$

Where D is the physical size of the largest dimensions of the antenna and λ is the wavelength of the E-field. In this region, we can assume that all field components are orthogonal to the direction of propagation, $+r$, with the antenna positioned in the center of the coordinate system. The electric far field can therefore be written on the form.

$$\mathbf{E}_F(r, \theta, \phi) = \hat{a}_\theta E_\theta e^{j\varphi_\theta} \frac{e^{-j\beta r}}{r} + \hat{a}_\phi E_\phi e^{j\varphi_\phi} \frac{e^{-j\beta r}}{r} \quad (\text{A.18})$$

Where \hat{a}_θ and \hat{a}_ϕ are the spherical basis vectors. Thus no field is aligned with the direction of propagation \hat{a}_r .

Where E_θ and E_ϕ are the magnitudes of the electric far-field components, and φ_θ and φ_ϕ are their respective phases.

²A more detailed version of this derivation can be found in [64]

³For the interested reader, the field of the simplest current configuration, a Hertzian dipole, is derived in appendix C

The \mathbf{E} field has to take an inverse relation with the radial distance because the power $P \propto |\mathbf{E}|^2$, has to have a squared inverse relation according to energy conservation. The orthogonality of the θ and ϕ components makes many of the relatively complex relations derived from the Maxwell equations significantly simpler.

This electric far field is still dependant on both the radial distance from the antenna, and the frequency of the field through β . We, therefore, define the radially normalized far-field, \mathbf{F} as

$$\boxed{\mathbf{E}_F(r, \theta, \phi) = \mathbf{F}(\theta, \phi) \frac{e^{-j\beta r}}{r}} \quad (\text{A.19})$$

And thus, the far-field in this thesis has the properties of frequency Independence, radial independence, and field-orthogonality to the original radial direction of propagation. Therefore, we can define this field as done in eq. (1.1).

$$\mathbf{F}(\theta, \phi) \in \mathbb{C}^2 \text{ for } \theta \in [0, \pi], \phi \in [0, 2\pi] \quad (\text{A.20})$$

And thus, we have shown how currents in the Maxwell equation (A.5-A.8) lead to propagating waves. By using the far-field approximation, we have shown that these fields are very close in form to eq. (A.16) and that by normalizing these fields, we arrive at the complex vector field on a sphere defined in the problem statement.

The intuition to draw from this is that the far fields \mathbf{F} eq. (1.1) are derived from the traveling waves, which have a direct relation to the currents \mathbf{J} , whose distribution we are indirectly modeling through the parameterization χ of the antenna models. I.e., in this thesis, we are indirectly making the mapping that leads to solutions, as seen above.

Another essential point is that even though we are talking about electromagnetic fields, the Maxwell equations fully connect the electric and magnetic fields. Thus, only one or the other is sufficient to define a unique field solution in a known medium. Therefore, we will generally only be discussing and modeling the electric fields.

A.1.3 Electromagnetic Field Polarisation

Since the far-field has two components orthogonal to the direction of propagation as seen in ??, the phase-magnitude relationship between these two components gives rise to the far-fields polarization [65].

Practically, polarization is also attributed to individual antennas by extension of their radiated fields polarization. If two antennas are transmitting with different polarization, the energy that is polarized orthogonally to the antenna polarization will be lost. I.e., if we imagine having a transmitting antenna with a polarization vector $\hat{\rho}_w$ and a receiving antenna with the polarization vector $\hat{\rho}_a$ with some fields.

$$\mathbf{E}_i = \hat{\rho}_w E_i \quad (\text{A.21})$$

$$\mathbf{E}_a = \hat{\rho}_a E_a \quad (\text{A.22})$$

The *polarization loss factor* is then given by

$$PLF = |\hat{\rho}_w \cdot \hat{\rho}_a|^2 \quad (\text{A.23})$$

The polarization of an antenna can be expressed by two components as a point on a unit-sphere called a Poincaré sphere [64], where the inner product can be defined as the distance between two points on the sphere. *This makes us able to express the field on a polarization-based basis.*

This idea leads to the concepts of *co-polarization* and *cross-polarization*, where co-Polarization is the polarization that is aligned to a chosen reference polarization $\hat{\rho}_{ref}$ such that

$$|\hat{\rho}_{co} \cdot \hat{\rho}_{ref}|^2 = 1 \quad (\text{A.24})$$

And conversely, *Cross-Polarization* is orthogonal to the reference polarization.

$$|\hat{\rho}_{cross} \cdot \hat{\rho}_{ref}|^2 = 0 \quad (\text{A.25})$$

Thus we can use these properties to make a basis change of an EM field from a spatially dependant one, whose components do not reveal much intuition about the performance of an antenna, to a polarization dependant basis that displays information about how close out antenna is to our chosen reference polarization ρ_{ref} . This polarization-derived basis is what is used throughout this thesis.

A.1.4 Antenna Performance Characteristics

Since the analytical field expressions are almost impossible to obtain for most practical antenna systems, many metrics have been developed to understand the performance of antenna systems that can be found from simulations or through measurements. All definitions used are from the IEEE 2014 standard of Antennas[1] outlined in a form adapted from [5].

The far-field radiation intensity from a propagating antenna can be defined as [5, eq. 2-12a]

$$U(\theta, \phi) = r^2 W_{rad} = \frac{r^2}{2\eta} |\mathbf{E}_F(r, \theta, \phi)|^2 \quad (\text{A.26})$$

Where η is the free space intrinsic impedance. The total energy radiated from an antenna can then be found from

$$P_{rad} = \int_0^{2\pi} \int_0^\pi U(\theta, \phi) \sin \theta d\theta d\phi \quad (\text{A.27})$$

And the *directivity*, D , is then defined as the power normalized radiation intensity.

$$D(\theta, \phi) = \frac{4\pi U(\theta, \phi)}{P_{rad}} \quad (\text{A.28})$$

The directivity is, therefore, a normalized power measure that describes what percentage of an antenna's total radiated energy is directed in a particular direction.

To account for multiple loss mechanisms, the term *gain* is introduced as the directivity of an antenna with the loss mechanisms of the antenna accounted for. Here we are using the definition of 'realized gain,' which we will refer to as 'gain' interchangeably throughout this thesis.

$$G(\theta, \phi) = D(\theta, \phi) e_c e_d (1 - \Gamma^2) \quad (\text{A.29})$$

Where e_c and e_d are the conduction and dielectric loss factors, and Γ is the voltage reflection coefficient of the antenna at its input terminals. These added coefficient values

can be considered the loss factors between the power used to excite an antenna and the power that will radiate away from the antenna. We will only model antennas with constant excitation and implicitly model these loss factors as functions of the antenna parameters. And the loss factors can even, in some cases, be included as antenna parameters.

A.1.5 Datasets Field Expressions

This section will provide a formal mathematical form of how the datasets modeled in this thesis are related to the theory shown appendix A. We will only be considering time-harmonic, frequency normalized, and power normalized far-fields written on a polarization-based orthogonal basis. I.e. far-fields on the form.

$$\mathbf{F}(\theta, \phi) = \hat{\rho}_{co} E_{co} e^{j\varphi_{co}} + \hat{\rho}_{cross} E_{cross} e^{j\varphi_{cross}} \quad (\text{A.30})$$

Where E_{co} and E_{cross} are scalar magnitudes, φ_{co} and φ_{cross} are their respective phases and $\hat{\rho}_{co}$ and $\hat{\rho}_{cross}$ are the co and cross polarization orthonormal basis vectors. E_{co} , E_{cross} , φ_{co} and φ_{cross} are functions of θ and ϕ . Thus as a sampled discretization of the field, the far-field takes the form:

$$\mathbf{F}_s[\theta_i, \phi_j] = \hat{\rho}_{co} E_{co}[\theta_i, \phi_j] e^{j\varphi_{co}[\theta_i, \phi_j]} + \hat{\rho}_{cross} E_{cross}[\theta_i, \phi_j] e^{j\varphi_{cross}[\theta_i, \phi_j]} \quad (\text{A.31})$$

Where we have adapted sampling notation from signal processing like in [8], where $[\cdot, \cdot]$ indicates the function input of a sampled version of a continuous function, with the function input (\cdot, \cdot) .

In eq. (A.31) θ_i, ϕ_j are the scalar values of the θ, ϕ vectors, with $i \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, M\}$. In this thesis we therefore have sampled far fields of dimensionality $\mathbf{F}_s[\theta, \phi] \in \mathbb{C}^{2 \times N \times M}$, though for most of the machine learning methods we have chosen to represent it with its euclidean counterpart $\mathbf{F}_s[\theta, \phi] \in \mathbb{R}^{4 \times N \times M}$. In all fields that are modelled in this thesis, $N = 361$ and $M = 3$. Since usually a large part of the field variation is limited to θ .⁴

A single example of a discretized $\mathbf{F}_s \in \mathbb{R}^{4 \times N \times M}$, will be referred to as a *far-field sample* since it is a single far-field coming from a single parameterization. We will use the term *sample on the far-field sphere* if we are referring to a single value like $\mathbf{F}_s[\theta_i, \phi_j] \in \mathbb{R}^4$.

Since these fields are power normalized to the input power, we can also relate them to the antenna's gain⁵, as defined in eq. (A.29), through.

$$G_{co}(\theta, \phi) = |E_{co}|^2 \quad (\text{A.32})$$

$$G_{cross}(\theta, \phi) = |E_{cross}|^2 \quad (\text{A.33})$$

Which will mainly be shown in power decibels, i.e., on the form:

$$G_{co,dB}(\theta, \phi) = 20 \log_{10}(|E_{co}|) \quad (\text{A.34})$$

$$G_{cross,dB}(\theta, \phi) = 20 \log_{10}(|E_{cross}|) \quad (\text{A.35})$$

⁴Many examples of far fields from different antennas can be seen in [5], here they also follow the antenna theoretic customs of limit ϕ -samples, since not much variation happens here.

⁵A metric for how much power is radiated in a certain direction

Which will mostly be referred to on a form like $|E|$ dB.

Therefore when the electric field is shown in decibel, it is accounting for an expression like eq. (A.29), with losses included.⁶

Summary

This chapter has introduced the Maxwell equations and derived the wave equation solution for expressing far-field radiation. We introduced rewriting this electric far-field on a polarization-based basis and defined a set of power normalized performance characteristics. Lastly, we have used this theory to state the field expressions modeled in this thesis.

⁶With the use of decibels, the unfamiliar reader should be careful in their interpretation of the magnitude plots in dB and their relation to other metrics like mean squared error. Of course, a relationship does exist, but it will also be common for a significant amount of the 'real' error to be so close in log-space to the actual value that it will be hard to differentiate the two visually, especially in high-gain regions of the gain-pattern. Practically, thinking of 6dB as a power doubling, 12 as a quadrupling, and 20dB as an order of magnitude scaling can be helpful.

B Phasor Notation

Phasor notation is convenient for time harmonic fields, it is in practice as transformation between the time domain and spacial domain done by introducing a time factor, in electrical engineering it is always chosen to be $e^{j\omega t}$.

Phasor notation is best shown by example, if we have a time dependant electrical field

$$\mathcal{E}(t) = E_0 \cos(\omega t + \varphi) \quad (\text{B.1})$$

By convention its phasor is given by

$$\mathcal{E}(x, y, z; t) = \text{Re}[\mathbf{E}(x, y, z)e^{j\omega t}] \quad (\text{B.2})$$

$$\mathcal{H}(x, y, z; t) = \text{Re}[\mathbf{H}(x, y, z)e^{j\omega t}] \quad (\text{B.3})$$

$$\mathcal{D}(x, y, z; t) = \text{Re}[\mathbf{D}(x, y, z)e^{j\omega t}] \quad (\text{B.4})$$

$$\mathcal{B}(x, y, z; t) = \text{Re}[\mathbf{B}(x, y, z)e^{j\omega t}] \quad (\text{B.5})$$

Thus we are able to express time varying real valued vector fields spatially varying complex vector fields, which greatly simplifies many problems and makes notation simpler. By convention this is seen as a transformation and the real part of time factor are only included in the notation when casting the expression back into time domain.

C Fields from a hertzian dipole

When a time harmonic current exiting an antenna gives rise to a time harmonic electromagnetic field the field will have multiple components that have different orders of inverse proportionality to the radial distance, r . An example being the electromagnetic field arising from the simplest current configuration, the infinitesimal 'hertzian' dipole. This is an example adapted from [5].

$$\mathbf{I}(z') = \hat{\mathbf{a}}_z I_0 \quad (\text{C.1})$$

This current gives rise to the magnetic field $\mathbf{H} = \hat{\mathbf{a}}_r H_r + \hat{\mathbf{a}}_\theta H_\theta + \hat{\mathbf{a}}_\phi H_\phi$ where

$$\begin{aligned} H_r &= H_\theta = 0 \\ H_\phi &= j \frac{\beta I_0 l \sin \theta}{4\pi r} \left[1 + \frac{1}{j\beta r} \right] e^{-j\beta r} \end{aligned} \quad (\text{C.2})$$

And the electric field $\mathbf{E} = \hat{\mathbf{a}}_r E_r + \hat{\mathbf{a}}_\theta E_\theta + \hat{\mathbf{a}}_\phi E_\phi$ where

$$\begin{aligned} E_r &= \eta \frac{I_0 l \cos \theta}{2\pi r^2} \left[1 + \frac{1}{j\beta r} \right] e^{-j\beta r} \\ E_\theta &= j\eta \frac{\beta I_0 l \sin \theta}{4\pi r} \left[1 + \frac{1}{j\beta r} - \frac{1}{(\beta r)^2} \right] e^{-j\beta r} \\ E_\phi &= 0 \end{aligned} \quad (\text{C.3})$$

Here we see that even the simplest current distribution will have these higher order inverse components. These higher order components are often referred to as the reactive fields. At the near field region, $\beta r \ll 1$, these reactive fields dominate the total electric field close to the antenna, but further away they diminish quickly and for $\beta r \gg r$ we obtain field expression denoted as the *far field*. For example the field of the infinitesimal Hertzian dipole reduces to.

$$\begin{aligned} H_r &= H_\theta = 0 \\ H_\phi &= j \frac{\beta I_0 l \sin \theta}{4\pi r} e^{-j\beta r} \end{aligned} \quad (\text{C.4})$$

And the electric field

$$\begin{aligned} E_r &= E_\phi = 0 \\ E_\theta &= j\eta \frac{\beta I_0 l \sin \theta}{4\pi r} e^{-j\beta r} \end{aligned} \quad (\text{C.5})$$

Here we can see that the fields reduce to being purely orthogonal to each other and to the direction from propagation, that is they are *transverse electromagnetic fields* (TEM). All fields that will be discussed and modelled in this thesis are fields that are assumed to be in the far field region, and they are therefore of significant importance to understand that behaviour and relations that allow us to make simplifying assumptions.

D Single Point of the Far-Field Sphere Parameter Sweeps

D.1 PATCH

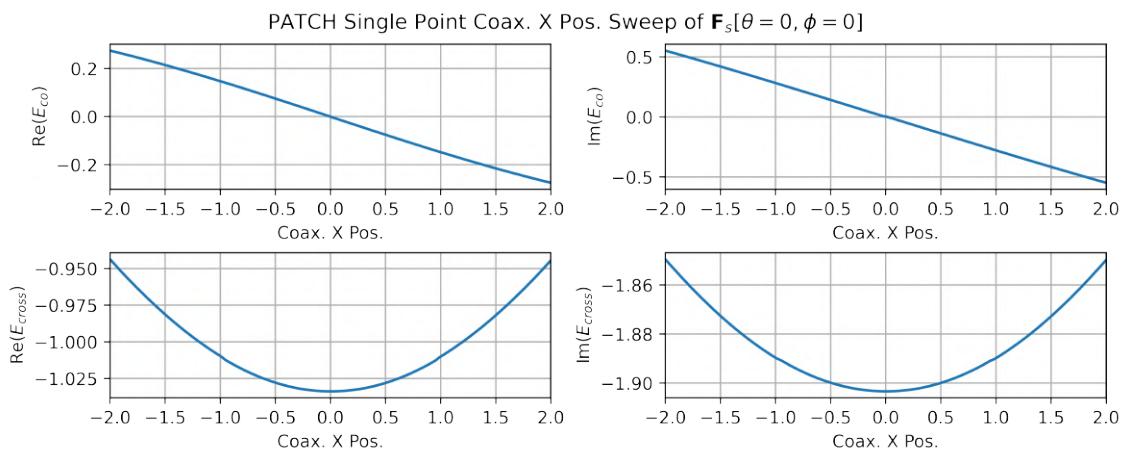


Figure D.1: Single point on the far-field sphere value as function of coaxial x placement in PATCH dataset

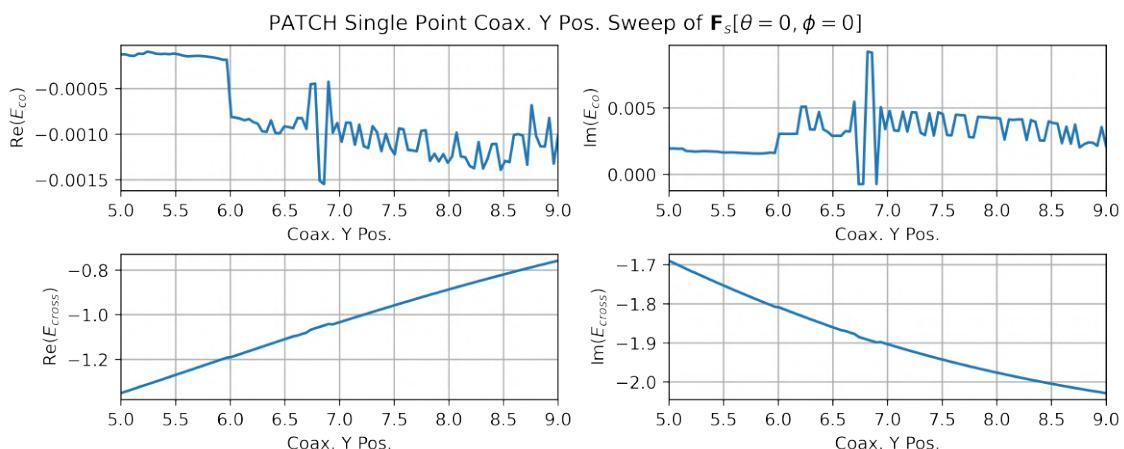


Figure D.2: Single point on the far-field sphere value as function of coaxial Y placement in PATCH dataset

D.2 CHA

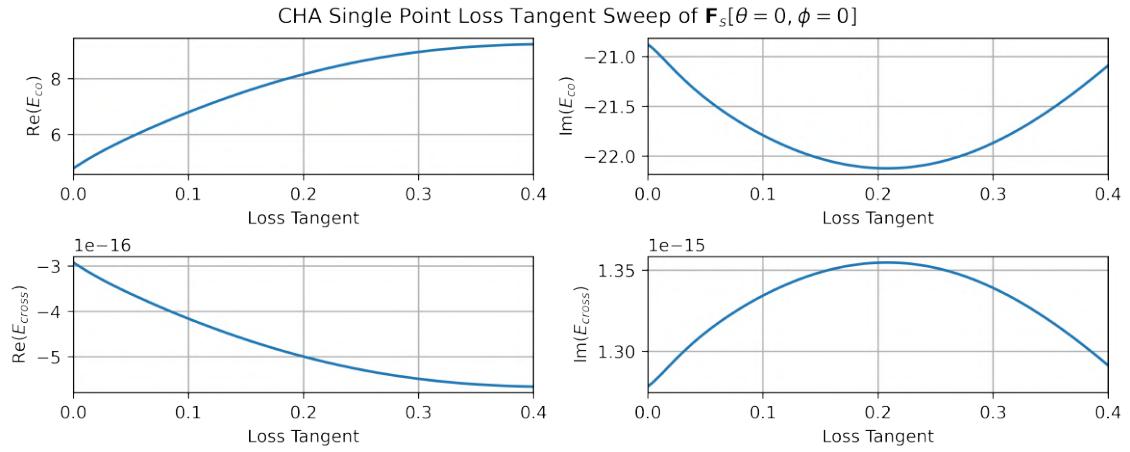


Figure D.3: Single point on the far-field sphere value as function of loss tangent in CHA dataset

D.3 RFLCT

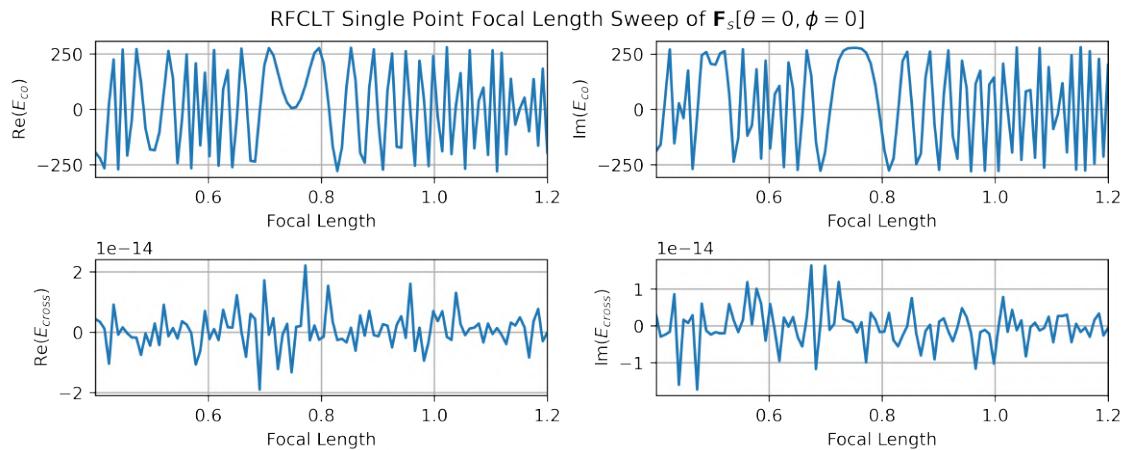


Figure D.4: Single point on the far-field sphere value as function of focal length in RFLCT dataset

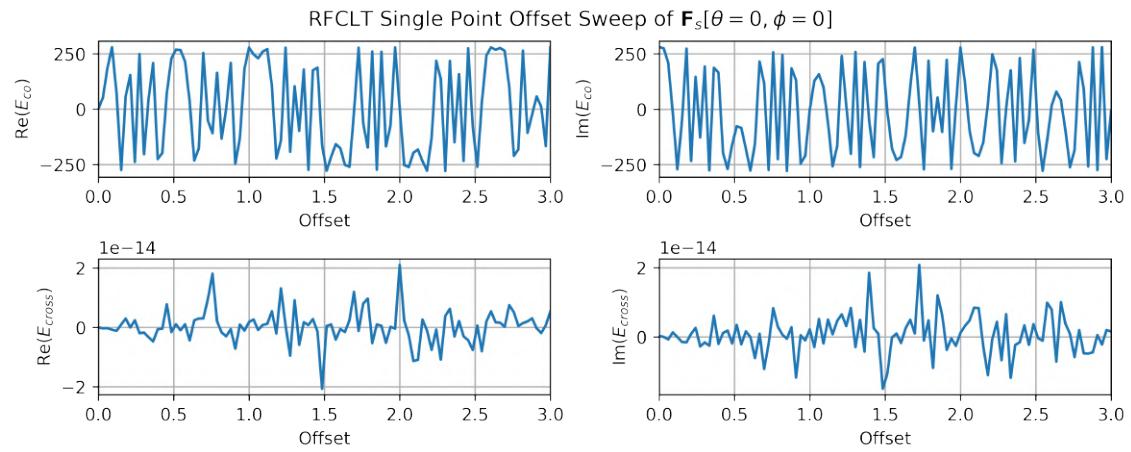


Figure D.5: Single point on the far-field sphere value as function of offset in RFLCT dataset

E Physics Informed Neural Networks in Far-Field Surrogates

A recent development within scientific machine learning is the formulation and application of physics-informed neural networks (PINNs) [66]. Physics-informed neural networks differentiate themselves from the usual physics-driven models outlined in section 3.1.1 by the fact that they are not relying on modification or tuning of an existing model; instead, they directly solve differential equations through the use of the auto differentiation and gradient descent.

Physics-informed neural networks (PINNs) are neural networks that use a physics-derived regularization term in their loss function to solve systems governed by differential equations. If then a set of boundary conditions are also provided as a target for the neural network, they can be able to converge upon the well-defined solution to the differential equation without any other data being provided.

Their implementation is usually done using auto-differentiation such as autograd from PyTorch[67]. These build-in graph-based auto differentiation schemes allow for easy calculation of derivatives between the input and output of a neural network. Thus a set of differential equations can easily be included as a loss factor for the networks.

It has even been proposed to substitute for the standard linear general PDE solvers in common use, since it has several advantages. Firstly, it does not require meshing, which is common in standard ODE solvers, such as finite difference and finite element analysis. Therefore it could be effective in problems limited by the meshing fineness and the curse of dimensionality. Secondly, meshing methods result in algebraic systems approximating solutions by piecewise polynomials which does not necessarily uphold the laws of physics implicitly. In contrast, PINNs are trained to be upholding the laws of physics from prior knowledge, such as energy conservation or symmetry throughout the region it is trained in.

Therefore one can consider PINNs as a data-driven surrogate model with a differential equation loss term. The data can be limited to only the boundary conditions needed to satisfy the uniqueness of the wanted solution. Still, multiple data points can be used to speed up the training and help the model achieve quicker convergence.

PINNs are also able to learn the constants in differential equations directly from data, as also showcased in [68]. By making a parameter in a differential equation part of the differentiating graph in the training of a neural network, the parameter can be discovered as part of the backpropagation step during network training.

Even though PINNs are in their infancy, a large amount of work has already been done to help produce them and give them a mathematical justification. [69] A python library built on TensorFlow specifically for PINNs is proposed. In [70] custom activation functions for use in PINNS are proposed, and claim to make PINN-convergence orders of magnitude higher than \tanh or \sinh activation functions. In [71] an attempt is made to formulate more rigorous mathematical foundations for PINNs.

A few electrostatic problems have been solved and published using PINNs[72, 73], though no published antenna surrogate model has used PINN-like physics-based regularization at the time of this writing.

The missing factor in using PINNs for far-field surrogate modeling is what physics-derived loss function to use. Of course, one could imagine gaining solutions directly from the Maxwell equations. However, in comparison with current numerical solvers, this likely be extremely slow and have to be very small in problem scope compared with what modern numerical solvers are capable of. Therefore for a current implementation that could help a far-field surrogate model converge faster or obtain more precise results, one could imagine using it for informed interpolation. Though going far away from a strongly data-driven solution does not seem feasible, as of yet.

F VAE Reconstruction Examples

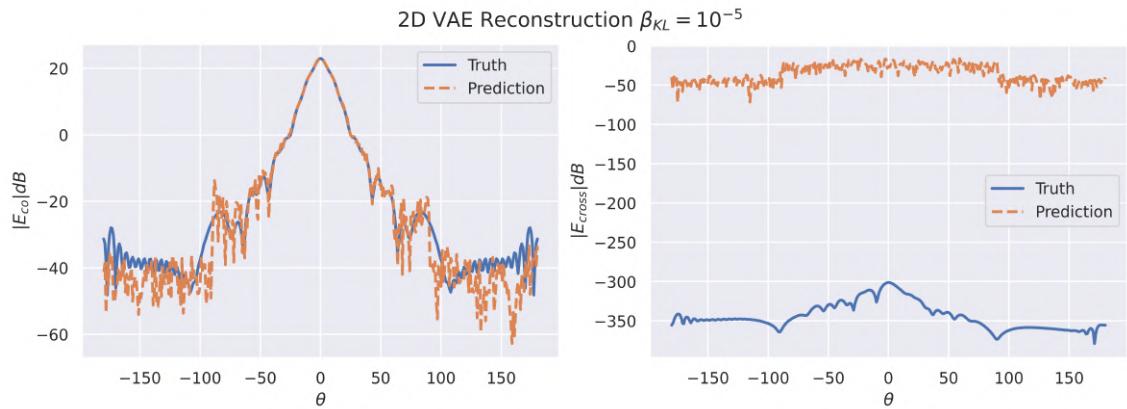


Figure F.1: 2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-5}$

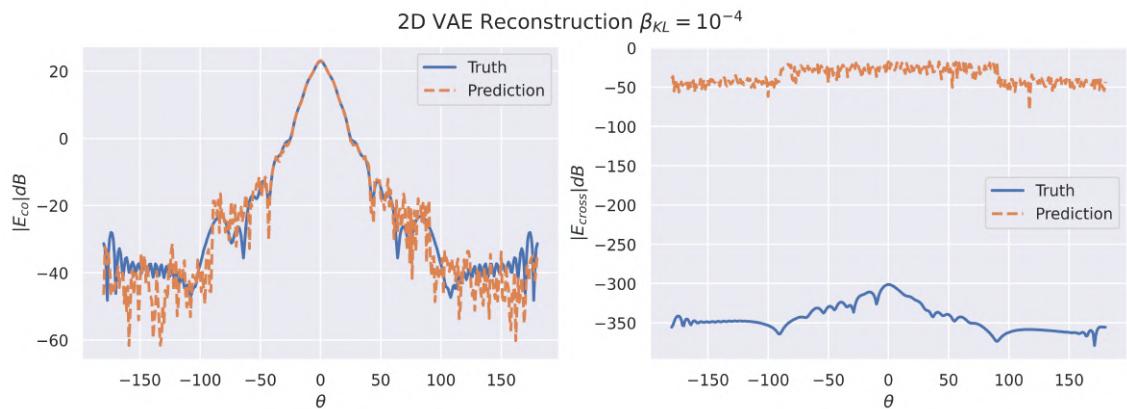


Figure F.2: 2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-4}$

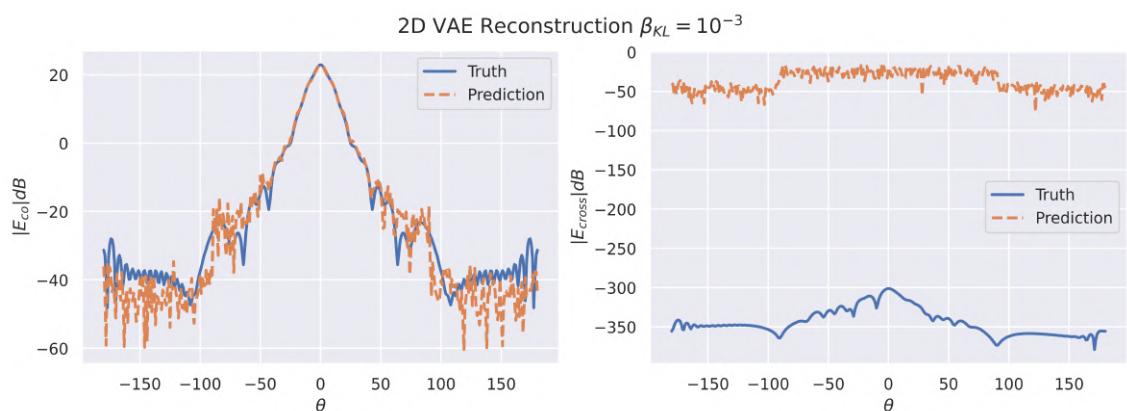


Figure F.3: 2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-3}$

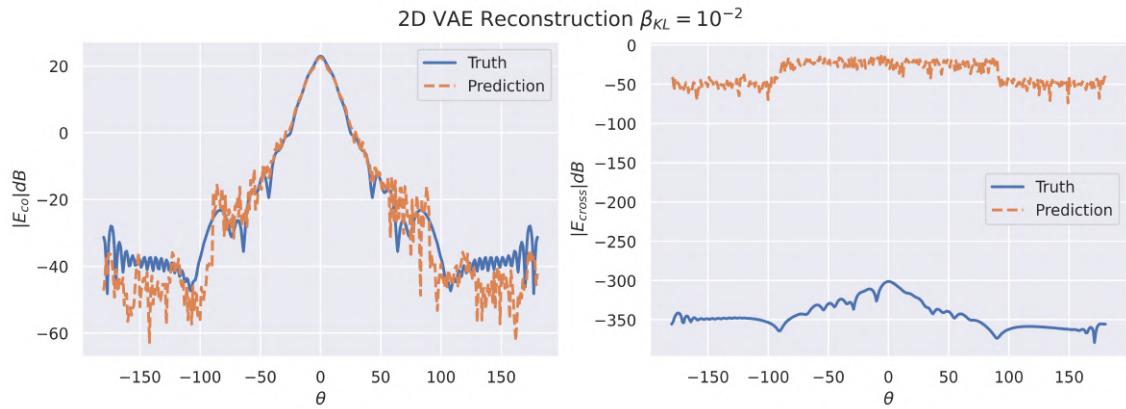


Figure F.4: 2D VAE reconstruction at $\beta_{KL} = \cdot 10^{-2}$

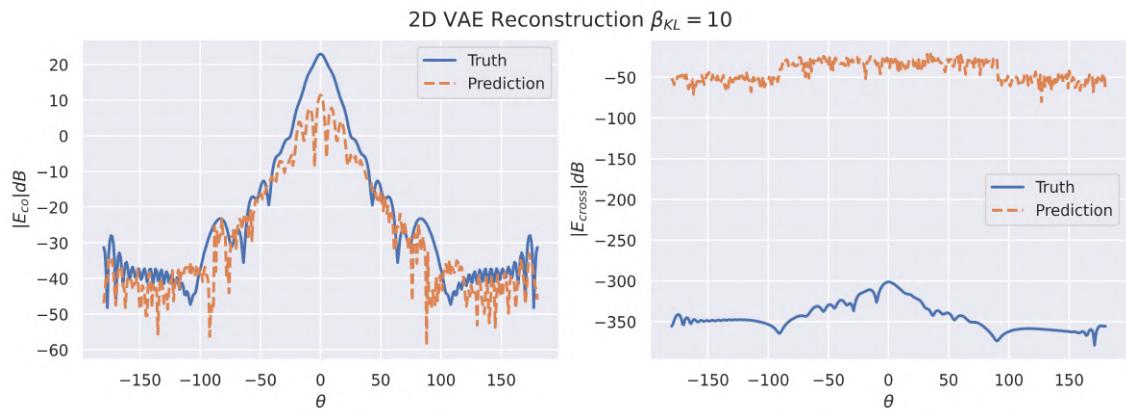


Figure F.5: 2D VAE reconstruction at $\beta_{KL} = 10$

Gradient descent can write code better than you. I'm sorry. - Andrej Karpathy

Technical
University of
Denmark

HC Ørsteds Plads, Building 348
2800 Kgs. Lyngby
Tlf. 4525 1700

www.elektro.dtu.dk