



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Railway infrastructure GPR dataset creation and modeling via black box and physics-informed neural networks

Master Thesis

Thomas Rigoni

April 2024

Advisors: Prof. Dr. Thomas Hoffmann, Prof. Dr. Eleni Chatzi,
Arcieri Giacomo, Dr. Marcus Haywood-Alexander

Department of Computer Science and Institute of Structural Engineering, ETH Zürich

Abstract

The assessment of track condition is important for the upkeep of modern rail infrastructure networks. Current methods for measuring and tracking the deterioration of the ballast layer are invasive, costly, and time-consuming, posing challenges which reduce the ability to adopt preventive maintenance strategies. Recent research investigates the utilization of Ground Penetrating Radar (GPR) as a cost-effective, fast, and non-invasive alternative for estimating ballast fouling and subsurface water content. Machine learning methods can be used to enable the automatic detection of these features, which however require high amounts of (labelled) training data. Despite the potential advantages of GPR, the generation of large-scale datasets using simulating software, such as gprMax, demands impractical amounts of computing power. In response to this challenge, we create two GPR datasets, which we use to train a CNN-based encoder-decoder architecture that serves as an accurate surrogate model for gprMax and investigate the application of physics-informed neural networks (PINNs) to this task. PINNs represent a plausible tool for their ability to exploit Maxwell's equations to approximate the underlying electromagnetic wave field propagation, but unfortunately our experiments show that a straightforward implementation faces significant challenges in this setting and we conclude that more research in this direction is needed. We report a reduction in computational cost of our model of ≈ 300000 times in a batched inference setting compared with the gprMax implementation, which results in a general speedup of the dataset generation process of ≈ 300 times. More efficient modelling will facilitate the creation of extensive datasets, paving the way for the subsequent monitoring methods aimed at supporting the maintenance of railway infrastructure.

Contents

Contents	ii
List of terms	1
1 Introduction	3
1.1 Problem description and motivation	3
1.2 Thesis objectives	4
1.3 Contributions	5
1.4 Thesis outline	5
2 Background	7
2.1 Railway track structure	7
2.1.1 Ballast fouling and water infiltration	9
2.2 GPR principles	9
2.3 gprMax	11
2.4 PINNs	11
3 Literature review	14
3.1 GPR	14
3.1.1 General GPR modelling	14
3.1.2 ML applied to GPR	15
3.2 PINNs	17
3.2.1 PINNs for wave propagation	17
3.2.2 PINNs for GPR data	18
4 Dataset generation	20
4.1 Initial model comparison	20
4.2 Implementation	27
4.2.1 Ballast compaction	30
4.2.2 Open Source contributions	32
4.3 Results	33

Contents

4.4	Limitations of the current design	33
5	ML surrogate models	37
5.1	Black box model	37
5.2	PINN models	42
5.2.1	MLP models	42
5.2.2	CNN model	49
6	Conclusions	54
6.1	Discussion	54
6.2	Future work	55
	Bibliography	57

Terms

B Magnetic flux density field.

D Electric displacement field.

E Electric field.

H Magnetic field strength.

*R*₀ Reflectance of an interface.

ε Electrical permittivity.

ε_r Relative electrical permittivity, dielectric constant.

μ Magnetic permeability.

σ Electrical conductivity.

σ^{}* Magnetic loss.

n Refractive index.

CNN Convolutional Neural Network.

DL Deep Learning.

EM Electromagnetic.

FDTD Finite Difference Time Domain method.

FWI Full Waveform Inversion.

GPR Ground Penetrating Radar.

ML Machine Learning.

MLP Multi-Layer Perceptron.

MSE Mean Squared Error.

NN Neural Network.

PCA Principal Component Analysis.

PDE Partial Differential Equation.

PINN Physics-Informed Neural Network.

PSS Planumsschutzschicht, compacted version of the gravel-sand subgrade.

RSA Random Sequential Adsorption algorithm.

SBB SBB CFF FFS - Swiss federal railways.

Chapter 1

Introduction

1.1 Problem description and motivation

According to the Swiss Federal Office of Transport [1], every day around 1.4 million people use the railway network in Switzerland, with Swiss citizens travelling more by train than any other nation in the world [2]. Consequently, the upkeep of the rail infrastructure network is of fundamental importance to the internal country's economy, even more so taking into account the around 45 million tonnes of freight cargo transported every year on Switzerland's railway networks [3]. Considering the real-world tear down cycles that the materials in railway tracks need to withstand, an important part of the network's upkeep resides in the timely inspection and maintenance of track conditions, which allows for longer periods of life and a more efficient and safe use.

Current methods for measuring and tracking the deterioration of the ballast and sub-ballast layers traditionally involve two procedures: a visual inspection of the top of the ballast layer and the excavation of possibly problematic areas [4]. The visual inspection is usually performed regularly, during which any inadequate sections are noted. This is a relatively slow and subjective process, where little information on the layers underneath the top is available, and thus locations with less than obvious deficiencies are often overlooked. On the other hand, more information is obtained by conducting an excavation within the track area, followed by a direct assessment of the quality of ballast and layers underneath. This latter procedure is invasive, very costly and time consuming and can possibly require a temporary closure of the track, which incurs in further drawbacks. All these characteristics pose additional challenges which hinder the ability of transportation companies to adopt preventive maintenance strategies, as no accurate and ubiquitous surveying of track conditions is possible with these methods.

Recent research investigates the utilization of Ground Penetrating Radar as a

cost-effective, fast, and non-invasive alternative for estimating ballast fouling and sub-surface water content [5]. When present, these two factors are, above others, the main safety risks for the usage of the railway infrastructure [6, 7]. The successful implementation of this technology would enable significantly faster evaluation of track conditions, which would result in a more comprehensive surveying of the railway network and a precise and timely maintenance of the tracks.

Machine learning methods can be used to enable the automatic detection of ballast fouling and groundwater from GPR-produced images of the railway tracks [8], which however require high amounts of labelled training data in order to generate accurate models. The necessary real-world data is not readily available, as the creation of labels would require extensive excavation of railway tracks. To circumvent this problem, it is possible to use computer simulations to create large amounts of GPR data comprising of both accurate geometry maps and labels. Unfortunately, the generation of large-scale datasets using simulating software, such as *gprMax*[9, 10], still demands impractical amounts of computing power, especially when considering realistic 3D geometries. To this end, this work explores the creation of deep-learning-based surrogate models for *gprMax*, which can be used to create large-scale labelled GPR datasets of railway track configurations in the future.

In particular, two approaches are followed for the modelling of GPR data. First, a black box CNN-based encoder-decoder network is trained following an end-to-end supervised ML approach. Second, the application of physics-informed neural networks is evaluated. PINNs represent a plausible tool on this task for their ability to include constraints expressed in the form of Partial Differential Equations (PDE) during their training. These are represented in this case by Maxwell's equations, which can be used to approximate the underlying electromagnetic wave field propagation. To the best of our knowledge no published work explores the application of PINNs on GPR data with a level of complexity comparable to railway track infrastructure.

1.2 Thesis objectives

The main objectives of this thesis are:

1. Creation of a (simulated) realistic and diverse dataset of railway track conditions, comprising both a precise geometry of the samples and their GPR measurements.
2. Training of machine learning models to approximate the response of the GPR measurements, such that they can act as surrogate models of current GPR simulation software. To this end, investigate the possibilities for application of physics-informed neural networks[11].

In particular, at the conclusion of this work, it should be possible to use the trained models to obtain GPR measurement data in a fraction of the time required by the simulation software. This would enable the generation of the large-scale datasets required for the creation of new ML-based monitoring methods for the maintenance of the railway infrastructure.

1.3 Contributions

The main contributions of this work can be summarized as follows:

- Creation of python scripts that enable the automatic generation of railway track GPR datasets, via the software library *gprMax*. Two different datasets are produced with these programs, each with a specific goal.
- Adaptation and training of a black box CNN model for the approximation of GPR responses, which can act as a surrogate model for *gprMax* with high accuracy. To the best of our knowledge, no published work applies ML-based solvers to similarly complex GPR geometries obtaining comparable levels of precision.
- Exploration of the application of physics-informed neural networks to the aforementioned approximation problem. We investigate two different architectures and show their strengths and limitations, discussing potential causes of their drawbacks and plausible future research directions.

1.4 Thesis outline

Chapter 2 discusses some of the background information and technologies on which this work is based, including railway track structures (Section 2.1), GPR operation principles and *gprMax* (Sections 2.2 and 2.3), and physics-informed neural networks in Section 2.4.

Chapter 3 discusses some of the most relevant literature in this field, both on GPR modeling (Section 3.1) and PINNs (Section 3.2).

Chapter 4 presents the dataset generation process, including an initial modelling comparison in Section 4.1, implementation choices (Section 4.2), discussion of the results, and some of the design limitations (Sections 4.3 and 4.4).

Chapter 5 shows the ML-based surrogate models developed for *gprMax*, with a focus on both their architectures, training methodologies, and results. First, a black box CNN model is presented in Section 5.1, then two physics-informed models are investigated in Sections 5.2.1 and 5.2.2 based on MLP and CNN PINN solvers.

1.4. Thesis outline

Chapter 6 summarizes the conducted work, underlines the strengths and weaknesses of the analyzed ML models, and discusses possible future applications and research directions.

Chapter 2

Background

2.1 Railway track structure

The structure of railway tracks has undergone constant technical upgrades and revision in the last century, but tended toward standardization, and it has nowadays few typical configurations [12]. Two main types of railway tracks exist today: slab tracks and ballasted tracks. Slab tracks are usually found in special areas, such as tunnels as well as on viaducts; this kind of track usually requires less maintenance and has a longer lifespan, but has the downside of a significantly higher cost for construction and smaller flexibility given by the rigid superstructure [13]. For these and other reasons, the predominant form of railway tracks in the world today consists of ballasted tracks [14].

The specifics of construction for ballasted tracks can vary, but in principle the steel rails are fixed on perpendicular sleepers (which can be made of wood, concrete or steel) and these sleepers are embedded into the compacted ballast layer. This layer is formed by crushed stones within roughly 3 - 6.5 cm in diameter, mainly composed of basalt, greywacke and granite [16]. The ballast layer is usually between 20 and 35cm deep, where to high speed tracks generally correspond higher ballast depth. The bottom of the ballast layer is always set at an angle of at least 5° as seen from a plane perpendicular to the rails, to facilitate water drainage.

Beneath the ballast, a subgrade layer of gravel and sand is found. This material has the function of securing the structural stability of the track, and the ability to drain water. Based on its composition and level of compactness, we can differentiate two similar subgrades: the regular subgrade is composed

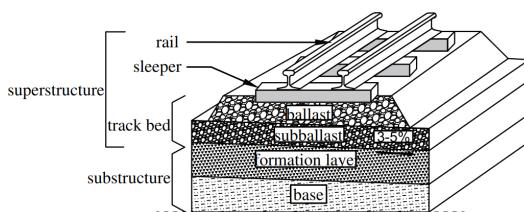


Figure 2.1: Structure of the railway track, from [15].

2.1. Railway track structure

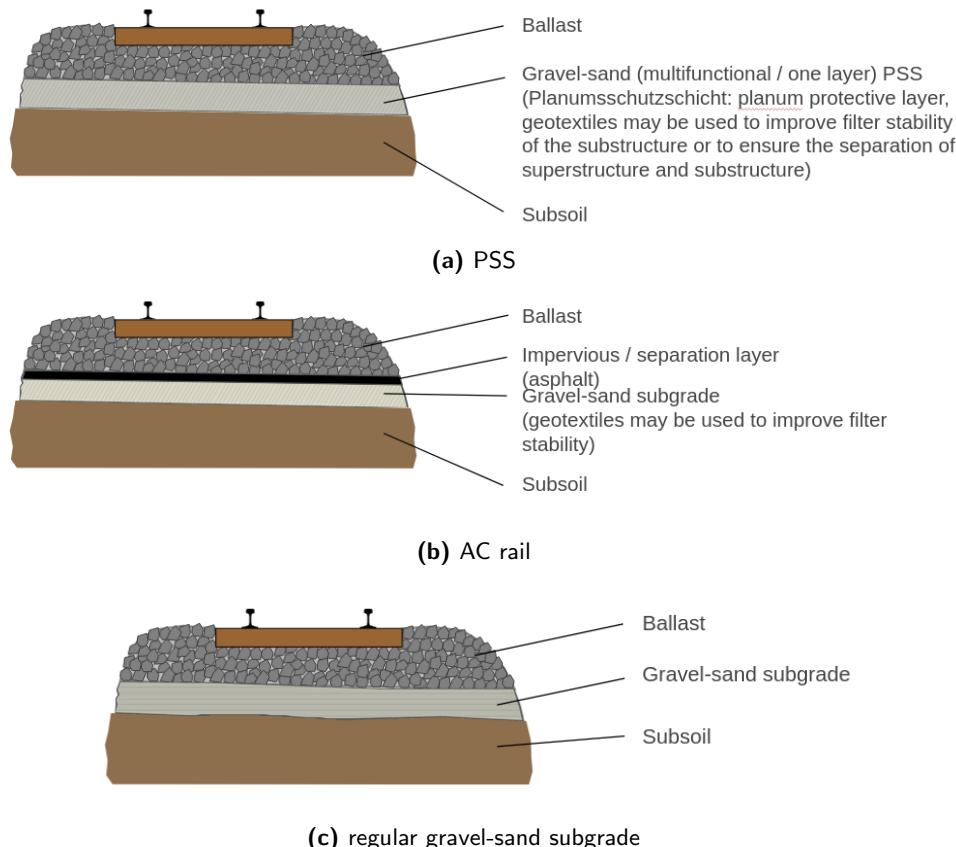


Figure 2.2: Railway track configurations, credit: David Haener from SBB.

of 75-80% gravel and 20-25% sand, with a bulk density of around 2100 kg m^{-3} . The PSS (from the german Planumsschutzschicht) is a more compacted version of this subgrade, with roughly the same amount of gravel and sand and a bulk density of around 2300 kg m^{-3} (source: David Haener from SBB). Between the ballast and the gravel-sand subgrade, a relatively thin layer of asphalt may be present, between 5 and 8cm. This latter configuration is termed AC rail, which forms a very durable railway track, but requires a higher construction cost with respect to regular subgrade tracks. Invariably, the lowest levels, termed subsoil, are composed of compacted soil layers or pre-existing structures, e.g. rock formations. Figure 2.2 shows the three aforementioned configurations.

The superstructure is formed by ballast, sleepers and rails. The substructure is composed of the layers placed between the superstructure and the subsoil, i.e., the natural ground.

2.1.1 Ballast fouling and water infiltration

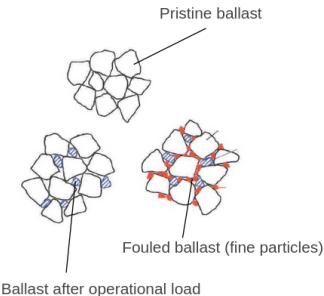


Figure 2.3: Depiction of ballast fouling, from [17].

The two main events that determine the deterioration of the ballast and the subgrade layers are ballast fouling and the presence of groundwater [6]. Ballast fouling is defined as the presence of fines between the ballast stones. Fouling can have multiple causes: the leading ones are the splintering and crumbling of the ballast stones themselves following numerous cycles of operational load [18, 7], and the infiltration of fine material from both above and below the ballast layer, respectively named descending and ascending fines. Infiltration from above can include the contamination of the ballast due to

vegetation, loss of cargo materials like coal and metal powder, and environmental impacts such as freezing and weathering. Ascending fines generally involve an insufficient stability of the substructure and underground material, or are caused directly by the presence of groundwater. The main effect of ballast fouling is a reduced elasticity of the railway track, which can cause damage to the sleepers and rails and pose safety concerns [7].

Water accumulation, for example due to a blocked drainage, can be the cause of a faster deterioration of the railway track. Standing water combined with loading and unloading cycles can cause the formation of mud and the pumping of fines into the ballast layer, causing fouling and the presence of voids both in the substructure and underneath the sleepers [19]. If the standing water persists for a prolonged period of time, the substructure may be damaged up to a total failure of the track.

2.2 GPR principles

Ground Penetrating Radar is a technology that allows non-destructive surveys of the sub-surface to investigate underground characteristics of the area, including the presence of defects, pipes, rods, etc. [20] This class of radar is usually composed of two antennas: the transmitter sends pulses of electromagnetic waves in the microwave band towards the soil (usually between tens of MHz to a few GHz). When an EM wave changes the medium in which it is propagating, a part of it gets refracted, while the rest is reflected based on the Fresnel coefficients (see Eq 2.7), derived from the Maxwell equations (see Eq 2.1 to 2.4). The reflected waves from the sub-soil are detected by the receiving antenna for a certain amount of time, creating a time-series of the electric field amplitude in time, which is called an A-scan. Usually, a GPR investigation involves multiple A-scans, taken at regular distances and then

2.2. GPR principles

stacked onto each other, obtaining a two-dimensional image called a B-scan, or radargram. In some cases, it is possible or necessary to scan multiple sections of the ground in parallel patterns with constant distance between them, which results in so-called C-scans.

The Maxwell equations in their common form are as follows [21]:

$$\nabla E = -\frac{\partial B}{\partial t} \quad (2.1)$$

$$\nabla H = \frac{\partial D}{\partial t} + J_c + J_s \quad (2.2)$$

$$\nabla B = 0 \quad (2.3)$$

$$\nabla D = q_v \quad (2.4)$$

where: E is the electric field, B the magnetic flux density field, D the electric displacement field, H the magnetic field strength, t time, J_i the current density, q_v the volume electric charge density.

When interacting with nonmagnetic materials, the equations can be simplified in [5]:

$$\nabla E = -\mu \frac{\partial H}{\partial t} \quad (2.5)$$

$$\nabla H = -\epsilon \frac{\partial E}{\partial t} + \sigma E \quad (2.6)$$

where ϵ , μ and σ are properties of the materials: respectively the electrical permittivity, the magnetic permeability and the electrical conductivity.

The derived Frensel coefficient in case of normal incidence is expressed as [22]:

$$R_0 = \left| \frac{n_1 - n_2}{n_1 + n_2} \right|^2 \quad (2.7)$$

where R_0 indicates the reflectance of the interface, i.e. the fraction of incident power reflected, and n_1 and n_2 indicate the refractive index of the two materials, defined as the ratio of the speed of light in a vacuum (c) to the speed of light in the medium, and can be expressed as $n = c \cdot \sqrt{\epsilon \mu}$. Note in particular that greater differences in ϵ_r and μ_r of the materials correspond stronger reflections, as $\epsilon = \epsilon_0 \cdot \epsilon_r$, where ϵ_0 is the permittivity of free space. Likewise for $\mu = \mu_0 \cdot \mu_r$.

2.3 gprMax

gprMax [9] [10] is an open-source software library written in Python and Cython that simulates the propagation of electromagnetic waves in both dielectric and conductive mediums. This is achieved using the Finite Difference Time Domain Method (FDTD) [23] applied to solve the Maxwell equations. Essentially, *gprMax* discretizes the domain space by creating a grid with sufficiently small cells, and advances the state of the simulation in very small time steps (in the order of picoseconds). These are the main cause of the high computational cost for FDTD simulations, but are necessary to guarantee the stability of the process, which must follow the CFL condition [23]:

$$\Delta t \leq \frac{1}{c \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}}$$

Where c is the speed of light in a vacuum. The stability condition in two dimensions is obtained by letting $\Delta z \rightarrow \infty$.

gprMax is a very flexible tool, it allows the creation of different kinds of materials given their EM properties, including multiple Debye, Drude and Lorentz dispersive materials and mixing models based on Peplinski soils [24, 25]. While *gprMax* offers a wide variety of geometry construction commands, arbitrary geometries can be constructed from defined materials by loading them from files in the standard HDF5 format. For instance, it is possible to create fractal boxes with rough surfaces on any face, with the addition of water and/or realistic grass on it.

GprMax has been widely used in many applications related to landmine detection [26], construction surveys for detection of defects and cracks in buildings and infrastructures [27], realistic GPR dataset creation [28], spectral soil property studies, even for conducting geological simulations for the RIMFAX radar equipped on the *Perseverance* rover on the mission *Mars 2020* [29].

2.4 PINNs

Physics-informed neural networks [11] are a deep learning framework that allows to embed partial differential equation constraints in the training process of the ML models. This is accomplished by introducing additional terms that contribute to the loss function adopted to train the networks. In particular, PINNs usually have no (or very minimal) architectural differences compared to regular NNs. This framework is mainly employed in regression tasks, where the value of some physical quantity is to be determined [30].

Usually, the loss is divided into three terms [31]:

- **Observations loss:** this term is the standard loss used in regular supervised learning, where the predictions generated from the observations of a training dataset are fit to their corresponding labels.
- **Collocations loss:** this term is directly derived from the PDE that describes the physical law under exam. Numerous points are sampled inside the so-called collocation domain and the PDE residual is evaluated for each of them.
- **Boundary loss:** this term, although not necessary, is derived from boundary conditions that can be prescribed to the model. It is usually very helpful to specify additional boundary conditions, as they restrict the solution space and allow for a faster and more precise convergence to the problem solution.

These loss terms can then be combined through a weighted sum, where coefficients λ are used to stabilize the training process:

$$L = \lambda_{obs} L_{obs} + \lambda_p L_p + \lambda_{bc} L_{bc} \quad (2.8)$$

where each L_i is usually obtained by applying a MSE on the loss residuals.

A popular choice for PINNs is the adoption of a simple architecture, namely the Multi Layer Perceptrons (MLPs), exploiting their capability as universal function approximators to model the underlying solution. Usually, both the inputs and outputs of the networks are physical quantities.

Physics-Informed Neural Networks are usually employed for three kinds of tasks [31]:

- **Domain extension:** given observations in a spatio-temporal domain, extend the domain in which the network can predict accurate results by training on collocation and boundary points in a wider range. One widely seen application of domain extension consists of employing observations only for the initial state of a system (one or multiple close time snapshots) and using a PINN to infer the state of the system at later times. This is for example the case for some of the experiments showcased in the original PINN paper [11].
- **Sparse reconstruction:** given observations in sparse spatial or time steps, use PINNS to accurately reconstruct the solution in unobserved parts of the domain, between observations.
- **Parameter estimation:** given observations, use a PINN to infer the physical parameters related to the studied problem. This task is also frequently referred to as the inverse problem.

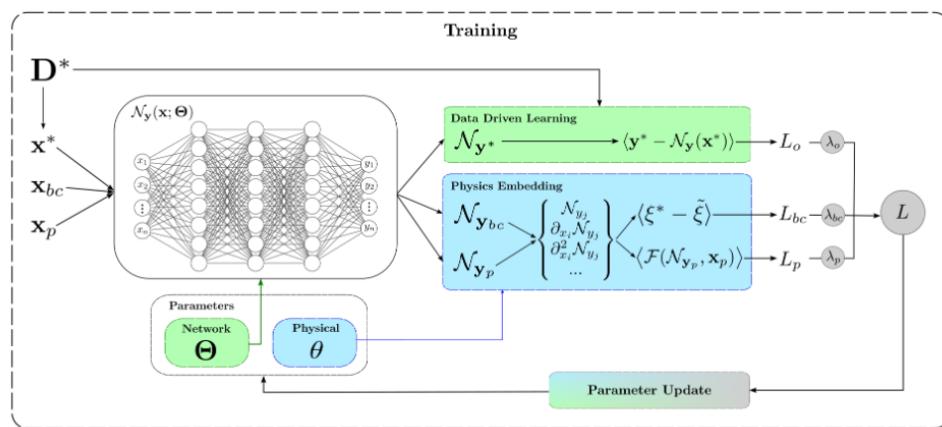


Figure 2.4: Physics-Informed Neural Network training procedure, from [31].

Chapter 3

Literature review

This chapter will discuss some of the most relevant works in the literature related to this thesis.

3.1 GPR

This section will review some publications regarding Ground Penetrating Radar. Section 3.1.1 will discuss some methods used for GPR data simulation, while section 3.1.2 will discuss the application of Machine Learning for GPR data simulation and inversion.

3.1.1 General GPR modelling

The modelling of GPR data is a fairly common practice and many papers explore this technique. Many implementations have emerged, mainly in Python, C++ and MATLAB. Irving et al. [32] produced a codebase in MATLAB for the numerical modelling of GPR wavefields in 2D using the Finite Difference Time Domain (FDTD) method. Islam et al. [33] use the Advanced Design System (ADS) software for calculating the distance, material and Doppler shift of targets buried in the ground. Zarei et al. [34] propose a MATLAB implementation for spectral element modelling of GPR waves for 2D geometries, using high-degree Lagrange interpolation to discretize the EM wavefield, and show that their method is both more efficient and more precise than Finite Element Methods. Deng et al. [35] implemented a C++/CUDA 3D forward and backward solver for EM wavefield propagation and Full Waveform Inversion (FWI) in Debye dispersive mediums. Their implementation uses FDTD for the forward solver and has multi-GPU capability, which allows to divide the domain into overlapping parts that are solved by each GPU in parallel.

gprMax [9, 10] is consolidating itself as the main software library to conduct GPR simulations for its flexibility and efficiency, especially since the release of a CUDA implementation. This feature allows to run the main FDTD cycles on NVIDIA GPUs, making the simulation process significantly faster. A high number of publications cite the use of *gprMax*, part of which presented by the same authors of the software. Koyan et al. [28] use *gprMax* to create realistic 3D sedimentary models and related GPR data. Giannakis, Warren et al. [36] [37] (the original authors of *gprMax*) use it for landmine detection. More examples are found in section 2.3.

3.1.2 ML applied to GPR

In the last 20 years, many Machine Learning (ML) algorithms have been applied for the analysis of GPR data. The problems that received the most attention focus on the interpretation of GPR radargrams, in the form of A,B or C-scans. The tasks to which the models are applied are mostly the automatic detection of imperfections or underground objects or rods, and the estimation of ground and objects physical properties.

The review article by Travassos X.L. et al. [38] underlines three main kinds of applications for ML models within the domain of GPR data: preprocessing, image segmentation and pattern recognition. Preprocessing ML techniques are applied to reduce noise, echo and other losses due to the variability of the real world, for example Patsia et al. [39] use a neural network for background noise removal in GPR B-scans. Image segmentation is the process of labelling the pixels of an image based on a class to which they belong. This application is more and more dominated by the use of Convolutional Neural Networks (CNNs). Pattern recognition is the process of detecting and categorizing patterns present in the input image. This can be done with a number of models, from k-nearest neighbours (k-NN), to Support Vector Machines (SVM), Decision Trees (DTs) and Random Forests (RF), Gradient Boosting (GB) and Hidden Markov Models (HMM). Here, as in image segmentation, CNNs are emerging as the most accurate models for the task, thanks to their ability to detect local patterns with very high precision. In fact, pattern recognition can be seen as a necessary step in image segmentation.

The survey article by Tong et al. [40] characterizes the methods of ML applied to GPR based on their use of A, B or C-scan data. For A-scan data, signal processing techniques are used, and high-level representations are often extracted from the A-scan data, by some tailored transformation or by Principal Component Analysis (PCA). B-scan based methods are divided into patch-based and region-based. The former use fixed-sized patches as input to ML models in a classification task. The latter methods input the full image to the model and get in return a region of interest, together with a class. Tailored CNN architectures have emerged in recent years to perform this

latter task, one example of such model is the Faster Region Convolutional Neural Network (Faster R-CNN).

Giannakis et al. [37] conducted an initial investigation for an algorithm for landmine detection using neural networks and GPR data, using the PCA of A-scan traces as input of the network and obtaining promising results. Picetti et al. [41] propose a convolutional autoencoder for anomaly detection, applied to landmines. Their work only uses landmine-free training samples, making the system robust to a wide variety of targets, as no strong assumptions are *a priori* made, but also leads the system to be more prone to false positives if another kind of anomaly is detected.

A smaller number of works focus on the application of machine learning to the full inversion of the GPR data, meaning the process of using GPR measurements to obtain an approximated permittivity map (or velocity map, since the two quantities are related) of the sub-surface. Leong et al. [42] develop a model based on DeepLabV3 called GPRNet for computing the wave speed starting from A-scan GPR data. Ji et al. [43] use a custom-built CNN composed by a convolutional time-dimensional data compressor, a MLP for feature encoding and convolutional permittivity decoder to obtain permittivity maps of the sub-surface from B-scans. Xie et al. [44] extend the Unet model for the same task and show superior results compared to a Fully Convolutional Network (FCN). Alvarex et al. [45] use an encoder-decoder network, Unet and a conditional Generative Adversarial Network (cGAN) for the same task and compare the results of the different networks with L1, L2 and the structured Similarity Index as loss functions to compare their results.

More inherent to this work, some publications explore the use of ML techniques for forward-simulation of GPR data. In particular, Giannakis et al. [46] use an MLP for predicting the mean A-scan trace of a realistic soil model (Peplinski soil [24]) given some parameters as input describing the sand fraction, volumetric fraction, water fraction, soil roughness and height of the antenna. In [47] the same authors use a combination of MLPs and PCA to estimate the A-scan response from a metallic rebar embedded in a concrete foundation. Dai et al. [48] use an encoder-decoder architecture exploiting attention-based feature fusion to calculate the B-scan response of a geometry, given its permittivity and permeability maps. Although the developed framework performs well on the test set, the geometries are still very simple, with only single extraneous objects embedded in near-uniform materials.

3.2 PINNs

From their introduction in 2017 by Raissi et al. [11], Physics-Informed Neural Networks have been used for modelling a variety of problems involving the minimization of PDEs, mainly directly physics related. Initially they were employed to solve Schrödinger, Burgers and Allen–Cahn equations[30]. In the next years, they were expanded to solve Euler equations on high-speed aerodynamic flows, the incompressible Navier-Stokes equations for fluid dynamics, then coupled with temperature equations for analyzing heat flow convection. The review article by Cuomo et al. [30] explains how the main architectures used with PINNs are Feed Forward NNs (FFNN, both shallow and deep), convolutional neural networks, and recurrent neural networks. Recently other architectures have been explored, such as AutoEncoders (both fully connected and convolutional) and Generative Adversarial Networks (GANs). Some real-world applications of PINNs include fluid dynamics, both for high and low speed scenarios, hemodynamics, wave dynamics both for elastic (geoscience and elastostatic problems) and electromagnetic waves, optics, molecular dynamics, and material related applications. Some other industry applications are for example related to lubricant deterioration, corrosion, machine damage estimation, heat transmission and more. Various frameworks have been developed in recent years to develop Physics-Informed Neural Networks: to name a few DeepXDE [49], NeuroDiffEq [50], based on PyTorch [51], NVIDIA Modulus [52] (previously known as NVIDIA SimNet), which aims to be a toolbox for solving both research and real-world related problems related to PINNs.

3.2.1 PINNs for wave propagation

In the context of wave propagation, the majority of published works explores the utilization of Physics-Informed Neural Networks applied to the scalar or acoustic wave equations. Already in 2020, Moseley et al. [53] solved the acoustic wave equation with a 10-layer PINN, for both homogeneous, layered and more complex velocity models, including conditioning the networks on the wave source location, by adding it as an input. They show that the networks they trained are able to extend the simulations effectively in the time domain, although for the more complex case the reflected wavefronts are not very well modeled. Voytan et al. [54] solve the scalar wave equation on a rectangular membrane with Dirichlet boundary conditions. They propose to use paraxial absorbing boundary conditions to restrict reflections from the computational boundaries and use an L4 loss at the beginning of the training process to avoid the trivial solution of predicting 0 everywhere. Song et al. [55] solve the acoustic wave equation in the frequency domain, proposing an adaptive sinusoidal activation function. Although their results seem to be promising, the terrain models are relatively simple. Alkhadhr et al. [56]

model ultrasound wave equations, with a focus on possible future medical applications. Their results match the underlying solution quite well, but also appear to be conducted in very simple domains. Rasht-Behesht et al. [57] conduct a series of experiments on PINNs applied to acoustic wave propagation for seismic data. Their approach shows very good results on both forward simulation and Full Waveform Inversion (FWI), introducing free-surface boundary conditions at the top of the simulation boundary to approximate the surface of the planet, which results in precise reflected waves. Zhang et al. [58] use a PINN to approximate the velocity and density fields based on acoustic wave equations to perform seismic inversion. Their approach is able to accurately predict seismographic data and also calculate accurate velocity and density models, albeit still quite simple.

3.2.2 PINNs for GPR data

The application of PINNs to electromagnetic (EM) waves, and in particular to GPR data, received less attention. Zheng and Wang [59] use a Physics-Informed Neural Network to perform time domain extension on GPR data. They show that this approach has the potential to overcome some limitations of Finite Difference Time Domain solvers, in particular numerical dispersion. They use an MLP with x , y , and time coordinates as input to predict the electric field at a specific point in space and time as output. They exploit the second-order scalar wave equation for the electric field to avoid the alternate computation of electric and magnetic fields, as it is done in FDTD. As it appears in their paper, the PDE is:

$$\frac{\partial^2 E}{\partial t^2} - \frac{1}{\mu\epsilon} \nabla^2 E + \sigma\mu \frac{\partial E}{\partial t} = 0 \quad (3.1)$$

From an analysis of the units of measure, it is possible to see that the first and second summands have units [Vm/s^2], while the third summand has units [V/m]. This suggests that an error is present in equation 3.1. Equations with correct units of measure are present in a publication by Zarei et al. [34] (cited by Zheng and Wang):

$$\frac{\partial^2 E}{\partial t^2} - \frac{1}{\mu\epsilon} \nabla^2 E + \frac{\sigma}{\epsilon} \frac{\partial E}{\partial t} = 0 \quad (3.2)$$

This (Equation 3.2) is the PDE that has been used for the experiments conducted in this work and, as can be seen in Section 5.2, it leads to promising results in multiple tests, vouching for its correctness.

In the aforementioned paper by Zheng and Wang, the authors suggest the possibility to use some boundary conditions like the paraxial adsorbing

3.2. PINNs

boundary condition [54], open boundary conditions [52], or free-surface boundary conditions [57], which were employed in previous acoustic wave simulation solvers, although it is not clear if any of these were actually used, or, if so, how they were adapted to the EM case.

The studied geometry models are still relatively simple and unfortunately no experiment showcases a strong reflection of the EM waves achieved only through domain extension, but observation data is used in the training that already showcases the reflection (see Figure 3.1). For the most complex geometry model, shown in Figure 3.2, the predictions are still not optimal for GPR data simulation, since, as remarked by the authors, most of the high-frequency components of the EM waves are not captured by the model. Moreover, the authors mention that the training time for their models is in the order of tens of hours, while a FDTD solver only needs some seconds to calculate the solution field.

All the aforementioned publications use PINNs as instance solvers, namely the physical parameters of the simulation (geometry maps) are fixed and, if the parameters change, it is necessary to train a new model. To the best of our knowledge, no published work explores the application of PINNs to more complex geometries or tries to create a model capable of predicting accurate results on multiple (unseen) geometries in the context of wave propagation.

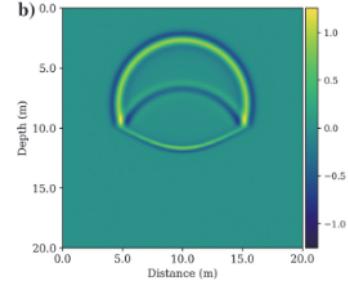


Figure 3.1: Snapshot of the E field used in training of a 2-layer model in [59].

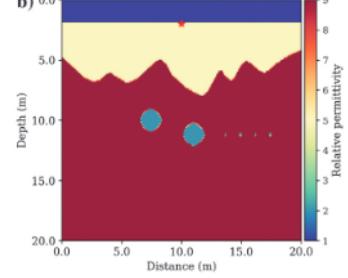


Figure 3.2: Most complex geometry model used in [59].

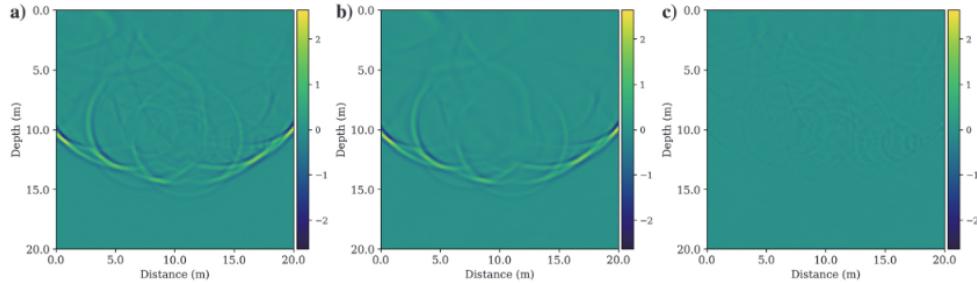


Figure 3.3: Electric wavefield predictions for the geometry in figure 3.2 from [59]. FDTD simulation on the left, PINN predictions in the center, their difference on the right.

Chapter 4

Dataset generation

This chapter expands on the first objective delineated in Section 1.2: we aim to create a realistic simulated GPR dataset of diverse railway track conditions, comprising of both full geometry maps and GPR measurement labels. This dataset will be used for the creation of ML-based surrogate models for gprMax.

First, we delineate some properties that we aim to fulfil for the data that we are going to generate:

- **Realistic:** The generated dataset should be as close to a real world scenario as possible, given the constraints below.
- **Varied and representative:** The data should include a wide range of conditions and situations encountered in the real world, with similar distribution probabilities.
- **Computationally feasible:** It should be possible to generate a dataset with a number of samples in the order of thousands in a reasonable time, on a sufficiently powerful machine.

These three principles will guide us in the modelling choices throughout all of this chapter, and will be discussed in more details at later points.

4.1 Initial model comparison

In order to evaluate the feasibility of different possible simulated models and analyze their main differences, the first step towards dataset generation is a model comparison. This step provides us with important information about the computational cost of each model and allows us to quantify the differences in the output B-scans. In total, six different simulated models are explored: three in 2D and three in 3D. All of them are composed of sleepers on top of 4 layers: ballast, PSS, asphalt and gravel. This structure is the one

found at the track built by SBB at ETH Hönggerberg and was adapted from the models in the Master Thesis by Lukas Heller [60]. The models are shown in Figure 4.1 and are:

- **2D box:** the simplest model is composed of 2D boxes representing all layers, including the ballast and PSS. The layers have a slightly rough surface to make them more realistic.
- **2D circles:** the ballast is modelled as 2D circles, while the PSS uses a mixture model for soil by Peplinski [24, 25].
- **3D cylinders:** simple extension of the 2D circles model in three dimensions, in which the circles become cylinders sharing the same projected positions.
- **3D cylinders with rails:** addition of rails on top of the sleepers of the 3D cylinders model, composed of an electrically conductive material.
- **3D spheres:** the most realistic model, where the ballast takes the shape of spheres of different diameters.
- **2D cut from 3D spheres:** a 2D vertical cut from the center of the 3D spheres model.

The main questions that we seek to answer with this choice of models are:

1. Is there a significant difference between the GPR radargrams generated by these different models?
2. If so, is it possible to decompose these discrepancies into factors, to understand the specific physical causes of the observed differences?
3. What is the computational cost of generating and simulating such models? Is this cost acceptable?

In particular, the answers to the second and third questions above will be decisive in the evaluation of these different models, to determine the best fit to the three properties outlined at the beginning of this chapter.

Results

All the geometries presented above were modelled and simulated through the open source software *gprMax*[9][10], using a 1 GHz Ricker (Mexican hat) pulse (see Figure 4.2) from a hertzian dipole initially placed 20 cm away from both the top and left boundaries of the simulation (depth corresponding to the center of the domain in the 3D simulations). The receiver is placed 15 cm on the right of the source and 55 A-scans are simulated with a step of 2 centimeters between them. All the A and B-scans presented in the following are resampled to the same time step (since, as briefly mentioned in Section 2.3, the CFL condition dictates different values of Δt for 2D and 3D models)

4.1. Initial model comparison

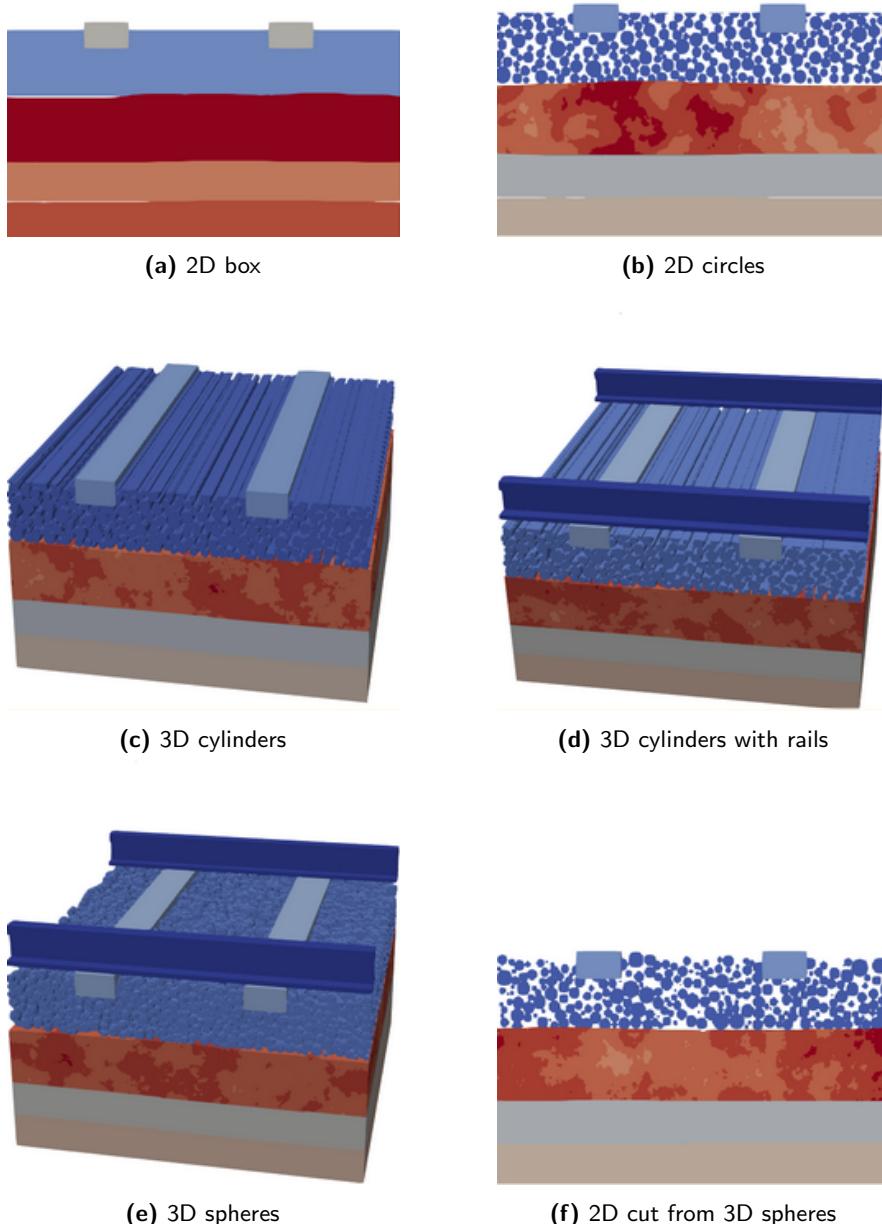


Figure 4.1: Simulated models considered in the initial comparison.

4.1. Initial model comparison

and adjusted to have the initial reflection at the same point in time, as 2D and 3D models differ slightly in this regard. Both the A and B-scans were pre-processed in their initial time steps by setting the response of the first 5ns to zero. This modification was made in order to remove the measurement of the direct wave between the source and the receiver, of small interest in this study.

As can be seen in Figure 4.3 2D box and 2D circles models significantly differ between time steps 900 and 1400 (corresponding roughly to 6.5ns and 10.3ns), where multiple stronger reflections are noticeable in the 2D circles model. This is likely due to the differences in the topology of the ballast, and in minor part also of the PSS. This result already answers the first question posed above: the difference in A-scan response for different models is indeed significant.

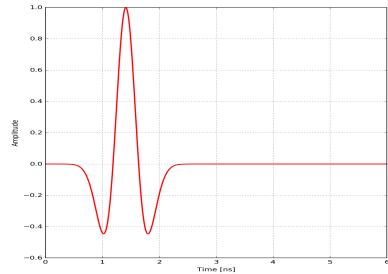


Figure 4.2: Ricker pulse, from the gprMax documentation.

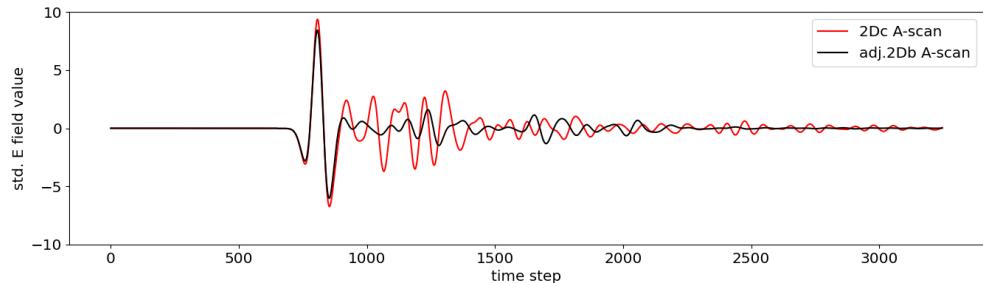


Figure 4.3: Comparison between central A-scans of the 2D box (black) and the 2D circles (red) models.

We analyze the discrepancy between the relatively simple 2D circles and the most realistic 3D spheres model, shown in Figure 4.4. We can see that the difference between the two A-scans is quite relevant: soon after the initial reflection (time step 1000), noticeable discrepancies emerge between the respective curves. We will now analyse the differences between subsequent models to determine what are the main physical factors that contribute to these variations.

Figure 4.5 shows the central A-scans of the 2D circles and 3D cylinder models. As can be seen, the disparities are minimal and, with the exception of minor discrepancies, the two A-scans are highly comparable. This is an important result, showing that the direct extension of a 2D model in 3D space does not significantly change the result of its GPR response.

To investigate the impact on the B-scan of the presence of rails, it is possible

4.1. Initial model comparison

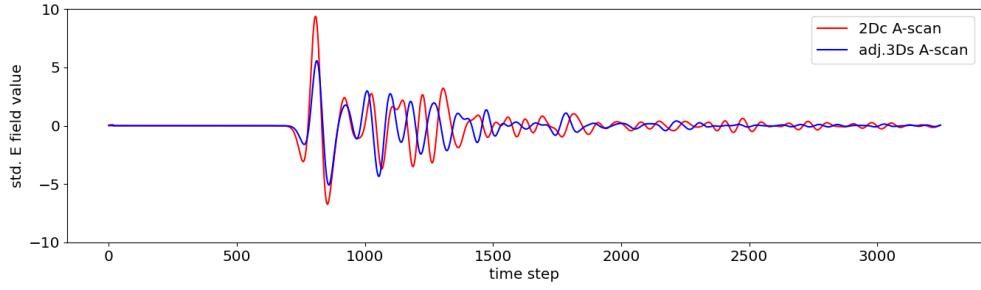


Figure 4.4: Comparison between central A-scans of the 2D circles (red) and the 3D spheres (blue) models.

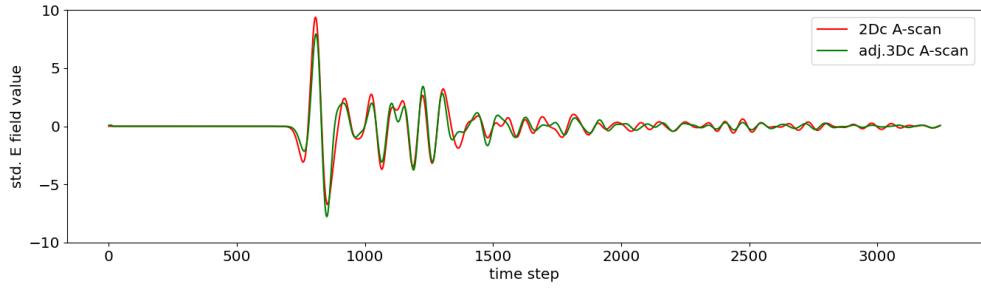


Figure 4.5: Comparison between central A-scans of the 2D circles (red) and the 3D cylinders (green) models.

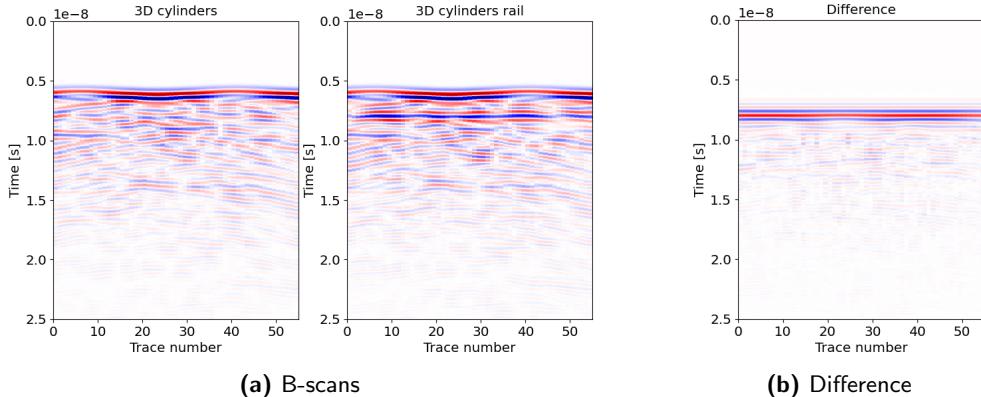


Figure 4.6: B-scans of 3D cylinders models with and without rails, and their difference.

to refer to Figure 4.6. On the left can be seen the B-scans generated for the 3D cylinders models with and without rails, while on the right their difference, with the same color interval. It is clear from the plot on the right that the two models lead to very similar reflections, with the exception of a well localized region due to the rail presence, between 0.7ns and 0.9ns. Because of its precise location, this region could, in theory, be added to 2D images as a sort of post-processing step in order to obtain more realistic B-scans.

4.1. Initial model comparison

The last difference that we consider in order to better understand the factors of change when we vary the model parameters is the one between 3D cylinders with rails and 3D spheres. Figure 4.7 shows that, after the initial reflection due to the surface of the ballast layer and to the rails, the two curves seem to decouple at time step 1200. This suggests that, with all other factors being equal, the topology and position of the ballast stones play a significant role in determining the EM response measured by GPR equipment.

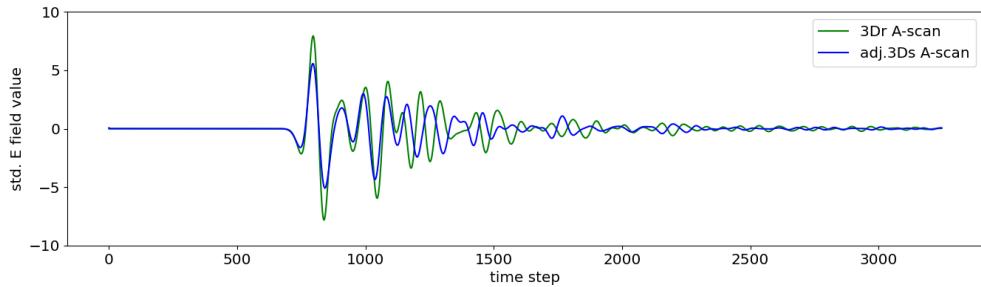


Figure 4.7: Comparison between central A-scans of the 3D cylinders with rails (green) and the 3D spheres (blue) models.

The last experiment we conduct on this initial model comparison consists of cutting the 3D spheres model geometry with a vertical plane parallel to the rails and placed at the same distance between them. Then we proceed to evaluate the similarity between central A-scans of the obtained 2D cut model and the original 3D spheres model. Figure 4.8 shows this comparison: it is possible to see that, already from the initial reflection, especially from time step 1000 onwards, the two curves diverge significantly from each other, lacking remarkable similarity. This suggests that, unlike our previous observations between 2D circles and 3D cylinders, the difference between 2D and 3D simulations is relevant when the 3D model is not directly conductible to an extension of the 2D one.

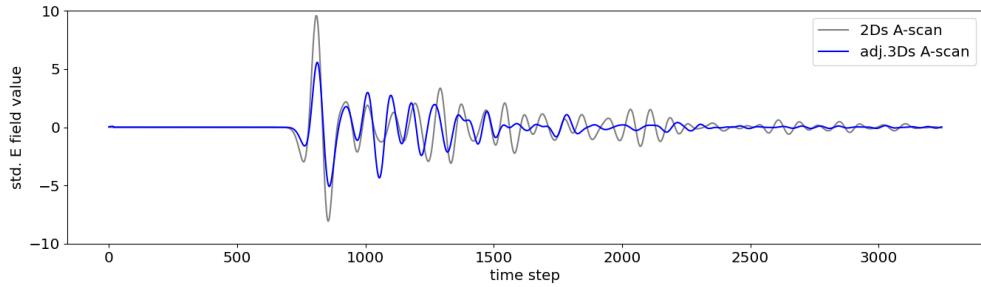


Figure 4.8: Comparison between central A-scans of the 3D spheres (blue) and the 2D cut from 3D spheres (grey) models.

Table 4.1 shows the Mean Squared Error (MSE) difference between the B-

4.1. Initial model comparison

scans obtained with the different models. Note in particular the smaller values of MSE between the 2D circles and 3D cylinders models, and between the latter variants with and without rails. Note also how, contrary to intuition, the values of MSE associated with the 2D cut from 3D spheres model are in general higher than any other model, even when compared with the visually very similar 2D circles model.

	2D box	2D circles	3D cylinders	3D cylinders with rails	3D spheres	2D cut spheres
2D box	-	0.42	0.37	0.67	0.67	0.74
2D circles	0.42	-	0.17	0.44	0.97	0.93
3D cylinders	0.37	0.17	-	0.31	0.88	0.88
3D cylinders with rails	0.67	0.44	0.31	-	0.51	1.06
3D spheres	0.67	0.97	0.88	0.51	-	1.03
2D cut spheres	0.74	0.93	0.88	1.06	1.03	-

Table 4.1: MSE between simulated models B-scans.

Concerning the computational time required to simulate these models, there exists a clear distinction between the 2D and their 3D counterparts: 2D models only require a couple of minutes to compute a B-scan on a machine equipped with a NVIDIA TITAN RTX graphics card, while 3D simulations require between two and three hours on the same machine.

To close this section, some conclusions can be drawn from the previous experiments:

- **3D models are too computationally expensive** to run when we need to generate thousands of samples. Running simulations non-stop for multiple days would only result in tens of B-scans, or a few hundreds of A-scans, which is simply not enough to train Deep Learning models on.
- **2D to 3D and rails have limited effects**, as long as the 3D model is a direct extension of the 2D one. Rail reflections are limited to a specific area and could be cut out and added later in a post-processing step to obtain B-scans more similar to those that would be obtained in a real-world scenario.
- **2D box and 2D circles are significantly different**, so we should adopt the more realistic (even if more computationally expensive) 2D circles model.
- **The topology and shape of the ballast matters**, as shown by the

difference between the 3D cylinders and 3D spheres models.

- **Cutting the 3D spheres model to obtain a 2D geometry leads to a very different result.**

As a result, the **2D circles** model is subsequently adopted for our further experiments, as this is both reasonably realistic and also computationally feasible.

4.2 Implementation

Most of the work regarding the implementation of the dataset generation concerns the fulfillment of the second objective delineated at the beginning of this chapter: ensuring that the created dataset will be sufficiently varied and representative of real world distributions. For both of these targets, the help, suggestions and real-world data provided by David Haener (our contact point at SBB) were extremely useful. Most of the design choices regarding the frequency of each configuration in this section are derived from data provided from this channel.

Initially, the design of the structure for the railway tracks was modelled after the layers present at the track at ETH Hönggerberg, and shown in the initial comparison (see section 4.1), with sleepers on top of ballast, PSS, asphalt and gravel. After meeting with David, a different and more realistic approach is followed. The track configurations specified in section 2.1 and shown in Figure 2.2 are implemented, namely with PSS, AC rail and regular gravel-sand subgrade, with a proportion of 40% for PSS tracks, 10% for AC rail and 50% for regular subgrade tracks.

All the track geometries and configurations are created with the *gprMax* software [9, 10], along with Python scripts. The pipeline process for dataset creation is shown in Figure 4.9. The first step is the creation of *gprMax* input files, which then get interpreted by the library to create 2D geometries with the specified materials. FDTD wavefield simulations are then run on these models. The output files from *gprMax* are automatically joined and post-processed, before being moved to the final dataset location.

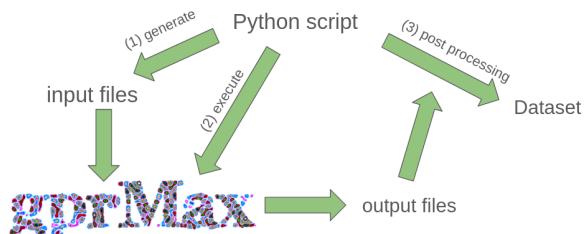


Figure 4.9: Schematic representation of the dataset creation pipeline. *gprMax* logo available on its official GitHub repository at <https://github.com/gprMax/gprMax>

In the following, the characteristics of the main elements present in the generated samples are outlined, as well as the reasoning behind their design choices.

Antenna

Two different antenna configurations are possible: a simpler hertzian dipole source combined with a receiver, or more complex antenna models created after real GSSI antennas, some of which are already available in *gprMax*. Unfortunately, this latter configuration is only possible when running 3D simulations, so the simpler approach is followed. The source is a Ricker pulse with a frequency of 1 GHz and both the source and the receiver are placed at a distance of 46.5 cm from the top of the sleepers, with 20 cm between them.

Rails

As already shown in the 2D models in section 4.1, rails are not added on top of the sleepers for 2D simulations. This design choice stems from the fact that this layer of steel would completely obscure the underlying elements and reflect most of the EM waves in the simulation. Consequently, minimal reflections would be measured from ballast and other layers beneath. This would also be a wrong design, as the 3D equivalent of including rails in the 2D simulation would involve a complete horizontal plane of steel at the rail level, which is clearly not the case in the real world. Furthermore, as remarked at the end of section 4.1, the rail reflection is very well localized in the output radargrams, and could be added as a post-processing step.

Sleepers

As briefly mentioned in chapter 2, three different materials are used for the construction of sleepers: concrete, wood and steel. All of these configurations are implemented into the random dataset generation, with frequencies of 50% for concrete, 25% for wood and 25% for steel. All the sleepers are modelled as rectangles (with each type their own size), and the distance between them (from the start of a sleeper to the start of the next one) is 65 cm. All the sleepers are embedded in the ballast layer by 70% of their height.

Ballast

As in the 2D circles model in section 4.1, ballast stones are modelled as circles stacked on top of each other. More details about the compaction mechanism used can be found in section 4.2.1.

Ballast fouling is simulated in two ways:

4.2. Implementation

- the average radius of stones is reduced, to mainly simulate light fouling due the crumbling of ballast material,
- a background layer of gravel and sand is added between ballast stones to simulate the finer particles of fouling material, indicating a high fouling index [61].

For each individual sample in the dataset, the diameter sieve for the ballast stones is determined by an internal variable, named *fouling level*, sampled between 0 and 1. The sieve constraints are calculated with linear interpolation between two pre-set sieves representing pristine and very fouled ballast, which are shown in Figure 4.10. The *fouling level* itself is not a uniformly random variable, but it follows a Beta(1.2, 2.5) distribution, depicted in Figure 4.11. When the *fouling level* is above a threshold (set at 0.7), the background fouling layer is added to the ballast, with a variable height to simulate different possible fouling conditions.

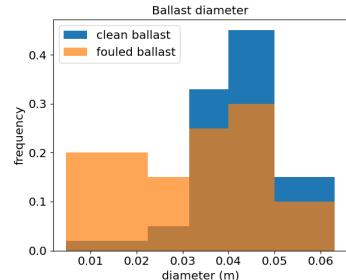


Figure 4.10: Ballast sieve distributions used as endpoints for the linear interpolation.

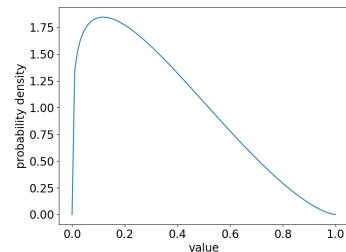


Figure 4.11: Probability density function for a Beta(1.2, 2.5) distribution.

Substructure and subsoil

Both the subgrade and subsoil layers are modelled using Peplinski [24, 25] mixed models of soil implemented in gprMax. The asphalt layer in the AC rail track type is added on top of the subgrade, while all the layers (including the fouling if present) have rough surfaces to better simulate a real scenario.

A *general deterioration* variable is sampled, in the same way as the *fouling level*, which influences the composition of the subgrade and subsoil layers in an analogous manner. By sampling from a Beta(1.2, 2.5) distribution, it is then used to calculate a linear interpolation between healthy and deteriorated compositions of the subgrade/PSS and the subsoil.

Water content and infiltration

The *general water content* of a sample is also sampled from a Beta(1.2, 2.5) distribution. This value is then used in two ways by sampling normal distributions centered on the *general water content*:

- The water contents of the subgrade and subsoil are drawn,

- The possibility to insert groundwater between the layers is added. If the sampled values are above a threshold, a water infiltration is added between layers, with the possibility of having this infiltration between each pair of subsequent layers.

The rationale behind this sampling approach stems from the premise that it is reasonable to assume a correlation between the moisture contents of various layers, as well as between these moisture levels and the occurrence of water infiltration. In other words, it is unlikely to find large differences between the moisture component of the subgrade and subsoil, or to find a water infiltration surrounded by very dry soil, as we assume is the case in reality.

Layer sizes

The layers have varying sizes, following a symmetric distribution given by a Beta(2, 2), depicted in Figure 4.12. As a result, the most frequent value for the size of a layer is its mean value. The ballast layer has a height between 20 cm and 35 cm, the asphalt in AC rails is between 5 cm and 8 cm thick, while PSS and regular subgrade have sizes between 25 cm and 40 cm.

Table 4.2 groups the properties of the materials used for dataset generation. All the regular materials present a magnetic loss $\sigma^* = 0 \Omega m^{-1}$, while all the Peplinski mixing soils use a sand particle density of $2.65 g cm^{-3}$, and a volumetric water fraction between 0.05 and 0.2.

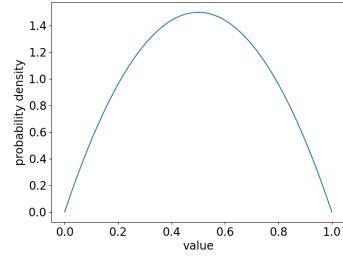


Figure 4.12: Probability density function for a Beta(2, 2) distribution.

4.2.1 Ballast compaction

The ballast creation algorithm presented in [60], while reasonably fast, presents some limitations. The main phases of this algorithm are:

1. Grading and generating ballast stones according to the underlying sieve curve.
2. Placing ballast stones in the domain size without overlap, using a Random Sequential Adsorption (RSA) algorithm.
3. Sequential compaction of the ballast stones in the horizontal and vertical directions.

Figure 4.13 shows the results of this algorithm. As can be seen, these are not optimal, as there are a lot of voids between the stones and in general they could be packed more closely, as in a real-world scenario where gravity acts.

material	ϵ_r	$\sigma (\text{S m}^{-1})$	μ_r
ballast	6.5 [62]	0	1
concrete	8 [63]	0.01	1
wood	2 [64]	0.01	1
steel	1	1.4e6 [65]	1000 [66]
asphalt	8 [63]	0.01	1

(a) EM Properties of regular materials			
material	sand fraction	clay fraction	bulk density (g cm^{-3})
fouling	0.28	0.12	1.9
PSS	0.35-0.48	0.02-0.3	2.3
regular subgrade	0.20-0.22	0.01-0.3	2.1-2.2
subsoil	0.27-0.65	0.03-0.35	2.0-2.1

(b) Properties of Peplinski materials, source: David Haener, SBB			
--	--	--	--

Table 4.2: EM properties of the materials used in dataset generation

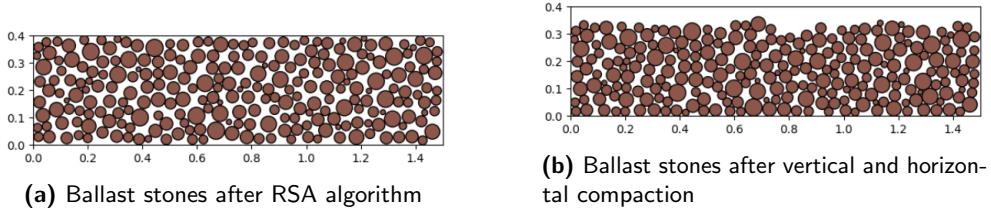


Figure 4.13: Ballast stones before and after compaction from [60]

For this reason, improved versions of the ballast compaction algorithm are developed, both in 2D and 3D. Both versions use physics engines to simulate gravity and compact the stones up to when relative movements are minimal.

2D physics simulation

The 2D compaction algorithm is implemented using the *Pymunk*[67] library, a flexible 2D rigid body simulation engine. This is the main implementation adopted for dataset generation: it creates and runs a different simulation for each sample. The total time required for generating compacted ballast stones is in general very similar to the code provided in [60], where both algorithms run on average in just over a second on our machine for the full compaction on a 1.5x0.4 meters space. This result is due to the

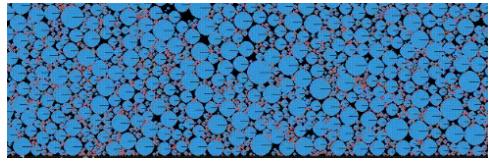


Figure 4.14: Result of ballast stones compaction using *Pymunk*[67].

fact that Pymunk supports efficient insertion and object collision detection primitives, allowing the RSA part of the algorithm to run in a fraction of time required by a more simple implementation.

3D physics simulation

A 3D counterpart to the 2D compaction implementation has been developed based on the PyChrono[68] library, as a standalone module. This module is used to generate the spheres in the 3D spheres model in section 4.1, and in general to test the feasibility of 3D simulations. Given the additional computational load of simulating a 3D space and the hugely increased number of objects, the simulation requires more time to compute, in the order of a couple hours.

4.2.2 Open Source contributions

During the development of the dataset generation scripts, multiple errors were found in gprMax, leading to their fix in the form of open source contributions to the library. The main problems found were:

- When using the library from the python APIs as opposed to the command line interface, and simulating multiple input files, in certain cases the FDTD grid was not re-generated for each file. This caused the simulations to be run only on the first file multiple times. When triggered, this bug induced all the outputs to be the same, even if different simulations were run.
- The built-in material used to simulate water was not set to be used as a Debye dispersive material, leading to the relative permittivity used in simulations to be the one at infinite frequency of 4.9, when the actual value at GPR frequencies is around 80 [69].
- An error in the Peplinski soil materials caused a possible repetition in material names, which in the best case resulted in wrong materials being used for part of the simulations. In many cases the simulations would simply crash because of the fact that less materials were present than indexes in the FDTD grid.
- An error involving the Perfectly Matched Layer (PML), caused the boundaries of the simulation to often reflect the EM waves when the steel material was used, causing completely wrong A and B-scan simulations.

4.3 Results

Two different datasets are generated with the configurations specified above.

- an **A-scan dataset** of 30000 samples, where source and receiver are placed (centrally) at 65 and 85 cm from the left border. Snapshots of the electric field are taken at 1 ns intervals, to enable the training of PINNs (see chapter 5).
- a **B-scan dataset** consisting of 4000 samples with 90 A-scans each, with a separation of 1 cm between them. No snapshots of the electric field are saved. This dataset is used in the training of the black box CNN model, see Section 5.1.

The computational and storage necessities of the two datasets are very different: the A-scan dataset only accounts for around 2 days of simulation time on a machine equipped with an AMD EPYC 7742 processor and a NVIDIA TITAN RTX graphic card, but it requires around 900 GB of storage to save all the field snapshots. On the other hand, the B-scan dataset only occupies around 80GB of data, but needs around 2 weeks for simulation, as the number of A-scans to simulate is much larger ($4000 \cdot 90 = 360000$).

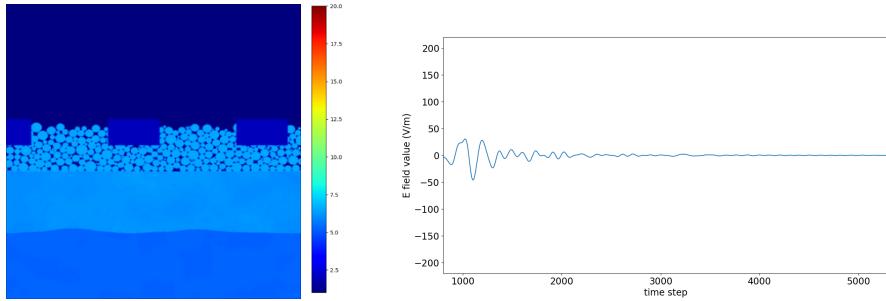
Figure 4.15 shows relative permittivity (ϵ_r) plots and Ascans of some generated samples from the first dataset. ϵ_r is used in the plots as it is one of the main EM properties of the materials, furthermore it is also a good indicator of the amount of moisture present, as water has a very high relative permittivity. This makes wet soils stand out, as ϵ_r increases with the moisture level. Figure 4.16, shows analogous samples from the B-scan dataset.

4.4 Limitations of the current design

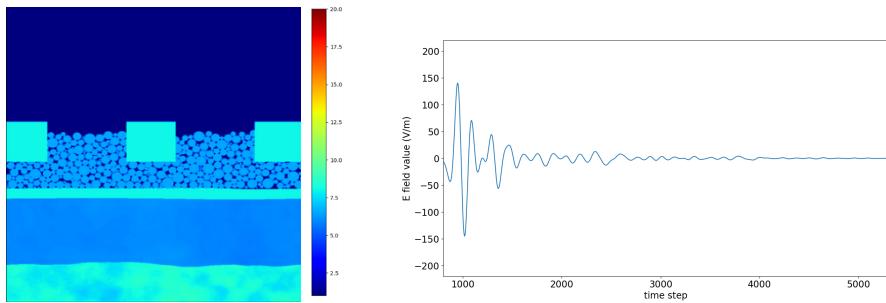
Although we believe that the presented dataset expresses a reasonably realistic implementation of different railway track conditions, some limitations are nonetheless present. In particular, we observe that:

- 2D simulation is not the most realistic way to represent real-world data. In particular, 2D simulations necessarily imply the absence of an antenna model and of the rails. This causes a lack of reflections due to these components in the output of the simulations, which could partially be added in future post-processing steps.
- The samples, although quite varied, do not represent extreme deterioration of the tracks, nor include any external object that could be present around a real railway track, e.g. pipes, rods, potholes, signals, tunnels etc.
- Ballast fouling is challenging to simulate accurately, as the number of objects to simulate grows significantly with the fouling level.

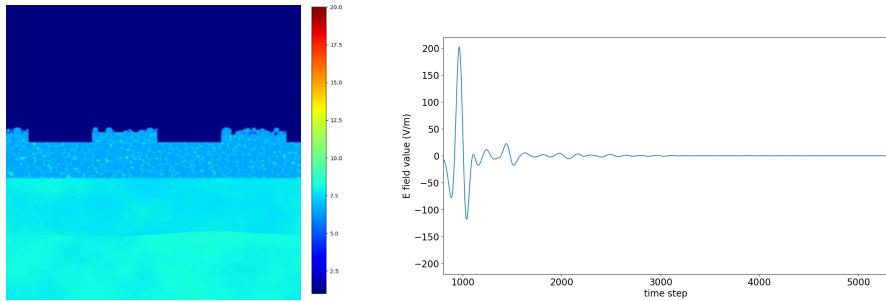
4.4. Limitations of the current design



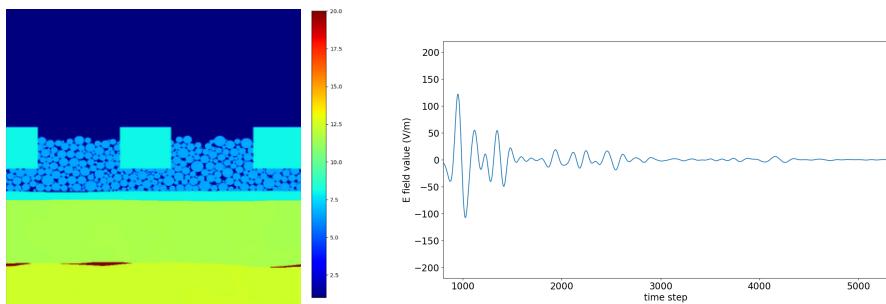
(a) Dry PSS track with wooden sleepers and clean ballast. Both the sleepers (time step 1000) and later reflections are small, compatible with the small change in ϵ_r between interfaces.



(b) Concrete sleepers on an AC-rail track. Notice that the initial reflection is much stronger compared with Figure 4.15a and the reflection from the asphalt layer is visible at time step 2500.



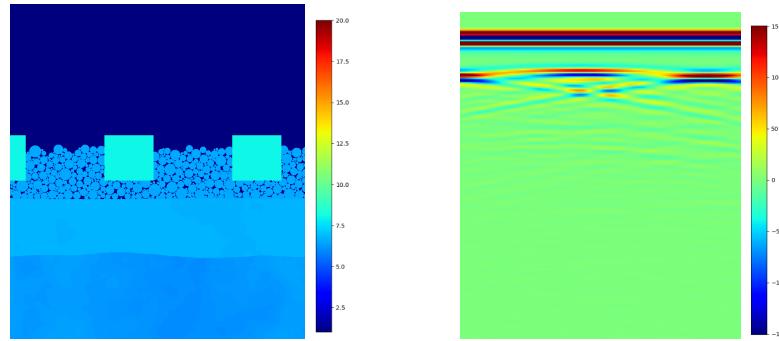
(c) Steel sleepers (invisible in the image) on a fouled ballast and mildly wet subgrade and subsoil. The sleepers create a very strong initial reflection due to conductive properties of the metal.



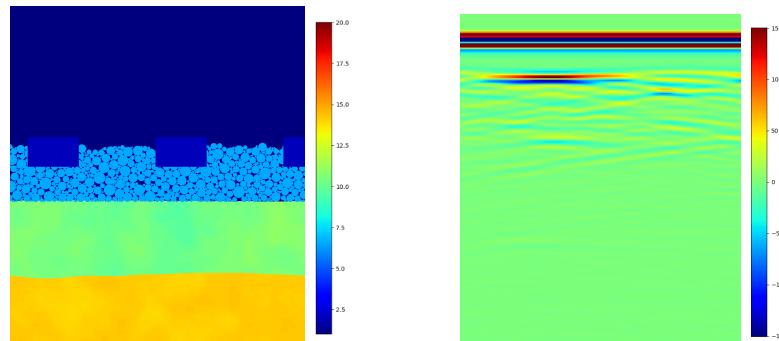
(d) Groundwater between the PSS and the subsoil, both of which are very wet. The A-scan shows both the reflection due to the asphalt layer and the one due to water, at time step 4100.

Figure 4.15: Relative permittivity plots (left) and central A-scans (right) of samples generated in the A-scan dataset.

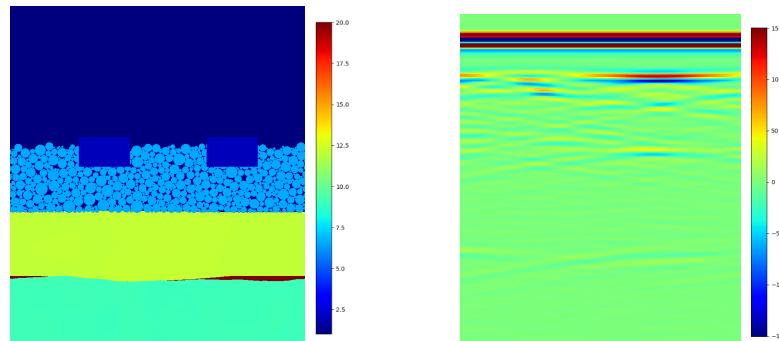
4.4. Limitations of the current design



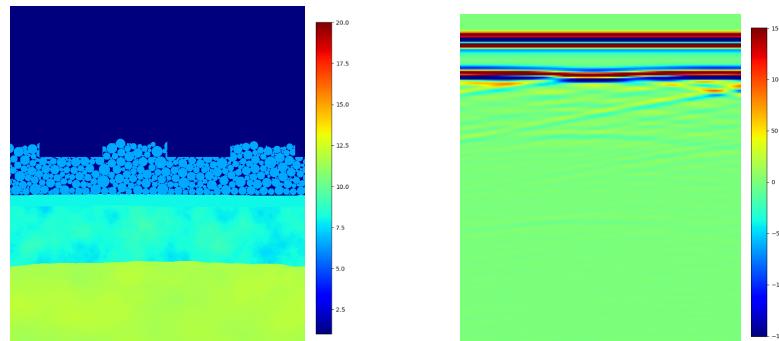
(a) Concrete sleepers on dry PSS and subsoil.



(b) Wooden sleepers on wet subgrade and deteriorated subsoil, with slightly fouled ballast.



(c) Water infiltration between PSS and subsoil, note the associated reflection in the B-scan.



(d) Steel sleepers, notice the strong reflection on the B-scan.

Figure 4.16: Relative permittivity plots (left) and resulting B-scans (right) of samples generated in the B-scan dataset.

4.4. Limitations of the current design

- Wooden sleepers and water infiltration are not extremely realistic. The sleepers are modelled with fixed EM properties, so wooden sleepers, which in reality are subject to the absorption of water, do not reflect it in the simulations. The materials around the groundwater are not (in general) gradually wetter as the distance to the infiltration diminishes, although in the dataset the presence of groundwater is more common where wetter soils are present.

Chapter 5

ML surrogate models

This chapter will expand on the second objective of this thesis: the implementation and training of ML-based models for fast inference of radargrams given material properties and positions, in the form of 2D geometry maps. Section 5.1 presents a purely data-driven black box approach in which no physical laws are considered, section 5.2.1 shows the implementation and limitations of a PINN based on the MLP architecture, section 5.2.2 discusses the application of a CNN-based PINN.

5.1 Black box model

The first model we present consists of a black box CNN architecture adapted from Dai et al. [48]. As partly mentioned in section 3.1.2, this architecture consists of two separate convolutional encoders for ε_r and σ maps, a feature fusion module based on attention mechanisms, and a convolutional decoder that outputs the predicted B-scans.

Each block in the encoders is formed by two 3x3 convolution, each followed by a ReLU activation, and a 2x2 max pooling operation. The adaptive feature fusion module, shown in Figure 5.1, employs two cross attention (CA) layers, followed by a connection block. The latter comprises a strided convolutional layer, followed by a transposed convolution and a regular convolution, each followed by a ReLU activation. The Decoder is also formed by blocks, each composed by an upsampling layer and two 3x3 convolutions, each of the latter followed by a nonlinearity. A final 1x1 convolution is applied to the output and acts as a regressor.

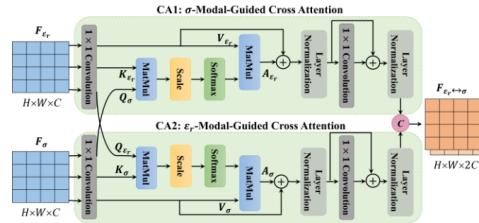


Figure 5.1: Attention-based feature fusion module, from [48].

5.1. Black box model

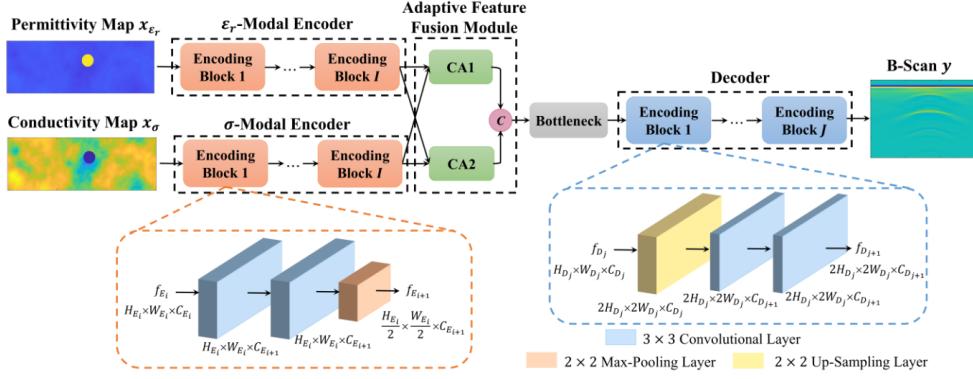


Figure 5.2: Structure of the black box CNN architecture, from [48].

Figure 5.2 shows the overall structure of the network.

The main modifications in the employed model compared to the one in [48] are i) both feature encoders employ an additional encoding block, and ii) the connection block responsible for the fusion of the attention-captured features employs a 3x3 convolution with 3x3 strides, which effectively acts as a parametrized downsampling operation. The model is implemented in Keras [70], using the Tensorflow [71] backend. It was trained for 300 epochs with a batch size of 10, the Adam optimizer was employed coupled with a MSE loss. The initial learning rate was set to 10^{-4} and a LR scheduler was adopted to reduce it automatically when needed. The best model was selected as the one with lowest validation loss. Some pre-processing of the input data was performed, namely a resizing of the labels to the prediction size of 192x224 pixels, and the pixel-wise median value of the labels was calculated and subtracted from the images: this allowed to remove the initial wave response from the function learned by the network without setting specific areas of the images to zero and thus possibly having sharp contrasts in the resulting labels. This processing makes the learning task easier and empirically improves the overall performance of the model. The median mask is shown in Figure 5.3.

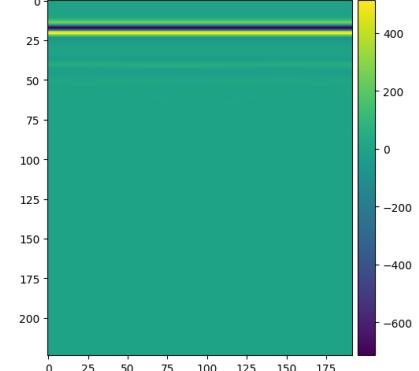


Figure 5.3: Median mask employed for pre-processing of the labels in the black box CNN model.

Results

Figure 5.4 shows some validation dataset samples, ground truth B-scan and predictions of the network. As we can observe, the model predictions show a

5.1. Black box model

remarkable similarity to the labels, with only minor differences. This result is, in our opinion, by itself a significant contribution of our work. The paper from Dai et al. [48] adopts extremely simple geometries consisting of a single extraneous body introduced in an heterogeneous medium (see upper left part of Figure 5.2), while we apply our model to realistic railway configurations. To our best knowledge no published study shows ML-based predictions of GPR data with a comparable level of complexity of the input geometries and accuracy of the predicted B-scans.

On the model predictions, sleepers responses are generally very accurate with only minor discrepancies, and are shown to be precise on all three sleeper types (concrete, wood and steel). An analysis of the sample loss values reveals that the least accurate predictions occur where groundwater appears between the layers, especially at the interface between subgrade and subsoil. Some of the highest loss samples are shown in figure 5.5. Another common issue appears to be a suboptimal subgrade reflection, as shown in Figure 5.5d.

Regarding computational cost, the black-box model is extremely efficient: inference for full B-scan predictions on a batch of samples only takes around 300ms on our setup, where a batch is composed by 30 samples. Comparatively, as mentioned in Chapter 4, computing the B-scan response of a single dataset geometry requires around 5 minutes on the same machine, which constitutes a speedup factor of ≈ 30000 . When using this model for B-scan generation, this very large speedup factor effectively shifts the bottleneck of dataset generation to the creation of the randomized sample geometries (which takes around 1s per sample). This allows for a general speedup of the process by around 300 times.

5.1. Black box model

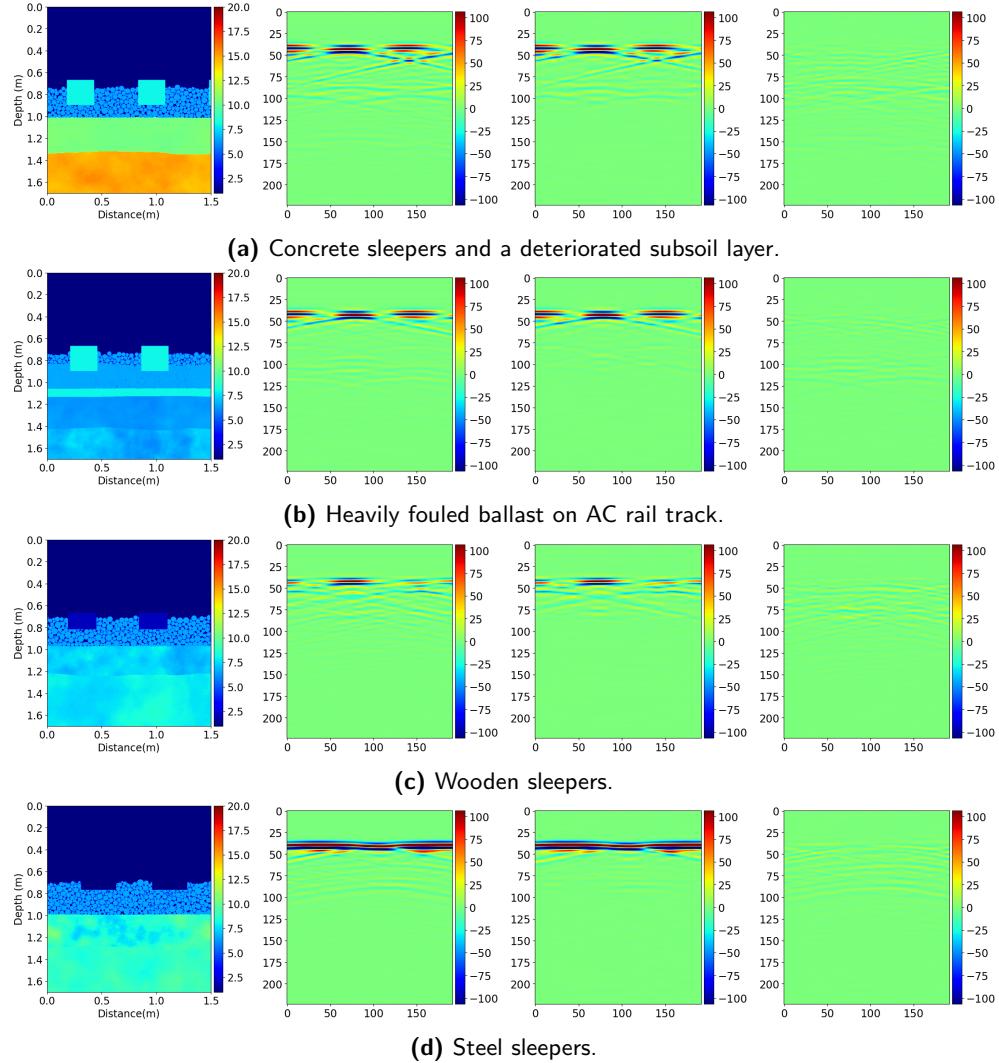


Figure 5.4: From left to right: sample ϵ_r geometry, ground truth, black-box model predictions and difference with ground truth.

5.1. Black box model

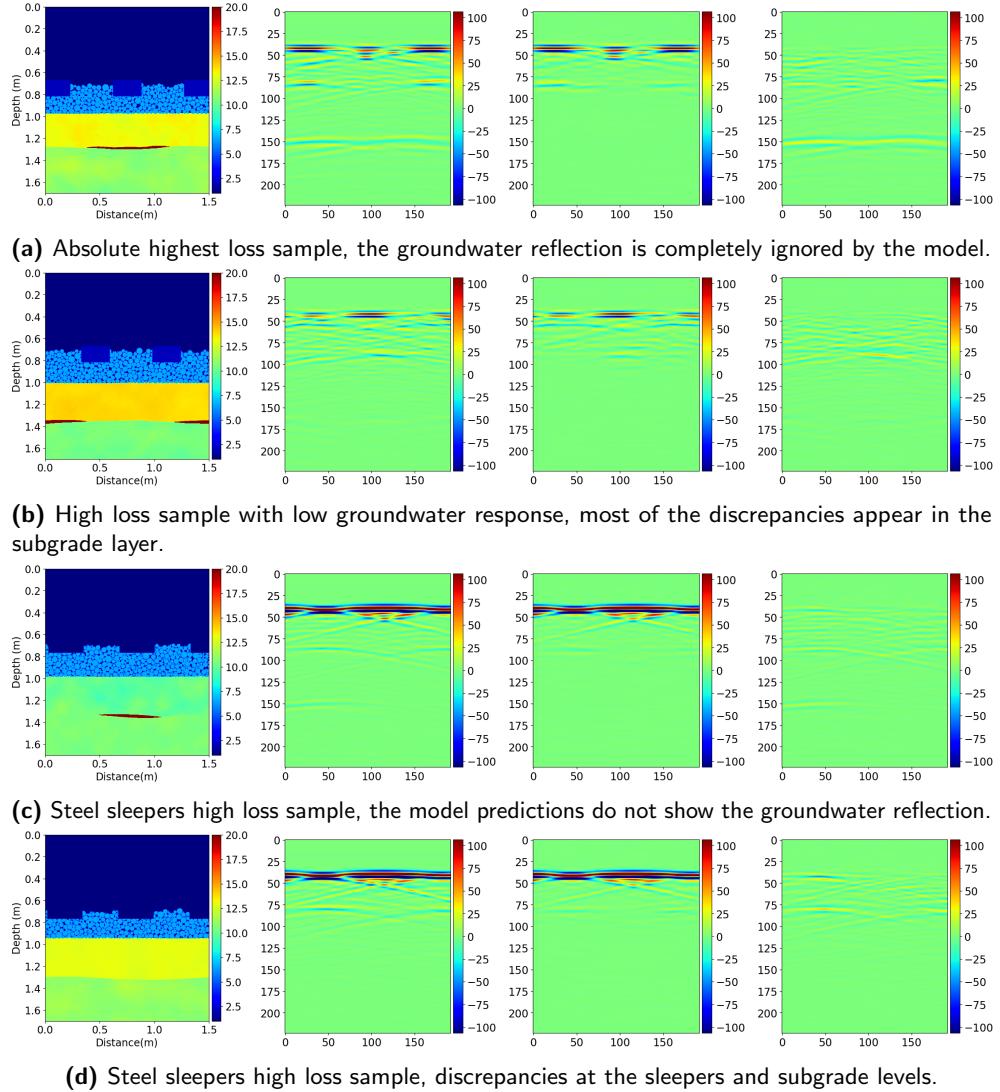


Figure 5.5: From left to right: sample ϵ_r geometry, ground truth, black-box model predictions and difference with ground truth of some of the samples with highest difference between the prediction and label.

5.2 PINN models

Physics-informed neural network models exploit physical PDEs and boundary conditions to apply additional loss terms to the training of NN models. In order to do so, the quantities expressed in those equations must be derived from some outputs of the networks, either final or intermediate results. Regarding in particular GPR measurements, the PDE satisfied by the propagation of the electric field in materials, from [34], is:

$$\frac{\partial^2 E}{\partial t^2} - \frac{1}{\mu\epsilon} \nabla^2 E + \frac{\sigma}{\epsilon} \frac{\partial E}{\partial t} = 0 \quad (5.1)$$

where $\nabla^2 E = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2}$ in two dimensions.

As can be seen, a pipeline used to train a PINN model which satisfies equation 5.1 necessarily needs to compute an estimate of three physical quantities related to the electric field E : its first and second derivatives in time and its second derivatives (w.r.t. x and y) in space. This necessity poses serious additional constraints into the design of PINN architectures. For instance, it is highly not trivial to apply the PINN framework to the model presented in Section 5.1. The only electric field measures present in that architecture are the predicted B-scan in the output, which only consist of values estimated at a single point in space for each A-scan, e.g. at the receiver. All the following models attempt to capture a evolving representation of the electric field through time and space in the full domain. In particular we focus the application of PINNs on the problems of sparse reconstruction and domain extension. The former is investigated with the purpose of obtaining precise GPR predictions in between training snapshots. The latter is explored for the possibility of using PINNs as forward modellers of GPR data with only limited initial observations.

5.2.1 MLP models

We present a family of physics-informed multi-layer perceptron (MLP) models for the prediction of electric field values at arbitrary points in space and time, associated with increasingly complex geometry maps. All the models have 3 inputs, which correspond to 2D spatial and temporal coordinates (x, y, t) and one output, which is the electric field. Figure 5.6 shows a representation of such models. We follow a

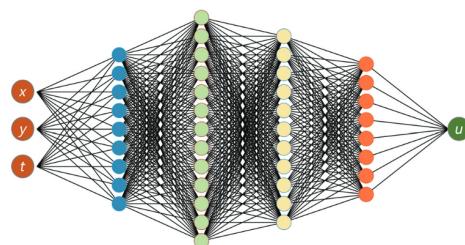


Figure 5.6: Schematic representation of the MLP pinn models, from [59]. The input is spatial and temporal coordinates (x, y, t), the output is the electric field.

similar approach to the one used in [59] for the first two models, namely the uniform and two-layer ones, which allows us to incrementally increase the complexity of the relative geometry maps and resulting wavefields. All the models are implemented in PyTorch[51]. The computation of the quantities required in Equation 5.1 are calculated using automatic differentiation between the inputs and output of the networks. All models used input and output pre-processing, namely all the input data was scaled into the $[0, 1]$ range and the labels between $[-1, 1]$.

Uniform model

The first and simplest model we implement simulates a Ricker waveform with a central frequency of 150 MHz propagating through a uniform medium with constant permittivity and conductivity. The medium parameters are: $\epsilon_r = 1$, $\sigma = 0$, $\mu = 1$, $\sigma^* = 0$. The space size is 20×20 m and the source is located at the center. gprMax is used to approximate the evolution of the wavefield using the FDTD method, generating snapshots of size 200×200 pixels. Figure 5.7 shows the two snapshots employed as observations, at 15 and 25ns. A total of 40000 static collocation points are sampled between 15 and 45ns, which are fixed during all the training of the network. No boundary conditions are applied. For this geometry, two models with 5 layers of 128 neurons each are trained: a PINN with SiLU activation functions and a black-box ReLU network to act as a non physics-informed baseline. The training process uses the Adam optimizer over full-batch gradient descent. As suggested in [54], a pre-training phase of 20000 iterations with L4 loss is applied, in which only observation points contribute to the loss of both models. Next, a 3000 iteration training over MSE loss takes place, with collocation points contributing to the loss of the SiLU model.

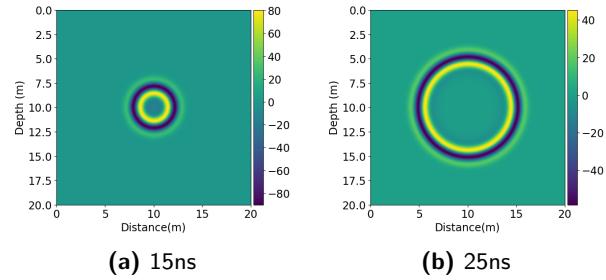


Figure 5.7: Training wavefields used in the uniform model

Figures 5.8 and 5.9 show the predictions of the models at 20ns and 40ns, to respectively emphasize their effectiveness at sparse reconstruction (20ns) and domain extension (40ns). As we can observe, the physics-informed model predictions closely align with the ground truth, while the ReLU network produces a poor prediction for $t = 20$ ns, and an even more inaccurate one for $t = 40$ ns.

This outcome, which reproduces the first result by Zheng and Wang [59], was not trivial to achieve. In particular, we observed that two main challenges

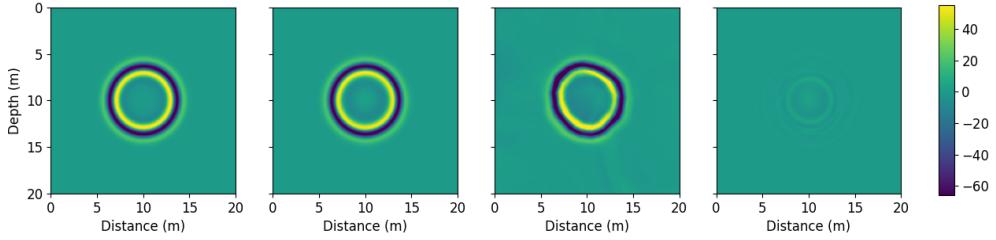


Figure 5.8: From left to right: ground truth, PINN predictions, NN prediction, difference between PINN prediction and ground truth. Uniform geometry, at $t = 20\text{ns}$.

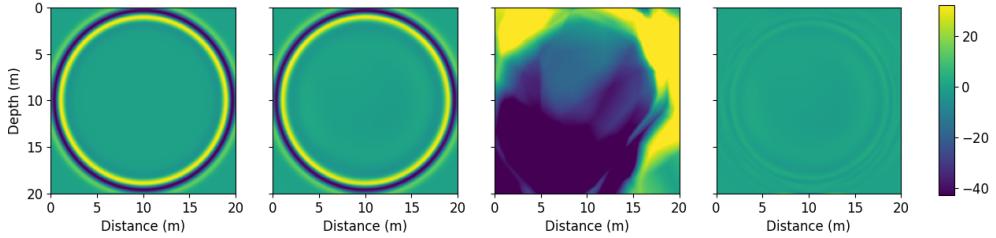


Figure 5.9: From left to right: ground truth, PINN predictions, NN prediction, difference between PINN prediction and ground truth. Uniform geometry, at $t = 40\text{ns}$.

arise when applying physics-informed training to this simple architecture. First, all the quantities and physical parameters forming the PDE must be rescaled to express the actual real-world quantities, e.g. time must be expressed in seconds (not ns), E must be expressed in V/m, $\epsilon = \epsilon_0 * \epsilon_r$, etc. Second, proper scaling values must be chosen for the combination of the observation and physics loss. EM waves travel (in void) at the speed of light, which causes very large factors to appear when computing both derivatives with respect to time and space. This results in a multiplicative factor of $2 \cdot 10^{-17}$ being applied to the PDE value, even before the MSE loss, which induces a value of $\lambda_p = 4 \cdot 10^{-34}$.

Two layer model

The second geometry that we analyze consists of a two layer model, where two materials with different EM properties are placed on top of each other, as seen in Figure 5.10. The analysis of the predictions of PINNs in this context enable us to gain useful insights on their ability to model reflection and refraction of EM waves, as in this scenario it is possible to observe in detail both phenomena.

Three models are trained with different configurations: the first model is trained (as in the

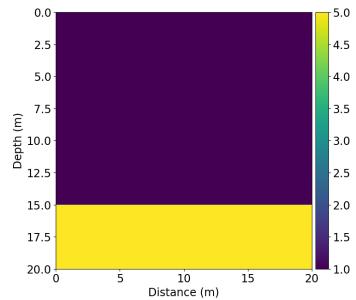


Figure 5.10: Relative permittivity plot of the two-layer model geometry. The source is placed at the center of the space.

5.2. PINN models

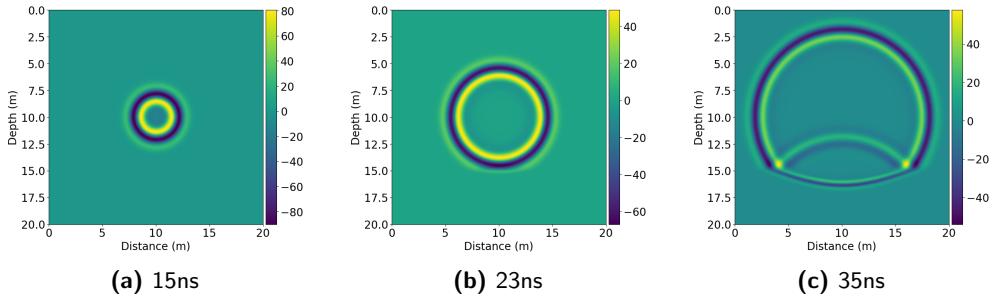


Figure 5.11: Some training wavefields in the two layer model

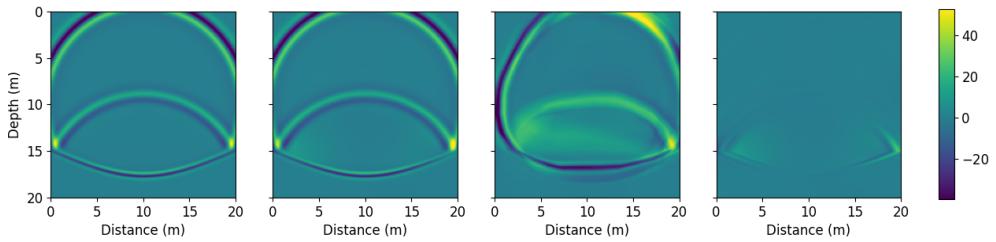


Figure 5.12: From left to right: ground truth, PINN predictions, NN prediction, difference between PINN prediction and ground truth. For the two layer geometry trained on snapshots between 15 and 35ns, at $t = 45$ ns.

publication of Zheng and Wang [59]) on snapshots between 15 and 35ns (see Figure 5.11), were it is already possible to see the reflected waves caused by the change of material. Figure 5.12 shows the predictions of the model and a non physics-informed counterpart on $t = 45$ ns. All the models are trained using 10000 steps of L4 pre-training followed by 20000 steps of MSE training with the Adam optimizer.

The second and third two-layer models are created to assess the ability of PINNs to model electromagnetic wave reflection and refraction when no observation data of these phenomena is provided to the network. Both are trained on snapshots between 15 and 23ns (see Figure 5.11). Model 2 employs the same loss function as in model 1, while in model 3 electromagnetic boundary conditions are applied at the interface between the two materials. These are, in the absence of unbound surface charges [72, 73, 74]:

$$\hat{n} \cdot D_2 = \hat{n} \cdot D_1 \quad (5.2)$$

$$\hat{n} \times E_2 = \hat{n} \times E_1 \quad (5.3)$$

Where \hat{n} represents the propagation direction of the EM waves. Since this experiment involves 2D simulation, the electric field is only calculated in the z direction, as it is null in both x and y orientations. This means that the

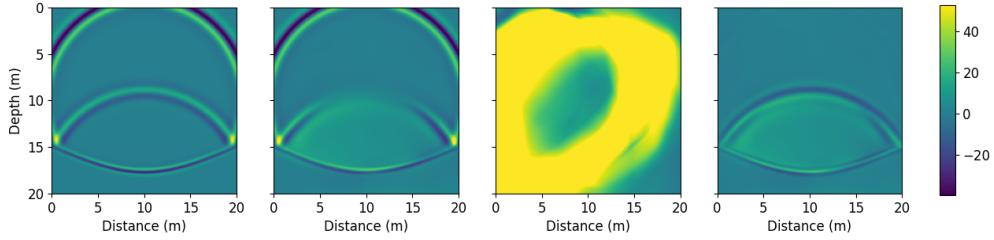


Figure 5.13: From left to right: ground truth, PINN predictions, NN prediction, difference between PINN prediction and ground truth. For the two layer geometry trained on snapshots between 15 and 23ns, at $t = 45\text{ns}$.

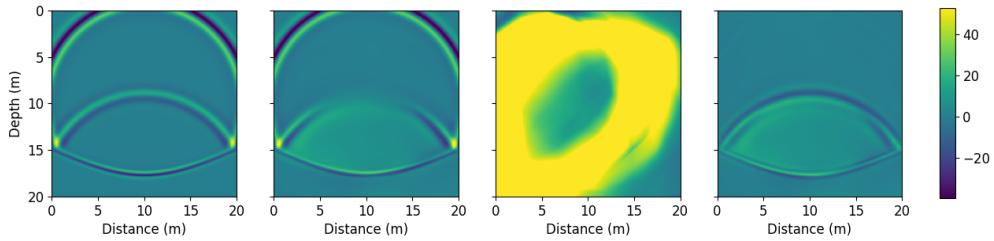


Figure 5.14: From left to right: ground truth, PINN predictions, NN prediction, difference between PINN prediction and ground truth. For the two layer geometry trained on snapshots between 15 and 23ns, with boundary condition, at $t = 45\text{ns}$.

propagation direction is in every point of the simulation perpendicular to the electric field, so Equation 5.2 results in $0 = 0$. For this reason, only Equation 5.3 is implemented. Figures 5.13 and 5.14 show the predictions of these two models at $t = 45\text{ns}$, coupled with a regular (non physics-informed) network. As we can see, the results are very similar, independently of the application of the boundary condition. They both show partial reflection, showing that, at least in the tested conditions, PINNs are not able to accurately capture the full wavefield propagation. Given these results, all the subsequent PINN models are only trained with the additional PDE loss and no boundary conditions are applied.

Application to the dataset

We show the application of the aforementioned PINN MLP model to the A-scan GPR dataset created in Chapter 4. Unfortunately, our experience is that no MLP model (physics-informed or not) is able to accurately capture the full wavefield obtained from a single railway dataset geometry. Additionally, the training of such models reveals extremely unstable. In this regard, they often only marginally differentiate between their predictions and the trivial solution of $E = 0$ everywhere. Many approaches are investigated in the attempt to tackle this problem, some lead to slightly better performance, but are still far from accurate predictions.

To name a few: varying the sampling rate of observations; adjusting the learning rates and λ values for the combination of PINN losses; applying time-dependent weighted losses; additional label pre-processing with a cube root to reduce the amplitude gap between main wave and reflected ones; different network activation functions, architecture shape and batch sizes; application of an additional loss penalizing samples that predicted values different from the label a-scan; usage of collocation points sampled on a per-batch basis; application of Xavier

initialization for network weights [75]; gradient clipping on the network parameters. Figure 5.16 shows predictions of the best (based on visual inspection and loss values) PINN and NN models. Both models were trained on samples from 2ns to 19ns, the PINN model had a observation snapshot interval of 1ns, while the non physics-informed one used an interval of 0.2ns. Additionally, the training of the black box NN model employed an additional cube root pre-processing of the labels before the data scaling. The computational cost for training both models is relatively high: the PINN architecture required two hours of training, while the NN employed almost 10 hours of training time. Both models were trained for 100 L4 warmup epochs with a batch size of 512 and 100 MSE training epochs with a batch size and a number of collocation points per batch equal to 8192.

Some hypotheses can be formulated on the reason behind these less than satisfactory results: the high number of wave reflections might be a contributing factor. Zheng and Wang [59] show a similar result on the most complex geometry in their experiments (see Section 3.2.2), where they speculate that the spectral bias of neural networks [76] might be involved. This is a phenomenon which causes the networks to not learn the high frequency components of the solution in a reasonable time frame. Other, conceptually simpler reasons might involve the size and architecture of the networks, which in this study is restricted to MLPs with constant hidden size of at maximum 512 neurons per layer, repeated for 5 layers.

The computational time needed to train these pixel based models is very high: training on the full A-scan dataset would require days, if not weeks of training time, and given the aforementioned results on a single geometry and other factors, we decide not to continue our research on MLPs in favor of investigating a different approach, namely the implementation of a physics-informed CNN.

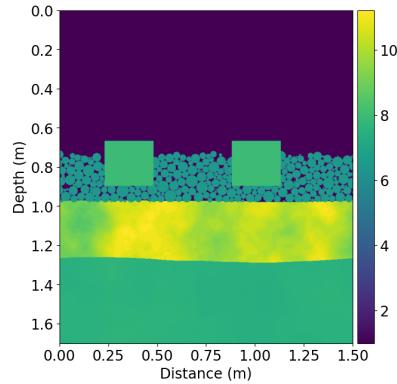


Figure 5.15: Relative permittivity plot of the rail track model geometry.

5.2. PINN models

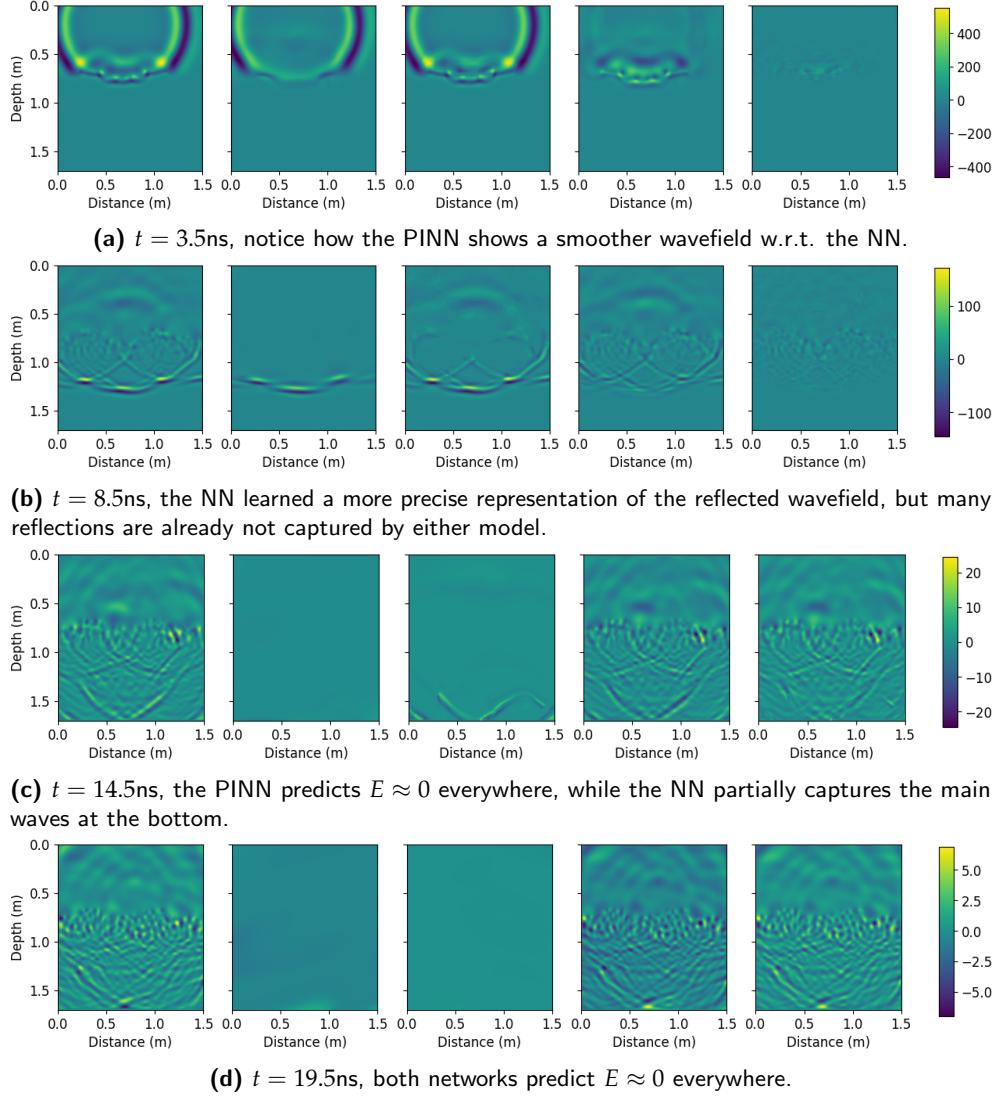


Figure 5.16: From left to right: ground truth, PINN predictions, NN predictions, PINN difference, NN difference with ground truth. Applied to the railway dataset geometry trained on snapshots between 2ns and 19ns every 1ns.

5.2.2 CNN model

We present a physics-informed CNN model denoted as *time2image*.

This model takes the time coordinate as input and outputs a 2D image of the electric field at the specified time step. A schematic representation of this architecture is displayed in Figure 5.17.

It features an initial MLP with increasing layer sizes, followed

by two convolutional decoding modules. Each of these is composed by an up-scaling transposed convolution and two 3x3 convolutions, each of the latter followed by an activation function. As in the black box CNN, a 1x1 convolution is placed at the end of the network to act as a regressor.

Two main differences are present in the computation of the PDE loss for this model with respect to the MLPs shown in Section 5.2.1:

- **Forward mode automatic differentiation** is used to calculate first and second order derivatives of the E field image with respect to the input time coordinate. This change is unavoidable because of the necessity of computing derivatives of multiple outputs with respect to a single input. Multiple backward differentiation steps (equivalent to the number of pixels in the output image) would be necessary, resulting in an extremely slow training process. With this adjustment, a single forward differentiation step is sufficient.
- **Numerical differentiation** is used on the output image to compute the second derivatives in space. A normalized Sobel [77] kernel is used as the differentiation operator. This modification is necessary due to the impossibility to exploit automatic differentiation with this architecture for the calculation of spatial derivatives. This limitation arises because neither x nor y are included as input or intermediate results of the network.

A PINN model and its non physics-informed counterpart are trained on the wavefield generated from the geometry in Figure 5.15. They use respectively ReLU and GELU activation functions and the MSE loss coupled with the Adam optimizer. Figure 5.18 shows the first and last training samples at 2ns and 10ns, while Figure 5.19 shows the pre-

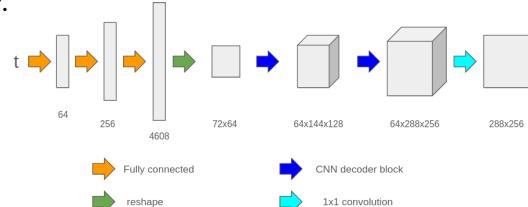


Figure 5.17: Schematic representation of the *time2image* model

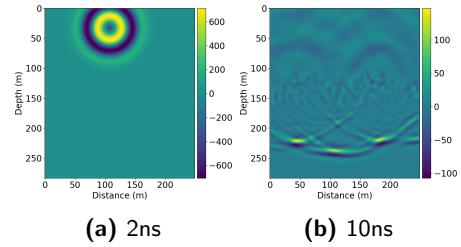


Figure 5.18: first and last training images for the *time2image* model.

dictions of the two networks at different points in time. As shown, both networks successfully capture the training data. However, they exhibit deficiencies in sparse reconstruction and domain extension capabilities. These properties are typically expected from a PINN model and in particular sparse reconstruction is crucial for precise B-scan prediction. Three hypothesis were formulated in order to explain this behaviour:

1. Since the time2image model requires a large amount of GPU memory, only a relatively **small number of collocation points** is used in each iteration: the aforementioned MLP models are trained on 8000-40000 collocation points in (x,y,t) coordinates, while the time2image model is only trained on 50 time coordinate collocation points resampled each iteration, which nevertheless produce a PDE residual on a full image, i.e. $284 \times 250 = 71000$ points each. The time sparsity of the points might be one of the causes that lead to the shown result.
2. **Numerical differentiation** could be the cause of instability for multiple reasons: firstly, the accuracy of the numerical differentiation is less than perfect. This particularly considering the second order derivative used in these experiments. These are calculated as a weighted sum of the values in neighboring pixels, which cannot always correctly estimate their real point-values. The measurement of PDE residuals is also always evaluated at the same grid-spaced spatial coordinates (determined by the position of pixels in the image) and this could influence the effectiveness of this method.
3. Other issues related to the **discretization of the output domain E** , which might also arise independently from numerical differentiation, could disrupt the convergence of the model to the physical solution. One explanation might involve the fact that, in a discrete domain, the enforcement of relations based on second order derivatives, which constraints the *shape* of a curve, might not be sufficient to influence the *value* of the discrete output.

Given the results shown in Figure 5.19, the fact that the black box model shown in section 5.1 captures the complexity of the dataset with good accuracy and the time limits for this work, we choose to investigate the underlying reasons for this phenomenon.

5.2. PINN models

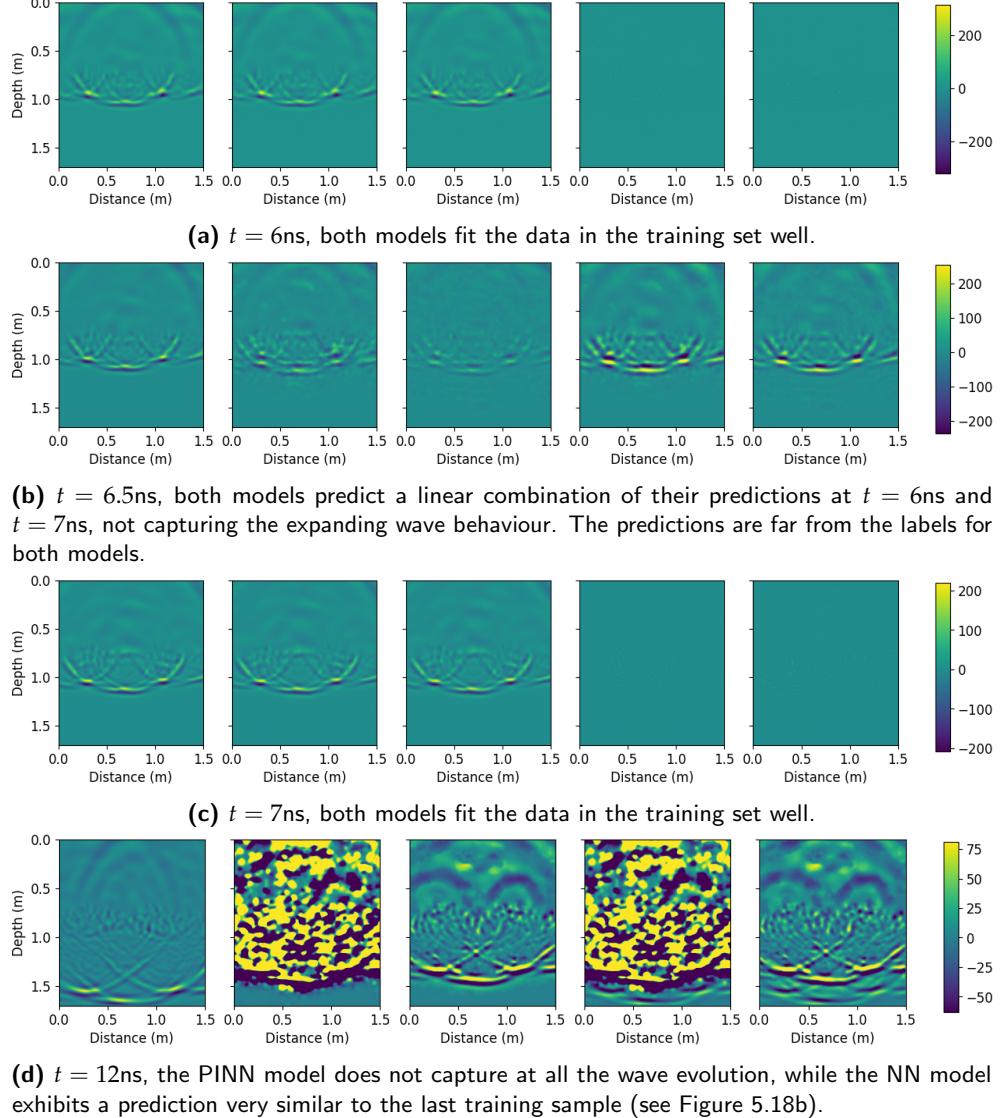


Figure 5.19: From left to right: ground truth, PINN predictions, NN predictions, PINN difference and NN difference with ground truth. Applied to the time2image model on the railway dataset geometry. Neither model exhibits sparse reconstruction (b) or domain extension (d) capabilities.

1D wave on MLP/CNN

In order to investigate and better understand the reasons for which the *time2image* model performance does not significantly improve when applying the physics-informed loss, we devise a further experiment: this consists of a 1D wave propagating towards the positive x direction in time. Three different models are trained on this data: an MLP with x and t as input as in Section 5.2.1, a *time2image* CNN model adapted to predict 1D sequences and a further MLP that takes the time coordinate as input and outputs a vector predicting the full wave at the corresponding time, which is denoted *expanding MLP*. Five observations were used for training, at times corresponding to $t = [0, 1, 2, 3, 4]$ s, and wave speed of 16 m/s. Figure 5.20 shows the first and last train samples of the networks. The spatial domain is 512m for the MLP model, with 20000 (x, t) collocation points spread in $t \in [0, 40]$ s and corresponding 512 pixels for both the CNN and expanding MLP architectures with 2000 collocations in the same time domain.

As can be seen in figure 5.21, both MLP models (with and without PDE loss) show remarkable sparse interpolation capabilities, while both the CNN and expanding MLP models are not as precise. For domain extension, both expanding MLP and CNN models are very imprecise, with the physics-informed predictions being generally more stable, but still not capturing the wave evolution at all.

Based on the presented result, we feel confident in excluding the small amount of collocation points as the cause for the *time2image* model imprecise results, as the number of collocation points in this last experiment is much higher and predictions are still very inaccurate. It is still not clear whether the cause is due to numerical differentiation and more research is necessary to determine the exact contributions to this phenomenon.

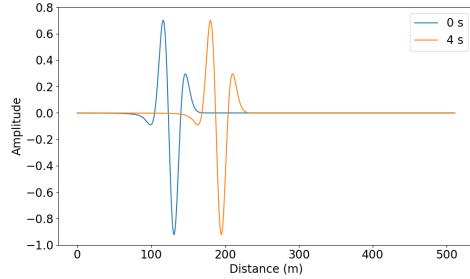


Figure 5.20: first and last training curves for the 1D wave model.

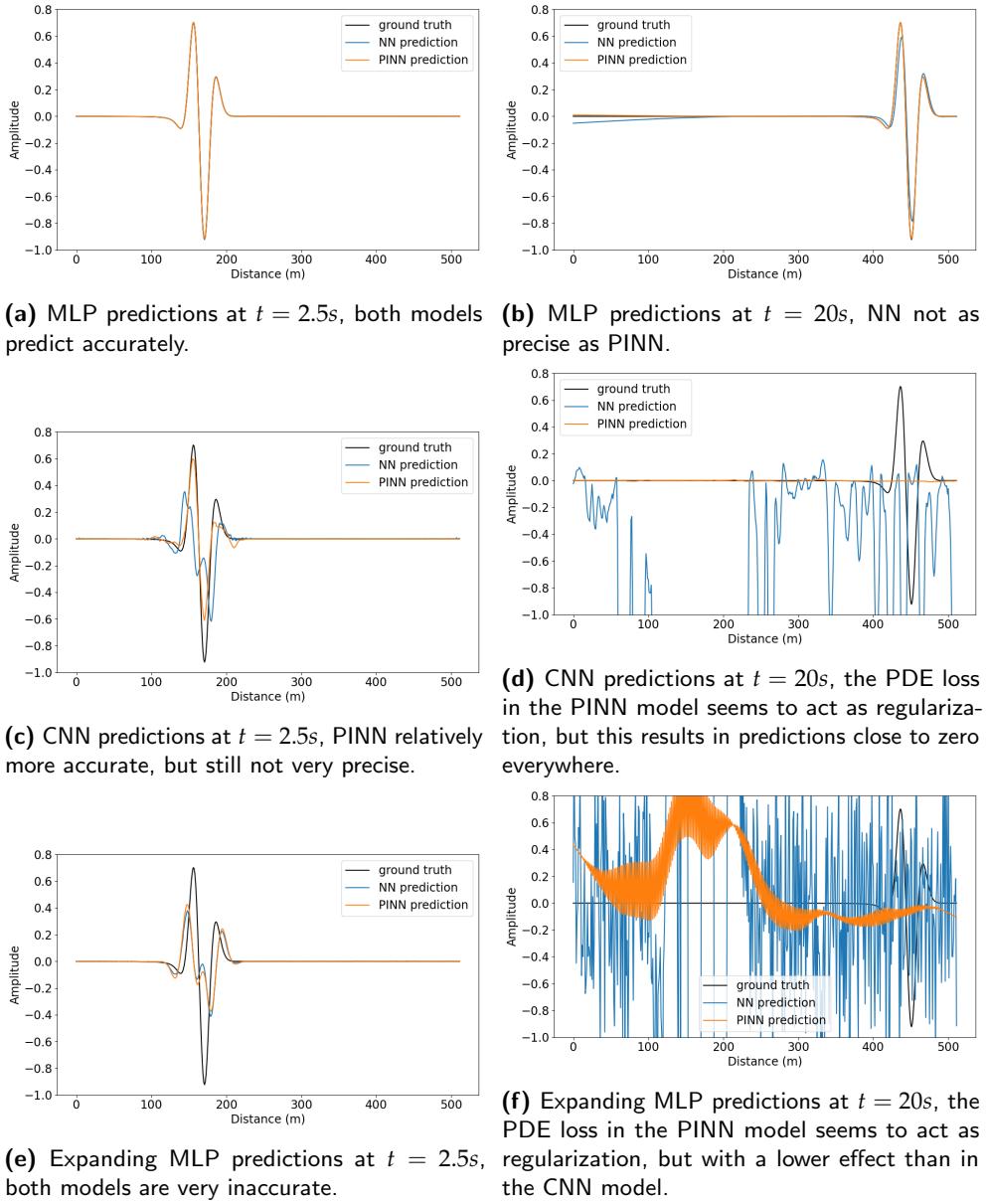


Figure 5.21: Predictions of MLP, CNN and expanding MLP models at $t = 2.5\text{s}$ (left, sparse reconstruction) and $t = 20\text{s}$ (right, domain extension).

Chapter 6

Conclusions

6.1 Discussion

Railway infrastructure requires regular surveys and timely maintenance to keep a high quality of ballast and sub-ballast layers and guarantee the safe and efficient transportation of people and cargo trains. The use of GPR technology for the estimation of ballast fouling and groundwater content requires large-scale labelled datasets. These are not available due to the high cost in time and money necessary for excavation. Computer simulated datasets can be generated, but still require substantial computational resources.

This work implements python scripts that enable the automatic generation of diverse and reasonably realistic GPR datasets by employing the open-source library gprMax. These tools are used to create two railway track GPR datasets, which are based on information provided by our contacts at SBB and are deemed to express realistic configurations of railway tracks. These datasets have some limitations, especially because, to limit computational cost, they only consider a 2D representation of the railway track geometry. This is clearly not the most realistic depiction of the real world and it does not account for rails and an antenna model.

One dataset, comprised of B-scan responses, is used to train a black box encoder-decoder CNN architecture. This approach shows very promising results, with B-scan predictions in most cases very close to the FDTD-simulated ones. The main discrepancies are identified to occur in samples where groundwater appears between the layers, in which the water reflection is sometimes not captured by the network. The computational cost at inference time of the proposed model is ≈ 300000 times lower than the one required by gprMax when using batched predictions. As a result, an overall speedup of the dataset generation process of ≈ 300 times is achieved.

The A-scan dataset is used in the investigation of PINN models. Two different

architectures are examined: an MLP and a CNN. The MLP-based models show a remarkable success in both sparse reconstruction and domain extension capabilities: on uniform and two layer geometries it fits the test data with high precision. Unfortunately, this first architecture shows some limits on its ability to capture a full wavefield generated by a rail track sample. This might be caused by the numerous reflections that arise in this case, and it is speculated that the spectral bias of neural networks [76] could be involved. On the contrary, the CNN model fits the data in the training set well, but does not exhibit remarkable interpolation or extrapolation properties. We conduct a further experiment on a simpler 1D wave to investigate this phenomenon, showing that it is not caused by the railway geometry or by the 2D nature of our experiment. Here, a CNN and an MLP-based architecture are investigated: both output a sequence corresponding to a discretized version of the field. This last analysis shows that in our experiments the enforcement of the wave equation on these models only acts as a regularization. It is not sufficient to expand the accuracy of the model predictions from the observation to the collocation domain. We speculate that numerical differentiation might be the cause of instabilities in this case, or that other factors caused by the discretization of the output domain might hinder the training process of these networks.

6.2 Future work

The natural continuation of this work involves the application of the trained black box model shown in Section 5.1 to the dataset creation task. This will enable the generation of large-scale datasets that will facilitate the creation of subsequent monitoring methods aimed at supporting the maintenance of railway infrastructure.

As briefly mentioned in the previous section, due to limitations in both the dataset and ML model, a real-world implementation of monitoring methods based on this work should consider the following factors:

- An even more realistic dataset modelling might be required to capture the complexity of the real world: the samples generated in the datasets only encompass limited variation. in the condition of the railway tracks. They do not take into consideration the presence of external objects such as pipes, signals, potholes, tunnels and other trains, which would highly affect the response of GPR equipment. Furthermore, the dataset only accounts for a 2D representation, which is missing an antenna model, rail and train carriage reflections. In this regard, we believe that the rail response could be added as a post-processing step with a limited additional complexity.
- The black box model predictions are not always accurate: the ground-

6.2. Future work

water response, which constitutes a relevant part in the motivation for this work, is sometimes not well captured by the network. To this extent, increasing the frequency of samples containing this feature in the training set might help alleviate this problem and produce more precise results.

Regarding the work conducted on physics-informed neural networks, multiple research directions arise following this work. In particular, some directions we believe are worth investigating are:

- The application of the boundary condition at the interface between layers seems to have no effect on the network predictions. This result might be caused by the application of the condition in 2 dimensions, leading to the omission of one of the constraining equations in our training procedure. More research is needed to investigate the causes of this phenomenon or whether a 3D version would result in different behaviour. A further motivation resides in the fact that the application of free-surface boundary conditions to seismic data results in very accurate reflected waves [57], so a comparison study could be conducted.
- More research should be conducted on the application of PINNs to problems with complex geometries, in particular for modelling EM waves.
- The application of PINNs to architectures with a discretized output domain seems in our experiments to only act as a mild regularization. It is not sufficient to obtain an accurate prediction inside the collocation domain. More research is needed to pinpoint the causes of this problem and to extend the framework of physics-informed NNs to these architectures.

Bibliography

- [1] F. O. o. T. FOT. [Online]. Available: <https://www.bav.admin.ch/bav/en/home/modes-of-transport/railways.html>.
- [2] E. D. f. a. A. EDA. [Online]. Available: <https://www.eda.admin.ch/aboutswitzerland/en/home/wirtschaft/verkehr/verkehr-fakten-und-zahlen.html>.
- [3] S. Facts and Figures. [Online]. Available: <https://reporting.sbb.ch/en/transportation?highlighted=9762ae54dd1e145e6b9870d594d1ca32>.
- [4] A. M. Zarembski, G. T. Grissom, and T. L. Euston, "On the use of ballast inspection technology for the management of track substructure," *Transportation Infrastructure Geotechnology*, vol. 1, no. 1, pp. 83–109, Mar. 2014, ISSN: 2196-7210. DOI: [10.1007/s40515-014-0004-5](https://doi.org/10.1007/s40515-014-0004-5). [Online]. Available: <https://doi.org/10.1007/s40515-014-0004-5>.
- [5] S. Wang, G. Liu, G. Jing, Q. Feng, H. Liu, and Y. Guo, "State-of-the-art review of ground penetrating radar (gpr) applications for railway ballast inspection," *Sensors*, vol. 22, no. 7, 2022, ISSN: 1424-8220. DOI: [10.3390/s22072450](https://doi.org/10.3390/s22072450). [Online]. Available: <https://www.mdpi.com/1424-8220/22/7/2450>.
- [6] C. Charoenwong, D. Connolly, P. Alves Costa, P. Galvín, and A. Romero, "The effect of ballast moisture content and fouling index on railway track settlement," *Transportation Geotechnics*, vol. 45, p. 101193, 2024, ISSN: 2214-3912. DOI: <https://doi.org/10.1016/j.trgeo.2024.101193>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221439122400014X>.
- [7] L. Wang, M. Meguid, and H. S. Mitri, "Impact of ballast fouling on the mechanical properties of railway ballast: Insights from discrete element analysis," *Processes*, vol. 9, no. 8, p. 1331, 2021.

- [8] N. S. Kahil *et al.*, "Automatic analysis of railway ground penetrating radar: Using signal processing and machine learning approaches to assess railroad track substructure," *Transportation Research Procedia*, vol. 72, pp. 3008–3015, 2023.
- [9] C. Warren, A. Giannopoulos, and I. Giannakis, "Gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Computer Physics Communications*, vol. 209, pp. 163–170, 2016, ISSN: 0010-4655. doi: <https://doi.org/10.1016/j.cpc.2016.08.020>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465516302533>.
- [10] C. Warren, A. Giannopoulos, and I. Giannakis, *Gprmax*. [Online]. Available: <https://www.gprmax.com/>.
- [11] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, ISSN: 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2018.10.045>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- [12] A. Dow, *The Railway - British Track Since 1804*. Pen & Sword Transport, 2017.
- [13] C. Charoenwong *et al.*, "Railway slab vs ballasted track: A comparison of track geometry degradation," *Construction and Building Materials*, vol. 378, p. 131 121, 2023, ISSN: 0950-0618. doi: <https://doi.org/10.1016/j.conbuildmat.2023.131121>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950061823008334>.
- [14] railsystem.net, *Track structure*. [Online]. Available: <https://railsystem.net/track-structure/>.
- [15] G. ARCIERI, C. Hoelzl, O. SCHWERY, D. Straub, K. Papakonstantinou, and E. Chatzi, "A comparison of value-based and policy-based reinforcement learning for monitoring informed railway maintenance planning," Sep. 2023. doi: <10.12783/shm2023/37015>.
- [16] S. B. S. - N. Schneeberger, *Der Stein, auf dem die bahn rollt*. [Online]. Available: <https://news.sbb.ch/artikel/99725/der-stein-auf-dem-die-bahn-rollt>.
- [17] B. Lichtberger, *Handbuch Gleis*. Eurail Press, 2010.

- [18] C. Shi, Z. Fan, D. P. Connolly, G. Jing, V. Markine, and Y. Guo, "Railway ballast performance: Recent advances in the understanding of geometry, distribution and degradation," *Transportation Geotechnics*, vol. 41, p. 101042, 2023, issn: 2214-3912. doi: <https://doi.org/10.1016/j.trgeo.2023.101042>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214391223001150>.
- [19] J. M. Mack, "Untersuchungen zum schichtübergang unterbau-schotter unter berücksichtigung verschiedener oberbaukonfigurationen," Ph.D. dissertation, TECHNISCHE UNIVERSITÄT MÜNCHEN, 2021.
- [20] D. J. Daniels, *Ground Penetrating Radar*, 2nd edition. IET, 2017.
- [21] D. P. Hampshire, "A derivation of maxwell's equations using the heaviside notation," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2018. doi: <10.1098/rsta.2017.0447>.
- [22] Z. H. Mohammed, "The fresnel coefficient of thin film multilayer using transfer matrix method tmm," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 518, 2019, p. 032026.
- [23] A. Taflove and S. C. Hagness, *Computational electrodynamics - 3rd edition*. Artech House, 2005.
- [24] N. Peplinski, F. Ulaby, and M. Dobson, "Dielectric properties of soils in the 0.3-1.3-ghz range," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 3, pp. 803–807, 1995. doi: <10.1109/36.387598>.
- [25] N. Peplinski, F. Ulaby, and M. Dobson, "Corrections to "dielectric properties of soils in the 0.3-1.3-ghz range"," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 6, pp. 1340–, 1995. doi: <10.1109/TGRS.1995.477193>.
- [26] I. Giannakis, A. Giannopoulos, and C. Warren, "A realistic fdtd numerical modeling framework of ground penetrating radar for landmine detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 1, pp. 37–51, 2016. doi: <10.1109/JSTARS.2015.2468597>.
- [27] Z. Liu, X. Gu, W. Wu, X. Zou, Q. Dong, and L. Wang, "Gpr-based detection of internal cracks in asphalt pavement: A combination method of deepaugment data and object detection," *Measurement*, vol. 197, p. 111281, 2022.
- [28] P. Koyan and J. Tronicke, "3d modeling of ground-penetrating radar data across a realistic sedimentary model," *Computers & Geosciences*, vol. 137, p. 104422, 2020, issn: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2020.104422>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098300419305540>.

- [29] S.-E. Hamran *et al.*, "Radar imager for mars' subsurface Experiment—RIMFAX," en, *Space Sci. Rev.*, vol. 216, no. 8, Dec. 2020. doi: <https://doi.org/10.1007/s11214-020-00740-4>.
- [30] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through Physics-Informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Jul. 2022.
- [31] M. Haywood-Alexander, W. Liu, K. Bacsa, Z. Lai, and E. Chatzi, *Discussing the spectra of physics-enhanced machine learning via a survey on structural mechanics applications*, 2023. arXiv: [2310.20425 \[cs.LG\]](https://arxiv.org/abs/2310.20425).
- [32] J. Irving and R. Knight, "Numerical modeling of ground-penetrating radar in 2-d using matlab," *Computers & Geosciences*, vol. 32, no. 9, pp. 1247–1258, 2006, issn: 0098-3004. doi: <https://doi.org/10.1016/j.cageo.2005.11.006>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098300405002621>.
- [33] M. Islam, M. Afzal, M. Ahmad, and T. Tauqeer, "Simulation and modeling of ground penetrating radars," in *2012 International Conference on Emerging Technologies*, 2012, pp. 1–6. doi: [10.1109/ICET.2012.6375487](https://doi.org/10.1109/ICET.2012.6375487).
- [34] S. Zarei, B. Oskooi, N. Amini, and A. R. Dalkhani, "2d spectral element modeling of gpr wave propagation in inhomogeneous media," *Journal of Applied Geophysics*, vol. 133, pp. 92–97, 2016, issn: 0926-9851. doi: <https://doi.org/10.1016/j.jappgeo.2016.07.027>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0926985116302178>.
- [35] J. Deng, Y. Rogez, P. Zhu, A. Herique, J. Jiang, and W. Kofman, "3d time-domain electromagnetic full waveform inversion in debye dispersive medium accelerated by multi-gpu paralleling," *Computer Physics Communications*, vol. 265, p. 108 002, 2021, issn: 0010-4655. doi: <https://doi.org/10.1016/j.cpc.2021.108002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465521001144>.
- [36] I. Giannakis, A. Giannopoulos, and C. Warren, "A realistic fdtd numerical modeling framework of ground penetrating radar for landmine detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 1, pp. 37–51, 2016. doi: [10.1109/JSTARS.2015.2468597](https://doi.org/10.1109/JSTARS.2015.2468597).
- [37] I. Giannakis, A. Giannopoulos, C. Warren, and N. Davidson, "Numerical modeling and neural networks for landmine detection using ground penetrating radar," Jul. 2015. doi: [10.1109/IWAGPR.2015.7292682](https://doi.org/10.1109/IWAGPR.2015.7292682).

- [38] X. L. Travassos, S. L. Avila, and N. Ida, "Artificial neural networks and machine learning techniques applied to ground penetrating radar: A review," *Applied Computing and Informatics*, vol. 17, no. 2, pp. 296–308, Jan. 2021.
- [39] O. Patsia, A. Giannopoulos, and I. Giannakis, "A deep learning framework for ground penetrating radar," in *2021 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, 2021, pp. 1–6. doi: [10.1109/IWAGPR50767.2021.9843168](https://doi.org/10.1109/IWAGPR50767.2021.9843168).
- [40] Z. Tong, J. Gao, and D. Yuan, "Advances of deep learning applications in ground-penetrating radar: A survey," *Construction and Building Materials*, vol. 258, p. 120371, 2020, issn: 0950-0618. doi: <https://doi.org/10.1016/j.conbuildmat.2020.120371>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095006182032376X>.
- [41] F. Picetti, G. Testa, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro, "Convolutional autoencoder for landmine detection on gpr scans," in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, 2018, pp. 1–4. doi: [10.1109/TSP.2018.8441206](https://doi.org/10.1109/TSP.2018.8441206).
- [42] Z. X. Leong and T. Zhu, "Direct velocity inversion of ground penetrating radar data using gprnet," *Journal of Geophysical Research: Solid Earth*, vol. 126, Jun. 2021. doi: [10.1029/2020JB021047](https://doi.org/10.1029/2020JB021047).
- [43] Y. Ji *et al.*, "Deep neural network-based permittivity inversions for ground penetrating radar data," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8172–8183, 2021. doi: [10.1109/JSEN.2021.3050618](https://doi.org/10.1109/JSEN.2021.3050618).
- [44] L. Xie, Q. Zhao, C. Ma, B. Liao, and J. Huo, "Ü-Net: Deep-Learning Schemes for Ground Penetrating Radar Data Inversion," *Journal of Environmental and Engineering Geophysics*, vol. 25, no. 2, pp. 287–292, Jun. 2020, issn: 1083-1363. doi: [10.2113/JEEG19-074](https://doi.org/10.2113/JEEG19-074). eprint: <https://pubs.geoscienceworld.org/jeeg/article-pdf/25/2/287/5102568/i1083-1363-25-2-287.pdf>. [Online]. Available: <https://doi.org/10.2113/JEEG19-074>.
- [45] J. K. Alvarez and S. Kodagoda, "Application of deep learning image-to-image transformation networks to gpr radargrams for sub-surface imaging in infrastructure monitoring," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018, pp. 611–616. doi: [10.1109/ICIEA.2018.8397788](https://doi.org/10.1109/ICIEA.2018.8397788).
- [46] I. Giannakis, A. Giannopoulos, and C. Warren, "A machine learning approach for simulating ground penetrating radar," in *2018 17th International Conference on Ground Penetrating Radar (GPR)*, 2018, pp. 1–4. doi: [10.1109/ICGPR.2018.8441558](https://doi.org/10.1109/ICGPR.2018.8441558).

- [47] I. Giannakis, A. Giannopoulos, and C. Warren, "A machine learning-based fast-forward solver for ground penetrating radar with application to full-waveform inversion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4417–4426, 2019. doi: [10.1109/TGRS.2019.2891206](https://doi.org/10.1109/TGRS.2019.2891206).
- [48] Q. Dai *et al.*, "A deep learning-based gpr forward solver for predicting b-scans of subsurface objects," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, issn: 1558-0571. doi: [10.1109/lgrs.2022.3192003](https://doi.org/10.1109/lgrs.2022.3192003). [Online]. Available: <http://dx.doi.org/10.1109/LGRS.2022.3192003>.
- [49] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," en, *SIAM Rev. Soc. Ind. Appl. Math.*, vol. 63, no. 1, pp. 208–228, Jan. 2021.
- [50] F. Chen *et al.*, "Neurodiffeq: A python package for solving differential equations with neural networks," *Journal of Open Source Software*, vol. 5, no. 46, p. 1931, 2020.
- [51] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, <https://pytorch.org/>, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [52] O. Hennigh *et al.*, "Nvidia simnet™: An ai-accelerated multi-physics simulation framework," in *Computational Science – ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds., Cham: Springer International Publishing, 2021, pp. 447–461.
- [53] B. Moseley, A. Markham, and T. Nissen-Meyer, *Solving the wave equation with physics-informed deep learning*, 2020. arXiv: [2006.11894 \[physics.comp-ph\]](https://arxiv.org/abs/2006.11894).
- [54] D. Voytan and M. K. Sen, "Wave propagation with physics informed neural networks," in *SEG Technical Program Expanded Abstracts 2020*, Virtual: Society of Exploration Geophysicists, Sep. 2020.
- [55] C. Song, T. Alkhalifah, and U. Bin Waheed, "A versatile framework to solve the helmholtz equation using physics-informed neural networks," *Geophysical Journal International*, vol. 228, Oct. 2021. doi: [10.1093/gji/ggab434](https://doi.org/10.1093/gji/ggab434).
- [56] S. Alkhadhr, X. Liu, and M. Almekkawy, "Modeling of the forward wave propagation using physics-informed neural networks," in *2021 IEEE International Ultrasonics Symposium (IUS)*, 2021, pp. 1–4. doi: [10.1109/IUS52206.2021.9593574](https://doi.org/10.1109/IUS52206.2021.9593574).

- [57] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (pinns) for wave propagation and full waveform inversions," *Journal of Geophysical Research: Solid Earth*, vol. 127, no. 5, Apr. 2022, ISSN: 2169-9356. doi: [10.1029/2021JB023120](https://doi.org/10.1029/2021JB023120). [Online]. Available: <http://dx.doi.org/10.1029/2021JB023120>.
- [58] Y. Zhang, X. Zhu, and J. Gao, "Seismic inversion based on acoustic wave equations using physics-informed neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023. doi: [10.1109/TGRS.2023.3236973](https://doi.org/10.1109/TGRS.2023.3236973).
- [59] Y. Zheng and Y. Wang, "Ground-penetrating radar wavefield simulation via physics-informed neural network solver," *Geophysics*, vol. 88, no. 2, KS47–KS57, Feb. 2023, ISSN: 0016-8033. doi: [10.1190/geo2022-0293.1](https://doi.org/10.1190/geo2022-0293.1). eprint: <https://pubs.geoscienceworld.org/geophysics/article-pdf/88/2/KS47/5831077/geo-2022-0293.1.pdf>. [Online]. Available: <https://doi.org/10.1190/geo2022-0293.1>.
- [60] L. Heller, *Synthetic gpr data generation for automatic ballast condition assessment*, 2023.
- [61] E. T. Selig and J. M. Waters, *TRACK GEOTECHNOLOGY and SUBSTRUCTURE MANAGEMENT*. Thomas Telford Publishing, 1994. doi: [10.1680/tgasm.20139](https://doi.org/10.1680/tgasm.20139). eprint: <https://www.icevirtuallibrary.com/doi/pdf/10.1680/tgasm.20139>. [Online]. Available: <https://www.icevirtuallibrary.com/doi/abs/10.1680/tgasm.20139>.
- [62] T. Freitas, A. Guimarães, S. Roels, V. Freitas, and A. Cataldo, "Is the time-domain reflectometry (tdr) technique suitable for moisture content measurement in low-porosity building materials?" *Sustainability*, vol. 12, p. 7855, Sep. 2020. doi: [10.3390/su12197855](https://doi.org/10.3390/su12197855).
- [63] K. Meyer, E. Erdogmus, G. Morcous, and M. Naughtin, "Use of ground penetrating radar for accurate concrete thickness measurements," vol. 328, Sep. 2008, pp. 1–10, ISBN: 978-0-7844-1002-8. doi: [10.1061/\(ASCE\)1080-0832\(2008\)328:1\(1\)](https://doi.org/10.1061/(ASCE)1080-0832(2008)328:1(1)).
- [64] R. Olmi, M. Bini, A. Ignesti, and C. Riminesi, "Dielectric properties of wood from 2 to 3 ghz," *The Journal of microwave power and electromagnetic energy : a publication of the International Microwave Power Institute*, vol. 35, pp. 135–43, Feb. 2000. doi: [10.1080/08327823.2000.11688430](https://doi.org/10.1080/08327823.2000.11688430).
- [65] R. Kenneth M., C. Thomas H., and W. John, *An Introduction to Materials Science and Engineering*. Wiley, 1991.
- [66] S. W. Ellingson, *Electromagnetics*, Vol. 1. VT Publishing, 2018. doi: [10.21061/electromagnetics-vol-1](https://doi.org/10.21061/electromagnetics-vol-1).
- [67] V. Blomqvist, *Pymunk*, 2023. [Online]. Available: <https://pymunk.org>.

Bibliography

- [68] S. Benatti *et al.*, “Pychrono and gym-chrono: A deep reinforcement learning framework leveraging multibody dynamics to control autonomous vehicles and robots,” in Jan. 2022, pp. 573–584, ISBN: 978-3-030-81165-5. doi: [10.1007/978-3-030-81166-2_50](https://doi.org/10.1007/978-3-030-81166-2_50). [Online]. Available: <https://projectchrono.org/pychrono/>.
- [69] F. D.V., *Treatise on Geophysics (Second Edition)*. Elsevier, 2015.
- [70] F. Chollet *et al.*, *Keras*, <https://keras.io>, 2015.
- [71] Martín Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: <https://www.tensorflow.org/>.
- [72] E. Zürich, *Material boundaries*, Accessed: 01-04-2024.
- [73] U. of Toronto, *Reflection/refraction boundary conditions*, Accessed: 01-04-2024.
- [74] A. Wolski, *Theory of electromagnetic fields*, Accessed: 01-04-2024.
- [75] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., ser. Proceedings of Machine Learning Research, vol. 9, Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. [Online]. Available: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [76] N. Rahaman *et al.*, *On the spectral bias of neural networks*, 2019. arXiv: [1806.08734 \[stat.ML\]](https://arxiv.org/abs/1806.08734).
- [77] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the sobel operator,” *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies¹.
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies².
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies³. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Railway infrastructure GPR dataset creation and modeling via black box and physics-informed neural networks

Authored by:

If the work was compiled in a group, the names of all authors are required.

Last name(s):

Rigoni

First name(s):

Thomas

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date

Zürich, 14/04/2024

Signature(s)

<i>Thomas Rigoni</i>

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

¹ E.g. ChatGPT, DALL E 2, Google Bard

² E.g. ChatGPT, DALL E 2, Google Bard

³ E.g. ChatGPT, DALL E 2, Google Bard