

BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Thesis

**PHYSICS-INFORMED NEURAL NETWORKS FOR MAGNETIC RESONANCE
ELECTRICAL PROPERTY TOMOGRAPHY AND PHONON BOLTZMANN
TRANSPORT**

by

Wiktor Lidwin

Submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Computer Engineering

2025

© 2025 by
Wiktor Lidwin
All rights reserved

Approved by

First Reader

Luca Dal Negro, Ph.D.
Professor of Electrical & Computer Engineering
Professor of Materials Science & Engineering
Professor of Physics

ACKNOWLEDGMENTS

I would like to express my thanks to my advisor, Professor Luca Dal Negro, for his support and guidance throughout this project. Special thanks to Roberto Riganti for introducing me to PINN, technical help with code development, and for many helpful discussions. Also, I would like to thank Dr. Ilias I. Giannakopoulos and Prof. Riccardo Lattanzi from New York University Department of Radiology for help with software validation and for the insightful discussions.

Wiktor Lidwin

Spring 2025

Boston, MA

Wiktor Lidwin

Boston University, College of Engineering, 2025

Advisor: Luca Dal Negro, Professor ECE, MSE, Physics

ABSTRACT

Physics-Informed Neural Networks (PINNs) have emerged as a powerful tool for solving forward and inverse problems governed by partial differential equations (PDEs). This work advances two state-of-the-art applications of PINNs in biomedical imaging and thermal transport.

First, we improve the Physics-Informed Fourier Networks (PIFON) framework, developed by collaborators at the NYU Department of Radiology for Magnetic Resonance Electrical Properties Tomography (MR-EPT), by introducing two novel architectures: EigenFunction-PINN (EF-PINN) and Multiscale Dropout-PINN (MS-DO-PINN), along with a Modified Mean Square Error loss function. These innovations address a critical limitation of classical PINNs, where high errors near material interfaces are smeared across neighboring regions, leading to poor reconstructions. Our approach significantly enhances reconstruction accuracy, reducing the error from $\leq 10\%$ (baseline PIFON) to as low as $\leq 0.01\%$. These improvements generalize well to 3D and realistic 2D scenarios, achieving $\leq 0.1\%$ error even under noise.

Second, we refine a PINN-based solver for the mode-resolved phonon Boltzmann Transport Equation (BTE) by developing a Hard Constraint PINN (HC-PINN) that enforces boundary conditions exactly. With additional optimization techniques, we reduce training time from 4 hours to under 10 minutes, enabling efficient simulation of thermal transport across ballistic to diffusive regimes.

These contributions not only advance the state of the art in PINN-based modeling but also highlight the power of tailored architectures and training strategies in overcoming long-standing challenges such as interface reconstruction and computational inefficiency. We demonstrate PINN’s capabilities by enabling precise, scalable solutions in biomedical imaging and nanoscale heat transport.

TABLE OF CONTENTS	
ACKNOWLEDGMENTS	3
ABSTRACT	4
Introduction	8
1.1 Physics-informed Neural Networks in Scientific Computing	8
1.1.1 Current Uses	10
1.1.2 Comparison to Data-Driven Approaches	11
1.1.3 Best Practices	15
MR-EPT Study	18
2.1 Motivation	18
2.1.1 Proposed PINN Solution	20
2.2 PINN Objectives	21
2.3 General Techniques	23
2.3.1 Residual Selection	23
2.3.2 Loss Scaling	23
2.3.3 Number of Collocation Points	24
2.3.4 β -Swish	25
2.3.5 Weight Initialization Scheme	26
2.4 High-Precision Techniques	27
2.4.1 Ideal Focusing Effect	28
2.4.2 Modified MSE	29
2.4.3 Collocation Points Scheme	32
2.4.4 Dropout	33
2.5 Frequency Tuning	34
2.5.1 Multiscale PINN	34
2.5.2 Eigenfunction Decomposition and Random Fourier Features	36
2.6 Architecture Overview	38
2.6.1 Training and Testing	41
2.7 Results	41
2.7.1 2D Cylinder under a plane wave	41
2.7.2 Comparison to the original PIFON	44
2.7.3 Comparison between MSE and modified MSE	45
2.7.4 Realistic tissue EPs	47
2.7.5 3D Sphere under a plane wave	52
2.8 Summary	57
BTE Study	58
3.1 Motivation	58
3.1.1 Proposed PINN Solution	60
3.2 PINN Objectives	61

3.3 Efficiency Techniques	62
3.3.1 Loss Scaling	62
3.3.2 Network Initialization	64
3.3.3 Hard boundary constraints	66
3.4 Architecture Overview	69
3.5 Results - 2D rectangle phonon transport	70
3.5.1 Comparison to Li et al.'s Work	70
3.5.2 Length Scale Sweep Analysis	73
General Conclusions	74
4.1 Future work	76
Appendix	78
5.1. Helmholtz in MRI	78
5.2 Simplified Helmholtz	79
5.3 Convection-Reaction EPT	79
5.4 Phonon Boltzmann Transport Equation	80
BIBLIOGRAPHY	82

Introduction

1.1 Physics-informed Neural Networks in Scientific Computing

Ongoing progress in modern machine learning, fueled by increases in computational resources and large-scale datasets, have led to transformative breakthroughs across a wide array of scientific and engineering disciplines. Despite this progress, a fundamental challenge persists in the application of machine learning to modeling complex physical, biological, and engineering systems: the scarcity of high-quality data. In many real-world scenarios, acquiring sufficient training data is expensive, invasive, or even physically infeasible. Furthermore, the majority of state-of-the-art machine learning models are unreliable in these small-data regimes, exhibiting poor generalization and robustness.

To overcome the limitations of data scarcity, machine learning models must use prior knowledge about the engineering system. Instead of relying purely on data, these models can be constrained using known mathematical structures, such as conservation laws, symmetries, or partial differential equations (PDEs). These principles are utilized to regularize the learning process, enabling models to generalize from sparse observations. Specifically, Physics-informed neural networks (PINNs) represent an emerging framework in natural science and engineering to accurately solve complex differential equation models both in a forward and inverse modality[1-8]. Building on the universal approximation theorem for multilayer neural networks, in PINNs, we use the entire

network as a surrogate solution for PDE models. These PINNs incorporate known physical laws by embedding the governing equations directly into the loss function.

Training a PINN involves minimizing the residuals of the PDE, along with errors in the initial and boundary conditions, over a set of randomly distributed collocation points that span the solution domain. These residuals are evaluated using automatic differentiation, which allows for the analytical computation of derivatives with respect to network inputs, avoiding the numerical inaccuracies associated with numerical methods like the finite element method(FEM). By enforcing the PDE, along with the initial and boundary conditions, throughout the domain, PINNs generate a unique solution that satisfies both the data and the physical laws. Consequently, any modification to these conditions necessitates retraining the network to obtain a new solution.

One of the most powerful features of PINNs is their ability to solve inverse problems, where unknown parameters or fields, such as material properties, source terms, boundary conditions, or initial conditions, must be inferred from partial or indirect observations. Unlike traditional approaches that often require separate forward and inverse solvers with iterative optimization loops, PINNs incorporate these unknowns directly into the neural network as trainable variables. This enables a unified framework in which the forward and inverse components are solved simultaneously. Such an approach is particularly advantageous for ill-posed problems, where traditional numerical methods are computationally intensive and sensitive to noisy or incomplete data.

1.1.1 Current Uses

The original formulation of PINNs, introduced by Raissi et al. (2019)[1], has since inspired a wide array of extensions and applications. Researchers have applied PINNs to forward and inverse problems in photonics, fluid dynamics, and heat transfer[1-10].

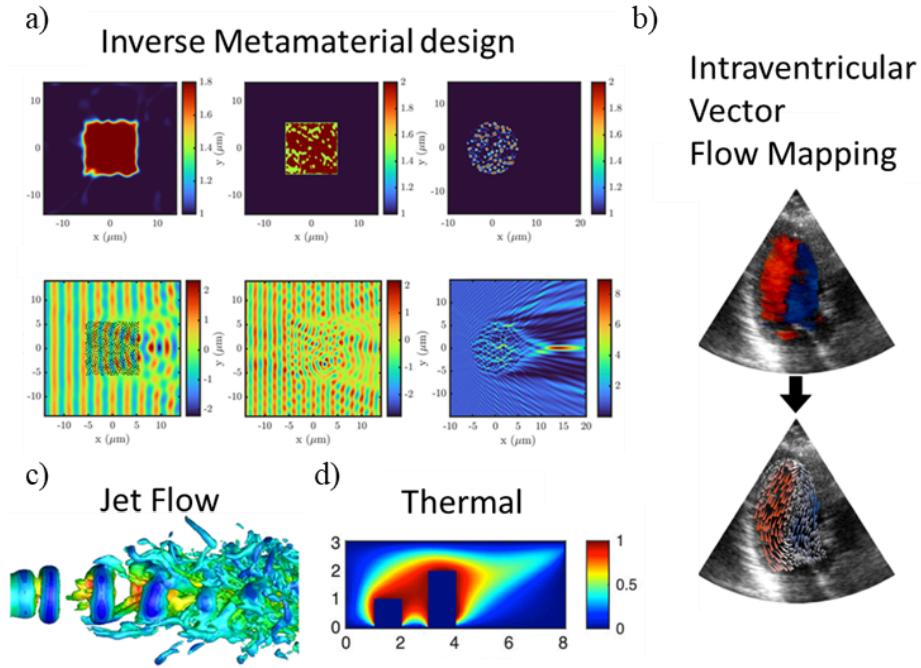


Figure 1.1: (a) PINN for inverse metamaterial design[8], (b) Intraventricular Vector Flow Mapping[13], (c) Jet Flow[11], and (d) Thermal Transport[12]

Recent advances in the application of PINNs highlight their growing ability to model increasingly complex physical systems. For example, Figure 1.1 demonstrates the parameter-retrieval capabilities of PINN for inverse metamaterial design, where fine-scale structures are identified to reproduce desired macroscopic field behavior. Additionally, Figure 1.1 presents a PINN framework for forward heat transfer problem, as developed by Cai et al. (2021) [12]. In this case, the model learns the temperature distribution within a domain by solving the heat equation under specified boundary and

initial conditions. As the field continues to grow, integrating PINNs with advanced architectures and efficient solvers promises to further expand their impact across disciplines.

1.1.2 Comparison to Data-Driven Approaches

It is important to examine in detail the fundamental differences between physics-driven and data-driven machine learning approaches. This comparison helps clarify why conventional data-driven neural networks, though powerful in many applications, often fall short in scientific and engineering problems governed by physical laws. Understanding these distinctions also sheds light on the respective advantages and limitations of each method. To illustrate the differences, a high-level overview of the data flow for both approaches for Electrical Property Tomography (EPT) is illustrated in Figure 1.2.

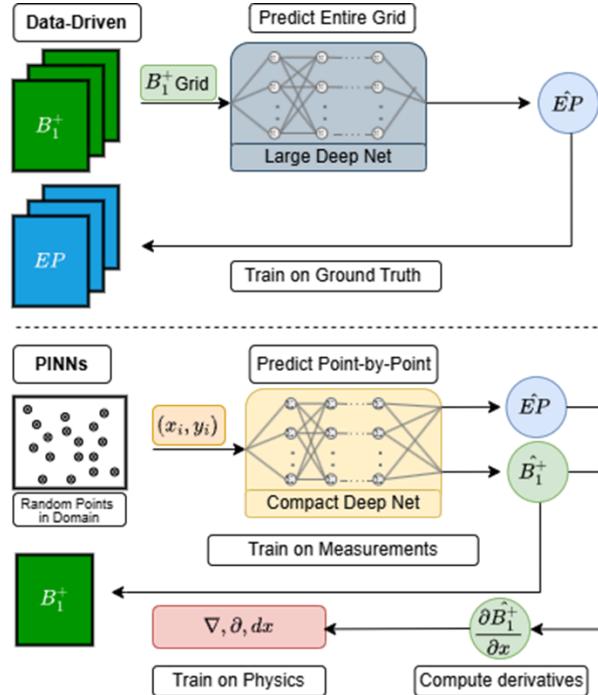


Figure 1.2: A high-level comparison between data-driven and PINNs for EPT

Suppose we aim to predict a 2D spatial grid of some Electrical Properties(EP)

given Magnetic Field (B_1^+) measurements on the same grid. In the data-driven approach, we assume access to a large, labeled dataset containing multiple unique entries of different EP distributions and their corresponding B_1^+ measurements, each sampled over an $M \times M$ grid. Traditional neural networks process the entire B_1^+ grid as input, predict the corresponding EP grid, and compute the loss by comparing the prediction to the ground truth. The model parameters are then updated via backpropagation, relying entirely on the availability and diversity of training data.

These models perform best when large, high-quality datasets are available and are often intricate and computationally heavy, requiring days to train. However, once trained, they offer extremely fast inference by leveraging prior knowledge from the entire dataset. The key advantages include rapid predictions and strong performance on familiar

distributions. The downsides are the reliance on high-quality labeled data, fixed resolution output, no guarantee of generalization to unseen cases, and long initial training times. Finally, all data-driven models, including popular LLMs like ChatGPT, are susceptible to a phenomenon called "hallucination," where the network produces outputs that appear plausible but are factually incorrect or physically invalid[13]. In the context of EP prediction, hallucination could lead the model to generate mappings that violate basic physical laws, due to the absence of built-in mechanisms enforcing physical correctness. Despite these challenges, data-driven neural networks remain valuable as fast and efficient initial approximators.

In contrast, PINNs need to retain the neural network parameters for each EPT problem, eliminating the need for large datasets. Rather than inputting a spatial grid of B_1^+ measurements into the model, this model approximates the function B_1^+ with a deep neural network and takes spatial coordinates (x,y) as inputs. The training data is interpreted as the initial or boundary conditions of the governing PDE and consist of a collection of N spatial points, each associated with a corresponding measured B_1^+ value. The model then predicts both B_1^+ and EP at each spatial coordinate. Critically, these outputs are not learned in isolation; they are constrained by a governing physics equation that couples B_1^+ and EP, typically involving spatial gradients. This physical constraint is enforced by introducing a separate set of training points, known as *collocation points*, where the residual of the PDE is evaluated. By training on these points using traditional gradient-based methods, the network incrementally reduces the error residuals of the governing PDEs and converges toward a valid solution across the entire domain. While

traditional data-driven neural networks typically rely on a single loss term to measure the mismatch between predicted and observed data, PINNs incorporate multiple loss components. These include terms that enforce any constraints, which are the B_1^+ measurements for EPT, as well as the residuals of the governing PDE.

In general, PINN architectures are compact and employ smooth, differentiable activation functions to facilitate gradient-based optimization via automatic differentiation. The advantages of PINNs include strong noise robustness, a built-in mechanism for quantifying and minimizing physical inconsistencies, the flexibility to evaluate solutions at arbitrary spatial locations, and access to exact analytical derivatives of the learned fields. However, several challenges remain: the need to retrain the model for every new problem or initialization, difficulty in cases where the governing equations are unknown or poorly defined, and nontrivial training dynamics that can lead to convergence issues. The last issue will be addressed in subsequent sections of this thesis, with the goal of mitigating it through architectural and training improvements.

In summary, data-driven neural networks excel at fast inference once trained but require large, high-quality datasets and do not guarantee physical validity. In contrast, physics-driven approaches like PINNs enforce physical laws directly, enabling robust solutions from sparse data, but often at the cost of increased training complexity and time.

1.1.3 Best Practices

When training a PINN, there are a few key general points to keep in mind. These guiding principles will be referenced throughout the remainder of this paper and will

motivate the methods used for loss balancing, network precision, and the avoidance of local minima. This overview is brief, as the detailed techniques for fine-tuning each principle will be explained later.

Ultimately, a PINN model is trained by minimizing an objective function. Regardless of how well-tuned the network architecture is, a poorly defined loss function will inevitably lead to poor results. There are several ways a loss function can be poorly defined, and all of these can hinder the training and performance of a PINN. A common and critical mistake is not balancing the various loss terms. Ideally, each loss component should contribute equally to the overall optimization process. If a particular term is too small, its influence on the learning process becomes negligible, effectively causing it to be ignored. Conversely, if one term dominates, it can overshadow the others, preventing the network from satisfying essential constraints. Therefore, balancing these loss terms is crucial to ensure that the model learns a physically meaningful and well-constrained solution.

Another common mistake is overlooking the fact that neural networks minimize loss functions via their gradients, not the loss values themselves. This distinction is crucial: if the gradients are poorly scaled or unstable, the network will struggle to learn effectively, regardless of the loss values. Ensuring that the loss gradients are well-behaved is therefore essential for efficient training. The mean squared error (MSE) is a popular choice due to its stability and smooth optimization landscape, making it a strong starting point in many cases. However, in the context of PINNs, the MSE has

limitations. Specifically, when the PDE residuals involve large derivative discontinuities, the resulting loss gradients can become erratic, leading to stalling in training behavior, a challenge that will be explored in more detail later.

Next, stochastic gradient descent (SGD) is most effective when fine-tuning small values, which is why a standard learning rate is typically around 10^{-3} rather than 10^3 . Choosing values that are either too large or too small for the inputs, outputs, or trainable parameters can lead to a rough or sparse optimization landscape. To illustrate, using a meter stick to measure the circumference of the Earth would not be practical, even with 2000 years and 5000 helpers. This analogy emphasizes the importance of selecting the right tools for the job, and not just increasing the number of epochs or neurons. Furthermore, in neural network training, most of the learning occurs early on, with the later stages focused on fine-tuning toward convergence. Given this, it is crucial to initialize the network as close as possible to the final solution, particularly in the case of PINNs, where the loss function involves higher-order derivatives. When training with nth-order derivatives, the network must also compute $(n+1)$ th-order derivatives for backpropagation. Since multiple derivative orders are involved in backpropagation, large deviations in any of these derivatives can cause training to stall or even result in instability. Generally, the higher the order of derivative calculation, the greater the likelihood of encountering "derivative spikes," where the derivative becomes unpredictably large or small. Proper initialization allows the PINN to focus on learning the underlying physics rather than wasting time navigating a distant solution space, while also reducing the risk of these spikes.

Often, PINNs are capable of modeling highly complex functions using relatively compact architectures. However, when a PINN struggles to approach the ideal solution, many practitioners mistakenly resort to adding more data or increasing the network size instead of investigating where the network is struggling. Understanding and applying proper initialization strategies is essential and will recur throughout our discussion, including in the context of boundary conditions, activation functions, weight initializations, and more.

In summary, creating a successful PINN framework relies on two key principles: defining a correct loss landscape and ensuring proper network initialization. These ideas will be explored in detail through the solution of both an inverse (EPT) and a forward (phonon transport) PDE.

MR-EPT Study

2.1 Motivation

Magnetic resonance electrical property tomography (MR-EPT) is a noninvasive imaging technique that utilizes clinical magnetic resonance imaging (MRI) systems to map the spatial distribution of the electrical properties(EP) of biological tissues[14,15]. These EP, namely relative permittivity and electric conductivity, determine the perturbation of electromagnetic waves through the body and are used as biomarkers for pathologies such as cancer[16-19]. Enhancing the resolution of the reconstructed EP could be used to improve the effectiveness of early cancer detection and therapeutic modalities such as radiofrequency hyperthermia[20-22].

MR-EPT reconstructs tissue EPs by solving the differential form of Maxwell's equations using the radiofrequency (RF) magnetic field data acquired during MRI scans[23]. This RF field, referred to as the B_1 field, is typically decomposed into its circularly polarized components: B_1^+ (left-handed) and B_1^- (right-handed). In current MRI implementations, only the magnitude of the B_1^+ field is directly measured, along with the transceive phase, which represents the average phase of B_1^+ and B_1^- . To make the problem tractable, a common assumption known as the “transceive phase approximation” is employed, wherein the phases of B_1^+ and B_1^- are presumed to be equal[24]. Under this assumption, the phase of B_1^+ can be approximated as half the transceive phase, allowing for the isolation of the real and imaginary components of the B_1^+ field. Conventional MR-EPT methods often reconstruct the EP distribution with

artifacts, primarily due to numerical errors introduced by numerical differentiation, which are further exacerbated by the presence of noise in the MRI measurements[45].

Recent advancements in deep learning (DL) have demonstrated EP reconstruction by learning from large-scale datasets[26-29]. These DL-based methods have been introduced to address challenges of noisy data and the high computational cost associated with conventional reconstruction techniques. Data-driven approaches typically treat MR measurements as 2D or 3D grids and train convolutional neural networks to approximate the EP. While these supervised learning models have shown promising results in simulation environments, their performance *in vivo* remains limited due to the scarcity of diverse and accurate training datasets. The lack of ground-truth EP maps from real tissues in clinical settings further exacerbates this issue, hindering generalizability and often resulting in predictions that deviate from the underlying physical laws of MR-EPT.

This paper builds upon the Physics-Informed Fourier Networks Electrical Properties Tomography (PIFON-EPT) framework[30]. In this MR-EPT PINN solver, X. Yu et al. utilized a dual PINN approach, modeling the complex B_1^+ and complex EP directly. These two networks are trained simultaneously in a physics-constrained manner, where the training process is guided by MRI measurements and the governing Helmholtz PDE that describes the relationship between B_1^+ and EP. By incorporating Random Fourier Features (RFF), the PIFON-EPT overcomes spectral bias and captures high-frequency variations in EP and B_1^+ [31-37]. While this framework achieved

impressive reconstruction performance on noisy data, it continued to face challenges in accurately resolving the boundaries between regions with differing electrical properties.

2.1.1 Proposed PINN Solution

To address the challenges inherent in previous methods, we reinterpreted the RFF layer as an eigenfunction(EF) decomposition. This reinterpretation provides a clearer understanding of RFF hyperparameter tuning and demonstrates that RFF is, in fact, a subset of the EFs of the Helmholtz equation. From here, we identify additional modes in the EF decomposition. By leveraging this insight, we decide to use these modes as a first layer pre-processing step, providing a functional basis for our deep neural network to learn from. The utilization of the eigen modes in our network architecture is termed EigenFunction-PINN (EF-PINN). This EF layer serves as a powerful basis for the B Net, enhancing its ability to model complex phenomena with higher accuracy and efficiency. To achieve high precision EP reconstruction, a Multiscale-PINN architecture[8] is used, enhanced with dropout layers to prevent overfitting and improve scale utilization. This novel approach is referred to as Multiscale-Dropout Out-PINN (MS-DO-PINN) and is utilized by the EP network. Furthermore, the standard mean squared error (MSE) loss function was modified to resolve reconstruction errors at EP interfaces. In addition, residual selection, loss scaling, β -Swish activations, and weight initialization methods are introduced to further enhance the accuracy and robustness of the reconstruction process.

We compare our method to the original PIFON using a 2D cylindrical phantom with plane-wave excitation as a baseline. The PIFON framework, using the Simplified

Helmholtz, produces EP reconstruction errors of $\leq 10\%$; our models produce errors of $\leq .01\%$. Building on this, we apply our approach to a grey matter and white matter 2D cylindrical phantom excited by a wire source, demonstrating the network's adaptability to proper MRI systems. Our models produced reconstructions errors of $\leq .1\%$ using B_1^+ data with a signal-to-noise (SNR) of 200. Finally, we extend our validation to a 3D spherical phantom under plane-wave excitation and reach $\leq .1\%$ reconstruction errors. The joint EF-PINN and MS-DO-PINN architectures combined with a modified MSE enable high-resolution reconstructions from noisy data, both in 2D and 3D. This work enhances the capabilities of PIFON and lays the groundwork for further advancements in MR-EPT.

2.2 PINN Objectives

The goal of the MR-EPT PINN is to reconstruct the EP inside a domain Ω , given the noisy complex B_1^+ measurements at N spatial points. The neural networks B and EP predict the complex values of B and EP and are referred to as B Net and EP Net, respectively. In general, reconstruction will be done by minimizing the following loss function:

$$\mathcal{L}(\tilde{\theta}_1, \tilde{\theta}_2) = w_{data}\mathcal{L}_{data}(\tilde{\theta}_1) + w_r\mathcal{L}_r(\tilde{\theta}_1, \tilde{\theta}_2) \quad (2.1)$$

Where L is the total loss we want to minimize, L_{data} is the MSE between the measured B_1^+ and the predicted \hat{B}_1^+ , L_r is the MSE of the PDE residual, θ_1 is the learnable parameters of B Net, and θ_2 is the learnable parameters of EP Net.

$$\mathcal{L}_{data} = \frac{1}{N_d} \sum_i^{N_d} |B(\tilde{\theta}_1; \mathbf{r}_i) - \mathbf{B}_1^+(\mathbf{r}_i)|^2 \quad (2.2)$$

$$\mathcal{L}_r = \frac{1}{N_c} \sum_i^{N_c} |\mathcal{R}\{B(\tilde{\theta}_1; \mathbf{r}_i); EP(\tilde{\theta}_2; \mathbf{r}_i)\}|^2 \quad (2.3)$$

Where N_d is the number of data points and N_c is the number of collocation points. L_r is computed on the collocation points, which can be randomly or uniformly distributed across the domain. Since EP and B are complex, each loss term is divided into its real and imaginary components.

Using the Eq. 5.4 and Eq. 5.5 provided in the Appendix, we have the following two options for the physics residual:

Assuming a smooth EP distribution, the Simplified Helmholtz residual is :

$$R_H = \nabla^2 B + k_0^2 \varepsilon B = 0 \quad (2.4)$$

Assuming B_z is approximately 0, the Generalized Helmholtz residual is :

$$R_{GH} = \nabla^2 B + k_0^2 \varepsilon B - \frac{1}{\varepsilon} \left(\frac{\partial B}{\partial x} - i \frac{\partial B}{\partial y} \right) \left(\frac{\partial \varepsilon}{\partial x} + i \frac{\partial \varepsilon}{\partial y} \right) - \frac{1}{\varepsilon} \left(\frac{\partial B}{\partial z} \cdot \frac{\partial \varepsilon}{\partial z} \right) = 0 \quad (2.5)$$

Both RH and RGH should equal 0 to satisfy the Helmholtz under their respective set of assumptions.

2.3 General Techniques

2.3.1 Residual Selection

The choice between different forms of the Helmholtz equation, simplified or generalized, depended on the size of the domain and the wavelength of interest. The Simplified Helmholtz worked best for large domains with multiple scattered wavelengths, where wave interactions are more oscillatory and spread out. On the other hand, the Generalized Helmholtz was more effective for small domains like those found in MRI-based scattering problems, where the wavelength is much larger than the domain size. In these cases, the extra terms in the generalized equation help capture small variations in material properties that would otherwise be missed.

2.3.2 Loss Scaling

A critical component of PINN training is the proper balancing of loss terms. In general, each loss component should contribute equally to the overall loss, which can be achieved by ensuring that their orders of magnitude are of the same order: $O(L_i) \sim O(1)$. This balance is critical for stable and efficient optimization. To achieve this, the magnetic field data is normalized such that its maximum magnitude is 1, resulting in the corresponding data loss term $O(L_d) \sim O(1)$. For the Helmholtz residual loss, the expected order of magnitude is estimated by substituting typical values into the PDE, allowing consistent scaling across all components of the total loss.

$$R_H = \nabla^2 B + k_0^2 \varepsilon B = 0, \quad O(\nabla^2 B) \sim O(k_0^2 \varepsilon B) \quad (2.6)$$

$$O(k_0^2 \varepsilon B) \sim O(k_0^2 \varepsilon) * O(B) \sim k_0^2 E[\varepsilon]$$

$$Assume : \quad O(\nabla^2 B) >= O\left(\frac{\nabla \varepsilon \times \nabla \times B}{\varepsilon}\right), \quad O(R_{GH}) \sim O(R_H)$$

We define $c := k_0^2 E[|\varepsilon|]$. Where $E[|\varepsilon|]$ is the expected value of the magnitude of the ground truth complex EP. The expected value is an approximation of the order of magnitude of $|\varepsilon|$. In practice, this expected value is treated as a constant and can be approximated using the mean, minimum, or maximum of the EP magnitude, depending on the specific characteristics of the profile. In scenarios where the EP exhibits high-magnitude outliers, using the maximum value provides a more stable estimate.

2.3.3 Number of Collocation Points

We aim for a collocation point resolution that provides coverage for both the data points and the spaces between them. This leads to the following heuristic:

$$N_c = (2N_d)^d \quad (2.7)$$

where N_c is the number of collocation points, N_d is the number of data points, and d is the dimension of the data(e.g., $d=2$ for 2D domains).

2.3.4 β -Swish

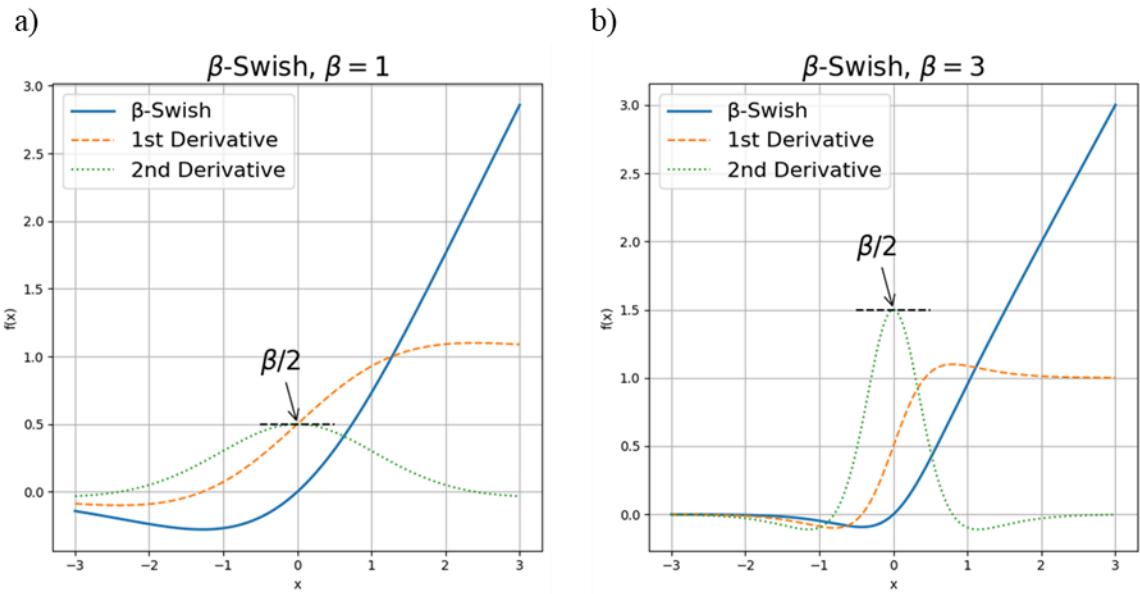


Figure 2.1. β -Swish activation with different β values. When $\beta=1$, β -Swish simplifies to normal Swish. Higher β parameters cause a sharper resemblance to ReLU (a) $\beta=1$ (b) $\beta=1$

The classical Swish activation has been extremely successful in PINNs. Initially, Swish was developed as a smooth ReLU. Although the exact reason for its superior performance over typical functions like tanh, sigmoid, or ReLU remains unclear, its behavior provides some clues (Fig. 2.1). As x approaches negative infinity, swish effectively blocks any signal. When x approaches positive infinity, swish acts as a linear region. This is extremely important because it conserves the gradient of x . Finally, the interpolation of these two regions can be thought of as a “buffer zone”, with a special point around -1.3 where the activation has a global minimum. In many ways, the Swish activation resembles a transistor for PINNs, regulating signal flow in a dynamic, differentiable manner that helps the network balance expressivity and stability.

The β -Swish activation, defined as $X^* \text{Sigmoid}(\beta^* X)$, with a learnable parameter β , is a promising choice for B Net[63]. Its second derivative resembles a Gaussian centered at zero with a peak magnitude of $\beta/2$, making it well-suited for interpolating sharp interfaces with discontinuous derivatives. At the same time, its first derivative remains well-behaved, staying within the approximate range $[0, 1]$. Finally, because β controls the sharpness of the transition, this can potentially provide a pathway for the network to “sharpen” its neuron outputs.

2.3.5 Weight Initialization Scheme

In typical neural network initialization schemes, the standard deviation σ of weight initialization governs the variance of layer outputs. Weight initialization schemes like Xavier aim to conserve the variance of each layer to be 1, which is done by setting

$$\sigma = \frac{\sqrt{2}}{\sqrt{N_{in} + N_{out}}} \quad (2.8)$$

where N_{in} and N_{out} are the input and output layer sizes, respectively. Assuming swish activation functions, the layer variance impacts the gradient initialization of the network with

$$E\left[\frac{\partial B}{\partial x}\right] \sim O(B) * O\left(\prod_0^L var_l\right) \quad (2.9)$$

where var_l is the variance of layer l and $\frac{\partial B}{\partial x}$ is the gradient of the Network output w.r.t. to the input x calculated using automatic differentiation.

2.4 High-Precision Techniques

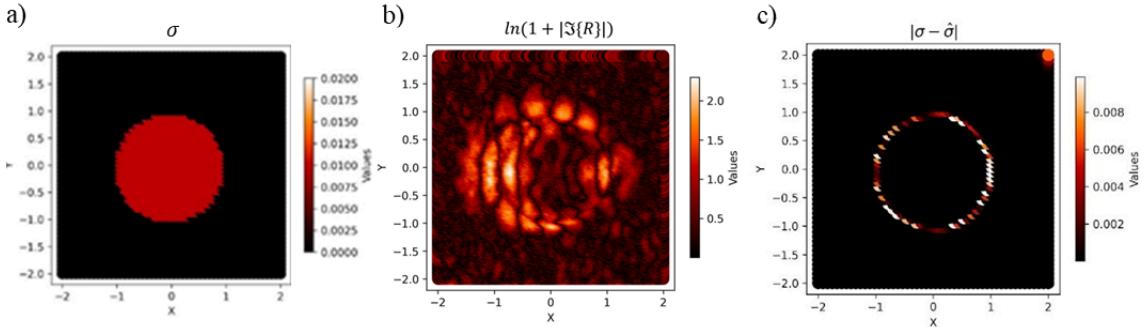


Figure 2.2: (a) Ground truth of the conductivity for the 2D cylinder (b) Logarithmic plot of the imaginary component of the Helmholtz residual(physics error) using a traditional MSE (c) Absolute difference between the ground truth and reconstructed conductivity corresponding to the model trained for (b).

The choice of the loss function plays a pivotal role in training neural networks effectively. A common approach is to minimize the Mean Squared Error (MSE) on all loss terms. While the MSE is a great choice for modeling data with gaussian noise, the MSE is not well suited for data with many outliers. Specifically, the MSE disproportionately weights extreme outliers due to squaring the loss, causing the optimization process to be biased towards these outliers. Now, when solving domains with discontinuous EPs, we expect the largest Helmholtz residual error to be located along the interface due to the discontinuities. This disproportionately high error region will dominate the MSE and the MSE will immediately try to resolve this area. However, this becomes a critical flaw of the MSE since it will distribute the high error associated with the interface to surrounding points. To demonstrate this effect, a standard deep neural network was trained with the traditional MSE for EPT on a 2D cylinder. The resulting absolute difference between the ground truth and predicted conductivity is displayed in Figure 2.2. We notice the reconstruction error lies exactly on the boundary of

the cylinder. Additionally, we see the Helmholtz residual vaguely outlining an artifact around this cylinder boundary. This diffusion of error leads to blurred reconstructions and prevents the network from accurately learning the sharp transitions critical to reconstructing the medium.

2.4.1 Ideal Focusing Effect

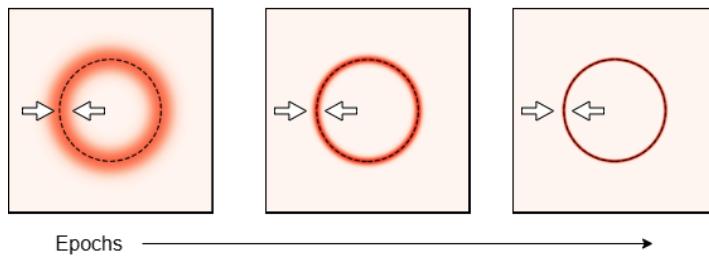


Figure 2.3: Example illustrating the loss distribution on a circular material interface. Stronger red colors indicate a high loss. This figure demonstrates the ideal focusing effect over the training epochs.

We aimed to create a strategy that causes the network to temporarily downweight high-error regions, allowing the network to first resolve the well-defined homogeneous regions. This strategy should result in a 'focusing' effect where reconstruction progresses inward from the error cloud edges (Fig. 2.3) In order to accomplish this, we utilize a novel loss function, collocation point resampling, and dropout layers as seen in Fig 2.4.

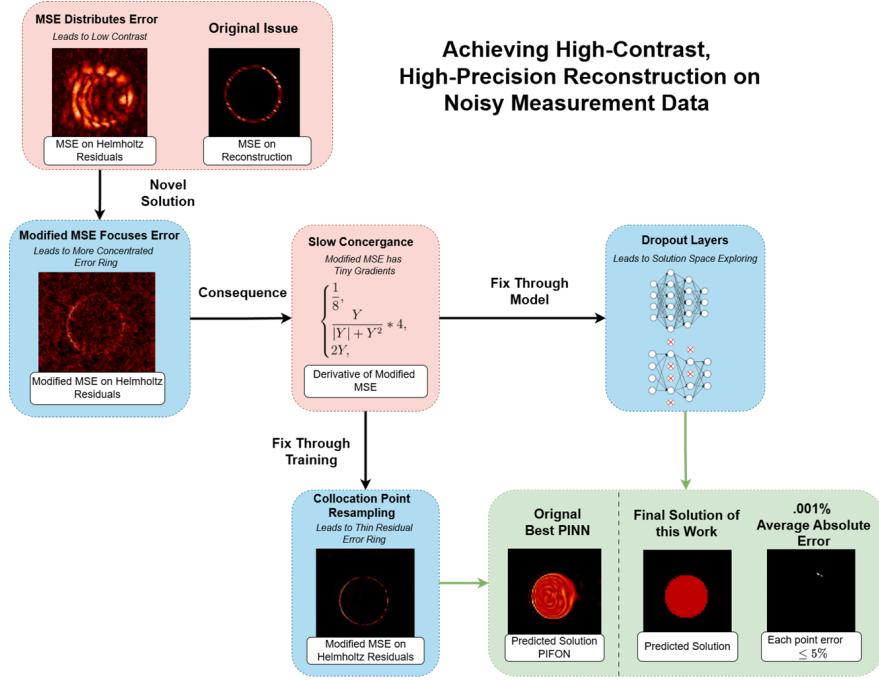


Figure 2.4: The methodological road map for achieving high precision on material interfaces. Key components of: Modified MSE, Collocation Point Resampling, and Dropout

2.4.2 Modified MSE

While the MSE loss is highly effective in homogeneous regions, where the solution tends to be smooth and well-behaved, it often performs poorly near material interfaces. In these boundary regions, sharp derivative transitions can lead to large local errors, and the MSE loss disproportionately emphasizes these errors, destabilizing training (Top Left of Fig 2.4). To address the large gradients of MSE in high-error regions we introduce a modified loss formulation. In areas of high residual, the standard MSE loss is locally replaced with an alternative function chosen for its reduced gradients. We adopt the following novel loss schemes:

$$L_{Proposed}(X) = \begin{cases} \frac{|X|}{8}, & |X| > 30.8 \\ \log_{1.286}(|X| + 1), & 1 < |X| \leq 30.8 \\ X^2, & |X| \leq 1 \end{cases} \quad (2.10)$$

$$L'_{Proposed}(X) = \begin{cases} \frac{1}{8}, & |X| > 30.8 \\ \frac{X}{|X| + X^2} * 4, & 1 < |X| \leq 30.8 \\ 2X, & |X| \leq 1 \end{cases}$$

$$L_{Proposed}(X, Y) = \begin{cases} \frac{|Y|}{8}, & \text{if } |X| > 30.8 \\ \log_{1.286}(|Y| + 1), & \text{if } 1 < |X| \leq 30.8 \\ Y^2, & \text{if } |X| \leq 1 \end{cases}$$

$$L'_{Proposed}(X, Y) = \begin{cases} \frac{1}{8}, & \text{if } |X| > 30.8 \\ \frac{Y}{|Y| + Y^2} * 4, & \text{if } 1 < |X| \leq 30.8 \\ 2Y, & \text{if } |X| \leq 1 \end{cases}$$

For the simplified Helmholtz residual, we use

$$\Re\{L_r\} = L_{Proposed}(|\Re\{R_H\}| + |\Im\{R_H\}|, \operatorname{Re}\{R_H\}) \quad (2.11)$$

$$\Im\{L_r\} = L_{Proposed}(|\Re\{R_H\}| + |\Im\{R_H\}|, \Im\{R_H\})$$

By computing the terms $|\Re\{R_H\}| + |\Im\{R_H\}|$ with `tf.stop_gradient`, these terms will behave as constants during backpropagation. When using the Generalized Helmholtz, the following adjustments are made:

$$\Re\{L_r\} = L_{Proposed}\left(\frac{|\Re\{R_H\}|}{|\hat{B}|}, \operatorname{Re}\{R_{GH}\}\right) \quad (2.12)$$

$$\Im\{L_r\} = L_{Proposed}\left(\frac{|\Im\{R_H\}|}{|\hat{B}|}, \Im\{R_{GH}\}\right)$$

Here, the Simplified Helmholtz residual is used as a diagnostic tool to identify regions where the solution is highly dependent on the gradients EP. Specifically, areas where the Simplified residual R_H is large but the Generalized residual R_{GH} remains small suggest that the network is heavily relying on gradient terms, often a sign of undertraining rather than physical accuracy. As a result, these regions are flagged and treated similarly to high-error zones in the simplified case, allowing targeted improvements during training.

Since the Generalized Helmholtz formulation is applied in MRI-like scattering cases, where the predicted B_1^+ field decays rapidly, we normalize the Simplified residual by the magnitude of B_1^+ to avoid biasing the metric toward low-field regions. This normalized residual gives a clearer picture of which areas need further attention during training. We have a final loss of

$$L = \Re\{L_d\} + \Im\{L_d\} + \frac{\Re\{L_r\} + \Im\{L_r\}}{c^2} \quad (2.13)$$

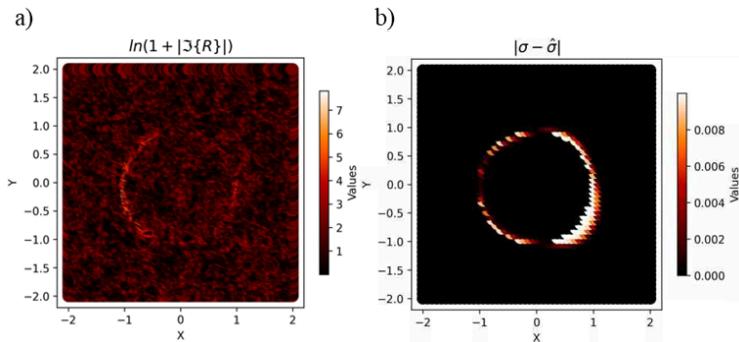


Figure 2.5: (a) Logarithmic plot of the imaginary component of the Helmholtz residual using Modified MSE (b) Absolute difference between the ground truth and reconstructed conductivity corresponding to the model trained for (a).

Compared with the classical MSE, the Modified MSE loss concentrates high error

regions, particularly near material interfaces, with even more error. While this is the behavior we expect, we notice that solely changing the MSE to the Modified MSE does not produce high-precision results (Fig 2.5). This is because the Modified MSE's small gradients and stiff EP Network inhibit learning.

2.4.3 Collocation Points Scheme

To fully leverage the proposed modified MSE, we resample the collocation points at each training iteration. Since the loss function is designed to resolve the boundaries of the error cloud through a "focusing" effect, it is crucial that the network is repeatedly exposed to these boundary regions. Randomly reassigning collocation points ensures diverse sampling of the domain, allowing the network to learn the full spatial structure of the residual error. This strategy also mitigates overfitting, enhancing the network's generalization across the entire domain once training is complete.

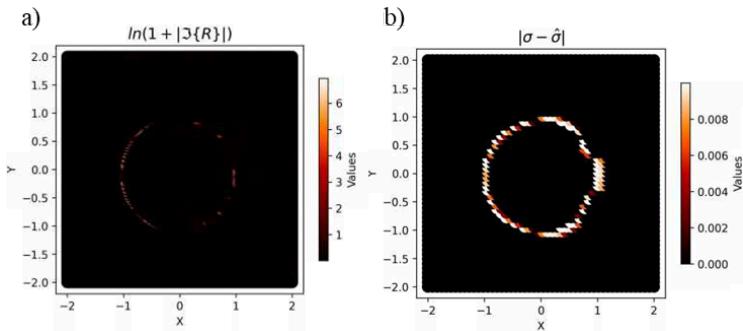


Figure 2.6: (a) Logarithmic plot of the imaginary component of the Helmholtz residual using Modified MSE and collocation point resampling (b) Absolute difference between the ground truth and reconstructed conductivity corresponding to the model trained for (a).

From Figure 2.6 we see that the residual loss now matches perfectly with the "focusing" theory. However, the reconstruction is still poor. This is an issue with the EP Net's tendency to get stuck in a local minimum, stalling further learning.

2.4.4 Dropout

Dropout layers, commonly used to prevent overfitting in deep classifiers, are particularly useful in PINNs, which are prone to overfitting and getting stuck in local minima[62]. The standard definition of a dropout layer is:

$$y = \frac{1}{p}m \odot x \quad (2.14)$$

where m is a binary mask resampled from a $\text{Bernoulli}(p)\{0,1\}$ each iteration with shape equal to x , \odot is element-wise multiplication, and x is an input layer (N, l_{size}) where N is the batch size and l_{size} is the number of neurons in the layer.

When using dropout in PINN, the $1/p$ scaling is typically removed to vary dropout probability without affecting activation magnitudes. Additionally, we introduce shared mask dropout, where a single binary mask is shared with all samples:

$$y_{shared} = S \odot x \quad (2.15)$$

where S is constructed by stacking the tensor s N times; while s has shape $(1, l_{size})$ and is sampled from $\text{Bernoulli}(p)\{0,1\}$. This causes the dropout mask to remain the same for each batch.

These dropout layers allow and force the PINN to explore nearby solutions and are an excellent choice when the loss landscape is very flat (near the end of training) or when resolving sharp boundaries. This is the final component of the “pinching theory”

methodology and final results will be detailed in 2.6.2 Comparison between MSE and Modified MSE.

Separately, when working with a linear PDE like the Helmholtz, a final dropout layer on the B Net could force each neuron connection to obey the linearity and cause better network utilization. Further investigation into dropout layers is needed, as these have a very interesting interpretation. By training a network with dropout layers, we minimize the objective function on the set of all possible subnet works. This pseudo-infinite sum of sub-networks might have very interesting uses yet to be uncovered.

2.5 Frequency Tuning

2.5.1 Multiscale PINN

To enable high-resolution reconstruction, a multiscale[8,64-65] design was introduced for the EP network. This architecture concatenates multiple deep sub-networks, or "scales," each focused on capturing different spatial frequencies in the data. A key feature of the multiscale design is the introduction of an input scaling factor for each scale. This scaling factor multiplies the input features in the first layer, setting the scale's preferred frequency. Lower scales capture coarse features, while higher scales focus on finer details, leading to multi-resolution training. The outputs from these scales are concatenated, passed through a Swish activation, and processed through a standard layer. A final linear layer maps the results to the two outputs representing the real and imaginary parts of the EP distribution.

To promote effective multi-scale learning, we propose a novel architecture: the Multiscale Dropout PINN (MS-DO-PINN). Our MS-DO-PINN architecture incorporates dropout layers, applied to the outputs of each scale, to regularize the model and reduce overfitting. This leads to an architecture that is both highly flexible and compact, allowing it to adapt to a wide variety of inverse problems and resolution requirements. However, the primary challenge with this architecture lies in determining the appropriate number of scales and selecting their corresponding scaling factors. An intuitive, but currently unverified, heuristic assumes that the input domain is normalized to the range $[0,1]$, in which case the lowest scale is typically set to 1. The choice of scaling factors should align with the expected feature sizes in the target solution. For instance, if the goal is to reconstruct structures with characteristic dimensions on the order of 0.01, then a suitable scaling factor would be $s_f = 100$, allowing the network to resolve such fine-scale features effectively.

Increasing the number of scales broadens the network's receptive field across frequency space, enabling it to learn features of varying granularity. In practice, a well-balanced set of scales such as $[1,2,4,6,8,10]$ offers a good compromise between model complexity and expressive power. A shared dropout mask is applied across the outputs of all scales, typically with a dropout probability of 10%, to promote generalization and reduce redundancy across scales.

In summary, the MS-DO-PINN enables robust, high-resolution reconstructions by effectively leveraging information across multiple spatial frequencies. Through careful selection of scaling factors and dropout strategies, this framework mitigates spectral bias,

while maintaining compactness and flexibility. These properties make it a powerful and practical solution for challenging inverse problems in medical imaging, particularly in the accurate recovery of complex-valued EP distributions.

2.5.2 Eigenfunction Decomposition and Random Fourier Features

Random Fourier Features (RFF) are traditionally used to approximate radial basis function (RBF) kernels and are defined as

$$\gamma(\mathbf{r}) = \begin{bmatrix} \cos(\mathbf{Br}) \\ \sin(\mathbf{Br}) \end{bmatrix} \quad (2.16)$$

Where r is the network input and $\mathbf{B} \in R^{m \times d}$ is sample from a random normal distribution $\mathcal{N}(0, s^2)$, where s is a hyperparameter. This transformation serves as a preprocessing step applied directly to the inputs, and \mathbf{B} is typically kept fixed and non-trainable. Also, γ is the input to the deep neural network. In the context of the B Net, the RFF layer can be interpreted as an eigenfunction decomposition of the Helmholtz operator. The eigenfunctions for the 2D Simplified Helmholtz are defined as

$$\sin(k_1x + k_2y + b_1) \quad (2.17)$$

$$\sin(k_1x + b_1) * \sin(k_2y + b_2)$$

$$\sin(k_3x + b_1) + \sin(k_3y + b_2)$$

Where $k_1^2 + k_2^2 = k_0^2$ and $\frac{k_3^2}{2} = k_0^2$, for $\nabla^2 B + k_0^2 B = 0$, and b is uniformly sampled from [-pi to pi]. To demonstrate these modes as valid Helmholtz solutions, we solve for their Laplacian:

$$\nabla^2 \sin(k_1x + k_2y + b_1) = -k_1^2 \sin(k_1x + k_2y + b_1) - k_2^2 \sin(k_1x + k_2y + b_1) \quad (2.18)$$

$$\nabla^2 \sin(k_1 x + b_1) * \sin(k_2 y + b_2) = -(k_1^2 + k_2^2) \sin(k_1 x + b_1) * \sin(k_2 y + b_2)$$

$$\nabla^2 \sin(k_3 x + b_1) + \sin(k_3 y + b_2) = -\frac{k_0^2}{2} \sin(k_0 x + b_1) - \frac{k_0^2}{2} \sin(k_0 y + b_2)$$

All of these basis functions satisfy the Simplified Helmholtz equation, allowing the network output to be interpreted as a weighted summation over these modes. Among these, the first two modes generally contribute the most to the final solution. As a result, the RFF transformation can be viewed as effectively decomposing the solution into its first eigenmode. If we choose to extend RFF into the second mode, we construct what will be referred to as the EigenFunction-PINN (EF-PINN):

$$EF(x, y, z) := \begin{bmatrix} [\sin(W_1 x + W_2 y + W_3 z + b_1)] \\ [\sin(W_4 x + b_2) * \sin(W_5 y + b_3) * \sin(W_6 z + b_4)] \end{bmatrix} \quad (2.19)$$

Since this is applied as a preprocessing layer, the EF-PINN effectively defines the functional basis upon which the neural network learns.

To imbed the network with the correct values of k , the standard deviation hyperparameter s of the weights W is selected according to the following scheme:

$$s = k_0 * \sqrt{E[\varepsilon_c]} * \frac{1}{\sqrt{d}} \quad (2.20)$$

where d is the size of the spatial inputs, in this 2D case $d=2$, $E[\varepsilon_c]$ is the expected magnitude of the ground truth complex EP.

If we choose to use an EF-PINN for the B Net, this scheme initializes the network already tuned to the dominant frequency, substantially increasing convergence in high-frequency spaces. Additionally, the rest of the B Net layers follow an Xavier

distribution on their weight initialization to make their variance = 1, resulting in a final Laplacian $O(\nabla^2) \sim O(k_0^2 * E[\varepsilon_c])$.

2.6 Architecture Overview

The proposed architecture uses a separate network for ε_c and B_1^+ , the EP Net and B Net. Both of these networks have 3 spatial inputs, (x,y,z), and predict B_1^+ and ε_c at the (x,y,z) spatial coordinate. Since B_1^+ and ε_c are complex valued, each submodel has 2 outputs, one for the real and one for the imaginary part. Hence, all equations are separated into real and imaginary components. The complete workflow of the architecture can be seen in Fig 2.7 and Fig. 2.8.

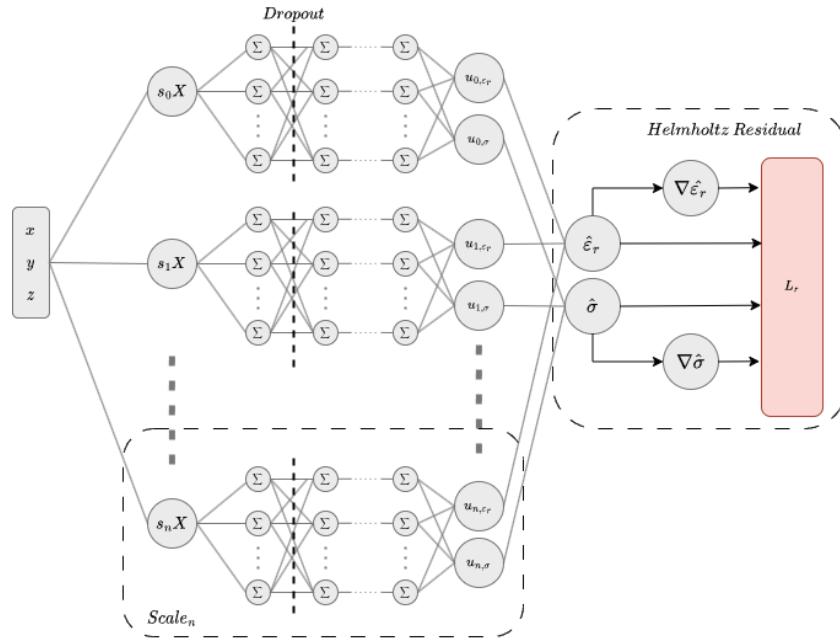


Figure 2.7: High level diagram of EP Net used in this thesis

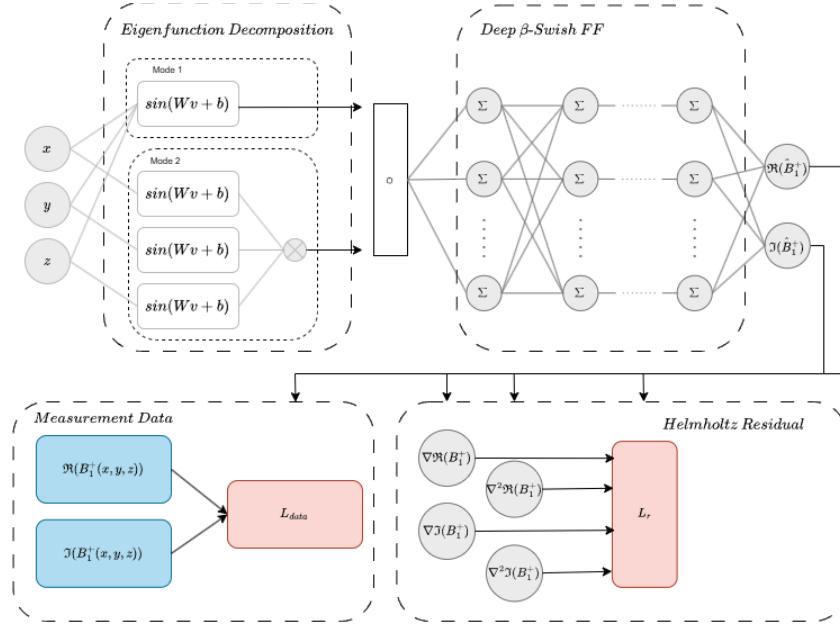


Figure 2.8: High level diagram of B Net used in this thesis

The MS-DO-PINN EP network adopts a multi-scale architecture consisting of 4 sub-networks (scales) with input scaling factors [1,2,4,8]. Each sub-network contains 3 hidden layers with 16 neurons per layer and using the Swish activation function. A Shared Mask Dropout layer with a dropout probability of 1% follows each hidden layer to mitigate overfitting and enhance generalization. Each scale produces two outputs, a real and imaginary component, which are passed through Swish activations. The outputs corresponding to the real and imaginary parts are concatenated into separate layers, each followed by another Shared Mask Dropout layer with a 5% dropout rate. This encourages the model to utilize all scales during training. A final linear layer then combines these to produce the complex EP output.

For cases where expected EP values are known (e.g., in medical imaging with known tissue properties), a sigmoid mapping can be applied to the final output to

constrain the solution space. In a two-material system, the EP Net output passes through a sigmoid activation where 0 and 1 represent the two materials. A single learnable neuron then maps this range to the corresponding material permittivities, e.g., from air ($\varepsilon_r=1$) to glass ($\varepsilon_r=4.7$). The final neuron would be initialized with a learnable weight of (4.7–1 = 3.7) and a bias of 1.

The EF-PINN B Net begins with an eigenfunction mode decomposition initial layer that includes the 1st and 2nd modes, each represented by 256 neurons. These mode components use independent weight initializations and are set to be non-trainable. The outputs of the eigenfunction modes are concatenated to form the input to the next layer. Following this, the network consists of two fully connected layers employing β -Swish activation functions. Each β -Swish layer has a trainable β parameter, initialized from normal distributions with standard deviations of 6.0 and 2.0, respectively. Since the order of the Laplacian for β -Swish is equal to half of β , the 6.0 initialization is designed to represent a transition in the EP from a value of 1.0 to 3.0. The 2.0 initialization, on the other hand, corresponds to an expected Laplacian value of 1.0, aiming to maintain the overall network Laplacian. Finally, the final output layer produces both real and imaginary components, with bias disabled.

The EP Net and B Net were designed with high precision capabilities and a focus on "initializing the network to closely match the expected solution." Through various techniques, the EP Net excels at interface modeling, while B Net is optimized for frequency tuning.

2.6.1 Training and Testing

For all experiments, we used simulated B_1^+ maps and corrupted them with white Gaussian noise with a standard deviation equal to the ratio of the peak value of $|B_1^+|$ to a prescribed peak signal-to-noise-ratio (SNR) value. All experiments used a SNR value of 200. The B_1^+ maps are grids of size N^d . In this thesis, we assume that the testing resolution is the same as the data point resolution.

To measure the error on the reconstruction between the $\text{pred}(\hat{v})$ and the ground truth(v), we used the peak normalized absolute error (PNAE) as defined

$$\text{PNAE}(\hat{v}, v) = \frac{\|\mathbf{v} - \hat{\mathbf{v}}\|_1}{\|\mathbf{v}\|_\infty} \quad (2.21)$$

We aim to achieve an average error of .1%.

2.7 Results

2.7.1 2D Cylinder under a plane wave

The dual MS-DO-PINN and EF-PINN architecture is compared to the original PIFON on an analytical 2D solution. Using Mie Scattering theory, the B_1^+ field is computed for an infinitely long homogenous cylinder under a TMz plane wave excitation. The EPs of the cylinder are of 3.0 relative permittivity and .01 conductivity S/m. The outside of the centered cylinder is air with values of 1.0 for relative permittivity and 0.0 for the conductivity. The operating wavelength is 1.00 m, and the cylinder has a radius equal to the wavelength. The computation domain is $[-2r, 2r] \times [-2r, 2r]$ with a resolution of 50x50 points, corresponding to a total of 2500 voxels. The number of

collocation points is set to $(2*50)^2 = 10000$ points. The resulting B_1^+ field is corrupted with white Gaussian noise and used as the input to the PINN.

The network consists of a MS-DO-PINN EP network and an EF-PINN B network using modes 1 and 2. The models were trained jointly with an Adam optimizer for 60k epochs. The learning rate was a step function with values $[5 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$ with transitions at 20k and 40k iterations.

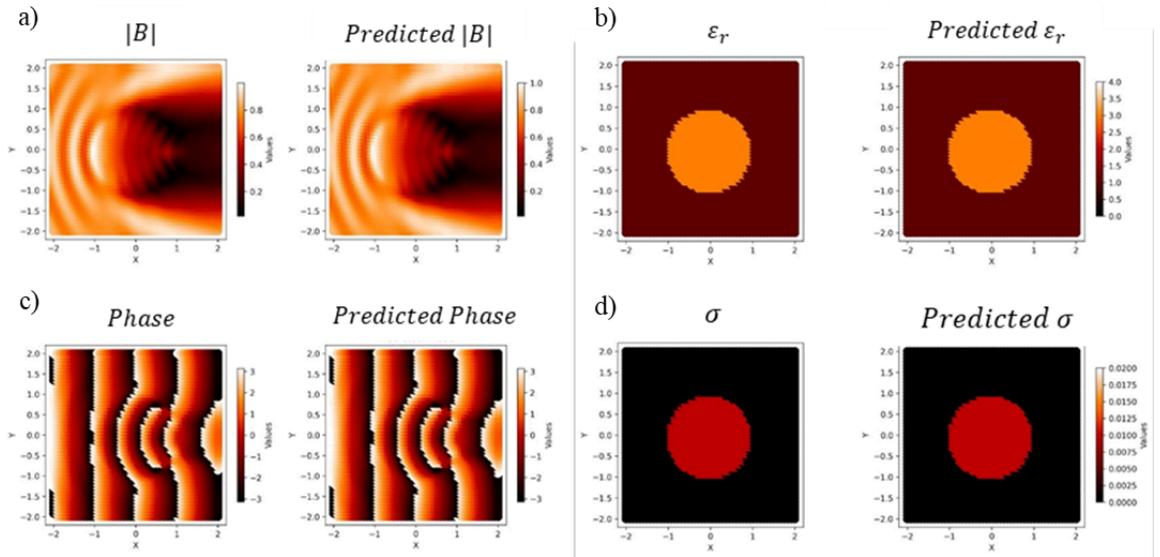


Figure 2.9: Ground truth(left) and predicted(right) values for (a) $|B|$, (b) relative permittivity, (c) phase(B), and (d) conductivity, respectively

The EPT reconstruction results are shown in Figure 2.9. The proposed PINN approach achieves near machine-precision accuracy, successfully reconstructing both the relative permittivity and conductivity across the entire domain, including at the boundaries. The average PNAE on the domain was measured to be 0.005% for the relative permittivity and 0.009% for the conductivity.

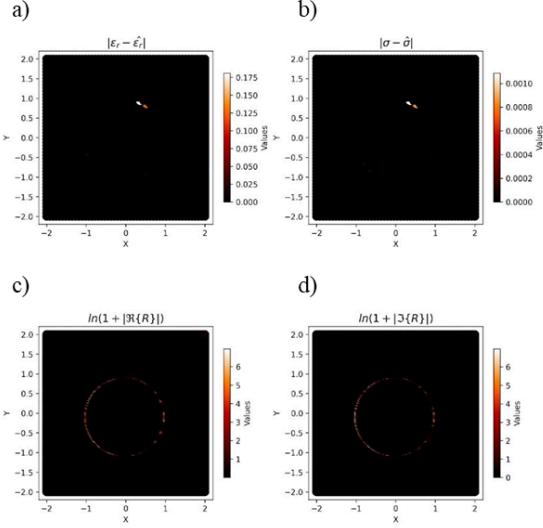


Figure 2.10: Absolute difference between the ground truth and predicted values for (a) relative permittivity and (b) conductivity evaluated on the data grid. The log plots for the (c) real and (d) imaginary parts of the Simplified Helmholtz residual on a dense grid

Figure 2.10 shows the absolute difference between the ground truth and the predicted values, evaluated on the same grid as the original data points. The reconstruction error is concentrated within two voxels where the predicted values differ from the ground truth by approximately 5%. Additionally, Figure 2.10 presents the logarithmic plot of the simplified Helmholtz residual, evaluated over a dense grid. This residual highlights how well the reconstructed fields satisfy the underlying physics throughout the domain. From the log plot of the residual, a thin ring of high-error is observed precisely along the material interface, matching the pinching theory. The thinning out of this ring is the main motivation of the modified MSE and is demonstrated to be extremely successful.

The dual MS-DO-PINN and EF-PINN architecture accurately reconstruct both permittivity and conductivity in the analytical 2D Mie scattering test case. With near

machine-precision accuracy and minimal error concentrated at the material interface, the results demonstrate the effectiveness of the joint architecture and the modified MSE loss in capturing sharp transitions and preserving physical consistency across the domain.

2.7.2 Comparison to the original PIFON

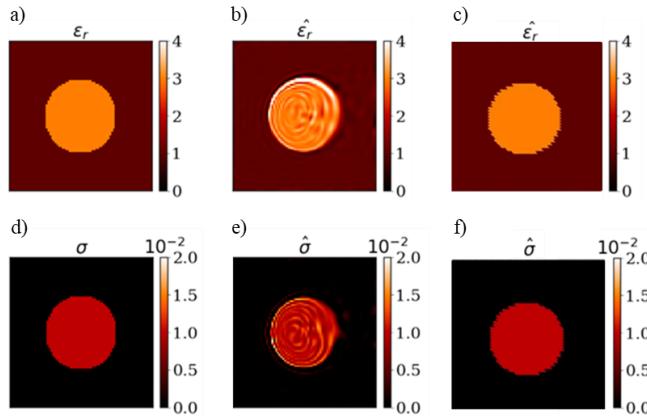


Figure 2.11: Comparison to PIFON. (a and d) ground-truth, (b and e) PIFON prediction using the Simplified Helmholtz, (c and f) this work’s prediction using the Simplified Helmholtz for relative permittivity and conductivity, respectively

Figure 2.11 presents a direct comparison between our results and those obtained in the original PIFON work. The average PNAE over the domain for the reconstructed relative permittivity and conductivity has been dramatically reduced from 3.96% and 9.67% to 0.005% and 0.009%, respectively. This substantial improvement over PIFON was achieved through the introduction of a modified MSE function, along with careful fine-tuning of the network architectures. In our approach, discrepancies at material interfaces are typically confined to isolated, single-voxel-width deviations, demonstrating the network’s enhanced ability to capture sharp discontinuities while maintaining global solution accuracy.

2.7.3 Comparison between MSE and modified MSE

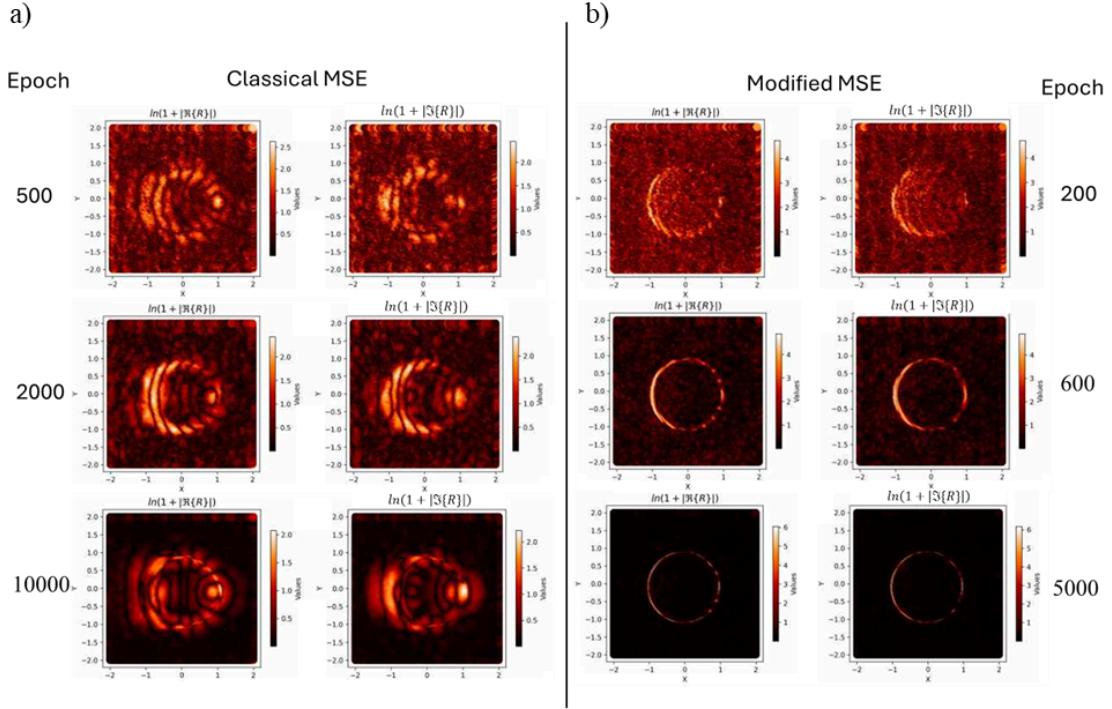


Figure 2.12: Logarithmic plots of the real and imaginary components of the Simplified Helmholtz residual during training with collocation point resampling and dropout layers. (a) Classical MSE shows distributed error. (b) Modified MSE creates a high-error ring

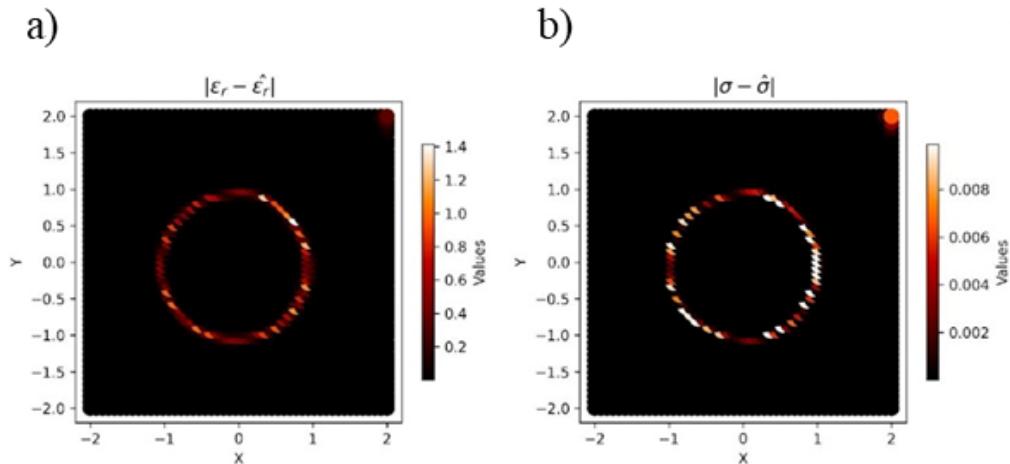


Figure 2.13: Final reconstruction error for (a) relative permittivity and (b) conductivity using classical MSE with collocation point resampling and dropout layers

To better understand the critical role of the modified MSE, we conducted a controlled study where the model architecture was held constant and only the MSE formulation for the Helmholtz residual was changed. Figure 2.12 shows a side-by-side comparison between the classical MSE and the modified MSE distributions. The reconstruction absolute error for the classical MSE is shown in Figure 2.13, where it becomes evident that the standard approach fails to accurately reconstruct the material interface.

Interestingly, the mean loss for both the classical and modified MSE approaches is approximately the same, but the distribution of errors is different. Due to the presence of sharp derivative discontinuities at the material interface, the classical MSE struggles. The disproportionately high residuals at the interface are smoothed out and redistributed into the surrounding regions, creating a pronounced "blurring" effect (visible in Figure 2.12). Even after extensive training, the residual cloud remains broad and diffuse. This behavior is a consequence of the classical MSE's hard treatment of outliers, forcing the network to immediately reduce large residual errors, but in doing so, it compromises the resolution of nearby regions. Once this blurred error structure is established, further learning becomes highly inefficient, as any local updates along the interface sharply increase the overall residual, effectively trapping the network in a poor local minimum.

In contrast, the modified MSE leads to a very different training dynamic, as seen by the sharp high-error ring at the interface in Figure 2.12. The modified MSE relaxes high-magnitude residuals, allowing the network to first fit the remainder of the domain

more accurately. This comes at the cost of higher maximum residuals concentrated along the material boundary. As can be seen in Figure 2.12, the maximum residual errors grow over the epochs to approximately 1000 for the modified MSE, compared to only 12 for the classical MSE.

Thanks to collocation point resampling at each training iteration, the architecture can effectively "pinch" the high-error ring thinner over time. When resampling points on the domain, most points fall into the already well-resolved homogeneous regions, stabilizing the solution; some land directly on the high-error interface, while others land on the boundary of the residual ring, driving the learning needed to shrink and sharpen the interface error. Without resampling, only a limited amount of the high-error boundary would be explored, leading to a lack of convergence.

Thus, the combination of the modified MSE, the collocation point resampling scheme, and dropout layers is essential for achieving near-machine precision reconstruction across sharp material interfaces.

2.7.4 Realistic tissue EPs

To reconstruct similar situations to MRI environments, we will use the EPs of grey matter (GM) and white matter(WM) with the following profiles: GM $\{\varepsilon_r = 60, \sigma = .692\}$, WM $\{\varepsilon_r = 43.8, \sigma = .413\}$. We assume the EPs follow a +-5% variance and add noise to the EP during B_1^+ simulation. Furthermore, we will use a wire source excitation operating at 300 MHz, similar to what is expected in MRI systems. A 2D circular domain

of radius $R = .11$ m is constructed, with an inner phantom cylinder. The outer region will correspond to WM, and the inner cylinder will be GM. The system configuration and geometry can be seen in Figure 2.14. The data was acquired by computing the magnetic field on a grid $[-r, r] \times [-r, r]$ and then only points inside the radius of the boundary will be extracted. We use 50 points to sample the grid, resulting in 1568 points inside the circular domain. The data was acquired with Mie Scattering theory.

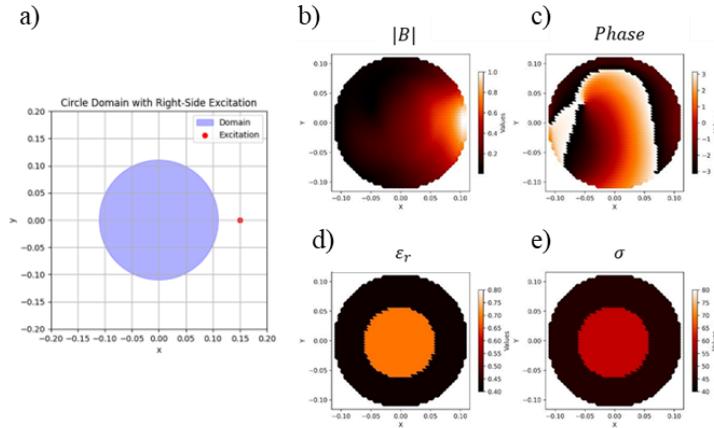


Figure 2.14: (a) The domain of the system in meters. The ground truth for the (b) B field intensity, (c) B field phase, (d) relative permittivity and (e) conductivity

Since the domain under consideration is small and the feature sizes are tiny compared to the wavelength, we employ the generalized Helmholtz residual for reconstruction. This setup presents a significantly more challenging problem: the spatial domain only captures a small fraction of the wavelength, and the system's B_1^+ magnitude decays rapidly, as illustrated in Figure 2.14.

Moreover, the presence of noise further complicates reconstruction. Because we assume a constant noise variance across the entire domain, the left-hand side, where the

true signal magnitude is much smaller, experiences substantially higher relative noise levels. As a result, the SNR is dramatically lower in this region, making accurate recovery of fine features even more difficult. Despite these challenges, the dual MS-DO-PINN and EF-PINN approach demonstrates strong resilience, showcasing the method's robustness under highly non-ideal conditions.

The model architecture follows the standard MS-DO-PINN for EP Net, with 6 scales each with a scaling factor of 1. The concatenated scale outputs are fed through a shared mask dropout with probability of 10%. Finally, the output for of the complex valued EP is passed through a learnable sigmoid layer to simulate different materials. This layer is designed for material reconstruction, with initial weights and biases calibrated to map the canonical sigmoid range [0,1] to material-specific permittivity and conductivity values, WM and GM in this case. These material values are learnable to allow deviations from the estimated properties. To improve the convergence of the loss, a contrast loss term is introduced in addition to the standard losses.

The contrast loss is defined by first rounding the outputs of the final sigmoid layer, which fall between 0 and 1, to their nearest integer value, also either 0 or 1. The MSE is then computed between these sigmoid outputs and their rounded values. This contrast loss is scaled by a small hyperparameter, $\alpha = .0001$, to prevent it from dominating the total loss, but is sufficient to promote more distinct transitions between materials.

The architecture was trained for 120k iterations using 4,096 collocation points and a piecewise constant learning rate schedule of: $[5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$ decreasing every 30k iterations.

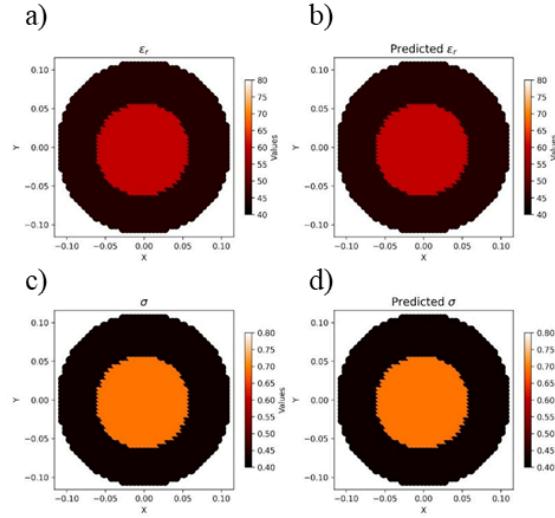


Figure 2.15: Ground truth for (a) relative permittivity and (c) conductivity. Predicted values for (b) relative permittivity and (d) conductivity.

The results are shown in Figures 2.15 and 2.16. The final reconstruction achieved an average PNAE of 0.12% for relative permittivity and 0.08% for conductivity. Importantly, the maximum point-by-point deviation between the predicted and true material properties is less than 1%. Thus, if a final post-processing step of “cutting” the solution by rounding each prediction to the nearest expected material property is applied, the network perfectly recovers the true underlying material distribution.

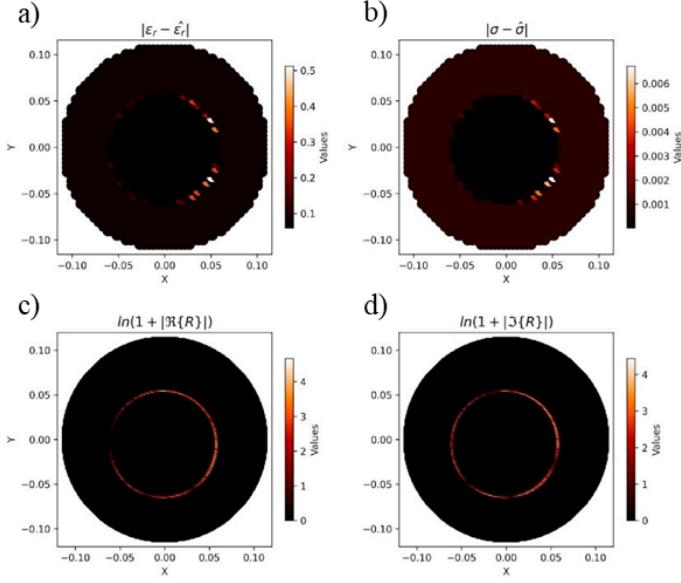


Figure 2.16: Absolute difference between the ground truth and predicted values for (a) relative permittivity and (b) conductivity evaluated on the data grid. The logarithmic plots for the (c) real and (d) imaginary parts of the Generalized Helmholtz residual on a dense grid

Similar to the plane wave excitation case, a thin high-error ring is observed around material interfaces due to the use of the modified MSE error when solving the Generalized Helmholtz equation (Fig 2.16). This ring structure again reflects the network's strategy of solving the high-error region by pinching. It is important to note, however, that the modified MSE converges more slowly for the generalized Helmholtz compared to the simplified Helmholtz. This is likely because the generalized Helmholtz includes EP gradient calculations.

Despite the challenges of low signal-to-noise ratio, fine-scale features, and small domain size relative to the wavelength, the dual MS-DO-PINN and EF-PINN architecture successfully reconstructs brain-like EP distributions, demonstrating strong robustness and precision under MRI-like conditions.

2.7.5 3D Sphere under a plane wave

In this example, we extend the methodology from 2D to 3D. The setup mirrors that of the 2D plane wave system, but is adapted to a 3D spherical geometry under a 300 MHz plane wave excitation. The sphere has a radius of 0.5 meters, a relative permittivity of 3, and an electrical conductivity of 0.01 S/m. The computational domain is a cube defined by $[-1.5r, 1.5r]^3$, discretized with 25 sample points along each spatial dimension, yielding a total of $25^3 = 15,625$ points. The number of collocation points is set equal to the number of data points.

The network is trained using a piecewise learning rate schedule of $[5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$, decreasing every 20k iterations over a total of 80k epochs. This network comprises six multi-scale branches with scaling factors [1, 2, 3, 4, 0.5, 0.25]. Each scale branch consists of a trainable Sine-activated first layer with 16 neurons, three Swish-activated hidden layers with 16 neurons, followed by a Swish-activated output layer. A shared mask dropout with a probability of 1% is applied after each scale layer, and an additional 20% shared mask dropout is applied across the combined scale outputs. The final output is passed through a learnable sigmoid activation.

The B Net uses an untrainable 2nd mode EF layer with 128 neurons, followed by two Swish-activated layers with 64 neurons each. Training is conducted using the simplified Helmholtz residual formulation.

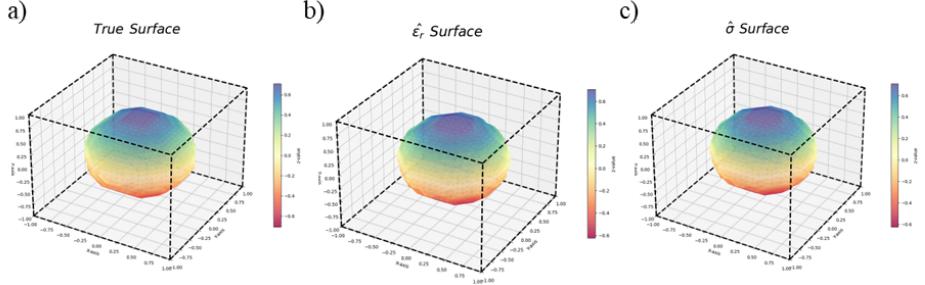


Figure 2.17: Surface rendering of (a) ground truth sphere, (b) predicted relative permittivity, and (c) conductivity. The color signifies the z-value of a point for depth perception. The spatial coordinates are normalized.

The 3D material reconstruction results are visualized in Figure 2.17 using surface rendering. The reconstructed material maps are obtained by rounding the predicted relative permittivity to the nearest material class, either 1 or 3, and applying a similar rounding for the conductivity. Despite the resolution limitations, the network successfully captures the spherical structure and underlying material contrast. The average PNAE is 0.12% for relative permittivity and 0.17% for conductivity. For relative permittivity, this error translates to only 30 misidentified points out of the 15,625-point grid. Furthermore, this 3D sphere has 2,668 points on its boundary, resulting in a boundary reconstruction error of approximately 1%. Figures 2.18 and 2.19 show 2D plane slices through the sphere's midpoints. Notably, the X-Y and Y-Z slices exhibit machine-precision accuracy, with only 2 misidentified points on the X-Z plane. These results are especially impressive given the relatively low collocation-to-data point ratio, far from the ideal minimum of 8:1 for 3D. This suggests that if data and collocation points were scaled to mirror the 2D case, near machine-precision reconstructions could be achievable on the noisy 3D B_1^+ measurements.

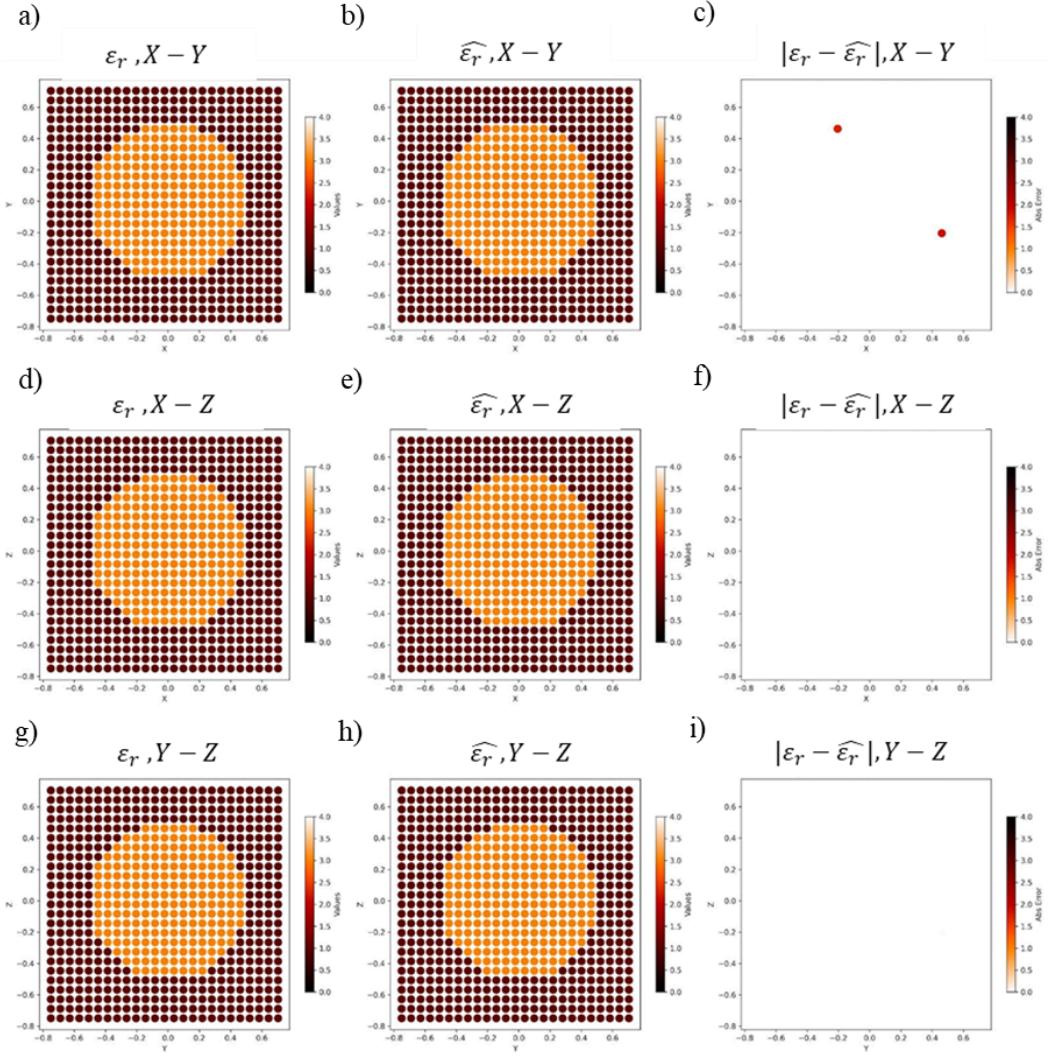


Figure 2.18: 3D Relative Permittivity Reconstruction. Plots of the ground truth, predicted value, and absolute difference for the (a-c) X-Y plane, (d-f) X-Z plane, and (g-i) Y-Z plane, respectively. White represents no effective difference in the absolute difference plots.

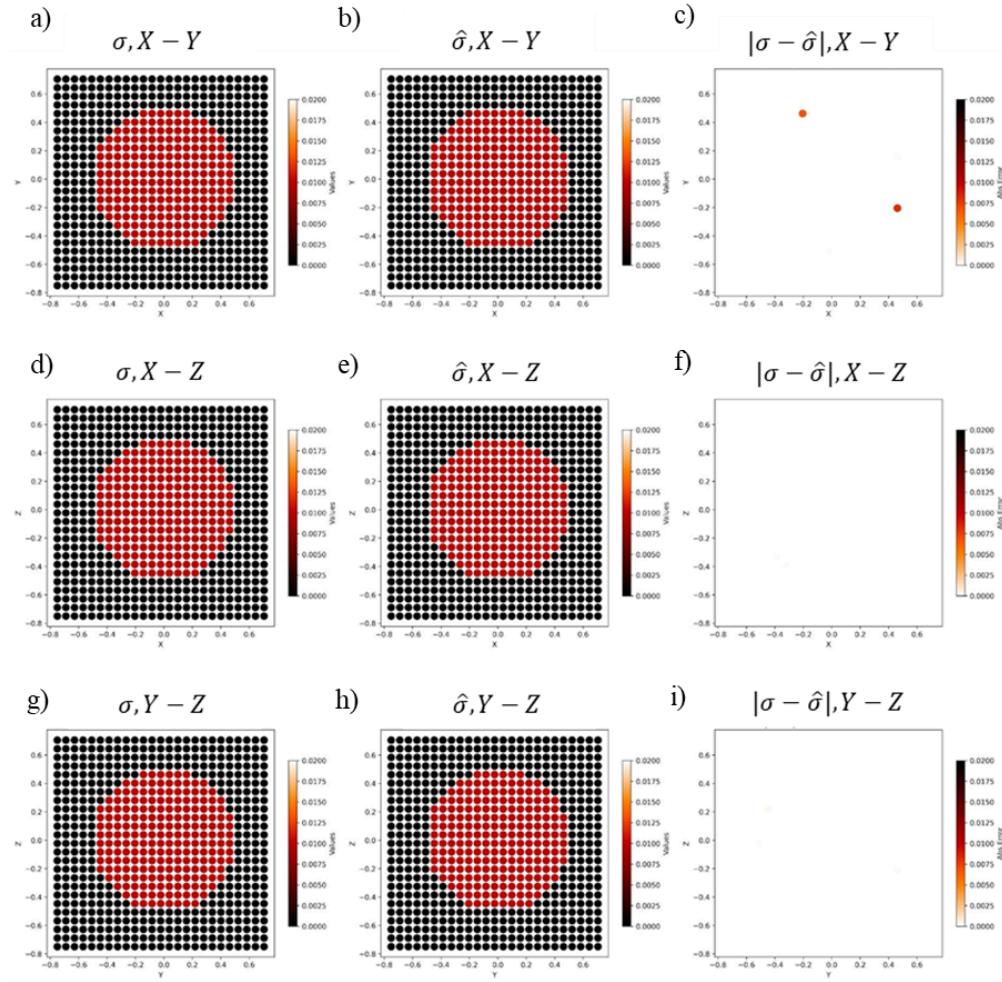


Figure 2.19: 3D Conductivity Reconstruction. Plots of the ground truth, predicted value, and absolute difference for the (a-c) X-Y plane, (d-f) X-Z plane, and (g-i) Y-Z plane, respectively. White represents no effective difference in the absolute difference plots.

2.8 Summary

Method	Effect
Loss Scaling	Faster Convergence
Number of Collocation Points	Determines Resolution
β -Swish	Faster Convergence
Weight Initialization	Ensures correct output/derivative scaling
Focusing Effect	Focuses error into thin ring and enables high-precision reconstruction
-- Modified MSE	Ignores high-error regions
-- Collocation Point Resampling	Ensures exploration of entire domain
-- Dropout Layers	Avoids local minima and overfitting
MS-DO-PINN	Avoids local minima and overfitting
EF-PINN	Allows tuning to specific spatial frequencies

Table 2.1: Displays the methods used and a summary of their effects on training and reconstruction

Hence, by incorporating general techniques, high-precision techniques, and finally frequency tuning techniques, we are able to improve the baseline PIFON framework and achieve high-precision $\leq .1\%$ reconstruction on noisy data. The general techniques of loss scaling, picking number of collocation points, using β -Swish activation functions, and weight initialization are essential to understanding the dynamics of the network. By constructing a balanced loss function, we provide the network with a navigable loss landscape that is neither too flat nor too steep. Building and initializing a network that immediately reflects the expected output is essential for fast and stable convergence. These techniques serve as the basis for further expansion into advanced high-precision and frequency tuning techniques. By refining the basic loss balancing method, we develop a novel “Focusing Effect” approach that spatially balances localized

errors. Furthermore, with the understanding of general network initialization, we are able to develop a Helmholtz-specific functional basis. Thus, each method presented simply balances the loss or provides the network with a “good first guess”. These simple concepts nudge the learning and network in the correct direction, further impeding the previously known “physics” into the Physics-Informed Neural Networks.

BTE Study

After developing the successful MR-EPT PINN framework, a secondary part was added to this thesis. By using the deep understanding of loss balancing and network initialization, we apply our methods to a difficult forward problem. In this case, instead of attempting to retrieve an unknown parameter with data, we will be solving for a PDE with boundary conditions. Specifically, we will study the phonon Boltzmann Transport Equation with the goal of increasing computational efficiency.

3.1 Motivation

Understanding thermal transport across a wide range of length scales is crucial for advancing technologies like thermoelectric materials[38,39] and compact electronic devices, like CPUs[41,42]. The phonon BTE provides a comprehensive framework for modeling thermal profiles at various length scales and thermal ranges[43,44]. In nano-sized systems or under large temperature gradients, phonon transport transitions between diffusive and ballistic regimes, significantly impacting material behavior and thermal profiles[45,46]. As illustrated in Figure 1.3, the phonon BTE enables detailed simulation of heat transport in next-generation transistor architectures, providing insights critical for optimizing thermal management. These simulations support the design of transistors with improved heat dissipation capabilities. However, solving the BTE accurately across different transport regimes remains challenging due to high computational demands caused by its high dimensionality.

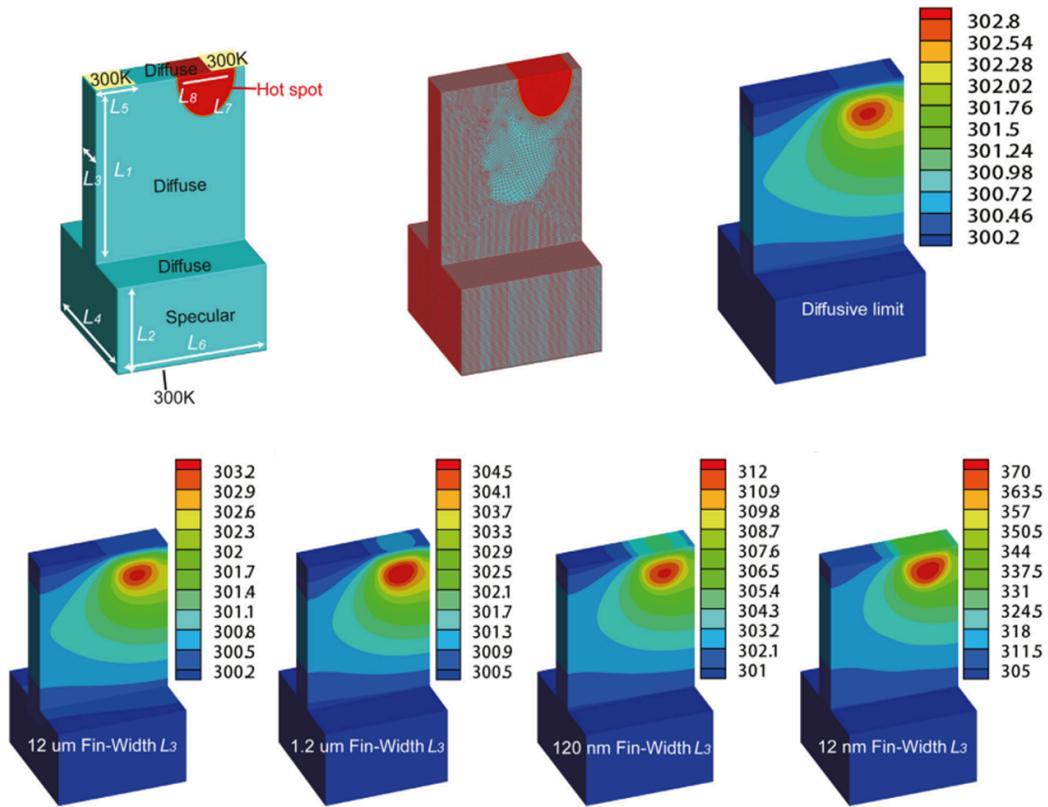


Figure 1.3: Example of thermal profiling in a silicon Fin Field-Effect Transistor using the Discrete Ordinates Methods[47]

Traditionally, simplifying assumptions are made to reduce computational complexity. These include treating phonon dispersion as isotropic, applying the single-mode relaxation time approximation, and assuming only small temperature deviations from a reference state[48-51]. Under these conditions, the phonon distribution can be linearized around temperatures[52,53] and relaxation times are often considered temperature-independent and spatially uniform. While such approximations have yielded valuable insights into nanoscale heat transport, they fall short when large temperature gradients are present. In reality, phonon relaxation times and mean free paths vary significantly with temperature[44,45], altering the local thermal conductivity and causing

phonons of different frequencies and polarizations to behave differently across a device. This variance causes the BTE to become multiscale in both frequency and space, making it a challenging task to solve. Although various numerical techniques have been developed, such as Monte Carlo (MC) simulations[53-55] and deterministic discretization methods[56-58], each has limitations. MC methods suffer from statistical noise and inefficiency at low Knudsen numbers, while the widely used deterministic approach, discrete ordinate method (DOM), tends to converge slowly in diffusive regimes and requires large memory resources[59]. Given these limitations, few methods today can predict phonon heat transport both accurately and efficiently when arbitrary temperature gradients are involved.

To address these challenges, Ruiyang Li *et al.* proposed a PINN framework for solving the mode-resolved phonon Boltzmann Transport Equation (BTE) [60,61]. Their original implementation, developed in the Python library PyTorch, focused on solving the forward problem of thermal transport in silicon across various length scales and system geometries. However, due to the use of large neural networks and extensive data arrays for boundary conditions, their models were computationally inefficient, requiring approximately 4 hours to complete a single 2D rectangular simulation.

3.1.1 Proposed PINN Solution

First, Li et al.'s code was translated into the Python library TensorFlow for compatibility with our research group's existing infrastructure. Building on Li et al.'s foundation, we developed an optimized framework prioritizing computational efficiency.

Our framework includes strict enforcement of boundary conditions, tailored network initialization, and balanced loss weighting. This improved framework will be referred to as Hard Constraint PINN (HC-PINN) and will be compared with the original PINN developed by Li et al. Exact boundary enforcement allows the PINN to focus exclusively on learning the interior solution of the PDE, while well-designed initialization and loss balancing promote faster convergence and improved training stability. Together, these enhancements enable the use of smaller networks and reduce training time from 4 hours to approximately 10 minutes per length scale simulation.

Overall, this work shows that our TensorFlow-based HC-PINN framework serves as an extremely efficient solver for high-dimensional forward problems. In particular, HC-PINN solves the phonon BTE across multiple length scales, offering a scalable tool for simulating thermal transport in complex semiconductor devices

3.2 PINN Objectives

The goal of PINN for the forward phonon BTE solver is to produce a temperature graph on the domain $\Omega \in R^2$ given isothermal boundary conditions, material relaxation times, phonon velocities, and the domain characteristic length. To achieve this, we define the phonon BTE residuals with the help of Eq. (5.6) and Eq. (5.9) from the Appendix as

$$R(\mathcal{F}(x, s, k, p)) := \begin{cases} \vec{v} \cdot \nabla(\hat{f}_{eq} + \hat{f}_{neq}) + \frac{\hat{f}_{neq}}{\tau(\hat{T})} \\ \sum_p \int_0^{\omega_{\max,p}} \int_{4\pi} \vec{v} \hbar \omega D(\tau(\hat{T})[\vec{v} \cdot \nabla^2(\hat{f}_{eq} + \hat{f}_{neq})]) d\Omega d\omega \end{cases} \quad (3.1)$$

where R is the residual function, \mathcal{F} is the entire network, and $\hat{f}_{eq}, \hat{f}_{neq}, \hat{T}$ are the predicted f_{eq}, f_{neq} , and T respectively. To solve the BTE, the residual function will be minimized, which involves solving for f_{eq} and f_{neq} . The first equation in the BTE residual is the PDE residual, and it is further split into 2 residuals, one for $p = 0$ and the other for $p=1$. Hence, the total number of loss terms is three, two for the PDE and one for the physical constraint. The architecture consists of separate models for f_{eq} and f_{neq} , but these models are jointly trained. The equilibrium distribution f_{eq} is determined by the Bose-Einstein distribution which is a function of temperature, wavenumber, and polarization(Eq. 2). Since only T is unknown and it is a function of (x,y) , the f_{eq} network, referred to as Feq Net, has inputs (x,y) and outputs $T(x,y)$ as an intermediate step, with f_{eq} being the final output. The non-equilibrium distribution f_{neq} network, referred to as Fneq Net, has input of the spatial coordinates (x,y) , normalized directional velocities (μ, η) , wavenumber k , and polarization p .

3.3 Efficiency Techniques

3.3.1 Loss Scaling

A critical component of PINN training is the proper balancing of loss terms. In general, each loss component should contribute equally to the overall loss, which can be achieved by ensuring that their orders of magnitude are of the same order: $O(L_i) \sim O(1)$. To estimate the expected order of magnitude for each term, we begin by substituting approximate values into the governing residuals. We typically start with a

term whose order we understand and whose expected range is known, often a term determined by the boundary conditions. In a phonon BTE system, that would be ∇T , which is determined by the boundary conditions. By figuring out the order of the terms multiplying ∇T , we can determine the order of the equation.

$$\begin{aligned}
\tau \vec{\mathbf{v}} \cdot \nabla(f_{eq} + f_{neq}) + f_{neq} &= \tau \vec{\mathbf{v}} \cdot \left(\frac{df_{eq}}{dT} \nabla T + \nabla f_{neq} \right) + f_{neq} \quad (3.2) \\
O \left(\tau \vec{\mathbf{v}} \cdot \left(\frac{df_{eq}}{dT} \nabla T + \nabla f_{neq} \right) + f_{neq} \right) &\approx O \left(\tau \vec{\mathbf{v}} \cdot \left(\frac{df_{eq}}{dT} \nabla T + \nabla f_{neq} \right) \right) \\
&\approx O \left(\tau \vec{\mathbf{v}} \cdot \frac{df_{eq}}{dT} \nabla T \right) \approx \mathbb{E}[\tau \vec{\mathbf{v}} \frac{df_{eq}}{dT}] \times \mathbb{E}[\nabla T] \\
\mathbb{E}[\tau \vec{\mathbf{v}} \frac{df_{eq}}{dT}] &\approx \tau v \mathbb{E}\left[\frac{df_{eq}}{dT}\right] \\
\mathbb{E}\left[\frac{df_{eq}}{dT}\right] &\approx \frac{df_{eq}(\mathbb{E}[\omega], \mathbb{E}[T])}{dT} \\
\mathbb{E}[\nabla T] &= 100
\end{aligned}$$

Since $\tau(T, k, p)$, $v(k, p)$, $\omega(k, p)$, $\mathbb{E}[T]$, and $\mathbb{E}[\nabla T]$ are all known beforehand, can set the following scaling factor:

$$c(k, p) = \tau v \left[\frac{df_{eq}(\mathbb{E}[\omega], \mathbb{E}[T])}{dT} * \mathbb{E}[\nabla T] \right] \quad (3.3)$$

$$L_{PDE_{p=0}} = L^2 \left(\frac{R_{PDE_{p=0}}}{c(k, 0)} \right)$$

where L^2 is the mean squared error (MSE). The expected values are approximations of the orders of magnitude and can be reasonably approximated using the minimum values, mean values, or maximum values, but are commonly chosen to be the

average and are constant throughout training. Usually, c is a constant to avoid a scenario where the network attempts to minimize L by maximizing c . Since c does not depend on any learnable parameters, this definition does not pose an issue. Furthermore, if the gradients of c need to be ignored, we can use TensorFlow's `tf.stop_gradient` function, which treats c as a constant during backpropagation. A similar procedure can be done for $p = 1$, and for the residual of the physical constraint. When matrix multiplication is used for summations, we normalize by dividing the expected value by the square root of the number of elements in the summation. Additionally, we can approximate $\mathbb{E}[\nabla^2 T]$ to be the maximum of the $\mathbb{E}[\nabla^2 T]$ on $\partial\Omega$.

Balancing loss terms in PINN training is key to stable optimization and fast convergence, which decreases the effective training time. By estimating each term's expected magnitude and scaling them to $O(1)$, we ensure equal contribution during training.

3.3.2 Network Initialization

Ideally, we want the expected value of our initialized network output to match the expected value of the final solution. This is similar to why data-driven approaches often normalize data to have a maximum value of 1. However, for PDEs, normalization is not always feasible. Instead, we "normalize" the output of our neural network by multiplying it by a scaling factor. This scaling factor should correspond to the order of the final solution, which can be determined similarly to the order of the loss function. However, determining the correct scaling factor requires a deep understanding of the mathematical

behavior of the PDE, and can be tricky, especially in multi-scale applications. For f_{eq} Net, we expect the predicted temperature profile to be on the same order as the BC, hence

$$O(f_{eq}) \sim f_{eq}(\max(T \text{ on } \partial\Omega), \mathbb{E}[\omega]) \quad (3.4)$$

where $\mathbb{E}[\omega]$ is the expected value of $\omega(k, p)$. This expected value is the average of ω over the inputs of k and p .

For f_{neq} , the magnitude of the solution is understood by taking the approximate limits of the BTE as the group velocity approaches infinity (ballistic limit) and zero (diffusive limit). In the ballistic regime, we approximate the right-hand side of the equation to be zero, which results in an approximation that the gradients of f_{eq} and f_{neq} have equal orders of magnitude.

$$\lim_{v \rightarrow \infty} \left[\vec{s}(\nabla f_{eq} + \nabla f_{neq}) = -\frac{f_{neq}}{\tau v} \right] \sim 0, \quad O(\nabla f_{eq}) \sim O(\nabla f_{neq}) \quad (3.5)$$

In the diffusive regime we assume f_{neq} to be a small perturbation in f such that

$$f_{eq} \gg f_{neq}, \quad \nabla f_{neq} \sim 0, \quad (3.6)$$

hence

$$\lim_{v \rightarrow 0} [\tau \vec{v} \cdot \nabla (f_{eq} + f_{neq}) = -f_{neq}], \quad O(\tau v \nabla f_{eq}) \sim O(f_{neq}) \quad (3.7)$$

We bring attention to the ballistic limit scaling, we are only given the order of magnitude of the derivative. This is resolved by using the isothermal boundary condition stating that f_{neq} is 0 on the boundary for some given direction input. This implies that the $O(f_{neq}) \sim L \nabla f_{neq}$ where L is the length of the system. Finally, we interpolated between these two limits by assuming the ballistic limit magnitude scaling works when $\tau v(L) \leq 1$ and otherwise we use the diffusive scaling.

3.3.3 Hard boundary constraints

To efficiently handle the isothermal boundary condition described in Eq. 6, a hard boundary constraint(HBC) scheme is utilized, also referred to as hard constraint(HC). HBC is a post-processing technique with no learnable parameters. The goal is to reduce loss terms in our objective function, speeding up convergence and reducing the needed model complexity. Additionally, since PINN is immediately initialized with the correct boundary conditions(BC), it can immediately start learning the dynamics of the BTE. This is crucial as most of the “learning” of NNs is done in the first few epochs, and the remainder is fine-tuning. In Li et al.’s original BTE paper, the 2D rectangular simulations had a total of 8 loss terms, 4 of which corresponded to boundaries; by replacing boundary loss terms with HBC, the number of loss terms is reduced by half. Furthermore, large temperature gradients are very harsh on the network and are difficult to learn, meaning that the network takes significantly “expressivity” to resolve the BCs. In general, it is not uncommon for PINN networks to only partially satisfy the BC even halfway through training.

This partial satisfaction of the BC presents a massive issue for the ill-posed nature of the BTE, which can have multiple transport regimes in a single solution. If the BC is even 10% inaccurate, the error can cascade into significant distortions of the temperature gradient and distribution functions, especially when working with nano-scaled devices. In the best-case scenario, this creates unnecessary overhead as the network continuously adjusts transport regimes as it tries to solve the BCs. In the worst case, the network becomes gridlocked at its current local-minimum solution because updating the BCs

would require large updates over the entire domain. Hence, creating an HBC post-processing layer is extremely important in ill-posed forward problems.

However, there are important requirements for designing an effective HBC. The HBC must provide a continuous interpolation between the boundary and the domain to ensure smooth solutions. Further, the nullifying function is carefully chosen to equal zero exactly at the boundary, thus ensuring that the network output satisfies the boundary condition by construction. Special care must be taken when using such functions, particularly regarding their derivatives. For instance, a simple nullifier like x^2 enforces the output to be zero at $x = 0$, but when we multiply it by an NN output, the derivative of this product will vanish as well. While this behavior might be desirable when enforcing a Neumann boundary condition, it is generally incorrect for Dirichlet conditions, where only the function value, not its derivative, should be fixed. Using such nullifiers improperly can therefore distort the solution and lead to inadequate or biased results.

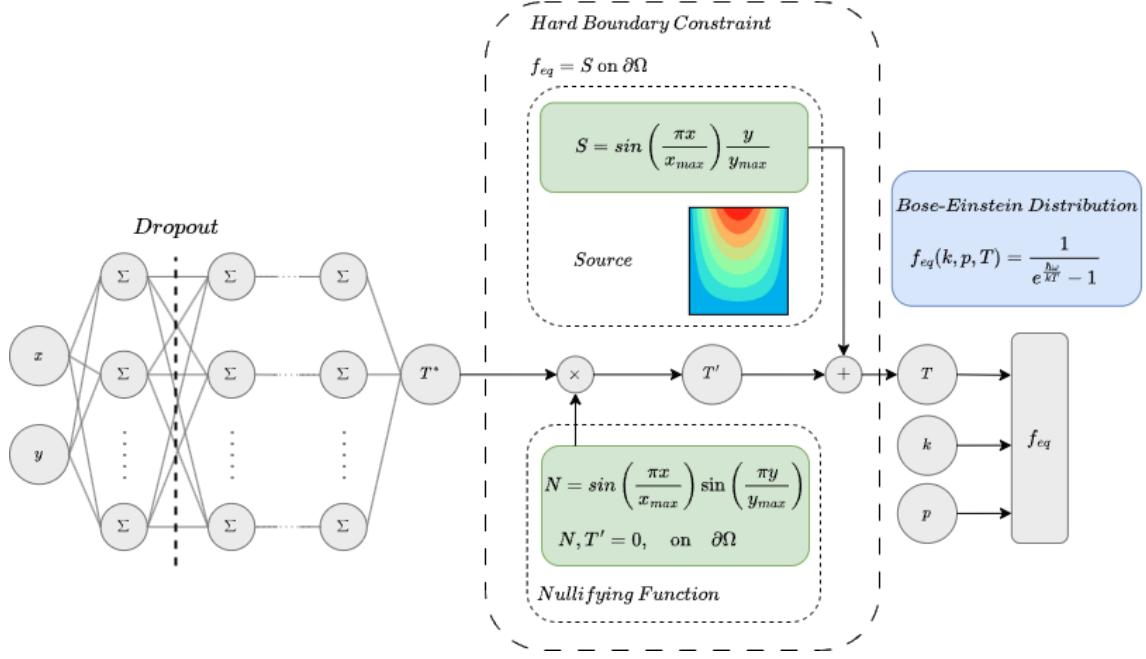


Figure 3.1: A high-level diagram of the Feq Net used in this work

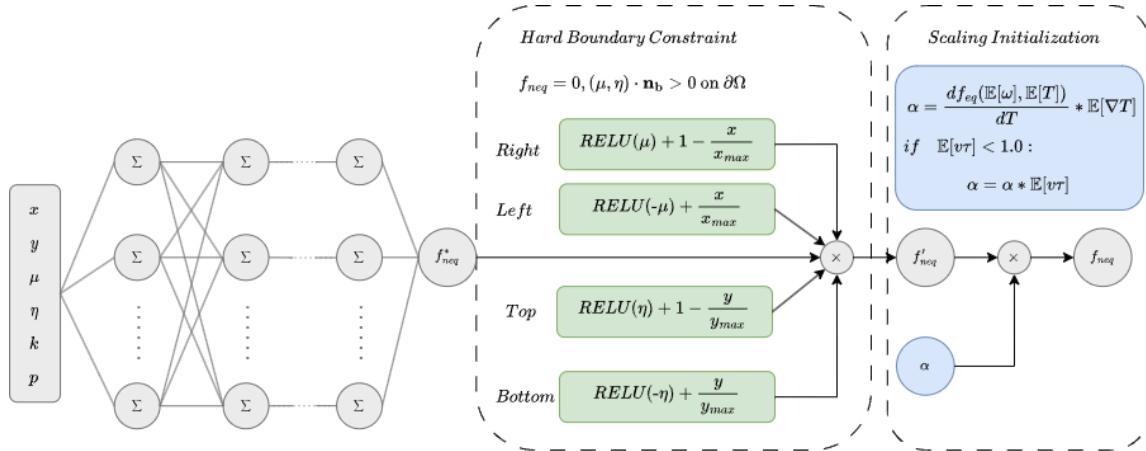


Figure 3.2: A high-level diagram of the Fneq Net used in this work

In our approach for the BTE, the HBC is constructed by taking the trainable outputs of the neural networks, T^* and f_{neq}^* , and multiplying them by a nullifying function (Fig. 3.1 and Fig. 3.2). We denote the result of this multiplication as T' in F_{eq} Net and f'_{neq} in F_{neq} Net. Since the f_{neq} isothermal boundary condition depends on

spatial and direction vectors, the isothermal boundary condition is separated into the left, right, top, and bottom boundaries. An appropriate nullifying function is selected for each boundary, and all of these functions are multiplied with f_{neq}^* .

After multiplication with the appropriate nullifying function, the resulting T' and f'_{neq} are guaranteed to be zero on the boundary throughout training. However, since we aim to enforce a specific temperature profile at the boundary, a source function S is introduced. This source function S describes the correct temperature values defined on the domain's boundary. By adding S to T' , we obtain the final predicted temperature distribution T (Fig. 3.1). In contrast, a source function is unnecessary for the Fneq Net because, under an isothermal boundary condition, the nonequilibrium distribution function f_{neq} is expected to be precisely zero at the boundaries, satisfying the condition directly through the nullification.

3.4 Architecture Overview

The high-level architecture for Feq Net and Fneq Net can be seen in Fig. 3.1 and Fig. 3.2. Both networks are designed with three hidden layers of 30 neurons each and use the Swish activation function. The first layer is initialized using Kaiming initialization, while the remaining layers use Xavier normal initialization to ensure stable training.

The architecture remains compact and efficient due to the use of HBC, which eliminates the need for explicit boundary condition loss terms. This not only reduces the computational burden associated with large, high-dimensional boundary grids but also

allows for shallow networks, significantly improving training speed. A shared mask dropout with 10% probability is applied to the Feq Net, which helps smooth the optimization path and accelerates convergence. Careful loss scaling and initialization further enhance training efficiency, enabling fast and stable learning despite the stiff problem structure.

Combining all of these optimizations and porting of the code into Tensorflow, the training time was reduced to approximately 10 minutes per simulation, compared to the 4-hour simulations in Li et al.'s work.

3.5 Results - 2D rectangle phonon transport

3.5.1 Comparison to Li et al.'s Work

The HBC-PINN architecture is applied to simulate phonon transport in a 2D rectangular domain of Silicon, where the width is set to L and the height to $3L$ (Fig. 8). Following the setup in Li et al.'s work, a heat source is applied along the top boundary, centered along the x -axis, and maintained at a high temperature T_h , while the remaining boundaries are fixed at a cooler temperature T_c . These boundary conditions are enforced using HBC. Consistent with Li et al.'s parameters, we set $T_h=400$ K and $T_c=300$ K, with all boundaries treated as isothermal. The heating source profile follows a sine function, as illustrated in Fig. 3.3.

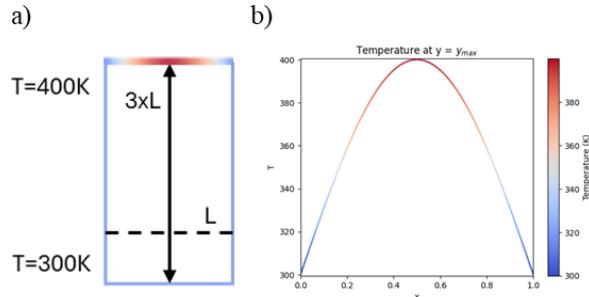


Figure 3.3: Rectangular domain geometry(a) and heating function(b)

Results are compared with Li et al.'s work for characteristic lengths of 100 nm, 1 μ m, 100 μ m, and in the diffusive limit in Fig. 3.4 and Fig. 3.5. The diffusive limit is separately modeled using a dedicated PINN trained on the steady-state heat diffusion equation $\nabla^2 T=0$, under the same HBC. The resulting temperature profiles, obtained from the Feq Net, are displayed in the figures.

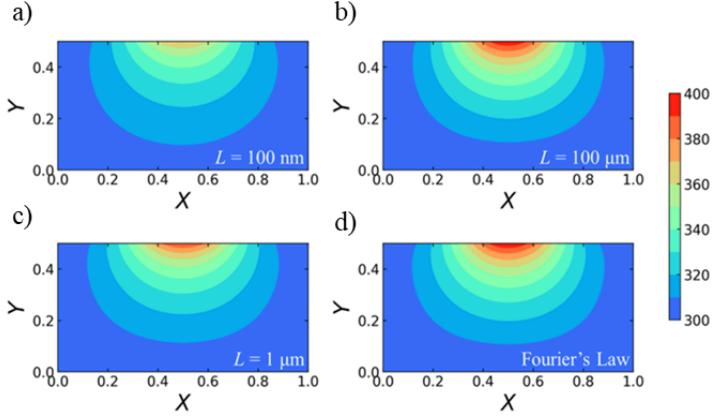


Figure 3.4: Li et al.’s work. The 2D temperature profiles for silicon at (a) $L = 100 \text{ nm}$, (c) $1 \mu\text{m}$, and (b) $100 \mu\text{m}$ obtained through the BTE. (d) The 2D temperature profile obtained from Fourier’s Law.

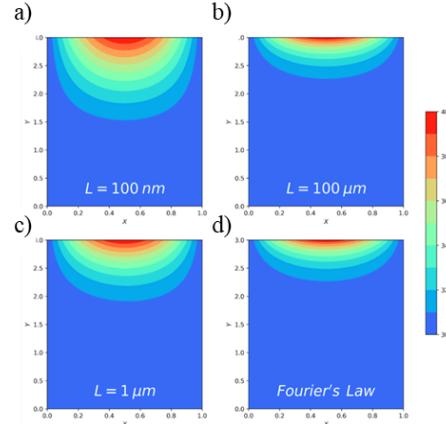


Figure 3.5: Our work. The 2D temperature profiles for silicon at (a) $L = 100 \text{ nm}$, (c) $1 \mu\text{m}$, and (b) $100 \mu\text{m}$ obtained through the BTE. (d) The 2D temperature profile obtained from Fourier’s Law.

Unlike Li et al.’s original results, where boundary conditions are not satisfied, our revised PINN maintains the peak temperature exactly at T_h across all scales. Specifically, Li et al.’s PINN exhibits boundary violations, leading to lower-than-expected peak temperatures at the heated boundary. Although no exact analytical solution is available for this configuration, the PINN solution in the diffusive limit matches the classical solution of the 2D steady-state heat equation, validating HC-PINN in the diffusive limit.

3.5.2 Length Scale Sweep Analysis

Center axis cut of Temperature Profile

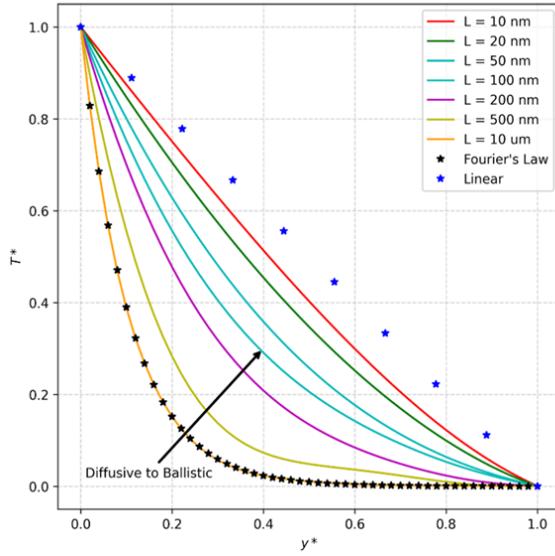


Figure 3.6: Our work. The center-axial ($x = .5L$) cut along the y -axis for the 2D temperature profiles for Silicon at multiple length scales obtained through the BTE. Where y^* is the normalized distance from the top boundary and T^* is the normalized temperature.

Finally, in Fig. 3.6, we present a centerline cut along the y -axis of the normalized temperature, T^* , where the temperature is scaled such that T_h maps to 1 and T_c maps to 0. Similarly, y^* denotes the normalized vertical distance from the top boundary. We simulate length scales ranging from 10 nm up to 10 μm . At 10 μm , the BTE solution aligns with the diffusive limit predicted by Fourier's law, confirming the model's accuracy in this regime. Additionally, we observe that as the length scale decreases, the temperature profile extends progressively deeper into the domain, reflecting stronger ballistic transport effects and approaching a linear profile.

3.6 Summary

Method	Description
Loss Scaling	Scales each loss term to the order of magnitude of 1, $L_i \sim O(1)$
Network Initialization	Sets expected magnitude separately for each phonon at (k, p)
Hard Constraints	Decreases loss terms Removes required training grid for boundaries Allows for reducing size of network

Table 3.1:The utilized methods for the phonon BTE and their description.

Building upon the methodologies established in our MR-EPT research, we focus on the two recurring strategies in the development of PINNs: loss balancing and the use of a "good first guess". These serve as fundamental principles when developing any PINN frameworks. Table 3.1 outlines the major methods used in the phonon BTE PINN framework. Notably, loss scaling is used to balance the optimization landscape between multiple different transport regimes. In addition, network initialization and hard constraints provide the model with a high-quality initial approximation and reduce the complexity associated with learning boundary conditions. Collectively, these techniques contribute to faster and more stable convergence, significantly reducing the overall computational time. Among them, hard constraints are particularly impactful, as they eliminate the need to store additional input grids for the boundary conditions and allow for a simplification of the deep neural network. As previously emphasized in the MR-EPT discussion, integrating as much prior knowledge as possible into the PINN architecture consistently yields the most successful results.

General Conclusions

This thesis demonstrates a significant improvement over the PIFON framework for EPT. While the original PIFON, using the Simplified Helmholtz, yielded reconstruction errors of $\leq 10\%$, our models achieved improved accuracy with errors of $\leq .01\%$. We further validated these improvements on a realistic 2D cylindrical phantom excited by a wire source, demonstrating the model's adaptability to practical MRI acquisition scenarios. Under data with a SNR of 200, our models achieved reconstruction errors $\leq .01\%$. We extended our validation to a 3D spherical phantom under plane-wave excitation and again achieved high-fidelity reconstructions with $\leq .01\%$ error.

These results are enabled by a new training methodology we term the “pinching effect,” designed to address challenges at material interfaces and discontinuities. This approach combines the Modified MSE loss, resampling of collocation points, and dropout layers to focus network learning on well-defined homogeneous regions before progressively resolving high-error boundaries. Furthermore, the EF-PINN and MS-DO-PINN architectures enable frequency tunability and multiscale feature extraction. This correct network initialization accelerates convergence by aligning the model’s starting point with the expected physical structure of the solution.

Building upon the baseline model by Li et al., we apply our Hard Constraint PINN (HC-PINN) formulation to solve the mode-resolved phonon Boltzmann Transport Equation (BTE) for thermal transport in silicon. Specifically, we address a 2D rectangular transport problem with isothermal boundary conditions, and compare our results to those

of Li et al.'s baseline PINN model. Our HC-PINN enforces boundary conditions exactly, incorporates proper scaling of outputs, and balances the loss function to enable efficient solving of the phonon BTE. These innovations reduce simulation time from 4 hours to just 10 minutes. Importantly, all of our models were trained on the BU Shared Computing Cluster with a NVIDIA L40S.

Additionally, we tested the HC-PINN across a wide range of silicon device length scales, from 10 μm down to 10 nm, capturing the transition from diffusive to ballistic heat transport. In the diffusive regime (10 μm), the temperature profile predicted by our model closely matches the classical Fourier-based solution. As the length scale decreases, we observe a gradual departure from this behavior, with the temperature distribution exhibiting features characteristic of the ballistic regime.

4.1 Future work

While this thesis presents substantial progress, several directions remain open for further investigation. In the biomedical domain, a key next step is the evaluation of the proposed PINN methodologies using realistic human brain models. This will provide a more accurate assessment of their potential for clinical application. Next, validating the phonon BTE PINN solution against experimental data is critical to confirm its practical viability. Once validated, this approach should be extended to handle complex geometries and multi-material systems, offering broader applicability to real-world thermal transport scenarios. Specifically, interface modeling between materials in different transport

regimes is currently extremely difficult and solving this problem with PINN can lead to massive breakthroughs in semiconductor device designs.

From a broader perspective, the "focusing effect" methodology warrants further investigation in other PDEs with interface-dependent behaviors. Exploring its generalizability across a range of PDE configurations could lead to new insights and techniques for solving problems involving sharp interfaces or discontinuous derivatives.

Finally, dropout layers have proven to be highly versatile and effective tools in our PINNs. A deeper understanding of the various ways in which dropout layers influence training and generalization could establish it as a standard, indispensable component in future PINN models.

Appendix

5.1. Helmholtz in MRI

The Helmholtz PDE describes the perturbation of the complex magnetic vector field (\mathbf{B}) caused by the complex EP(ε_c):

$$\nabla^2 B + k_0^2 \varepsilon_c B + \frac{\nabla \varepsilon_c \times \nabla \times B}{\varepsilon_c} = 0 \quad (5.1)$$

Where $B = (B_x, B_y, B_z)$ represents the magnetic flux density from the RF coil.

Further, k_0^2 is the wavenumber in vacuum and

$$\varepsilon_c = \varepsilon_r - \frac{i\sigma}{\omega \epsilon_0} \quad (5.2)$$

is the relative complex permittivity. Here, ε_c and σ represent relative permittivity and conductivity, respectively. Furthermore, i refers to the imaginary unit, ω is the angular frequency, and ϵ_0 is the permittivity of free space.

In current MRI setups, only the left-handed circularly polarized magnetic field B_1^+ can be measured, $B_1^+ = \frac{B_x + iB_y}{2}$. In the case of high-field MRI scanners (<7 T) the B_z component of the coil's magnetic field can be assumed negligible near the center regions. Equation 1 can now be rewritten as:

$$\begin{aligned} \nabla^2 B_1^+ + k_0^2 \varepsilon B_1^+ &= \left(\frac{\partial B_1^+}{\partial x} - i \frac{\partial B_1^+}{\partial y} + \frac{1}{2} \frac{\partial B_z}{\partial z} \right) (g_x + ig_y) \\ &\quad + \left(\frac{\partial B_1^+}{\partial z} - \frac{1}{2} \frac{\partial B_z}{\partial x} - i \frac{1}{2} \frac{\partial B_z}{\partial y} \right) (g_z) \end{aligned} \quad (5.3)$$

with $\mathbf{g} := (g_x, g_y, g_z) := \nabla \ln \varepsilon_c$.

5.2 Simplified Helmholtz

If we assume a piecewise constant distribution of EP ($\nabla \varepsilon_c = 0$, $g = 0$), equation 3 can be reduced to the Simplified Helmholtz:

$$\nabla^2 B_1^+ + k_0^2 \varepsilon_c B_1^+ = 0 \quad (5.4)$$

Typical solvers will use a Savitzky-Golay filter to smooth out B_1^+ and then compute the Laplacian of B_1^+ with a finite differences approach. In the case of PINN, we use automatic differentiation on the network to solve directly for the derivatives of B_1^+ .

5.3 Convection-Reaction EPT

If we let $\gamma = \frac{1}{\varepsilon_c}$ and we assume Bz to be negligible we get:

$$\nabla^2 B_1^+ \cdot \gamma + k_0^2 B_1^+ = -(\frac{\partial B_1^+}{\partial x} - i \frac{\partial B_1^+}{\partial y})(\frac{\partial \gamma}{\partial x} + i \frac{\partial \gamma}{\partial y}) - \frac{\partial B_1^+}{\partial z} \cdot \frac{\partial \gamma}{\partial z} \quad (5.5)$$

This convection reaction equation (5.5) is typically solved with a mesh-based finite difference scheme for γ . However, the resulting EP distributions suffer from numerical errors, such as global spurious oscillations, caused by mesh density for discretization and inherent mathematical limitations of cr-MREPT.

5.4 Phonon Boltzmann Transport Equation

The 2D steady-state phonon BTE under the single-mode relaxation time approximation[42] can be written as:

$$\vec{v} \cdot \nabla(f_{eq} + f_{neq}) = \frac{-f_{neq}(T)}{\tau(T)}, f = f_{eq} + f_{neq} \quad (5.6)$$

where $f = f(\vec{x}, \vec{s}, k, p)$ is the phonon distribution function with inputs of: spatial vector $\vec{x} = (x, y)$, directional unit vector $\vec{s} = (\mu, \eta)$, wave number k , and polarization p . The vector $\vec{v} = v\vec{s}$, where v is the phonon group velocity, assuming isotropic velocity. The temperature-dependent $\tau(T, k, p)$ is the relaxation time. The phonon distribution function can be divided into f_{eq} and f_{neq} , where f_{eq} is the equilibrium distribution governed by the Bose-Einstein distribution,

$$f^{eq}(T, k, p) = \frac{1}{e^{\frac{\hbar\omega(k, p)}{k_B T}} - 1} \quad (5.7)$$

where \hbar is the reduced Planck's constant, k_B is the Boltzmann constant, and ω is the frequency and is a function of k and p . Finally, $f_{neq}(\vec{x}, \vec{s}, k, p)$ is the non-equilibrium distribution.

In the case of no internal heat source, we have a physical constraint that the divergence of the heat flux q must be zero, which can be obtained by integrating the energy-based form of Eq. (3.1) over the solid angle space (Ω) and the frequency space (ω, p)

$$\vec{q} = \sum_p \int_0^{\omega_{\max,p}} \int_{4\pi} \vec{v} \hbar \omega Df d\Omega d\omega \quad (5.8)$$

$$\nabla \cdot \vec{q} = 0$$

where $D(\omega, p)$ is the phonon density of states and $\omega_{\max,p}$ is the maximum frequency.

We can express equation 4 as the following:

$$\nabla \cdot \vec{q} = \sum_p \int_0^{\omega_{\max,p}} \int_{4\pi} \vec{v} \hbar \omega D(\tau(T)[\vec{v} \cdot \nabla^2(f_{eq} + f_{neq})]) d\Omega d\omega \quad (5.9)$$

We use a 2D rectangular domain for our simulations. The boundaries of our domain are isothermal, which are defined as follows:

$$f = f_{eq}, (\mu, \eta) \cdot \mathbf{n}_b > 0 \text{ on } \partial\Omega \quad (5.10)$$

where \mathbf{n}_b is the normal unit vector pointing into the simulation domain.

BIBLIOGRAPHY

- [1] M. Raissi, et al. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." *Journal of Computational physics* 378 (2019): 686-707.
- [2] G. E. Karniadakis, et al. "Physics-informed machine learning." *Nature Reviews Physics* 3.6 (2021): 422-440.
- [3] S. Qi, et al. "Two-dimensional electromagnetic solver based on deep learning technique." *IEEE Journal on Multiscale and Multiphysics Computational Techniques* 5 (2020): 83-88.
- [4] L. Lu, et al. "Physics-informed neural networks with hard constraints for inverse design." *SIAM Journal on Scientific Computing* 43.6 (2021): B1105-B1132.
- [5] Y. Chen, et al. "Physics-informed neural networks for inverse problems in nano-optics and metamaterials." *Optics express* 28.8 (2020): 11618-11633.
- [6] Y. Chen, and Luca Dal Negro. "Physics-informed neural networks for imaging and parameter retrieval of photonic nanostructures from near-field data." *APL Photonics* 7.1 (2022).
- [7] R. Riganti, and L. Dal Negro. "Auxiliary physics-informed neural networks for forward, inverse, and coupled radiative transfer problems." *Applied Physics Letters* 123.17 (2023).
- [8] R. Riganti, et al. "Multiscale Physics-Informed Neural Networks for the Inverse Design of Hyperuniform Optical Materials." *arXiv preprint arXiv:2405.07878* (2024).
- [9] J.D. Toscano, et al. "From pinns to pikans: Recent advances in physics-informed machine learning." *Machine Learning for Computational Science and Engineering* 1.1 (2025): 1-43.
- [10] S. Cai, et al. "Physics-informed neural networks enhanced particle tracking velocimetry: An example for turbulent jet flow." *IEEE Transactions on Instrumentation and Measurement* (2024).
- [11] S. Cai, et al. "Physics-informed neural networks for heat transfer problems." *Journal of Heat Transfer* 143.6 (2021): 060801.

- [12] H. J. Ling, et al. "Physics-guided neural networks for intraventricular vector flow mapping." *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* (2024).
- [13] Z. Xu, et al. "Hallucination is inevitable: An innate limitation of large language models." *arXiv preprint arXiv:2401.11817* (2024).
- [14] C. M. Collins, and Z. Wang. "Calculation of radiofrequency electromagnetic fields and their effects in MRI of human subjects." *Magnetic resonance in medicine* 65.5 (2011): 1470-1482.
- [15] J. W. Hand, "Modelling the interaction of electromagnetic fields (10 MHz–10 GHz) with the human body: methods and applications." *Physics in Medicine & Biology* 53.16 (2008): R243.
- [16] K. K. Tha, et al. "Noninvasive evaluation of electrical conductivity of the normal brain and brain tumors." *Proc. Int. Soc. Magn. Reson. Med.* Vol. 22. 2014.
- [17] K. K. Tha, et al. "Noninvasive electrical conductivity measurement by MRI: a test of its validity and the electrical conductivity characteristics of glioma." *European radiology* 28 (2018): 348-355.
- [18] E. Balidemaj, et al. "Feasibility of electric property tomography of pelvic tumors at 3T." *Magnetic resonance in medicine* 73.4 (2015): 1505-1513.
- [19] J. Shin, et al. "Initial study on in vivo conductivity mapping of breast cancer using MRI." *Journal of Magnetic Resonance Imaging* 42.2 (2015): 371-378.
- [20] C. Rossmann and D. Haemmerich. "Review of temperature dependence of thermal properties, dielectric properties, and perfusion of biological tissues at hyperthermic and ablation temperatures." *Critical Reviews™ in Biomedical Engineering* 42.6 (2014).
- [21] E. Balidemaj, et al. "Hyperthermia treatment planning for cervical cancer patients based on electrical conductivity tissue properties acquired in vivo with EPT at 3 T MRI." *International Journal of Hyperthermia* 32.5 (2016): 558-568.
- [22] M. M.de Oliveira, et al. "Bio-heat transfer model of electroconvulsive therapy: Effect of biological properties on induced temperature variation." *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016.
- [23] U. Katscher, et al. "Determination of electric conductivity and local SAR via B1 mapping." *IEEE transactions on medical imaging* 28.9 (2009): 1365-1374.

- [24] H. Wen. "Noninvasive quantitative mapping of conductivity and dielectric distributions using RF wave propagation effects in high-field MRI." *Medical Imaging 2003: Physics of Medical Imaging*. Vol. 5030. SPIE, 2003.
- [25] G. Ruan, et al. "Magnetic resonance electrical properties tomography based on modified physics-informed neural network and multi constraints." *IEEE Transactions on Medical Imaging* (2024).
- [26] S. Mandija, et al. "Opening a new window on MR-based electrical properties tomography with deep learning." *Scientific reports* 9.1 (2019): 8895.
- [27] N. Hampe, et al. "Investigating the challenges and generalizability of deep learning brain conductivity mapping." *Physics in Medicine & Biology* 65.13 (2020): 135001.
- [28] S. Gavazzi, et al. "Deep learning-based reconstruction of in vivo pelvis conductivity with a 3D patch-based convolutional neural network trained on simulated MR data." *Magnetic resonance in medicine* 84.5 (2020): 2772-2787.
- [29] I. Giannakopoulos, et al. "On the usage of deep neural networks as a tensor-to-tensor translation between mr measurements and electrical properties." *Proc. ISMRM*. 2020.
- [30] X. Yu, et al. "PIFON-EPT: Mr-based electrical property tomography using physics-informed fourier networks." *IEEE journal on multiscale and multiphysics computational techniques* 9 (2023): 49-60.
- [31] Z. Q. J. Xu, et al. "Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks." *Communications in Computational Physics* 28.5 (2020): 1746-1767.
- [32] N. Rahaman, et al. "On the spectral bias of neural networks." *International conference on machine learning*. PMLR, 2019.
- [33] J. C. Wong, et al. "Learning in sinusoidal spaces with physics-informed neural networks." *IEEE Transactions on Artificial Intelligence* 5.3 (2022): 985-1000.
- [34] B. Ronen, et al. "The convergence rate of neural networks for learned functions of different frequencies." *Advances in Neural Information Processing Systems* 32 (2019).
- [35] S. Wang, et al. "On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks." *Computer Methods in Applied Mechanics and Engineering* 384 (2021): 113938.
- [36] A. Rahimi, and B. Recht. "Random features for large-scale kernel machines." *Advances in neural information processing systems* 20 (2007).

- [37] M. Tancik, et al. "Fourier features let networks learn high frequency functions in low dimensional domains." *Advances in neural information processing systems* 33 (2020): 7537-7547.
- [38] Z. G. Chen, et al. "Nanostructured thermoelectric materials: Current research and future challenge." *Progress in Natural Science: Materials International* 22.6 (2012): 535-549.
- [39] M. Zebarjadi, et al. "Perspectives on thermoelectrics: from fundamentals to device applications." *Energy & Environmental Science* 5.1 (2012): 5147-5162.
- [40] E. Pop, et al. "Heat generation and transport in nanometer-scale transistors." *Proceedings of the IEEE* 94.8 (2006): 1587-1601.
- [41] A. L. Moore, and L. Shi. "Emerging challenges and materials for thermal management of electronics." *Materials today* 17.4 (2014): 163-174.
- [42] G. Chen. *Nanoscale energy transport and conversion: a parallel treatment of electrons, molecules, phonons, and photons*. Oxford university press, 2005.
- [43] Z. M. Zhang, et al. *Nano/microscale heat transfer*. Vol. 410. New York: McGraw-Hill, 2007.
- [44] M. G. Holland, "Analysis of lattice thermal conductivity." *Physical review* 132.6 (1963): 2461.
- [45] D. Terris, et al. "Modeling semiconductor nanostructures thermal properties: The dispersion role." *Journal of Applied Physics* 105.7 (2009).
- [46] J. P. M. Péraud, and N. G. Hadjiconstantinou. "Efficient simulation of multidimensional phonon transport using energy-based variance-reduced Monte Carlo formulations." *Physical Review B—Condensed Matter and Materials Physics* 84.20 (2011): 205331.
- [47] Y. Hu, et al. "Ultra-efficient and parameter-free computation of submicron thermal transport with phonon Boltzmann transport equation." *Fundamental Research* 4.4 (2024): 907-915.
- [48] C. Hua, and A. J. Minnich. "Semi-analytical solution to the frequency-dependent Boltzmann transport equation for cross-plane heat conduction in thin films." *Journal of Applied Physics* 117.17 (2015).
- [49] X. P. Luo, and H. L. Yi. "A discrete unified gas kinetic scheme for the phonon Boltzmann transport equation accounting for phonon dispersion and polarization." *International Journal of Heat and Mass Transfer* 114 (2017): 970-980.

- [50] C. Zhang, et al. "An implicit kinetic scheme for multiscale heat transfer problem accounting for phonon dispersion and polarization." *International Journal of Heat and Mass Transfer* 130 (2019): 1366-1376.
- [51] A. J. Minnich, et al. "Quasiballistic heat transfer studied using the frequency-dependent Boltzmann transport equation." *Physical Review B—Condensed Matter and Materials Physics* 84.23 (2011): 235207.
- [52] J. M. Loy, et al. "A fast hybrid Fourier–Boltzmann transport equation solver for nongray phonon transport." *Journal of heat transfer* 135.1 (2013): 011008.
- [53] S. Mazumder, and A. Majumdar. "Monte Carlo study of phonon transport in solid thin films including dispersion and polarization." *J. Heat Transfer* 123.4 (2001): 749-759.
- [54] D. Lacroix, et al. "Monte Carlo transient phonon transport in silicon and germanium at nanoscales." *Physical Review B—Condensed Matter and Materials Physics* 72.6 (2005): 064305.
- [55] Q. Hao, et al.. "Frequency-dependent Monte Carlo simulations of phonon transport in two-dimensional porous silicon with aligned pores." *Journal of Applied Physics* 106.11 (2009).
- [56] C. Ni, and J. Y. Murthy. "Parallel computation of the phonon Boltzmann transport equation." *Numerical Heat Transfer, Part B: Fundamentals* 55.6 (2009): 435-456.
- [57] S. F. Ali, et al. "Large-scale parallel computation of the phonon Boltzmann transport equation." *International journal of thermal sciences* 86 (2014): 341-351.
- [58] S. V. J. Narumanchi, et al. "Comparison of different phonon transport models for predicting heat conduction in silicon-on-insulator transistors." (2005): 713-723.
- [59] J. M. Loy, et al. "A coupled ordinates method for convergence acceleration of the phonon Boltzmann transport equation." *Journal of Heat Transfer* 137.1 (2015): 012402.
- [60] R. Li, et al. "Physics-informed neural networks for solving multiscale mode-resolved phonon Boltzmann transport equation." *Materials Today Physics* 19 (2021): 100429.
- [61] R. Li, et al. "Physics-informed deep learning for solving phonon Boltzmann transport equation with large temperature non-equilibrium." *npj Computational Materials* 8.1 (2022): 29.
- [62] N. Srivastava, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

- [63] P. Ramachandran, et al. "Searching for activation functions." arXiv preprint arXiv:1710.05941 (2017).
- [64] Z. You, et al. "MscaleFNO: Multi-scale Fourier Neural Operator Learning for Oscillatory Function Spaces." arXiv preprint arXiv:2412.20183 (2024).
- [65] Y. Wang, et al. "A practical PINN framework for multi-scale problems with multi-magnitude loss terms." Journal of Computational Physics 510 (2024): 113112.