ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

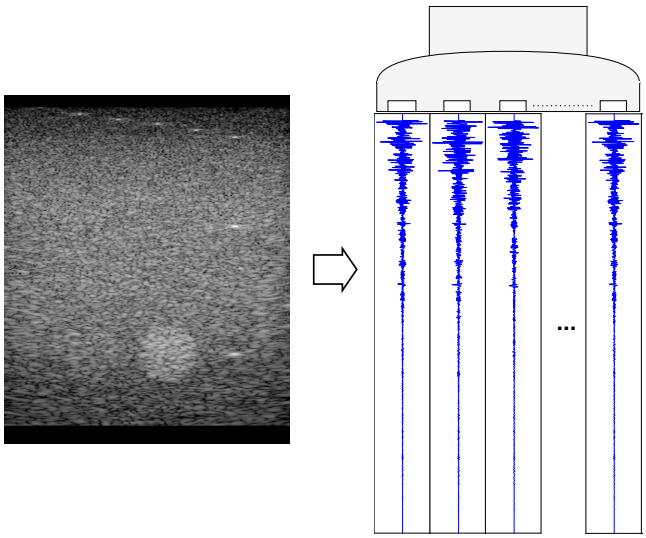
Institut für Integrierte Systeme
Integrated Systems Laboratory

Department of Information Technology and
Electrical Engineering

Autumn Semester 2023

Ultrasound Image Recycling: Predicting Raw Data with Simulation-Driven Deep Learning

Master Project



Maurits Reitsma
mreitsma@student.ethz.ch

April 2024

Supervisors: Dr. Christoph Leitner, christoph.leitner@iis.ee.ethz.ch
Dr. Yawei Li, yawei.li@vision.ee.ethz.ch

Professor: Prof. Dr. Luca Benini, luca.benini@iis.ee.ethz.ch
Prof. Dr. Luc van Gool, vangool@vision.ee.ethz.ch

Acknowledgements

I sincerely thank my supervisors Christoph Leitner and Yawei Li for their guidance during the project and for always providing valuable input. I also want to thank my family and friends for supporting me throughout the project. Most of all I want to thank Felizitas for her daily support and for always believing in me.

Abstract

Medical ultrasound (US) imaging is a vital diagnostic tool and has many areas of application. The raw data from US imaging, known as radio-frequency (RF) data, contains more information than US images and has valuable use cases. Although large datasets of processed US images are widely available, raw RF data remains scarce.
This thesis introduces the first system designed to predict raw RF data from US images. We have developed a data-driven, physically-informed model that combines deep learning (DL) techniques with numerical simulations of ultrasound waves. Specifically, we employed a TransUNet neural network architecture to predict the acoustic properties of the underlying tissue. The neural network is trained using a custom synthetic dataset generated by us. To create extensive and meaningful synthetic datasets of paired RF data and US images, we propose four different strategies. These include using geometric features from standard images in the ImageNet dataset and information from medical CT images.
The performance of our model using different synthetic datasets is compared and evaluated for real RF data and we find that the model which generalizes best is trained on the CT phantom dataset. Additionally, we explore fine-tuning the trained models to predict RF data for medial gastrocnemius (MG) muscle images using a dedicated synthetic muscle phantom dataset created by us. We verify that fine-tuning the model for prior-known tissue types significantly improves the predicted RF data. By comparing our physics-based model to an end-to-end DL model trained for RF data prediction, the benefit of introducing the numerical simulation into our predictions is highlighted.

Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor. For a detailed version of the declaration of originality, please refer to Appendix B

Maurits Reitsma,
Zurich, April 2024

Contents

List of Figures

List of Tables

List of Acronyms

CC . . . . . .Cross-correlation

CDF . . . . . .Cumulative Distribution Function

CIRS . . . . . .Computerized Imaging Reference Systems

CIRS GP . . . .CIRS Model 054GS General Purpose Ultrasound Phantom

CNR . . . . . .Contrast-to-noise Ratio

COCO . . . . . .Common Objects in Context

CT . . . . . .Computed Tomography

CUDA . . . . . .Compute Unified Device Architecture

DAS . . . . . .Delay and Sum

DICOM . . . . .Digital Imaging and Communications in Medicine

DL . . . . . .Deep Learning

FFT . . . . . .Fast Fourier Transform

FWHM . . . . .Full Width at Half Maximum

GP . . . . . .Gaussian Process

GPU . . . . . .Graphics processing unit

HDF . . . . . .Hierarchical Data Format

HU . . . . . .Hounsfield Unit

IoU . . . . . Intersection Over Union

IT'IS . . . . . Information Technologies in Society

LG . . . . . Lateral Gastrocnemius

MAE . . . . . Mean Absolute Error

MG . . . . . Medial Gastrocnemius

MRI . . . . . Magnetic Resonance Imaging

MSE . . . . . Mean Square Error

MTJ . . . . . Muscle Tendon Junction

NLP . . . . . Natural Language Processing

NMDID . . . . . New Mexico Decedent Image Database

NN . . . . . Neural Network

PSD . . . . . Power Spectral density

PW . . . . . Plane Wave

RF . . . . . Radio Frequency

RMS . . . . . Root Mean Square

RMSE . . . . . Root Mean Square Error

SAM . . . . . Segment Anything Model

SNR . . . . . Signal-to-noise Ratio

SoS . . . . . Speed Of Sound

SSIM . . . . . Structural Similarity Index Measure

TGC . . . . . Time Gain Compensation

US . . . . . Ultrasound

Chapter 1

Introduction

1.1. Motivation

Medical ultrasound (US) has been a cornerstone of diagnostic imaging for decades, due to its noninvasive nature. Unlike CT scans, which use ionizing radiation, US imaging provides a safer alternative that avoids radiation exposure. Additionally, ultrasound stands out for its cost-effectiveness and ease of use. In comparison to the more expensive MRI and CT scanners, ultrasound devices are not only more affordable but also highly portable, which increases their utility in various clinical and emergency settings. [11]

During the US acquisition raw US data, also called radio-frequency (RF) data, is collected and then processed to construct a US image. For data analysis, there are multiple advantages that RF data provides over traditional US images. Firstly, the RF data contains more information than the processed US image, since it is available at a higher resolution than the image. Additionally, it is not affected by post-processing steps such as log-compression, which is used to construct a visually more appealing image. Due to the enriched information present in the RF data, it has been used for spatial alignment of ultrasound data by the work of Klein [12].
Recent studies have developed wearable US sensors and used them to measure muscle activity, such as the system by Frey *et al.* [13]. An example of such research is the study conducted by Vostrikov *et al.* [14], where they use the collected RF data to estimate the pennation angles of muscles using machine learning (ML) algorithms directly on the edge device. Another work of Vostrikov *et al.* [15] designed a data-driven ML model to perform hand gesture recognition from RF data collected using a wearable US sensor. These research projects are interested in directly processing the RF data, since the image formation step takes up both computational and memory resources, that are not available on edge devices.

The training of such ML algorithms would benefit from large RF data sets. However, RF data is scarce, as medical ultrasound acquisition systems are often proprietary and closed, and manufacturers offer no access to the originally acquired RF data. Meanwhile, there are large datasets of medical US images freely available [16] [17]. Instead of collecting large new datasets for RF data, it would be useful to be able to "recycle" existing US image datasets.

Therefore, in this thesis, we want to establish a method to predict the original raw RF data from existing US images. To the best of our knowledge, no previous methods have attempted to predict raw RF data from images.

1.2. Objective

Our main objective is to develop a method to predict the raw RF data given a US image. As the US image is constructed from the RF data using non-linear transformation, where part of the information is lost due to processing steps such as envelope detection and log-compression, there exists no closed-form solution to this problem. To predict the RF data given the US image, we present a data-driven physically-informed model that can be separated into two main parts:

1. A numerical simulation used to simulate a US acquisition for the predicted imaging medium. This simulation computes the desired RF data based on the physics of US and therefore forms the physics-informed part of our system.
2. A deep learning (DL) model that predicts the underlying acoustic properties of the imaging medium, which we refer to as the phantom maps, given a US image. This is the data-driven part of our system.

Utilizing the numerical simulation ensures the generation of realistic RF data that aligns with the fundamental principles of physics. This would not be guaranteed when attempting to directly predict RF data using a DL model. For the implementation of the numerical simulation, the k-Wave toolbox [18] is used.

To train a DL model to predict the medium phantom maps, a large dataset of matching US images and phantom maps is needed, which was not previously available. Therefore, we introduce a framework to create large synthetic datasets consisting of paired phantom maps, US images, and RF signals. We present five different strategies to create meaningful phantoms, aiming to generalize beyond the synthetic dataset:

1. The Ellipse phantom dataset, where phantoms are created using geometric primitives.

2.-3. The ImageNet and ImageNet-enhanced phantom dataset, where phantoms are created using geometric structures extracted from general-purpose images in the ImageNet dataset. The geometric structures are extracted using segmentation and further enhanced using the edge information contained in the images.

4. The CT phantom dataset, where phantoms are created using information contained in medical CT images.

5. The Muscle phantom dataset, where phantoms are created using medical US images of the medial gastrocnemius (MG) muscle.

Using these five strategies, a total of 210'000 synthetic data samples are generated for this project. For each of the first 4 datasets, a DL model is trained from scratch.
For the DL model, two different model architectures, the UNet[19] and TransUNet[8], are explored and their prediction performance is compared.

We evaluate the performance of our physically-informed model for each of the synthetic datasets on real US images and compare the real and predicted RF data using custom metrics we introduce.
The muscle phantom dataset is used to fine-tune the previously trained models for the task of predicting RF data for US images of MG muscles. Like this, we can investigate, if the model's predictions can benefit from a dedicated synthetic dataset if the target tissue is known in advance.

Lastly, an end-to-end DL model is trained to predict the RF data directly from the US image. The end-to-end model's performance is compared to our physically-informed model. The goal is to provide a comparison and to evaluate the potential value of integrating the numerical simulation into our system.

Chapter 2

Preliminaries - Ultrasound Imaging

This thesis concerns itself with ultrasound (US) imaging and how to extract the original measured radio-frequency data (RF data) from a US image. Because the modeling of the acoustic properties of tissue and the simulation of US data is a large focus, we first explain how US data is acquired and what physical principles the imaging technique is built on.

Basic Principle of Ultrasound

A medical US image gives a cross-sectional view of tissue and organ boundaries for a target region. The US image is generated by sending an ultrasound wave through a tissue of interest using a transducer. This causes echoes that arise as ultrasound waves reflect from tissue borders and scatter from slight anomalies within the tissues. Using the pulse-echo principle, which is the same principle used for sonar, one can reconstruct the exact locations of the tissue borders with respect to the transducer and generate the resulting US image. [1]

2.1. Physics of Sound Waves

An ultrasound wave is defined as a high-frequency sound wave with a frequency higher than 20kHz, out of the range of audible frequencies [20].

Medical ultrasound sound waves are longitudinal waves that propagate through a physical medium, which is typically some form of tissue or liquid. Particles oscillate along the direction of travel and cause regions of high pressure (compression) and low pressure (rarefaction), which is visualized in Figure 2.1. As per usual for waves, the particles don't propagate, only the disturbance and its associated energy do. [1]

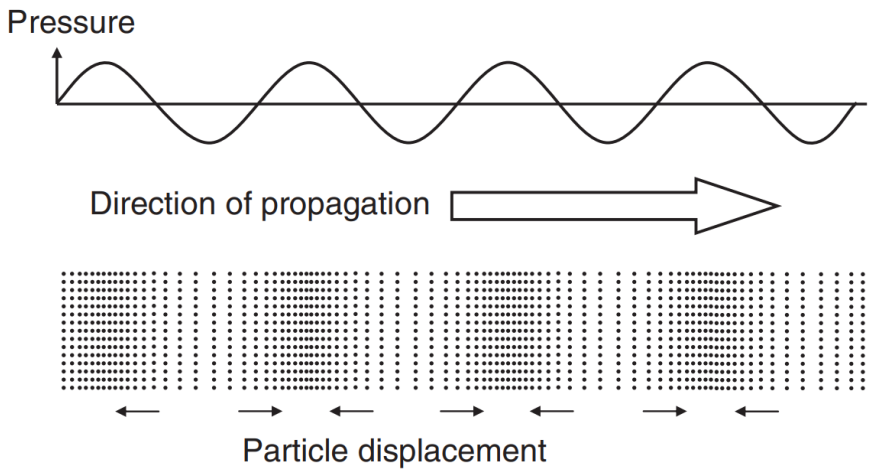


Figure 2.1.: The visualization of a sound wave, showing the particle displacement in the direction of propagation. [1]

The propagation of a sound wave depends on the medium it is traveling in [3]. The frequency of the wave f is independent of the propagation medium and stays constant as the sound wave crosses from one medium to another and is determined by the excitation frequency of the transducer which usually ranges between 2-15 MHz. However the speed of sound (SoS) c of a sound wave changes across different media and is fully determined by the current propagation medium. The properties of the medium that determine the acoustic SoS are mass density ρ and stiffness k. The latter is a measure of how much a material resists being deformed under pressure. The relationship between them is captured with the following formula:

$$c = \sqrt{\frac{k}{\rho}} \tag{2.1}$$

The wavelength λ is determined by the frequency f and SoS c as usual for waves with the following equation:

$$\lambda = \frac{c}{f} \tag{2.2}$$

The wavelength of the ultrasound pulse determines the ability of a US imaging system to resolve fine anatomical details. It directly impacts the axial resolution of the US system.

2.1.1. Interactions with Matter

The signal the transducer receives at the transducer is the result of the echoes originating from the interactions of the ultrasound waves with the propagation medium. To better

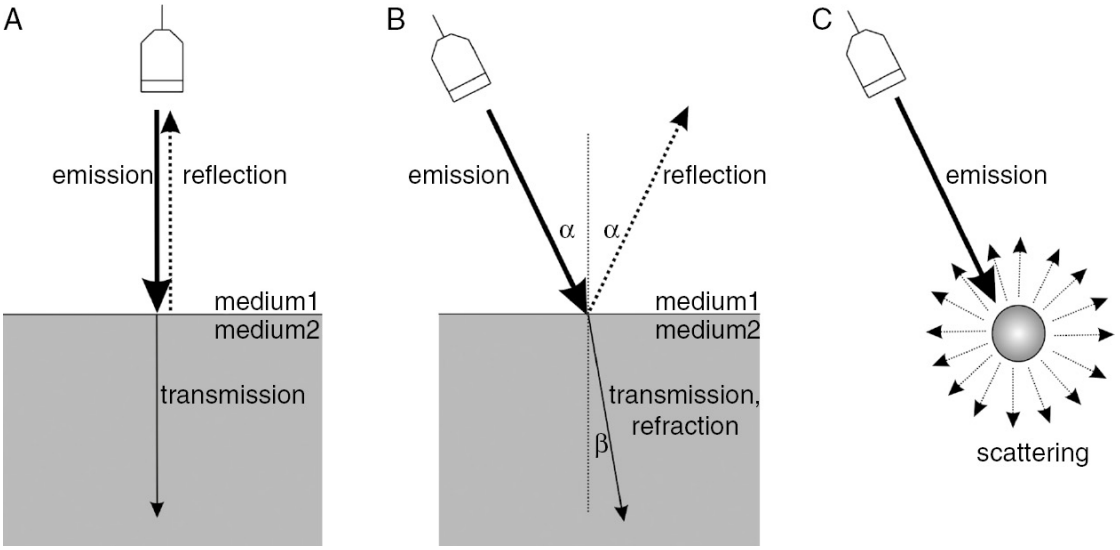


Figure 2.2.: The different types of interactions of sound waves with matter. (A) Reflection (B) Reflection and Refraction (C) Scattering. [2]

understand the echoes we receive at our transducer this section elaborates on the different interactions of sound waves with the propagation medium.

Reflection and Refraction

Reflection and refraction of US waves occur at boundaries, where the acoustic impedance of the medium changes [2]. The acoustic impedance Z of a medium represents the resistance to the propagation of ultrasound waves through a medium and it is defined as the product of the medium density ρ and the SoS c of the medium:

$$Z = \rho \cdot c \tag{2.3}$$

Reflection of ultrasound waves occurs when an ultrasound wave traveling through one type of tissue encounters an interface with another tissue of different acoustic impedance. Specifically, if the US wave travels from medium 1 to medium 2, the reflected pressure amplitude can be computed with [1]:

$$\frac{p_r}{p_i} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.4}$$

where p_r and p_i reference the reflected and incident pressure amplitude and Z_1 and Z_2 are the acoustic impedances of the two media. This assumes that the incident wave arrives in the normal direction of the tissue boundary (see Figure 2.2 (A)).

If the incident wave arrives at an angle of incidence $theta_i$ (denoted as α in Figure 2.2 (B)) then by the law of reflection the angle of reflection $\theta_r = \theta_i = \alpha$ for a smooth interface. If the SoS changes between the two media, analogously to the, due to refraction the direction of the transmitted wave changes. The angle of the transmitted wave can be computed, analogously as for electromagnetic waves, using Snell's law:

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{c_1}{c_2} \tag{2.5}$$

Scattering

The reflection laws described above hold only for large interfaces, for example at boundaries between different organs. Within most tissues, there are also many small-scale variations of the acoustic impedance. These small structures we refer to as scatterers and for them the reflections do not follow the previously described laws of reflection. For targets smaller than the wavelength of the incident ultrasound wave, it is scattered across a large range of angles. Formally for targets with a much smaller size d than the wavelength λ the scattered power is related to the frequency.

$$W_s \propto \frac{d^6}{\lambda^4} \propto d^6 f^4 \tag{2.6}$$

This relation is called Rayleigh scattering. [1]

The scattering makes up most of the echoes that are received by a transducer and the scattering is what gives different tissues their contrasts. [21]

Attenuation

As the ultrasound wave travels through the medium, the amplitude and intensity decreases. This is caused by both scattering and absorption. Both mechanisms are dependent on the frequency. The attenuation of a medium can be described using the attenuation coefficient α. The attenuation coefficent α is given in units $[\frac{\text{dB}}{\text{cm}\cdot\text{MHz}}]$. The attenuation for a signal of frequency f at depth z can therefore be calculated using

$$\text{Attenuation}\,[\text{dB}] = \alpha \left[\frac{\text{dB}}{\text{cm}\cdot\text{MHz}}\right] \cdot f\,[\text{MHz}] \cdot z\,[\text{cm}]. \tag{2.7}$$

2.2. Transducers

The device that generates the ultrasound wave and then records the echo is called the transducer. Medical transducers usually consist of multiple transducer elements, where each transducer element converts an electrical signal into a sound wave and then also

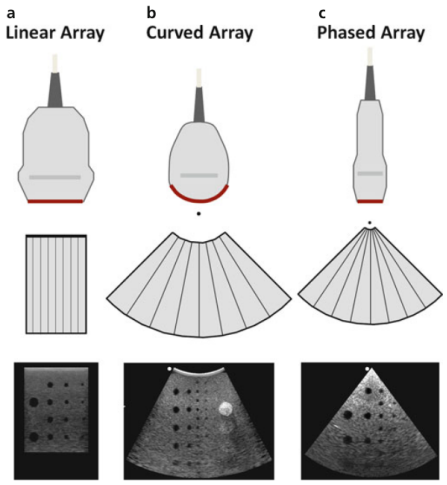


Figure 2.3.: Three different transducer types. [3]

converts the returning ultrasound echo to an electrical signal again. This electrical signal generated from the returning sound waves is the RF data we measure which is subsequently used to generate the ultrasound image. [3]

Transducers make use of the piezoelectric effect to convert the electrical signals to sound waves and the reverse. The piezoelectric effect describes the generation of an electric potential out of mechanical stress on a piezo-crystal. Vice versa, an applied electric potential produces a mechanical deformation on the piezo-crystal. By applying an oscillating electric potential to the crystal we can make it expand and contract to generate a sound wave of a desired frequency. Inversely it can also be used to sense the reflected echo and measure the incoming sound waves [22].

A medical transducer usually consists of between 128 and 256 transducer elements. Each transducer element has a piezo-element to generate a sound wave and receive a signal. For each transducer element, there is a corresponding channel in the captured RF signals. The resonance frequency of the transducer is also referred to as the center frequency and typically corresponds to the frequency at which the ultrasound wave pulse is emitted by the transducer.

2.2.1. Transducer Types

There is a variety of different types of transducers that have different applications. Three such examples can be seen in Figure 2.3.

Depending on how the transducer elements are arranged you get different transducer types. For the linear array (Figure 2.3 (A), there are usually between 128 and 192 transducer elements arranged equidistantly on a straight line, and the parallel form a

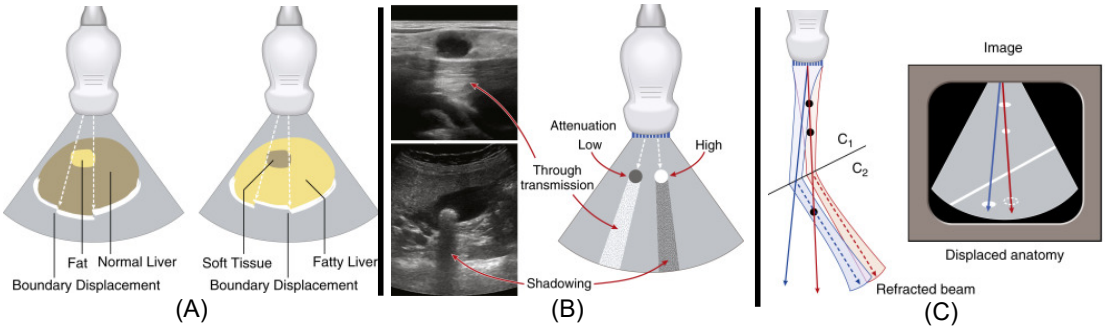


Figure 2.4.: Visualization of three different US imaging artefacts. (A) visualizes the speed displacement artefact, (B) shows the Shadowing artefact, and (C) shows the artefact caused by refraction. [4]

rectangular image. In (B), the transducer elements in the curved array are arranged in a curved configuration, leading to an image with an expanded field of view (FOV). On the other hand, a phased array (C) features elements arranged in a straight line but with a notably smaller operating aperture compared to a linear transducer, resulting in an image shaped like a circle segment.

2.3. Ultrasound Imaging Modes

There are multiple different modes for US acquisitions. The simplest one is called A-mode imaging, where the A stands for amplitude. For A-mode imaging, a single focused beam is sent and echoes are received on a single channel and the height of the reflected amplitudes are plotted.
When we talk about US images we usually refer to B-mode images, where the B stands for brightness, where the raw US data is collected for many channels and formed into a US image. This is the imaging mode that this thesis focuses on.
Lastly, there is also M-mode imaging, where the M stands for motion. In motion mode, a stationary transducer emits ultrasonic pulses to capture either A-mode or B-mode images sequentially, enabling the measurement of organ movement over time.[23]

For traditional B-mode imaging, the US image is formed line by line using focused transmit US beams, to achieve a high image resolution in the lateral direction. In contrast, there also exists ultrafast US images, which rely on sending a single unfocused plane wave (PW) through the entire field of view at once. With this method, higher frame rates of US images are possible, with the downside of lower lateral image resolution. [24]

2.4. Ultrasound Imaging Artefacts

During US image formation several assumptions are made that can lead to the following imaging artefacts.

Speed Displacement

During image formation, a constant SoS of 1540 m/s is assumed. Deviations in the SoS in various tissues from the assumed mean velocity result in errors in distance measurement and distortions in the image [3]. This is visualized in Figure 2.4 (A) for the example of fatty tissue with a lower SoS of 1450 m/s.

Refraction

During image formation, it's assumed that ultrasound waves travel in straight lines within the imaging medium. However, as previously discussed, when sound waves encounter an interface of two media with different SoS at an angle, refraction occurs, causing the positions of structures in the image to be altered [3]. This is visualized in Figure 2.4 (B).

Shadowing and Enhancement

Another assumption in image formation is the consistent attenuation of the ultrasound (US) beam as it passes through tissue. Consequently, regions with lower attenuation may result in signal enhancement compared to neighboring areas, causing the region behind them to appear brighter. Conversely, regions with high attenuation can produce shadows behind them, as illustrated in Figure 2.4 (C).

Chapter 3

Related Work

To the best of our knowledge, no previous research has focused on predicting the RF data of US acquisitions from US images. However, our method uses several techniques and addresses subproblems that have been previously explored by others, such as using deep learning models on US images, extracting acoustic properties of the medium from a US image, and creating synthetic US datasets. In this section, other studies that have explored these subtasks previously will be reviewed.

3.1. Synthetic Phantom Data for Ultrasound

In this section previous works are discussed that have created synthetic datasets, that match tissue property maps and corresponding US acquisitions.

One previous work that has explored creating a synthetic dataset for US images was Perdios *et al.*[25]. Their strategy involved creating numerical phantoms by placing many elliptical targets of varying scattering. The other acoustic properties (density, SoS, attenuation) remain constant through the numerical phantom. They use an optimized version of the Field II simulator [26] to simulate US images. They create 30'000 synthetic data images that they use to train a DL model, which can successfully improve the image quality of low-quality US during image reconstruction.

The work of Zhang *et al.* [27] introduces a method to predict the scattering distribution of an acoustic medium given the US image. For this, they create a statistical scattering model and for each point in the numerical phantom assign distribution parameters, that in turn, they try to predict using a DL model. To create the underlying scattering parameter maps in their synthetic dataset they use overlapping geometric shapes from thresholding random coarse gray-scale patterns interpolated at finer resolutions. The final dataset they use has a size of 4000 parameter maps.

It is important to note that both the methods by Zhang *et al.* [27] and Perdios *et al.*[25] use convolutional methods for simulation, which leads to fast simulation times. While convolutional methods are fast, they have the disadvantage of only being able to simulate phantoms that have homogeneous acoustic properties aside from the scattering pattern [28].

Our method relies on modeling a virtual phantom of real tissue structures. This is also a topic explored for other medical imaging modalities such as computed tomography (CT) and not just US. Therefore we also investigate two works that extract meaningful phantoms used for CT simulation. The research of Shi *et al.* [29] explores creating accurate numerical phantoms of the human body from a set of CT images. The images are segmented into different tissue regions and average physical properties are extracted based on the radiation measurements in the CT scan. In the work of Segars *et al.* [30] they create virtual phantom models by projecting phantoms based on patient data from CT images, such as those created by the work of Shi *et al.* [29], to mathematically defined functions such as Cubic Bezier surfaces. This reduces the computation time for CT projection calculation.

There also exist several different works that use generative models to create large synthetic US image datasets using generative deep learning methods. The method presented by Maack *et al.* [31] uses a generative adversarial network (GAN) to create synthetic US images based on a small existing dataset of real US images. Meanwhile Stojanovski *et al.* [32] uses the more recent generative models called Denoising Diffusion Probabilistic Models [33]. They use their synthetic dataset to successfully train a network to perform semantic segmentation on real data. However, these methods are less relevant for our work, since we are interested in having matching RF data and possibly also the corresponding numerical phantom.

3.2. Prediction of Tissue Properties from Ultrasound Images

Our method relies on training a DL model to predict the point-wise acoustic properties of tissues from a US image. Here are some previous works, that have discussed strategies to predict acoustic properties from US images.

As discussed in the previous section the work of Zhang *et al.* [27], has successfully trained a UNet [19] architecture to predict the scattering pattern of the acoustic medium from an US image. The model was trained on a synthetic dataset they created specifically for the task.

There have been recent works to predict the SoS for an acoustic medium from US data. One such work is the dissertation by Simson [34]. The presented method uses a UNet [19] model to predict the SoS for a numerical phantom directly from the beamformed

RF data. The model was trained on synthetic RF data that was simulated using k-Wave [18] on numerical phantoms they create specifically for breast anatomy.

The earliest work that tried to predict the attenuation value from ultrasound data is by Kuc [35] and a more recent technique was developed by Ghoshal *et al.* [36]. Both works present analytic solutions based on US physics for predicting the global attenuation value from the measured RF time signal. Predicting it from the RF data is an easier task than from the US image since the RF signal carries more information. We have not found previous works, that try predicting the acoustic attenuation properties point-wise for the imaging medium.

The work of Chean *et al.* [37] uses time-domain time domain reflectometry theory, to analytically predict the acoustic impedance, using restricting assumptions such as no significant scattering or attenuation occurring in the propagation medium.

To the best of our knowledge, there is no existing work on trying to predict the mass density of the imaging medium from medical US images. The work that comes closest would be that of Sanabria *et al.* [38], where they use air-coupled ultrasound to predict the density of particle boards during inspection. However, since this is barely related to our case of medical US images, we will not get into their method.

Chapter 4

Methodology

This chapter will first outline the virtual phantom model we have developed. Subsequently, it discusses the methods used to simulate RF data given a virtual phantom and to generate a corresponding US image. The simulation will be used for both the synthetic data generation and our physically-informed model. Various strategies for creating diverse datasets of virtual phantoms are then explored. Finally, we will describe the architectures of the deep learning (DL) models we use to predict RF data given a US image.

4.1. Virtual Phantom Model

Our virtual phantom model specifies all relevant acoustical properties of the imaging medium. We define the underlying propagation medium as a two-dimensional grid, denoted as G, with height N_z and width N_x:

$$G = \{(i, j) \mid 1 \leq i \leq N_z,\, 1 \leq j \leq N_x\}$$

The lateral direction is referred to as the x-axis and the axial direction as the z-axis. The grid points represent the discretization of the physical propagation medium, with the physical grid spacing defined by the distance Δxz, which is uniform in both the lateral and axial directions (See Figure 4.1). The transducer is positioned at the top of the grid along the x-axis, with the transmission pulse directed along the z-axis, as visualized in Figure 4.1.

At each grid point with coordinates (i, j), we assign four acoustical properties that characterize our virtual phantom:

- Density $\boldsymbol{\rho}(i, j)$, with unit $[\frac{kg}{m^3}]$

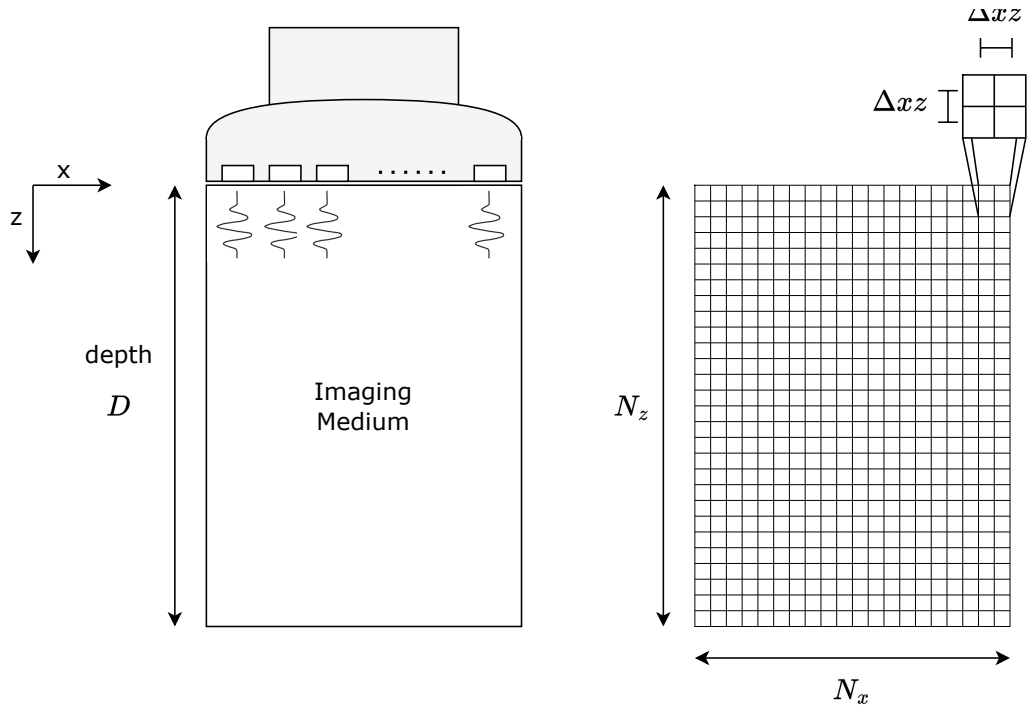


Figure 4.1.: A visualization of the virtual phantom grid G with dimensions N_x and N_z. During the simulation, the transducer will be placed at the top of the grid.

- Speed of sound $\boldsymbol{c}(i,j)$, with unit $[\frac{m}{s}]$
- Attenuation coefficient $\boldsymbol{\alpha}(i,j)$, with unit $[\frac{\text{dB}}{\text{MHz}\cdot\text{cm}}]$
- Scattering $\boldsymbol{s}(i,j)$, which has no absolute unit. For further details, refer to Section 4.1.1

These maps collectively describe the virtual phantom for all points in our imaging medium.

Thus, we can represent our virtual phantom as a function $\boldsymbol{P}$ that maps each grid point to these four phantom values:

$$\boldsymbol{P} : (i,j) \mapsto \begin{bmatrix} \rho \; c \; \alpha \; s \end{bmatrix} \tag{4.1}$$

In practice, the virtual phantom is represented as four matrices of dimensions $N_z \times N_x$: the density map $\boldsymbol{D}$, the speed of sound (SoS) map $\boldsymbol{C}$, the attenuation coefficient map $\boldsymbol{A}$, and the scattering parameter map $\boldsymbol{S}$. When stacked, these form a $4 \times N_z \times N_x$ tensor referred to as the virtual phantom $\boldsymbol{P}$. An illustration of such a virtual phantom is shown in Figure 4.2.

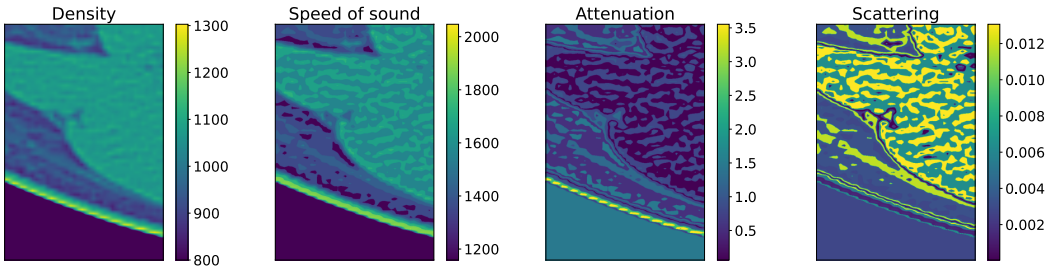


Figure 4.2.: An example virtual phantom $\boldsymbol{P}$. It is characterized by four maps that model the acoustic properties of the medium: Density, SoS, Attenuation, and Scattering. This phantom was generated using a CT slice (see Section 4.3.4).

The density $\rho(i, j)$ with unit $[\frac{kg}{m^3}]$ models the mass density of the medium at any grid point, while the SoS value $c(i, j)$ with unit $[\frac{m}{s}]$ models the acoustic speed of sound for any grid point. Together, they determine the acoustic impedance at every point, which is essential to characterize how the incident sound wave is reflected.

Additionally, the attenuation coefficient $\alpha(i, j)$ models the attenuation of a signal considering scattering and absorption, which is essential for simulating the sought-after physical properties.

We do not model the attenuation power exponent, due to limitations of the simulation software we use (See Section 4.2.3). Furthermore, we do not consider the non-linearity parameter B/A of the tissue, and the ultrasound acquisitions are always modeled linearly. The model is made simpler and more efficient by disregarding this.

The scattering is not a physical measure but describes variances of a statistical model. Using this statistical model we can draw realizations of the scattering distribution for the density, resulting in the density map ρ_s, which includes scattering elements. The definition of the statistical model and the precise interpretation of the scattering parameter are detailed in the following section.

4.1.1. Scattering Model

In the virtual phantom the density, speed of sound, and attenuation can be considered inherent characteristics of the medium, which can be either looked up in reference materials or measured directly through physical experimentation. On the other hand, our scattering model tries to capture the effect of scattering using a statistical model, denoted as S. By applying S, we can then generate a density map that incorporates the scattering components, $\rho_s(i, j) \sim S(i, j, \rho(i, j))$, where ρ_s signifies the estimated density distribution influenced by scattering $s(i, j)$ at the coordinates i and j, based on the initial density values $\rho(i, j)$.

Scattering occurs when structures smaller than the wavelength of the incoming ultrasound wave have a different acoustic impedance than the surroundings. We call these structures *scatterers* going forward. These scatterers cause a diffusive scatter, in contrast to the specular reflections that occur at larger surfaces at the boundaries of different acoustic impedance.

Since scatterers differ in acoustic impedance from their immediate surrounding, one could model the change of the acoustic impedance either with the density, the speed of sound, or both. We decided to model them using changes in the density. Larger speeds of sound can lead to speed displacement artifacts and lead to a longer computation time of the simulation to achieve stability.

Therefore, by applying S, we generate a density map, $\rho_s \sim S(i, j, \rho(i, j))$, where ρ_s signifies the estimated density distribution influenced by scattering at the coordinates i and j, based on the initial density values $\rho(i, j)$.

Statistical Model

We model the scattering with an adapted statistical model that is described by Zhang *et al.* [27]. The scattering of each tissue type is characterized by three parameters $(\upsilon_s, \mu_s, \sigma_s)$. The parameter $\upsilon_s \in [0, 1]$ describes the scatterer density, meaning the number of scatterers in a given area, which is not to be confused with the mass density ρ. Further μ_s describes the mean and σ_s the standard deviation of the scatter amplitude.

Drawing a sample from this probabilistic model means that for each point p_{ij} in our grid we draw from a Bernoulli distributed random variable (RV):

$$X_{ij} \sim \text{Bernoulli}(\upsilon_s)$$

The RV draws value 1 with probability υ_s and 0 otherwise. The points where the RV equals 1 are classified as scatterers. If a point is a scatterer, we sample the density value from the Gaussian distribution $\mathcal{N}(\mu_s(i, j), \sigma_s(i, j)^2)$. Otherwise, the mean density $\rho(i, j)$ is assigned. The density with the scattering applied can therefore be described to be drawn randomly in the following way:

$$\upsilon_s(i, j) = \begin{cases} \sim \mathcal{N}(\mu_s(i, j), \sigma_s(i, j)^2) & \text{if } X_{ij} = 1 \\ \rho(i, j) & \text{otherwise} \end{cases} \tag{4.2}$$

The parameter $\upsilon_s \in [0, 1]$ describes the scatterer density, meaning the number of scatterers in a given area. This density is fixed to always be 10 scatterers per resolution cell which is in line with Pinton *et al.* [39] and Perdios *et al.* [25]. Given our simulation setup, the resolution cell was experimentally determined to be 0.6357mm in axial and 0.6949mm in lateral direction, which results in approximately 9×10 grid points given our simulation parameters described in Section 4.2.1. In practice, the grid is separated

into distinct resolution cells of the above-mentioned size and for each resolution cell, 10 points are chosen at random as our scatterers.

For the mean of our scattering, we choose $\mu_s(i,j) = \rho(i,j)$ to have it centered around the density. The scattering variance $\sigma_s(i,j)^2$ is defined with relation $\rho(i,j)$ and $s(i,j)$.

To accomplish this, we draw a scattering factor for each location. If a grid point is a scatterer, we will draw the factor with a variance $s(I,j)$for that grid point.

$$s^*(i,j) = \begin{cases} \sim \mathcal{N}(1, s(i,j)) & \text{if } X_{ij} = 1 \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

Then we multiply the drawn scattering factor at each grid point with the density at this grid point:

$$\rho_s(i,j) = s^*(i,j) \cdot \rho(i,j) \tag{4.4}$$

If we combine formulas 4.3 with 4.4 using the linearity of the normal distribution, it leads to our desired statistical model 4.2 with mean $\mu_s(i,j) = \rho(i,j)$ and variance $\sigma_s(i,j)^2 = s(i,j) \cdot \rho(i,j)^2$ leading to a standard deviation of $\sigma_s(i,j) = \sqrt{s(i,j)} \cdot \rho(i,j)$.

For all the phantoms we generate, $s(i,j)$ is in the range $\in [0, 0.015625]$ which means the standard deviation ranges between $\in [0, 0.125 \cdot \rho(i,j)]$. Experimentally, we find that higher scattering does not increase contrast but only leads to an artifact, where the target appears larger than it actually is.

4.2. Data Generation Pipeline

We want to create large datasets of virtual phantoms, corresponding RF data from US acquisitions and the thereupon formed US image of the RF data. In this section, we provide a detailed explanation of how we can simulate the acquired RF data and form the US image given a virtual phantom.

The data generation can be broken down into 3 steps:

1. Phantom Generation
2. Simulation
3. Image Formation

An overview can be seen in Figure 4.3. First, we also specify the exact specifications that are used to create all the synthetic samples in the datasets we produced.

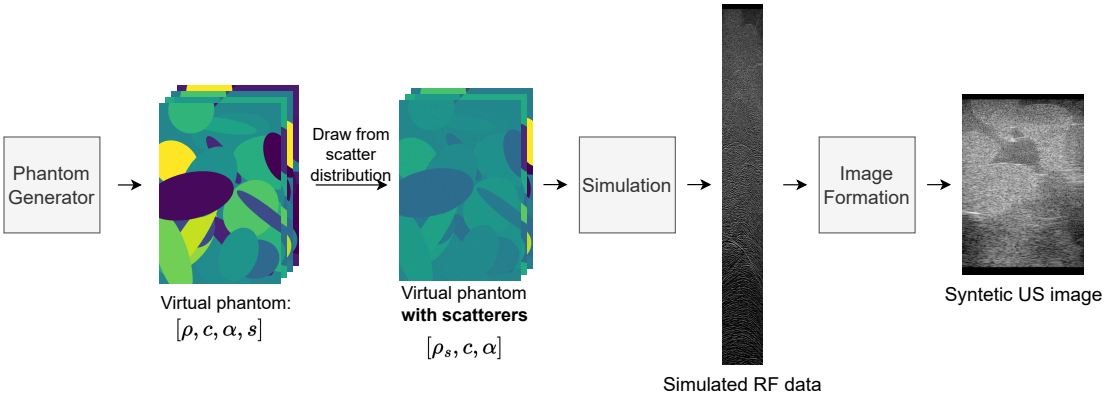


Figure 4.3.: Overview of the data generation pipeline.

4.2.1. Setup Specifications

The phantom grid size, simulation grid size, and final image size are identical for all samples in the synthetic datasets. By fixing the setup specifications across our synthetic datasets, consistency can be guaranteed.

Furthermore, a point for point correspondence between phantom values and image pixels is achieved, by choosing equal phantom and image dimensions. This correspondence makes the point wise prediction task for our DL models easier.

Our setup models a single plane wave (PW) US acquisitions using the Verasonics Vantage 256 ultrasound research system (Kirkland, WA, USA) coupled with a 192-channel transducer, specifically the GE 9L-D model from General Electric Healthcare (Chicago, IL, USA). This is modeled using the following grid and transducer parameters described in Tables 4.1 and 4.2.

Grid parameter	Value
N_x	576
N_z	856
Δxz	0.074mm

Table 4.1.: Overview of the grid parameters used in creating the synthetic datasets.

The grid width N_x is chosen to match the width of the transducer using the grid spacing Δxz. Δxz is selected such that the transducer element pitch p is a multiple of Δxz. In our case, we have $p = 3 \cdot \Delta xz$. The choice of both the N_x and N_z is also motivated by the utilized simulation model, since simulation grids with a smaller prime factorization are simulated faster [40].

Transducer parameter	Value
# of Transducer Elements	192
Transducer Element Pitch p	0.23mm
Center Frequency f_{center}	5.2083 MHz
Bandwidth	(3.1325, 7.2875) MHz
Sampling Frequency	20.8332 MHz
Input Signal Length (Wavelengths)	3
Elevation Focus	28 mm

Table 4.2.: Overview of the transducer parameters used in creating the synthetic datasets.

From the above-mentioned parameters, we can calculate all relevant parameters for our phantom grid. Here are a few relevant parameters that follow from these specifications:

$$\text{height} = N_z \cdot \Delta xz = 63.28\text{mm} \tag{4.5}$$

$$\text{width} = N_x \cdot \Delta xz = 42.58\text{mm} \tag{4.6}$$

$$\text{wavelength} \quad \lambda = \frac{c_{ref}}{f_{center}} = 0.29\text{mm} \tag{4.7}$$

Here c_{ref} refers to the assumed average SoS in tissue and we apply $c_{ref} = 1540\frac{m}{s}$.

Here, it is important to note, that when testing our implemented models, the size of the grid or the transducer specifications may be chosen differently.

In this project, solely single plane wave imaging is modeled. While image quality could be improved by using multiple plane waves from different angles, this would also increase simulation time significantly.

4.2.2. Phantom Generation

In the first step, we need to create the virtual phantom $\boldsymbol{P}$ that fully characterizes the imaging medium captured using the US simulation. The virtual phantom $\boldsymbol{P}$ is defined as the concatenation of the four maps that model acoustic properties $\boldsymbol{P} = [\boldsymbol{D}, \boldsymbol{C}, \boldsymbol{A}, \boldsymbol{S}]$, where $\boldsymbol{D}$ is the density map, $\boldsymbol{C}$ corresponds to the SoS map, $\boldsymbol{A}$ describes the attenuation map and $\boldsymbol{S}$ the scattering map.

The virtual phantom $\boldsymbol{P}$ can either be constructed using the various strategies mentioned in Section 4.3 or it can be predicted by our DL model using a US image during inference.

In any case, the first step before passing the virtual phantom to the simulation, is to sample the scatterers using the scattering distribution defined by our scattering parameters in the phantom map $\boldsymbol{S}$. This is also illustrated in Figure 4.3. In Section 4.1.1, a detailed explanation of the sampling process of the scattering model is provided.

4.2.3. Simulation

Selecting an appropriate simulation tool is crucial for generating realistic synthetic US data. The importance of the simulation method for this project necessitates a discussion on the selection of the US simulation approach. Two common techniques for simulating US are convolution-based methods and wave-based numerical methods.

In convolutional methods, the interactions of US are estimated through the process of convolving the spatial impulse response of the acquisition system with the medium's scatterer pattern. An example of this implementation is found in the widely used Field II simulation tool by Jensen *et al.* [26]. The primary benefit of this approach lies in its fast computation times, which enables its application even for real-time applications [27]. However, the convolution based method is based on the assumption that the propagation medium consists of a homogeneous background with point scatterers. In addition, the attenuation can only be modelled using a linear frequency dependence for convolutional simulation methods [28]. Therefore this simulation method falls short for our purposes due to the presence of larger-scale heterogeneities in our virtual phantoms.

To simulate a heterogeneous medium with spatially varying speed of sound, density and attenuation, we need to use numerical-simulation methods [28]. The idea is to simulate the US waves propagating through the medium using numerical methods to compute the solutions of the partial differential equations that describe the physical system. Consequently, the simulation tool we chose that fits this criterion is the k-Wave toolbox by Treeby *et al.*. [18].

The k-Wave toolbox gets its name from the utilization of the k-space pseudo-spectral method for efficient computation of time-domain solutions to differential equations. It supports simulations across one, two, and three dimensions, though our focus lies on two-dimensional simulations. Primarily designed as a Matlab toolbox, k-Wave also offers a C++ version, which we take advantage of to improve simulation performance. Moreover, both the MATLAB and C++ versions of k-Wave implement optimizations for parallel processing on a GPU, boosting simulation efficiency even further.

MATLAB Implementation

All properties to execute the k-Wave simulation functions are defined using the following four input structures: The computational grid `kgrid`, the acoustic medium `medium`, the acoustic source `source` and the sensor object `sensor`.

For the 2D computational grid `kgrid` we need to define the following parameters [40]:

- the grid height `Nx` where phantom height N_z is chosen.
- the grid width `Ny` where phantom width N_x is chosen
- the spatial resolutions `dx` and `dy` grid spacing Δxz is selected for both

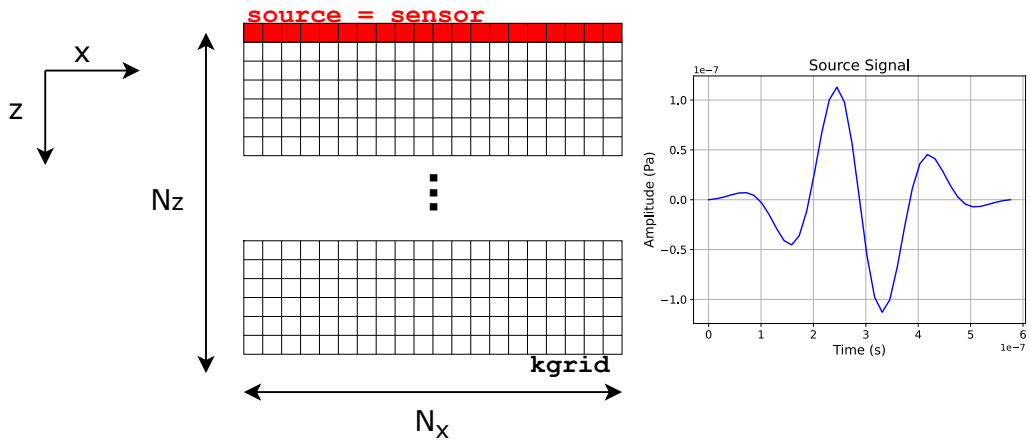


Figure 4.4.: The left side displays the simulation grid, while the right side shows the source driving the transducer elements.

- the temporal resolution `dt` and the total number of time steps `Nt`

The temporal resolution `dt` is chosen as suggested by k-Wave based on the maximum speed of sound in our system and based on the spatial resolution Δxz and the maximum speed of sound in our phantom c_{max} [40]. We avoid selecting it based on the transducer's officially provided sampling frequency, as this could result in unstable simulations. The total number of simulation steps `Nt` is determined by first specifying the total simulation time T. T is chosen to be the duration for an ultrasound pulse to travel twice the depth D of the simulation grid, ensuring that all echoes reflected from the imaging medium are recorded:

$$T = 2 \cdot \frac{D}{c_{avg}} \tag{4.8}$$

$$\texttt{Nt} = \frac{T}{\texttt{dt}} \tag{4.9}$$

Here, c_{avg} is the approximation of the average speed of sound and equals 1540 m/s.

The acoustic medium `medium` is defined using the following parameters [40]:

- 2D matrices for `density`, `sound_speed` and attenuation coefficient `alpha_coeff`. Here, we assign the respective maps of the virtual phantom. For the density the density map with the added scatterers ρ_s is assigned.
- the attenuation power law exponent `alpha_power`. This is fixed to value 1.5, since realistic values for this parameter range between 1 and 2 [9]. Using k-Wave this can only be chosen globally for the entire imaging medium.

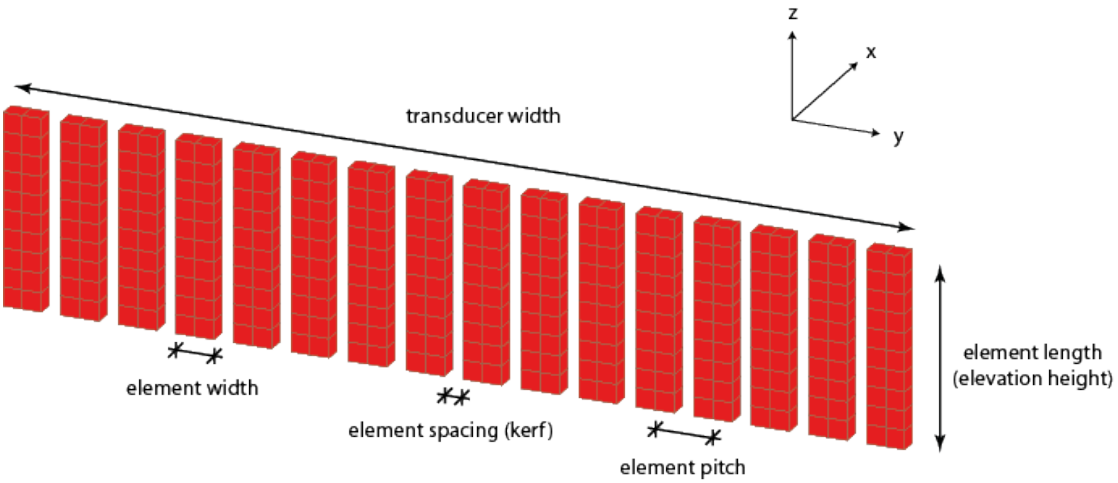


Figure 4.5.: Visualization of the transducer element pitch, width and kerf. [5]

- the non-linearity parameter `BonA`, which we do not define. This parameter describes the non-linearity parameter B/A which indicates the relative impact of the non-linear finite-amplitude effects on the speed of sound. By not defining it, k-Wave will discard the non-linear term and only use linear differential equations for the simulation [40].

Using the structures `sensor` and `source` we define the location of the transducer in our simulation grid. For each structure a mask is defined that specifies the grid points, where the input signal will be generated and where the pressure data will be recorded. The transducer is positioned at the top and since our simulation grid matches the width of the transducer, the transducer mask includes all grid points where $z = 0$ (See Figure 4.4). The same mask is assigned to `sensor.mask` and `source.p_mask`. The setup specifications (see Section 4.2.1) ensure that each transducer element corresponds to an integer number of grid points. For our specifications, one transducer element is consistently represented by three adjacent grid points.

For a real transducer, the width of the transducer elements is not identical to the element pitch (see Figure 4.5). The physical distance between the transducer elements is called kerf. However, we disregard the kerf and approximate the element width as being equal to the pitch. Our reasoning is that accurately representing the kerf, typically one-tenth the size of the pitch, would require reducing the spatial resolution of our phantom's grid Δxz by a similar factor. This adjustment would quadratically increase the size of the phantom and, by extension, the computational time of the simulation.

The final thing to define is the excitation signal that drives the transducer elements and the US acquisition. This is achieved using the `toneBurst` function in k-wave, which utilizes the transducer's center frequency f_{center}, the length of the input pulse in cycles, and a Gaussian envelope (See Figure 4.4).

Once these objects are defined, the simulation function to solve the coupled first order system of equations in 2D can be called:

```
kspaceFirstOrder2D(kgrid, medium, source, sensor);
```

The simulation function by k-Wave is optimized to run more efficiently using a GPU.

This process yields recorded pressure data for each time step at each grid point included in the sensor mask we defined. As previously noted, a transducer element is represented by a fixed number of grid points within the sensor mask—three grid points, according to the setup specifications in Section 4.2.1. To obtain the channel data for a single transducer element, we sum the pressure data signals received by each grid point within that element.

All that remains is to resample the received RF data from each channel to match the expected sampling frequency of the transducer. This step is necessary because the simulation's temporal resolution is determined independently from the transducer's sampling frequency. We accomplish this using the 1D interpolation method `interp1` provided by MATLAB.

C++ Implementation

While the MATLAB code performs well, each simulation requires a substantial amount of time to complete, even with GPU acceleration. The long simulation time is problematic, because we aim to create large datasets of synthetic data using the simulation tool. The amount of data we can generate is constrained by the duration of the simulations.

To address this issue, we take advantage of the C++ implementation that k-Wave offers. The C++ implementation can reduce simulation time significantly, by exploiting the temporal and spatial locality of the data during the simulation, which cannot be achieved using MATLAB code [40]. For our simulations, we use the CUDA-optimized binaries of k-Wave, version 1.3. This allowed us to also take advantage of GPU acceleration, further reducing the simulation speed.

To interface the C++ binary, we need to create a simulation input file, which is stored as a hierarchical data format (HDF) file [41]. The input file contains all relevant information for the simulation, that was previously stored in the four structures `kgrid`, `medium`, `source` and `sensor`.

Although k-Wave provides a MATLAB function to create a correctly formatted simulation input file from the four input structures, our codebase predominantly utilizes Python, making the repeated invocation of MATLAB processes time-consuming. Therefore we have implemented our own version to create the simulation input file using Python, which allows us to directly generate the simulation input file after we generate the virtual phantom. Our implementation translates all MATLAB code and functions used by

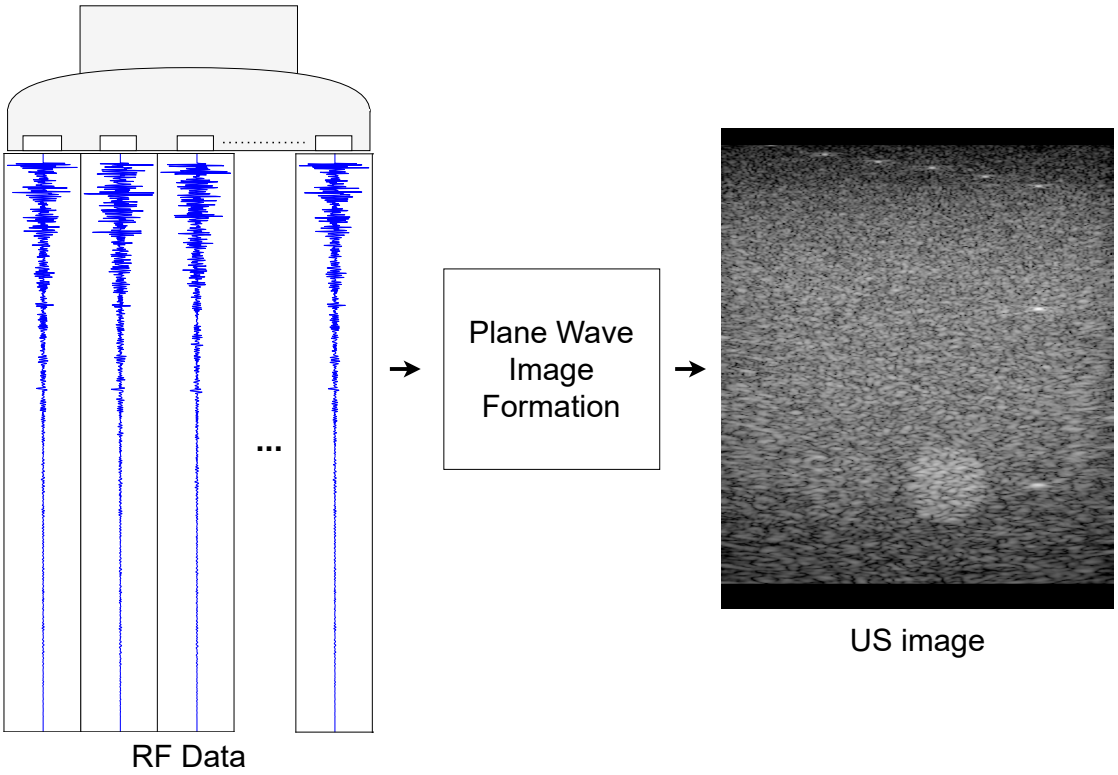


Figure 4.6.: Visualization of the plane wave image formation to generate a US image from RF data

k-Wave for establishing the simulation grid and input signal into Python, streamlining the process.

After the simulation is complete, the C++ binary returns another HDF file containing the simulation outputs. Using Python we can then extract the received pressure signal for each grid point in the sensor mask from the output file. Then we need to resample the pressure signals and aggregate the signals across the grid points for each transducer element. This last step was described in more detail for the MATLAB implementation.

4.2.4. Single Plane Wave Image Formation

After obtaining the RF data from the simulation, we have to construct the US image from a single plane wave recording. This task is illustrated in Figure 4.6.

Creating high-quality ultrasound images from single PW recordings remains an unresolved challenge, with numerous open competitions aimed at addressing this issue [42] [43]. We employ the standard beamforming method, called the delay-and-sum (DAS) beamforming [44].

To form an ultrasound image from the simulated RF data, the following processing steps in addition to the DAS-beamforming are performed:

1. Remove Input Signal
2. Time Gain Compensation (TGC)
3. Filtering
4. Hilbert Transform
5. DAS Beamforming
6. Envelope
7. Interpolation
8. Log Compression

To implement these steps, we utilize the dasIT library [45]. This library offers functions and guidelines necessary for implementing each processing step.

Remove Input Signal

In the initial processing step, we set the beginning segment of the received RF data to zero. This adjustment is necessary because, during the early phases of the simulation, the sensor captures the excitation signal emitted by the source. To focus solely on the echoes of the excitation pulse, we eliminate the RF data from these initial time steps.

Time Gain Compensation (TGC)

In the second step, we address the attenuation of ultrasound signals using time gain compensation (TGC). As the ultrasound pulse propagates through tissue, it undergoes attenuation, leading to weaker echoes from deeper structures [7]. The TGC step compensates for the attenuation, by amplifying the signals from deeper structures.

In the Verasonics US acquisition systems that we aim to model (see Section 4.2.1), the gain correction is achieved by applying a TGC waveform to the signal [46]. To model such systems as closely as possible, we also apply a TGC waveform. The TGC waveform is a piecewise linear function, that is multiplied with the received RF data signal[45]. The TGC waveform is defined using a set of 8 control points that are interpolated.

We calibrate the TGC waveform based on the experiment at Section 6.1, where we utilized the desired acquisition system and transducer to capture a known phantom structure. This captured data was then compared to a virtual recreation of the phantom. The control points derived from this calibration are detailed in Table 4.3. The impact of the TGC is illustrated in Figure 4.7.

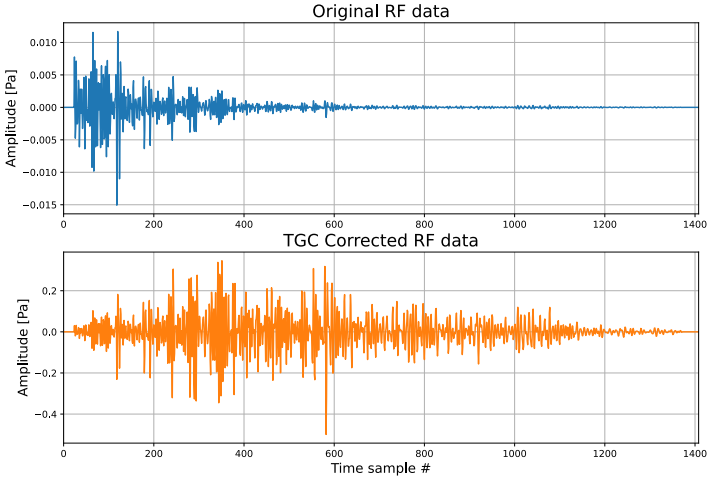


Figure 4.7.: Comparison of the RF data before and after the TGC correction.

1	20	100	250	350	500	500	300	250

Table 4.3.: TGC control points

In a real US acquisition system, TGC correction is applied directly to the analog signal before it reaches the analog-to-digital converter stage. This is important, because the echoes from deep tissue regions likely fall below the detectable voltage range of the analog-to-digital converter before the TGC is applied. Therefore, when comparing our synthetic RF data with real acquired RF data, it is important to compare the two TGC-corrected signals.

Filtering

To eliminate noise outside of the desired bandwidth of the transducer we apply finite impulse response (FIR) bandpass filter. The width of this bandpass filter is set to match the bandwidth of the transducer, as specified by the manufacturer. We use the bandwidth specifications for the GE-9LD transducer (see Section 4.2.1)

Hilbert Transform

The RF data contains information about both the amplitude and the phase of the signal. The Hilbert transform is a tool to extract both the instantaneous amplitude and phase, by mapping our real RF signal to the complex analytic signal. The Hilbert transform is defined as [47]:

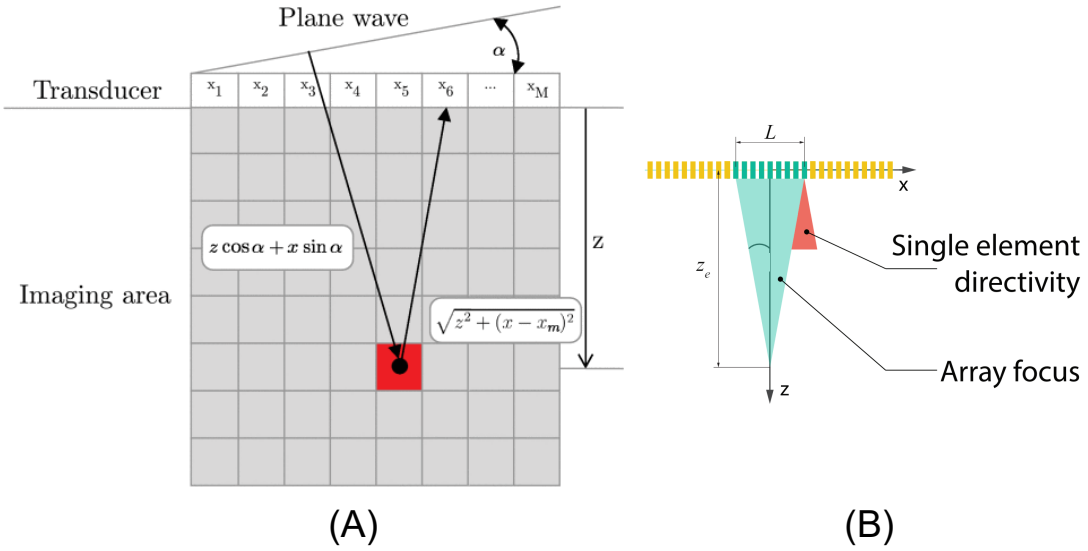


Figure 4.8.: (A) visualizes of the beamforming delays for any point on the image plane. [6] (B) shows the single element directivity of a transducer element [7].

$$H[x(t)] = \frac{1}{\pi} \int_{-\infty}^{+\infty} x(\tau) \frac{1}{t-\tau} d\tau = r(t) + j \cdot s(t) \tag{4.10}$$

By applying the Hilbert transform to the signal, we can extract the real part $r(t)$ and the imaginary part $s(t)$ of the analytic signal. These components provide information about the instantaneous amplitude, also known as the envelope $E(t)$ and instantaneous phase $\phi(t)$ [47]:

$$E(t) = |H[x(t)]| = \sqrt{r^2(t) + s^2(t)} \tag{4.11}$$

$$\phi(t) = \arg(H[x(t)]) = \arctan(\frac{s(t)}{r(t)}) \tag{4.12}$$

In a subsequent processing step, we can utilize the analytic signal to extract the envelope of our beamformed data using Equation 4.11.

DAS beamforming

The beamforming is the heart of the image formation pipeline. It maps our analytic time from the time domain of the received signal (n_channels, time) to the image plane (z,x).

The fundamental concept of the delay and sum (DAS) beamformer involves initially calculating the delays for each pixel P_e with coordinates (z_e, x_e) on the image plane

to each transducer element x_m. These delays measure the time τ_{TX} it takes for the ultrasound wave to travel from the transducer to point P_e and the time τ_{RX} it takes the echo to propagate to transducer element m. The delays can be calculated using [6]:

$$\tau_{TX}(z, x, \alpha) = (z\cos(\alpha) + x\sin(\alpha))/c \tag{4.13}$$

$$\tau_{RX}(z, x, x_m) = \sqrt{z^2 + (x - x_m)^2}/c \tag{4.14}$$

$$\tau(z, x, x_m, \alpha) = \tau_{tx} + \tau_{rx} \tag{4.15}$$

Here, it is assumed that the speed of sound c stays constant at c =1540 m/s through the entire image medium. This is an approximation that can lead to speed displacement artefacts (See Section 2.4). Although the formula can be generalized to accommodate different angles of incidence α of the PW, for our purposes, we exclusively consider $\alpha = 0$.

Once the delays are calculated, the beamformed signal for all points in the image plane can be obtained by summing the delayed analytic signals from each channel [6]:

$$I_{\mathrm{DAS}}[z, x] = \sum_m w_m[z, x] y_m[\tau(z, x, x_m)] \tag{4.16}$$

Here, $y_m[\tau(z, x, x_m)]$ represents the analytic signal for channel m at the time delay calculated in equation (4.15). Due to the directivity of the individual transducer elements (as shown in Figure 4.8 (B)), not every transducer element is influenced by the point at location P_e. Therefore, one can define $w_m[z, x]$ to only consider the signals of the relevant channels in the sub-aperture defined by the element directivity [6]:

$$w_m[z, x] = \begin{cases} 1, & \text{if } |x - x_m| \leq \dfrac{z}{2f_\#} \\ 0, & \text{otherwise} \end{cases} \tag{4.17}$$

The element directivity of a transducer element is modeled using f-number $f_\#$. This is a transducer-specific constant and is chosen by dasIT in relation to the elevation focus of the transducer.

The DAS-beamforming function we use is efficiently implemented by the dasIT library [45].

Envelope, Interpolation and Logarithmic Compression

Having obtained our analytic beamformed imageI_{DAS}, we now aim to calculate the actual pixel intensities for the final image.

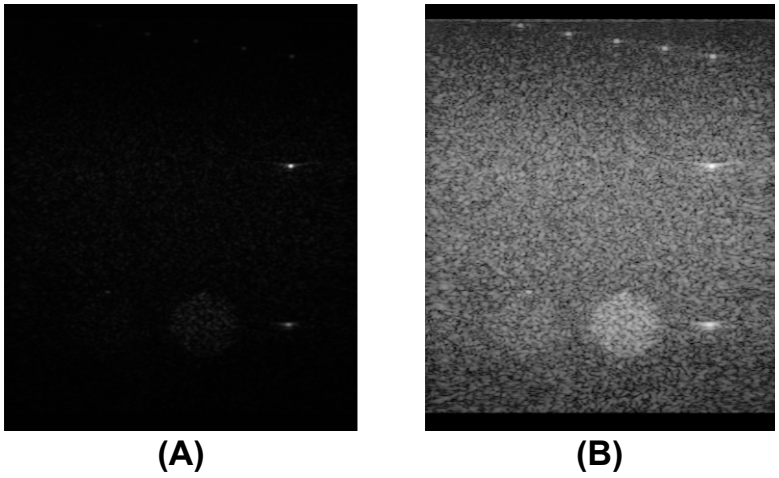


Figure 4.9.: Comparison of the resulting image before (A) and after (B) applying the logarithmic compression step.

We begin by extracting the envelope by taking the magnitude of the analytic signal (Equation 4.11):

$$I_{\text{env}} = |I_{\text{DAS}}| \tag{4.18}$$

Before the final processing step signal I_{env} has shape (N_z, n_transducer_channels). We need to interpolate our signal in the lateral x-direction, to ensure the physical pixel size is consistent in both directions and matches the original phantom grid spacing Δxz. According to our setup specifications, each transducer channel corresponds to a grid points—and by extension, image pixels—where a is an integer. Consequently, we can employ bilinear interpolation in the x-direction to expand the image width by a factor of a, which for our synthetic datasets is 3. This allows us to adjust our signal image to the desired dimensions (N_z, N_x).

Due to significant amplitude differences in the image signal, we must apply logarithmic compression to the beamformed signal before quantizing it into 8-bit pixel values. Following Palmer's method *et al.* [6], we first convert the signal to dB:

$$I_{\text{dB}} = 20 \cdot \log_{10}(I_{\text{env}}) \tag{4.19}$$

Then, as suggested by Palmer, we normalize the signal [6]:

$$I_{\text{dB norm.}} = I_{\text{dB}} - \max(I_{\text{dB}}) \tag{4.20}$$

and clamp values that fall outside of the selected dynamic range D_r [6]:

$$I_{\text{dB final}} = \begin{cases} I_{\text{dB norm.}}, & \text{if } I_{\text{dB norm.}} < -D_r \\ -D_r, & \text{otherwise.} \end{cases} \tag{4.21}$$

For the dynamic range, the value $D_r = 60$ is chosen for all samples.

Finally, we obtain the desired grayscale image by quantizing the dB values into integer numbers within the interval [0,255]:

$$I_{\text{grayscale}} = \left\lfloor \frac{255(I_{\text{dB final}} + D_r)}{D_r} \right\rceil \tag{4.22}$$

where $\lfloor \rceil$ represents the rounding to the nearest integer operator. The image in $I_{\text{grayscale}}$ corresponds to the final US image.

A comparison between the image before and after the logarithmic compression step we described is illustrated in Figure 4.9.

4.3. Virtual Phantom Datasets

A primary challenge and focus of this work is the creation of a synthetic virtual phantom dataset that exhibits sufficient diversity and complexity. This is crucial for training DL models on the synthetic dataset in a way that enables them to generalize effectively from synthetic samples to real tissue data.

We present four different strategies to create virtual phantoms:

1. Ellipse Phantoms
2. ImageNet Segmentation Phantoms
3. ImageNet-enhanced Phantoms
4. CT Phantoms
5. Muscle Texture Phantoms

For each dataset of virtual phantoms we create, we utilize the data generation pipeline outlined in Section 4.2 to generate corresponding RF data and US images for each phantom. These datasets can then be employed to train DL models.

4.3.1. Ellipse Phantoms

For our first method, we follow the work of Perdios *et al.* [25], where they created virtual phantoms by placing random ellipses. Their models demonstrated good generalization to both numerical phantoms and real tissue images. However, in their work, only the scattering parameters were varied between the targets and no other acoustic attributes [25].

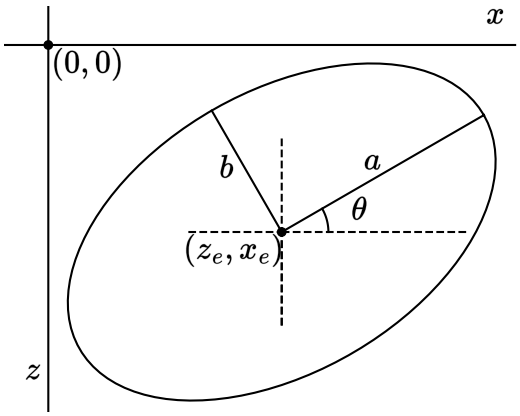


Figure 4.10.: Visualization of the ellipse parameters.

Analogously to them, we create a virtual phantom by placing a fixed number of targets within our phantom on a fixed background. Each target is a set of points of the imaging medium. The targets have the shape of geometric primitives, where we choose ellipses of random size, shape and orientation. They are placed randomly across our phantom and for each target, we assign a fixed number of targets. A further advantage of this method is that it offers an easy way to create any amount of different virtual phantoms, without any additional external data. In contrast to Perdios *et al.*, we do not only vary the scattering for the different targets but also the other four acoustic parameters defined in the virtual phantom. The method used for assigning the acoustic properties to the targets is described below.

We set the number of targets at 25 ellipses. This number can be increased to enhance the complexity of the phantom. For each target, we sample the following geometric parameters of the ellipse to determine its shape and size on the phantom grid:

$$z_e \sim \text{Uniform}(0, N_z) \tag{4.23}$$
$$x_e \sim \text{Uniform}(0, N_x) \tag{4.24}$$
$$a \sim \text{Uniform}(\frac{N_z}{20}, \frac{N_z}{4}) \tag{4.25}$$
$$b \sim \text{Uniform}(\frac{N_x}{20}, \frac{N_x}{4}) \tag{4.26}$$
$$\theta_e \sim \text{Uniform}(0, 360) \tag{4.27}$$

All geometric parameters and the location of the ellipse are independently and uniformly sampled. The lower limit for the ellipse axis lengths a and b is chosen in relation to the grid size, such that our target ellipse is still clearly visible. The upper limit is

also selected in relation to the grid size, such that the last placed ellipse cannot cover all previous ellipses. The ellipse parameters are visualized in Figure 4.10. By placing targets sequentially, we get overlapping ellipses, which lead to more complexity and more intricate boundaries than with non-overlapping ellipses.

For the background, we set the following phantom values as listed in Table 4.4. For the SoS we choose 1540 $\frac{m}{s}$, as for the average soft tissue, and which is also assumed in the image formation for all standard beamforming techniques. For the attenuation coefficient we choose 0.54 dB/(MHz·cm, the average value for soft tissue [48]. The density is chosen to be that of water, which is also close to the average density of soft tissue [48]. For the scattering parameter, we selected a value that experimentally yielded a realistic speckle pattern.

Parameter	**Value with Units**
ρ_b	1000 kg/m^3
c_b	1540 m/s
α_b	0.54 dB/(MHz·cm)
s_b	0.000625

Table 4.4.: Default values for the phantom background

We create a dataset of 50'000 virtual ellipse phantoms using this method.

Assigning Phantom Values to Targets

For each target region we have placed on the simulation grid, we assign the phantom values with a single value that is the same for all grid points in the target. If we formalize this, each target T_k is a subset of grid points $T_k \subseteq G$. Now for each target, we assign a homogeneous value to each of the phantom maps:

$$\boldsymbol{P}(T_k) = \begin{bmatrix} \rho(T_k) \\ c(T_k) \\ \alpha(T_k) \\ s(T_k) \end{bmatrix} = \begin{bmatrix} \rho_k \\ c_k \\ \alpha_k \\ s_k \end{bmatrix} \quad \text{where } \boldsymbol{P}(T_k) = \boldsymbol{P}(i,j) \quad \forall\,(i,j) \in T_k \tag{4.28}$$

The strategy we employ here to assign tissue values to the targets is not limited to the shape of the targets and is similarly implemented in the other phantom generation strategies (Sections 4.3.2, 4.3.3, 4.3.4 and 4.3.5)

There are a number of strategies on how you could choose and assign the values to each map $[\rho_k, c_k, \alpha_k, s_k]^T$. Since our goal is to model a distribution of virtual phantoms that accurately represents real tissue structures, we strive to select realistic tissue values.

Therefore, we make use of a dataset on tissue properties collected by the Foundation for Research on Information Technologies in Society (IT'IS) [9]. They provide a list of tissue types, where they provide information on the density, SoS and attenuation coefficient for each tissue type. In Table 4.5 some examples of the data they provide are listed:

Tissue Name	Density		Speed of sound		Attenuation
	Mean	Std	Mean	Std	coefficient
Blood	1050	17	1578.2	11.3	0.206
Breast Fat	911	53	1440.2	21.9	0.379
Cervix	1105	74	1629.0	5.7	0.705
Connective Tissue	1027	0	1545.0	0.0	1.260
Gallbladder	1071	63	1583.6	0.0	0.131
Kidney	1066	56	1554.3	18.2	0.244
Muscle	1090	52	1588.4	21.6	0.617
Nerve	1075	52	1629.5	20.5	1.150
Ovary	1048	0	1595.0	0.0	0.680
Pancreas	1087	59	1591.0	0.0	0.830

Table 4.5.: Small excerpt of the tissue data provided by the IT'IS Foundation[9].

For the density and for the speed of sound a standard deviation is provided in addition to the mean, while for the attenuation coefficient only the mean is given. Furthermore, they do not provide any information on scattering values or the closely related echogenicity of the tissue types.

For each target T_k we can assign a tissue $\tau(k)$ at random from the above-mentioned list of tissues:

$$\tau(k) \sim \text{Uniform}(\{\tau_{\text{Blood}}, \tau_{\text{Nerve}}, \ldots, \tau_{\text{Muscle}}\})$$

Then, we can appoint the tissue properties of our target based on the assigned tissue type. Since, for the density and SoS, we are also provided with an estimate of the standard deviation in addition to the mean, we can model the variance by sampling from an appropriate distribution:

$$\rho_k \sim \mathcal{N}(\mu_{\rho_\tau}, \sigma_{\rho_\tau} | \tau = \tau_k) \tag{4.29}$$

$$c_k \sim \mathcal{N}(\mu_{c_\tau}, \sigma_{c_\tau} | \tau = \tau_k) \tag{4.30}$$

$$\alpha_k = \alpha_{\tau_k} \tag{4.31}$$

For scattering or, by extension, echogenicity, there appears to be no well-established tabular information correlating tissue type and echogenicity in the level of detail that we need. A very broad categorization for a few tissue types is provided by [49], however,

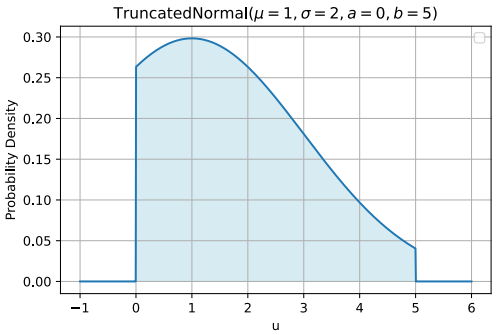


Figure 4.11.: This figure shows the truncated normal distribution used to sample the factor u_k to determine the scattering of target T_k

this list is too short and not narrow enough for our purposes, as it only differentiates between hyper- and hypo-echoic tissue.

Therefore, we assign the scattering of each target independently from the selected tissue type. We draw the scattering value s_k for target T_k at random with respect to the background scattering. This is achieved by randomly sampling a factor a, that relates the target scattering s_k and background scattering s_b:

$$s_k = u_k^2 \cdot s_b \quad \text{where} \quad u_k \sim \text{TruncatedNormal}(\mu = 1, \sigma = 2, a = 0, b = 5) \tag{4.32}$$

Intuitively, for $u_k = 1$, we get the same scattering as the background and the target could not be distinguished. For $u_k < 1$ the target is hypo-echoic, which looks dark compared to the background, whereas at $u_k = 0$ we have a region with no scattering. For $u_k > 1$, the region is hyper-echoic, which appears bright on an image.

Here, we sample u_k from the truncated normal distribution [50] which is plotted in Figure 4.11. We do not take the regular normal distribution here, and restrict values of u to be positive, as negative values for scattering do not make sense in our context. Furthermore, we avoid too large scattering values, as these can deteriorate our image quality. Therefore, we only consider $u < 5$, a range we determined experimentally. The standard deviation of this distribution is chosen arbitrarily and it could be set otherwise.

If u_k would be sampled from the uniform distribution between $[0, 5]$, it would result in a greater amount of hyper-echoic targets. This is due to the higher quantity of hyper-echoic targets in the range $[1, 5]$, than the amount of hyper-echoic targets in the range $[0, 1]$.

One thing to keep in mind here is that while all targets represent real possible tissue values, their arrangement and composition are random and are not enforced to be realistic. This also introduces unrealistic scenarios, such as brain tissue located right next to kidney tissue, as found in our dataset of synthetic phantoms. To further bias a model to learn realistic tissue regions, it would be advantageous to introduce this bias into our

dataset. This is one of the motivations for creating the dataset based on medical CT images 4.3.4.

4.3.2. ImageNet Phantoms

While ellipse phantoms serve as a practical initial model due to their simplicity and the minimal information required for generating individual data points, their utility is bounded by their inherent limitation to basic geometric forms. Real-world anatomical structures and boundaries exhibit more complex geometries, which we cannot capture using our ellipse-phantom dataset. Therefore, we aimed to create another dataset, which contains more complex geometric information about our targets. Thus, we utilize the geometric structures and information contained in the ImageNet dataset [51].

First released in 2009, the ImageNet dataset [51] is one of the most important datasets in the field of computer vision [52]. The dataset is vast, containing over 14 million annotated high-resolution images and is diverse enough to be categorized into over 20,000 categories [51].

We make use of the large variations of geometric structures that appear in ImageNet images and extract them to generate virtual phantoms containing more complex structures. To extract the shapes of all the objects that are occurring in these diverse images, we use instance segmentation. Instance segmentation is a computer vision technique, that involves identifying and delineating each distinct object of interest appearing in an image [53].

Segment Anything Model (SAM)

We perform the image segmentation using the Segment Anything Model (SAM), which is developed and released by Kirillov *et al.* [54]. SAM is a foundation model [55] for the task of image segmentation, which means it was trained on a broad web-scale dataset. This makes it ideal for our purpose, due to the large diversity of objects we want to segment in the ImageNet dataset.

Inspired from foundation models in Natural Language Processing (NLP), SAM is a promptable model, meaning, that given an image and a prompt, it will predict a segmentation mask. The prompt specifies which object to segment in the image and can range from a point to a bounding box or even text [54].

We use the implementation of the SAM provided directly by Kirillov *et al.* [54]. Because our goal is to detect all objects in an image, we use a fixed grid of points of $32x32$ as points. For each prompt, a segmentation mask is computed. Predicted masks with either a too low stability score or with a too large intersection over union (IoU) with other predicted masks are discarded. These are the filtering methods Kirillov *et al.* implement

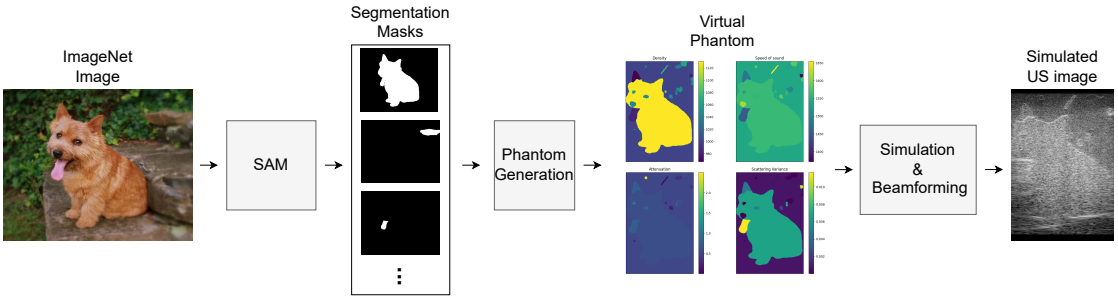


Figure 4.12.: Pipeline for generating a virtual phantom from an ImageNet image.

as a standard and we adopt these thresholds. This leaves us for the ImageNet images we have segmented with an average of between 43 and 44 segmentation maps.

Using the SAM, we extract segmentation masks for the 50'000 images of the ImageNet validation set. Subsequently, for each of these images, a virtual phantom using the segmentation masks is generated.

Creating a Virtual Phantom from Segmentation Masks

We can generate the virtual phantom the same way as we have done for the ellipse phantoms, by sequentially placing targets on a background. We reuse the same background parameters $[\rho_b, c_b, \alpha_b, s_b]^T$ as in Section 4.3.1. This time, however, each target T_k is not a simple geometric shape but an extracted segmentation mask. This means, that the number of targets on a virtual phantom corresponds to the number of extracted segmentation maps for a given ImageNet image. For each target, we assign phantom values $[\rho_k, c_k, \alpha_k, s_k]^T$ using the same methodology as for the ellipse phantoms described in Section 4.3.1. Because the ImageNet images generally have different dimensions than our phantom dimension (N_z, N_x), we must resize and crop each segmentation mask to fit our phantom grid.

The flow of data from an original ImageNet image to a virtual phantom and corresponding US image is depicted in Figure 4.12. The approach presented is not limited to ImageNet specifically and could be applied equally to other datasets. We choose the ImageNet dataset due to its diverse pool of images and popularity. An alternative could be the Microsoft Common Objects in Context (COCO) dataset [56], which is used for object detection and segmentation. For these examples, segmentation maps would already be provided, which could directly be used to create our phantoms.

4.3.3. ImageNet-Enhanced Phantoms

For both the datasets we have created so far, we have placed target areas across a background. While the targets for the ImageNet phantom dataset are geometrically

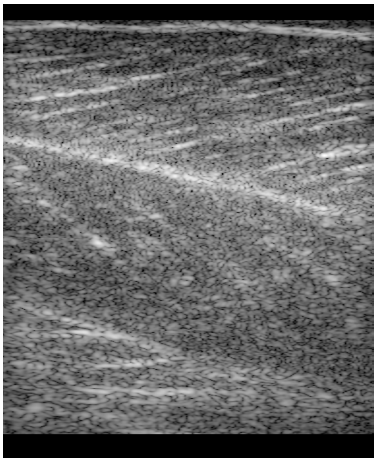


Figure 4.13.: An in vivo US image of a muscle. Structures of different shapes, densities and textures are visible inside the tissue in the US image, which we want to be represented within our synthetic datasets.

more complex, the area within the target T_k is homogeneous and the phantom values $[\rho_k, c_k, \alpha_k, s_k]^T$ remain constant. This will bias a model trained on this synthetic dataset toward predicting larger regions with homogeneous phantom parameters.

However, in real tissue structures more fine-grained structures and texture patterns can be found within a given tissue type. An example of a muscle tissue US image is illustrated in Figure 4.13. It is apparent, that there are many smaller, finer-grained structures present within this tissue, and we aim to represent such structures in our synthetic dataset.

For the **ImageNet-Enhanced** phantom dataset, we refine the ImageNet-phantom dataset by integrating more detailed structural elements into the phantoms. This enhancement is achieved by incorporating edge information from the ImageNet images in addition to the previously predicted segmentation masks.

In summary, we start by generating the same phantoms as described in Section 4.3.2 using the segmentation masks. Then for the points belonging to the edge map, we introduce an additive term. A broad overview of this approach is shown in Figure 4.14.

Edge Detection and Dilation

To extract the edges from the ImageNet images, we employ an established edge detection technique, namely the Canny edge detector algorithm [57]. This program executes the following steps to detect edges in the image: smoothing the image to reduce noise, calculating gradients to identify potential edges, applying non-maximum suppression and finally using double thresholding with hysteresis to refine the detected edges. As it

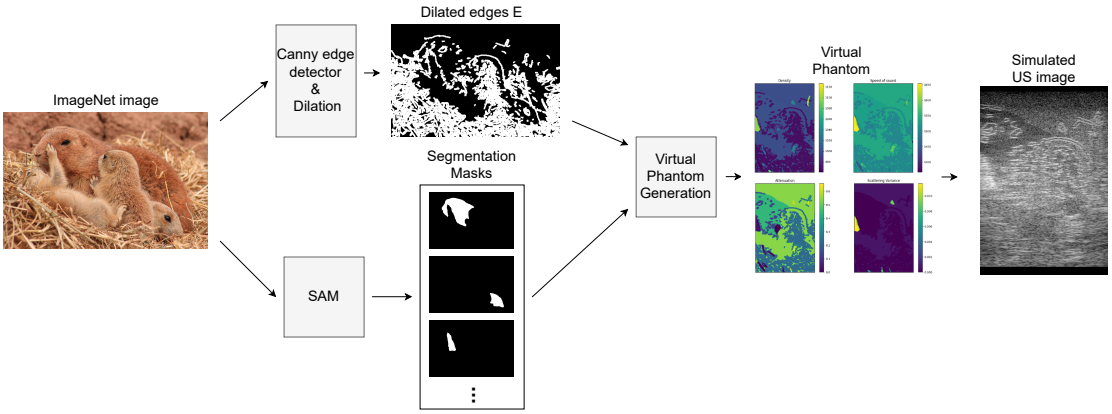


Figure 4.14.: Methodology showing the flow of data for the Imagenet-enhanced dataset.

is an already well-explored technique, we use the implementation provided by the python library `OpenCV` [58].

Due to the non-maximum suppression step in the Canny edge detector, the obtained edges are generally thin and often only one pixel wide [57]. When integrating such fine structures into our virtual imaging medium, these would introduce structures smaller than a single wavelength. As discussed in Section 4.1.1, structures of sub-wavelength dimensions primarily contribute to scattering. Given that our scattering model already accounts for such interactions, we want to avoid additional sub-wavelength structures.

Therefore, we use the morphological operation known as dilation [53] to broaden the edges. Within the simulation grid utilized for generating all synthetic data (refer to Section 4.2.1), a single wavelength is equivalent to four grid points. Bearing this in mind, we adopt a 4x4 dilation kernel. To achieve a smoother output map, an ellipse-shaped dilation kernel rather than a square one is implemented.

Choosing Appropriate Thresholds for Canny Edge Detector

The Canny edge detector uses two gradient thresholds as hyperparameters, a *low threshold* and a *high threshold*, which must be carefully selected and adapted for each individual image [53]. Intuitively, by increasing both thresholds, fewer edges with stronger gradient magnitudes are considered. This selection is critical for our purposes, as we aim to avoid edge maps where, after dilation, large portions of the map are characterized as edge points. This would result in more homogeneous maps and thereby the finer structures we strive to introduce would be lost.

Given the extensive variety of images in the ImageNet dataset we try to process, choosing the thresholds uniformly across all the images leads to the above-mentioned issues

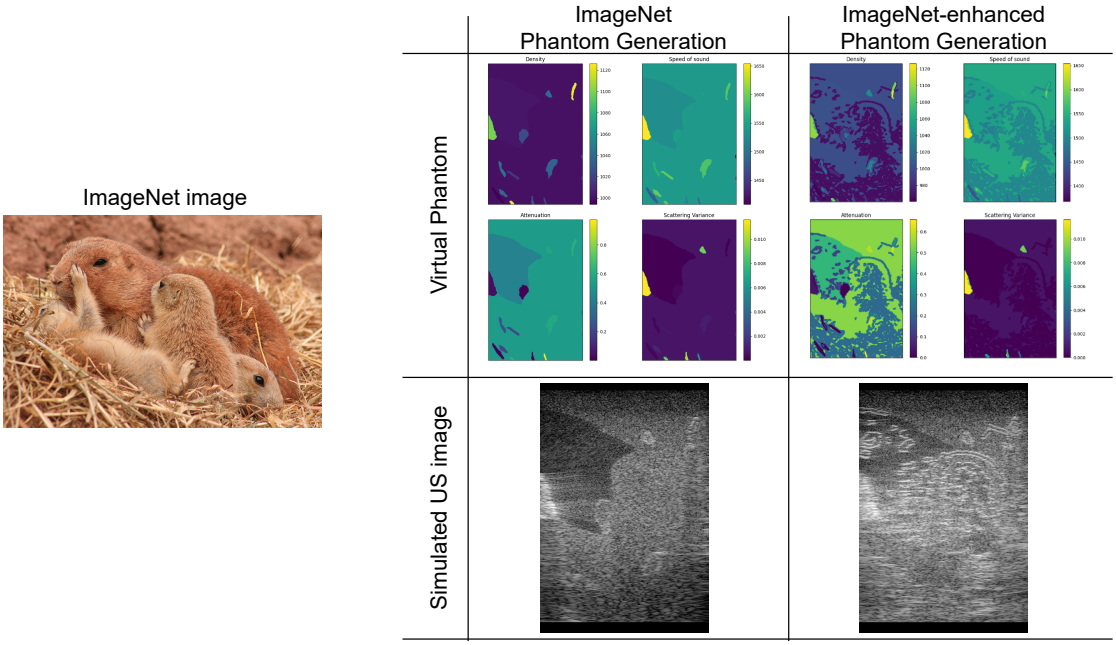


Figure 4.15.: Comparison of two phantoms with (right) and without (left) edge enhancement

and selecting the thresholds manually for each image is not feasible. To tackle this challenge we have developed and employed the following heuristics to choose the thresholds dynamically:

We begin with a low threshold of 100 and a high threshold of 200 and apply the edge detection. Then, we check for the resulting edge map if any of the four quadrant areas of the map contains at least 20% edge points. If this is the case, then we will dynamically increase both thresholds by a factor of 1.5 and recursively repeat this process until the detected edge maps are sufficiently sparse.

While this heuristic has proven effective for our applications, one could choose different initial thresholds or a different factor between the recursive iterations.

Incorporating edges into Virtual Phantom Generation

We can now incorporate the extracted edge information into the virtual phantom generation.

Initially, we perform the same steps as described in Section 4.3.2: We start with an empty simulation grid, with background phantom values $[\rho_b, c_b, \alpha_b, s_b]^T$. Then, we sequentially place targets on this grid, where each target T_k is a set of points, which is given using an ImageNet segmentation mask. By randomly assigning a tissue τ_k from

our list of tissues, we can appoint the phantom values $[\rho_k, c_k, \alpha_k, s_k]^T$ to all points in T_k. Hereafter, we obtain phantom values for every grid point (i, j) that we reference as $[\rho'(i,j), c'(i,j), \alpha'(i,j), s'(i,j)]^T$.

Now, we also have a set of points E, which corresponds to the extracted edge map and holds the obtained fine-structure information. For these points, we introduce an additive term in addition to the previously assigned phantom values:

$$\begin{bmatrix} \rho(i,j) \\ c(i,j) \\ \alpha(i,j) \\ s(i,j) \end{bmatrix} = \begin{bmatrix} \rho'(i,j) + e_\rho \\ c'(i,j) + e_c \\ \alpha'(i,j) + e_\alpha \\ s'(i,j) + e_s \end{bmatrix} \quad \forall\, (i,j) \in E \tag{4.33}$$

A different value is picked for each phantom term, since the four phantom values have different scales. Intuitively, we want this additive to be in relation to the standard deviation of each phantom map.

Since we assign ρ, c and α based on the random tissues, we approximate the standard deviations σ for the values based on the empirical distributions of the means across the tissue values presented in our list of tissues. The estimation can be achieved by using the sample variance

$$\mu = \frac{1}{N_T} \sum_{j=1}^{n} x_i \tag{4.34}$$

$$\sigma^2 = \frac{1}{N_T - 1} \sum_{i=1}^{n} (x_i - \mu)^2 \tag{4.35}$$

Here, N_T refers to the number of tissues. The aforementioned formula for the sample variance can be used to acquire the standard deviation σ for the density, SoS and attenuation coefficient.

The scatter parameter s is a bit different, as we do not extract it from an external dataset but sample it from a distribution. It is important to note that we do not directly sample s; rather, we sample a factor u from a truncated normal distribution (See Figure 4.11).

$$s = u^2 \cdot s_b \quad \text{where} \quad u \sim \text{TruncatedNormal}(\mu = 1, \sigma = 2, a = 0, b = 5) \tag{4.36}$$

To compute the variance of random variable S, we can write it in relation to random variable U, where random variable U is distributed using the truncated normal distribution [50].

$$\text{Var}(S) = \text{Var}(s_b \cdot U^2) = s_b^2 \cdot \text{Var}(U^2) \tag{4.37}$$

The variance for U^2 can be calculated using the moments of U [59]:

$$\mathrm{Var}(U^2) = (\mathrm{E}[U^4] - \mathrm{E}[U^2]^2) \tag{4.38}$$

We can approximate the moments of U $\mathrm{E}[U^4]$ and $\mathrm{E}[U^2]$ by numerically computing the integrals of the expectations[59]:

$$\mathrm{E}[U^4] = \int_0^5 u^4 \cdot \mathrm{TruncatedNormal}(\mu = 1, \sigma = 2, a = 0, b = 5)(u)\, du \tag{4.39}$$

$$\mathrm{E}[U^2] = \int_0^5 u^2 \cdot \mathrm{TruncatedNormal}(\mu = 1, \sigma = 2, a = 0, b = 5)(u)\, du \tag{4.40}$$

Then, the standard deviation of scattering parameter s is simply:

$$\sigma_s = \sqrt{\mathrm{Var}(S)} \tag{4.41}$$

Once we have the standard variations for each of our four phantom values, a random factor e of the standard deviation will be added to each map. We sample e from:

$$e = \mathrm{Uniform}(0.5, 2) \cdot \mathrm{choice}(\{-1, 1\}) \tag{4.42}$$

and we define the additive term e_j to be:

$$e_p = e \cdot \sigma_p \quad \forall p \in \{\rho, c, \alpha, s\} \tag{4.43}$$

The additive term e_j is then added for each map p for all points belonging to the edge map. Therefore, the fine structure regions will have an offset of between 0.5 and 2 standard deviations for each phantom value. We do not include the smaller offsets, since we require the fine structures to be significant to get different samples from the original ImageNet dataset.

4.3.4. CT Phantoms

While the previous two phantom generation techniques explore generating phantoms using complex geometric structures with realistic tissue properties, their composition is not based on real tissue structures. In vivo tissue structures might have reoccurring patterns of tissue composition that are not present in the previously established datasets.

Medical images provide a method for capturing tissue structures, with a broad spectrum of imaging modalities available. While our work's primary focus lies on US images, other popular imaging modalities such as X-ray, Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) can be considered to extract information about tissue

structures. Each of these modalities offers unique advantages and is capable of revealing aspects of tissues that may not be visible through other imaging methods [60].

In this dataset, we aim to capture realistic tissue structures in the created virtual phantoms using a previously collected dataset of 3D CT scans.

CT scans are generated by rotating an X-ray source and detector around the body to capture cross-sectional, high-resolution image slices[61]. These images offer a higher resolution compared to other modalities such as US, which makes them suitable for capturing even fine tissue structures. A distinctive feature of CT imaging, compared to MRI and ultrasound, is that the intensity values within a voxel correspond directly to the physical properties of the tissue. The intensities are measured in Hounsfield Units (HU) [62] which can be used to estimate realistic values for the acoustic properties of each pixel.

The Hounsfield unit is defined through the linear attenuation coefficient μ, which quantifies the penetration of a material by a beam of radiation:

$$\text{HU} = 1000 \cdot \frac{\mu - \mu_{Water}}{\mu_{Water}} \tag{4.44}$$

where μ_{Water} is the linear attenuation coefficient of water.

Using CT images we attempt to generate realistic virtual phantoms, based on authentic medical data. To generate this data, we must first preprocess the CT scan data available to us, creating CT slices that match the dimensions of our virtual phantoms. Then, we apply the HU intensities for the extracted CT slices to create a realistic virtual phantom. An overview of the process can be seen in Figure 4.16. We describe now first the method of the pre-processing, followed by how we generate a realistic virtual phantom using the extracted CT data.

The CT scans used are from the New Mexico Decedent Image Database (NMDID) [63]. They provide a dataset of whole-body CT scans of over 15'000 different subjects. For each subject, around 25 different scan volumes covering different parts of the body are available. Utilizing the data of four subjects we generate a dataset of 50'000 virtual phantoms.

Pre-Processing

The pre-processing includes the following processing steps that are also visible in Figure 4.16 (A) - (C):

1. Extract CT slices from a provided CT volume
2. Extract a region of interest through thresholding
3. Extract a crop of the CT slice that matches the size of our virtual phantom.

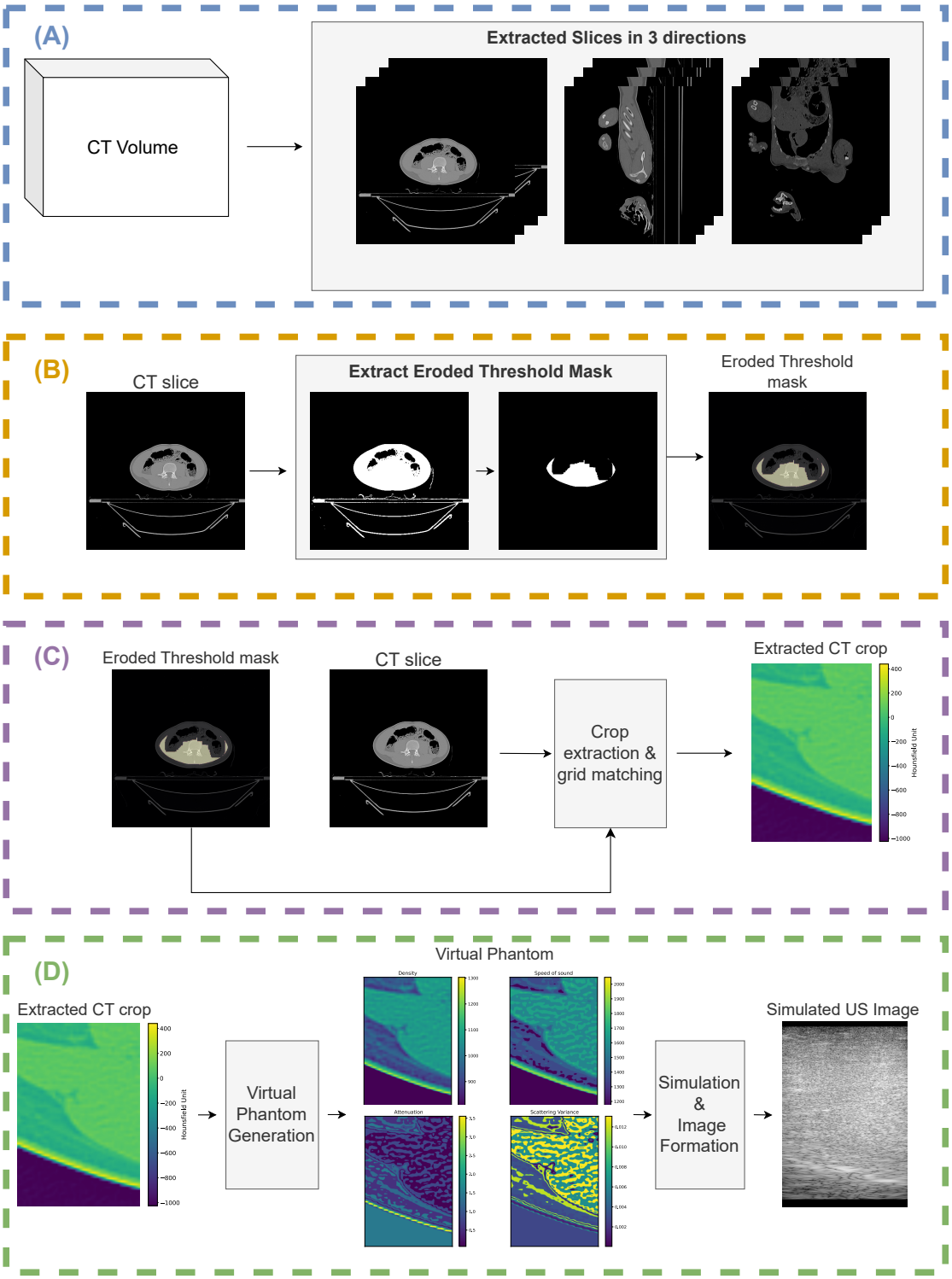


Figure 4.16.: Overview of the preprocessing of CT scans to extract CT crops (A-C), followed by generating a virtual phantom and simulating corresponding RF data and an US image (D).

The individual processing steps are described in more detail below.

The initial CT data is provided in volumes, where one volume consists of many slices contained in DICOM files. These files also include data on the physical size of the pixels and the thickness of each slice. We can concatenate the 2D slices to form a 3D volume. From this 3D volume slices in the sagittal, axial and coronal planes can be extracted, by slicing the 3D volume in the respective axis. Due to the potential difference between the slice thickness and the pixel spacings, we interpolate the extracted slices. This ensures uniform pixel spacings in both directions of the extracted CT slice. Since neighboring slices are often very similar and to increase diversity between the extracted slices, we only extract every n-th slice in every direction, where n is chosen between 5 and 20 given the processed volume. This is a heuristic that could be adapted, and if more data is needed we could decrease n. This initial step of extracting diverse slices from the CT volumes is shown in Figure 4.16 (A).

For the NMDID dataset we have approximately 25 volumes per subject and using our method around 150 slices are extracted for each CT volume.

The extracted slices are typically far larger physically than our virtual phantom size. We therefore want to extract a crop of the CT slice with the equal physical size as our virtual phantom. As depicted in Figure 4.16 (A), the extracted CT slice encompasses not only regions of tissue but also substantial areas of background. Therefore, random cropping of the image may frequently result in extracts that do not include any tissue regions. Consequently, we first need to identify the tissue regions within the CT scans. From previous experimental data [64] we learn that all tissue types have HU units larger than -700. Hence, we threshold the CT slice to only consider points with HU > -700. Now, we refine the threshold mask of points by applying the morphological operation of erosion [65] with a fixed kernel size of 25. On the one hand, we aim to eliminate equipment regions, which may have been part of the threshold mask and which are sufficiently thin in our dataset to be removed. On the one hand, we aim to eliminate equipment regions, which may have been part of the threshold mask and which are sufficiently thin in our dataset to be removed. The eroded threshold mask is referred to as the set of points M.

After extracting the region of interest in a CT slice, we can obtain multiple random crops of this CT slice, where each crop has the exact physical size of our simulation grid. Thus, we select a random point from our region of interest M and we crop around this a patch of the slice which has the desired physical size. Now, because the phantom grid spacing Δxz is generally much smaller than the pixel spacing of the cropped patch, we upsample the patch using bilinear interpolation. After the upsampling, the crop matches the size of the simulation grid.

We decide to extract 5 crops per CT slice. This could be increased if more data is needed or it can be decreased for greater variety between individual samples.

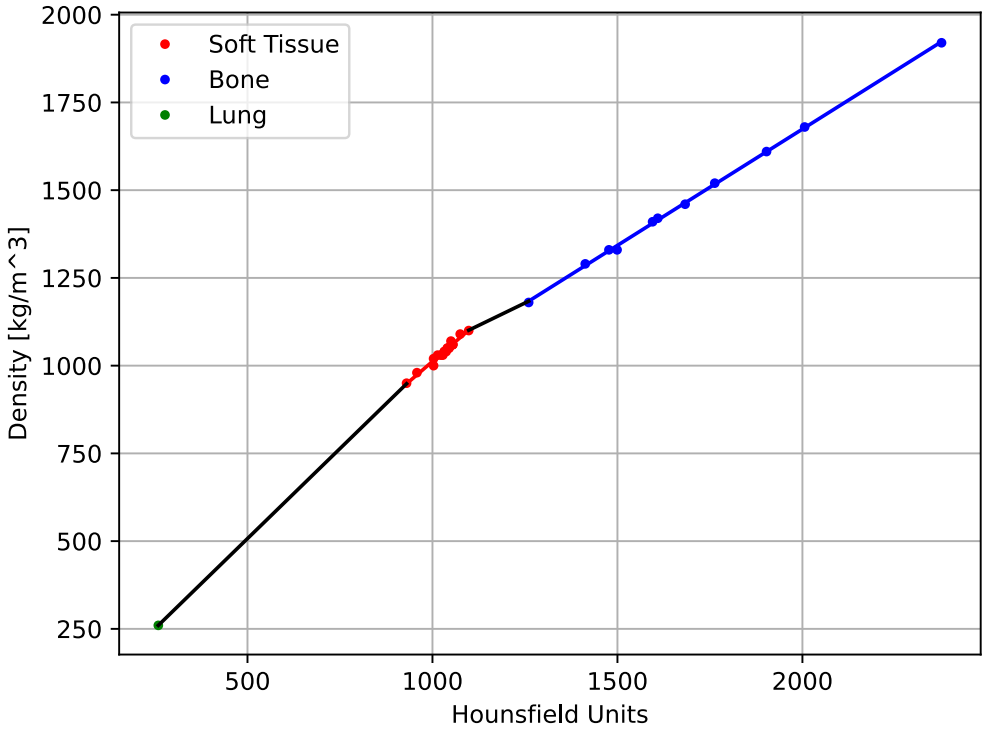


Figure 4.17.: The piecewise linear function mapping from Hounsfield unit to mass density.

Creating a Virtual Phantom from a CT Slice Crop

For the extracted CT crop, the intensities are measured in Hounsfield units. The Hounsfield unit is strongly related to the mass density ρ of the measured tissue[66]. Using the experimental data provided by Uwe Schneider *et al.*, a piecewise linear function can be fitted using linear regression to map the HU to the mass density ρ [67]. The piecewise linear function utilized was fitted by the k-Wave library function `hounsfield2density` [68]. The function is visualized in Figure 4.17.

Note that in the work of Uwe Schneider *et al.* [67] the scaled HU unit is employed, which is defined as:

$$\text{scaled HU} = 1000 \cdot \frac{\mu}{\mu_{Water}} \tag{4.45}$$

Given that our data is provided in regular Hounsfield units, we must initially shift the intensities by 1000 before applying the piecewise linear function shown in Figure 4.17.. Using this mapping, we can estimate a realistic density value for every point in our virtual phantom.

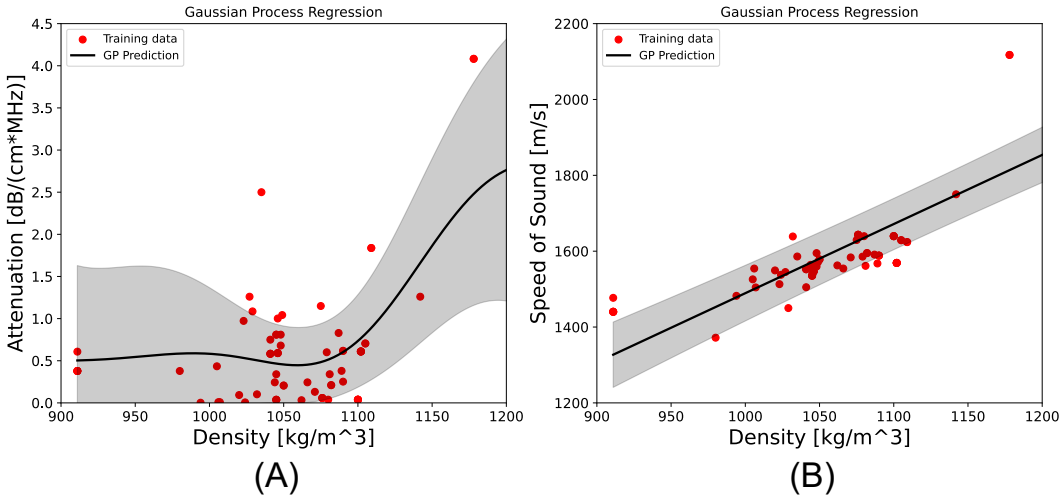


Figure 4.18.: The GP models that are fitted to the IT'IS dataset. The mean is represented by the bold black line, while the gray shaded areas denote two standard deviations around the mean. (A) shows the GP model mapping the density ρ to attenuation coefficient α. (B) shows the GP model mapping the ρ to speed of sound s.

Now, based on the density values, we choose appropriate values for the other phantom values c, α and s. Each parameter is sampled from a probabilistic model.

To enforce spatial consistency between neighboring points, we group the points into level sets based on the estimated density value $\rho(i, j)$. We can define these level sets as subsets of grid points where the density is in a range:

$$L_k = \{(i, j) \mid \text{low\_threshold}_k < \rho(i, j) <= \text{high\_threshold}_k\}$$

The thresholds are chosen manually to divide our range of densities into 21 level sets. The ranges do not have the same size and are selected to obtain more level sets for ranges where many different tissue types exist.

We then assign values c_k, α_k and s_k to all points in level set L_k given the mean estimated density over the level set, which we call ρ_k. If we were to predict these values for each grid point individually and independently instead, we would lose spatial consistency, which would be unrealistic.

It was previously shown that there exists a significant empirical relationship between the density and the speed of sound [48]. Therefore, we can fit a linear model through the data on the acoustical properties derived from IT'IS [9]. Since this is not an exact linear mapping, we want to likewise incorporate the uncertainty of the SoS value given

the density into our model. To include the uncertainty, we fit a Gaussian process (GP) regression model using a linear kernel [69].

A GP model characterizes a probability distribution over possible functions that fit the data they try to model [69]. In addition to a predicted value, it also provides an uncertainty estimate, which allows to sample a probability distribution given an input at inference. Using different kernel functions, one can include prior information about the data distribution [69]. Since there is a linear relationship between density and speed of sound, we use a linear kernel to fit the GP model [48]. This is performed using the Python implementation provided by `scikit-learn` [70].

For the attenuation coefficient α only a weak correlation with the density exists [48]. However, modeling using a GP model provides us with the uncertainty within the data and allows us to predict diverse realistic estimates for the attenuation coefficient α_k, given a density value ρ_k. Due to the absence of a direct linear correlation between the attenuation coefficient and density, we use an RBF kernel in addition to the linear kernel.

Both fitted GP models are plotted in Figure 4.18

To the best of our knowledge, there does not exist a documented correlation between the mass density and the scattering parameters s. Therefore, we assign it independently at random similar to the other phantom generation strategies described in the previous two sections.

Using this strategy we create virtual phantoms with realistic values for all 4 phantom parameters $[\rho, c, \alpha, s]$.

Filtered Data

When utilizing this method to generate a virtual phantom dataset from CT scans, we encounter a specific issue. Some samples will have large sections of bone tissue right at the top of the phantom near the transducer lens. Bones have a high SoS and attenuation compared to soft tissues. Therefore, the resulting image can be heavily distorted by speed displacement artefacts and shadowing, so that the entire image provides little information beyond the initial bone tissue section. Also, while bone tissue might occur in US acquisitions, it is usually not located right at the surface of the transducer, due to the previously mentioned artefacts.

Therefore, we filter low quality data samples from our dataset before fitting our deep learning models. Heuristically, we exclude any virtual phantoms where the top area A contains at least one-quarter of its area with density or SoS values above the threshold for soft tissue. Area A is visualized in Figure 4.19 and it spans the intervals $[0,N_z/3]$ in axial direction and $[N_x/8,\ 7/8 \cdot N_x]$ in lateral direction. The thresholds are chosen to be the maximum density and SoS values for soft tissue [9], with the density threshold being at 1178 $\frac{kg}{m^3}$ and the SoS threshold at 2000 $\frac{m}{s}$.

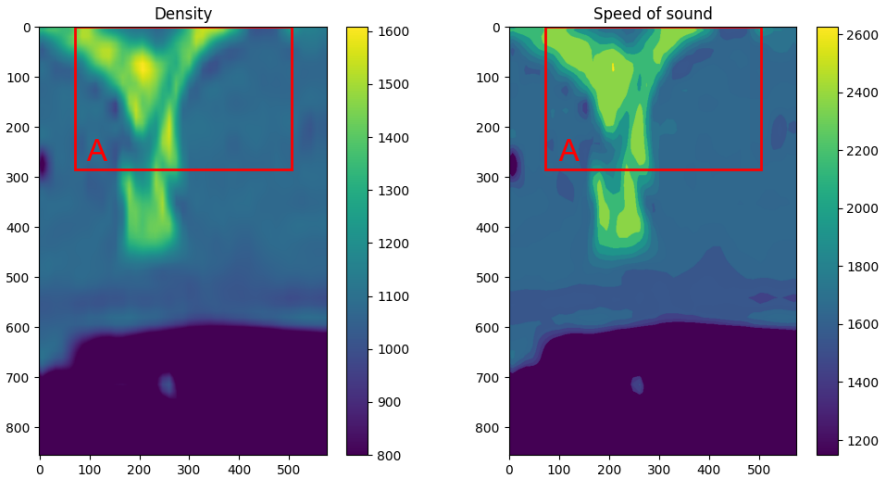


Figure 4.19.: An overview of the filtering of the phantoms with bone tissue at the transducer lens. If over 25% of the points in the phantom in area A have values over a threshold, the phantom is removed from the dataset.

4.3.5. Muscle Texture Phantoms

The previously generated datasets model a great variety of geometric shapes and different tissue structures. In this final dataset, we aim to model one specific tissue structure type. If the type of tissue structure is known in advance, it is beneficial for training to fine-tune the model to a dataset of the known tissue type.

We choose the muscle tissue of the medial gastrocnemius (MG) muscle of the lower leg, because the data for both creating such virtual phantoms [16] and evaluating the performance of our models on the domain of this muscle is available [71].

The virtual phantoms are created using a dataset of US muscle images that were collected in the work of Leitner *et al.* [16] in the context of training a DL model to predict the muscle-tendon junctions (MTJ). The dataset consists of images of the medial gastrocnemius (MG) and lateral gastrocnemius (LG) muscles and was recorded using three different acquisition systems. The US acquisition systems are the Aixplorer V6, the Esaote MyLab60 system, and the Telemed ArtUs US [16].

To extract the specific texture patterns of the MG from an existing dataset of real-life US images on the muscle tissue, we make the following assumptions: Within the MG muscle tissue sample, we have exactly two different types of tissue, muscle and tendon tissue. Additionally, we assume the tendon structures to appear brighter and more echoic in the US images than the surrounding muscle tissue.

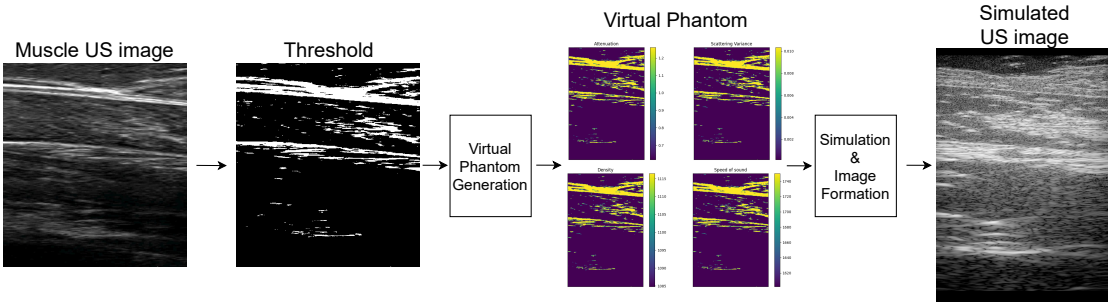


Figure 4.20.: Overview of the creation process of the muscle phantoms

Using these assumptions, we implement thresholding to receive the set of pixels M belonging to the muscle tissue and the set of pixels $\bar{M}$ corresponding to the tendon tissue. Given a crop of the original US image matching the size of our virtual phantom, we can assign the acoustic properties $[\rho_M, c_M, \alpha_M]^T$ for all points of the muscle tissue and likewise $[\rho_{\bar{M}}, c_{\bar{M}}, \alpha_{\bar{M}}]^T$ for the tendon tissue points. These values are determined using the data from the IT'IS dataset [9] and using the probabilistic method described in Section 4.3.1. Using the probabilistic method here guarantees that our model will not converge to only ever predicting the fixed tissue values of tendon and muscle.

We do not have tabular values for the scattering parameters. However, since the tendon tissue appears brighter, it has a higher scattering Therefore, we assign the muscle tissue the fixed scattering parameter that we previously used for the background scattering s_M, which is 0.000625. The scattering parameter of the brighter tendon tissue is then drawn randomly in relation to the background scattering, analogously to how we have assigned target scatterings before (see Section 4.3.1):

$$s_{\bar{M}} = u^2 \cdot s_M \quad \text{where} \quad u \sim \text{TruncatedNormal}(\mu = 3, \sigma = 2, a = 2, b = 5) \tag{4.46}$$

Note that here we draw u from a shifted distribution compared to before because we require a scattering parameter higher than the muscle tissue.

When thresholding the original US images to differentiate the muscle and tendon structures, the choice of the threshold is important. Across different acquisition systems and different subjects, this threshold might vary. For this reason, we established not a fixed threshold value but applied the histogram distribution of intensities within the US image. We determine that setting the threshold at 87.5% of the cumulative distribution function (CDF) yields optimal results in our analysis (see Figure 4.21). Intuitively, this results in one-eighth of the pixels having a higher intensity than our threshold.

Using this method, we can create virtual phantoms, which replicate real tissue structures of the MG muscle. We generate 10'000 virtual phantoms for this dataset. We do not match the dataset size of the previous datasets, because this dataset is used for fine-tuning a previously trained model and therefore needs fewer samples.

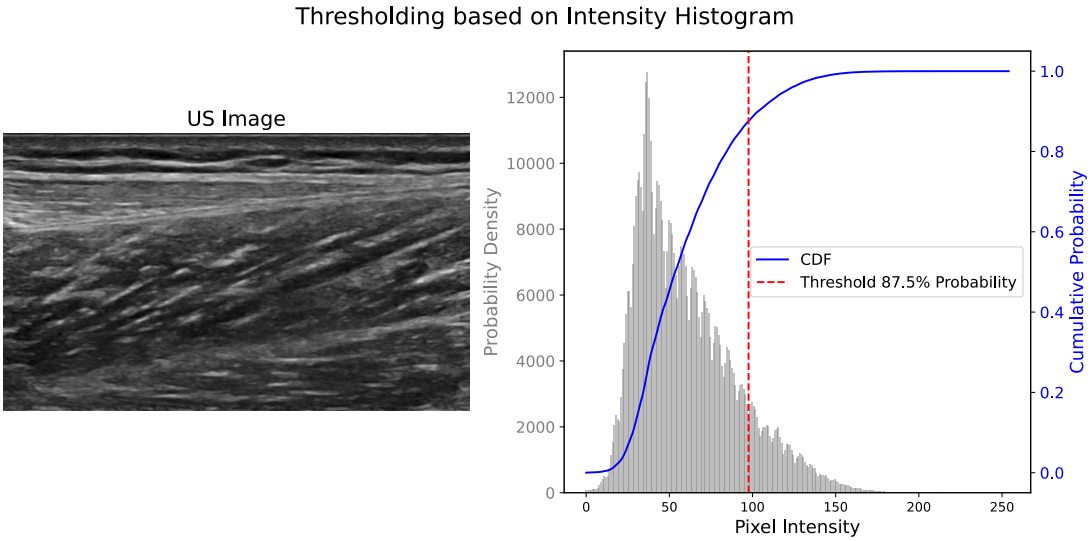


Figure 4.21.: Dynamic thresholding based on the CDF of the intensity distribution. Pixels above the threshold are classified as tendon tissue, while those below it are categorized as muscle tissue.

To consider the greatest diversity of US images possible from the US image dataset by Leitner *et al.* [16], we take the same amount of images from each acquisition machine. The images taken are also balanced across the different subjects per machine.

4.4. Physically-Informed Model for RF Data Prediction

Our goal is, given an US image, to predict the raw RF channel data. Due to the complexity and non-linearity of the image formation processing (See Section 4.2.4), there exists no closed-form solution to the proposed problem.

Formally, the problem of predicting RF data from the US image can be defined as a regression task, where we try to predict a time signal for each transducer channel. As this process involves analyzing image data to extract information, it falls within the research field of computer vision (CV). In the past decade, through the introduction of convolutional neural networks (CNN) by AlexNet [72], deep neural network (NN) architectures have established themselves as a very effective technique for various fields of CV such as image classification, segmentation, image restoration and many more. These deep learning (DL) models have also been used to great effect for tasks on US images [27] [25] [73].

It might be tempting to train a DL model to directly predict the RF channel data from the US image. However, the resulting RF data would not have any guarantee to be realistic

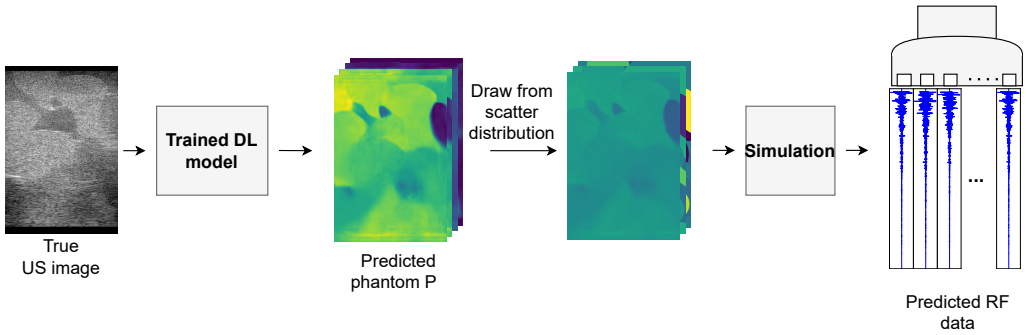


Figure 4.22.: Overview of the inference steps to make a prediction using our physically-informed model.

and be in agreement with all the differential equations, that describe the physical model of ultrasound wave propagation.

To introduce this prior information into our system, we propose a physically-informed model. Instead of predicting the RF data using a NN directly, we train a NN model to predict the acoustic properties of the imaging medium, which we model using the virtual phantom P (see Section 4.1). During inference, we can then use the simulation environment k-Wave established in Section 4.2.3 to acquire RF data for the predicted virtual phantom P. This can be seen as a form of regularization of our DL model, where we include the prior information of the laws of physics for the propagation of ultrasound waves through the simulation into our predicted RF data.

An overview of our physically-informed model during inference can be seen in Figure 4.22.

4.4.1. Problem Formulation and Loss Function

We want the DL model in our physically-informed system to learn to predict the acoustic properties described in virtual phantom P. To train a machine learning (ML) model, we first need to formalize our problem to a loss function for our neural network (NN) architecture to optimize during training.

Our problem statement can be formalized by defining the input to our model as the image $\mathbf{I} \in \mathbb{Z}^{N_z \times N_x}$, that has height N_z and width N_x. The target output of our model is the virtual phantom $\mathbf{P} \in \mathbb{R}^{4 \times N_z \times N_x}$ and can be defined as a tensor, which is a concatenation of the four phantom maps that describe the acoustic medium $[\boldsymbol{D}, \boldsymbol{C}, \boldsymbol{A}, \boldsymbol{S}]$. As a reminder, these maps model the density, SoS, attenuation coefficient, and scattering for each point

of our phantom grid. The prediction of our NN model M with trainable parameters θ can therefore be described as

$$\hat{\mathbf{P}} = M(\mathbf{I}, \theta). \tag{4.47}$$

Then, we train the parameters θ of our NN model to optimize a loss function over our training dataset:

$$\mathcal{L}_{total} = \frac{1}{N_{train}} \sum_{i}^{N_{train}} \mathcal{L}(\mathbf{P_i}, M(\mathbf{I_i}, \theta)) \tag{4.48}$$

$$= \frac{1}{N_{train}} \sum_{i}^{N_{train}} \mathcal{L}(\mathbf{P_i}, \hat{\mathbf{P}}_\mathbf{i}) \tag{4.49}$$

where N_{train} is the number of samples in our training set, and $(\mathbf{I_i}, \mathbf{P_i})$ is the i-th data sample. $\mathcal{L}$ is the loss function for a single data sample.

For the loss function, we state that the Mean Absolute Error (MAE) fits our needs best. It is a simple loss function, which is defined as the absolute difference for each map value averaged over all maps and grid points. Incorporating the MAE, we get the following loss function

$$\mathcal{L}(\mathbf{P_i}, \hat{\mathbf{P}}_\mathbf{i}) = \frac{1}{4N_xN_z} \sum_{p=1}^{4} \sum_{w=1}^{N_x} \sum_{h=1}^{N_z} |\mathbf{P_i}(p, h, w) - \hat{\mathbf{P}}_\mathbf{i}(p, h, w)| \tag{4.50}$$

where p indexes the different phantom maps by defining

$$[\mathbf{P}(1, \cdot, \cdot), \mathbf{P}(2, \cdot, \cdot), \mathbf{P}(3, \cdot, \cdot), \mathbf{P}(4, \cdot, \cdot)] = [\boldsymbol{D}(\cdot, \cdot), \boldsymbol{C}(\cdot, \cdot), \boldsymbol{A}(\cdot, \cdot), \boldsymbol{S}(\cdot, \cdot)] \tag{4.51}$$

To get the alternative Mean Squared Error (MSE), instead of the absolute value, the squared value of the difference is used.

Here, it is important to notice that our phantom maps do not have the same scale. While the density and SoS map both take values in between [700, 2200], the scattering takes values in the much smaller range [0, 0.015625]. Despite our initial attempts to address this issue by using a weighted loss function, we still encounter the challenge that our DL model must predict values of vastly different magnitudes. Predicting different targets at different scales is an inherent limitation for a DL model [74].

We resolve this issue by standardizing the phantom maps across the training dataset first. For each phantom map p, we compute the mean μ_p and standard deviation σ_p of the phantom map across the entire training dataset. Then, we can standardize each phantom map in a training sample to receive the normalized phantom map $\bar{\mathbf{P}}$:

$$\bar{\mathbf{P}}(p, \cdot, \cdot) = \frac{\mathbf{P}(p, \cdot, \cdot) - \mu_p}{\sigma_p} \tag{4.52}$$

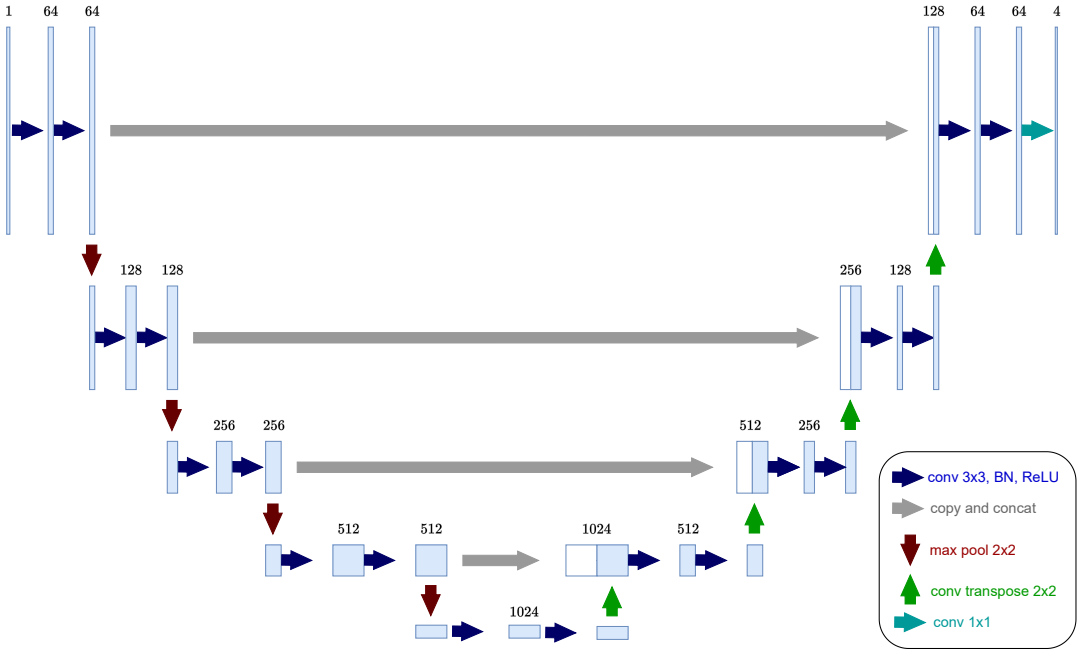


Figure 4.23.: The visualization of the UNet architecture we use, displaying the number of channels at each layer.

During inference, we must subsequently rescale the predicted phantom maps from the standardized targets back to their original scales:

$$\mathbf{P}(p,\cdot,\cdot) = \bar{\mathbf{P}}(p,\cdot,\cdot) \cdot \sigma_p + \mu_p \tag{4.53}$$

Here, we use one DL model M to predict all four phantom maps at once. Alternatively, One could opt to train a separate DL model for each phantom map individually instead of predicting all four maps with a single model. This would however increase the computational and memory load by a factor of four. Additionally, we hypothesize that the shared features might enhance the predictions of the phantom maps, since correlations between the different the acoustic parameters exist. This is related to multi-task learning [75].

4.4.2. Model Architectures

UNet

The UNet, initially developed by Ronneberger *et al.* [19], is a very popular neural network architecture for medical image processing. The architecture uses an encoder-decoder structure connected by skip connections, utilizing a series of convolutional layers. The

UNet architecture is particularly useful for pixel-wise classification or regression, since it can effectively combine both low resolution contextual information using the deep encoded features and high-resolution local information of the image.

There are numerous works using the UNet architecture on US images for numerous different applications [25] [76]. In particular the previous work of Zhang *et al.* [27] for predicting the scattering map and the work by Simson [34] to predict the SoS map both employ UNet models. Therefore the UNet is chosen for the initial model architecture.

The architecture with the exact model hyperparameters we use is shown in Figure 4.23. We choose an initial channel width of 64 and a total depth of 4. For the downsampling in the encoder, 2x2 Max-pooling is employed, while for the upsampling a 2x2 transpose convolution is used. This results in a model with a total of approximately 31 million trainable parameters.

The input dimensions for UNet models are variable and do not need to be fixed beforehand. However, both dimensions need to be divisible by factor 2^d where d is the depth of the UNet. We chose a depth of $d = 4$, which means the image dimensions need to be divisible by 16. Since the images in our synthetic datasets have size 856×576, the images are padded with zeros to the next highest image divisible dimensions, which is 864×576, before being passed to the model.

TransUNet

While the UNet has shown great results in many fields of processing medical images, there exist more recent model architectures that aim to improve the performance of the UNet. One such model that we try is called the TransUNet architecture introduced by Chen *et al.* in 2021 [8]. Like the UNet, the architecture was originally designed for medical image segmentation. The TransUNet architecture integrates a Vision Transformer (ViT) with the conventional UNet framework, significantly enhancing the encoder's capability to extract more sophisticated high-level features. The ViT model architecture was introduced by Dosovitskiy *et al.* [77] that outperformed traditional CNN architectures for image classification. The ViT is built on the transformer model architecture introduced by Vaswani *et al.* [78] for the task of natural language processing, using an attention mechanism, which helps the model to focus on relevant parts of the input data. The ViT accesses the transformer architecture, by splitting the image into a fixed number of patches, which are then embedded and processed analogously to words for the original transformer.

The TransUNet architecture is visualized in Figure 4.24. Like in a regular UNet, the input image is processed by convolutional layers and downsampled to extract high-level low resolution feature maps. Instead of immediately upsampling this feature map, they are passed to a ViT model to extract improved feature maps, profiting from attention mechanisms. The resulting feature maps are then upsampled using a traditional UNet decoder architecture using cascaded upsampling and convolutional layers. Using this

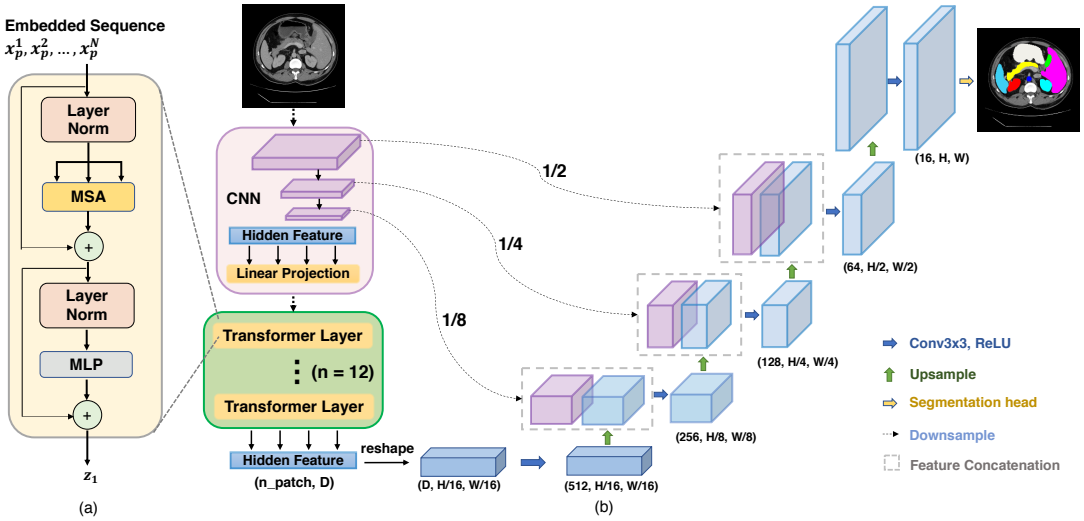


Figure 4.24.: An overview of the TransUNet architecture is illustrated.[8]

model architecture Chen *et al.* [8] achieve better results than using a traditional UNet. The TransUNet has also been successfully applied to US images in the research by Li *et al.* [73].

We use the implementation of the network by Chen *et al.* [8], which is openly available. For our purposes, we reduce the model size of the vision transformer, to roughly achieve the same number of trainable parameters as our UNet architecture. For the model parameters, we choose the configuration of the ViT-B/16. However, to keep the number of trainable consistent with those of our UNet-architecture, for a valid comparison, we reduce the following two model parameters: The hidden size, which refers to the dimensions of the patch embedding, is reduced from 768 to 384; The transformer MLP (see Figure 4.24 (a)) dimension is reduced from 3072 to 768. The resulting model has roughly 32 million trainable parameters.

4.4.3. Inference Using the Physically-Informed Model

The steps of the Inference are shown in Figure 4.22. Given an US image, we can use the trained DL model to predict the corresponding virtual phantom.

There are some post-processing steps applied to the predicted phantom. As established, the predictions of the DL model are standardized by the normalization parameters of the training set, therefore the predicted phantoms need to be rescaled first using equation 4.53.
In the next step, the predicted scattering map is clipped to be in a range of 0 and 1, since values outside of this range do not make sense the way the parameters were defined (see Section 4.1.1). Then the scattering is applied to the density map, by drawing scatterers

using the scattering distribution parameterized by the predicted scattering map. In the last step, the density, SoS and attenuation coefficient maps are also clipped to sensible ranges, based on the maximum values that occur for tissues in the human body:

$$\begin{aligned} \rho &\in [0, 2200] \\ c &\in [0, 4200] \\ \alpha &\in [0, 21] \end{aligned}$$

The predicted virtual phantom can then be simulated using the k-Wave simulation tool, as it was described in Section 4.2.3 to deterministically compute the final predicted RF data.

4.5. End-to-End Deep Learning Model for RF Data Prediction

To evaluate the effectiveness of our physically-informed model, we also employ an end-to-end DL model. The model is designed and trained given an US image to directly predict the corresponding RF data.

To formalize this problem, we can refer to the input image as $\mathbf{I} \in \mathbb{Z}^{N_z \times N_x}$ and the output RF data as $\mathbf{R} \in \mathbb{R}^{N_t \times M}$ where M is the number of channels and N_t is the length of each time signal in R. We note here that the number of channels M is always smaller than the image width N_x, while the signal length N_t is always larger than the image height N_z. For our synthetic data, we have $M = \frac{N_x}{3}$ and $N_t = 2 \cdot N_z$.

Compared to our previous prediction problem, the output data has a different shape for this problem than the input data. For this reason, we need to slightly adapt the model architecture from before. This is done by extending the previous model architecture with an additional tail architecture. The model architecture we use has as a first block either the UNet or TransUNet model. Now instead of using the output of this architecture directly as our model output, we add an additional tail architecture. The tail architecture first resizes the outputted feature map of the TransUNet or UNet to the desired output shape $N_t \times M$ and then follows up with several convolutional layers in the form of two ResNet-blocks (see Figure 4.25). These final convolutional layers should allow the network to recreate the typical time-signal structures at the new higher output resolution.

For the loss function, both the MSE and MAE are valid choices. We choose the sample-wise MAE loss, which can be analogously defined as before:

$$\mathcal{L}_{time}(\mathbf{R_i}, \hat{\mathbf{R}}_\mathbf{i}) = \frac{1}{N_t \cdot M} \sum_{t=1}^{N_t} \sum_{m=1}^{M} |\mathbf{R_i}(t, m) - \hat{\mathbf{R}}_\mathbf{i}(t, m)| \tag{4.54}$$

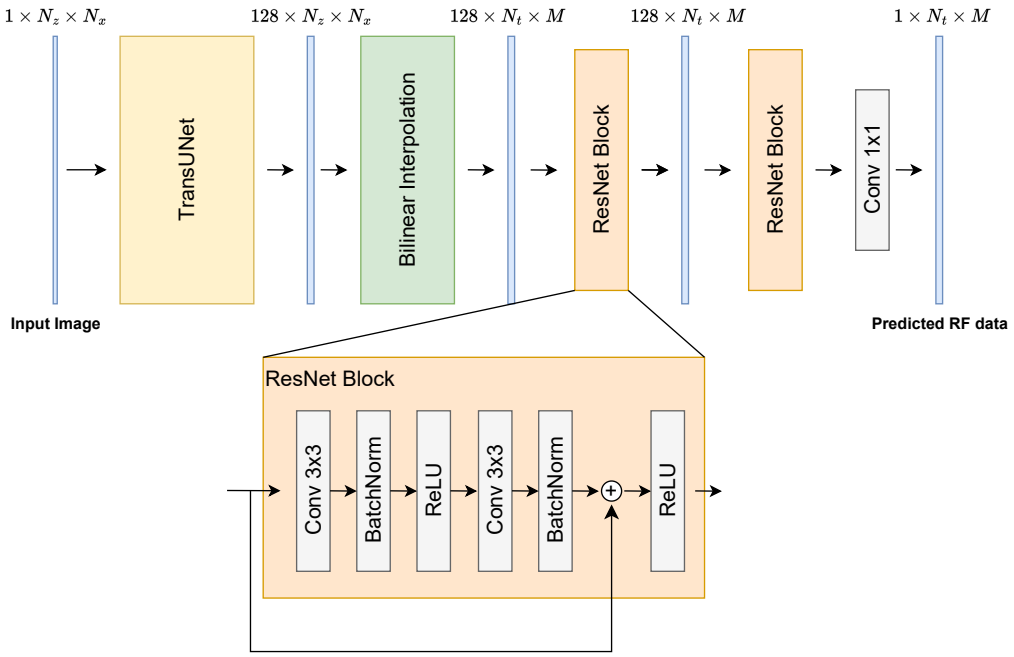


Figure 4.25.: The visualization of the end-to-end model architecture that predicts the RF data directly from the US image.

where $\mathbf{R_i}$ and $\hat{\mathbf{R}}_\mathbf{i}$ refer to the true and predicted RF signal for sample i. Similar to the physically-informed model, we train the end-to-end model on the standardized RF signals and then rescale the predicted signals during inference. The standardization is done across the entire training dataset.

Due to the relevant frequency nature of the target RF signals, an additional loss term can be introduced. We refer to this loss as the spectral loss and it should bias the model to predict RF signals with a realistic frequency spectrum. The spectral loss is defined by us to be the mean squared differences of the spectral magnitudes:

$$\mathcal{L}_{spectrum}(\mathbf{R_i}, \hat{\mathbf{R}}_\mathbf{i}) = \frac{1}{N_t \cdot M} \sum_{f=1}^{N_t} \sum_{m=1}^{M} (|\mathcal{F}\{\mathbf{R_i}\}(f,m)| - |\mathcal{F}\{\hat{\mathbf{R}}_\mathbf{i}\}(f,m)|)^2 \tag{4.55}$$

Here $\mathcal{F}$ denotes the Discrete Fourier Transform (DFT) for the target and predicted RF signals, which is computed using the Fast Fourier Transform (FFT) algorithm [79]. It is worth mentioning that the DFT is a linear mapping involving a simple matrix multiplication. Therefore, the spectral loss term is differentiable and can be optimized using backpropagation.

The total loss function we optimize during training then becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{time} + \lambda \mathcal{L}_{spectrum} \tag{4.56}$$

where λ is a trade-off parameter to balance the two loss terms, and it needs to be determined experimentally. We find $\lambda = 0.1$ works well for our use-case, however, the choice of λ could be further investigated.

Chapter 5

Experiments

5.1. Overview of the Physically-Informed Model Evaluation

To evaluate the accuracy of our physically-informed model for a single sample, we perform model inference as it was described in Section 4.4.3 to get the predicted RF data from an image. Subsequently, we apply image formation to compute the predicted US image. Figure 5.1 illustrates the inference steps of our physically-informed model. We then use selected metrics to compare the predicted and actual RF and US image data. If the true virtual phantom data is available, it can additionally be employed to assess the DL model's performance in predicting the virtual phantom.

In the following sections, we will elaborate on the datasets used to test our trained models and discuss the metrics used to evaluate the accuracy. We will further discuss the different training regimes selected to train our DL models.

5.2. Evaluation Datasets

The evaluation datasets that are used to assess our model's performance are of either of the following two types:

- **Synthetic data**: This dataset is constructed using simulations from our simulator and virtual phantoms.
- **External test data**: This dataset comprises a diverse set of data, including synthetic data generated with an alternative simulator, and real acquisitions from both physical gelatin phantoms and in vivo tissues.

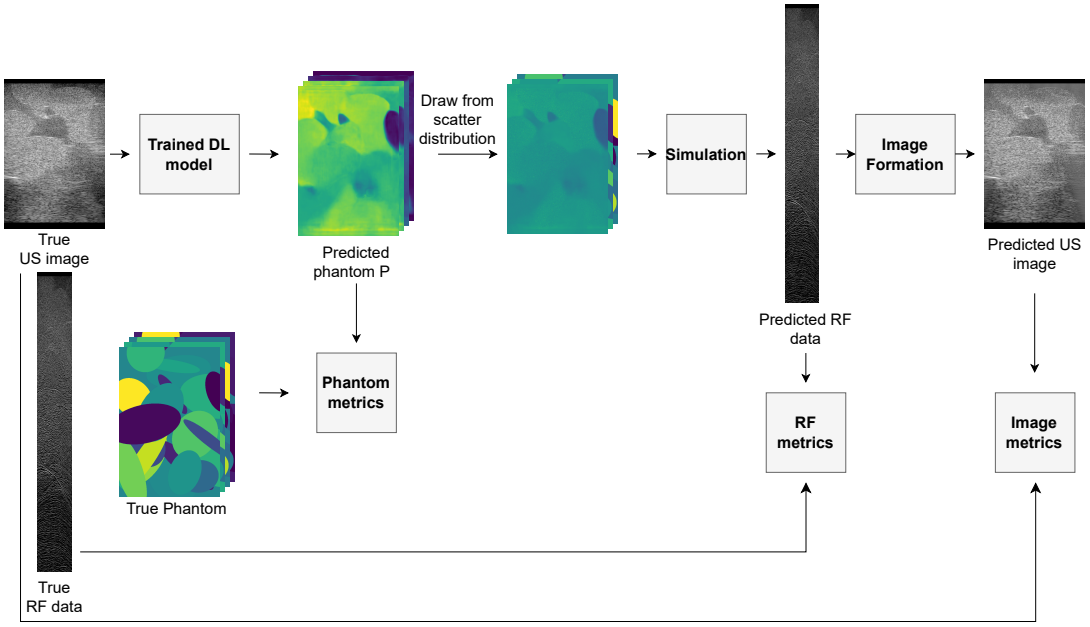


Figure 5.1.: Overview of the evaluation of the physically-informed model's performance for a single sample.

5.2.1. Synthetic Evaluation Set

The test set, containing synthetic data, is generated using the same methodology as the training set, employing our simulator and beamformer consistently. This approach ensures that the test data is perfectly aligned with the training domain, maintaining consistency in the creation of virtual phantoms and the simulation of the ultrasound acquisition. This ensures that no new sources of error are introduced, allowing for a precise and accurate evaluation of the model's performance to generalize and accurately predict within the domain it was trained on.

When evaluating the physically-informed model, this test dataset offers the added benefit of facilitating a direct comparison between the original and the predicted virtual phantoms. Moreover, it allows us to evaluate the impact any errors within the predicted phantoms may have on the simulated RF data and the subsequently formed images.

In practice, the inference and evaluation take around one minute per sample. While we assign a dataset of 5'000 samples to this evaluation set for each dataset, due to timing constraints we limit the size of this dataset to 1'000 samples.

5.2.2. External Evaluation Set

It is crucial to evaluate our model's performance on data acquired from various sources aside from data generated by ourselves: virtual phantoms from different simulation models, acquisitions on real physical phantoms, and importantly, data samples from actual in vivo tissue. This approach enables us to assess how effectively our model generalizes to diverse examples, with a focus on in vivo samples, since many of our strategies aim to lead to a generalization to in vivo tissue data.

When selecting samples for evaluation from different open-source datasets, ensuring consistency in the US imaging method of the selected samples with our simulated US imaging method is crucial. For our objective, we look for RF data from a single plane wave that was acquired using a linear transducer array.

We solely require the RF data, as it enables us to directly form images using our beamformer (see Section 4.2.4). Using a different beamformer could introduce additional sources of error, which we aim to avoid when comparing the original and predicted images.

We obtain such RF data from 3 different sources:

1. **Verasonics System Recordings** [71]: This dataset was acquired in the context of the work of Leitner *et al.*[71]. The dataset can be categorized into two groups: Samples that were either acquired on the CIRS GP phantom model [10] or on in vivo MG muscle tissue. All acquisitions were made using the research US imaging system Verasonics Vantage 256 with a GE 9L-D model transducer.
2. **Plane-wave Imaging Challenge in Medical Ultrasound (PICMUS) 2016**: The PICMUS challenge [43] aimed to assess various beamforming strategies for extracting high-quality ultrasound images from plane wave RF data. The evaluation set from PICMUS, fitting our requirements, consists of four phantom data samples and two in vivo carotid tissue samples. The phantom data is further divided into two simulated samples using the Field II simulator and two samples containing data of real physical phantoms. The platform used to acquire the data is the Verasonics Vantage 256 with the L11-4v transducer.
3. **Challenge on Ultrasound Beamforming with Deep Learning (CUBDL) 2020**: The evaluation dataset from CUBDL [80] consists of a diverse collection of RF data that was acquired on both phantom and in vivo data[42]. For our purposes, we use the dataset provided by the Department of Biomedical Engineering at Tsinghua University, China (TSH) [81]. This dataset features 500 single plane wave recordings of the forearm muscle brachioradialis. Given the similarity among many recordings, likely due to repeated acquisitions on the same subject, we select 6 distinct-looking samples to make the evaluation more efficient. The data was acquired using the Verasonics Vantage 256 US system using a L10-5 transducer.

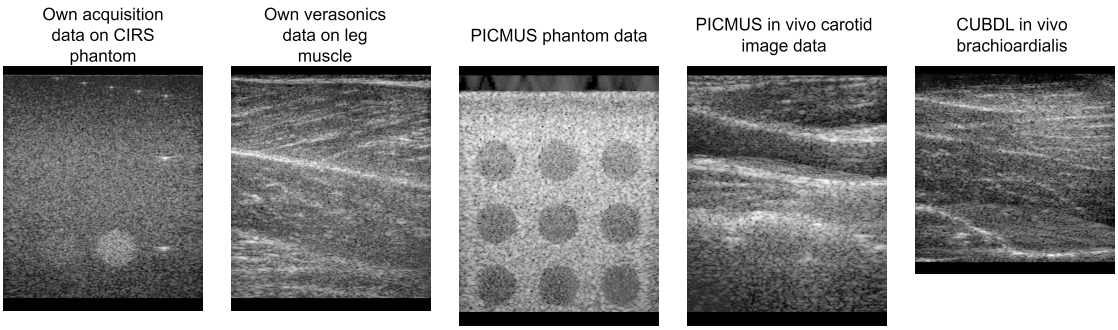


Figure 5.2.: Example images for all 5 external datasets

For each selected dataset, while multiple plane wave acquisitions at various angles are available, we consistently take the acquisitions, where the incident angle of the plane wave is zero.

As an overview, we categorize our external test sets into the following distinct datasets:

- **Verasonics CIRS**: acquisitions on the CIRS GP phantom model [71].
- **Verasonics muscle data** in vivo acquisitions on the MG muscle in the lower leg [71].
- **PICMUS phantom** acquisitions of physical phantoms and virtual phantoms using Field II simulator [43].
- **PICMUS in vivo** in vivo acquisitions of carotid artery data [43].
- **CUBDL in vivo** in vivo acquisitions of the brachioradialis muscle (BM) in the lower arm [82][81].

Example images for all 5 external datasets are shown in Figure 5.2.

This approach ensures a comprehensive evaluation across different types of data, highlighting our model's generalization capability, particularly towards in vivo tissue samples.

One consideration we need to keep in mind for the PICMUS and CUBDL data is that they employ different grid and transducer settings compared to those used for training our model. Therefore, results on these datasets will show the generalization ability of our model across different transducer and acquisition settings.

For our physically-informed model, we need to define a simulation grid for each of the external test sets. The heuristic we have developed to choose the simulation grid and all simulation parameters is explained in the following section.

Finding Appropriate Simulation Grid Specifications

To replicate an acquisition using our simulation environment, we need to define the transducer and simulation grid. Most of our simulation parameters can be read directly from the data sheet of these datasets, however, the simulation grid size is less trivial. A systematic approach to selecting the simulation grid size (N_x, N_z) and grid spacing Δxz is explained here.

To precisely replicate the transducer, the initial step involves choosing a grid spacing Δxz, which allows us to accurately depict the transducer element pitch. For an accurate representation of the transducer in a discrete simulation grid, the element pitch should be a multiple of Δxz.

In our training set, the grid spacing Δxz is equal to the temporal displacement z_{temp}. As a reminder, z_{temp} represents the distance that the sound wave travels in one time step of the sampled RF data, and it is determined by the sampling frequency of our signal. For all the datasets we consider in this project, the sampling frequency is four times the central frequency of the transducers, which results in a temporal displacement $z_{temp} = \lambda/4$, where λ represents the wavelength of the sound wave. Ideally, we choose $\Delta xz = z_{temp}$. However, this does not guarantee that Δxz is a divisor of the element pitch. Therefore, we choose Δxz to be the divisor of the element pitch that is closest to z_{temp}. We can calculate this in the following way:

$$\Delta xz = \frac{\text{element\_pitch}}{n} \quad \text{where} \quad n = \left\lfloor \frac{\text{element\_pitch}}{z_{temp}} \right\rfloor \tag{5.1}$$

Here, n is the number of grid points per transducer element.

We define the width of our simulation grid as the width of the transducer (refer to Section 4.2.3). Thus, to specify the width of the simulation grid N_x, we multiply the number of transducer elements by the number of grid points per transducer element:

$$N_x = \text{n\_transducer\_elements} \cdot n \tag{5.2}$$

Using the provided image depth D, we calculate an appropriate grid height N_z. Then we define N_z such that it is a multiple that closely matches this depth $D \approx N_z \cdot \Delta xz$, which we can compute with

$$N_z = \left\lfloor \frac{D}{\Delta xz} \right\rfloor \tag{5.3}$$

Here, we round down to disregard any excess parts of the provided RF data.

5.3. Evaluation Metrics

The metrics chosen for the model evaluation are of great importance, since conclusions and subsequent decisions are made based on the metrics. For our system, we establish metrics for the RF data we aim to predict, as well as metrics to compare predicted and true US images. While the primary objective is to accurately predict RF data, the beamformed US image also reveals information contained in the RF data. Likewise, for the physically-informed model, we need metrics to evaluate the phantom maps, if they are available (see Figure 5.1).

5.3.1. RF Metrics

Before we go over the RF data metrics, there are two important notes when comparing the RF data signals. First, we always compare the TGC-corrected RF signals. We do this because the RF data provided to us in the external datasets is already TGC compensated since the TGC happens before the analog to digital conversion step in a real US acquisition system (see Section 4.2.4).

Second, we always compare the normalized RF data signals. We do this because the simulated and real RF data have different physical measurement units. The real recorded RF data has unit Voltage while our simulated RF data is recorded in Pascal. To compare them independent of the scale we can compare the normalized signals. The normalized signal $\bar{R}$ is computed using the mean μ and standard deviation σ of the RF data R over all channels:

$$\bar{R}[t, m] = \frac{R[t, m] - \mu}{\sigma} \tag{5.4}$$

Now we will describe the different metrics we use to compare the RF data signals. Because we are the first that try to predict RF channel data from US images, there are no established metrics to evaluate the accuracy of the prediction. We propose the following comparison metrics to compare the true RF data and predicted RF data we denote with $\hat{R}$:

- Time-series mean absolute error (MAE)
- Envelope MAE
- Mean cross-correlation (CC) peak.
- Windowed Root mean square (RMS) difference.
- Power spectral density (PSD) MAE

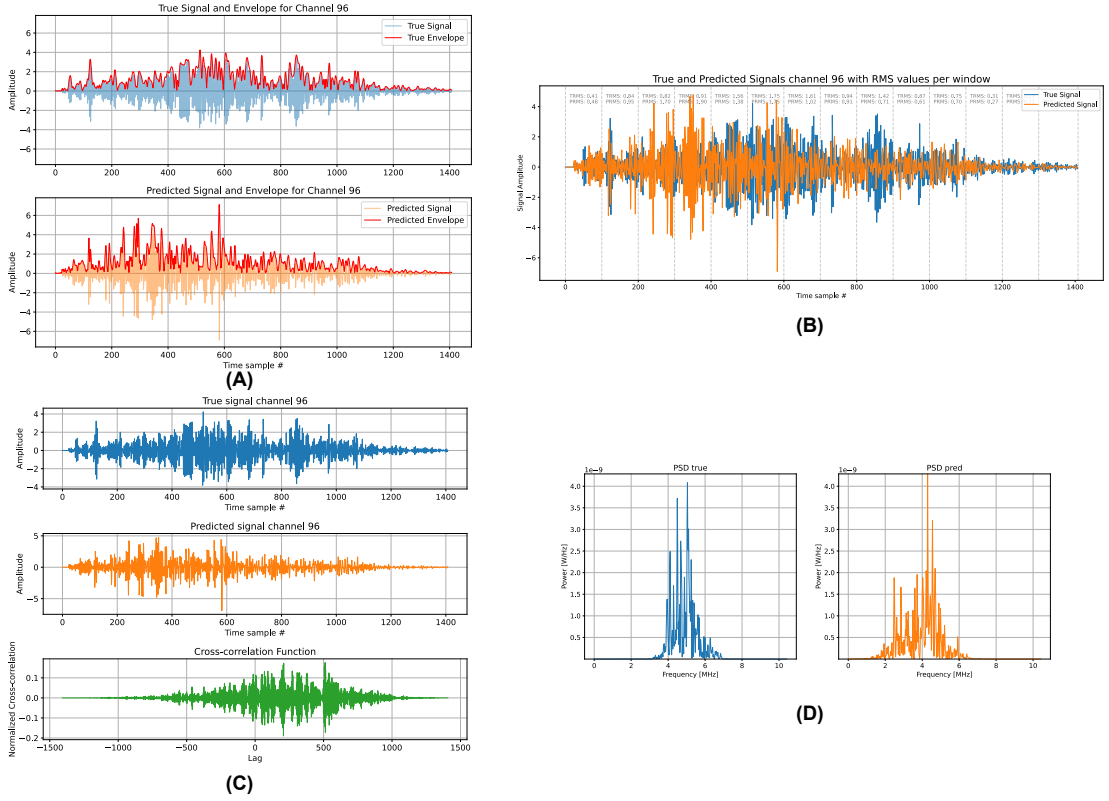


Figure 5.3.: An overview of the RF metrics used showing the computed metrics for a single channel. (A) Here we have the envelopes of the two signals to be compared. (B) This shows the windowed RMS difference. (C) This shows the CC function of the two signals. (D) shows the comparison of the PSD of the two signals.

We denote the true RF data with $R[t, m]$ and the predicted RF data we denote with $\hat{R}[t, m]$, where t is the time sample index which ranges from 1 to T and m is the channel index which ranges from 1 to M, where M is the total number of channels

The first metric we choose is the sample-wise mean absolute error (MAE) to compare the time signals. This takes the absolute difference of the time signals and averages them over the whole time signal and overall channel data:

$$\text{Time Series MAE}(R, \hat{R}) = \frac{1}{MT} \sum_{m=1}^{M} \sum_{t=1}^{T} |R[t, m] - \hat{R}[t, m]| \tag{5.5}$$

This is the most straightforward way to compare two time signals. However, this metric might be misleading due to phase shifts and the periodic nature of our time series. As an example two identical signals where one is phase shifted by 90 degrees would yield very similar images and should be very close, however, they will yield a very high sample-wise MAE. For this reason, we also use additional metrics.

One way to get around this problem is to compare the envelopes of the two RF data signals. The envelope is an important part of US image formation (see Section 4.2.4) and it can be computed using the magnitude of the Hilbert transform H [47]. We can then compute the MAE over the two envelopes:

$$\text{Envelope MAE}(R, \hat{R}) = \frac{1}{MT} \sum_{m=1}^{M} \sum_{t=1}^{T} |E[t, m] - \hat{E}[t, m]| \quad \text{with} \quad E[t, m] = |H(R)[t, m]| \tag{5.6}$$

The envelopes of an example signal are shown in Figure 5.3 (A).

Another method to compare the similarity of two time signals is the cross-correlation (CC) function. The CC function is defined as the product of the two functions, where one is displaced by the lag τ. We can calculate it for each channel m using the operator $\star$ [83]:

$$((R[m] \star \hat{R}[m])[\tau] = \sum_{t=1}^{T} R[t, m] \cdot \hat{R}[t + \tau, m] \tag{5.7}$$

As it is common practice we use the normalized cross-correlation function. It gives an easier interpretation of maximum correlation at 1 or -1 and minimum correlation at 0 and can be seen as a time-dependent Pearson correlation coefficient [83]. To extract a single number for this metric, we take the maximum absolute value for each channel and then take the average peak CC over all channels:

$$\text{Average CC peak}(R, \hat{R}) = \frac{1}{M} \sum_{m=1}^{M} \max_{\tau} \{|(R[m] \star \hat{R}[m])[\tau]|\} \tag{5.8}$$

An example of the cross-correlation can be seen in Figure 5.3 (C).

We can also compare the root mean squares of the two RF signals. For a voltage signal, the RMS is closely related to the power of the signal [84]. Because our RF signals can be scaled to voltage, the RMS can also be used to compare the power of the two RF signals. Because the location of large echoes is important to us in our RF data, we want to keep some spatial information about the location of the power in the signals. Therefore we divide the RF signals into windows and compute the RMS over each window w:

$$\mathrm{RMS}_w(R[m]) = \sqrt{\frac{1}{T_w}\sum_{t}^{T_w} R[t,m]} \tag{5.9}$$

Here T_w denotes the number of samples in a window We can then compare the absolute differences of the RMS over the windows and average them over all windows and all channels:

$$\text{Mean RMS differences}(R,\hat{R}) = \frac{1}{MN_w}\sum_{m=1}^{M}\sum_{w}|\mathrm{RMS}_w(R[m]) - \mathrm{RMS}_w(\hat{R}[m])| \tag{5.10}$$

where N_w is the number of windows per channel. We use as a heuristic a window size T_w of 100 samples. This is illustrated in Figure 5.3 (B).

Finally, we also compare the two signals in the frequency domain, as it is also an important feature of the RF data. To compute the frequency power spectra for each channel we use Welch's method [85] over just a plain FFT, as it is less sensitive to noise and reduces the variance. Then we compare the two power spectral densities (PSD) using the MAE and average it over all channels. Such PSDs are portrayed in Figure 5.3 (D).

5.3.2. Image metrics

The images that are constructed from the predicted RF data also hold value, since they reveal intuitive information that is hidden in the RF signals and allow for a visual comparison. To compare the predicted and true US image, we can rely on metrics that were introduced and used by previous works such as by Zhang *et al.* [27] to compare the predicted and true US images. In addition, more advanced established image similarity metrics are used. Here is an overview of the metrics we measure when comparing the true and predicted US image:

1. Pixel-wise Root Mean Squared Error (RMSE)
2. Mean image intensity difference
3. Signal-to-noise ratio (SNR) difference
4. Patch-wise KL divergence
5. Structural Similarity Index (SSIM)

6. Perceptual Loss [86]

The first metric we choose is the pixel-wise root mean squared error defined as:

$$\text{RMSE} = \frac{1}{N} \sum_{j=1...N} s_j \tag{5.11}$$

This is a straightforward way to compare the individual pixel intensities, which can be a useful indicator of image similarity. However a mismatch between the speckling patterns between two images, due to differently drawn scattering patterns can increase this. Additionally, the RMSE cannot capture the structural similarity between the two images.

The next three metrics are chosen analogously to Zhang *et al.* [27]. We rely on their metrics, since the model they have developed similarly to ours also predicts US images.

First, we have the mean image intensity difference (I), which is defined using the following formula:

$$\Delta\text{I} = \frac{|\text{I}_t - \text{I}_p|}{\text{I}_t}, \quad \text{with} \quad \text{I} = \frac{1}{N} \sum_{j=1...N} s_j \tag{5.12}$$

The subscripts p and t denote predicted and true US image, s_j denotes pixel intensity for the j-th pixel and N are the number of pixels. This metric captures global intensity shifts in the images.

As proposed by Zhang *et al.*[27], we can also capture the global SNR of the images and compute their difference:

$$\Delta\text{SNR} = \frac{|\text{SNR}_t - \text{SNR}_p|}{\text{SNR}_t}, \quad \text{with} \quad \text{SNR} = \frac{\mu}{\sigma} \tag{5.13}$$

Here μ and σ reference the global mean and standard deviation of the image intensities.

They also measure the patch-wise local intensity statistics, by computing the Kullback-Leibler (KL) divergence on intensity histograms of image patches[27]. These intensity statistics are of importance for downstream tasks of tissue characterization and should be accurately replicated [87] [88]. The KL divergence can be computed using

$$\text{KL}(h_p||h_t) = \sum_{l=1...D} h_p[l] \log\left(\frac{h_p[l]}{h_t[l]}\right) \tag{5.14}$$

where h denotes the intensity histogram and D is the number of histogram bins, where $D = 50$ bins are chosen. The KL divergence is a common tool in statistics to compare two probability distributions. If the distributions match perfectly, the KL divergence is zero. Higher KL divergence indicates a higher mismatch between the two distributions [89]. Just as Zhang *et al.*[27] propose, we choose a patch size of 10 wavelengths λ of our ultrasound pulse in both directions. A visualization of the comparison of the intensity

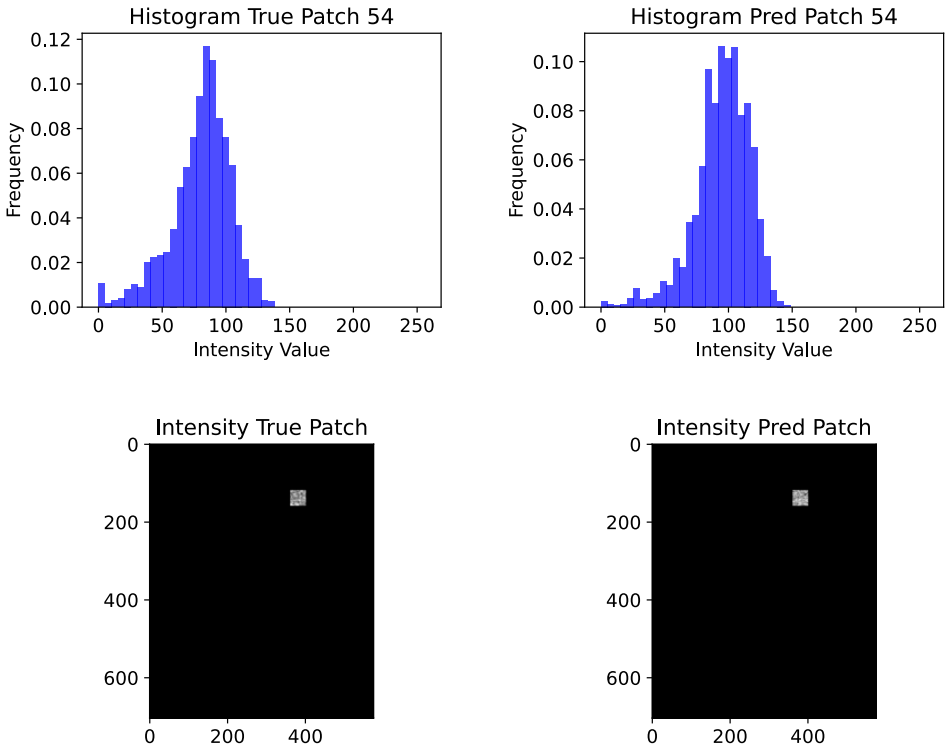


Figure 5.4.: Example for patch-wise the histogram comparison using the KL divergence. Here illustrated are the predicted and true image histograms that are compared using the KL divergence for one of the patches.

histogram in an image patch is shown in Figure 5.4. The final value we report is the mean KL divergence overall image patches.

In addition to the metrics introduced by Zhang *et al.*, we also use the following two metrics to capture structural differences between the two images. One such metric is the Structural Similarity Index Measure (SSIM) [90]. It is a metric based on human visual perception and tries to quantify and combine the concepts of luminance, contrast, and structure. Formally it is defined using the following formula:

$$\mathrm{SSIM}(x,y) = l(x,y)^{\alpha} \times c(x,y)^{\beta} \times v(x,y)^{\gamma} \tag{5.15}$$

$$= \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right)^{\alpha} \times \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right)^{\beta} \times \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right)^{\gamma} \tag{5.16}$$

where the luminence term is denoted with l, the contrast term with c and the structure

term with s. To go into detail about the meaning of each term would go beyond the scope of this report.

Another advanced metric we choose to compare images is the perceptual loss [86]. The perceptual loss leverages deep features extracted by a NN trained for image classification. These deep features contain high-level and contextual information about the images that other metrics such as the SSIM cannot capture [86]. We use their implementation and the default trained AlexNet [72] they provide is used to extract the features.

5.3.3. Phantom Comparison

If the true phantom is available, we can also compare it with the predicted virtual phantom. Note that this is only the case for us for the synthetic evaluation set we have created ourselves, since we do not know the virtual phantoms for the real acquisitions in the external evaluation set.

For the phantom comparison, we can use both the pixel-wise RMSE and the patch-wise KL divergence analogously to the Image metrics in the previous section. Here we report both metrics for each of the four phantom values individually. This choice is made, because we want to evaluate how well our DL model performs for each of the phantom maps and see if certain acoustic properties are harder to predict.

5.4. Training of The Physically-Informed model

The physically-informed model we design leverages a DL model to predict the phantom maps. In this section, we specify the training datasets used for training the DL models and we describe the training specifications used.

5.4.1. Overview of Training Datasets

We create large virtual phantom datasets for the following phantom types that we have developed in Section 4.3 :

- Ellipse Phantoms
- ImageNet Phantoms
- ImageNet-enhanced Phantoms
- CT Phantoms
- Muscle Phantoms

For each of these datasets, except for the muscle phantoms, we create 50'000 data samples using the respective virtual phantom generation strategy, followed by the data generation steps. The CT phantom set is a bit smaller, however, since we filter samples that have bone tissue at the transducer lens (see Section 4.3.4). This removes approximately 17% of the data samples. Since the muscle dataset is only used for fine-tuning a trained model, we choose a smaller size of 10'000 samples.

We distribute these data samples at random into a training, validation, and test set. The training set is used for training our DL model, while the validation set is used to track the validation loss during training, to tune training hyperparameters and to select the best model. The test set is used in the end for evaluating the performance and makes up the synthetic evaluation set discussed in Section 5.2.1. We assign 80% of the total data samples to the training set and 10% each for the validation and test set. The dataset sizes for all the datasets we use to train our DL model from scratch are presented in Table 5.1.

Dataset	Ellipse	ImageNet	ImageNet-enhanced	CT	Combined
Training Set	40,000	40,000	40,000	33,351	153,351
Validation Set	5,000	5,000	5,000	4,151	19,151
Test Set	5,000	5,000	5,000	4,151	19,151

Table 5.1.: The size for each of the datasets that we use to train our DL model from scratch and how they are distributed across the training, validation and test set.

Then we train a TransUnet model (see Section 4.4.2) from scratch for each training dataset except for the Muscle phantoms, since they are used for fine-tuning.

We also decide to train a model that learns from all four phantom types. For this reason, a model is trained on what is referenced as the Combined dataset, which is the union of all the previous datasets and contains all the data samples generated. This model is therefore trained on the largest dataset and can potentially learn from all the information introduced in our different datasets.

After training a NN from scratch for each of the large datasets in Table 5.1, we can then fine-tune the trained models on the muscle phantom datasets. This gives us a total of ten trained models, five models that were trained on larger datasets from scratch and for each a model that was fine-tuned on the muscle data. An overview of this can be seen in Figure 5.5

5.4.2. Training Specifications

Here we briefly describe the training hyperparameters for each of the DL models we train to predict the virtual phantom. These hyperparameters were experimentally found by

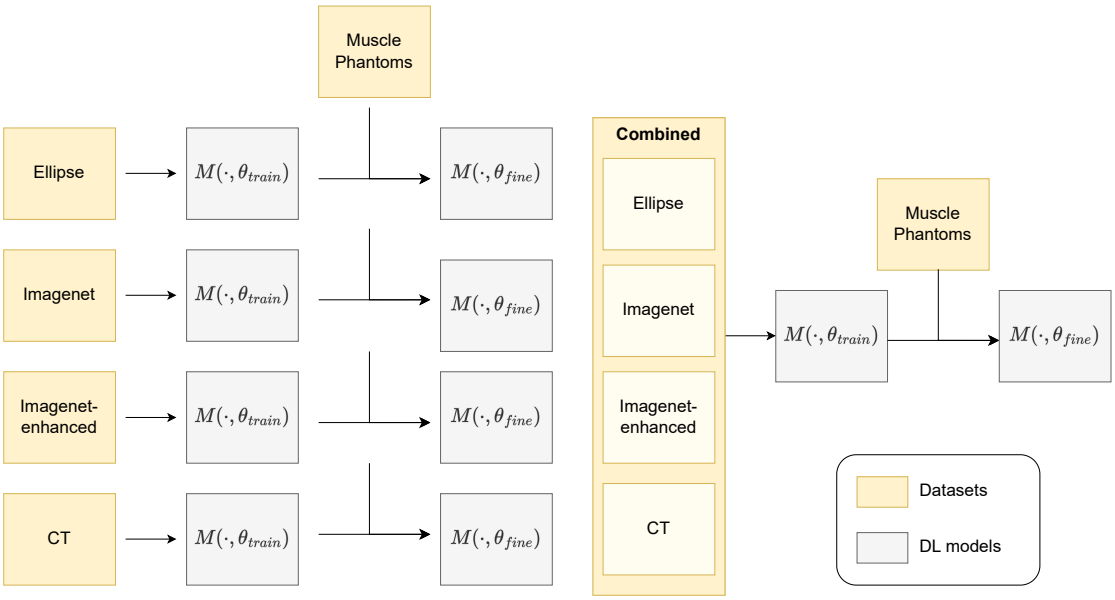


Figure 5.5.: An overview of the DL models we train and on what dataset they are trained on.

trying a variety of different hyperparameter configurations, and to that extent, we have implemented a grid search.

The model architecture we found performs best is the TransUNet model architecture as it is described in Section 4.4.2.

For the loss function, we choose the MAE loss (see Section 4.4.1) and we optimize it using the popular Adam optimizer [91]. We choose an initial learning rate of $5 \cdot 10^{-5}$ and we decrease the learning rate by a factor of 0.8 if the training loss does not decrease for five consecutive epochs. We found that a batch size of 8 worked the best for us. However, due to memory limitations on the GPU, we were not able to try higher batch sizes, which might be interesting to explore.

To artificially increase the dataset size and prevent overfitting, we use the concept of data augmentation. For the data augmentation methods, we choose techniques analogous to the work of Leitner *et al.* [16]. They also used augmentation techniques specifically for US images, which we can therefore make use of for our model. Specifically, we apply the following augmentations:

1. We randomly crop the image by 10% of the image size, while keeping the original image dimensions by resizing the images afterwards.
2. With probability 0.5 we flip the image horizontally.
3. With probability 0.5 we apply random rotations by $\pm 20°$.

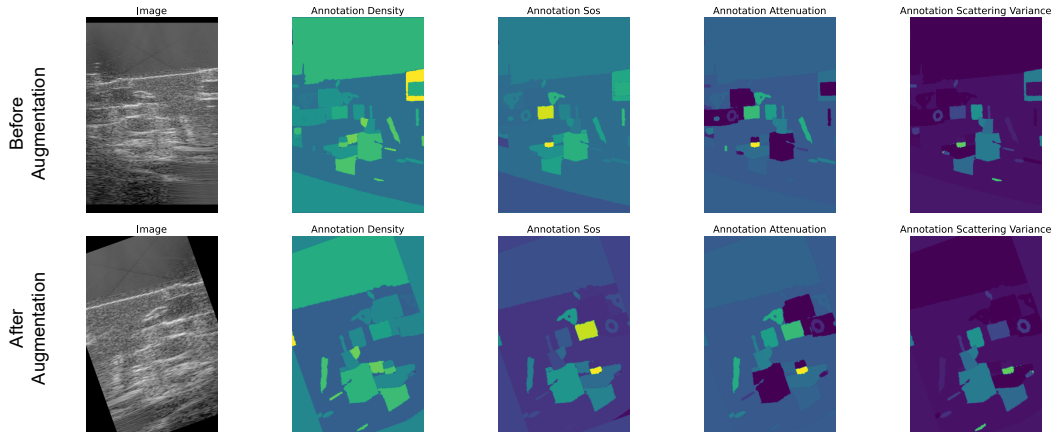


Figure 5.6.: An example of the augmentation applied to both the input US image and the target virtual phantom.

4. With probability 0.5 we apply a random affine transformation that includes: translations in both image directions by 10%; zooming the images by scale factors in the range [0.7, 1.3]; randomly shearing the image in the range [-0.2,0.2].

For all augmentations, it is important to apply the same transformation to both the image and the virtual phantom. An example of the augmentation is shown in Figure 5.6.

We implement the training pipeline in Python using the PyTorch library [92]. All models are trained on the GPU clusters provided by the computer vision lab at ETH. The models are trained on two GPUs at a time.

The models for the **Ellipse phantom**, **ImageNet phantom**, **ImageNet-enhanced phantom**, and the **CT phantom** dataset are trained for 75 epochs each. The model for the **Combined** dataset is trained only for 25 epochs, where it is important to consider that one epoch considers updates over the entire dataset. This means training one epoch on the combined dataset leads to around 4 times as many parameter updates compared to training on the other datasets, because it contains roughly 4 times as many data samples. This takes more time, the combined dataset was trained for around 8 days compared to the other datasets where training takes around 6 days.

The fine-tuning of the models on the muscle dataset is done using the same training parameters, but only for 10 epochs. However, here we use the mean squared error (MSE) instead of the MAE, since it experimentally gives us better results.

5.5. Training of the End-to-End Model

The end-to-end model is trained on the Ellipse phantom dataset for 25 epochs using the model architecture and loss function described in Section 4.5.

The same initial learning rate as for the physically-informed model of $5 \cdot 10^{-5}$ is chosen and the loss function is also optimized using the Adam optimizer [91]. For the learning rate schedule, the same specifications are chosen as for the physically-informed model.

For the data augmentation, we only use random horizontal flip and do not use the other augmentation techniques, that are used for the physically-informed model. Rotations and affine transformations are avoided, since the transformation cannot be accurately translated for the RF data. Random cropping could be applied, however, this would reduce the bias of the TGC function, which we also want the model to learn.

Again the training is implemented using PyTorch [92] and the model is trained on the GPU clusters provided by the computer vision lab at ETH.

Chapter 6

Results

Here, we present and discuss the results of the experiments described in the previous chapter. Initially, we discuss two experiments to get a better understanding of the limitations of our approach. Subsequently, we will look at the performance outcomes of training our DL models on the synthetic datasets we have created. In the end, an ablation study will be conducted, providing explanations for the selected loss function and model architectures.

6.1. Data Generation Validation

Our aim is to demonstrate the efficacy of our virtual phantoms and simulations in replicating real-world acquisitions. We utilize the physical CIRS phantom model because of its known physical dimensions and target placements. Employing a virtual phantom, we accurately replicate the physical phantom's characteristics. This enables us to compare the real acquisitions on the physical phantom to the simulated data from the virtual counterpart.

This experiment establishes the maximum accuracy with which our physically-informed system can recreate real RF data and US images. Here, we hand-design the virtual phantom to match the physical phantom as closely as possible. Therefore, we cannot expect our results, derived from a DL model predicting a virtual phantom, to exceed this benchmark.

Additionally, this analysis can also be useful for calibrating different model parameters utilized during our synthetic data generation, such as the TGC waveform of the signal (see Section 4.2.4).

For this experiment, we use a physical phantom model. Physical phantom models are used in practice for calibration and testing the performance of ultrasound acquisition

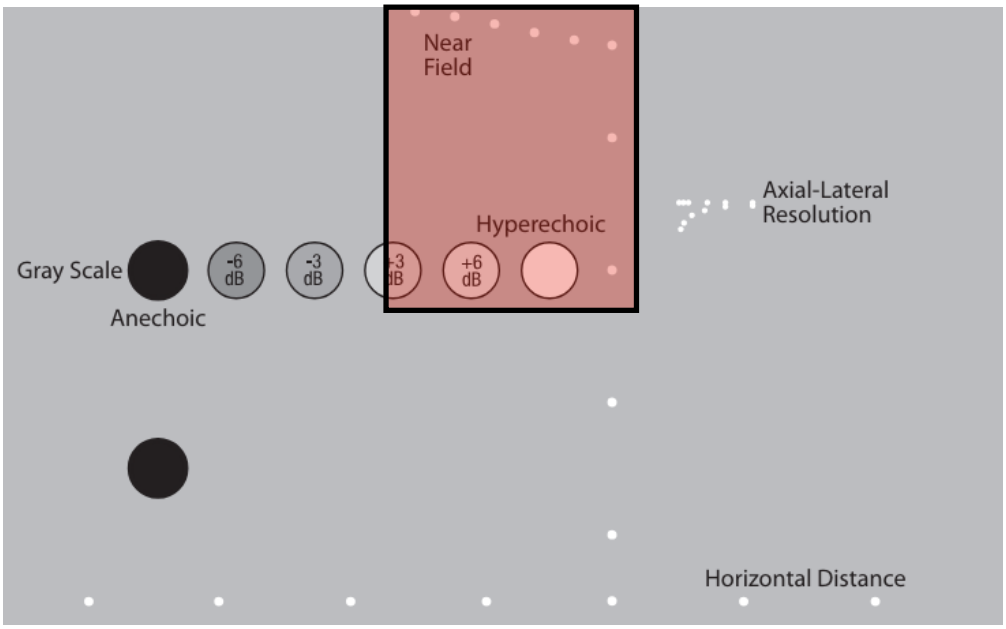


Figure 6.1.: Acquired section of the CIRS GP phantom model

systems. By utilizing the CIRS phantom model we know thanks to a data sheet the physical size and locations of all the targets inside the medium. This enables us to recreate a virtual phantom with the identical measurements. For the grayscale level targets in the phantom, the only values provided are the dB level of the target with respect to the background.

Initially, we implemented this experiment by acquiring data ourselves on the CIRS Model 040GSE Multi-Purpose, Multi-Tissue Ultrasound Phantom [93] using an Ultrasonix acquisition system. However, this presented an issue, since the returned signal data from the acquisition was already beamformed by the beamformer of the Ultrasonix machine. We aim to compare the original RF data, which prompts us to use a different dataset.

To avoid above-mentioned issue, we use the RF data on the CIRS Model 054GS General Purpose Ultrasound Phantom (CIRS GP) [10], which was collected on the research US acquisition system Verasonics Vantage 256 using a GE 9L-D model transducer in the work of Leitner *et al.* [71].

We recreate the CIRS GP using a virtual phantom. The geometric locations and size of all targets are known from the provided data sheet [10] and are visualized in Figure 6.1. The background values are also known, apart from the attenuation and scattering, for which we opt to utilize the standard values available for the synthetic datasets. The chosen background parameters are presented in Table 6.1.

To recreate the small point targets, we increase the acoustic impedance by increasing both SoS and density by 25% w.r.t the background at the specified locations. For the

Acoustic properties	Value with Units
ρ_b	900 kg/m^3
c_b	1540 m/s
α_b	0.54 dB/(MHz·cm)
s_b	0.000625

Table 6.1.: Background phantom values for the CIRS GP Phantom [10].

grayscale targets, we only increase the scattering. We calibrate the scattering values of these targets, aiming to closely match the appearance and the contrast-to-noise ratio (CNR) of the real acquired targets. This resulting in scattering values s are presented in Table 6.2.

Target	Scattering parameter s
+3dB target	0.00069
+6dB target	0.0012
Hyperechoic target	0.0077

Table 6.2.: Scattering values s for the grayscale targets in the CIRS phantom.

The transducer specifications and simulation grid are the same as the ones for the synthetic data setup specifications provided in tables 4.1 and 4.2 in Section 4.2.1, except for the US image to have less depth with $N_z = 704$. The depth is chosen lower, as the RF data contains fewer samples. To identify the exact location of the transducer on the GP phantom, we measured the pixel distances of the known targets to the sides of the image. The resulting phantom can be seen in Figure 6.2.

To compare the real and simulated RF data and US images, we employ the evaluation metrics introduced in Section 5.3. Since the exact locations of all the grayscale and point targets are known, two additional metrics can be calculated. The contrast-to-noise ratio (CNR) is a good indication of image quality and is computed for each known grayscale target with [94]:

$$\mathrm{CNR} = \frac{|\mu_i - \mu_o|}{\sigma_i + \sigma_o} \tag{6.1}$$

where μ_i, μ_o, σ_i and σ_o denote the means and standard deviations of the image intensities within an area inside the target and an area outside the target.
In addition, using the point targets we can compute the axial and lateral resolution by computing the full width at half maximum (FWHM). The reported resolutions are averaged across all point targets.

The resulting RF and image metrics between the real and simulated US image is presented in Tables 6.3 and 6.4.

The acquired and the simulated images are compared in Figure 6.3. They are visually confirmed a close match, which shows that our data generation pipeline is capable of

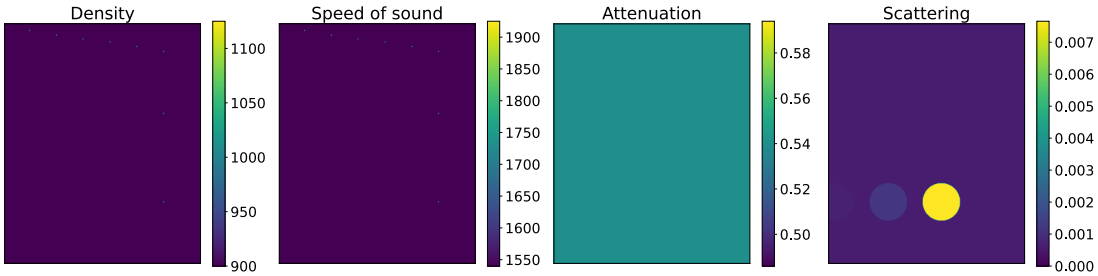


Figure 6.2.: Virtual phantom designed to replicate the physical CIRS phantom

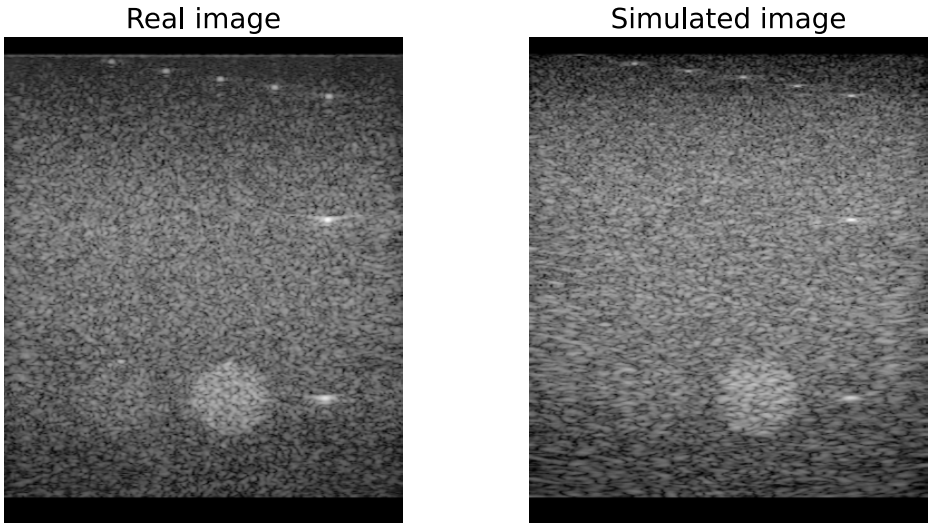


Figure 6.3.: Comparison of the US image of the real acquired RF data on the left and the US image of the simulated RF data on the right.

approximating and yielding visually close results.

Because we know the locations of the point and gray-scale targets, we can compare the CNR of the grayscale targets and the axial and lateral resolutions between the real and acquired images. The results are presented in Tables 6.5 and 6.6.

For the CNR, we get higher values for the real US image, indicating a better image quality for the real data. This discrepancy might be caused by our choice of background scattering parameter or by our choice of the scattering density. This could be further investigated in the future.

For the average axial FWHM of the real US image that we report in Table 6.6, we should note here that the average is heavily disturbed by two outliers. For the two outliers, the calculation of the FWHM seems to be faulty. By disregarding the outliers, we get an average axial FWHM of 0.47mm, which is much closer to the average axial resolution of

RF Metrics	
MAE	0.95
Envelope RMSE	1.03
Cross-Correlation Metrics (Average Peak)	0.16
RMS Difference	0.28
PSD MAE	2.59×10^{-10}

Table 6.3.: Summary of RF metrics for the manually created CIRS validation.

Image metrics	
Mean Intensity Difference	0.11
SNR Difference	0.00058
SSIM	0.15
RMSE	33
Perceptual Loss	0.11
Mean KL Divergence	0.22

Table 6.4.: Summary of image metrics for the manually created CIRS validation.

the simulated image. Either way, the simulated images seem to get a higher resolution both in axial and lateral directions.

These values in tables 6.3 and 6.4 can be used as a baseline when comparing the predicted values for the predicted RF and image data our physically-informed model generates.

As we can see from our results, we do not get a perfect match between the real and simulated RF data and the resulting US image. disparity exists between the real and simulated data and various possible sources for these discrepancies are listed below:

- One reason for the gap could be due to the imperfect representation of the physical world by our virtual phantom model. One contributing factor is the lack of knowledge of the actual scatterer pattern, which we approximate using our scattering model. In addition, the virtual phantom is only a discrete approximation of the continuous acoustic medium. Choosing a higher grid resolution might further close this gap, however, this would lead to longer simulation times.
- We approximate the 3D model using a 2D simulation grid. This simplification is another source that could introduce discrepancies between the real and simulated recordings. Introducing a third dimension is possible, however, this increases the simulation time significantly.
- Our simulation grid stops at the border of the virtual phantom, whereas in reality, the areas beyond the virtual phantom also contribute to the echoes received by the transducer. Extending the simulation grid beyond the virtual phantom could mitigate this issue while leading to longer simulation times.

Target	CNR True	CNR Predicted	CNR Absolute Difference
Plus 3dB Target	-26.52	-62.32	35.80
Plus 6dB Target	-5.12	-7.68	2.56
Hyperechoic Target	3.79	2.73	1.06

Table 6.5.: CNR Metrics for the real and simulated US image for the CIRS validation.

Metric	Real image	Simulated image	Absolute difference
Avg Axial FWHM	4.02mm	0.26mm	3.76mm
Avg Lateral FWHM	1.15mm	0.97mm	0.19mm

Table 6.6.: The average FWHM metrics for the real and simulated US image for the CIRS validation.

- The transducer modeled in our simulation grid does not account for the kerf between transducer elements, for efficiency reasons (see Section 4.2.3). This omission could also contribute to the gap.
- Another source of error is the simulation model itself. For example, because we do not model the non-linear properties of the acoustic medium, they are also not considered.
- Another contributing factor is the uncertainty surrounding the specific TGC waveform applied to the signal by the acquisition system. We empirically calibrate the TGC for the Verasonics system in this experiment, by manually adjusting the control points to closely align the synthetic and acquired RF data and images. A similar calibration should be done for a new machine if the TGC waveform is not provided.
- Another source of error stems from our limited knowledge regarding the exact input wave function of the transducer; we approximate it using a pulse with 3 cycles in length.

One of the things we noticed in the simulated image compared to the real image is the triangular areas in the bottom right and left corner of the image, which seem to be somehow distorted (see Figure 6.3 right). We hypothesized that this might be connected to the simulation grid ending right at the border of the transducer. To test this, we expand the virtual phantom in the simulation using edge padding, where we repeat the edge values on both sides of the phantom. However, we find that this does not remove the triangular artefacts while increasing the simulation time.

6.2. Visual Effect of Each Acoustic Property

Using our physically-informed model we try to predict 4 different acoustic properties for each pixel. In this experiment, we evaluate the difficulty of predicting the individual virtual maps density D, SoS C, attenuation A and scattering S.

The purpose of this experiment is to explain discrepancies in results. The scattering and attenuation regularly achieve lower normalized validation losses compared to the density and SoS losses, which indicates that they are easier to determine.

The intensity of a pixel in a US image depends on the reflected pressure amplitudes from that physical point. Further, the density and SoS values can not be predicted solely from the reflected pressure echoes of an image. As a reminder (see Section 2.1.1), when a sound wave passes from medium 1 to medium 2 the reflected pressure amplitude is described by the formula [1]:

$$\frac{p_r}{p_i} = \frac{Z_2 - Z_1}{Z_1 + Z_2} \quad \text{where} \quad Z_i = \rho_i c_i \tag{6.2}$$

where p_r is the reflected and p_i the incident pressure amplitude and Z_i are the medium acoustic impedances. Therefore, knowing the reflected pressure amplitude, we have one equation and two unknowns in formula (6.2), which does not allow for an exact solution for both acoustic impedances Z_1 and Z_2.

There also exists the ambiguity that a high reflected pressure amplitude might either indicate a switch to lower or to higher acoustic impedance in equation 6.2. In addition, even if we could predict the acoustic impedances exactly, we could not predict exact values for density and SoS since we have two unknowns with only one equation.

In attempt to evaluate these ambiguities and the difficulty predicting various acoustic properties from the US image, we have devised the following experiment: We create a template virtual phantom with a single circular target placed in the middle of a standard background (see background values as in Table 4.4). Then, for each phantom value ρ, c, α and s we create two phantoms. One phantom where we increase the phantom value for the target shape and one where we decrease the phantom value.

Because the different phantom maps are on different scales and have different variances, the choice of the increase and decrease is not trivial. Here we increase and decrease the maps by 3 standard deviations of the respective phantom value. These are the same standard deviations, that we have calculated in Section 4.3.3.

Then we can simulate all eight virtual phantoms to get corresponding US images. We can now compare the two images for each acoustic using our regular image metrics. Here, we select the RMSE and the perceptual loss. If the images are very similar for an acoustic property, we hypothesize it to be harder to predict, since the change does not have a big impact on the visual appearance of the image. If the two images are very different,

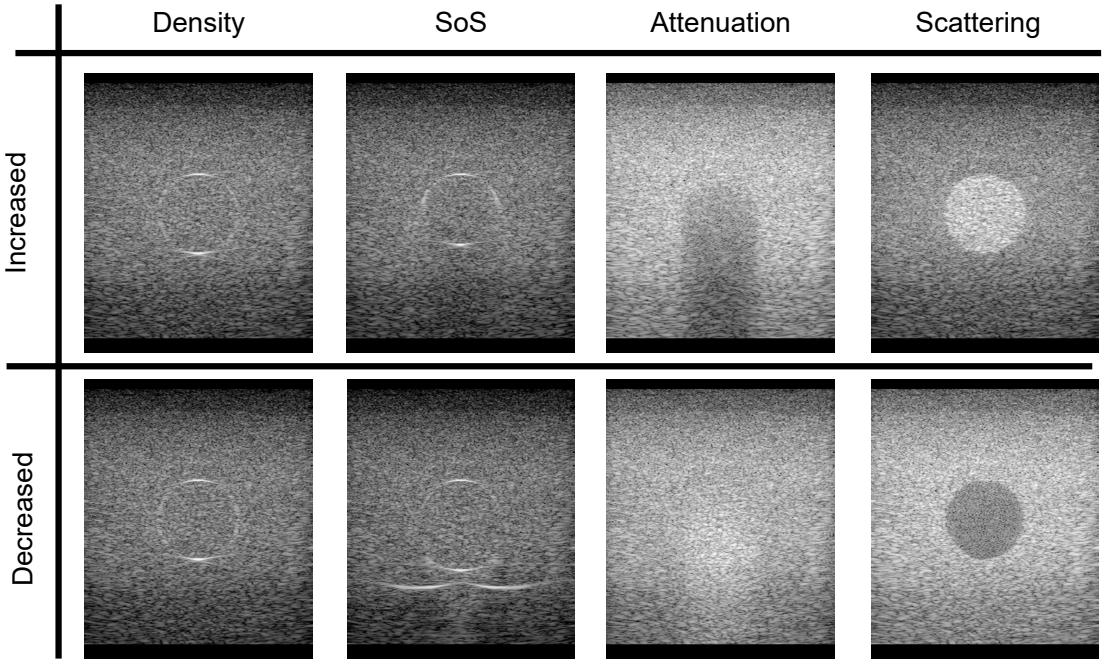


Figure 6.4.: Overview of the resulting images after either increasing or decreasing a specific acoustic parameter by the respective standard deviation.

we hypothesize it to be easier for a learning algorithm to pick up on the differences and predict the correct phantom value.

The resulting images are shown in Figure 6.4 and the quantitative resulting image metrics are shown in Figure 6.5. It becomes clear that the two density images look very similar even though the density values are very different. We conclude that this ambiguity makes the prediction harder. Meanwhile, for the scattering and attenuation, a clear distinction between the two images can be seen, due to the different pixel-wise intensities of the targets. This is also reflected in the quantitative results in Figure 6.5, where the image RMSE and perceptual loss are lower for the density than for the scattering and attenuation, letting us conclude they are harder to distinguish. For the SoS, while the reflection echo intensities are similar, a distinction can be made due to the speed displacement artefacts (see Section 2.4). This distinction is reflected in the high perceptual loss.

Additionally, it becomes apparent that the acoustic impedance differences for a target are only seen in the received echo at the target boundary regions, while scattering and attenuation affect all the pixels in the target. Therefore, to predict the density and SoS of pixel P, we need more context than to predict scattering or attenuation at pixel P.

We can therefore find an explanation for why certain acoustic properties are harder to predict for each pixel than others.

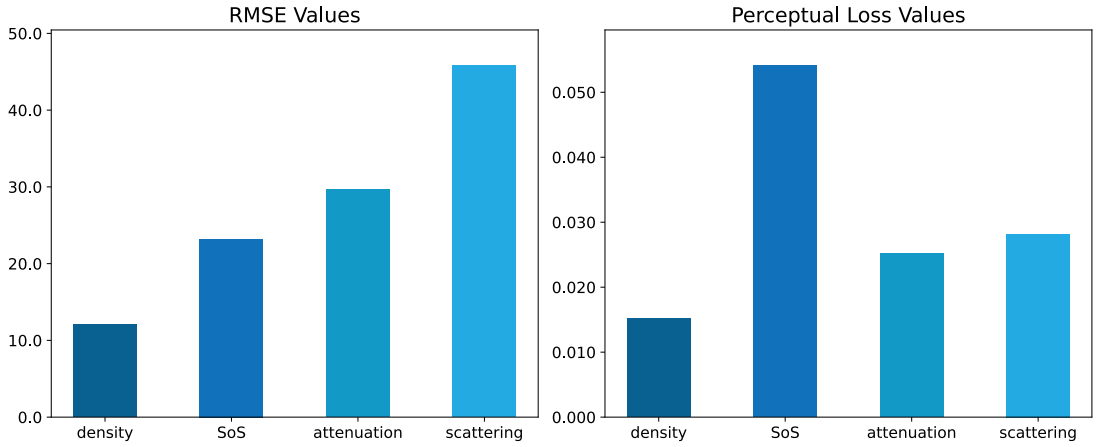


Figure 6.5.: Resulting image metrics for each phantom value. For each acoustic property, the images of the increase and decrease of the acoustic property are compared. On the left, we have the RMSE between the images and on the right the perceptual loss of the two images.

6.3. Physically-Informed Model Results

In this Section, we evaluate the models that we have trained on the five main datasets. These datasets and the training model architecture and specifications were established in Section 5.4. We reference them in the following using these names:

- **Ellipse model**: The TransUnet model trained on the Ellipse phantom dataset.
- **ImageNet model**: The TransUnet model trained on the ImageNet phantom dataset.
- **ImageNet-enhanced model**: The TransUnet model trained on the ImageNet-enhanced phantom dataset.
- **CT model**: The TransUnet model trained on the CT phantom dataset.
- **Combined model**: The TransUnet model trained on the Combined phantom dataset.

First, we investigate the validation loss curves during training for each model to see how well the models converge. Then we can evaluate the performance of each model on the corresponding synthetic datasets, where each such set contains synthetic images we have generated from the same phantom types as the model was trained on. Finally, we can evaluate and compare the generalization capabilities of our trained models on the external test sets.

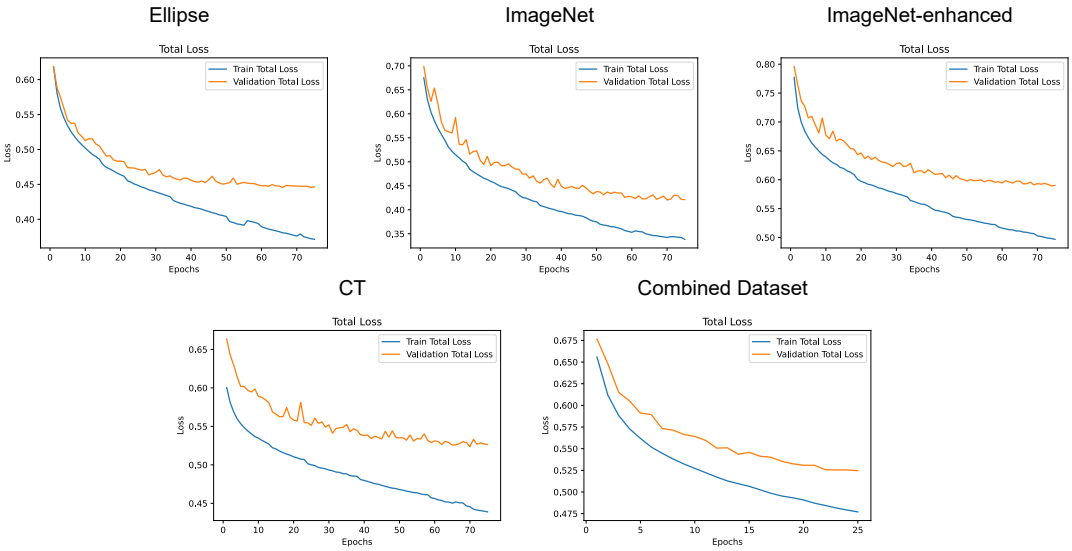


Figure 6.6.: The validation loss plotted for each epoch during the training for each of our five models.

6.3.1. Model Convergence and Validation Loss

We train the models on the Ellipse, ImageNet, ImageNet-enhanced, and CT phantom datasets all for 75 epochs, while the larger combined dataset was trained for 25 epochs (refer to Section 5.4.2 for details). Here we can see for of the models, at which epoch the best validation loss was achieved in Table 6.7:

Trained Model	**Epochs Trained**	**Epoch best validation loss**
Ellipse model	75	66
ImageNet model	75	70
ImageNet-enhanced model	75	74
CT model	75	75
Combined model	25	25

Table 6.7.: For each model we present how many epochs we train and at which epoch we achieve the best validation loss.

The results in Table 6.7 show that the Ellipse and ImageNet models have converged while the others might benefit from training for more epochs. It is also intuitive that these two models converge faster, since the Ellipse and ImageNet phantom datasets are less complex than the other datasets. These datasets contain less complexity and therefore less information that the model can learn. The ImageNet-enhanced, the CT and in particular the Combined model, on the other hand, are trained on more complex data and could benefit from longer training. All the validation losses together with the training

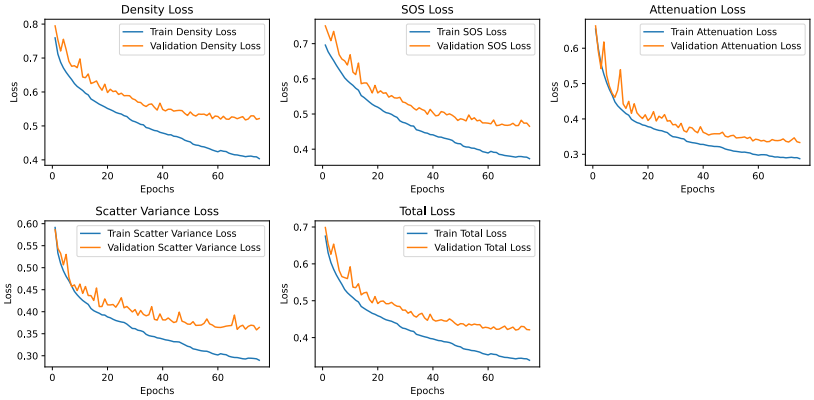


Figure 6.7.: The validation loss for each phantom map (density, SoS, attenuation coefficient and scattering parameter) individually.

losses are plotted in Figure 6.6. It's important to note that we cannot directly compare the values of the validation losses shown in Figure 6.6, since they depend on the dataset specific normalization parameters.

We also evaluated the validation loss curves for predicting each phantom map (density, SoS, attenuation coefficient and scattering parameter) individually. These losses are shown for the ImageNet model in Figure 6.7, but the plots look similar for the other models as well. When comparing the normalized losses in Figure 6.7, we find that the scattering and attenuation achieve a lower normalized loss than the density and SoS. This means that the trained model can more accurately predict the attenuation and scattering values, with respect to the respective standard deviation, than the density and SoS. We can explain this discrepancy between the different phantom maps using the results from the experiment in Section 6.2, where we see the effect of changing the phantom parameters with respect to the standard deviation on the image. We conclude that the density and SoS maps are harder to predict, which is reflected in the validation losses reported in Figure 6.7.

6.3.2. Results for Synthetic Evaluation Datasets

Here we present the performance of each of the trained models on their corresponding synthetic dataset. The synthetic dataset contains images created from the corresponding phantom type that the model was trained on. The Combined model is evaluated on the test sets of each of the previous four datasets. The values reported in Table 6.8 are the average metrics across the entire evaluation set.

Please be cautious that we cannot directly compare the performance metrics of these 5 models on the synthetic test set since each of these metrics is computed for different evaluation sets. Each of these datasets has different distributions, since the strategy

Metric	Ellipse	ImageNet	ImageNet-enhanced	CT	Combined
RF Metrics					
Timesignal MAE	0.824	0.876	0.879	0.851	0.852
Envelope RMSE	1.045	1.062	1.082	1.053	1.073
CC average peak	0.187	0.179	0.178	0.187	0.183
RMS Difference	0.192	0.248	0.271	0.222	0.240
PSD MAE	1.351e-10	1.362e-10	1.347e-10	1.360e-10	1.359e-10
Image Metrics					
Image RMSE	42.822	41.993	42.368	39.458	44.285
Mean Intensity Diff	0.282	0.145	0.151	0.123	0.228
SNR Difference	0.156	0.122	0.120	0.095	0.143
Average KL-Div	0.512	1.267	1.346	1.064	1.066
SSIM	0.128	0.128	0.133	0.134	0.129
Perceptual Loss	0.110	0.104	0.103	0.085	0.106
Phantom Metrics					
RMSE Density	47.414	58.663	49.373	109.428	60.589
RMSE SoS	44.771	62.248	49.521	189.959	80.434
RMSE Attenuation	0.276	0.332	0.292	0.538	0.331
RMSE Scattering	0.0019	0.0029	0.0022	0.002985	0.002255

Table 6.8.: Results for each model on their respective test set containing synthetic data samples of the same virtual phantom type as the model was trained on. We show the RF, Image and Phantom metrics for each of the Ellipse, Imagenet, Imagenet-enhanced, CT and Combined models and can compare the different metrics.

to create the phantoms is different, and some phantom structures could be harder to approximate than others. However, the results in Table 6.8 can still give us some valuable insights, that are elaborated on in the following.

First, we can compare the RF metrics achieved for all models in Table 6.8 with the results in Table 6.3 from the data generation validation experiment in Section 6.1. We find that the RF metrics for all models on the synthetic datasets outperform the "baseline" of the CIRS experiment. This can be easily explained by the gap between real and simulated RF data, which is present for the CIRS experiment but not for the metrics on the synthetic data. For the synthetic datasets, we compare two simulated RF signals, while for the experiment in Section 6.1 real and simulated RF data are compared. The discrepancy in the results across the RF metrics shows the sim-to-real gap for the RF data, that we need to be aware of.

The Ellipse model's predictions give the best results for the RF metrics and part of the Image metrics on its respective ellipse test set. This shows that it can match the respective training domain the best, which can be explained due to the simpler phantoms

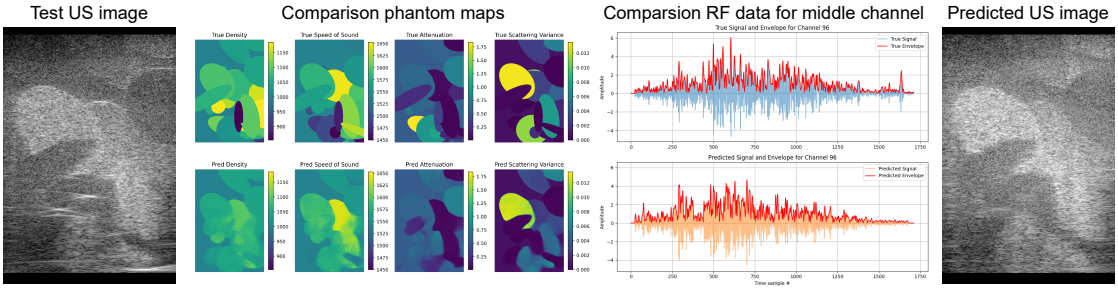


Figure 6.8.: Example of visual results for the ellipse model on the ellipse test set.

it needs to predict. However, it is interesting to see that the intuitive perceptual loss and SSIM metrics indicate the best results for the CT model, compared to the Ellipse model.

We can see that the Combined model shows worse metrics compared to the other four models. This can be explained, because the Combined model needs to recreate all four data distributions, while the other models can each fit their corresponding dataset and the particular distribution of virtual phantoms.

When comparing the phantom metrics across the datasets, we have to be aware that each model was evaluated on different datasets with different distributions. We therefore cannot conclude that the Ellipse model performs better than the CT model just because the RMSE is larger. In the CT phantom dataset, we have a larger variance for the density, SoS and across the different virtual phantoms. This makes it in turn harder to predict these phantom parameters with the same accuracy, which leads to the higher error for the CT phantom model.

We can also qualitatively evaluate the visualization of predictions of each model on their respective test set and find promising results.

In Figure 6.8 we have the visual results for an image the Ellipse model processes. It is evident that the predicted phantom matches the true phantom quite closely for all 4 acoustic properties and the resulting US image looks visually very close to the original image. This shows the effectiveness of our approach and the capability to recreate these phantoms.
However, we find that the predicted RF data generally has lower peaks than the true RF data, which can be seen in Figure 6.8. We find this for most samples in the ellipse dataset, however, this is also an issue we find for the other four models we train. High peaks in the RF data indicate echoes with high amplitudes that are received by the transducer, which are caused by large jumps in acoustic impedance. We can see that the predicted virtual phantoms have smoother and less abrupt borders between targets for the SoS and density maps, compared to the original phantoms, which leads to smaller peaks in the resulting RF data. One way to potentially achieve clearer borders in the predicted

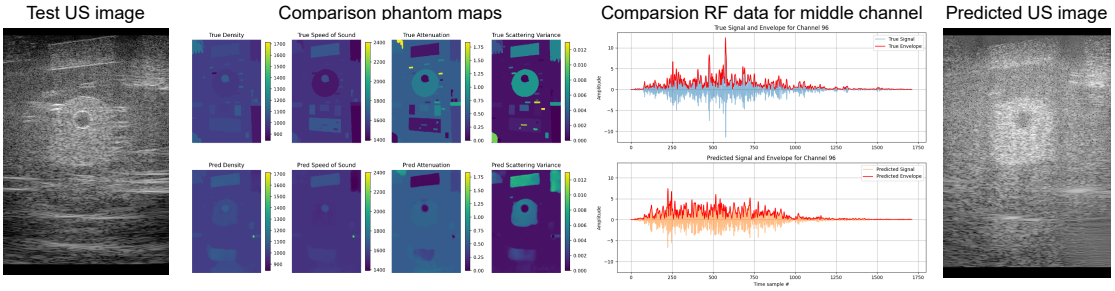


Figure 6.9.: Example of visual results for the ImageNet model on the ImageNet phantom test set.

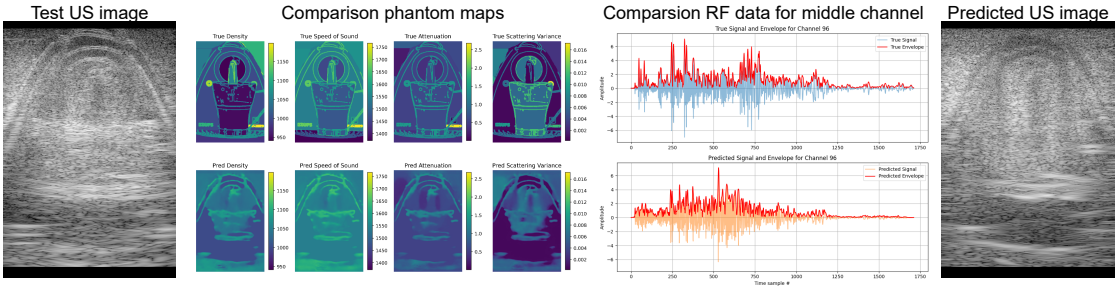


Figure 6.10.: Example of visual results for the ImageNet-enhanced model on the ImageNet-enhanced test set.

phantom could be by introducing an additional loss term when training the model, that tries to enforce the same gradients between the target and predicted phantom maps.

For the ImageNet model, we find that large targets are generally well detected, which is a good indication. An example is shown in Figure 6.9. However smaller targets, as can be seen in the phantom plots in Figure 6.9, are not detected with the same accuracy by the model. This could be partly due to the inherent resolution of the US image, which cannot capture structures that are too small. However, it can also be related to the loss function we use. By using the global MAE loss, small structures tend to not weigh in as much towards the loss function compared to the larger more homogeneous structures. To get around this problem a different and more complex loss function could be explored such as the perceptual loss that is commonly used for generative models [95].

An example of a prediction of the ImageNet-enhanced model can be seen in Figure 6.10. We can see in this example that the finer lines are recognized by the model, which shows promise. However predicting smaller structures can still be challenging for the model, as discussed for the ImageNet model.
What we find here is that the bottom half of the phantom is predicted with less accuracy than the top half. This can be explained, because at greater depths, the ultrasound transducer needs to send and receive signals over a longer distance and the echoes might

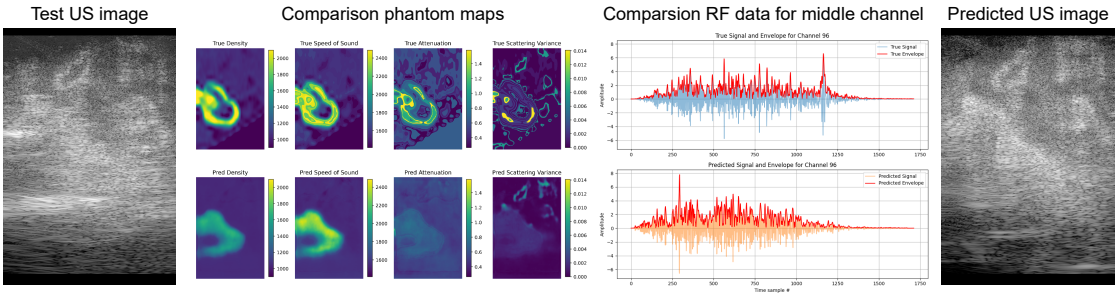


Figure 6.11.: Example of visual results for the CT model on the CT phantom test set.

get distorted by the structures above it. Therefore we have less direct information at a lower resolution for the bottom section of the US image, which makes the prediction of the bottom parts inherently harder. It is most evident for this dataset, however, the issue of getting worse predictions for the bottom half of the images can also be found for the other models we trained.

Figure 6.11 shows an example of a prediction of the CT model. Visually, the predicted US image closely resembles the original image, particularly in the upper regions. However, we find that the predicted density and SoS maps from the CT model are generally more uniform than those of previous models, likely due to the CT phantom dataset's wider variation in densities and SoS. Even after excluding data samples containing bone tissue at the transducer lens, a significantly larger proportion of phantoms still include bone structures compared to the other datasets. The other datasets select tissues at random from the IT'IS tissue list [9], where soft tissues are much more frequent than bone tissue. Therefore, the CT model is less biased towards soft tissue and learns to predict density and SoS on a larger range, which leads to less precise predictions for these two maps. We find that the predicted scattering maps include more details, as can be seen in Figure 6.11, since it follows the same distribution as in the previous datasets.

6.3.3. Results for External Evaluation Datasets

The external evaluation set (refer to Section 5.2.2) can be used to evaluate how well the models generalize beyond the domains that they were trained on, and how well they can recreate US data that was acquired using real systems.

The metrics obtained on the external test set are compared to the upper bound results established in Section 6.1, referred to as the ceiling. Since the ceiling involves a hand-crafted virtual phantom accurately representing the correct acoustic properties, the models are not expected to surpass these results.

To evaluate and compare the model performances, the external evaluation sets are split into two subgroups. On the one hand, we have phantom data that was acquired on a phantom model, which includes the PICMUS phantom set and the Verasonics CIRS

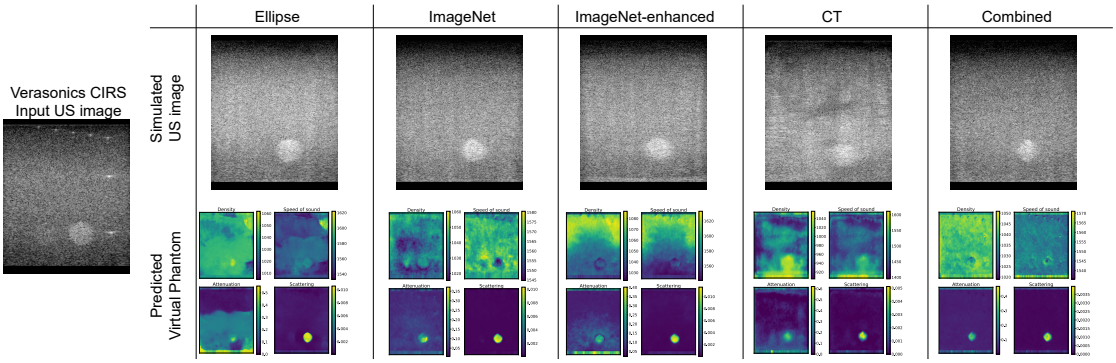


Figure 6.12.: The visual results for a sample of the Verasonics phantom evaluation set.

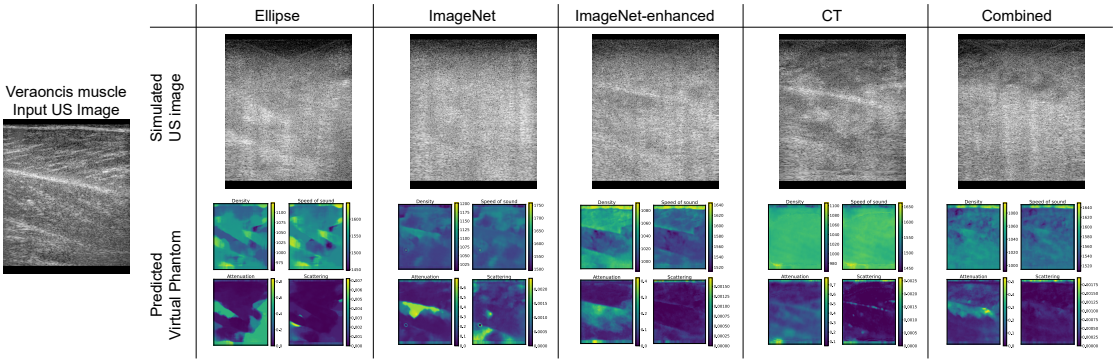


Figure 6.13.: The visual results for a sample of the Verasonics muscle evaluation set.

evaluation sets. On the other hand, we have in vivo data that was acquired on real-life tissue, which includes the CUBDL in vivo, the PICMUS in vivo and the Verasonics muscle data evaluation sets.

The quantitative results for RF and Image metrics are presented for both the phantom and in vivo datasets in Figure 6.15. We also evaluate and compare the predictions of the different models visually.

One thing to note from the quantitative results in Figure 6.15 is the high time-signal and frequency spectrum MAE for the ceiling experiment. The error being higher than that of the other predictions seems very counter-intuitive. Meanwhile, all other RF metrics we have introduced show the ceiling achieving significantly better results. This might be an indication that the MAE metrics are not the most reliable, when comparing the RF metrics, compared to the other three RF metrics we have introduced.

In Figure 6.12 the results for the Verasonics CIRS phantom input image are shown. It can be seen that all the models recognize the high contrast target in the phantom they predict. However, the Ellipse, ImageNet, and Combined models have the cleanest resulting image. This makes sense since the Ellipse and ImageNet phantom datasets are

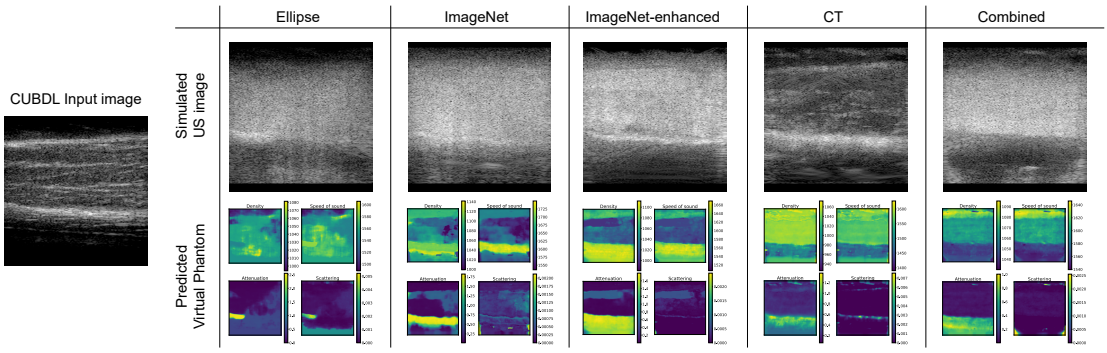


Figure 6.14.: The visual results for a sample of the CUBDL in vivo evaluation set.

trained on clear targets placed on a homogeneous background.
Meanwhile, the CT model predicts disturbing patterns in the density and SoS maps that reduce the resulting image quality. These patterns are likely caused by the presence of similar patterns in the CT phantom dataset.
None of the models successfully recreate the point targets in the resulting images, which can be linked to the lack of similar single-point targets in all datasets. However, it is noteworthy that the predicted phantoms from the Combined and both ImageNet models exhibit traces of the point targets, albeit with an acoustic impedance difference too small to cause a visible echo in the resulting image.

The visual results for the CIRS phantom are also largely in line with the Image and RF metrics for the phantom evaluation set shown in Figure 6.15, with a few exceptions. The Ellipse model has the lowest envelope RMSE and mean RMS difference and achieves a high average CC peak indicating that it is the model suited the best for predicting phantom data. This is also intuitive, since physical phantoms consist largely of simple geometric shapes. The low scores for the RF metrics for the ImageNet model might be disturbed by an outlier, since visually there is no clear indication of this.

The results are also visually inspected for the in vivo data samples. Figures 6.13 and 6.14 show the results of the models for two different types of muscle data. For reference, the finer brighter structures in the muscle tissue are called aponeuroses.
Immediately it can be seen that the predicted US image of the CT phantom is visually the most similar to the real acquired input image. This indicates that the predicted RF data holds contents most similar to the original RF data. The CT model also achieves the best score for the relevant RF and most of the Image metrics, which lets us conclude that the CT model achieves the best generalization for in vivo data samples.

It becomes clear that the Ellipse and ImageNet models do not perform well for the in vivo examples, and cannot capture the finer longer structures of the aponeuroses in the muscle tissue. In the phantom map for the ellipse model in Figure 6.13, we can see the hints of ellipses and simple geometric shapes, however, they are not enough to capture the fine structures. The predicted image by the ImageNet-enhanced model in Figure 6.13

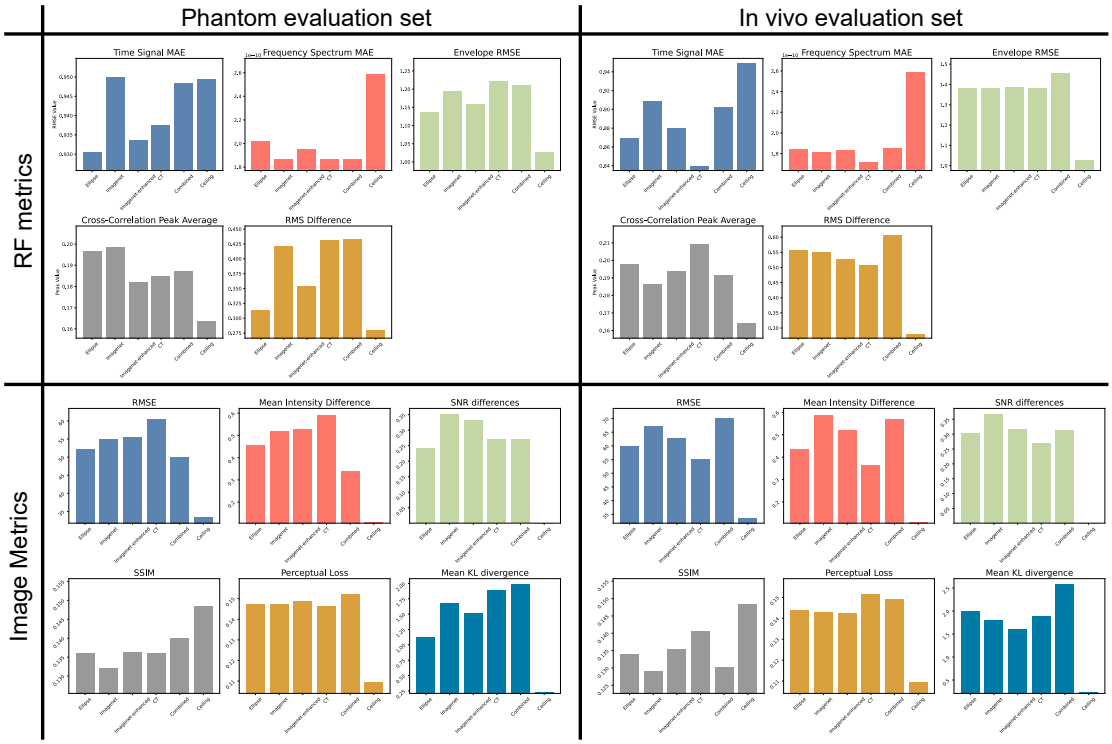


Figure 6.15.: A comparison for the Image and RF metrics between the models between the phantom and in vivo dataset. The results are also compared with the Ceiling results we find.

features slightly more visible lines for the two largest aponeuroses sheets. It makes sense for the ImageNet-enhanced model to perform better than the ImageNet model to detect the finer structures, although the effect is not as large as we would hope.

The Combined model's quantitative and visual results are also underwhelming and similar to the Ellipse and ImageNet phantom models. One reason for this is likely the fact that the CT samples make up a relatively small part of the phantom dataset, compared to the combination of the other three datasets. Therefore it might make sense to introduce more CT phantoms into the combined dataset, relative to the other phantom types. Another factor that might hinder its performance could be that the Combined model has not fully converged yet, therefore training the model longer might be beneficial.

Furthermore, a cascaded training regime could benefit the training of the Combined mode. This is further elaborated on in Section 7.2. However the cascaded learning would require carefully fine-tuning the number of epochs we train on each dataset and unfortunately, this was outside the scope of this project.

One issue that we found when evaluating our results on the external dataset, was reoccurring shading patterns that were present in the formed images for all models. These

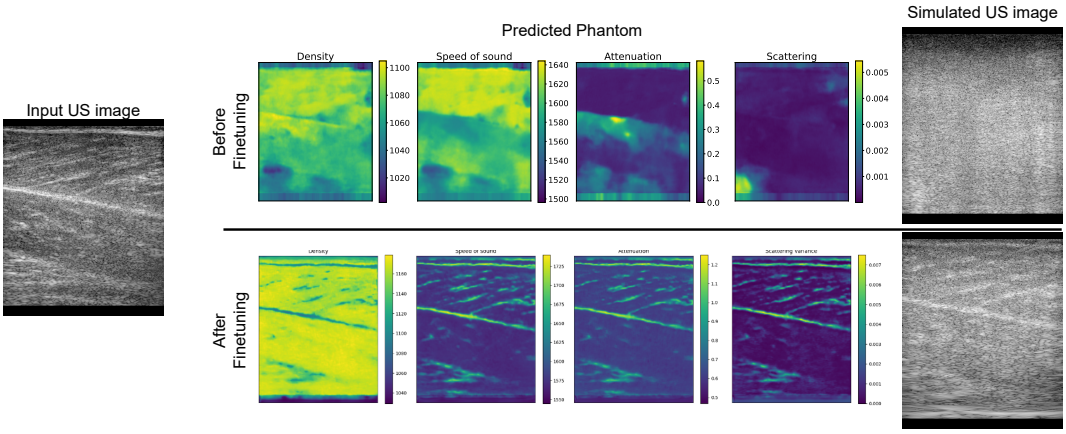


Figure 6.16.: An example visual result, comparing the prediction of an MG muscle US image before and after fine-tuning for the Combined model.

patterns could not be explained using the underlying numerical phantoms and eventually, it was found to be an error in the dasIT beamformer. This issue was caused by the binary implementation of the apodization function in dasIT [45] (see Equation 4.17). The problem can be resolved by using a windowing function such as the Henning or Blackman window instead.

6.3.4. Results for the Fine-Tuned Model

After fine-tuning each of our models on the Muscle phantom set, based on MG muscle images, very promising results can be shown. Here only the numerical and visual results for the Combined model are presented, however, the same trends can be seen for the other models as well.

In Figure 6.16 the improvement of the model's prediction is evident. This example compares the model's prediction for an MG muscle image before and after the fine-tuning step. It can be seen in the predicted phantom maps, that the fine-tuned model picks up on the smaller aponeuroses structures. These then also become visible in the corresponding simulated image, which appears much closer to the original input image.

The improvement is also reflected in the numerical results presented in Table 6.9. For the Verasonics muscle evaluation set, significant improvements for all the Image metrics become apparent. While the RF metrics results are mixed, the envelope RMSE and RMS difference clearly show an improvement.

It should be noted that, while this fine-tuned model shows good results, the model generally learns a fairly simple pattern from the muscle phantom data. Due to the binary threshold masks that are used to create the muscle training data, the model tends to divide the pixels into two distinct different tissue types. If unforeseen tissue types are

Metric	MG Muscle		In vivo		Phantom	
	Before	After	Before	After	Before	After
RF Metrics						
Timesignal MAE	0.929	**0.853**	0.902	**0.788**	0.948	**0.886**
Envelope RMSE	1.410	**1.21**	1.455	**1.334**	**1.211**	1.243
CC average peak	**0.200**	0.179	0.191	**0.211**	0.187	**0.195**
RMS Difference	0.591	**0.372**	0.606	**0.480**	**0.433**	0.458
PSD MAE	**2.11e-10**	3.00e-10	**1.86e-10**	2.12e-10	**1.86e-10**	2.24e-10
Image Metrics						
Image RMSE	57.34	**37.41**	70.16	**55.20**	**50.09**	65.22
Mean Intensity Diff	0.266	**0.111**	0.569	**0.448**	**0.340**	0.712
SNR Difference	0.116	**0.048**	**0.314**	0.319	**0.269**	0.346
Average KL-Div	1.944	**0.413**	2.575	**1.536**	1.977	**1.363**
SSIM	0.137	**0.183**	0.130	**0.168**	**0.139**	.138
Perceptual Loss	0.105	0.105	**0.149**	0.153	**0.152**	0.158

Table 6.9.: Numerical results for the Combined model, comparing the RF and image metrics before and after fine-tuning on the MG Muscle dataset. The average metrics are calculated for specific Verasonics Muscle evaluation sets containing MG muscle images; all in vivo evaluation sets; and all phantom evaluation sets. The better score for each metric is marked in bold.

present in the target US image, the model will likely struggle more to capture this, the longer the model was fine-tuned. Therefore, the number of fine-tuning epochs should be chosen carefully.

Still, it can be concluded, that if the acoustic properties and patterns of the target tissue domain are known in advance, the model performance can be significantly increased by fine-tuning on a small dedicated dataset.

6.4. End-to-End Model Results

In this section, the performance of the end-to-end model trained on the Ellipse dataset is assessed and compared to the physically-informed model trained on the same dataset. Both models are evaluated using the corresponding synthetic Ellipse test set, to avoid the domain gap of training and evaluation set present for the external evaluation datasets.

In Table 6.10 the resulting RF and image metrics are presented and in Figure 6.17 an example of a prediction is visualized for both the end-to-end and the physically-informed model.

Qualitatively a clear difference between the predicted images and RF data of the end-to-end and physically-informed model can be found, such as in Figure 6.17. In the example,

Metric	Physically-Informed Model	End-to-End Model
RF Metrics		
Timesignal MAE	0.824	**0.644**
Envelope RMSE	**1.045**	1.741
CC average peak	0.187	**0.225**
RMS Difference	**0.192**	0.736
PSD MAE	**1.351e-10**	1.462e-10
Image Metrics		
Image RMSE	**42.822**	53.47
Mean Intensity Diff	**0.282**	0.316
SNR Difference	**0.156**	0.221
Average KL-Div	**0.512**	3.562
SSIM	0.128	**0.168**
Perceptual Loss	**0.110**	0.322

Table 6.10.: Results for the physically-informed model and the end-to-end model that are both trained and evaluated on the ellipse phantom dataset.

the physically-informed model RF data prediction matches the original RF data much more closely and the US image also looks far closer to the ground truth. Meanwhile, the US image for the prediction of the end-to-end model looks very unrealistic. This can be explained, since the end-to-end model's predicted RF data is not guaranteed to agree with the laws of physics that the US acquisition follows. Due to the numerical simulation model in the physically-informed model, this is intrinsic to its RF data prediction. This explains the more realistic RF data and corresponding US image for the physically-informed model.

These qualitative findings are also reflected in the quantitative metrics in Table 6.10. The physically-informed model has a much lower error for the envelope of the RF metrics and the windowed RMS difference is also considerably lower. This clearly shows a clear improvement in the quality of the predicted RF data of the physically-informed model. This is also reflected in the resulting image metrics, where the predicted image of the physically-informed model achieves better metrics for all of our metrics, except for the SSIM.

These results let us conclude that the physically-informed model produces superior results over the end-to-end model and supports our choice and design of the physically-informed model.

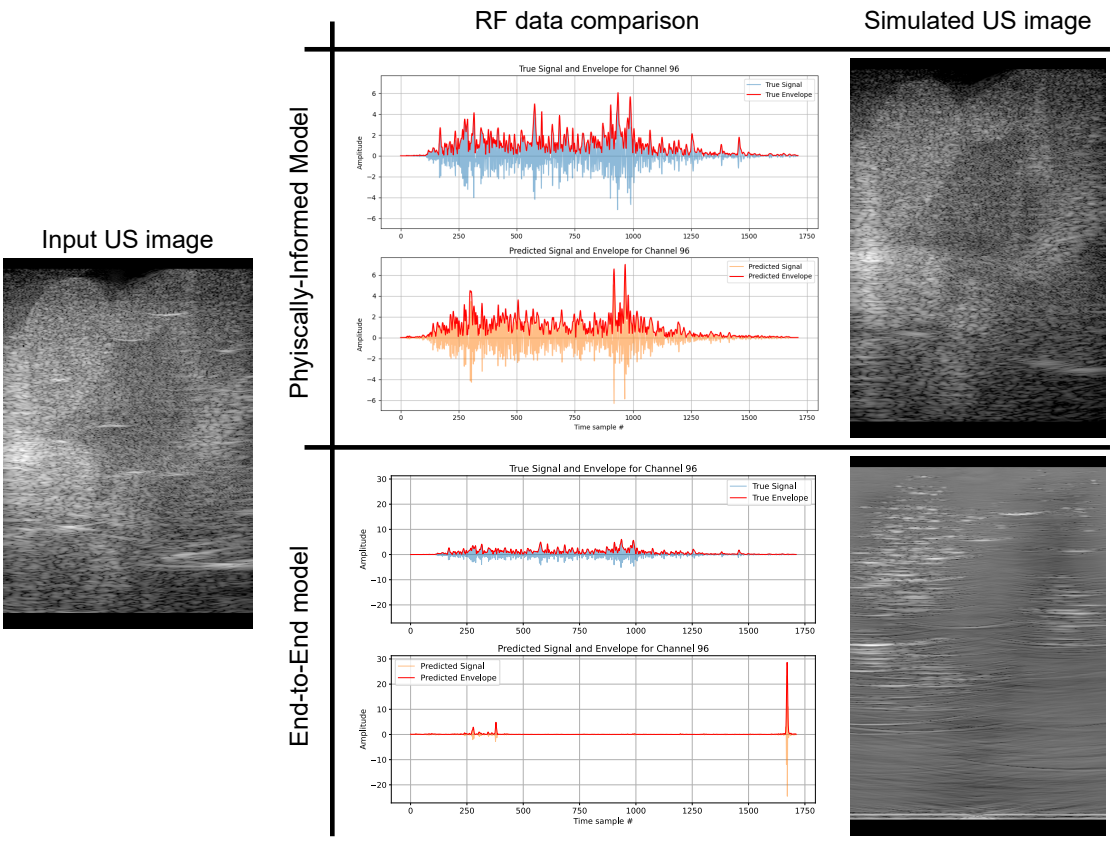


Figure 6.17.: An example comparing the predicted RF signal and corresponding US image of the physically-informed and the end-to-end model. For the RF data, the middle channel of the transducer is plotted for both results.

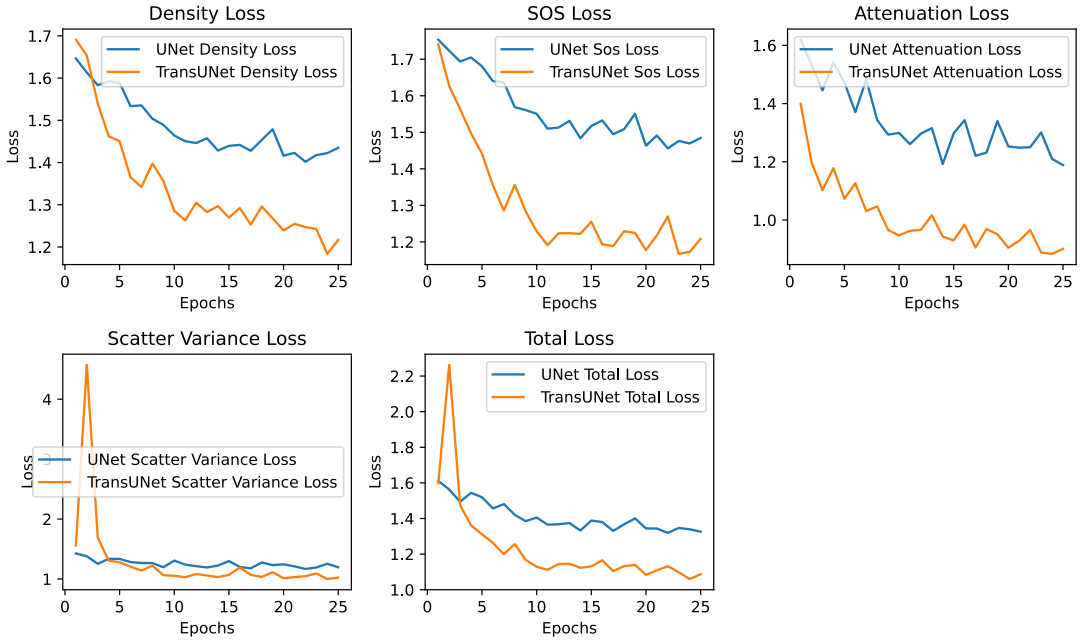


Figure 6.18.: The validation loss for each phantom map on the ImageNet phantom dataset. The TransUnet has the lower validation loss for all four phantom maps. Here the training is done without data augmentation.

6.5. Ablation Study

6.5.1. Model Architecture

We have investigated two different model architectures for predicting the acoustic parameter phantom from the US image. The UNet[19] and the TransUNet[8] model architectures are compared, where both models have the exact architectures described in Sections 4.4.2 and 4.4.2. The two models are designed to have a comparable number of learnable parameters. In this particular experiment, we have trained both models without data augmentation using the MSE loss for 25 epochs on the ImageNet phantom dataset.

Figure 6.18 shows the validation loss for both models for each phantom parameter map individually and their average in the total loss. It can be seen that the validation loss for the TransUNet model is significantly smaller for each phantom map than that of the traditional UNet architecture, for each of the predicted acoustic parameters. This clear reduction of the validation loss of the TransUNet over the UNet lets us conclude that it is more suitable for the task of predicting acoustic phantom maps from US images.

Based on these results the TransUNet model is selected as our DL model of choice.

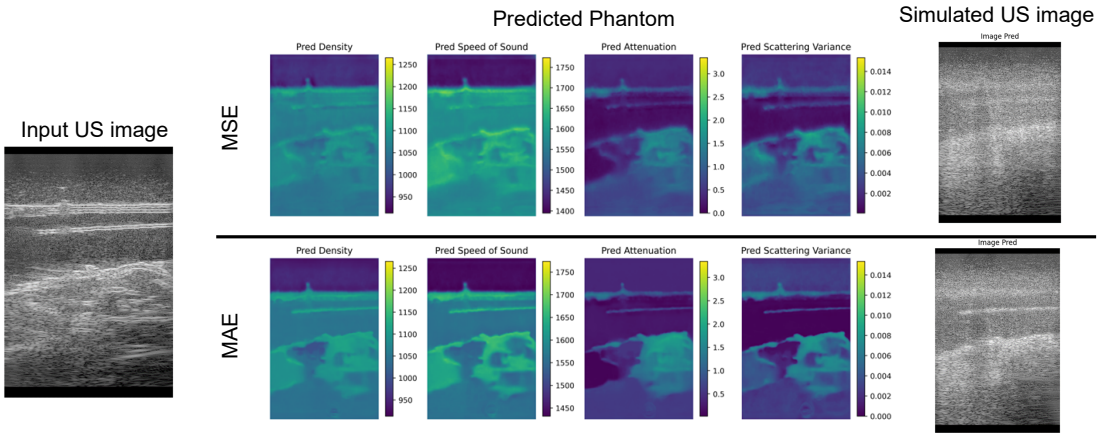


Figure 6.19.: An example that shows the effect of the MAE against the MSE loss function during training.

6.5.2. Loss Function

We investigate the choice of the loss function for training the DL model from scratch. To this end, we compare the predictions for the TransUNet model trained on the ImageNet-enhanced dataset for 50 epochs once using the MAE and once with the MSE loss. A visualization of a predicted phantom of an image in the test set is shown in Figure 6.19.

We find that the MSE tends to lead to smoother predicted phantom maps than the model trained using the MAE (see Figure 6.19). These smoother predictions could be explained by the higher penalization of outliers of the MSE compared to the MAE. The smoother phantom maps, lead to generally smoother transitions of acoustic impedances. Since the reflected echo amplitude depends on the acoustic impedance differences, the smoother transitions will lead to a smoother resulting US image with lower contrast and less clear edges. The US image from the MAE prediction will therefore have better contrast and clearer edges than the US image resulting from the MSE prediction, which can be seen in in Figure 6.19.

For this reason, the MAE is selected over the MSE for training the DL models from scratch.

6.5.3. Dataset size

An important hyperparameter to consider is the number of samples in the training dataset. The amount of samples needed to train an ML model varies considerably between different tasks. For this reason, we try to evaluate if the task of predicting the acoustic properties needs the full dataset size we are using or if fewer data points might also be sufficient. For this purpose, we compare three models trained on three different

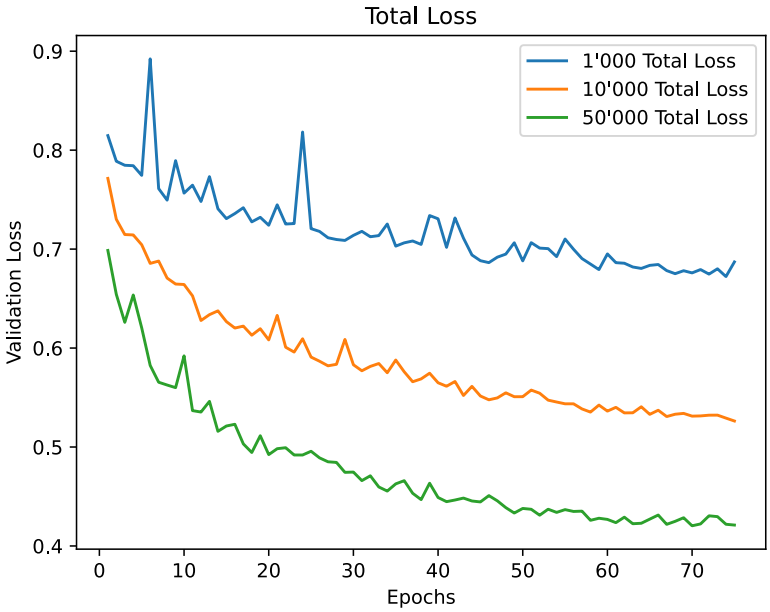


Figure 6.20.: The total validation loss comparing the model trained on 1'000, 10'000 and 50'000 samples of the ImageNet phantom dataset

subsets of the ImageNet phantom set training data. The dataset sizes are 1'000, 10'000 and 50'000 samples of which 80% are used for training. All models are validated on the full validation set of 5'000 images.

In Figure 6.20 the total validation loss is reported and compared for the three models. It becomes very clear that the model significantly improves if it is trained on larger datasets, as the validation loss converges at a considerably lower value. At some point you would expect a diminishing return from more training data, however, we see that this is not the case yet. This indicates that even larger datasets could potentially further improve the model's performance.

Chapter 7

Conclusion and Future Work

7.1. Conclusion

In summary, the following contributions have been made in this thesis:

1. We are the first to design a system to predict RF data from US images. We have designed and implemented a physically-informed model using a data-driven DL model to predict the acoustic properties of the image and a numerical simulation to predict realistic RF data from US images. For the numerical simulation, the k-Wave simulation tool was used.
2. We introduce a framework to create 5 distinct synthetic datasets of virtual phantoms for training our DL model: the Ellipse phantom dataset, the ImageNet phantom dataset ImageNet-enhanced dataset, and the CT phantom dataset in addition to a smaller Muscle phantom dataset. A total of 210'000 virtual phantoms, along with their corresponding RF data and processed US images, were generated. Our approach has the potential to create much larger datasets.
3. For the DL model, a TransUNet architecture was trained from scratch for each synthetic dataset. Furthermore, the performance of our physically-informed system for each synthetic dataset was evaluated on an external dataset of real US data. We determined both visually and numerically, that the model's predictions are generalizing best to the US data of in vivo tissue, when training on the CT phantom dataset.
4. We demonstrated that, by fine-tuning the model for a specific tissue type, using tissue specific synthetic data, the RF predictions of our physics-informed model can be improved for images of that tissue type.

5. Two different DL models were explored and we found that the TransUNet is the more suited model architecture for predicting the acoustic properties of the imaging medium, showing superior performance.
6. An end-to-end deep learning model was trained to predict the RF data directly from the US image, using the same synthetic datasets as the physics-informed model. We showed that the end-to-end model achieves significantly worse results compared to our physically-informed system, further validating the choice of the system we have designed.

7.2. Future Work

The next step in evaluating the quality of the predicted RF signals would be to train a model for a specific task on the synthetic RF data and then evaluate the trained model on real RF data. An example of such a task would be the prediction of pennation angles in lateral gastrocnemius (LG) muscles using RF data as described by Vostrikov *et al.* [14]. If the trained model generalizes well to real RF data, it further confirms the value of the synthetic RF data we predict.

One straightforward way to improve our method would be to generate and train on a larger synthetic dataset. Our method has the potential to create far larger datasets for all 4 phantom types. Only a small fraction of the ImageNet and the CT scan dataset available was used by us for the dataset generation. The ImageNet contains 14 million images [51], of which only 50'000 were utilized, while for the CT phantoms only 4 out of 15'248 subjects available in the NMDID [63] are currently used. Deep learning methods generalize better using more data and larger models with more trainable parameters can be trained successfully. This becomes a question of computational and time resources available.

Instead of using segmentation maps from ImageNet images, an interesting possibility is to explore extracting segmentation maps from medical images for generating virtual phantoms. While this approach might appear similar to the method used to create the CT phantoms, this would not be limited to CT scans and could be applied to other image modalities such as MRI images and US images, further increasing the amount of available datasets. Instead of the SAM model, the alternative and more recent MedSAM model presented by Ma *et al* [96] could be used for segmentation, as it is trained specifically for medical image segmentation.

Another way to improve the quality of the predicted RF data of our system would be to improve the US simulation, by making it more realistic. For efficiency in this work, a 2D simulation on a simulation grid with exactly the width of the transducer is chosen. For a real acquisition, the RF data is also affected by a wider region that spans three dimensions. Therefore, it might be interesting to explore if a larger phantom during

inference improves the predicted RF data quality. The modeling of the non-linearity of the acoustic medium could also improve the final results.

There is furthermore potential to improve our combined model trained on the union of the four large datasets. Instead, a cascaded training regime to combine the different synthetic datasets could be applied. This implies to start training on a simpler dataset (the ellipse dataset) and then for later training epochs on progressively more complex data (such as the CT phantom dataset). This approach has occasionally proven effective in the field of curriculum learning [97]. One work that explores training a model on datasets from multiple domains in this fashion and discusses the challenges that come with it is the study by Karthik *et al.* [98].

Since our model is trained on a synthetic dataset with the goal of generalizing well to real US data, the target domain and the training domain are inherently different. Therefore, one could explore strategies from the field of domain adaptation to generalize better from the training to the target domain. One such technique that could be explored is Meta-Learning introduced by Li *et al.* [99], a model-agnostic training procedure that encourages the model to generalize well to other domains during training.

Appendix A

Task Description

Master Thesis at the Department of Information Technology and Electrical Engineering

Autumn Semester 2023

Maurits Reitsma

Ultrasound Image Data Recycler

October 30, 2023

Advisors:	Dr. Christoph Leitner, ETZ J65, christoph.leitner@iis.ee.ethz.ch Dr. Yawei Li, ETF D114.2, yawei.li@vision.ee.ethz.ch
Supervisor:	Prof. Dr. Luca Benini, lbenini@iis.ee.ethz.ch Prof. Dr. Luc Van Gool, vangool@vision.ee.ethz.ch

Handout:	October 30, 2023
Due:	April 29, 2024

1 Project Outline

In clinics huge digital ultrasound dataset exist. Since medical ultrasound equipment used is proprietary and closed, it is not possible to access the raw data of these images. In emerging ultrasonic applications, e.g., AR/XR, robotics, human-machine-interfaces, networks are used for sensor control at the extreme edge. Training these models would benefit from raw data, but it is scarce. To overcome this dilemma, this project aims to develop an image-to-raw data converter. The project explores end2end and mixed deterministic/data-driven model and approaches all implemented in C++ and Python. The latter focuses on developing a "physically informed" model that includes an ultrasound simulator (deterministic), a CNN U-net model (data-driven), and a joint loss function to convert images to raw data.

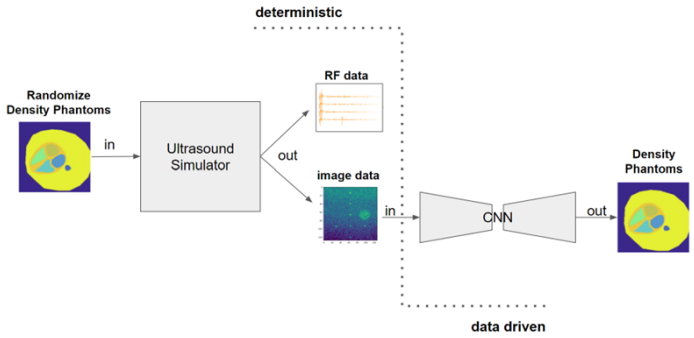


Figure 1: The proposed methodology for the USDataRacycler consists of 2 modlues: one simulation environment to create ultrasound RF data and images from density maps and a neural network architecture to convert images to density maps.

1.1 Research Question

The research questions of this thesis are:

- What are possible model pipelines for converting images to raw data. How large are the conversion losses and what are strategies to improve conversion?
- Quantification of how well your model generalises between different domains at all levels of the development.

1.2 Methodology

- **Workpackage I - Literature and Code review:** The aim of this task is to develop a general understanding of the physics of ultrasound, ultrasound image

formation, ultrasound beamforming and simulation. In addition, the state of the art in ML-based beamforming techniques will be assessed. Furthermore, neural network architectures for image restoration and denoising will be evaluated. During the literature review, a database of open code repositories and datasets will be maintained, e.g. as a Markdown file in the project's Gitlab repository.

Deliverables:

- **D1:** Fine Grained Project Plan and Gantt Chart.
- **D2:** Documentation of the literature review findings.
- **D3:** Maintained database of open code repositories and datasets.

- **Workpackage II - Mockup:** The main task of this work package is to setup a naive but complete processing pipline. This testbench should include the simulation environment and the neural network architecture as well as interfaces between each other.

  Deliverables:

  - **D4:** Setup and documentation of the preliminary processing pipeline.

- **Workpackage III - Iteration A, Synthetic Data:** The aim of this task is to explore density phantom creation methods. An important aspect in these investigations must be explainability. Why does one method work better than the other? And how can I quantify accuracy and losses? It will be important to also establish a statistical monitoring framework for quantification.

  Deliverables:

  - **D5:** Synthetic density maps with quantified accuracy measures.
  - **D6:** Report the methods' explainability and accuracy.

- **Workpackage IV - Iteration B, Interface Efficiency:** After we have defined and evaluated our model using synthetic data, we want to scale the training and testing. Therefore, the modules (e.g. physical simulation and neural networks) and the interfaces need to be as efficient as possible. How does scaling affect our results? and if why?

  Deliverables:

  - **D7:** Report on the optimization of simulation and neural network modules.
  - **D8:** Assessment of scalability impact on results.

- **Workpackage V - Iteration C, Transfer to Real World Data:** In this step we want to apply our model to real world data. Are the results comparable to those obtained with synthetic data? If not, how can I adapt my methodology to increase accuracy. How robust is our model architecture to data coming from source (e.g. instruments)?

  Deliverables:

  - **D9:** Results of applying the model to real-world data.
  - **D10:** Assessment of the model's robustness to various data sources.

- **Workpackage VI - Report and Presentation:** Work on the final report and thesis presentation.

  Deliverables:

  - **D11:** Final report summarizing the entire project and clean and documented code repository.
  - **D12:** Prepared materials and presentation for the thesis run-through and defense.

2 Project Realization

2.1 Project Plan

Within the first week of the project you will be asked to prepare a project plan. This plan should identify the tasks to be performed during the project and sets deadlines for those tasks. The prepared plan will be a topic of discussion of the first week's meeting between you and your advisers. Note that the project plan should be updated constantly depending on the project's status.

2.2 Meetings

Weekly meetings and reports must be held. The exact time and location of these meetings will be determined within the first week of the project in order to fit the student's and the assistant's schedule. These meetings will be used to evaluate the status and progress of the project. Besides these regular meetings, additional meetings can be organized to address urgent issues as well.

2.3 Report

Documentation is an important and often overlooked aspect of engineering. One final report has to be completed within this project. The common language of engineering is de facto English. Therefore, the final report of the work is preferred to be written in English. Any form of word processing software is allowed for writing the reports, nevertheless, the use of LaTeX with Tgif[1] or any other vector drawing software (for block diagrams) is strongly encouraged by the IIS staff.

Code Repository As many of our projects are heavily code-based, the documentation of the code and its repository is also considered in the grading of the final report. We suggest to follow coding standards and style guides when writing code, e.g. C [1], Python [2],...

Final Report The final report has to be presented at the end of the project and a digital copy need to be handed in. Note that this task description is part of your report and has to be attached to your final report.

2.4 Presentation

There will be a presentation (15 min for the semester thesis, and 20 min for the MS thesis presentation followed by 5 min Q&A) at the end of this project in order to present your results to a wider audience. The exact date will be determined towards the end of the work.

References

[1] "Recommended C Style and Coding Standards." [Online]. Available: https://www.doc.ic.ac.uk/lab/cplus/cstyle.html

[2] "PEP 8 – Style Guide for Python Code | peps.python.org." [Online]. Available: https://peps.python.org/pep-0008/

[1] See: `http://bourbon.usc.edu:8001/tgif/index.html` and `http://www.dz.ee.ethz.ch/en/information/how-to/drawing-schematics.html`.

Maurits Reitsma	Date, Place
Prof. Dr. Luca Benini	Date, Place

Appendix B

Declaration of Originality

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

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

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies[2].

◉ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies[3]. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Ultrasound Image Recycling: Predicting Raw Data with Simulation-Driven Deep Learning

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

Last name(s):	**First name(s):**
Reitsma	Maurits

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

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

Place, date	**Signature(s)**
28.04.2024	

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

[1] E.g. ChatGPT, DALL E 2, Google Bard
[2] E.g. ChatGPT, DALL E 2, Google Bard
[3] E.g. ChatGPT, DALL E 2, Google Bard

Bibliography

[1] A. T. Peter Hoskins, Kevin Martin, *Diagnostic Ultrasound: Physics and Equipment*, 2nd ed. Cambridge University Press, 2010.

[2] L. Oláh, *Ultrasound principles*. Cambridge University Press, 2016, p. 1–14.

[3] K.-V. Jenderka and S. Delorme, *Diagnostischer Ultraschall*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 285–305. [Online]. Available: https://doi.org/10.1007/978-3-662-54801-1_10

[4] "Ultrasound artifacts," https://radiologykey.com/ultrasound-artifacts/, 2024, accessed: 27.04.2024.

[5] B. Treeby, B. Cox, and J. Jaros, "Defining an ultrasound transducer example," http://www.k-wave.org/documentation/example_us_defining_transducer.php, 2024, accessed: 18.04.2024.

[6] C. L. Palmer and O. M. H. Rindal, "Wireless, real-time plane-wave coherent compounding on an iphone: A feasibility study," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 66, no. 7, pp. 1222–1231, 2019.

[7] C. Leitner, *New directions in recording and processing electro-mechanical signals from the human body*, 2022.

[8] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," 2021.

[9] IT'IS Foundation, "Tissue properties database v4.1," 2 2022.

[10] CIRS, *General Purpose Ultrasound Phantom Model 054GS: User guide*, Computerized Imaging Reference Systems, Inc., 2013.

[11] S. H. Contreras Ortiz, T. Chiu, and M. D. Fox, "Ultrasound image enhancement: A review," *Biomedical Signal Processing and Control*, vol. 7, no. 5, pp. 419–428, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809412000183

[12] T. J. Klein, "Statistical image processing of medical ultrasound radio frequency data," Ph.D. dissertation, Technische Universität München, 2012. [Online]. Available: https://mediatum.ub.tum.de/1100919

[13] S. Frey, V. Kartsch, C. Leitner, A. Cossettini, S. Vostrikov, S. Benatti, and L. Benini, "A wearable ultra-low-power semg-triggered ultrasound system for long-term muscle activity monitoring," in *2023 IEEE International Ultrasonics Symposium (IUS)*, 2023, pp. 1–4.

[14] S. Vostrikov, T. M. Ingolfsson, S. Hafthorsdottir, C. Leitner, M. Magno, L. Benini, and A. Cossettini, "A muscle pennation angle estimation framework from raw ultrasound data for wearable biomedical instrumentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.

[15] S. Vostrikov, M. Anderegg, C. Leitner, L. Benini, and A. Cossettini, "Hand gesture recognition via wearable ultra-low power ultrasound and gradient-boosted tree classifiers," in *2023 IEEE International Ultrasonics Symposium (IUS)*, 2023, pp. 1–4.

[16] C. Leitner, R. Jarolim, B. Englmair, A. Kruse, K. A. L. Hernandez, A. Konrad, E. Y.-S. Su, J. Schröttner, L. A. Kelly, G. A. Lichtwark, M. Tilp, and C. Baumgartner, "A human-centered machine-learning approach for muscle-tendon junction tracking in ultrasound images," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 6, pp. 1920–1930, 2022.

[17] Y. Chen, C. Zhang, L. Liu, C. Feng, C. Dong, Y. Luo, and X. Wan, "Uscl: Pretraining deep ultrasound image diagnosis model through video contrastive representation learning," 2021.

[18] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of Biomedical Optics*, vol. 15, no. 2, p. 021314, 2010. [Online]. Available: https://doi.org/10.1117/1.3360308

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[20] R. Millner, *Wissensspeicher Ultraschalltechnik*. Leipzig: VEB Fachbuchverlag Leipzig, 1987.

[21] P. Allan, G. Baxter, and M. Weston, *Clinical Ultrasound*, ser. Clinical Ultrasound. Churchill Livingstone, 2011, no. v. 1. [Online]. Available: https://books.google.ch/books?id=6xTlzAEACAAJ

[22] A. A. Vives, *Piezoelectric Transducers and Applications*, 1st ed. Berlin, Heidelberg: Springer, 2004.

[23] A. Maier, S. Steidl, V. Christlein, and J. Hornegger, *Medical Imaging Systems: An Introductory Guide*, ser. Lecture Notes in Computer Science. Cham CH: Springer Open, 2018, vol. 11111.

[24] H. Hasegawa and C. de Korte, *Ultrafast Ultrasound Imaging*. MDPI AG, 2018. [Online]. Available: https://books.google.ch/books?id=O0hvDwAAQBAJ

[25] D. Perdios, M. Vonlanthen, F. Martinez, M. Arditi, and J.-P. Thiran, "Cnn-based image reconstruction method for ultrafast ultrasound imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 69, no. 4, pp. 1154–1168, 2022.

[26] J. Jensen, "Field: A program for simulating ultrasound systems," vol. 4, 03 1999.

[27] L. Zhang, V. Vishnevskiy, and O. Goksel, "Deep network for scatterer distribution estimation for ultrasound image simulation," 2020.

[28] M. Verweij, B. Treeby, K. van Dongen, and L. Demi, "2.19 - simulation of ultrasound fields," in *Comprehensive Biomedical Physics*, A. Brahme, Ed. Oxford: Elsevier, 2014, pp. 465–500. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444536327002215

[29] C. Shi and X. G. Xu, "Development of a 30-week-pregnant female tomographic model from computed tomography (ct) images for monte carlo organ dose calculations," *Medical Physics*, vol. 31, no. 9, pp. 2491–2497. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.1778836

[30] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui, "Realistic ct simulation using the 4d xcat phantom," *Medical Physics*, vol. 35, no. 8, pp. 3800–3808. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2955743

[31] L. Maack, L. Holstein, and A. Schlaefer, "Gans for generation of synthetic ultrasound images from small datasets," *Current Directions in Biomedical Engineering*, vol. 8, no. 1, pp. 17–20, 2022. [Online]. Available: https://doi.org/10.1515/cdbme-2022-0005

[32] D. Stojanovski, U. Hermida, P. Lamata, A. Beqiri, and A. Gomez, "Echo from noise: synthetic ultrasound image generation using diffusion models for real image segmentation," 2023.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.

[34] W. A. Simson, "Physics-informed deep learning for advanced medical ultrasound," Ph.D. dissertation, Technische Universität München, 2022. [Online]. Available: https://mediatum.ub.tum.de/1634543

[35] R. Kuc, "Clinical application of an ultrasound attenuation coefficient estimation technique for liver pathology characterization," *IEEE Transactions on Biomedical Engineering*, vol. BME-27, no. 6, pp. 312–319, 1980.

[36] G. Ghoshal and M. L. Oelze, "Time domain attenuation estimation method from ultrasonic backscattered signals," *The Journal of the Acoustical Society of America*, vol. 132, no. 1, pp. 533–543, 07 2012. [Online]. Available: https://doi.org/10.1121/1.4728195

[37] T. W. Chean, N. Hozumi, S. Yoshida, K. Kobayashi, and Y. Ogura, "Mutual conversion between b-mode image and acoustic impedance image," *Japanese Journal of Applied Physics*, vol. 56, no. 7S1, p. 07JF18, jun 2017. [Online]. Available: https://dx.doi.org/10.7567/JJAP.56.07JF18

[38] S. J. Sanabria, U. Hilbers, J. Neuenschwander, P. Niemz, U. Sennhauser, H. Thömen, and J. L. Wenker, "Modeling and prediction of density distribution and microstructure in particleboards from acoustic properties by correlation of non-contact high-resolution pulsed air-coupled ultrasound and x-ray images," *Ultrasonics*, vol. 53, no. 1, pp. 157–170, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0041624X12001011

[39] G. F. Pinton, G. E. Trahey, and J. J. Dahl, "Sources of image degradation in fundamental and harmonic ultrasound imaging using nonlinear, full-wave simulations," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 58, no. 4, pp. 754–765, 2011.

[40] B. Treeby, B. Cox, and J. Jaros, *k-Wave: A MATLAB toolbox for the time-domain simulation of acoustic wave fields*, August 2016, toolbox Release 1.1. [Online]. Available: http://www.k-wave.org

[41] ——, "Running c++ simulations example," http://www.k-wave.org/documentation/example_cpp_running_simulations.php, 2024, accessed: 18.04.2024.

[42] D. Hyun, A. Wiacek, S. Goudarzi, S. Rothlübbers, A. Asif, K. Eickel, Y. C. Eldar, J. Huang, M. Mischi, H. Rivaz, D. Sinden, R. van Sloun, H. Strohm, and M. A. L. Bell, "Deep learning for ultrasound image formation: Cubdl evaluation framework & open datasets," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. x, no. x, p. xx, 2021.

[43] H. Liebgott, A. Rodriguez-Molares, J. Jensen, and O. Bernard, "Plane-wave imaging challenge in medical ultrasound," in *IEEE International Ultrasonics Symposium*, Tours, France, 2016, p. accepted.

[44] V. Perrot, M. Polichetti, F. Varray, and D. Garcia, "So you think you can das? a viewpoint on delay-and-sum beamforming," *Ultrasonics*, vol. 111, p. 106309, Mar. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.ultras.2020.106309

[45] C. Leitner, "dasit," https://github.com/luuleitner/dasIT, 2024, accessed: 2024-04-19.

[46] *Vantage User Manual*, Vantage software release 4.1.1-1910101200 ed., Verasonics, Oct. 2019, confidential documentation for users of the Verasonics Vantage ultrasound system. Provides comprehensive guidelines on system setup, safety precautions, software installation, system tests, and maintenance procedures.

[47] D. Benitez, P. Gaydecki, A. Zaidi, and A. Fitzpatrick, "The use of the hilbert transform in ecg signal analysis," *Computers in Biology and Medicine*, vol. 31, no. 5, pp. 399–406, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482501000099

[48] T. D. Mast, "Empirical relationships between acoustic parameters in human soft tissues," *Acoustics Research Letters Online*, vol. 1, no. 2, pp. 37–42, 10 2000. [Online]. Available: https://doi.org/10.1121/1.1336896

[49] X. Li, P. Yi, E. Curry, and A. Murakami, "Ultrasonography as a diagnostic, therapeutic, and research tool in orthopaedic surgery," *Journal of the American Academy of Orthopaedic Surgeons*, vol. 26, pp. 187–196, 03 2018.

[50] J. Burkardt, "The truncated normal distribution," *Department of Scientific Computing Website, Florida State University*, vol. 1, no. 35, p. 58, 2014.

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[53] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., ser. Texts in Computer Science. Springer Cham, 2022.

[54] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[55] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma,

A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2022.

[56] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[57] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[58] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[59] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2014. [Online]. Available: https://www.sciencedirect.com/book/9780123948113/introduction-to-probability-and-statistics-for-engineers-and-scientists

[60] S. Hussain, I. Mubeen, N. Ullah, S. S. U. D. Shah, B. A. Khan, M. Zahoor, R. Ullah, F. A. Khan, and M. A. Sultan, "Modern diagnostic imaging technique applications and risk factors in the medical field: A review," *BioMed Research International*, vol. 2022, pp. 1–19, 2022.

[61] J. Hsieh, *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*, 3rd ed. SPIE, 2015.

[62] E. Samei and N. J. Pelc, Eds., *Computed Tomography: Approaches, Applications, and Operations*, 1st ed. Springer Cham, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-26957-9

[63] H. Edgar, S. Daneshvari Berry, E. Moes, N. Adolphi, P. Bridges, and K. Nolte, "New mexico decedent image database," 2020. [Online]. Available: https://doi.org/10.25827/5s8c-n515

[64] V. Chougule, A. Mulay, and B. Ahuja, "Clinical case study: Spine modeling for minimum invasive spine surgeries (miss) using rapid prototyping," 02 2018.

[65] P. Soille, *Erosion and Dilation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–103. [Online]. Available: https://doi.org/10.1007/978-3-662-05088-0_3

[66] A. Sudhyadhom, "On the molecular relationship between hounsfield unit (hu), mass density, and electron density in computed tomography (ct)," *PLoS ONE*, vol. 15, no. 12, p. e0244861, 2020.

[67] U. Schneider, E. Pedroni, and A. Lomax, "The calibration of ct hounsfield units for radiotherapy treatment planning," *Physics in Medicine & Biology*, vol. 41, no. 1, p. 111, jan 1996. [Online]. Available: https://dx.doi.org/10.1088/0031-9155/41/1/009

[68] B. Treeby, B. Cox, and J. Jaros, "hounsfield2density," http://www.k-wave.org/documentation/hounsfield2density.php, 2024, accessed on 18.04.2024.

[69] J. Wang, "An intuitive tutorial to gaussian process regression," *Computing in Science & Engineering*, vol. 25, no. 4, p. 4–11, Jul. 2023. [Online]. Available: http://dx.doi.org/10.1109/MCSE.2023.3342149

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[71] C. Leitner, S. Vostrikov, H. Penasso, P. A. Hager, A. Cossettini, L. Benini, and C. Baumgartner, "Detection of motor endplates in deep and pennate skeletal muscles in-vivo using ultrafast ultrasound," in *2020 IEEE International Ultrasonics Symposium (IUS)*, 2020, pp. 1–7.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[73] G. Li, D. Jin, Q. Yu, Y. Zheng, and M. Qi, "Multiib-transunet: Transformer with multiple information bottleneck blocks for ct and ultrasound image segmentation," *Medical Physics*, vol. 51, no. 2, pp. 1178–1189, 2024. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.16662

[74] H. van Hasselt, A. Guez, M. Hessel, and D. Silver, "Learning functions across many orders of magnitudes," *CoRR*, vol. abs/1602.07714, 2016. [Online]. Available: http://arxiv.org/abs/1602.07714

[75] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. Madison, WI, USA: Omnipress, 2011, p. 521–528.

[76] V. A. Chenarlogh, M. G. Oghli, A. Shabanzadeh, N. Sirjani, A. Akhavan, I. Shiri, H. Arabi, M. S. Taheri, and M. K. Tarzamni, "Fast and accurate u-net model for fetal ultrasound image segmentation," *Ultrasonic Imaging*, vol. 44, no. 1, pp. 25–38, 2022, pMID: 34986724. [Online]. Available: https://doi.org/10.1177/01617346211069882

[77] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[79] E. O. Brigham and R. E. Morrow, "The fast fourier transform," *IEEE Spectrum*, vol. 4, no. 12, pp. 63–70, 1967.

[80] M. A. L. Bell, J. Huang, D. Hyun, Y. C. Eldar, R. van Sloun, and M. Mischi, "Challenge on ultrasound beamforming with deep learning (cubdl)," in *Proceedings of the 2020 IEEE International Ultrasonics Symposium*, 2020, pp. 1–5.

[81] X. Zhang, J. Li, Q. He, H. Zhang, and J. Luo, "High-quality reconstruction of plane-wave imaging using generative adversarial network," in *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2018, pp. 1–4.

[82] M. A. L. Bell, J. Huang, A. Wiacek, P. Gong, S. Chen, A. Ramalli, P. Tortoli, B. Luijten, M. Mischi, O. M. H. Rindal, V. Perrot, H. Liebgott, X. Zhang, J. Luo, E. Oluyemi, and E. Ambinder, "Challenge on Ultrasound Beamforming with Deep Learning (CUBDL) Datasets." [Online]. Available: http://dx.doi.org/10.21227/f0hn-8f92

[83] T. Derrick, *Time Series Analysis: The Cross-Correlation Function*, 01 2004, pp. 189–205.

[84] K. Cartwright, "Determining the effective or rms voltage or various waveforms without calculus," vol. 8, 01 2007.

[85] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[86] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[87] P.-H. Tsui, C.-K. Yeh, C.-C. Chang, and Y.-Y. Liao, "Classification of breast masses by ultrasonic nakagami imaging: a feasibility study," *Physics in Medicine & Biology*, vol. 53, no. 21, p. 6027, oct 2008. [Online]. Available: https://dx.doi.org/10.1088/0031-9155/53/21/009

[88] P. Shankar, J. Reid, H. Ortega, C. Piccoli, and B. Goldberg, "Use of non-rayleigh statistics for the identification of tumors in ultrasonic b-scans of the breast," *IEEE Transactions on Medical Imaging*, vol. 12, no. 4, pp. 687–692, 1993.

[89] J. Shlens, "Notes on kullback-leibler divergence and likelihood," *CoRR*, vol. abs/1404.2000, 2014. [Online]. Available: http://arxiv.org/abs/1404.2000

[90] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[91] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[92] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[93] CIRS, *Multi-Purpose, Multi-Tissue Ultrasound Phantom Model 040GSE*, Computerized Imaging Reference Systems, Inc., 2013.

[94] J. J. Dahl, S. A. Mcaleavey, G. F. Pinton, M. S. Soo, and G. E. Trahey, "Adaptive imaging on a diagnostic ultrasound scanner at quasi real-time rates," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 10, pp. 1832–1843, 2006.

[95] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[96] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, pp. 1–9, 2024.

[97] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," 2018.

[98] E. N. Karthik, A. Kerbrat, P. Labauge, T. Granberg, J. Talbott, D. S. Reich, M. Filippi, R. Bakshi, V. Callot, S. Chandar, and J. Cohen-Adad, "Segmentation of multiple sclerosis lesions across hospitals: Learn continually or train from scratch?" 2022.

[99] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," 2017.

[100] T. L. Szabo, "Chapter 1 - introduction," in *Diagnostic Ultrasound Imaging: Inside Out (Second Edition)*, second edition ed., T. L. Szabo, Ed. Boston: Academic Press, 2014, pp. 1–37. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012396487800001X

[101] T. Klein, M. Hansson, and N. Navab, "Spatial statistics based feature descriptor for rf ultrasound data," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 33–36.

[102] H.-A. Loeliger and C. Reller, "Signal processing with factor graphs: Beamforming and hilbert transform," in *2013 Information Theory and Applications Workshop (ITA)*, 2013, pp. 1–4.

[103] J. Zhu, S. Zhao, Y. Ye, and G. Wang, "Computed tomography simulation with superquadrics," *Medical Physics*, vol. 32, no. 10, pp. 3136–3143. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2040727

[104] F. Tom and D. Sheet, "Simulating patho-realistic ultrasound images using deep generative networks with adversarial learning," 2018.

[105] M. Jakovljevic, S.-W. Hsieh, R. Ali, G. Chau Loo Kung, D. Hyun, and J. J. Dahl, "Local speed of sound estimation in tissue using pulse-echo ultrasound: Model-based approach," *J Acoust Soc Am*, vol. 144, no. 1, p. 254, Jul 2018.