Learned Acoustic Reconstruction using Synthetic Aperture Focusing

by

Tim Straubinger

B.Sc., The University of British Columbia, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

October 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Learned Acoustic Reconstruction using Synthetic Aperture Focusing

submitted by **Tim Straubinger** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Robert Xiao, Assistant Professor, Department of Computer Science, UBC
Supervisor

Helge Rhodin, Assistant Professor, Department of Computer Science, UBC
Co-supervisor

Antony Hodgson, Professor, Department of Mechanical Engineering, UBC
Supervisory Committee Member

Abstract

Navigating and sensing the world through echolocation in air is an innate ability in many animals for which analogous human technologies remain rudimentary. Many engineered approaches to acoustic reconstruction have been devised which typically require unwieldy equipment and a lengthy measurement process, and are largely not applicable in air or in everyday human environments. Recent learning-based approaches to single-emission in-air acoustic reconstruction use simplified hardware and an experimentally-acquired dataset of echoes and the geometry that produced them to train models to predict novel geometry from similar but previously-unheard echoes. However, these learned approaches use spatially-dense representations and attempt to predict an entire scene all at once. Doing so requires a tremendous abundance of training examples in order to learn a model that generalizes, which leaves these techniques vulnerable to over-fitting.

We introduce an implicit representation for learned in-air acoustic reconstruction inspired by synthetic aperture focusing techniques. Our method trains a neural network to relate the coherency of multiple spatially-separated echo signals, after accounting for the expected time-of-flight along a straight-line path, to the presence or absence of an acoustically reflective object at any sampling location. Additionally, we use signed distance fields to represent geometric predictions which provide a better-behaved training signal and allow for efficient 3D rendering. Using acoustic wave simulation, we show that our method yields better generalization and behaves more intuitively than competing methods while requiring only a small fraction of the amount of training data.

Lay Summary

Many acoustic imaging systems in medicine and marine exploration produce ultrasound and process the returning echoes to infer what objects are located nearby, but these require complex hardware. Recently, attempts have been made to produce similar visualizations in air using minimal hardware and machine learning instead of conventional algorithms to produce 3D images. However, existing learning-based approaches try to learn entire scenes at once, which requires a large amount of training data. Instead, we use concepts from conventional acoustic imaging and focus only on individual points in space. We train a neural network to estimate the distance to the nearest obstacle at a single point using only audio signals aligned by the expected round-trip time of a wave to and from that point. We show that this implicit formulation leads to better interpolation and extrapolation and requires less training data than models which predict entire scenes at once.

Preface

The written contents of this thesis are the original work of Tim Straubinger under the guidance of Helge Rhodin and Robert Xiao. All software used in this work and all figures were created by Tim Straubinger except where otherwise noted. All experiments were performed by Tim Straubinger with the help of generous hardware resources provided by UBC ARC Sockeye.

Table of Contents

List of Tables

List of Figures

Glossary

ASRT apex shifted radon transform

CAD computer-aided design

FDTD finite-difference time-domain

FM frequency modulation

IOU intersection over union

MAE mean absolute error

PMD photonic mixer device

PML perfectly-matched layer

ROI region of interest

SAFT synthetic aperture focusing technique

SAU synthetic aperture ultrasound

SLAM simultaneous localization and mapping

SPAD single photon avalanche diode

SDF signed distance field

SNR signal-to-noise ratio

UCM ultrasound condenser microphone

UES ultrasonic electrostatic speaker

Acknowledgments

Firstly, I must thank both of my supervisors, Robert Xiao and Helge Rhodin, for providing me their mentorship, their inspiration, their patience, and their invaluable and endless questions over these past two years. Having two supervisors whose combined time and expertise I have been given has been a luxury that not many students have, and I offer my humble gratitude for this opportunity.

I am grateful for the many close friends that I have made during my time as a graduate student at UBC, who have helped learn me that being able to laugh at one's self is a valuable skill, and whose collaborations in art, music, and tea, and whose absurd conversations I will cherish: Dave, Paulette, Noah, Joe, Greg, the Tuesday Tea gang, and the honourable speakers of the UnDistinguished Lecture Series.

I want to thank Dinesh Pai for providing me with a path to academia, without which I would never have seen the richness of the Computer Science Department, its diverse research communities, and the many future scholars I will continue my journey with.

I thank the UBC ARC Sockeye high-performance computational cluster, for offering their enormous resources to budding academics such as myself, without which this work would not have been feasible.

To my parents, I am grateful for your support in my continued education and the welcoming environment you have always provided me.

Finally, to my partner Dinah, with whom my life's greatest adventures are perpetually only beginning, thank you for always being there beside me.

Chapter 1

Introduction

The physical world around us can convey enormous amounts of information simply through the acoustic echoes it produces in response to emitted sounds. This is made clear by the fact that many animals, such as bats, cetaceans, and even some humans [23, 49], are able to localize themselves and navigate by producing sound and listening to the echoes returning from their surroundings. Thus, there is potential to be able to infer spatial information from acoustic echoes using technology for localization and geometric reconstruction.

Compared to optical vision, sound presents a comparable but distinct modality for sensing the environment. Sound is reflected and absorbed differently by materials compared to light, and thus can yield novel information about the physical properties of nearby objects. Notably, sound interacts primarily with changes in density of the media it passes through, unlike light, which has far more complex interactions [3]. Sound also travels slowly enough for its time of flight to be easily measured, which allows for accurate distance information to be obtained with simple sensing techniques. Furthermore, while both light and sound may be absorbed, reflected, and refracted by objects due to wave phenomena, the wavelengths of sound are much larger than those of light, and diffraction phenomena become relevant at macroscopic scales. According to the Rayleigh criterion,

$$\theta \approx 1.22\frac{\lambda}{D}, \tag{1.1}$$

the angular resolution θ of a wave-based sensor is limited by the wavelength λ being used as well as the width of the aperture D. For example, an acoustic device with a 30 centimeter aperture working at 40 kHz can theoretically achieve an angular resolution of 0.035 radians, or about 3.5 centimeters at a 1 meter distance. While this is indeed a limitation, it can also be a benefit in human environments because it means that acoustic sensing devices can offer privacy by construction while still yielding useful spatial information - for example, the presence but not the identity of a person nearby. Additionally, the privacy guarantees due to the wavelengths used by an active ultrasonic sensing device can be verified by an end user using an ultrasonic microphone.

In medical contexts, ultrasonic imaging techniques are widely used to non-invasively study developing fetuses and internal organs. The speed of sound in human tissue is about 4.7 times faster than in air [43], which permits faster repeated measurements of the target sample, which in turn enables scanning-based approaches. Typical medical ultrasounds form images one line at a time by focusing waves and iteratively scanning the subject [12]. The spatial and temporal resolution that can be obtained in this way is limited by the speed of sound and the spatial extent of the region of interested being scanned. Given the slower speed of sound in air, and given that typical indoor environments are many orders of magnitude larger than the regions being examined by medical ultrasounds, the same scanning techniques simply would be far slower in air, and would be unable to achieve interactive frame rates in human environments.

Rather than carefully focusing sound waves and scanning in multiple directions sequentially, it also feasible to perform acoustic imaging with unfocused spherical waves by combining signals from multiple spatially-separated receivers and/or transmitters, using a family of techniques known as synthetic aperture methods. Instead of focusing waves through a single large aperture, a synthetic aperture uses multiple smaller sensors to gather waves at points scattered across a similar spatial extent. By recombining information from each point, the same angular resolution can be achieved as if one had used a single aperture of the same total size.

In medicine, synthetic aperture ultrasound (SAU) [7, 20, 32, 48] gathers returning echoes from all directions simultaneously, rather than one line at a time, and

uses software processing to disentangle individual reflections. To create an acoustic image, SAU sums the signal contributions of all echoes after accounting for the expected time delay between every combination of emitter, receiver, and point in the image. The result is that reflections due to true inhomogeneities become pronounced, while false reflections and noise are made unlikely due to destructive interference. The expected signal-to-noise ratio (SNR) thus improves steadily with the number of receiving and transmitting elements employed by the hardware, with typical medical devices using 64, 128, or 256 elements in total [20, 40, 48]. Conversely, one would expect the imaging quality of SAU without additional considerations to seriously deteriorate when using relatively few elements.

SAU is able to achieve higher quality images than scanning approaches using the same hardware, at the cost of additional data post-processing. Additionally, by avoiding the scanning process, synthetic aperture techniques can achieve higher frame rates, and they offer an important path toward in-air acoustic imaging at interactive speeds. Conventional SAU relies on multiple sequential acoustic emissions, as the redundancy offers better estimates in the presence of noise [4, 39]. In medical imaging, where the speed of sound is high and regions of interest are small, these improvements are worth the negligible increase in latency. However, in air and at larger scales, far more time is spent waiting for echoes to return, and as a result, relying on repeated measurements causes a significant drop in imaging speed.

In underwater exploration, where the speed of sound is similar to that of human tissue but the sizes of the regions of interest are on the order of many meters, the additional time cost of multiple emitted sounds due to the longer wave travel time means that they simply cannot be used for interactive applications, and scanning-based and iterative approaches are not viable [6]. Instead, underwater imaging systems typically use a single, unfocused acoustic emission whose echoes are recorded by a large planar array of omnidirectional receivers, and synthetic aperture focusing allows the same ensemble of acoustic recordings to be re-used for each viewing direction. As with SAU, underwater acoustic imaging temporally aligns all received signals with their expected time of arrival at a given point in the image to estimate the reflectivity at that point. However, in order to make de-

tailed three-dimensional images with conventional algorithms, a large abundance of acoustic receivers is needed, which increases the cost of both the hardware and the computation needed for visualization. Modern systems often use several hundred receivers, which increases resolution and reduces noise due to random wave interference, but which also introduces further design challenges, such as optimizing the spatial arrangement of the receivers and devising scalable algorithms which can efficiently handle such a volume of data [6, 53]. Given the large number of receivers, and that underwater acoustic imaging systems operate at frequencies in the hundreds of kilohertz, one would expect that the same algorithms would fail to produce meaningful visualizations at common in-air ultrasound frequencies such as 20 to 40 kHz, and with a minimal number of receivers. Instead, one would need algorithms which are better able to remove speckle artefacts and robustly reason about the presence of reflectors with far less information.

Imaging using a single acoustic emission hypothetically achieves the fastest possible frame rates but has not found much use in air, likely due to severe losses of accuracy that traditional methods encounter, since these methods generally use redundancy to overcome ambiguity, be it with an abundance of receivers, emitters, sensing iterations, or combinations thereof. With conventional algorithms, in-air acoustic imaging has to the author's knowledge only been demonstrated with iterative techniques requiring a lengthy capture process and static scenes [3, 25].

Recently, learning-based approaches have begun to demonstrate acoustic imaging using single emissions, and machine learning may offer a viable alternative to preexisting wave-based imaging techniques [8, 9, 16, 21]. Where algorithmic methods for acoustic imaging suffer due to their reliance on repeated measurements to handle challenging and non-obvious wave interactions, machine learning side-steps the need to explicitly model complex physical phenomena, and is able to specialize in common scenarios. If designed correctly, a learning-based approach can efficiently extract pertinent information from the acoustic echoes it receives and predict the most likely geometry according to a trained model. But the use of machine learning introduces unique challenges as well, such as the need for representative training data and the danger of over-fitting. Before learning-based approaches to acoustic imaging can be practically adopted, their behaviours and

limitations across diverse scenarios and their shortcomings must be well understood.

Existing machine learning methods for acoustic reconstruction use a convolutional neural network to learn to predict entire scenes at once in a dense representation, such as a three-dimensional occupancy map or two-dimensional depthmap, from a single multi-channel acoustic recording [8, 9, 16, 21]. While neural networks are able to generalize well when given a sufficient amount of training examples, the use of dense representations means that the models must learn to disentangle the myriad interactions between all points in the input audio and the output geometry. When collecting a dataset, the practical infeasibility of capturing all possible examples of geometric configurations means that models must either be capable of meaningfully interpolating and extrapolating from the data they have been shown or else risk over-fitting, producing meaningless results on previously-unseen examples after focusing too heavily on limited training data.

In order to map from the time domain of an acoustic echo to the spatial domain needed for estimating geometry, some manner of spatiotemporal conversion must be performed during acoustic reconstruction. While convolutional neural networks are translation-invariant, existing neural architectures for acoustic reconstruction follow a modular design consisting of a temporal acoustic encoder and spatial geometric decoder, with information passing between the two using only the hidden feature dimensions [8, 9, 16, 21]. As a consequence, the spatiotemporal conversion is performed only by the network's hidden activations where it is difficult to reason about.

Due to the difficulties in learned acoustic reconstruction of gathering a sufficiently large dataset for learning combined with the needs of existing convolutional neural networks for an enormous variety of training examples, we suggest that a different approach is needed. In particular, the use of a dense representation—training a neural network to map from an entire audio recording to an entire geometric scene—is not necessary. Implicit function neural networks have shown that geometric representations can instead be learned one point at a time, by providing spatial coordinates as inputs and making scalar predictions [30, 33, 41]. It would be possible to train an implicit neural network for acoustic reconstruction by providing both the recorded echoes and the coordinates of a desired spatial location as

inputs. However, by explicitly providing spatial information to the network, it may easily learn to give different treatment to different regions of space and thereby over-fit. Alternatively, one may provide the network only with sounds that have been pre-processed with respect to a given sampling location. Synthetic aperture focusing as used in non-learned acoustic imaging methods is such a process, and provides an input signal that is clearly rich in information. Importantly, aligning incoming echoes in time according to the travel time of a wave to and from an individual point in space performs a simple spatiotemporal conversion, allowing a network trained in this representation to specialize purely on the largely local relationship of such time-aligned audio signals with individual geometric details.

In this work, we explore the viability of end-to-end learned in-air acoustic reconstruction using close to the minimum viable amount of sensing hardware and experimental measurements, namely a single acoustic emitter, a small number of acoustic receivers, and a single acoustic emission whose echo is recorded simultaneously across all receivers. We achieve this using an implicit representation based on synthetic aperture focusing that directly relates the distance to the nearest reflector at a given point with the echo heard by all signals after aligning them by the expected delay due to the in-air time-of-flight. We use an acoustic wave simulation to generate a large dataset of randomly-placed obstacles and the echoes they produce, and we use this dataset to train a neural network to relate time-shifted echo signals to the geometry in the region of interest. Although we do not expect our trained neural networks and dataset to be immediately transferable to a physical realization, our technique is readily applicable in the physical world, provided that training data is available. In our explorations, the use of computer simulation enables rapid prototyping and allows for greater experimental control and fidelity, and is commonly used in the literature [3, 7, 19, 26, 32, 45–48, 57]. We evaluate our methods and compare them to competing learning-based approaches using a variety of experiments. We explore the robustness of our technique and provide commentary on a hypothetical physical implementation of our system.

Chapter 2

Related Work

Acoustic and wave-based imaging and reconstruction has a rich history spanning many fields. In medicine, ultrasonic imaging is widely used to non-invasively study internal organs at interactive speeds. In underwater acoustic imaging, similar techniques are used at a larger scale to aid in underwater operations and exploration. In robotics, acoustic sensing has been used to measure distances, establish landmarks, and to classify parts of the environment surrounding a robotic agent to aid in mapping and navigation. In seismology, wave-based sensing has been successfully applied to measure the composition and geometry of solid and subterranean structures. Between these fields, many useful developments to wave-based sensing have been made in signal processing techniques, geometric ray-based and wave-based modeling, synthetic aperture methods, migration methods, signal design, and more. Only recently have researchers turned to machine learning using end-to-end differentiable neural networks to solve acoustic reconstruction in a data-driven, example-based manner. For complex reconstructions, the amount of data needed to learn a generalized model becomes prohibitive in practice, motivating the need for carefully-chosen data representations and the application of domain-specific knowledge. For greater experimental control, some researchers turn to numeric simulation of physical wave interactions, and this demand has led to the development of sophisticated software for simulating wave propagation. In the following sections, we provide a summary of the relevant methodologies and findings across these areas.

2.1 Delay-and-Sum Methods

A simple technique for wave-based imaging that has found use in numerous domains is the delay-and-sum method, in which an image is formed point-by-point from an array of wave recordings. The reflectivity of each point in the medium is estimated as follows. First, assuming a constant speed of sound, the time of flight a wave would take to travel to and from the imaging location and each emitter and receiver pair is computed. This time delay is then used to synthetically focus the recorded waveforms, temporally aligning them with the expected wavefront due to a reflector at the present location. The focused waveforms are then summed and the resulting signal strength is the estimated reflectivity of that point as a fraction of the emitted signal's strength [3, 20, 48].

Conventional medical ultrasonic imaging uses a large array of acoustic elements that act as both emitters and receivers to capture each line of an image sequentially, in what is commonly referred to as a B-mode or B-scan image. Either a physical lens or beamforming is used to create a focused and directional pulsed ray of sound that passes into the target medium and whose returning echoes are recorded. The timing of each arriving echo is used to infer the total distance travelled and thereby the depth at which the wave was reflected. By repeating this line-based sensing many times across a range of directions, a two-dimensional image is obtained [38].

Burckhardt *et al.* first applied synthetic aperture techniques to acoustic sensing for medical applications, embracing spherical wave propagation and data post-processing to generate an acoustic image [5]. Their system uses optics rather than a digital computer to synthesize images, and is able to distinguish small reflectors at close range.

Corl *et al.* as well as Bennett *et al.* later refined these ideas and applied computational methods for producing images, albeit for nondestructive testing of solid materials, using an array of 32 elements [1, 10]. Their systems record the echoes received by an array of receivers after a controlled sound is emitted. An image is then formed point-by-point by delaying each received signal according to the expected round-trip time for wave hitting a deflector at the imaging point, and then summing the contributions of all signals.

Lockwood *et al.* explore the use of few and sparse acoustic elements to allow for more rapid imaging using similar synthetic focusing [27]. The faster two-dimensional imaging times allow the technique to be repeated while mechanically scanning the sensing device to create three-dimensional images.

Nikolov *et al.* measure the effect of coding schemes, which use recognizable patterns of acoustic elements in parallel, on the quality of images, finding that the increased complexity yields diminishing returns on imaging quality [32]. However, the use of an FM chirp as the emitted signal was found to yield a better SNR over comparably simple signals, regardless of the number of acoustic elements in use.

Later with the development of more sophisticated computational hardware, Jensen *et al.* demonstrate synthetic aperture ultrasonic imaging using an array of 64 elements operating at 5 MHz, and show that the use of synthetic focusing by means of simple weighted summations leads to faster and higher-quality images than B-scan images with beam-forming under an otherwise equivalent setting. Synthetic focusing in Jensen *et al.*'s work may be understood in terms of the closely-related synthetic aperture focusing technique (SAFT) [24] and is achieved in parallel for each point in the imaging plane by means of an apodization function which attenuates all received signals except those near the expected time-of-flight of an echo due to a deflector at the current image location. The same SAU techniques were further refined by modeling the angular distribution of wave energy from each acoustic emitter, in contrast to the implicit spherical wave assumption made by Jensen *et al*, and achieve further improvements in image quality [48].

In underwater settings, delay-and-sum methods closely related to those in SAU are commonly used in active acoustic sensing to create 3D visualizations of the environment from a 2D sensing aperture. Notably, the demand for interactive visualizations that can facilitate underwater exploration means that high frame rates are prioritized, and modern underwater acoustic imaging systems rely only a single acoustic emission and use a large array containing hundreds of acoustic receivers [6, 31]. A typical modern system uses the delay-and-sum technique, temporally aligning the signals from all receivers according to the time of flight to a single point in the space being imaged, in order to estimate the reflectivity of that point [31]. The abundance of data that must be processed to create a single visualization

in this way - at frequencies in the hundreds of kilohertz, with hundreds of receivers and for many thousands of imaging points - means that special attention is paid to efficient algorithms which can yield results at interactive rates [6]. Additionally, the large number of sensors needed for practical visualizations can be placed into diverse arrangements which greatly affects the resolution due to diffraction artefacts [6, 31]. Finally, whereas 2D visualizations can be displayed directly, the 3D reflectivity data produced by underwater acoustic imaging systems must be further processed to be displayed to a user, for example by estimating the boundaries of objects and rendering these as 3D surfaces [31].

2.2 Migration Methods

In seismology, migration methods have been used in imaging underground structures to correct for differences in reflection arrival time due to inconsistent speeds of sound in order to create visualizations that are better spatially-aligned to the ground truth [1]. In some applications, migration bears close resemblance to synthetic aperture focusing techniques as used in acoustic imaging, though alternative techniques exist which are capable of correcting spatial distortions due to media with varying densities, or which use a frequency domain representation rather than the purely temporal approach of SAU.

Loewenthal *et al.* demonstrated the use of migration on two-dimensional wave-based scans of seismic media to relocate visualized features to have a direct spatial correspondence with the physical structures being measured [28]. However, their work relied on a simplified wave model that only considers waves traveling to or from the receiver and disregards angle-dependent reflections. Kosloff *et al.* show how to overcome this and faithfully model the acoustic wave equation in migration techniques. Stolt *et al.* adapted wave migration from the spatial domain to the frequency-wavenumber (*f-k*) domain, showing improved coherence and fewer artefacts while providing a theoretical starting point for efficient migration techniques in three dimensions [44].

Trad *et al.* combine the use of the apex shifted radon transform (ASRT), which, like other methods, computes the sum of returned signals after accounting for

[1] or *ground truth*, as it were

round-trip delay, with least-squares optimization to provide better support for multiple spatially-distributed deflectors [50].

Ibrahim *et al.* devise a technique derived from both the ASRT and Stolt's *f-k* migration that separates seismic wave deflectors using numeric minimization [17]. Similarly, Garcia *et al.* apply Stolt's *f-k* migration to medical ultrasound imaging, using a planar rather than a spherical emitted pulse, achieving improved results over simpler delay-and-sum techniques used in other ultrasonic imaging [13].

2.3 Optical Reconstruction

With sound waves, the time of flight can easily be measured which enables many classes of algorithms for wave-based imaging. While the speed of traveling light waves is much harder to resolve, recent advances in optical hardware have enabled measuring the arrival times of individual light reflections, and this allows the use of 3D reconstruction algorithms similar to those used in acoustic and seismic imaging, as well as closely related synthetic aperture techniques such as back-projection.

Velten *et al.* demonstrated the use of a streak camera that records the spatiotemporal impulse response of a hidden scene after being struck by a laser pulse, and use a back-projection algorithm to reconstruct an approximation of the occluded geometry in three dimensions [55]. Notably, their system can only resolve one spatial direction during image capture because it must use one imaging axis effectively as the temporal dimension. Detailed reconstructions require iterated measurements with laser pulses at different locations, with around 60 repetitions reportedly in use in their experiments. This work was extended by Heide *et al.* to use a photonic mixer device (PMD) as the time-of-flight camera, using an optimization framework on top of a physically-based imaging model, which resulted in the ability to disentangle geometry from materials at the cost of increased computational burden. With the introduction of single photon avalanche diode (SPAD) cameras, Buttafava *et al.* were able to adapt the computationally cheaper back-projection techniques of Velten *et al.*, with a total of 185 imaging iterations at differing laser pulse locations.

Turning to numeric simulation and deep learning, Su *et al.* skip some of the signal pre-processing of time-of-flight cameras and instead train a neural network that is able to produce higher-quality geometric reconstructions than those of con-

ventional algorithms alone. Training data is created using a light transport simulation that explicitly models the optical time of flight. Though it is unclear whether the choice of training examples lead to a truly generalized model compared to a classical approach, the resulting imaging system is able to produce improved reconstructions on real-world data in typical indoor environments.

Drawing upon migration methods previously used in seismology, Lindell *et al.* adapt Stolt's *f-k* migration method to optical non-line of sight reconstruction, rather than using the backprojection technique of Velten *et al.*, in similar set of experiments with pulsed lasers and a time-resolved SPAD camera [26], leading to improved reconstructions but with up to 256 iterations with different laser pulse locations. A numeric simulation of time-resolve light transport is also used to validate the technique with increased experimental control.

Relying entirely on simulation to solve optical non-line-of-sight reconstruction, Iseringhausen *et al.* propose a highly optimized time-resolved light transport simulation which is used in the inner loop of an optimization algorithm that seeks a simple geometric reconstruction whose simulated spatiotemporal light impulse response best matches the observed data [19]. The underlying geometric representation consists of the level set of a sum of Gaussian functions which acts as a strong spatial regularizer, leading to much smoother and simpler reconstructions than the noisy volumetric image data given by back-projection.

2.4 Acoustic Reconstruction

Using between 4 and 16 acoustic elements, Wykes *et al.* compute the time delay of arrival of ultrasonic echoes of small objects in response to emitted signals and compute the positions of the reflectors using triangulation [58], in a technique reminiscent of that used in synthetic aperture ultrasound. Using bat echolocation as inspiration, Matsuo *et al.* use only a single emitted FM chirp and binaural audio to classify and localize multiple reflecting objects, relying on the time delay of arrival for localization and the spectral characteristics of each reflector to disentangle the multiple overlapping echoes. Also using the bat as an example, Steckel *et al.*, in their BatSLAM work, construct a biomimetic sonar device that is similarly able to classify and localize echoes by their time delay and spectral qualities respectively,

in order to perform acoustic simultaneous localization and mapping (SLAM) in an indoor environment [42]. Eliakim *et al* developed the RoBat system, additionally classify obstacles in the environment by their echoes as one of two categories using a neural network.

Borrowing techniques from optical imaging, Lindell *et al.* use an array of 16 receivers and 16 microphones scanning across 32 unique positions, emitting FM chirps to capture echoes of hidden objects, followed by computational reconstruction using the Light Cone Transform with several additional processing steps, resulting in crude geometric reproductions [25].

2.4.1 Learned Acoustic Reconstruction

Rather than using the highly sophisticated iterative, geometric, or optimization-based algorithms described previously, some researchers are turning to neural networks and deep learning as a black box alternative for acoustic reconstruction, which allows for greater computational efficiency and improved specificity for real-world scenarios.

In BatVision, Christensen *et al.* use a small robot equipped with a speaker, binaural microphones, and depth camera to gather a large dataset of indoor scenes and the echoes they produce in response to an FM chirp [9]. This dataset is then used to train fully-convolutional neural networks to predict greyscale and depth images solely from single echoes. In this setting, the use of spectrograms over waveform audio led to better reconstructions. In further experiments, an additional neural network was used as a discriminator during training to encourage predicted images to more closely match the distribution of the training data. Overall, depth images were found to be a more meaningful representation over greyscale images, owing to the colouring and lighting information that is not expected to be conveyed in acoustic echoes.

Hwang *et al.* similarly use a convolutional neural network modeled after the neural pathways of a bat's brain, referred to as Bat-G Net, to predict geometry from acoustic echoes [16]. To gather their dataset, dubbed ECHO-4CH, a stationary sensing device consisting of a central ultrasonic electrostatic speaker (UES) and 4 ultrasound condenser microphones (UCMS) that record the echoes produced in

response to an FM sweep by simple geometric objects. Each example in the dataset contains one or two obstacles in various orientations at a range of roughly 1.5 meters. Labels for training consist of binary voxel grids that are generated in CAD. The Bat-G Net model then is trained to predict voxel geometry from an entire echo, through a combination of fully-connected and convolutional neural network layers. Occlusions are explicitly disallowed by back-filling any obstructed voxels, resulting in what is effectively a height-map representation.

Later, Kim *et al.* further develop the Bat-G Net system by incorporating attention mechanisms, resulting in Bat-G2 Net [21]. The ECHO-4CH dataset from previous work is re-used, and rather than predicting back-filled three-dimensional data, two-dimensional heightmaps are generated instead. Experiments were performed measuring the performance degradation when competing echo signals appear as noise in the input audio, and the addition of a non-local attention mechanism was found to increase reconstruction quality even under the presence of severe noise relative to the earlier Bat-G Net.

2.5 Acoustic Simulation

To study the interactions between sound and obstacles in a virtual environment, a simulation is needed that can accurately model the propagation of acoustic waves through space and their interactions. This can be achieved simply by discretizing and integrating the wave equation

$$\nabla^2 p - \frac{1}{c_0^2}\frac{\delta^2 p}{\delta t^2} = 0 \tag{2.1}$$

which relates the acceleration of the acoustic pressure at any point in space to its local curvature and the speed of sound. This can be done using the finite-difference time-domain (FDTD) method, which represents the acoustic pressure as a uniform grid and computes the local curvature using the differences between neighbouring grid points. Though conceptually simple, the FDTD requires a very fine discretization to achieve a reasonable accuracy for high-bandwidth simulations, which effectively incurs a very high memory overhead [51]. An alternative method to finite differences is to use Fourier methods for derivative computation, which is what

the *k*-space pseudospectral method does. Treeby *et al.* provide an efficient implementation of the *k*-space pseudospectral method in the K-Wave acoustic simulation toolkit [51, 52]. K-Wave integrates the acoustic wave equation while allowing users to specify initial conditions and configurations such as spatially-varying acoustic properties and sound sources. The K-Wave toolkit has found use and validation in many publications since its initial development, in studies modeling various phenomena such as receiver directionality [11], non-linear ultrasound propagation in biological tissue [56], photoacoustic interactions with nanoparticles [36], and elastography [37], waves in inheterogeneous liquids [29, 35].

Chapter 3

Method

We propose a novel formulation of learned acoustic reconstruction that enables the use of classical synthetic aperture techniques with recent implicit neural network architectures. Rather than learning to predict entire scenes at once from whole audio recordings as with existing methods such as BatVision [9] and Bat-G Net [16], our neural network functions on an individual point in space, receiving cropped waveform audio signals according to the expected round-trip time of a wave deflected at that point. This leaves the network with no spatial awareness other than what can be inferred from the time-aligned audio signals. Additionally, we use signed distance fields, as opposed to the depth images or occupancy grids used in BatVision and Bat-G Net respectively, giving a more continuous representation which results in a more stable training objective . A high-level illustration of our system is shown in Figure 3.1.

Unlike BatVision and Bat-G Net, we use an acoustic wave simulation to gather datasets and perform our experimentation. This enables greater experimental control and access to accurate geometric information. The details of our simulated setting are further discussed in the following sections.

In our studies, we compare the effects of the hyperparameters of synthetic focusing with neural networks, such as the number of receivers used, the length of time-aligned audio seen by the network, and the network's internal architecture. We create special-purpose datasets to measure the ability of our implicit representation to interpolate and extrapolate to unseen data, and compare the results to those

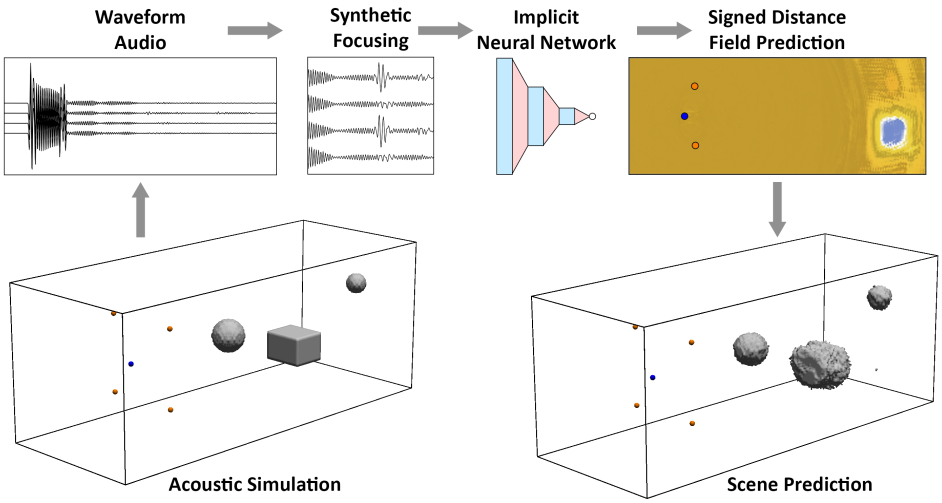


Figure 3.1: An overview of our proposed system. An acoustic simulation computes the echoes produced by various obstacles. These echoes are focused using synthetic aperture techniques at a single point in space. A neural network learns to predict the signed distance from these focused echoes, and is evaluated across many points in space to yield a signed distance field which can be rendered in 3D. Best viewed in colour.

of competing methods. Finally, we measure the ability of our model and others to generalize from very small datasets. For the results of our experiments, please refer to Chapter 4.

3.1 Synthetic Focusing

As the basic input representation to our models, we use temporally-aligned audio recordings based on the expected straight-line time-of-flight between the emitter, a given point location in the ROI, and the receivers. This synthetic focusing technique bears close resemblance to delay-and-sum techniques used in classical wave-based imaging systems, except that instead of simply summing the resulting waveforms to estimate their strength, we provide the time-aligned and windowed audio signals directly to a neural network that learns to relate them to the presence of solid obstacles. Effectively, our proposed method is a data-driven nonlinear delay-and-sum method for acoustic reconstruction. Like other implicit function neural networks,

we train on single points in space and produce scalar outputs. Unlike other implicit function neural networks, our model does not receive the desired spatial coordinates as inputs.

We assume that the emitter and receiver locations relative to one another are known, and that the speed of sound in air is constant. We perform synthetic focusing at a given spatial sampling location (x, y, z) as follows. Given the known emitter location and receiver locations (x^e, y^e, z^e) and (x_i^r, y_i^r, z_i^r) respectively, we first compute the linear distance from the emitter to the sampling location d^e, and the distance from the sampling location to the i-th receiver d_i^r. The sum of d^e and d_i^r gives us the round-trip distance that a wave would travel if deflected at that point, and from this we can compute the total expected time of flight Δt_i for each receiver as

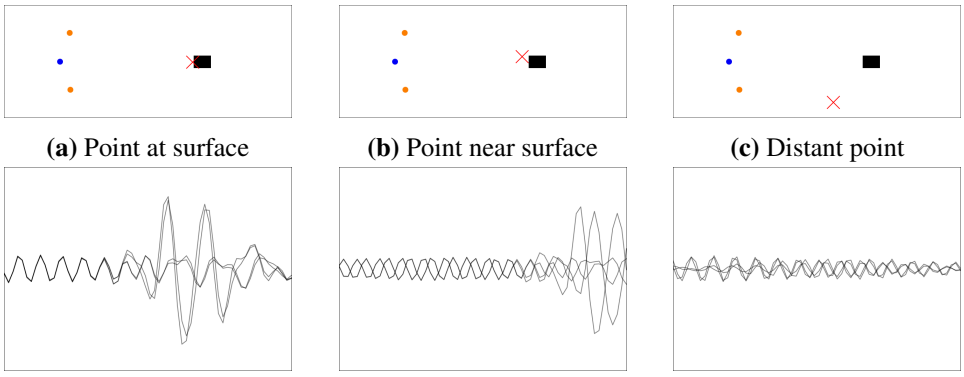
$$
\begin{aligned}
d^e &= \sqrt{(x-x^e)^2 + (y-y^e)^2 + (z-z^e)^2}, \\
d_i^r &= \sqrt{(x-x_i^r)^2 + (y-y_i^r)^2 + (z-z_i^r)^2}, \\
\Delta t_i &= \frac{d^e + d_i^r}{c},
\end{aligned}
$$

where $c = 343$ m/s is the speed of sound in air as used in the simulation. The per-receiver delay Δt_i is then used to align and window each received signal S_i, resulting in synthetically focused audio signals $\hat{S}_i$, defined as

$$
\hat{S}_i(t) = k\,(d^e)^2\,(d_i^r)^2\,S_i\left(t + f_s \Delta t_i - \frac{W}{2}\right), \tag{3.1}
$$

with $t \in \{0, \ldots, W-1\}$ where $k = 30$ dB is an experimentally-tuned gain constant, $(d^e)^2\,(d_i^r)^2$ applies an amplitude compensation based on the distance traveled assuming spherical wave propagation to and from a small deflector, and W is the window length, in samples. In our experiments, W ranges from 64 to 256 samples. It should be noted that the expected moment of arrival occurs in the middle of the focused audio at $t = \frac{W}{2}$, thus providing nearby information from both before and after this point in time.

By considering only the linear distance between the emitters, receivers, and



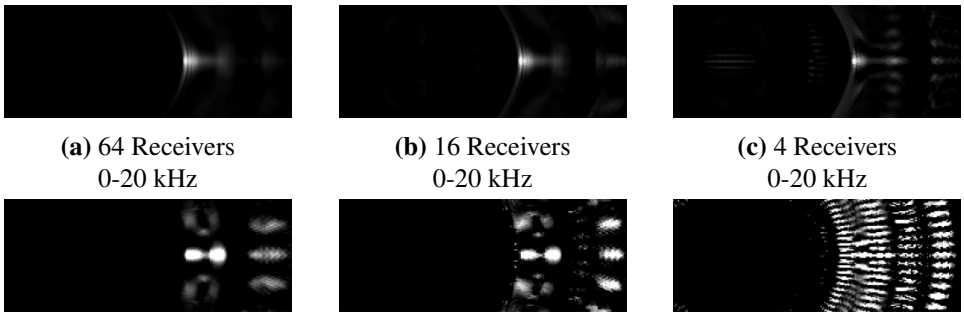
(a) Point at surface **(b)** Point near surface **(c)** Distant point

(d) Signals at surface **(e)** Signals near surface **(f)** Distant signals

Figure 3.2: Visualization of the synthetically focused signals $\hat{S}_i$ at various locations relative to the closest surface of rectangular prism. In the top row, the emitter and receivers are shown in blue and orange respectively, and the sampling location is depicted with a red 'x'. The waveforms of the focused signals for each receiver with a window size of $W = 64$ samples are overlaid in the bottom row for the same sampling locations illustrated above. Best viewed in colour.

sampling location, we implicitly are neglecting to model the effects of secondary reflections and occlusions which may be caused by other obstacles in or near the straight paths. In classical approaches, these sources of interference are overcome with abundantly many receivers and emitters, causing such interactions to become insignificant relative to true reflections. However in our learned setting, particularly when using very few receivers, our models must learn to account for this interference in order to make robust predictions in the presence of multiple distinct reflectors. When the sampling location is on the nearest surface of a solid obstacle, we expect the synthetically focused signals $\hat{S}_i$ to contain a detectable echo centered at $t = \frac{W}{2}$ across all receivers. In all other cases, we expect the $\hat{S}_i$ to contain various amounts of silence and interfering echoes from other obstacles that are unlikely to be in phase across all receivers. The effect of the proximity of the sampling location to an obstacle on the synthetically-focused signals $\hat{S}_i$ is depicted in Figure 3.2.

We illustrate how our synthetic focusing can be used even without machine learning in simple visualizations closely related to those in SAU and underwater acoustic imaging in Figure 3.3.



(a) 64 Receivers 0-20 kHz | **(b)** 16 Receivers 0-20 kHz | **(c)** 4 Receivers 0-20 kHz

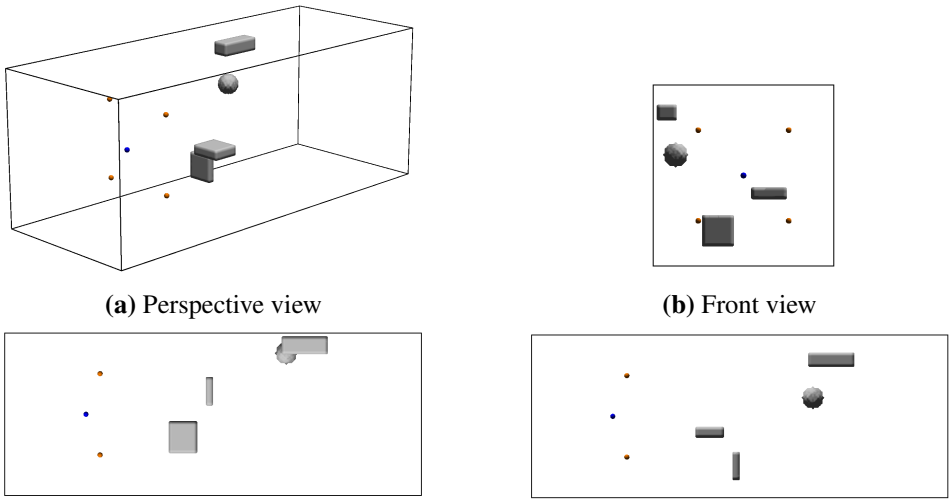
(d) 64 Receivers 18-22 kHz | **(e)** 16 Receivers 18-22 kHz | **(f)** 4 Receivers 18-22 kHz

Figure 3.3: A simple and non-learned imaging method using our synthetic focusing technique, rendered from the echoes of a single rectangular prism. Image brightness is the logarithm of the mean signal power divided by the average variance between all channels. For a large bandwidth such as the 0-20 kHz FM chirp used in the top row, this clearly localizes the obstacles. At a narrower bandwidth and with fewer receivers, the image becomes distorted by wave interference, and this simple technique breaks down.

3.1.1 Simulation Environment

Each example in our dataset is implemented as a $177 \times 69 \times 69$ centimeter volume at a resolution of 7.5 millimeters per grid cell. This choice of parameters is motivated in Section 3.1.3. Absorbing boundaries are used to prevent waves that are leaving the simulation area from returning as unwanted echoes, thus simulating an effectively infinite empty volume, and are realized using 10 perfectly-matched layers (PMLS) [2] on all sides of the simulation, resulting in a total computational grid size of $256 \times 112 \times 112$ units. We denote the longest dimension as the x direction, and the shorter dimensions as y and z. We reserve a cubic region consisting of the first 69 centimeters of the x-dimension for the emitter and receivers.

The remaining $108 \times 69 \times 69$ centimeter volume at the other end of the x-dimension is reserved for placing obstacles and is referred to as the region of interest (ROI). Obstacles are discretized and emplaced into the simulation by assigning the underlying simulation grid within the obstacle interior the acoustic properties of typical wood. All other grid locations are modelled as air at standard atmospheric



(a) Perspective view

(b) Front view

(c) Top view

(d) Side view

Figure 3.4: Our simulation volume and experimental setup rendered in 3D from multiple views, showing only the 4 receivers used in a majority of our experiments. The blue sphere represents the location of the virtual emitter, and each orange sphere denotes a receiver location. Best viewed in colour.

conditions.

The simulations used for our experiments were performed using the hardware-accelerated CUDA K-Wave executable [51] and with hardware resources kindly provided by the UBC ARC Sockeye high-performance computational cluster [54].

An illustration of our simulation volume including obstacles and hardware locations can be seen in Figure 3.4.

The initial wave condition is defined by placing a single high-pressure impulse at the location of our single emitter, in the center of the $69 \times 69 \times 69$ reserved region, as is further described in Section 3.2. This initial impulse is then smoothed by applying a Blackman window function to the spatial pressure distribution in the frequency domain, which prevents high-frequency "ringing" artefacts from arising elsewhere in space due to K-Wave's use of the spatial frequency domain. The reasons for placing a single impulse rather than simulating a realistic emitted signal such as an FM chirp are discussed in Section 3.1.2.

Once the initial conditions are fully specified, the simulation is stepped at 2×10^{-7} seconds per time step, a number that was experimentally found to prevent numerical instability, and for a sufficient number of time steps for a wave travelling through air to traverse the length of simulation twice. At every time step, the acoustic pressure is sampled at each receiver location and recorded. After the simulation is complete, we re-sample the recorded signals from the extremely high simulation step frequency of 5 MHz to the desired sampling frequency of $f_s = 96$ kHz, after applying a 10th order Butterworth low-pass filter with a cut-off frequency of $\frac{f_s}{2} = 48$ kHz to remove any high-frequency components that would otherwise introduce aliasing. The resulting acoustic recordings are each 2048 samples long, spanning a duration of approximately 21 milliseconds. We refer to the signal recorded by the i-th receiver as $S_i(t)$ where t denotes time samples and is expected to range from 0 to 2047. If t falls outside this range, S_i is defined to be 0, and if t has a fractional value, we implicitly linearly interpolate the sample values at the nearest integer locations.

3.1.2 Choice of Emitted Signal

Although K-Wave supports time-varying sound sources which can be used to directly simulate an FM chirp as used in other acoustic reconstruction datasets, we simulate only a single impulse. Because obstacles are held fixed and are not moving, the simulation is a linear time-invariant system and therefore we can later use a convolution to perfectly emulate the echo produced by any desirable emitted signal. Effectively, our dataset is made generic with respect to the emitted signal at negligible extra computational cost. Although we do not investigate the effects of different emitted signals in this work, our dataset allows such experimentation to be done efficiently in future work without needing to perform additional simulations.

We consider using only a single acoustic emitter in our current experiments for two reasons. Firstly, while extra receivers may be simulated at negligible extra cost, recording the signals from multiple emitters in a separable manner requires re-running each simulation once per additional emitter, which increases the cost of creating our datasets. Secondly, using multiple acoustic emissions would increase the total capture time of our system in a physical implementation, resulting

in slower frame rates.

In all our experiments, when training and evaluating both our own methods as well as competing methods, we use a linear FM chirp consisting of a sine wave rising in frequency from 18 kHz to 22 kHz and lasting 1 millisecond, for a total of 20 wavelengths.

3.1.3 Spatiotemporal Tradeoffs

The size and spatial resolution of the simulation grid represents an important compromise between the spatial extent of the simulation volume, the computational cost, and the maximum representable frequency. For generating a large dataset as needed for machine learning, increasing computational costs become prohibitive. On the other hand, due to the Nyquist theorem, the highest representable frequency has a wavelength equal in length to two grid cells, and thus we need a fine discretization in order to represent waves of ultrasound. To make our dataset comparable with the real world, we ideally want to simulate the highest possible frequencies used in existing ultrasound methods. For modeling acoustic interactions in close-range human environments, we would also like to consider a volume of space several meters in size. However, increasing the resolution causes at least a cubic increase in the computational workload. Considering that the total simulation time must be enough to capture echoes from the most distance possible obstacles, the total number of simulation time steps must also increase linearly with the simulation scale, leading to a quartic increase in computational cost when increasing the simulation scale.

The spatial parameters chosen for our simulations are derived from early explorations into the feasibility of a synthetic dataset, and they enable the representation of ultrasonic frequencies up to 22.8 kHz with a maximum spatial extent of just under 2 meters. While the frequency range is less than that of the physically-acquired ECHO-4CH dataset [16] it still represents a usable ultrasonic bandwidth of a few kHz and extends beyond the 20 kHz limit of the BatVision [9] dataset. Additionally, the ROI in our dataset has a physical volume nearly twice as large as the $64 \times 64 \times 64$ cm ROI used in ECHO-4CH while also offering 33% higher resolution along each axis. The maximum range in our dataset remains close to that of

ECHO-4CH at approximately 2 meters.

3.1.4 Practical Benefits for Dataset Curation

The use of acoustic simulation to generate a synthetic dataset, rather than using physical hardware to measure the echo responses of real-world obstacles environments, comes with its own tradeoffs which are outlined here. A real-world dataset firstly requires hardware, which, for producing and measuring ultrasound in air, especially with multiple receivers, can be prohibitively expensive. By using an acoustic simulation, we can simply drive and record the acoustic pressure at any desirable point in space with perfect fidelity, without having to worry about the impedance, directionality, and selective frequency response that hardware introduces. Additionally, by using a deterministic and wholly self-contained simulation, we are able to rule out environmental factors such as inaudible noise sources and varying speeds of sound due to atmospheric conditions, which future work into a physical implementation of our technique will need to address.

The final major complication for a real-world dataset is the positioning of obstacles in the environment and capturing those positions accurately for ground truth labels. In ECHO-4CH, the obstacle positions are limited to a horizontal plane and rotated at fixed intervals. This simplifies mechanically positioning and automatically generating labels in software for the obstacles, but the statistical distribution remains limited in this fashion. Conversely, in BatVision, obstacle labels are drawn from a much more diverse distribution of indoor environments, but at the cost of relying on a depth camera rather than a mathematical description as the single source of truth for training labels. Notably, the depth camera does not capture occluded objects and is often unable to estimate depth for portions of an image, resulting in missing information which a machine learning method will be taxed with modeling if no extra precautions are taken. By using a simulation consisting of a dense, volumetric grid, we are able to use the same mathematical description for both capturing the echo produced by an obstacle and for the ground truth label used during training. Additionally, we are free to place obstacles anywhere in space, without mechanical support, and we have direct access to occluded regions.

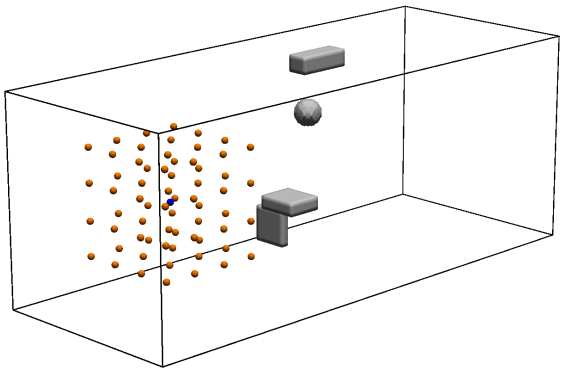


Figure 3.5: A perspective view of our simulation volume and experimental setup with all 64 receiver locations in their $4 \times 4 \times 4$ grid formation shown as orange spheres. The blue sphere at the center denotes the location of the emitter. Best viewed in colour.

3.2 Emitter and Receiver Arrangement

For each individual example in our dataset, we choose an obstacle configuration, place obstacles into the ROI, and run an acoustic simulation to obtain an impulse response for each acoustic receiver. The initial impulse is placed at the exact center of the $69 \times 69 \times 69$ cm region at the low end of the x-dimension. A total of 64 receivers are placed in a uniform grid surrounding the emitter with 4 receivers along each grid axis and with 11.5 cm between adjacent receivers, for a total span of 34.5 cm in each dimension. This locations of all these receivers within the simulation volume are shown in Figure 3.5. We denote the spatial location of the emitter as (x^e, y^e, z^e), and the location of the i-th receiver is referred to as (x_i^r, y_i^r, z_i^r).

While a total of 64 receivers would be difficult to realize in hardware, additional receivers in our virtual dataset come at no additional simulation cost, and any desirable subset of the receivers may be used during training. The receiver and emitters have an nominal sample rate of 96 kHz as discussed in Section 3.1.2. However, the effective range of practical frequencies is limited by the simulation's spatial resolution to 22.8 kHz, short of the theoretical 48 kHz due to the sampling rate, as explained in Section 3.1.3.

3.3 Obstacle Distributions

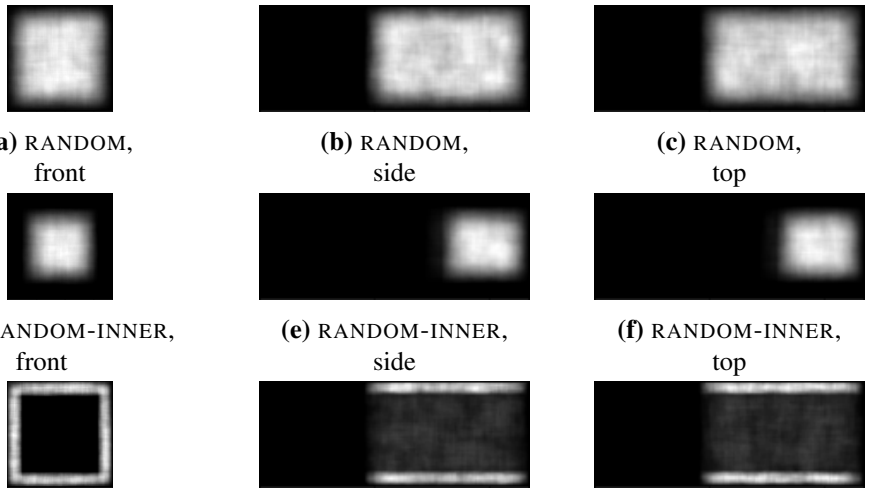
We use a total of three simulated datasets in our experiments for training and evaluating models, all consisting of a variety of random obstacle types, sizes, and orientations. While the scenes in our dataset are fairly simplistic compared to the real-world indoor environments used by Christensen *et al.* for the BatVision model [9], the sizes and positions of are more diverse than those of the ECHO-4CH dataset used to train Bat-G Net [16] as discussed further in Appending A.2. The three datasets vary primarily in which regions of space are withheld and kept free of any obstacles for the purposes of cross-validation.

The obstacles in each dataset consist of spheres and axis-aligned rectangular prisms whose positions are drawn from a uniform random distribution spanning the entire ROI or a subset thereof. The sphere diameters as well as the widths, heights, and lengths of the rectangular prisms are similarly drawn from a uniform random distribution of between 2 and 20 centimeters. Between 1 and 4 obstacles in total are placed in each example, and spheres and rectangular prisms are both chosen with a 50% probability. No precautions are taken to prevent occlusions or intersections between obstacles.

The first dataset, dubbed RANDOM, consists of obstacles placed at random anywhere in the entire ROI without constraint. The second dataset, which we refer to as RANDOM-INNER, restricts obstacles to lie only within a centered rectangular subset of the ROI with half the total volume, spanning the entire extent of the x direction of the ROI, such that an outer margin along the x-direction is kept free. The third dataset, dubbed RANDOM-OUTER, is the complement of the RANDOM-INNER dataset, consisting of obstacles located strictly outside of the same inner half, with the inner half kept empty as seen from the perspective of the emitter and receivers. Each of the RANDOM, RANDOM-INNER, and RANDOM-OUTER datasets consists of 5000 examples in total for training, 500 for validation, and 500 for testing. All three of our datasets are visually summarized in Figure 3.6.

3.4 Geometric Representation

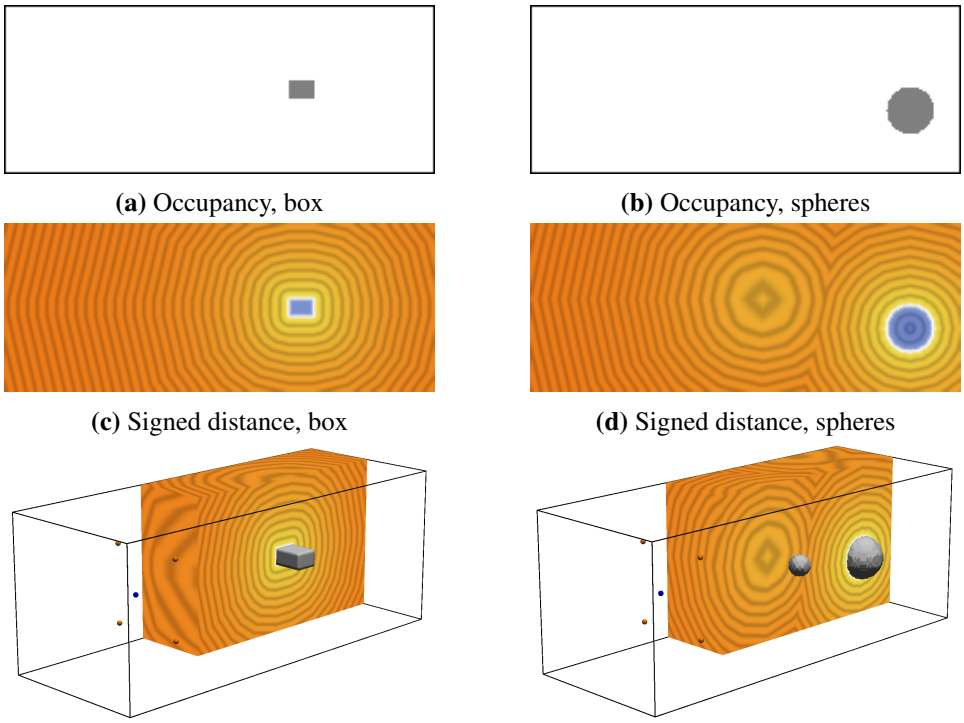
The choice of output representation has a significant effect on neural network performance in practice, where dataset sizes are limited, and domain-specific knowl-



(a) RANDOM, front
(b) RANDOM, side
(c) RANDOM, top
(d) RANDOM-INNER, front
(e) RANDOM-INNER, side
(f) RANDOM-INNER, top
(g) RANDOM-OUTER, front
(h) RANDOM-OUTER, side
(i) RANDOM-OUTER, top

Figure 3.6: Spatially-varying density of obstacles in our three datasets as viewed from different directions. These images were creating by summing the occupancy maps of all examples in each dataset before projecting along the viewing direction.

edge is often applied to improve learning. We use the signed distance to the nearest obstacle surface as our choice of geometric representation for training neural networks on synthetically focused audio. Signed distance fields offer several advantages for our setting over binary occupancy fields as a dense 3D format. Firstly, the signal varies more smoothly than occupancy fields, which jump discontinuously at obstacle boundaries and are thus less favourable for approximation by a implicit neural network. Secondly, the signed distance field is trivially convertible to an occupancy grid for evaluation purposes by using a threshold—locations with a signed distance of zero or less are occupied, and other locations are unoccupied. Thirdly, signed distance fields lend themselves naturally to convenient rendering techniques, such as sphere tracing [14] for efficient ray-intersection and perspective rendering, and shading using spatial derivatives to compute surface normal vectors. A visual comparison of binary occupancy fields and signed distance fields is shown in Figure 3.7. 3D renderings in this and other figures created using sphere-tracing to find the nearest obstacle surface along camera rays, and shading was performed



(a) Occupancy, box

(b) Occupancy, spheres

(c) Signed distance, box

(d) Signed distance, spheres

(e) Perspective rendering, box

(f) Perspective rendering, spheres

Figure 3.7: Colour-coded visualizations of signed distance fields as compared to occupancy maps, in two and three dimensions. Regions with a negative distance are shown in blue, positive distance is shown in shades of orange, and white represents a value close to zero. Notably, the signed distance field can be seen to yield to more information about the proximity of obstacles, such as the smaller sphere located away from the middle plane, as well as the depth inside of obstacles. In the 3D perspective renderings in the bottom row, the signed distance fields are shown filling one half of the ROI. Best viewed in colour.

using basic Lambertian reflection, where the brightness of a surface varies with dot product of the SDF surface normal and a light direction vector.

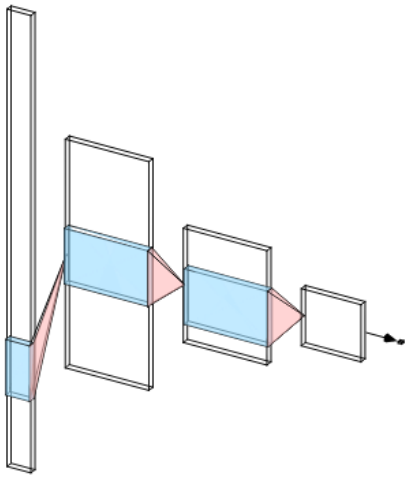


Figure 3.8: A depiction of our convolutional neural network architecture. The white rectangles denote the sizes of hidden activations, and the blue rectangles represent convolutional kernels. The final arrow represents a fully-connected layer. Best viewed in colour.

3.5 Network Model

In order to make a physical realization of our method most feasible, we consider the minimum number of receivers needed for acoustic reconstruction to be 4, which matches the hardware setup used in Bat-G Net. This is the number we use when evaluating our model against other techniques and for many of our experiments. The network architecture stated below is the result of a parameter search, which is described along with the effects of the number of receivers on network performance in Chapter 4.

We use a convolutional neural network to map from synthetically-focused temporal domain audio signals to the signed distance at a given point in space. We represent the input as a 1-dimensional tensor with 4 feature channels corresponding to each reciever and with a temporal length defined by our synthetic focusing as $W = 256$. Three sequential convolutional layers are applied, each of which has a stride of 2 and halves the input length, producing a feature vector with a temporal length of 32 and with 32 feature channels. This is then flattened, producing a

Convolutional Layers					
Layer	Input Length	Kernel Size	Padding	Output Channels	Stride
0	256	31	15	128	2
1	128	31	15	128	2
2	64	31	15	32	2

Fully-Connected Layers		
Layer	Input Features	Output Features
3	1024	1

Table 3.1: Parameters of all convolutional and fully-connected layers of our neural network model. Batch normalization layers and activation functions are not depicted.

1024-dimensional feature vector, and given to a fully-connected layer which contains a single output neuron that we interpret as the estimated signed distance at the sampling location. The parameters of all network layers are given in Table 3.1. We apply Batch Normalization [18] to the inputs of all layers and the Leaky ReLU activation function with a negative slope of 0.1 between all hidden layers.

3.5.1 Network Training

When training our network, we clamp the signed distance to a maximum of 10 centimeters away from the nearest obstacle, on the premise that the synthetically focused audio signals limit the network's global awareness, and that forcing it to learn the distance to far-away obstacles would degrade its performance near the surface of obstacles.

Like other neural networks, our model is trained on random mini-batches of geometric examples to improve stochastic gradient estimates. Additionally, within each example, we randomly select a large number of sampling locations according to a custom probability distribution which preferentially places training samples near or within the surface of obstacles. This is necessary because the simulation volume consists predominantly of empty space, and uniform random sampling would bring the network's attention mostly away from the obstacles we are interested in. We perform weighted random sampling as follows. Given the signed distance field discretized at all simulation grid locations $\mathrm{sdf}(i, j, k)$ where

$i \in \{0, ..., 107\}$ and $j, k \in \{0, ..., 68\}$, we assign the relative weight w to each grid location as

$$w(i, j, k) = e^{(-r \times \max(\text{sdf}(i,j,k) - d_{\min}, 0))}, \tag{3.2}$$

where $r \approx 69.318$ m^{-1} is a spatial decay rate causing a decrease of 50% every centimeter, and $d_{\min} = 2$ cm is the distance below which the weight is limited and remains constant. Effectively, all points inside or very close to obstacles are given a high weight, and other locations have a weight that decays exponentially with distance. We normalize all weights to sum to 1 and draw samples according to the discrete distribution

$$P(x = i, y = j, z = k) = \frac{w(i, j, k)}{\sum_{i,j,k} w(i, j, k)}, \tag{3.3}$$

where x, y, z are the grid indices being sampled. These indices are then mapped to spatial coordinates, at which point they are used to perform synthetic focusing and create network training inputs.

At each training step, our model computes the estimated signed distance $\hat{d}$ at each sampling point from the synthetically-focused signals $\hat{S}$. The loss function $\mathscr{L}$ we compute is the mean absolute error (MAE) between the estimated signed distance $\hat{d}$ and the ground truth signed distance d

$$\mathscr{L} = \mathop{\mathbb{E}}_{P} \left[\left| d - \hat{d} \right| \right] \tag{3.4}$$

where $\mathbb{E}_P$ is the expectation over all grid locations as drawn from the distribution P.

From this loss, our network is optimized using the Adam algorithm [22]. We use the PyTorch framework for hardware-accelerated numeric computing and automatic differentiation to implement and train our models [34].

Chapter 4

Results

To understand how our synthetic focusing procedure behaves as an input acoustic representation, we first train a set of neural networks using different numbers of acoustic receivers and different sizes of audio windows W, and report the effects on geometric accuracy in Section 4.1.

Then, to optimize our network model for the special case of 4 receivers and a window size of $W = 256$ samples, we perform a hyper-parameter search in Section 4.2 in which we test the effects of the convolutional kernel size and hidden feature dimensions on network behaviour. We additionally measure the effects of using the frequency domain by applying a Fourier transform to the synthetically-focused audio, as well the performance of a simple fully-connected network given the same inputs.

In our main experiment, we train our model, a re-implementation of Bat-G Net [16], and the reportedly best-performing variants of BatVision [9] using the original source code, on each of our three simulated datasets, all with the same 4 receivers. In these experiments, we seek to understand how each model is able to generalize to new scenarios from limited training data, in terms of both the number of available training examples and their spatial diversity. Additionally, we hope to explore how our domain-specific audio representation using synthetic aperture focusing affects model performance given the practical limitations of datasets.

As a simple comparison and initial baseline, we first train all models on the RANDOM training dataset and evaluate their performance on the RANDOM test set

which is drawn from the same distribution. Then, to measure the extrapolation behaviour of all models, we train each network on the RANDOM-INNER dataset and evaluate on the RANDOM-OUTER dataset, forcing models to predict geometry located outside the distribution they have previously seen. Finally, in a complementary trial, we train all models on the RANDOM-OUTER dataset and before evaluating on the RANDOM-INNER dataset, to measure how each model performs when predicting obstacles located within the same spatial extent as their training data but whose exact positions have never been observed while learning. These experiments help us to draw simple conclusions about how these models may perform on physical datasets with inherently-limited diversity and with training distributions that in general may differ from the data seen at test time. We provide further details and results in Section 4.3.

To demonstrate how our model compares to others when training datasets are small, we perform three additional comparisons between our model, Bat-G Net [16], and BatVision [9] using subsets of the RANDOM dataset for training and evaluation. This serves both to measure the relative sample efficiency of each method as well to better understand their applicability in real-world conditions where dataset curation is expensive. The RANDOM dataset contains 5000 training examples, but in these experiments, we use only the first 500, 50, and 5 examples, respectively, to train each model, in order to measure how each model's performance degrades with limiting dataset sizes. Please refer to Section 4.4 for a full description and the results of this experiment.

Finally, as justification for the many additional design choices made in our proposed input representation and network training procedure, we conduct an ablation study in Section 4.5 in which we selectively bypass or replace components of our system and quantify the effect on test performance.

4.1 Receiver Count and Window Size

To measure how our synthetically-focused audio representation behaves as an input for learned acoustic reconstruction, we first hold fixed the architecture of a simple neural network while varying the number of acoustic receivers as well as the window size W of the focused audio that the network is given as an input. We train

each network for 24 hours on the RANDOM training dataset. During training, we periodically evaluate each model's performance on the RANDOM validation, and we report its best results on this unseen data. Then at test time, we quantify model performance by densely evaluating the network at each point in the ROI volume, yielding a signed distance field at our simulation's grid resolution, which we then threshold to yield a binary occupancy map. This map is compared to the ground truth by computing the F1 score and the intersection over union (IOU), both of which range from a minimum score of 0 to a perfect score of 1. The results of this study are shown in Table 4.1.

The network model used in this study follows the same general architecture as our final model that was found after performing our hyper-parameter search, but differs in that it uses a relatively small convolutional kernel size of 5 and only 32 hidden feature channels, compared to the kernel size of 31 and the 128 hidden feature channels of our primary neural network.

In each trial, as we vary the number of receivers in use, we maintain a total spatial extend for the receiving array of 34.5 cm in the y and z directions. Similarly, the receiving array covers a span of 34.5 cm in the x direction unless $N_x = 1$ in which case the receivers all lie on a plane facing the ROI.

We observe that the model's test performance improves monotonically with the number of receivers being used. Performance also improves consistently when using a window size of $W = 128$ rather than 64. The improvements when increasing the window size to 256 relative to 128 are somewhat insignificant for this choice of network. We note that for the two choices of 16 total receivers we investigated, the planar receiver arrangement ($N_x = 1, N_y = 4, N_z = 4$) performed better than the volumetric arrangement ($N_x = 4, N_y = 2, N_z = 2$). This may in part be due to the finer cross-range resolution of the planar arrangement, which has 4×4 spacing in the y and z axes compared to the 2×2 spacing of the volumetric arrangement.

4.2 Network Hyperparameter Search

To make our proposed model more physically realizable and to allow a more fair comparison between our technique and competing models in terms of the amount of hardware being used, we use a total of 4 receivers in all our remaining studies

W	Total Receivers	N_x	N_y	N_z	F1 Score ↑	IoU ↑
64	4	1	2	2	0.2018	0.1284
	8	2	2	2	0.3459	0.2337
	16	4	2	2	0.4566	0.3214
	16	1	4	4	0.4945	0.3534
	32	2	4	4	0.5501	0.4051
	64	4	4	4	0.5839	0.4409
128	4	1	2	2	0.3247	0.2232
	8	2	2	2	0.4238	0.3028
	16	4	2	2	0.5025	0.3694
	16	1	4	4	0.5350	0.4018
	32	2	4	4	0.5959	0.4553
	64	4	4	4	0.6122	0.4731
256	4	1	2	2	0.3418	0.2439
	8	2	2	2	0.4520	0.3286
	16	4	2	2	0.5023	0.3704
	16	1	4	4	0.5532	0.4166
	32	2	4	4	0.5826	0.4434
	64	4	4	4	0.6048	0.4664

Table 4.1: Neural network performance versus receiver count and arrangement. The reported F1 scores and IOU are computed on the RANDOM validation set. N_x, N_y, and N_z denote the size of the receiver array along each axis.

for both our own models and those we compare against. These four receivers are placed at the corners of a 34.5×34.5 cm plane facing the ROI and correspond to $N_x = 1, N_y = 2, N_z = 2$ in Table 4.1. In our synthetic focusing procedure, we hold the window size fixed at $W = 256$.

Although fully-connected neural networks are known to be universal function approximators as the size or number of hidden layers is increased arbitrarily [15], we are interested in pursuing small and efficient networks which are optimized for our acoustic setting and which can generalize to unseen data, and so we compare the performance of basic network architectures and audio representations.

Firstly, we measure how a 3-layer fully-connected network compares to a 3-layer convolutional neural network, while varying the number of hidden features and kernel sizes. When passing synthetically-focused audio to each network, we

additionally compare the effect of the direct time-series representation against a frequency domain representation, in which we first apply a Fourier transform to the input audio. Our evaluation using the F1 score and IOU using the RANDOM validation set is identical to that of Section 4.1. Importantly, we do not use the RANDOM test set in this hyper-parameter search because doing so would bias our results on the test set in later experiments.

The results for all convolutional neural network variants are given in Table 4.2, and those for fully-connected neural networks are shown in Table 4.3.

For all convolutional networks, we use a stride of 2 and pad each convolution with $\frac{k-1}{2}$ units on both sides where k is the kernel size, which serves to make the output length precisely half the input length. We additionally limit the feature channels of the final convolutional layer such that the final fully-connected layer receives exactly 1024 inputs after flattening.

The single best-performing model was found to be the convolutional neural network with a kernel size of 31, 128 hidden channels, and temporal domain inputs, and this is the model we use for the remainder of our experiments. We note that other convolutional networks with larger kernel sizes perform similarly well. Temporal domain inputs improved convolutional network performance in all cases except for the smallest kernel size of 5, and the best-performing frequency domain convolutional network did significantly worse than the best temporal domain model. For convolutional models working in both the temporal and frequency domain, we observe that increasing the model size did not consistently improve performance. This may be due in part to the number of training iterations each model was subjected to, as these were trained for 24 hours in total and the larger models were slower to train and evaluate. Because we are interested in models that learn and yield predictions quickly, we consider this to be a meaningful limitation to impose. In our main experiments, we train all models for a total of 72 hours.

Among our fully-connected networks, we find that performance improved monotonically with the number of hidden features, and that frequency domain inputs out-performed time domain inputs in all cases. However, the best performance was worse than that of the convolution networks. The best-performing fully-connected models had a larger number of parameters than the best performing convolutional models in either input domain. Conceivably, we could have investigated using

Convolutional Neural Networks					
Input Domain	Kernel Size	Channels	Parameters	F1 Score ↑	IoU ↑
Temporal	5	64	33.4k	0.4853	0.3509
		128	107k	0.5359	0.4022
		256	376k	0.5367	0.4032
	15	64	97.4k	0.6357	0.5034
		128	317k	0.6736	0.5444
		256	1.12M	0.6586	0.5309
	31	64	200k	0.6744	0.5452
		128	653k	**0.7073**	**0.5816**
		256	2.32M	0.6614	0.5366
	63	32	138k	0.6613	0.5339
		64	405k	0.6770	0.5499
		128	1.32M	0.6972	0.5720
		256	4.71M	0.6213	0.4945
	127	32	278k	0.6704	0.5449
		64	814k	0.6910	0.5669
		128	2.67M	0.6740	0.5501
		256	9.50M	0.5376	0.4128
Frequency	5	64	43.7k	0.5761	0.4453
		128	127k	0.5922	0.4645
		256	417k	**0.6296**	**0.5015**
	15	64	128k	0.5582	0.4268
		128	378k	0.5941	0.4640
		256	1.25M	0.6003	0.4705
	31	64	263k	0.5629	0.4323
		128	780k	0.5876	0.4551
		256	2.57M	0.5658	0.4377
	63	64	534k	0.5552	0.4259
		128	1.58M	0.5625	0.4306
		256	5.23M	0.4999	0.3698
	127	64	1.07M	0.5687	0.4349
		128	3.19M	0.5362	0.4053
		256	10.5M	0.4505	0.3240

Table 4.2: Results of our hyper-parameter search on convolutional neural networks. Networks with a temporal domain input receive the synthetically focused signals directly, and networks with a frequency domain input receive Fourier-transformed inputs. F1 scores and IOU are computed on the RANDOM validation set.

Fully-Connected Neural Networks				
Input Domain	Hidden Features	Parameters	F1 Score ↑	IoU ↑
Temporal	64	69.8k	0.3685	0.2502
	128	128k	0.4339	0.3067
	256	328k	0.4948	0.3653
	512	788k	**0.5197**	**0.3893**
Frequency	64	70.4k	0.4462	0.3173
	128	149k	0.5064	0.3748
	256	331k	0.5528	0.4201
	512	792k	**0.5803**	**0.4457**

Table 4.3: Results of our hyper-parameter search on fully-connected neural networks. Networks with a temporal domain input receive the synthetically focused signals directly, and networks with a frequency domain input receive Fourier-transformed inputs. F1 scores and IOU are computed on the RANDOM validation set.

more hidden features in the fully-connected networks, but we chose not to in this work and were satisfied with the performance of our convolutional networks.

4.3 Model Comparison and Cross Validation

The three neural network models for learned acoustic reconstruction that we evaluate our model against in this work are Bat-G Net [16] which we re-implement as described in Appendix A.1, and the two BatVision models for waveform audio and spectrogram audio [9] using source code from Christensen *et al*.

The Bat-G Net model by design receives 4 channels of audio input in the form of pairs of spectrograms, one favouring frequency resolution and one with better temporal resolution, for a total of eight spectrograms. From our simulated 2048-sample audio signals, we likewise compute pairs of spectrograms, with a long window of 256 samples and a short window of 64 samples. These spectrograms are then resampled slightly to 256×256 to exactly match the expected input size of Bat-G Net. As an output representation, Bat-G Net produces a $64 \times 64 \times 64$ voxel occupancy grid, and we resample our dataset's occupancy grid labels between this size and our $108 \times 69 \times 69$ voxel ROI for training and evaluation. In their original experiments, Hwang *et al.* modify all labels in their dataset to fill in occluded

regions along the x-axis as being occupied, resulting in a representation with the same expressive capacity as a depth map, and we use this representation in our experiments here. Because we are additionally interested in detecting occluded obstacles, and because Bat-G Net is capable of representing occluded obstacles, we additionally train Bat-G Net on unmodified obstacle labels. We indicate whether or not obstacle labels were modified in this way with the term "back-fill" when presenting our results. Further details of our Bat-G Net reimplementation and training are given in Appendix A.1.

Both BatVision models as proposed receive only binaural audio input, and so we trivially modify the first layer of each network to accept four input channels instead of two. In the case of the waveform audio encoder, we resample our 2048-sample audio inputs to the network's expected 3200 samples. For the spectrogram encoder, as with Christensen *et al.*, we compute spectrograms from our input audio using a window size of 64, and we slightly resample this to match the network's expected 2D input size. Instead of a fully-3D output, both BatVision models produce a 2D depthmap, and so for training we project our dataset occupancy maps along the x-dimension, producing a normalized depthmap that varies from 0 at the near surface of the ROI to 1 at the far end. Additional details about our use of the BatVision models may be found in Appendix A.3.

4.3.1 Quantitative Comparison

Fundamentally, each of Bat-G Net, BatVision, and our implicit model produce a different output representation, and so we must carefully translate between these in our evaluations to draw fair conclusions. When comparing our model performance to that of Bat-G Net without back-filling, we again evaluate our network at every grid location to produce a dense signed distance field, which we then threshold to yield a binary occupancy map. This yields a geometric representation equivalent to that of the Bat-G Net model, and so for both we compute and report the F1 score and IOU against the ground truth occupancy map.

For a fair comparison between our model, Bat-G Net with back-filling, and both BatVision variants, we use 3D occupancy maps in which all occluded regions have been back-filled along the x-direction. We generate these back-filled

occupancy maps from the 2D depthmaps predicted by BatVision using an inverse projection. To create the same representation from our implicit model, we follow the same dense evaluation and thresholding procedure used elsewhere, and additionally fill in all occluded regions along the x-dimension as being occupied. The Bat-G Net model trained with back-filled obstacle labels produces this representation directly. We then compute the F1 score and IOU with respect to the similarly back-filled ground truth occupancy. It should be carefully noted that although we use the same quantitative metrics when reporting results for models with and without back-filled obstacle labels, because these scores have different physical interpretations, they should not be used to draw conclusions between models that differ in whether their predicted obstacles are back-filled.

4.3.2 Results on the RANDOM dataset

We train our model, Bat-G Net, BatVision with the waveform encoder, and BatVision with the spectrogram encoder on the RANDOM training set for a total of 72 hours. We use a batch size of 128 examples and 256 sampling locations per example for our model. For Bat-G Net, we use a batch size of 8, and for both BatVision models, we use a batch size of 16. All models are trained using the Adam optimizer with a learning rate of 2×10^{-4} and parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Throughout training for each model, we persist the network state achieving the highest performance on the RANDOM validation set and use this to compute the F1 scores and IOU as described above on the RANDOM test set. Our results are given in Table 4.4.

We find that our method performs significantly better than both Bat-G Net and BatVision with either the waveform or spectrogram encoder. Like Christensen *et al.*, we observe that the BatVision model performs better with the spectrogram encoder than with the waveform encoder. Sample outputs from the RANDOM test set are shown in 3D in figures 4.1 and 4.2.

Train on RANDOM, Test on RANDOM			
Model	Back-fill?	F1 Score ↑	IoU ↑
Bat-G Net	No	0.4094	0.2913
Ours	No	**0.6817**	**0.5506**
BatVision, waveform	Yes	0.1115	0.0723
BatVision, spectrogram	Yes	0.2770	0.1883
Bat-G Net	Yes	0.5601	0.4256
Ours	Yes	**0.7520**	**0.6289**

Table 4.4: Network test performance on the RANDOM test set after training for 72 hours on the RANDOM training set.

4.3.3 Results on the RANDOM-INNER dataset and cross-validation with RANDOM-OUTER

Next, we re-train all models exactly as in Section 4.3.2, but instead of the RANDOM dataset, we use the RANDOM-INNER dataset with its excluded outer region for both training and validation. All other training procedures remain identical. We then evaluate each model both on the RANDOM-INNER test set as well as the out-of-distribution RANDOM-OUTER test set, to measure to what extent each network is able to extrapolate and predict obstacles located outside the spatial extent of the training data. Our results are given in Table 4.5.

We observe that all networks perform better on the RANDOM-INNER dataset after being trained on the same distribution than when training and testing on the RANDOM dataset. However, when being tested on RANDOM-OUTER, which contains obstacles in regions of space that were held empty during training, we observe that while both Bat-G Net and both BatVision variants perform abysmally, our network still achieves a significant result. In particular, both Bat-G Net and BatVision simply fail to predict obstacles at all in the outer areas withheld during training, and instead predict obstacles only in the known inner region or nothing at all. We show visual results for each network on the RANDOM-INNER dataset in figures 4.3 and 4.4.

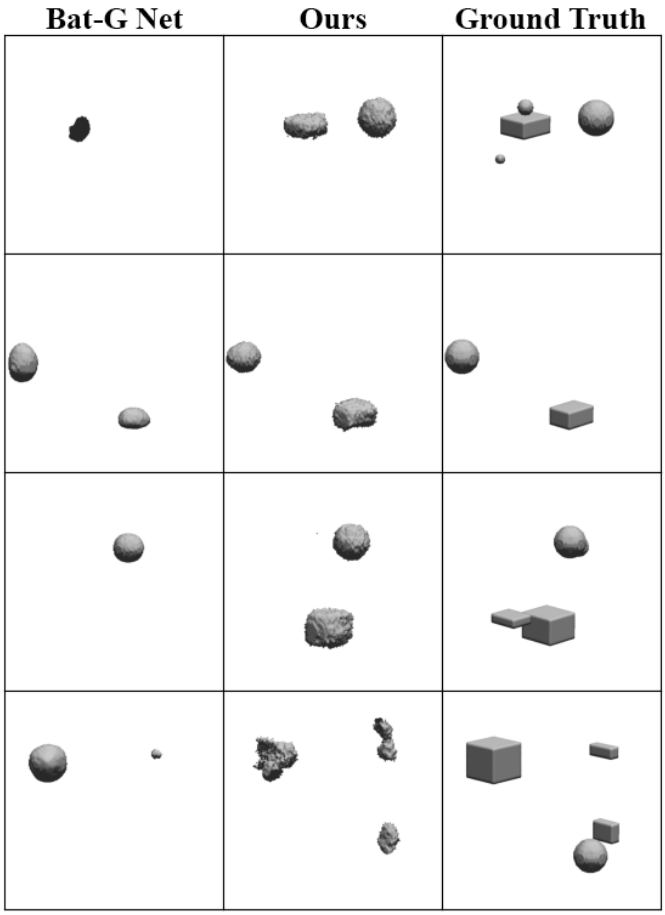


Figure 4.1: Sample outputs from Bat-G Net without back-filled obstacles and our model on the RANDOM test set after being trained on the RANDOM training set.

4.3.4 Results on the RANDOM-OUTER dataset and cross-validation with RANDOM-INNER

In a complementary experiment, we re-train all networks as before, this time using the RANDOM-OUTER train and validation sets to optimize each model. We then quantify the in-distribution performance on the RANDOM-OUTER test set as well as the out-of-distribution performance on the RANDOM-INNER test set, to determine how well each model is able to predict obstacles located in a withheld region that

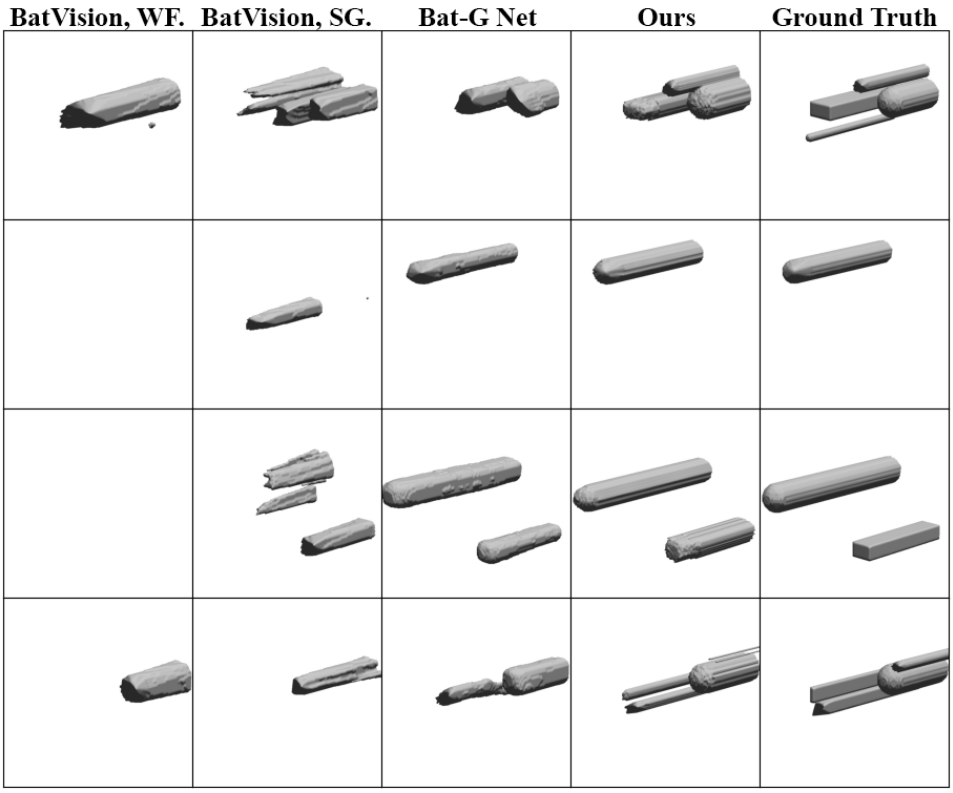


Figure 4.2: Sample outputs from all models with back-filled obstacles on the RANDOM test set after being trained on the RANDOM training set.

lies within the overall extent of the training distribution. We report the quantitative results of this experiment in Table 4.6 and provide illustrations of the predictions of each model in figures 4.5 and 4.6.

Unlike in the previous experiments, all networks performed worse when both trained and evaluated on the RANDOM-OUTER dataset than when trained and evaluated on the RANDOM dataset in Section 4.3.2. We find that BatVision with either the waveform encoder or the spectrogram encoder consistently fails to effectively learn the RANDOM-OUTER distribution. When cross-evaluating on the RANDOM-INNER dataset, we again find that while Bat-G Net and both BatVision variants perform extremely poorly, our model suffers only a fractional loss in performance. Curiously, BatVision with the waveform encoder performs better on the

Train on RANDOM-INNER				
Test Set	Model	Back-fill?	F1 Score ↑	IoU ↑
RANDOM-INNER	Bat-G Net	No	0.6129	0.4968
	Ours	No	**0.7958**	**0.6880**
RANDOM-OUTER	Bat-G Net	No	0.0000	0.0000
	Ours	No	**0.3134**	**0.2363**
RANDOM-INNER	BatVision, WF.	Yes	0.2895	0.2104
	BatVision, SG.	Yes	0.6248	0.5071
	Bat-G Net	Yes	0.7665	0.6605
	Ours	Yes	**0.8421**	**0.7490**
RANDOM-OUTER	BatVision, WF.	Yes	0.0000	0.0000
	BatVision, SG.	Yes	0.0000	0.0000
	Bat-G Net	Yes	0.0003	0.0002
	Ours	Yes	**0.3644**	**0.2902**

Table 4.5: Network test performance on the RANDOM-INNER test set and the RANDOM-OUTER test set after being trained for 72 hours on the RANDOM-INNER training dataset.

Train on RANDOM-OUTER				
Test Set	Model	Back-fill?	F1 Score ↑	IoU ↑
RANDOM-OUTER	Bat-G Net	No	0.2620	0.1810
	Ours	No	**0.5863**	**0.4561**
RANDOM-INNER	Bat-G Net	No	0.0000	0.0000
	Ours	No	**0.4635**	**0.3245**
RANDOM-OUTER	BatVision, WF.	Yes	0.0001	0.0000
	BatVision, SG.	Yes	0.0000	0.0000
	Bat-G Net	Yes	0.4820	0.3723
	Ours	Yes	**0.6594**	**0.5288**
RANDOM-INNER	BatVision, WF.	Yes	0.0130	0.0067
	BatVision, SG.	Yes	0.0000	0.0000
	Bat-G Net	Yes	0.0002	0.0001
	Ours	Yes	**0.6312**	**0.4841**

Table 4.6: Network test performance on the RANDOM-OUTER test set and the RANDOM-INNER test set after being trained for 72 hours on the RANDOM-OUTER training dataset.

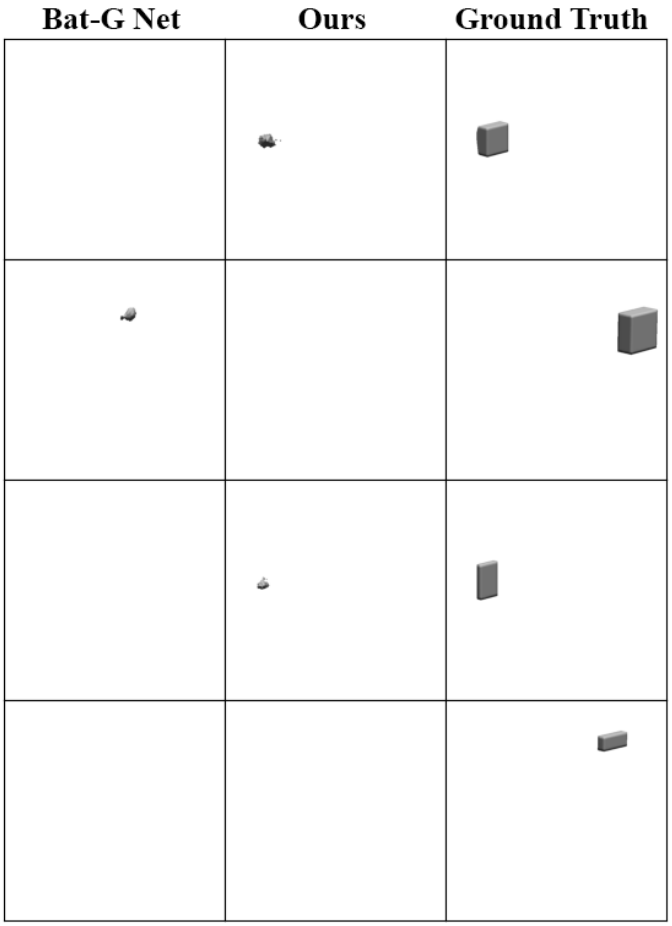


Figure 4.3: Sample outputs from Bat-G Net without back-filling and our model on the RANDOM-OUTER test set after being trained on the RANDOM-INNER training set. Empty images denote cases where a model failed to predict any obstacles.

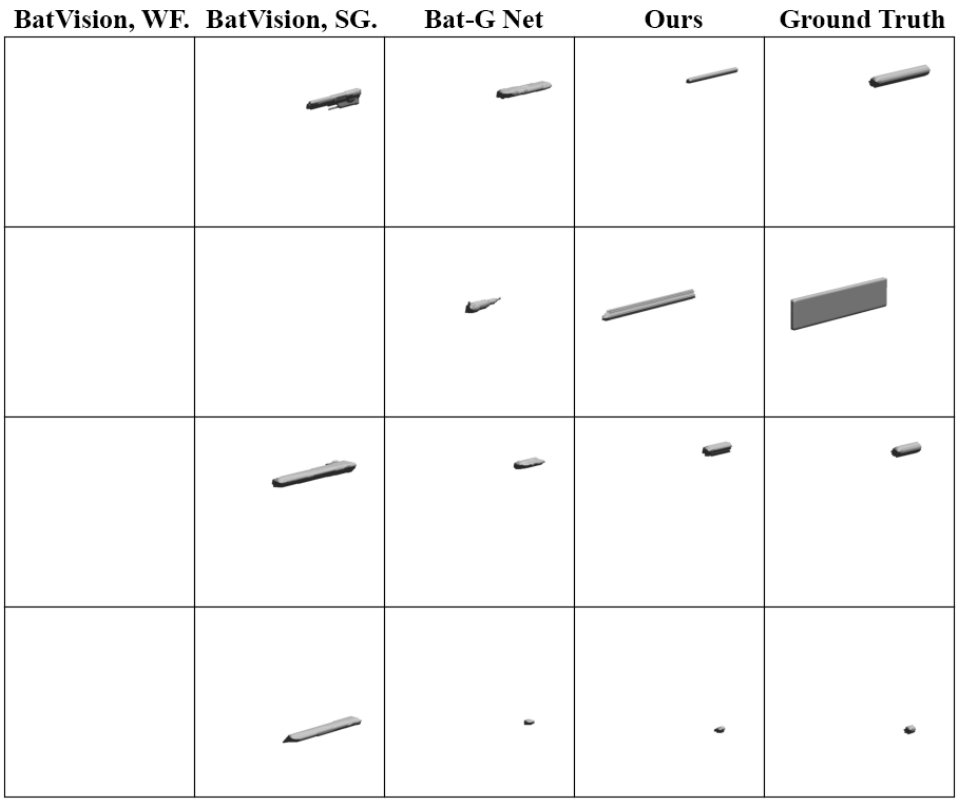


Figure 4.4: Sample outputs from all models with back-filled obstacles on the RANDOM-OUTER test set after using the RANDOM-INNER training set for learning. Empty images represent cases where a model failed to predict any obstacles.

out-of-distrubution test set than on the in-distribution RANDOM-OUTER. We attribute this to the presence of peculiar artefacts that this model produces when evaluated on the RANDOM-INNER dataset, which can be seen in the last two columns of Figure 4.6.

4.4 Model Comparison on Small Datasets

In our final experiment, we quantify how the performance of each model decreases when the training dataset becomes progressively smaller. The training and evaluation is identical to that of Section 4.3.2, except that we provide only the first 10%,

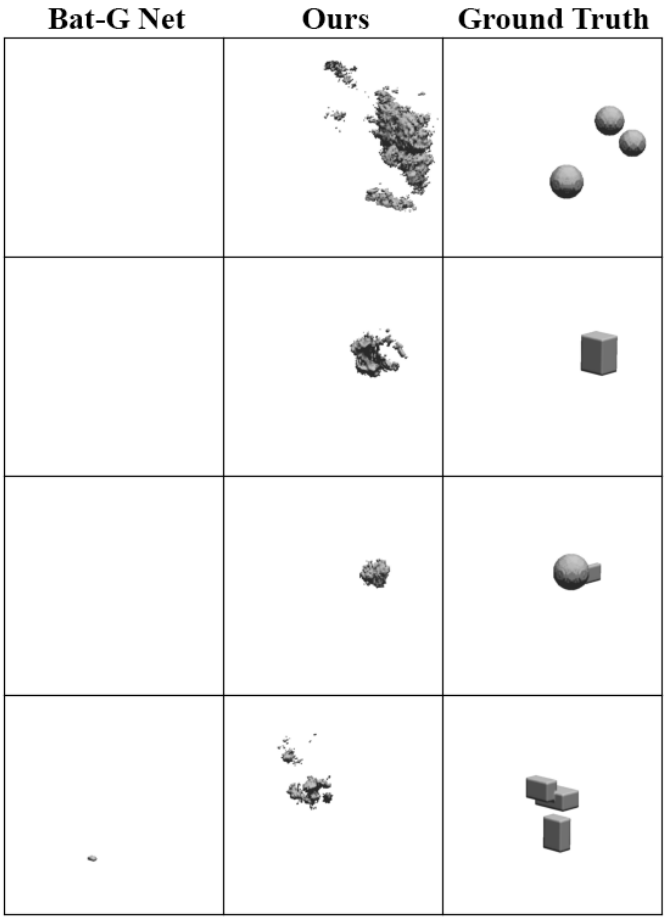


Figure 4.5: Sample outputs from Bat-G Net without back-filling and our model on the RANDOM-INNER test set after being trained on the RANDOM-OUTER training set. Empty images represent cases where a model failed to predict any obstacles.

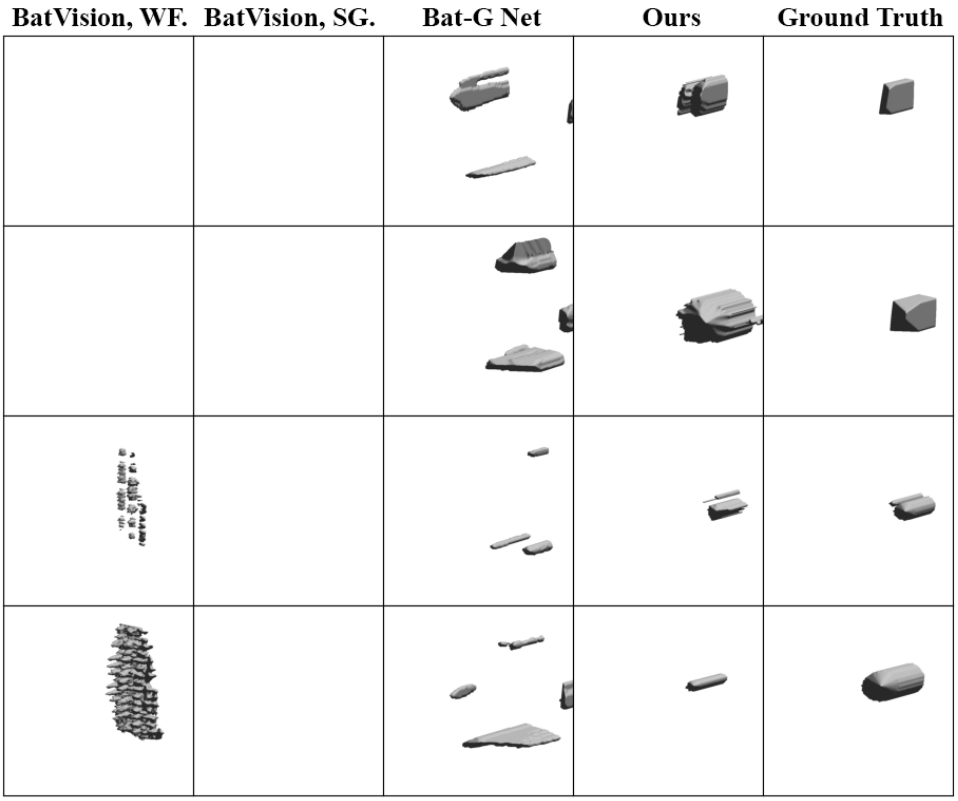


Figure 4.6: Sample outputs from all models on the RANDOM-INNER test set with back-filling after being trained on the RANDOM-OUTER training set. Empty images represent cases where a model failed to predict any obstacles.

1%, and 0.1% of examples in the RANDOM train dataset to each network during training. We however evaluate each model after training on the entire RANDOM test set. Results are given in Table 4.7 and illustrated in figures 4.7 and 4.8.

Unsurprisingly, the relative test performance of each model decreases as the training dataset is reduced in size. Most notably, we observe that the performance of Bat-G Net and both BatVision models decreases drastically as the training dataset becomes progressively smaller, but that our model suffers only moderate losses. We note that in all cases, our model outperforms Bat-G Net and BatVision when trained on exactly $10\times$ less data, and BatVision is consistently

Train on Subset of RANDOM, Test on RANDOM				
Dataset Size	Model	Back-fill?	F1 Score ↑	IoU ↑
5000	Bat-G Net	No	0.4094	0.2913
	Ours	No	**0.6817**	**0.5506**
500	Bat-G Net	No	0.0317	0.0185
	Ours	No	**0.5603**	**0.4190**
50	Bat-G Net	No	0.0116	0.0063
	Ours	No	**0.3566**	**0.2405**
5	Bat-G Net	No	0.0067	0.0038
	Ours	No	**0.1109**	**0.0666**
5000	BatVision, WF.	Yes	0.1115	0.0723
	BatVision, SG.	Yes	0.2770	0.1883
	Bat-G Net	Yes	0.5608	0.4256
	Ours	Yes	**0.7520**	**0.6289**
500	BatVision, WF.	Yes	0.0268	0.0162
	BatVision, SG.	Yes	0.0873	0.0512
	Bat-G Net	Yes	0.3133	0.2090
	Ours	Yes	**0.6657**	**0.5253**
50	BatVision, WF.	Yes	0.0000	0.0000
	BatVision, SG.	Yes	0.0441	0.0245
	Bat-G Net	Yes	0.0734	0.0434
	Ours	Yes	**0.4669**	**0.3353**
5	BatVision, WF.	Yes	0.0079	0.0043
	BatVision, SG.	Yes	0.0334	0.0179
	Bat-G Net	Yes	0.0122	0.0068
	Ours	Yes	**0.1628**	**0.1018**

Table 4.7: Results across the entire RANDOM test set after training each model on limited subsets of the RANDOM training set.

outperformed by our model with 100× fewer training examples.

4.5 Ablation Study

In our final series of experiments, we test the effect of design decisions made in our synthetic focusing procedure, our output representation, and our network training procedure, by disabling or replacing these aspects and measuring the result of training.

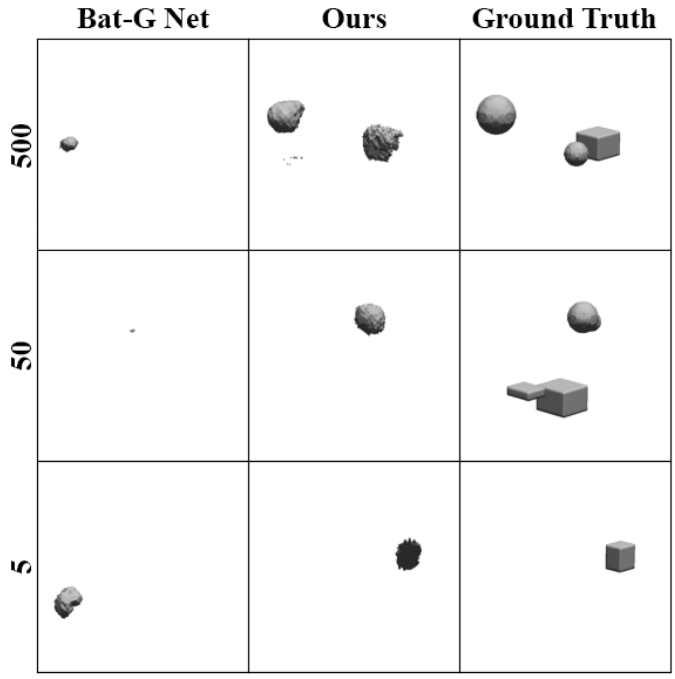


Figure 4.7: Sample outputs from Bat-G Net without back-filling and our model on the RANDOM test set after being trained on subsets of the RANDOM training set of decreasing size. Empty images represent cases where a model failed to predict any obstacles.

In the given formulation of our synthetically-focused audio inputs, we apply a distance-dependent amplification factor $(d^e)^2 (d_i^r)^2$ which serves to counteract the loss of signal strength due to spherical wave propagation. In the first trial of our ablation study, we remove this term completely. In the second trial, we similarly remove this term, but apply a sign-preserving clipped logarithm to each point in the input audio, as an alternative method for dynamic range compression, defined as

$$\hat{S}_i^*(t) = \text{sign}\left(\hat{S}_i(t)\right) \frac{\log\left(\min\left(\max\left(\left|\hat{S}_i(t)\right|, v_{\min}\right), v_{\max}\right)\right) - \log\left(v_{\min}\right)}{\log\left(v_{\max}\right) - \log\left(v_{\min}\right)} \tag{4.1}$$

where $v_{\min} = 10^{-5}$ and $v_{\max} = 1$ are experimentally-tuned constants denoting the minimum and maximum expected amplitudes of the recorded audio. Effectively,

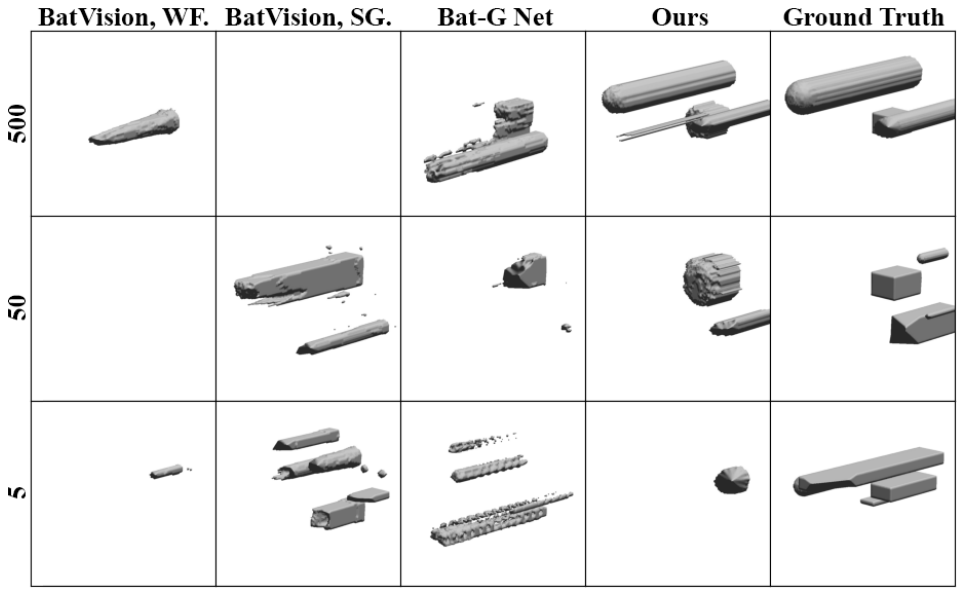


Figure 4.8: Sample outputs from all models with back-filled obstacles on the RANDOM test set after training on subsets of the RANDOM training set of decreasing size. Empty images represent cases where a model failed to predict any obstacles.

$\hat{S}_i^*(t)$ relates linearly to the logarithm of the magnitude $\hat{S}_i(t)$ within the range of $v_{\min}$ and $v_{\max}$, has a magnitude normalized to $[0,1]$, and has the same sign as the input.

To measure the impact of our importance sampling procedure in which training locations are chosen preferentially near or inside obstacle surfaces, in our next trial, we re-train a network using grid locations that are sampled from a uniform random distribution U throughout the ROI

$$U(x=i, y=j, z=k) = \frac{1}{LWH}, \tag{4.2}$$

where $L = 144$, $W = 112$, and $H = 112$ are the sizes of our simulation grid. We then train our model with an otherwise unmodified loss function

$$\mathscr{L} = \mathop{\mathbb{E}}_{U}\left[\left|d - \hat{d}\right|\right] \tag{4.3}$$

Experiment	F1 Score ↑	IoU ↑
Uniform sampling without importance-weighted loss	0.0000	0.0000
Logarithmic amplitude compensation	0.3322	0.2150
No amplitude compensation	0.4007	0.2842
Uniform sampling with importance-weighted loss	0.4449	0.3124
Binary occupancy instead of SDF	0.5052	0.3740
Control	**0.5316**	**0.3981**

Table 4.8: Results on the RANDOM test set after training on the RANDOM training set with select features disabled or replaced with alternatives.

where $\mathbb{E}_U$ is the expectation over all grid locations according to the distribution U. In a closely-related experiment, we draw samples uniformly, but weight the contribution of each sampling location to the loss according to the custom probability distribution P in which the proximity and interior of obstacles is given preference. For this trial, the loss function we use is

$$\mathscr{L} = \mathop{\mathbb{E}}_{(x,y,z)\sim U} \left[P(x,y,z) \left| d - \hat{d} \right| \right]. \tag{4.4}$$

This achieves importance sampling similarly to drawing sampling locations directly from our custom distribution P, except that a majority of sampling locations now lie in empty space where they receive a low weight.

Finally, we replace the signed distance field representation with a binary occupancy field, whose possible values are 1 denoting occupied and -1 denoting unoccupied. We report all test results and those of an unmodified neural network in Table 4.8.

From these results, we conclude that each of our distance-dependent amplitude compensation, importance sampling, and signed distance field representation improves network performance. Most notably, training on grid locations drawn uniformly with an unweighted loss function resulted in the network failing to predict any obstacles, likely due to the fact that the vast majority of spatial locations in our dataset are empty space. Using an importance-weighted loss with uniformly sampling yielded performance that was similar to but not as good as that of directly sampling locations from the custom distribution P. The quantitative benefits of us-

ing the signed distance field over a binary occupancy grid in this setting appear to be relatively minor, but significant nonetheless.

Chapter 5

Discussion

In our first experiment testing the behaviour of our synthetically-focused audio representation in Section 4.1, we varied the number of receivers that a simple convolutional network receives, and we found that performance improved readily with the number of receivers that are used. We find this to be in agreement with the trend in acoustic imaging literature towards increasing numbers of receivers for better visualizations, and thus validating for our input representation in this context. In practice, when creating a physical implementation of our system, this also means that the quality of predicted visualizations may be improved simply by using more acoustic receivers.

Across all our experiments comparing our methods to those of Bat-G Net and BatVision, we find our method performs significantly better in every setting that we tested. In our simplest comparison experiment in Section 4.3.2 where we trained and tested models on the entire RANDOM dataset, we conclude that our models perform significantly better than other models. But importantly, in our remaining experiments on cross-validation in Section 4.3.3 and Section 4.3.4, as well on the effect of limiting dataset sizes in Section 4.4, we find that the Bat-G Net and BatVision fail to generalize to obstacles in previously-unseen regions of space and require very large numbers of training examples to achieve noteworthy performance. We believe this to be due to the use of dense representations, in which a single output neuron is devoted to each spatial location, and each such neuron receives

Model	Learnable Parameters
Bat-G Net	25585129
BatVision, waveform input	47563201
BatVision, spectrogram input	71533697
Ours	652585

Table 5.1: Total number of learnable scalar parameters for each model used in our experiments.

separate treatment by design. In stark contrast, our proposed model, which has no direct spatial awareness, is able to meaningfully interpolate, extrapolate, and learn from relatively few examples. We believe this marks an important step towards practical adoption of in-air learned acoustic reconstruction, due to the technical challenges of dataset curation.

We also hold these results to be especially significant given that our proposed model has only a small fraction of the number of learnable parameters of Bat-G Net or BatVision as shown in Table 5.1. We attribute the ability of our proposed model to generalize well despite its size and lack of spatial awareness to the amount of usable information already present in our synthetically-focused audio inputs.

In our simulated datasets, we consider only small reflectors. This results in relatively quiet echoes, due to the compounding signal loss of spherical wave propagation both to and from the obstacles. In our synthetic focusing process, we compensate for this explicitly in our input representation by applying a distance-dependent amplification. In Bat-G Net, the ECHO-4CH dataset used by Hwang *et al.* is similar with respect to its small reflectors, and the use of a spectrogram, which computes the logarithm of the input signal strength across time and frequency, helps to account for such tiny echoes. The BatVision model with the spectrogram encoder shares this benefit, but the BatVision model for waveforms is instead forced to account for an enormous dynamic range. In the BatVision dataset of indoor office environments, where typical reflectors are large rooms, hallways, and furniture, we expect the returning echoes to be stronger than in the case of small reflectors, and we attribute the poor performance of BatVision with the waveform encoder in our experiments to this effect.

Model	Dense Evaluation Time (s)
Bat-G Net	0.018
BatVision, waveform input	0.0038
BatVision, spectrogram input	0.0064
Ours	9.1

Table 5.2: Total time taken by each model to produce an estimate for every point in the experimental volume at our simulation grid resolution. Bat-G Net and BatVision produce a dense representation directly, but our model must be re-evaluated separately at every point. Run times were computed on an Nvidia GeForce RTX 2080 Ti GPU.

5.1 Limitations

One major drawback of our implicit neural network formulation is that as proposed, it must be separately re-evaluated at every location in space, without any clear opportunity for reducing redundant work when predicting each point in a large volumetric grid. Although our model is much smaller in terms of the number of trainable parameters, in our evaluations, we find our model to be orders of magnitude slower to evaluate on the entire simulation grid relative to Bat-G Net and BatVision, which give predictions for the entire scene in a single invocation. We report the time taken by each model to predict an entire volume in Table 5.2. For practical adoption, one major avenue for future work will be finding more efficient formulations for our method, such as using the estimated signed distance field to adaptively avoid extra work when making visualizations, different network architectures that are able to exploit parallel workloads better, or possibly different forms of synthetically-focused input audio.

By working in simulation, we have been able to carefully control our dataset curation, our geometric training labels, and sources of noise and measurement errors, and in this clean virtual setting, we have shown that our implicit neural network with synthetically-focused audio as an input representation performs far better than convolutional neural networks operating on whole audio recordings and whole scenes at once in a dense representation. However, this does not directly allow us to conclude that our method can be easily realized in a physical setting. One limitation we anticipate is the need for accurate geometric labels during train-

ing which will be expensive to produce from physical environments. Fortunately, as we have shown, our implicit method is very sample-efficient, and thus can be expected to require fewer real-world examples than competing methods. While we anticipate our model to still perform well in the presence of light or moderate noise when trained on similarly noisy data, this remains to be demonstrated. Additionally, in our distance-based amplitude compensation, we have assumed that recorded echoes are produced by small deflectors. In human environments, in the presence of large walls, rooms, and furniture, the returning echoes may be much stronger than those produced by small obstacles, and consequently models will need to learn to account for a much larger dynamic range to produce useful predictions in such settings. Extra precautions in the design and training of the neural network may be needed to prevent large echoes from over-powering those from small objects.

By design, our neural network has no direct spatial awareness, and is only able to reason about the presence of reflective obstacles using audio signals that have been transparently shifted and narrowed in time. While our network clearly outperforms existing fully-convolutional networks, this lack of information has the consequence that our network is less able to reason globally or about distant interactions, such as secondary reflections and occluding obstacles near the receiver but far from the sampling location. While it is theoretically possible to see around corners using sound, our current formulation limits this by assuming only straight paths and primary reflections.

5.2 Future Work

The robustness of our method to environmental noise and measurement errors remains to be evaluated, and may be further explored in subsequent research. While we expect these factors to be detrimental to our model's performance, we also note that our method is easy to extend with the use of additional receivers which can be expected to improve performance. Additionally, while our current neural network does not explicitly know the spatial locations of its receivers and treats each input channel separately, we believe an even more generalized model may result from giving the model some limited awareness of its receiver arrangement and a shared

learnable parameterization of each receiver's audio signal conditioned on its location. If done correctly, this would incorporate the inherent symmetry of the receiver arrangement into the model's training procedure for even better sample efficiency. At present, our model is able to learn different behaviours for each incoming audio channel separately, due to our use of one feature channel per audio recording in our input convolutional layers.

In our simulation, we have limited ourselves to obstacles with a single acoustic material and static scenes. When acquiring a physical dataset, if the acoustic properties of obstacles are captured, it may be straightforward to augment our neural network model with additional outputs for the material properties at any sampling location, and it may then be determined to what extent such predictions are possible or reliable. Additionally, if obstacles are allowed to move while being measured, and this motion is accurately captured, it may similarly be possible to estimate the instantaneous velocity of obstacles with our model due to Doppler shifting, although this may require reformulating our time-of-flight calculations when synthetically focusing the recorded audio signals.

An additional course for future work may be ways to relax the straight-line assumptions made by our synthetic focusing procedure, in order to better model secondary reflections and occlusions. This may be possible using recursive techniques, for example by using estimates of one nearby obstacle location to find secondary reflections it contributes to. However, this may compromise the efficient parallel nature of our proposed technique in which all sampling locations may be evaluated independently.

In the shorter term however, the practical usability of our model in real life depends on its visualization speed which presently is very slow, and on the availability of detailed, volumetric scene information, which must be somehow acquired in a physical dataset. Addressing these two factors—for example with a more efficient parallel formulation of our model and a dataset capture system with acoustic hardware and precise measurements of the nearby physical environment—means that our system will be ready for interactive applications.

Chapter 6

Conclusion

We have proposed a novel audio representation for learned acoustic reconstruction inspired by synthetic aperture techniques, and shown in simulation that this representation leads to far better performance and generalization than that of competing models which use fully-convolutional neural networks and dense geometric representations. Our implicit formulation means that trained models can be made much smaller and require smaller training datasets relative to other existing networks which use dense representations and which consider entire scenes at once. Because of its sample efficiency, our model can be trained using fewer examples which makes it more readily applicable in the real world where dataset curation remains a significant challenge.

Bibliography

[1] S. Bennett, D. Peterson, D. Corl, and G. S. Kino. A real-time synthetic aperture digital acoustic imaging system. In *Acoustical Imaging*, pages 669–692. Springer, 1982. → page 8

[2] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200, 1994. → page 20

[3] N. Blaunstein, V. Yakubov, and T. . F. eBooks A-Z. *Electromagnetic and acoustic wave tomography: direct and inverse problems in practical applications*. CRC Press, Taylor & Francis Group, Boca Raton, 2019. → pages 1, 4, 6, 8

[4] N. Bleistein and S. H. Gray. From the hagedoorn imaging technique to kirchhoff migration and inversion. *Geophysical Prospecting*, 49(6): 629–643, 2001. → page 3

[5] C. Burckhardt, P.-A. Grandchamp, and H. Hoffmann. An experimental 2 mhz synthetic aperture sonar system intended for medical use. *IEEE Transactions on Sonics and Ultrasonics*, 21(1):1–6, 1974. → page 8

[6] C. Chi, S. ebooks Engineering, and S. O. service). *Underwater Real-Time 3D Acoustical Imaging: Theory, Algorithm and System Design*. Springer Singapore, Singapore, 1st 2019. edition, 2019. ISBN 9789811337437; 9789811337451; 9811337446; 9811337438; 9789811337444; 9811337454. → pages 3, 4, 9, 10

[7] R. Y. Chiao and X. Hao. Coded excitation for diagnostic ultrasound: a system developer's perspective. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 52(2):160–170, 2005. → pages 2, 6

[8] J. H. Christensen, S. Hornauer, and S. Yu. Batvision with gcc-phat features for better sound to vision predictions. 2020. → pages 4, 5

[9] J. H. Christensen, S. Hornauer, and S. X. Yu. Batvision: Learning to see 3d spatial layout with two ears. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1581–1587, 2020. doi:10.1109/ICRA40945.2020.9196934. → pages 4, 5, 13, 16, 23, 26, 32, 33, 38, 69

[10] P. Corl, G. Kino, C. DeSilets, and P. Grant. A digital synthetic focus acoustic imaging system. In *Acoustical Imaging*, pages 39–53. Springer, 1980. → page 8

[11] B. Cox and B. Treeby. Effect of sensor directionality on photoacoustic imaging: a study using the k-wave toolbox. In *Photons Plus Ultrasound: Imaging and Sensing 2010*, volume 7564, page 75640I. International Society for Optics and Photonics, 2010. → page 15

[12] W. S. Gan, E. Corporation, and I. Books24x7. *Acoustical imaging: techniques and applications for engineers*. Wiley, Chichester, West Sussex, U.K, 1. aufl. edition, 2012. ISBN 1119941083; 9781119941071; 1119941075; 9781119941088; 0470661607; 9780470661604. → page 2

[13] D. Garcia, L. Le Tarnec, S. Muth, E. Montagnon, J. Porée, and G. Cloutier. Stolt's fk migration for plane wave ultrasound imaging. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 60(9):1853–1867, 2013. → page 11

[14] J. C. Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. → page 27

[15] K. Hornik. Multilayer feed-forward networks are universal approximators. *Artificial neural networks: Approximation and learning theory*, 1992. → page 35

[16] G. Hwang, S. Kim, and H.-M. Bae. Bat-g net: Bat-inspired high-resolution 3d image reconstruction using ultrasonic echoes. In *Advances in Neural Information Processing Systems*, pages 3720–3731, 2019. → pages 4, 5, 13, 16, 23, 26, 32, 33, 38, 66

[17] A. Ibrahim and M. D. Sacchi. Fast simultaneous seismic source separation using stolt migration and demigration operators. *Geophysics*, 80(6): WD27–WD36, 2015. → page 11

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. → page 30

[19] J. Iseringhausen and M. Hullin. Non-line-of-sight reconstruction using efficient transient rendering. *ACM transactions on graphics*, 39(1):1–14, 2020. → pages 6, 12

[20] J. A. Jensen, S. I. Nikolov, K. L. Gammelmark, and M. H. Pedersen. Synthetic aperture ultrasound imaging. *Ultrasonics*, 44:e5–e15, 2006. → pages 2, 3, 8

[21] S. Kim, G. Hwang, and H.-M. Bae. Bat-g2 net: Bat-inspired graphical visualization network guided by radiated ultrasonic call. *IEEE access*, 8: 189673–189683, 2020. → pages 4, 5, 14

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. → page 31

[23] D. Kish. Human echolocation: How to "see" like a bat. *New Scientist*, 202 (2703):31–33, 2009. → page 1

[24] K. Langenberg, M. Berger, T. Kreutter, K. Mayer, and V. Schmitz. Synthetic aperture focusing technique signal processing. *NDT international*, 19(3): 177–189, 1986. → page 9

[25] D. B. Lindell, G. Wetzstein, and V. Koltun. Acoustic non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6780–6789, 2019. → pages 4, 13

[26] D. B. Lindell, G. Wetzstein, and M. O'Toole. Wave-based non-line-of-sight imaging using fast f-k migration. *ACM Trans. Graph. (SIGGRAPH)*, 38(4): 116, 2019. → pages 6, 12

[27] G. R. Lockwood, J. R. Talman, and S. S. Brunke. Real-time 3-d ultrasound imaging using sparse synthetic aperture beamforming. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 45(4):980–988, 1998. → page 9

[28] D. Loewenthal, L. Lu, R. Roberson, and J. Sherwood. The wave equation applied to migration. *Geophysical Prospecting*, 24(2):380–399, 1976. → page 10

[29] E. Martin, J. Jaros, and B. E. Treeby. Experimental validation of k-wave: Nonlinear wave propagation in layered, absorbing fluid media. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 67(1): 81–91, 2019. → page 15

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. → page 5

[31] V. Murino and A. Trucco. Three-dimensional image generation and processing in underwater acoustic vision. *Proceedings of the IEEE*, 88(12): 1903–1948, 2000. → pages 9, 10

[32] S. Nikolov and J. A. Jensen. Comparison between different encoding schemes for synthetic aperture imaging. In *Medical Imaging 2002: Ultrasonic Imaging and Signal Processing*, volume 4687, pages 1–12. SPIE, 2002. ISBN 0277-786X. → pages 2, 6, 9

[33] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. → page 5

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. → pages 31, 66

[35] S. Paul, S. Rajendran, and M. S. Singh. k-wave toolbox for studying elastic property in photoacoustic imaging. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 91–95. IEEE, 2019. → page 15

[36] M. Pramanik et al. Simulating photoacoustic waves from individual nanoparticle of various shapes using k-wave. *Biomedical Physics & Engineering Express*, 2(3):035013, 2016. → page 15

[37] F. Prieur and S. Catheline. Simulation of shear wave elastography imaging using the toolbox "k-wave". In *Proceedings of Meetings on Acoustics 172ASA*, volume 140, page 020002. Acoustical Society of America, 2016. → page 15

[38] T. D. Rossing, S. ebooks Physics, and Astronomy. *Springer handbook of acoustics*. Springer, Dordrecht, 2nd edition, 2014;2015;. ISBN 9781493907540;1493907549;. → page 8

[39] J. M. Sanches, A. Laine, J. S. Suri, S. ebooks Biomedical, and L. Sciences. *Ultrasound imaging: advances and applications*. Springer, New York, 1. aufl.;2012; edition, 2012;2011;. ISBN 1461411793;9781461411796;. → page 3

[40] S. Schmidt, N. Duric, C. Li, O. Roy, and Z.-F. Huang. Modification of kirchhoff migration with variable sound speed and attenuation for acoustic imaging of media and application to tomographic imaging of the breast. *Medical physics (Lancaster)*, 38(2):998–1007, 2011. → page 3

[41] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf. → page 5

[42] J. Steckel and H. Peremans. Batslam: Simultaneous localization and mapping using biomimetic sonar. *PloS one*, 8(1):e54076, 2013. → page 13

[43] B. Stern. The basic concepts of diagnostic ultrasound1. https://teachersinstitute.yale.edu/curriculum/units/1983/7/83.07.05.x.html, 1983. Accessed: 2021-08-09. → page 2

[44] R. H. Stolt. Migration by fourier transform. *Geophysics*, 43(1):23–48, 1978. → page 10

[45] S. Su, F. Heide, G. Wetzstein, and W. Heidrich. Deep end-to-end time-of-flight imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6383–6392, 2018. → page 6

[46] H. Sun, F. Yang, F. Meng, Z. Zhang, C. Gao, and M. Liu. A topographic kirchhoff dynamic focused beam migration method based on compressed sensing. *IEEE access*, 6:56666–56674, 2018.

[47] H. Sun, Z. Zhang, G. Hu, F. Meng, C. Gao, M. Liu, J. Tang, Y. Wang, and F. Yang. Kirchhoff beam migration based on compressive sensing. *IEEE access*, 6:26520–26529, 2018.

[48] Y. Tasinkevych, I. Trots, A. Nowicki, and P. A. Lewin. Modified synthetic transmit aperture algorithm for ultrasound imaging. *Ultrasonics*, 52(2): 333–342, 2012. → pages 2, 3, 6, 8, 9

[49] L. Thaler, G. M. Reich, X. Zhang, D. Wang, G. E. Smith, Z. Tao, R. S. A. B. R. Abdullah, M. Cherniakov, C. J. Baker, D. Kish, and M. Antoniou. Mouth-clicks used by blind expert human echolocators - signal description and model based signal synthesis. *PLoS computational biology*, 13(8): e1005670–e1005670, 2017. → page 1

[50] D. Trad, R. Siliqi, G. Poole, and J.-L. Boelle. Fast and robust deblending using apex shifted radon transform. In *SEG Technical Program Expanded Abstracts 2012*, pages 1–5. Society of Exploration Geophysicists, 2012. → page 11

[51] B. E. Treeby and B. T. Cox. k-wave: Matlab toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of biomedical optics*, 15(2):021314, 2010. → pages 14, 15, 21

[52] B. E. Treeby, J. Jaros, D. Rohrbach, and B. Cox. Modelling elastic wave propagation using the k-wave matlab toolbox. In *2014 IEEE international ultrasonics symposium*, pages 146–149. IEEE, 2014. → page 15

[53] A. Trucco, M. Palmese, and S. Repetto. Devising an affordable sonar system for underwater 3-d vision. *IEEE transactions on instrumentation and measurement*, 57(10):2348–2354, 2008. → page 4

[54] UBC Advanced Research Computing. Ubc arc sockeye, 2019. URL https://arc.ubc.ca/ubc-arc-sockeye. → page 21

[55] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3(1):1–8, 2012. → page 11

[56] K. Wang, E. Teoh, J. Jaros, and B. E. Treeby. Modelling nonlinear ultrasound propagation in absorbing media using the k-wave toolbox: experimental validation. In *2012 IEEE International Ultrasonics Symposium*, pages 523–526. IEEE, 2012. → page 15

[57] L. V. Wang and T. . F. eBooks A-Z. *Photoacoustic imaging and spectroscopy*, volume 144. CRC, Boca Raton, 1 edition, 2009;2017;. ISBN 1420059912;9781420059915;. → page 6

[58] C. Wykes, F. Nagi, and P. Webb. Ultrasound imaging in air. In *International Conference on Acoustic Sensing and Imaging, 1993.*, pages 77–81. IET, 1993. → page 12

Appendix A

Supporting Materials

A.1 Bat-G Net Implementation and Training

The re-implementation of Bat-G Net used in our studies follows the architecture described by Hwang *et al.* [16] as closely as possible in nearly all regards, but differs in a number of minor aspects where we made simplifications that we describe in this section.

Firstly, unlike Hwang *et al.*, we do not use deformable convolutions and instead use standard convolutional layers with identical kernel sizes when defining the neural encoder which maps from spectrograms to a latent representation, simply because at the time of writing, deformable convolutions are not implemented in the PyTorch framework [34]. We additionally do not use dropout when training our Bat-G Net implementation.

According to our best understanding of the network diagram given in Figure 4 of Hwang *et al.*'s paper, and according to samples of source code shared by the paper authors, the fully-connected layer at the middle of the Bat-G Net model has 65,536 input neurons and 65,536 output neurons. This implies a total of 4,294,967,296 learnable weights for the fully-connected layer alone, which, assuming 4 bytes per floating point number in hardware, has a memory footprint of 16 GB which exceeds the total memory of most modern high-end GPUs. We believe this to be a either a misunderstanding or a misrepresentation of the network architecture that was used in the experiments of Hwang *et al.*, and so in our

re-implementation, we limit the size of this fully-connected layer to 4096 input neurons and 4096 output neurons, for a total of 16,777,216 learnable parameters and 64 MB of memory for the weight matrix. We similarly adapt the feature dimensions of the nearby convolutional layers. As shown in Table 5.1, this results in a total number of learnable parameters similar to that of the BatVision models.

As proposed by Hwang *et al.*, the Bat-G Net model produces a $64 \times 64 \times 64$ volumetric output containing two output channels which are interpreted as the logits of a binary probability distribution, as demonstrated by the use of the cross-entropy loss in Equation 9 of their paper. Instead, we use a single output channel in our final layer of the Bat-G Net decoder, which we interpret directly as the occupancy at each grid location, ranging from 0 as unoccupied to 1 as occupied. During training, we instead minimize the mean squared error

$$L(\hat{y}, y) = \frac{1}{64^3} \sum_{i=1}^{64} \sum_{j=1}^{64} \sum_{k=1}^{64} (y_{i,j,k} - \hat{y}_{i,j,k})^2. \tag{A.1}$$

To verify that our re-implementation remains faithful to the original, we train our Bat-G Net model on the ECHO-4CH dataset as used by Hwang *et al.* and compute its test results after training for 24 hours. Like Hwang *et al.*, we reserve 2600 examples for testing and use the remainder for training. In the ECHO-4CH dataset, examples are ordered first by their obstacle type and secondly by their orientation, and each obstacle configuration is repeatedly recorded up to 5 times. According to correspondence with Hwang *et al.*, in their original experiments, a purely random partition was used. However, because of the repeated measurements, we believe this may result in an unrealistic advantage during testing, as it implies that each test example may have multiple similar examples in the training set. For this reason, in addition to the random train/test split used by Hwang *et al.*, we perform a second training experiment where we use the final 2600 examples in the ordered dataset for testing and the remaining first portion for training, thus largely withholding unique pairs of obstacles and repetitions of exact configurations from the test set. We report the F1 score and IOU of both our trained Bat-G Net models in Table A.1.

With our slightly modified Bat-G Net re-implementation, on the ECHO-4CH dataset, we achieve a result that improves on the F1 score of 0.896 reported by

Train/Test Split	F1 Score ↑	IOU ↑
Random	0.9863	0.9742
Contiguous	0.9303	0.8721

Table A.1: Test results of our Bat-G Net re-implementation after training for 24 hours on the ECHO-4CH dataset, using a contiguous or random partition of the full dataset for training and testing.

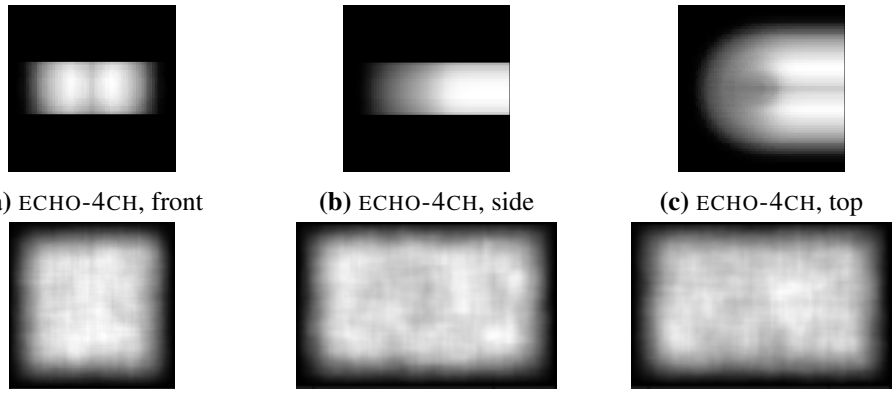
Hwang *et al.* using both the random and the contiguous train/test split. We present this as validation that our re-implementation is at least as good as the one used by Hwang *et al.* in their original work and thus a fair comparison in our broader experiments. We additionally observe that with the contiguous train/test split, the model performance suffers, suggesting the the test set in this case truly was more distinct from the corresponding training data than in the purely random split.

A.2 Comparing the ECHO-4CH and RANDOM Datasets

While Bat-G Net achieves very high test performance on the ECHO-4CH dataset, we found its performance to be significantly lower on our RANDOM dataset even though it is conceptually very similar. In this section, we give a variety of reasons why this may be expected given only differences in these two datasets.

The obstacles in ECHO-4CH follow a deterministic but limited distribution, remaining fixed on a plane and being rotated at fixed intervals. From the visualizations summarizing the obstacles of both datasets we provide in Figure A.1, it is apparent that while our RANDOM dataset places obstacles throughout the entire volume of the ROI rather uniformly, the ECHO-4CH dataset uses less than a third of the available volume. Consequently, the distribution of obstacle configurations that a model trained on this distrubition is much smaller.

A further reason for losses in performance when training on the RANDOM dataset relative to the ECHO-4CH dataset lies in the differences in the audio resolution used in both datasets. The spectrogram of the ECHO-4CH dataset are each 256×256 pixels in size, and are computed from high-bandwidth ultrasonic microphones working with FM sweeps between 20 and 120 kHz. By comparison, the bandwidth used in our acoustic simulation is much narrower at 18 to 22 kHz, and



(a) ECHO-4CH, front **(b)** ECHO-4CH, side **(c)** ECHO-4CH, top

(d) RANDOM, front **(e)** RANDOM, side **(f)** RANDOM, top

Figure A.1: Spatially-varying density of obstacles in the ROIs of both the RANDOM and ECHO-4CH datasets as viewed from different directions. These images were creating by summing the occupancy maps of all examples in each dataset before projecting along the viewing direction.

consequently the spectrograms we generate from our simulated datasets may be expected to contain less information.

Although the ROI of our RANDOM dataset is larger than that of ECHO-4CH, both in terms of physical dimensions at $108 \times 69 \times 69$ cm versus $64 \times 64 \times 64$ cm respectively, and in terms of grid size at $144 \times 112 \times 112$ units versus $64 \times 64 \times 64$, we do not believe this to be a factor in relative model performance. When training the Bat-G Net model on our simulated datasets, we hold the output resolution of Bat-G Net fixed at its original 64^3 grid size, and linearly resample between this and our simulation grid to generate training labels and compute losses, and perform our evaluations. Thus, although our RANDOM dataset is spatially larger, Bat-G Net effectively maintains a fixed output grid size throughout our experiments.

A.3 BatVision Implementation and Training

The authors of BatVision [9] kindly have made their model source code available in a public GitHub repository at https://github.com/SaschaHornauer/Batvision, which we use in our experiments. We cloned the repository at the commit identified

by the hash value `15ba875aadfd1deb39ece3922ecb87b9d8700aa9` and trivially modified the waveform and spectrogram encoders to use 4 input audio channels instead of 2, in order to allow better spatial reasoning and allow for a more fair comparison against our model and Bat-G Net. The loss function we minimize when training both BatVision models is identical to that used by Christensen *et al.*