The Dissertation Committee for Tyler James Wiest Certifies that this is the approved version of the following Dissertation:

Computational Automation for Efficient Design of Acoustic Metamaterials

Committee:

Carolyn C. Seepersad, Supervisor

Michael R. Haberman, Co-Supervisor

Richard Crawford

Kevin Lee

Computational Automation for Efficient Design of Acoustic Metamaterials

by

Tyler James Wiest

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2021

Acknowledgements

This work would not have been possible without the support and guidance of my supervisor, Dr. Carolyn Seepersad. She has had a profound influence on my development as a researcher and provided me all the tools necessary to make contributions to the engineering design automation community. Dr. Seepersad, I will be forever grateful for your help completing this dissertation and my degree. Dr. Michael Haberman has also been immensely influential to me by supporting a novice acoustician in his pursuits into the field. Without his guidance and expertise, I never could have made meaningful contributions on acoustic metamaterials. Thank you Dr. Haberman for taking me into your research group and committing so much time getting me up to speed on acoustics topics that would have been elementary for a fully focused acoustician.

Many others aided my studies and research. Foremost, I would like to thank Applied Research Laboratories (ARL) for supporting me with a graduate research assistant appointment. By working under the umbrella of ARL I had access to powerful computing systems that proved incredibly useful and a support system of experienced researchers that were willing to assist and gracious with their time. Key ARL members include Dr. Kevin Lee, Chris Mericle, and Michael Lee. Thank you Dr. Richard Crawford for teaching my first class on product design and providing an excellent and thorough introduction to the engineering design process. Your instruction inspired my approach to design research and will be invaluable throughout my career as an engineer.

I would also like to thank the colleagues and labmates that conducted research alongside me. Thank you Clint Morris and Conner Sharpe for your willingness to discuss design automation topics in great detail and share in the spirit of developing the capabilities

of the entire research group. Thanks to AJ Lawrence, Jared Allison, David Debeau, Nick Rodriguez, Hongtao Song, Connor Gunsbury, Andrew Allan, Ademola Oridate, Oliver Uitz, and Sofia Valdez, as well as all members of both Dr. Seepersad's and Dr. Haberman's labs that worked with me concurrently. At last, thank you to all my family and friends that have supported me up to this point and been understanding of sacrifices I have chosen in order to pursue a doctorate degree. Thanks Mom and Dad for making me into the person that I am. Thanks Mallory for standing by me through the difficult times and encouraging me to celebrate the good ones during this journey. I could not have done it without you all.

This material is based on work supported by the National Science Foundation under Grant No. EFRI-1641078. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Abstract

Computational Automation for Efficient Design of Acoustic Metamaterials

Tyler James Wiest, Ph.D.

The University of Texas at Austin, 2021

Supervisor: Carolyn C. Seepersad

Co-Supervisor: Michael R. Haberman

Acoustic metamaterials (AMMs) are an exciting technology because they are capable of responding to vibrations in ways that are impossible to achieve with conventional materials. However, realization of AMMs requires engineering design to provide a connection between first-principles research and production of parts that perform as expected. Designing AMMs is a challenging endeavor because evaluating designs is costly and manufacturing metamaterials requires precise techniques with small minimum resolutions. To address these challenges, new computational tools are necessary to aid design. This work proposes three tasks that improve the capabilities of design for AMM while being extensible to other engineering design automation tasks. The first task is to develop a design exploration tool that improves the computational efficiency of identifying sets of high-performing designs in a design space that is sparse and comprises mixed discrete/continuous data. The second task is to develop a process for designers to evaluate manufacturability of difficult-to-manufacture parts and drive co-development of manufacturing methods and AMM. In the final task, a machine learning based method is

developed to efficiently model AMM with heterogeneous arrangements of their microstructures such that strict homogenization is infeasible. The outcomes from completing these tasks will provide a significant and novel improvement over existing methods of designing AMMs.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction to Acoustic Metamaterial (AMM) Design

Acoustic metamaterials (AMMs) are an exciting technology because they are capable of responding to vibrations in ways that are impossible to achieve with conventional materials. However, realization of AMMs requires engineering design to provide a connection between first-principles research and production of parts that perform as expected. Designing AMMs is a challenging endeavor because evaluating designs is costly and manufacturing metamaterials requires precise techniques with small minimum resolutions. To address these challenges, new computational tools are necessary to aid design. This work proposes three tasks that improve the capabilities of design for AMM while being extensible to other engineering design automation tasks. The first task is to develop a design exploration tool that improves the computational efficiency of identifying sets of high-performing designs in a design space that is sparse and comprises mixed discrete/continuous data. The second task is to develop a process for designers to evaluate manufacturability of difficult-to-manufacture parts and drive co-development of manufacturing methods and AMM. In the final task, a machine learning based method is developed to efficiently model AMM with heterogeneous arrangements of their microstructures such that strict homogenization is infeasible. The outcomes from completing these tasks will provide a significant and novel improvement over existing methods of designing AMMs.

1.1 Acoustic Metamaterial Definition

Metamaterials are structures, architected on the micro-scale, that achieve their properties not only from the choice of constituent material but also from the arrangement of that material to effect properties that are not achievable with conventional materials [1].

Because of ongoing advancements in manufacturing such as additive manufacturing (AM), metamaterial designs with previously unrealizable architectures are possible or nearly possible to fabricate [2] [3] [4]. This technology has accelerated metamaterial research for numerous applications and for parts under all types of static and dynamic loads [5]. However, designing and building metamaterials for service in a part is much more complicated than using conventional materials because of the near-infinite, and ever-increasing, potential for customizability. Rather than simple selection, creating a metamaterial requires an application-specific design process to meet performance specifications for the envisioned application. The many combinations of design feature settings available and the limited number of arrangements that exhibit novel, interesting, or useful behavior leads to many design challenges.

One classic example of a metamaterial is the microstructured lattice. Lattices are used in quasi-static structural applications by providing a higher strength-to-weight ratio when designed with respect to the load paths in the object [6]. An example of how a lattice might serve in a bracket is shown in Figure 1.1. For a part like this one under relatively simple static loads, it is evident how that the lattice design is much more complex than a solid bracket. The microstructure is designed to perform its task under the geometric constraints and expected loading on the part. As requirements of the microstructured part become more complex, so does the process of designing it.

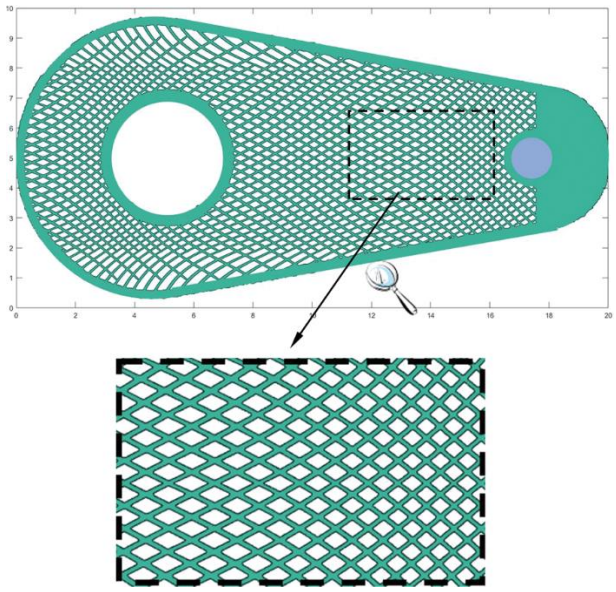

Figure 1.1: Lightweight bracket made with a lattice metamaterial [7].

The study of metamaterials subject to vibrational and acoustic loading has become its own research field that focuses on metamaterials exhibiting acoustic responses not found in nature or conventional materials. Research and development of AMMs that exhibit unique and interesting acoustic performance has shown great potential and a high likelihood of future applicability [8] [9]. Acoustic metamaterials demonstrate interesting response behavior based on the interaction of sub-wavelength features with incident waves. Dynamic interactions and wave responses at the micro-scale of AMMs lead to macro-scale effective properties of the medium such as negative density, ρ, and compressibility, C, which are not present in natural materials. For illustration, Figure 1.2 shows an Ashby chart over the effective ρ-C parameter space of AMMs. Conventional materials are located in the quadrant having both positive density and compressibility. This figure also denotes a few behaviors that occur when an AMM exhibits effective properties that are not present in conventional materials such as negative refraction and enhanced absorption.

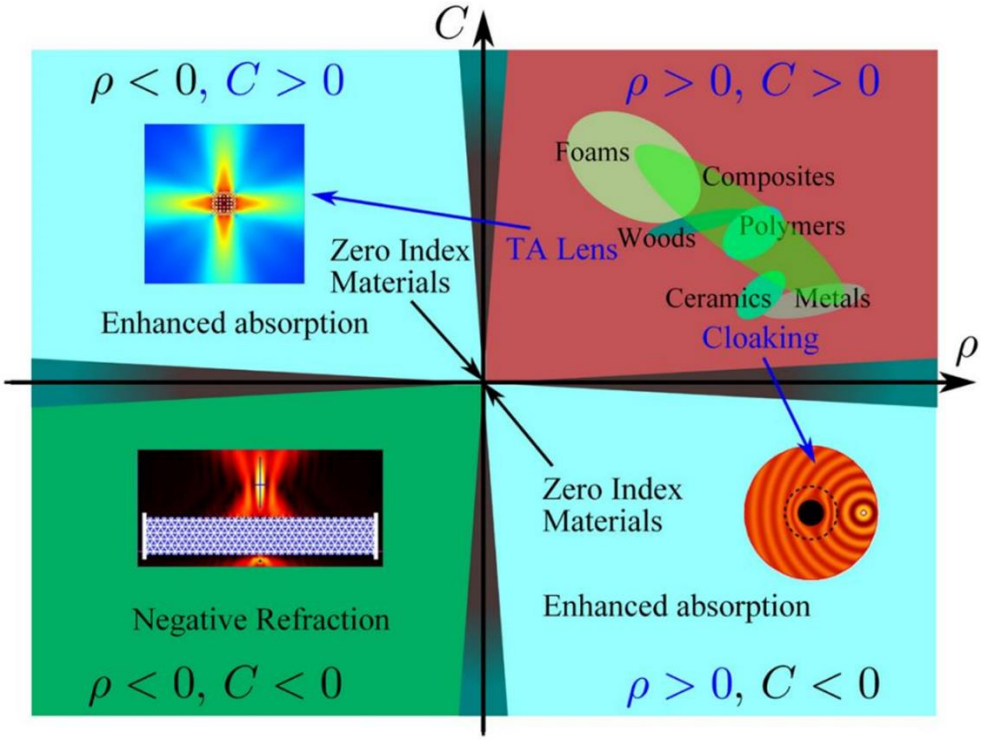

Figure 1.2: Ashby chart over the ρ-C parameter space. Acoustic metamaterials are able to achieve negative effective density and/or compressibility on the macro-scale while conventional materials are restricted to the positive ρ-C space. Some examples of achievable AMM behavior are noted [10].

The variety of AMM geometries and physical relationships that are used to realize these unconventional behaviors are abundant. The following section discusses some of the types of AMMs that show these behaviors and how they can be used.

1.1.1 Types of AMMs and Motivating Applicability

Although the scope of AMM research is very broad, at the highest level these advanced acoustic materials can be separated into two classes: active and passive [11]. Whereas active AMMs derive their unique response from external control systems [12] [13] or modulation [14] to affect sound propagation behavior, the response of passive AMMs relies solely on material composition and geometry of the structure [15] [16] [11].

By designing a passive metamaterial's micro-scale structure properly, macro-scale behavior of the material can vary with respect to incident wave parameters in ways that are unachievable in conventional materials. AMMs designed to exhibit an asymmetric [17] [18] or non-reciprocal [19] [20] response with respect to the direction of propagation are particularly interesting. These responses enable manipulation of acoustic fields such as non-reciprocal transmission [19], wave steering [21], selective absorption of wave energy [22], and cloaking [23]. Directional asymmetry can be achieved by exploiting a few different underlying physical phenomena such as Bragg scattering [24] [25], local resonances [26], and strain-momentum constitutive coupling (Willis coupling) [27].

Controlling sound in application is a common practice and AMMs are simply a tool to achieve sound control in new ways. One of the most common applications of sound controlling structures is a bass trap, which absorbs low-frequency waves in lecture halls and theatres, for example. Metamaterials could manifest themselves in many applications. One possibility is for sensors that allow signals to be sent and received simultaneously. Another is using wave steering to route vibrations around sensitive machine components. Cloaking AMMs have applications in stealth for military operations as well as the capability to reduce environmental noise. Directionally asymmetric absorbers are useful for reducing measurement noise in sensors that create their own signals. This work will focus on design efforts for two types of AMMs: strain-momentum coupled (Willis) metamaterials and directionally asymmetric absorbers. Chapter 2 provides greater detail on these two types of AMMs and the primitive geometries that motivate efficient computational design automation. Both of these AMMs are acoustically reciprocal although they exhibit scattering behavior that is directionally asymmetric. The next section introduces scattering and asymmetry in acoustic media.

1.1.2 Acoustic Wave Scattering, Asymmetry, and Reciprocity

Acoustic waves originate from a vibrating source that displaces the matter around it. Outgoing waves from a source may take any number of shapes such as spherical, cylindrical, or planar. Regardless of the outgoing shape, wavefronts appear increasingly planar as the distance of an observation point increases from the source because the degree of curvature diminishes. Planar waves appear as parallel compressional fronts whose amplitude, A, can be modeled as a sinusoidal function of one variable, x, as shown in Figure 1.3(a). Analysis of acoustic systems in the far-field may treat incident waves as planar with minimal error. Additionally, planar waves provide valuable simplifications of the wave equation. In this work, waves in the far-field are assumed to be planar. With this assumption we may evaluate acoustic excitation and AMM response with wave fields that vary in only one spatial direction.

The plane waves scattered from an AMM can be represented using a scattering matrix form where the scattering matrix, [S], relates incoming (incident) waves to outgoing (scattered) plane waves [28], [29], [30]. When plane waves propagate along the *x*-direction, this system can be presented by a two-port system using the matrix relationship

$$\begin{Bmatrix} p_o^- \\ p_o^+ \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{Bmatrix} p_i^+ \\ p_i^- \end{Bmatrix} \tag{1.1}$$

where p_i^+ and p_i^- are incident pressure waves travelling in the $\pm x$-directions, respectively. Likewise, p_o^+ and p_o^- represent outgoing (scattered) plane waves in the $\pm x$-directions, respectively. Assigning $+x$ as direction 1 and $-x$ as direction 2 the scattering for each incident wave direction may be written as

$$\begin{Bmatrix} p_{r,1} \\ p_{t,1} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{Bmatrix} p_{i,1} \\ 0 \end{Bmatrix} \text{ and } \begin{Bmatrix} p_{t,2} \\ p_{r,2} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{Bmatrix} 0 \\ p_{i,2} \end{Bmatrix},$$

where the incoming waves for cases 1 and 2 have been identified as $p_{i,1}$ and $p_{i,2}$ respectively, and the outgoing waves with subscripts "r" and "t" to denote reflected and transmitted plane waves in agreement with the notation provided in Figure 1.3(b).

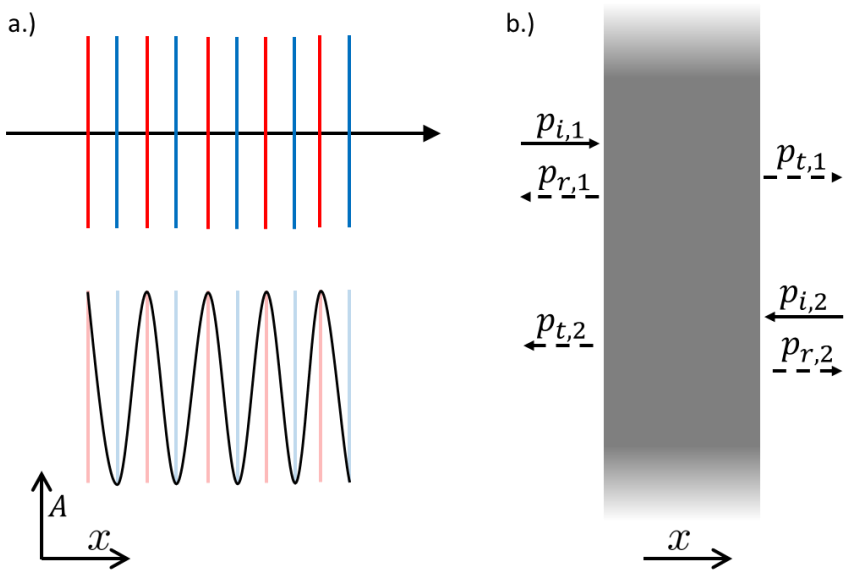

Figure 1.3: a.) Plane waves have parallel wave fronts orthogonal to the direction of propagation and their amplitude in space may be modeled as a sinusoid. b.) Scattering of planar waves is defined by the pressure fields that are reflected and transmitted by a medium. Transmitted and reflected components of a scattered field are dependent on the propagation direction of an incident wave.

Analysis of scattering by an AMM is essential to evaluate its macro-scale response behavior such as coupling between strain and momentum. Additionally, from these scattering definitions it is possible to provide an expression for the absorption coefficient of an acoustic medium with propagation direction case n as defined in Equations 1.2:

$$\alpha_n(x,f) = 1 - |R_n|^2 - |T_n|^2, \tag{1.2a}$$

$$S = \begin{bmatrix} R_1 & T_2 \\ T_1 & R_2 \end{bmatrix}. \tag{1.2b}$$

Note that Equation 1.2(b) identifies the coefficients of the scattering matrix as the reflection and transmission coefficients for each case, which are defined as $R_1 = p_{r,1}/p_{i,1}$, $R_2 = p_{r,2}/p_{i,2}$, $T_1 = p_{t,1}/p_{i,1}$, and $T_2 = p_{t,2}/p_{i,2}$.

Expressing absorption in this form assumes that both incident and scattered wave fields are planar. This assumption is valid in the metamaterial paradigm where characteristic scattering feature sizes a and feature spacing are sub-wavelength, i.e. $a \ll \lambda$ and $d_p \ll \lambda$, and that the scattered field is probed multiple wavelengths from the scattering surface such that evanescent fields are extinguished. Absorption asymmetry can then be characterized with the ratio

$$\bar{\alpha} = \frac{\alpha_2}{\alpha_1} \tag{1.3}$$

which is a measure of the dependence of acoustic absorption with respect to the direction of incidence and serves as the performance metric for directionally asymmetric absorbers explored in this work. While the performance goal is application specific, one obvious goal would be to maximize asymmetry as a function of design features, namely the excitation frequency and the composition of the AMM medium. It is worth noting that asymmetry of absorption is not necessarily indicative of non-reciprocal behavior. Non-reciprocity in acoustics is defined by the breaking of transmission symmetry [31]. A non-reciprocal response requires that $T_2 \neq T_1$.

1.1.3 Manufacturability of AMMs

Fabrication of subwavelength features for metamaterial parts may require manufacturing methods that can resolve very small features with complex geometries. Because wavelength is related to frequency and a medium's sound speed as $\lambda = \frac{v}{f}$, an AMM's specifications are preconditioned by the environment in which it will be used and the characteristics of the waves it will encounter in service. For example, a borderline

ultrasound signal at 20 kHz in air ($v = 343\ ^{m}/_{s}$) has $\lambda = 1.7cm$ and requires features with characteristic sizes, a, smaller than that. At the low end of the MHz range in air, characteristic feature sizes must be in the hundreds of microns. Producing features this small is reasonably achievable for many fabrication methods but controlling the arrangement of features on this size scale with complex geometries is challenging. However, additive manufacturing (AM) methods are uniquely suited for this task [10]. As a result AM and metamaterial technology have developed in tandem in recent years. While fields of research on both AM and AMM are burgeoning, there is a considerable need to connect the capabilities and future needs of each field so that AMMs can harness AM technology and AM may focus progress toward capturing fruitful applications of their processes. Making the connection between these fields is an exercise in designing for specifications. Design of AMMs with the expectation of fabrication via AM presents many challenges and inspires the work of this dissertation.

1.2 Design Approaches and Challenges for AMM

Much of the research published to date on AMMs succeeds in showing specific cases that lead to interesting system responses but stops short of systematically exploring feasible designs and mapping physical behaviors back to system features. One reason is that, except for the simplest of cases, it is costly to evaluate AMM designs using either numerical methods or experiments. Another reason is AMM design spaces commonly have high-dimensionality and complex topologies that make global optimization infeasible. The expense to acquire a sufficient set of performance data about prospective AMM designs presents a significant challenge for designers to effectively explore and characterize the space of design features. As a result, acoustic metamaterial design is often carried out by

heavily constraining the problem and then manually adjusting the design features [32] [33] [26].

The crux of the problem is the difficulty of automating a process that tailors detailed architectures for specific performance criteria under the broad and varied nature of operating conditions in an acoustically excited environment. Because of expense and the very large (often infinite) number of possible designs, automating the process requires deliberate techniques to efficiently explore and exploit the design space. From the engineering design perspective, computational tools are sought that meet the demands of these problems using approaches such as metamodeling, optimization, statistical modeling, and machine learning. Although many tools exist [34], new ones are needed to match the ever-increasing complexity of design opportunities and minimize the cost to arrive at a satisfactory design. For AMM, the need for design tools is currently far from satisfied. Three design challenges of great importance are introduced in the following discussion.

Challenge One

To explore and characterize design spaces in the early stages of design, a sufficient number of feasible designs must be evaluated through experimentation or simulation. For compatibility with design and optimization conventions, design features are represented as dimensions of a design space, and dimensionality is often very large for acoustic metamaterial design problems. In addition to high dimensionality, these problems tend to have a variety of data types such as continuous, discrete, and categorical values for both design features and exogenous variables. When performance criteria are difficult to achieve, designing in high dimensional spaces leads to a very sparsely populated design space as the total hypervolume of the space grows exponentially with respect to the number of dimensions [35]. Consequently, the quantity of design evaluations (simulations or

experiments) required to characterize a design space also grows rapidly. This phenomenon is known as the "curse of dimensionality" [36]. The combination of mixed data types and high dimensionality requires design methods that can mitigate these challenges to efficiently explore the space and yield an accurate mapping of promising designs.

Challenge Two

When designing metamaterials with behavior tailored to specific acoustic loads, the performance of a candidate design can be especially sensitive to manufacturing variability. For passive metamaterials to exhibit directionally asymmetric responses, for example, the characteristics of the expected acoustic load dictate the metamaterial architecture. Depending on the sensitivity of the acoustic response to the details of the underlying geometry, dimensional deviations may lead to a non-functional system. Such is the case with frequency-dependent phenomena like local resonance and Bragg scattering [24] [26] [33]. To reliably and repeatably produce AMM that behave as designed, the accuracy and precision of the manufacturing method must be considered during the design process. AM has enabled the manufacturability of modern metamaterials but also presents challenges in bridging the gap between design and physical demonstration because of the many sources of variability associated with many AM processes. A computationally inexpensive approach is needed to support the repetitive sampling typically associated with uncertainty quantification, specifically with respect to the influence of manufacturing-induced variation on the acoustic performance of candidate metamaterials.

Challenge Three

Statistical and machine learning methods have been used to support exploration of complex design spaces, but the benefits of these methods are limited if they are unable to

efficiently handle the interactions between dynamic loads and spatial features and the influence of those interactions on metamaterial performance. Although conducting design exploration separately for each unique loading condition and class of candidate designs is common, a single method that is valid for a variety of loading conditions and classes of candidate designs would reduce the expense of data acquisition required to carry out a thorough design process. For the case of AMM, which are composed of many micro-scale features distributed in space, design automation is sought for a variable quantity of features that make up a representative system without repeating expensive sampling and characterization efforts each time the number or arrangement of features changes. In practice this characterization would be a metamodel: a cheaply query-able model that represents the functional relationship between inputs and outputs of a more expensive simulation model. By using a metamodel with these capabilities, a metamaterial design space with a greater variety of candidate designs and more spatial complexity (e.g. heterogeneous arrangement of micro-structures, functional grading, etc.) could be explored with manageable levels of computational expense.

Computational design automation such as statistical and machine learning (ML) can be used to address each of these challenges. However, off-the-shelf methods are largely insufficient when designing AMMs so new methods are required to best address the challenges described henceforth. Chapter 3 provides background on ML algorithms and considerations that must be made when implementing them in a design task.

1.3 Research Objectives to Improve Design of AMM

Addressing the three challenges requires completing individual tasks, but they are intertwined and aligned toward the general goal of efficiently automating design of AMMs. The central thesis of this work may be expressed as follows:

The characteristics of AMM production and performance evaluation demand design approaches that are efficient in all stages: exploration of potential designs, exploitation or refinement of promising candidate designs, and quality assurance. Computational design automation may be leveraged, but AMM design presents some unique challenges that motivate development of new methods to meet those challenges. Specification, production, and application of AMMs as effective parts is enabled by developing a method for each stage of the AMM design process.

With this thesis in mind, three research tasks have been identified that improve our capability to design AMMs. These tasks are organized in Table 1.1 and described in the chapters that follow.

Chapter	Task
4	Improve sampling efficiency during design exploration in sparse design spaces with mixed (continuous, discrete, categorical) data types.
5	Evaluate the robustness of metamaterial performance to spatial variability induced by manufacturing processes in a computationally efficient way.
6	Establish an automated, simulation-based method to efficiently design metamaterials with a variable number of repeating features and functionally graded spatial properties.

Table 1.1: Research tasks of this dissertation.

Although motivated by, and focused on, AMM design, the computational tools developed for each task are also extendable to other engineering design domains. The rest of this dissertation is organized to present background on the topics, detail each task, and finally summarize and conclude the work. Chapters 2 and 3 provide background on analysis of AMMs and computational design methods, respectively. Chapters 4 through 6 address the

tasks in Table 1.1. Finally, Chapter 7 provides a discussion of the research contributions and suggests potential for future work.

Chapter 2: Analysis of Acoustic Metamaterials

A significant amount of acoustic analysis makes use of simplifying assumptions such as dimension reduction and material isotropy to minimize the number of mathematical expressions required to define a system and allow the wave equation to be solvable. Analytical solutions are developed to better understand the underlying physics and predict acoustic metamaterial (AMM) behaviors, but this approach is often limited to idealized systems of discrete elements and geometries that are mathematically well-described for continuum systems. Geometrically simple, passive, structural AMMs and acoustic metasurfaces (AMSs) that exhibit asymmetric responses can be described and analyzed in this manner, but opportunities for metamaterial design and optimization are limited. Increasing complexity of geometric and material parameters quickly leads to physical behavior that is impractical, if not impossible, to express with analytical continuum mechanics approaches. As a result, discrete numerical approaches are preferred over analytical approaches to acoustics problems when geometry is irregular or inhomogeneous and physical responses include complicated behaviors such as near-field evanescent effects, viscoelasticity, and multi-modal responses.

By definition, passive AMMs contain geometric features and material interfaces that affect the wavefield both local to those features and beyond the AMM boundaries. In this paradigm, discrete numerical methods are used to maintain mass and energy continuity across interfaces where constitutive parameters make a step-change. A change in constitutive parameters may also require modifications to the underlying physics across interfaces between material domains with differing constitutive behaviors. For AMMs, common material models include fluids, linear elastic solids, and viscoelastic solids. FEA has been developed and demonstrated to be an effective and practical method for analyzing structures with these geometric and material features.

Although FEA can be computationally expensive for design tasks (as addressed in Section 1.2), using it is necessary to gather deterministic analyses to serve as ground-truth data for the design methods that will be developed and presented in Chapters 4-6. This chapter presents the physical behavior and geometry of two types of passive AMM that exhibit Willis coupling and asymmetric absorption in Section 2.1. Section 2.2 then explains how these AMMs are analyzed with FEA.

2.1 Passive AMM for Asymmetric Responses

The successful design of the sub-wavelength structure of an AMM leads to macroscopically observable behavior that is unachievable in conventional materials [15] [16]. Traditionally, exceptional control of acoustic wave propagation has been achieved by exploiting a few different underlying physical phenomena. Bragg scattering can be used to construct phononic crystals [24] [25], local resonances generate negative effective density, stiffness, and phase speed [26], direction-dependent inertia leads to the emergence of anisotropic dynamic density [37] [38], and elastic lattice designs lead to pentamodal effective properties [39] [40]. However, passive AMMs designed to exhibit an asymmetric scattering response [41] [42] [43] [44] with respect to the direction of propagation and active AMMs displaying non-reciprocal behavior [19] [45] are particularly interesting for applications where one desires to tune the direction-dependent acoustical response of a material.

2.1.1 Willis Materials

Direction-dependent strain-momentum coupling is referred to as Willis coupling or acoustic bianisotropy [46] and the metamaterials based on this physical behavior are broadly referred to as Willis materials. Analogous to magnetoelectric coupling in

electromagnetism [47], Willis materials are described with non-classical coupled constitutive relations for the local momentum and pressure-strain behavior, as shown in Equation (2.1), due to the presence of an asymmetric sub-wavelength structure. In the frequency domain, the coupled constitutive relations are:

$$\boldsymbol{\mu} = \underline{\boldsymbol{\rho}} \cdot \boldsymbol{u} - \boldsymbol{\eta} p, \tag{2.1a}$$

$$\epsilon = \boldsymbol{\gamma} \cdot \boldsymbol{u} - \beta p, \tag{2.1b}$$

with coupling vectors

$$\boldsymbol{\eta} = \boldsymbol{\chi}^{o} + i\boldsymbol{\chi}^{e}, \tag{2.2a}$$

$$\boldsymbol{\gamma} = \boldsymbol{\chi}^{o} - i\boldsymbol{\chi}^{e}, \tag{2.2b}$$

where

$\boldsymbol{\mu}$: momentum density

$\boldsymbol{u}$: particle velocity

ϵ : volume strain

p : acoustic pressure

$\underline{\boldsymbol{\rho}}$: anisotropic mass density

β : adiabatic compressibility

$\boldsymbol{\chi}^{o}$: odd coupling

$\boldsymbol{\chi}^{e}$: even coupling.

For reciprocal media the coupling vectors vanish, i.e. $\boldsymbol{\eta} = \boldsymbol{\gamma} = 0$, and the constitutive relations are consistent with the classical relations for momentum (Equation 2.1a) and strain (Equation 2.1b) in an isentropic fluid. Heterogeneities with asymmetric distributions of density and/or stiffness embedded in a fluid background will lead to macroscopically observable Willis coupling due to the fact that local spatially-uniform, but time-varying pressure fields will simultaneously generate volume and translation of the center of volume. This leads to coupling between the monopole and dipole contributions to the total

scattered field. The seminal work by Sieck and colleagues showed that composites of this type can be homogenized to quantify Willis coupling on the macroscopic level [27]. The approach they presented motivates the designer to identify composites that exhibit non-negligible coupling and to seek instances of strong coupling under various conditions.

Many researchers have demonstrated systems that exhibit Willis coupling. Muhlestein *et al.* were the first to design and analyze an effective material element made in 1D using discrete mechanisms [41]. Experimentation on this structure succeeded in showing measurable Willis coupling. Li *et al.* demonstrated wave steering with a metasurface comprised of a 2D array of Helmholtz resonators [43]. Also working with an array of resonators, Quan *et al.* analytically derived the maximum achievable macro-scale bianisotropy for a grate-like structure of resonators with functionally graded geometries [44]. These approaches show viable designs made with fluid filled channels serving as resonators. Willis coupling was also demonstrated in multiple instances where two or more membranes encase a fluid by Ma *et al.* [48] [49]. The preceding examples are based on fluid systems, but analyzing Willis materials with scatterers made of continuous elastic material is another active research area.

By utilizing a uniform fluid cylinder with prescribed coupling vectors Muhlestein *et al.* both analytically described and numerically calculated the scattered fields that indicate moderate Willis coupling [50]. Extension from fluid models to those with solid inclusions is ongoing. Isotropic cantilever beam resonators in a fluid have been shown to exhibit Willis coupling [51], for example. In pursuit of a fully elastic Willis material, Muhlestein and Haberman developed a homogenization method for a random distribution of elastic heterogeneities in an elastic medium [52]. There is a wealth of recent and ongoing research on Willis materials in addition to the literature cited in what follows, but a comprehensive review of this literature is beyond the scope of this work.

As recent literature shows, a bianisotropic acoustic response is demonstrable using several design paradigms. In pursuit of Willis material designs that are manufacturable and deployable into application, scattering inclusions made of solid material provide many advantages. Because heterogeneities are required to be sub-wavelength, they must be extremely small for high frequency waves. Much of the Willis material literature uses a non-dimensional constraint $ka \ll 1$ where a is characteristic feature size and k is wavenumber, $k = 2\pi/\lambda$ to avoid limiting the scope to specific frequencies. The experimental demonstration by Muhlestein *et al.* [41] utilizes frequencies in the 10^3 Hz order of magnitude. Working at low frequencies yields tenable size scales for fabrication of experimental structures such as discrete element systems. At higher frequencies fabrication becomes a challenge as the required size of heterogeneities decreases. Solid, asymmetric inclusions become a more attractive option than fluid channel resonators in this case to improve manufacturability at small scales. While fluid filled inclusions and other resonators such as side-wall cavities and membranes [53] are still promising opportunities, this work focuses on continuous structural inclusions. Continuous inclusions are less challenging to manufacture than fluid filled inclusions. Using AM, multi-material inclusions can be built at small enough scales to create Willis coupling behavior for a broad frequency spectrum. As AM technology improves in resolution and dimensional accuracy, fabrication and design opportunities will continue to expand.

Two different geometric primitives are used in this work to demonstrate coupled Willis and asymmetric material response behavior. Targeted response behavior is demonstrable with both, but because desirable performance is narrow-band in nature and strongly dependent on constitutive material properties, designing them to any performance specification requires powerful design methods. These primitives are a starting point for the design processes that explores the space of features and exploits feature-to-performance

mappings to identify sets of satisfactory designs. Chapter 3 provides background on the design methods adapted for addressing the Research Tasks, and Chapters 4-6 detail how each task was accomplished. The following sections in this chapter detail the analysis methods for both types of primitive geometries using FEA.

2.2 MULTI-MATERIAL LAYERED WILLIS MATERIALS

2.2.1 Pseudo-1D Willis Material Model

The simplest expression of Willis material uses inclusions made with semi-infinite layers of different materials. Planar layers that are normal to the direction of wave propagation eliminate boundary effects and transverse scattering that would exist in geometries with higher dimensionality. In applications, a Willis material must be fabricated in 3D with boundaries on each side but to simplify the analysis we can consider a 1D composite model ensonified with plane waves. Figure 2.1 shows the configuration of periodic unit cells under plane wave excitation in the +*x*-direction. The configuration of potential unit cells is limited to one primitive design consisting of a two-layer inhomogeneity in a background of liquid water. The composition of the layers constituting the inhomogeneity is limited to common materials with well-understood bulk material properties. In particular, density, ρ, sound speed, c, and compressibility, β, in the layers and the background water are the important factors affecting wave propagation through the unit cell. By convention the properties of the background fluid are labeled ρ_0, c_0, and β_0. Each inhomogeneity includes two layers, and each layer is assumed to have identical thickness because thickness has less effect on the wave propagation than the existence of boundaries between materials. For a unit cell of length L and a heterogeneity of length l, the volume fraction is $VF = \frac{l}{L}$.

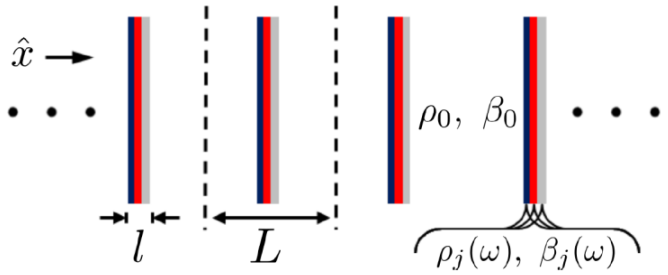

Figure 2.1: Plane wave propagation along the x-axis in a 1D periodic medium composed of repeating multi-layer heterogeneities of length l in unit cells of length L with a background of liquid water with properties ρ_0, c_0, and β_0. Adapted from [27] with permission.

Under the assumption that Willis coupling vectors are zero, this system may be analyzed as a 1D scattering system as described in Equations 1.1-1.3 (scattering, reflection, transmission). However, when using the coupled constitutive equations (Equations 2.1 and 2.2) this system exhibits non-zero coupling. It is possible to evaluate the Willis coupling terms $\chi^e$and χ^o of the unit cell response. Plane waves are incident on the unit cell propagating from left-to-right and then right-to-left, and the scattered pressure field is calculated on both sides of the unit cell. Since the homogenization procedure is dynamic, the frequency of the plane waves affects the effective material behavior and therefore must be considered. For the homogenization, frequency is incorporated as an element of the wavenumber in the background material, k_0, and normalized to the unit cell length. The non-dimensional wavenumber takes the form

$$k_0 L = \frac{2\pi f L}{c_0} \tag{2.3}$$

in the background material and

$$kL = \frac{2\pi f L}{c} \tag{2.4}$$

in the inclusion layer. To quantify the Willis coupling, we use the same convention to normalize the coupling terms as Sieck *et al.* [27], namely $c_o\chi^e/(k_0 L)$ and $c_o\chi^o/(k_0 L k L)$.

The normalized terms $c_o\chi^e/(k_0L)$ and $c_o\chi^o/(k_0LkL)$ are comparable for different unit cell designs. In addition to the coupling, the impedance ratio between the composite and water, $\mathrm{Re}(Z_{\mathrm{eff}})/Z_0$, is measured, along with the effective normalized wavenumber of the composite, kL. This analysis is carried out entirely using non-dimensionalized parameters so performance is measured as homogenized effective normalized couplings, $c_o\chi^{\mathrm{e}}_{\mathrm{eff}}/(k_0L)$ and $c_o\chi^{\mathrm{o}}_{\mathrm{eff}}/(k_0LkL)$, as functions of the normalized wavenumber in the background fluid, k_0L.

2.2.2 Finite Element Analysis of 1D Willis Material

To measure bianisotropic responses of this system an FEA model was built using the commercial software package Comsol Multiphysics to simulate the Willis coupled response of individual unit cells. The physics are applied to the geometry using Comsol's *Pressure Acoustics, Frequency Domain* module [54] which solves the Helmholtz equation,

$$\nabla^2 p + \frac{\omega^2}{c^2} p = 0, \tag{2.5}$$

for the time-harmonic pressure field in an isentropic fluid using the $e^{-i\omega t}$ time convention,

$$p(\boldsymbol{x}, t) = p(\boldsymbol{x}) e^{-i\omega t}. \tag{2.6}$$

Geometry and Materials

Unit cell geometry used for simulation is shown in Figure 2.2. Model geometry is built in 3D with cuboids of material layered along the *x*-direction such that *y*- and *z*- spatial dimensions are equal for all layers. The inclusion is centered at $x = 0$ such that the interface between the two inclusion materials is on the *y*-*z* plane at $x = 0$. The background fluid serves as a waveguide on either side of the inclusion with probe planes at $x = \pm L/2$. Beyond the probe planes at either end along the x-direction, *Plane Wave Radiation* conditions are placed at $x = \pm L$. This condition models an open port for plane waves and

allows for no reflections when the incident plane wave is normal to the boundary. Neumann (sound-hard or rigid) conditions are applied on all boundary planes defined by unit normals in the *y*- and *z*-directions, $\boldsymbol{n}_{y,i} = (0,1,0)$ and $\boldsymbol{n}_{z,i} = (0,0,1)$ respectively. A sound-hard condition prescribes zero acceleration and velocity along the direction normal to it,

$$-\boldsymbol{n} \cdot \left(-\frac{1}{\rho_c}\nabla p_{tot}\right) = 0, \tag{2.5}$$

such that impedance goes to infinity $Z \to \infty$. In Equation 2.5 ρ_c is constant fluid density, p_{tot} is total pressure, and $\boldsymbol{n}$ represents the normal vector. Because only plane waves exist in the domain, the wavefield has no component normal to the sound-hard boundaries and therefore no reflection occurs from those boundaries. To excite the model, incident waves are applied as a background field in all domains except the inclusion layers. This background field is steady-state in the time-harmonic frequency domain.

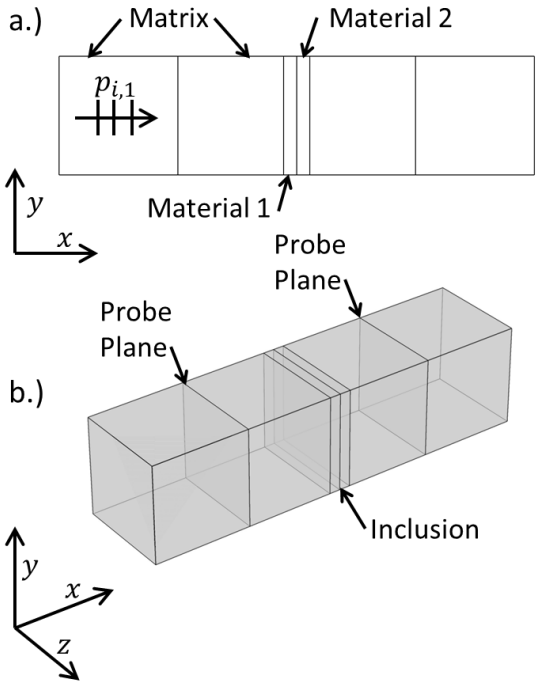

Figure 2.2: FEA model geometry of a scattering two-layer inclusion in background fluid. (a) A cross-section of the geometry is shown with constitutive material domains labeled and the background plane wave, $p_{i,1}$, indicating the direction of propagation in the $+x$-direction. (b) Isometric view showing the location of the inclusion and probe planes where measurements of p and v are taken and averaged across the surface.

Material definitions are simplified by the 1D, planar nature of the analysis. Solid materials generally experience elastic waves that sustain shear forces, but under 1D planar radiation conditions the shear wave component approaches zero as long as each material is isotropic. While the individual constitutive materials are isotropic, the homogenized material remains anisotropic as expressed in Equation 2.1. As a result, the expression for longitudinal elastic waves reduces to match the expression of pressure waves in a fluid (Equation 2.5). This treatment reduces computational expense by reducing the complexity of calculating equilibrium of continuity across multi-physics boundaries. Without the existence of shear, the pertinent parameters of the solid inclusions are density, ρ, sound speed, c, and compressibility, β. Material selections and their associated parameters are discussed further as part of the design exploration methodology in Chapter 4.

Meshing

Meshing the geometry for the two-layer Willis material simulation is relatively simple because both the waveguide and inclusion domains are regular cuboids. The background fluid domain is meshed separately from the inclusions, and the meshing function in Comsol automatically ensures node connectivity across domain boundaries. The mesh is generated from tetrahedrons using a Delaunay tessellation [55] and element size is based on bounds set by the user. Characteristic element sizes, l, are bounded as $300\ \mu m < l < 7mm$ with a maximum element growth rate of 1.35 and curvature factor of 0.3. Figure 2.3 shows the resolved mesh for the base geometry. The extruded mesh scales

with length of the matrix, so the element count is nearly identical for all simulations at approximately 6.8×10^5 elements with approximately 9.8×10^5 DOF.

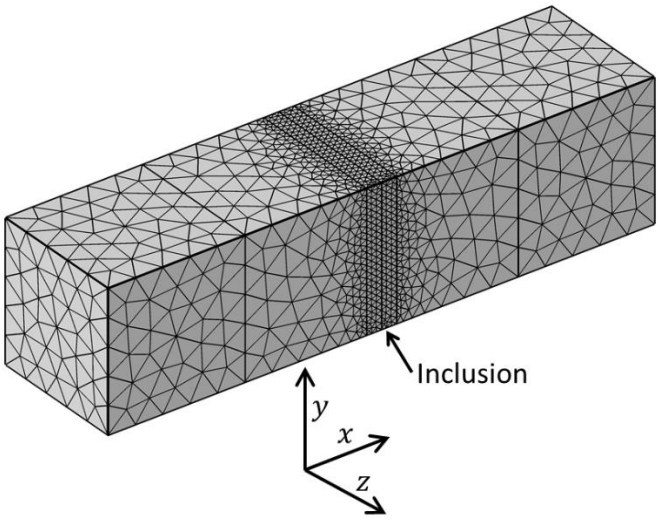

Figure 2.3: The meshing of the two-layer Willis material FEA model. A finer mesh resolution is used for the inclusion layers due to their thinness relative to the domain. In the background fluid the mesh resolution increases with proximity to the inclusion layers.

Study and Post-processing

The model is studied by solving Equation 2.5 for all elements in the domain and ensuring continuity from element to element with boundary and background conditions compiled. PARDISO [54] [56] is the linear numerical solver used to evaluate the system. Outputs are collected by averaging element values across the probe planes shown in Figure 2.2. For this study total pressure, $p_{\mathrm{tot},j}$, and the x-component of velocity, $v_{x,j}$, (both complex-valued) are the outputs gathered where index j indicates the number of the probe.

Simulation output results are post-processed to perform dynamic homogenization and yield effective macroscopic parameters of the composite. The full homogenization derivation and procedure and is too extensive to describe here. Interested readers may refer

to Sieck, Alu, and Haberman [27] for a detailed description. For the purposes of material design, the homogenized coupling parameters, $c_o\chi^{\mathrm{e}}_{\mathrm{eff}}/(k_0L)$ and $c_o\chi^{\mathrm{o}}_{\mathrm{eff}}/(k_0LkL)$, are most important because they indicate the coupling performance of the metamaterial. The post-processing that calculates these terms from $p_{\mathrm{tot},1}$, $p_{\mathrm{tot},2}$, $v_{x,1}$ and $v_{x,2}$ is described in the following paragraph.

For a unit cell geometry described above the scattering matrix (Equation 1.XX) can be expressed to consider the ingoing and outgoing waves at the $x = \pm l/2$ boundaries of the heterogeneity as

$$\begin{bmatrix} p_1^{out} \\ p_2^{out} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} p_1^{in} \\ p_2^{in} \end{bmatrix}, \tag{2.6}$$

$$p_1^{out} = S_{11}p_1^{in} + S_{12}p_2^{in} = p_2^{in}e^{ik_0l} + p_s(-l/2), \tag{2.7a}$$

$$p_2^{out} = S_{21}p_1^{in} + S_{12}p_2^{in} = p_1^{in}e^{ik_0l} + p_s(-l/2), \tag{2.7b}$$

Where p_s is scattered pressure. $\boldsymbol{u}_s \in \mathbb{R}^3$ is determined analogously. The fields p_{loc} and $\boldsymbol{v}_{loc}$ as they would exist in a heterogeneous medium are necessary for later calculation and expressed as

$$p_{loc} = p_1^{in}e^{ik_0l/2} + p_2^{in}e^{ik_0l/2}, \tag{2.8a}$$

$$\boldsymbol{v}_{loc} = \frac{\hat{x}}{Z_0}(p_1^{in}e^{ik_0l/2} - p_2^{in}e^{ik_0l/2}). \tag{2.8b}$$

Equations 2.7 and 2.8 are used in the following Equations 2.9 and 2.10 to determine the monopole polarizability, α_m, dipole polarizability, $\underline{\boldsymbol{\alpha}}_d$, and coupled polarizability, $\boldsymbol{\alpha}_c$:

$$\frac{\boldsymbol{d}_0}{\rho_0} = \underline{\boldsymbol{\alpha}}_d \cdot \boldsymbol{u}_{loc} - i\boldsymbol{\alpha}_c \frac{1}{Z_0} p_{loc}, \tag{2.9a}$$

$$\frac{m_0}{\beta_0} = -i\boldsymbol{\alpha}_c \cdot Z_0\boldsymbol{u}_{loc} - \alpha_m p_{loc}, \tag{2.9b}$$

$$i\frac{k_0}{A}e^{ik_0l/2}\frac{\boldsymbol{d}_0}{\rho_0} = \boldsymbol{v}_s(l/2) + \boldsymbol{v}_s(-l/2), \tag{2.10a}$$

$$-i\frac{k_0}{A}e^{ik_0l/2}\frac{m_0}{\beta_0} = p_s(l/2) + p_s(-l/2). \tag{2.10b}$$

The polarizabilities are then inverted to form complementary polarizabilities, $\tilde{\alpha}_m$, $\underline{\tilde{\boldsymbol{\alpha}}}_d$, and $\tilde{\boldsymbol{\alpha}}_c$:

$$\tilde{\alpha}_m = \alpha_m^{-1}/\Delta_\alpha, \tag{2.11a}$$

$$\underline{\tilde{\boldsymbol{\alpha}}}_d = \underline{\boldsymbol{\alpha}}_d^{-1}\left[\underline{\boldsymbol{I}} + \boldsymbol{\alpha}_c \otimes \tilde{\boldsymbol{\alpha}}_c\right], \tag{2.11b}$$

$$\tilde{\boldsymbol{\alpha}}_c = \boldsymbol{\alpha}_c \cdot \left(\alpha_m \underline{\boldsymbol{\alpha}}_d\right)^{-1}/\Delta_\alpha, \tag{2.11c}$$

with

$$\Delta_\alpha = 1 - \boldsymbol{\alpha}_c \cdot \left(\alpha_m \underline{\boldsymbol{\alpha}}_d\right)^{-1} \cdot \boldsymbol{\alpha}_c. \tag{2.11d}$$

Equations 2.6-2.11 develop all the terms necessary to calculate the effective coupling terms. Expression of the other effective parameters is excluded. As part of expressing the effective fields in terms of interaction coefficients and complementary polarizabilities, monopole fields, Λ_m, dipole fields, $\underline{\boldsymbol{\Lambda}}_d$, even coupled fields, $\boldsymbol{\Lambda}_c^o$, and odd coupled fields, $\boldsymbol{\Lambda}_c^e$ are calculated by way of the following equations:

$$\tilde{\Lambda}_m = V\tilde{\alpha}_m - \left[\frac{k_0 L \sin(k_0 L)}{2(\cos(k_0 L) - \cos(kL))}\right] - i\frac{k_0 L}{2} + \left[\frac{k_0 L}{kL}\right]^2 - (k_0 L)^2, \tag{2.12a}$$

$$\underline{\tilde{\boldsymbol{\Lambda}}}_d = V\underline{\tilde{\boldsymbol{\alpha}}}_d - \left[\frac{k_0 L \sin(k_0 L)}{2(\cos(k_0 L) - \cos(kL))}\right] - i\frac{k_0 L}{2} + \left[\frac{k_0 L}{kL}\right]^2 - (k_0 L)^2, \tag{2.12b}$$

$$\tilde{\boldsymbol{\Lambda}}_c^o = -\left[\frac{k_0 L \sin(k_0 L)}{2(\cos(k_0 L) - \cos(kL))}\right] + \frac{k_0 L k L}{(kL)^2} - (k_0 L)^2, \tag{2.12c}$$

$$\tilde{\boldsymbol{\Lambda}}_c^e = V\tilde{\boldsymbol{\alpha}}_c, \tag{2.12d}$$

and

$$\Lambda_m = \tilde{\Lambda}_m^{-1}/\Delta_{\tilde{\Lambda}}, \tag{2.13a}$$

$$\underline{\boldsymbol{\Lambda}}_d = \underline{\tilde{\boldsymbol{\Lambda}}}_d^{-1}\left[\underline{\boldsymbol{I}} + \left(\tilde{\boldsymbol{\Lambda}}_c^o + i\tilde{\boldsymbol{\Lambda}}_c^e\right) \otimes \left(\boldsymbol{\Lambda}_c^o - \boldsymbol{i}\boldsymbol{\Lambda}_c^e\right)\right], \tag{2.13b}$$

$$\boldsymbol{\Lambda}_c^o = \tilde{\boldsymbol{\Lambda}}_c^o \cdot \left(\tilde{\Lambda}_m \underline{\tilde{\boldsymbol{\Lambda}}}_d\right)^{-1}/\Delta_{\tilde{\Lambda}}, \tag{2.13c}$$

$$\boldsymbol{\Lambda}_c^e = \tilde{\boldsymbol{\Lambda}}_c^e \cdot \left(\tilde{\Lambda}_m \underline{\tilde{\boldsymbol{\Lambda}}}_d\right)^{-1}/\Delta_{\tilde{\Lambda}}, \tag{2.13d}$$

with

$$\Delta_{\tilde{\Lambda}} = 1 - \left(\tilde{\boldsymbol{\Lambda}}_c^o - i\tilde{\boldsymbol{\Lambda}}_c^e\right) \cdot \left(\tilde{\Lambda}_m \underline{\tilde{\boldsymbol{\Lambda}}}_d\right)^{-1} \cdot \left(\tilde{\boldsymbol{\Lambda}}_c^o + i\tilde{\boldsymbol{\Lambda}}_c^e\right). \tag{2.13e}$$

Finally, effective coupling can be defined as

$$c_0 \boldsymbol{\chi}_{eff}^0 = \boldsymbol{\Lambda}_c^o, \tag{2.14a}$$

$$c_0 \boldsymbol{\chi}_{eff}^e = \boldsymbol{\Lambda}_c^e. \tag{2.14b}$$

By calculating the macroscale effective coupling terms of the composite, strength of coupling is measured and used as the performance metric in material design. As discussed in [27], $\boldsymbol{\chi}^{0}_{eff} \neq 0$ if the heterogeneity is symmetric in geometry and material properties, so $\boldsymbol{\chi}^{e}_{eff}$ is the preferred performance metric for showing coupling strength.

Scattered wave fields are returned by the study and can be plotted in 3D throughout the domain. An example of the scattered field plot at a single frequency is shown for each direction of incident wave propagation in Figure 2.4. The background pressure wave is 1 Pa for both directions, $p^{in} = 1Pa$. Figure 2.4a represents the case where $p_1^{in} = 1Pa$ with $p_2^{in} = 0Pa$, and Figure 2.4b represents the case where $p_1^{in} = 0Pa$ with $p_2^{in} = 1Pa$.

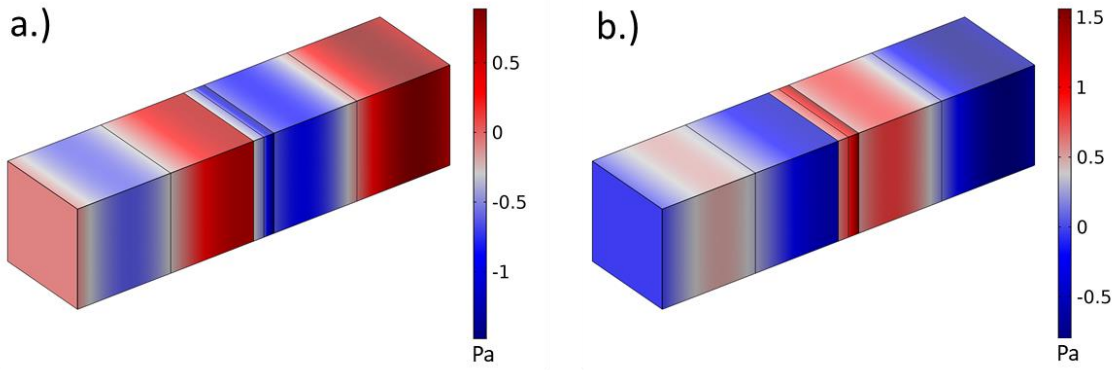

Figure 2.4: Scattered pressure field results when the background wave is incident in the (a) $+x$-direction, $p_{i,1}$, and in the (b) $-x$-direction, $p_{i,2}$.

Scattered fields are post-processed to calculate the even and odd coupling terms. This analysis flow is utilized to evaluate and design two-layer Willis materials over a broad frequency spectrum and for many combinations of materials and geometric features. Specific materials and geometric features explored and how this analysis is used for design is covered in Chapter 4.

2.3 MULTI-MATERIAL ELLIPSOIDAL ASYMMETRIC INCLUSIONS

2.3.1 Ellipsoidal Asymmetric Inclusion Model

More geometrically complex than layered models, AMMs of arranged solid inclusions have been shown to demonstrate Willis coupling behavior by Muhlestein and Haberman [52]. While the authors homogenized a structure with a random arrangement of elastic inclusions, positioning all inclusions along a single plane provides a structure that is more easily manufacturable than random arrangements. The planar layer of inclusions exists in 3D but this structure behaves as an AMS by definition because its thickness is significantly subwavelength [57]. Figure 2.5 illustrates the primitive design considered in this work. The AMS consists of identically oriented ellipsoidal inclusions embedded in a lossy elastomeric matrix. The inclusions are assumed to be fabricated using a mixture of two materials, one being a lossless, isotropic elastic solid, such as a metal, and the second a lossy isotropic material, such as a viscoelastic polymer. These heterogeneous inclusions are embedded in a background material (the matrix) that has good characteristic acoustic impedance match to the background fluid in which it operates, which is water in this work. The inclusions are assumed to have identical orientation with respect to the global coordinate system and are arranged with square periodicity in the *y*-*z* plane such that the AMS is orthogonal to incident waves propagating parallel to the *x*-axis. Figure 2.5 shows a general representation of heterogenous inclusions where the distribution of the two materials is graded in the *x*-direction.

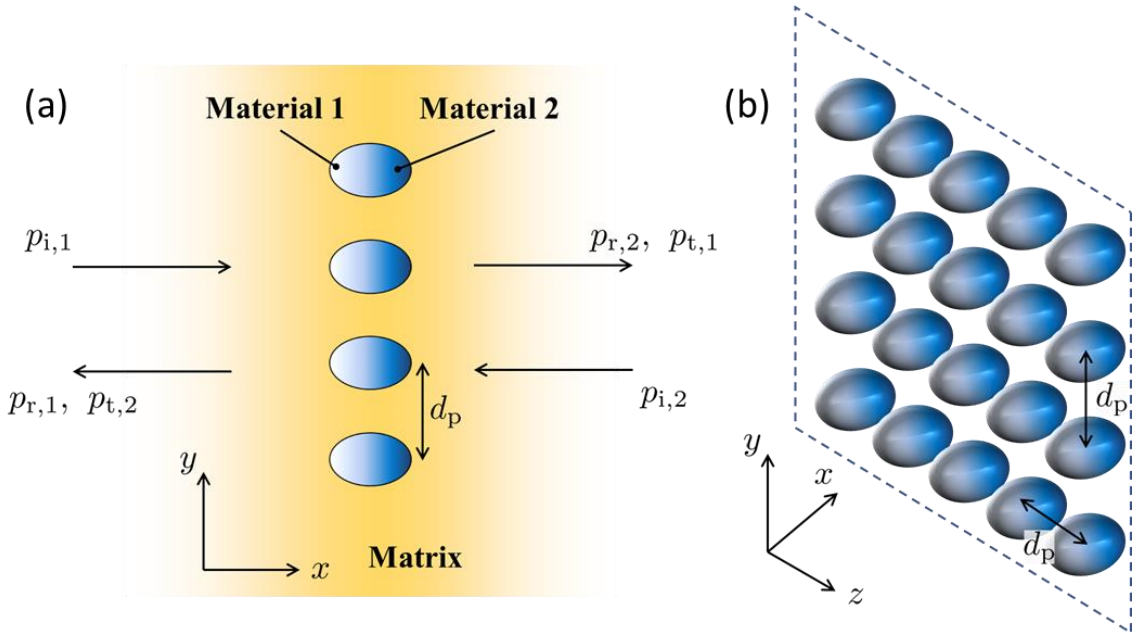

Figure 2.5: Representative geometry of the AMS consisting of distribution of inclusions fabricated from two materials and embedded in a lossy matrix material to generate asymmetric acoustic absorption. The inclusions have non-uniform distribution of material properties and identical orientation in space and are arranged with square periodicity in the *y*-*z* plane with inter-inclusion spacing $d_p \ll \boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is the wavelength of propagating plane waves in the background material. (a) Two-dimensional cross-section in the *x*-*y* plane showing incident, reflected, and transmitted waves for cases of waves propagating in the $\pm x$-direction. (b) Three-dimensional representation of the square lattice of scatterers aligned with the $\boldsymbol{x} = \mathbf{0}$ plane.

The design of an AMS displaying strongly asymmetric absorption is achieved by analyzing the scattering response of heterogeneous inclusions in a matrix material and measuring the back- and forward-scattered fields. The degree of asymmetry in absorption is determined by performing two numerical experiments, one with excitation via equal amplitude plane waves traveling in the positive *x*-direction and the second when the excitation is from the negative *x*-direction. The backscattered fields from both cases are compared for a wide range of frequencies of interest. The numerical experiments are denoted as case 1 and 2 for incident wave propagation directions along unit vectors $\boldsymbol{n}_{x,1} =$

$(1,0,0)$ and $\boldsymbol{n}_{x,2} = (-1,0,0)$, respectively. The scattered pressure field is then probed on planes parallel to the AMS on both sides of the AMS for cases 1 and 2 to determine absorption from the scattered field as expressed in Equations 1.1-1.3 (scattering and absorption) and summarized in the following paragraph [58] [28] [29] [59]. The model is constructed so that the geometry of the primitive can be varied in ways typical of AM fabrication error to support the robust design methodology described and demonstrated in Chapter 5.

The arrangement of dissimilar materials in the inclusions that compose the AMS are essential to generating asymmetric acoustic absorption from the Willis scattering primitive design, even when the inclusions of the AMS are deeply subwavelength [41] [28] [29] [59]. By arranging the inclusions as shown in Figure 2.5, the input impedance of the AMS is very different for incident waves from either direction [41]. When one inclusion material is assumed to be lossy, acoustic absorption is observed to be strongly dependent on the direction of incidence [28] [29] [59]. Here we define the material loss of the inclusion materials by assuming that they are constructed from materials displaying a complex plane wave modulus, $M = M'(1 - i\eta_s)$, where M' is the storage modulus, η_s is the isotropic viscoelastic loss factor, and we have assumed the $e^{-i\omega t}$ time convention. It is important to note that asymmetry in reflection and absorption does not violate reciprocity and thus transmission is symmetric, i.e. $T_1 = T_2$. The asymmetry in reflection is therefore solely responsible for asymmetric absorption as one would anticipate from previous work [45] [28] [29]. For this reason, asymmetry of reflection $R_2/R_1$is also a valid performance metric.

Some initial performance and design constraints are specified *a priori* to eliminate superfluous variables. Analysis is performed on a unit cell consisting of ellipsoidal inclusions inside a matrix material with a cross sectional length and width of 1.2 mm. The

cross-sectional dimensions of the unit cell have been chosen to be much smaller than acoustic wavelengths in the matrix material at the frequencies of interest (< 160 kHz) in order to avoid Bragg scattering. The characteristic size of inclusions is necessarily constrained to be sub-wavelength, $ka \ll 1$, to ensure that the scattered field is planar in the far-field and that we remain in the long-wavelength scattering limit. Exploration of the asymmetry with respect to ka is carried out by holding a constant with $a = d = 1$ mm and varying frequency in the range of 1 to 160 kHz. For satisfactory performance from a design perspective, the asymmetric absorber must exhibit at least a 6 dB difference of reflection with respect to the direction of incident wave propagation: $20\log_{10}(|R_2/R_1|) \leq -6$ dB.

2.3.2 Finite Element Analysis of Ellipsoidal Asymmetric Inclusions

Similar to the approach taken in Section 2.2.2, a FEA model was built using the commercial software package Comsol Multiphysics to simulate the asymmetric response of individual unit cells and supercells made of multiple unit cells arrayed in the *y*-*z* plane. The physics are applied to the geometry using Comsol's *Pressure Acoustics, Frequency Domain* module for fluid domains (Equations 2.5 and 2.6) and the *Solid Mechanics* module for solid, elastic domains. For frequency domain studies, the *Solid Mechanics* module solves the equations

$$Re(\boldsymbol{u}e^{i\omega t})\rho\omega^2 = \nabla \cdot S + \boldsymbol{F}_v e^{i\phi}, \tag{2.15a}$$

$$-ik = \lambda, \tag{2.15b}$$

where $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{F}_v$is a body force, S is the second Piola-Kirchhoff stress tensor.

Coupling between the pressure acoustics in the fluid matrix and solid mechanics of the inclusion is implemented with the *Acoustic-Structure Boundary* condition in Comsol.

This boundary condition couples the load on the structure to the acceleration experienced by the fluid matrix. At the boundaries the equations

$$-\boldsymbol{n} \cdot \left(-\frac{1}{\rho_c} \nabla p_{tot}\right) = -\boldsymbol{n} \cdot \boldsymbol{u}_{tt}, \tag{2.16a}$$

$$\boldsymbol{F}_A = p_t \boldsymbol{n}, \tag{2.16b}$$

describe the coupling where $\boldsymbol{F}_A$ is force per unit area, and $\boldsymbol{u}_{tt}$ is acceleration.

Geometry and Materials

The geometry for the model consists of a matrix of background material with a semi-infinite layer of ellipsoidal inclusions having radii $r_\mathrm{a}, r_\mathrm{b}$, and r_c aligned with the *x*-, *y*-, and *z*-axes, respectively, and centered at $(x_a, x_b, x_c) = (0, 0, 0)$ midway through the matrix domain with respect to the direction of incident wave propagation. Inclusions are arranged on the *y*-z plane in a square-packed lattice with center-to-center side-lengths of d_p as shown in Figure 2.5. Each inclusion includes two materials separated by—and bonded along—an interface defined by a y-z plane that intersects the ellipsoid at a distance h from its center (Figure 2.6).

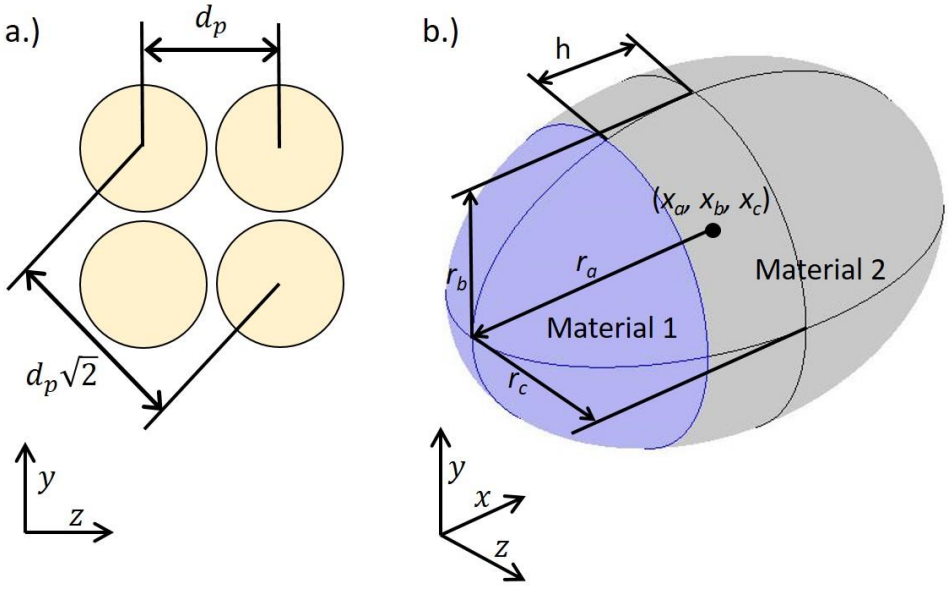

Figure 2.6: (a) A square-packed lattice of four inclusions and their arrangement shown in plane with the scattering layer. (b) Dimensions and material placements overlain on an example of ellipsoidal inclusion geometry where the center of the inclusion has a position defined by the triplet (x_a, x_b, x_c). These dimensions are subject to variation from imperfect manufacturing.

To model a layer of many inclusions efficiently, a single unit cell as shown in Figure 2.7 is treated as infinitely periodic in the $x = 0$ plane. Floquet periodic boundary conditions [60] are used along both the y- and z- directions. Elimination of internal reflections in the x- direction is performed with perfectly matched layers (PMLs) at either end of the matrix. The PMLs are domains that emulate the Sommerfeld radiation condition by progressively damping energy over a finite distance [54] [61] [62]. With these boundary and domain conditions, the simulated scattering behavior is the result of a square lattice of inhomogeneous inclusions, which constitutes the asymmetric absorbing AMS.

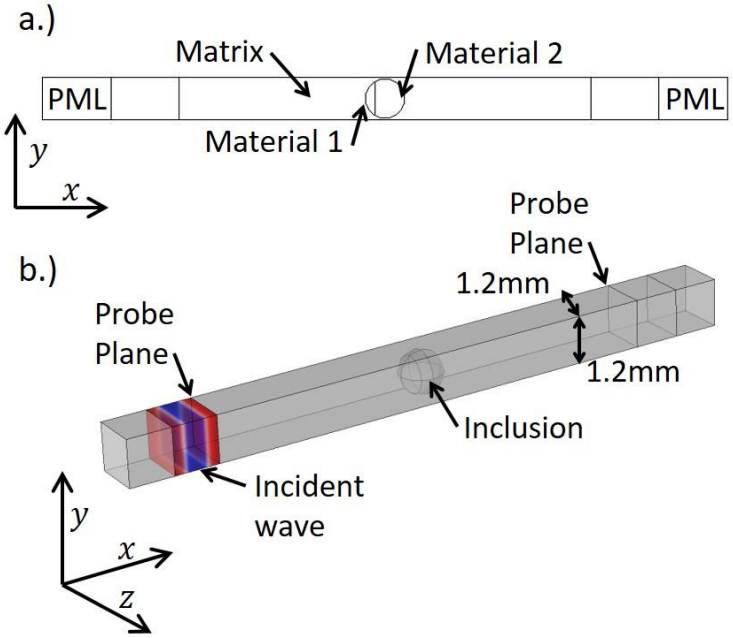

Figure 2.7: Example simulation geometry of a single cell of the periodic asymmetric absorber metamaterial. An inclusion composed of two separate materials is encased in a matrix. (a) Cross-section of the unit cell geometry indicating the constitutive material placement and perfectly matched layer (PML) conditions at both ends of the domain along the x-direction. (b) Isometric view of a single simulation with the incident plane wave, $p_{i,1}$, propagating along the $+x$-direction. The incident wave field is shown only on the left of the probe plane so that internal geometry is visible. A simulation solves for the scattered fields at the probe planes positioned at each end. The distal sections are PMLs to eliminate reflections and domain boundaries. Unlabeled domains represent the matrix material.

The incident wave is a planar background field that originates between the PML and the inclusions and propagates toward the inclusions with the matrix serving as a waveguide. The scattered field is measured as a steady state single-frequency result in each separate FE solution on both sides of the inclusion layer plane. Measurements are taken sufficiently distant from the inclusion layer to ensure that all evanescent fields have decayed such that the total scattered field is planar. Because the matrix is lossless, there is no loss of information by extending the distance between the inclusions and probe plane but for these analyses this distance is set at 3λ from the $x = 0$ plane. Field outputs are averaged across each probe plane to account for numerical noise and then post-processed to yield R and T for each simulation.

A physical implementation of this AMS would take the form of a layer of matrix material containing aligned heterogeneous inclusions embedded in another material or fluid. Scattering from external boundaries of the matrix is application specific and outside the scope of this work. However, the silicone-based polymer polydimethylsiloxane (PDMS) is the matrix material for its excellent impedance match to water in this case. The range of realizable impedance for unaltered PDMS is dependent on manufacturing factors but falls between 1-1.1 MRayl [63] [64]. This leads to reflection coefficients of approximately 0.03 at a PDMS-water boundary. Additives such as ZnO and TiO_2, can be

used to adjust the density, and therefore impedance, of PDMS while minimally affecting the loss such that the water-PDMS interfaces reflect almost no energy [63] [64]. The PDMS is treated as a lossless fluid to investigate the absorptive performance of the inclusion layer alone. Shear deformation in the PDMS can be ignored at these frequencies since shear waves attenuate very rapidly at the frequencies of interest (over 100 kHz) [65] [66].

Inclusion materials are elastic solids with the viscoelastic component dissipating energy by way of a non-zero isotropic structural loss parameter $\eta_s \neq 0$. Both inclusion materials are defined by their density, ρ, Young's modulus, E, and Poisson's ratio, ν. Specific material selections and their corresponding constitutive parameters are discussed further as part of the design methodology in Chapter 5.

Meshing

Simulating these structures with a sub-wavelength inclusion layer and sufficient waveguide requires a domain that is multiple orders of magnitude larger than the smallest feature. Consequently, it is both unnecessary and computationally infeasible for the mesh to be made of uniformly sized elements throughout. Rather, mesh density must be higher in and around the inclusion to capture near-field effects. Mesh density is lower where the scattered field is approximately planar with wavelengths of the order of the incident wave. This mesh refinement is implemented by densely meshing the near-field region of the domain within $\pm 2.5 r_{\mathrm{a}}$ and then extruding the mesh to the extents of the remaining domain with a much coarser resolution in the x-direction. The near-field region is meshed with quadrilaterals on the boundaries normal to *y*- and *z*- directions and tetrahedrons throughout the domain inside the boundaries. Mesh extrusion from the near-field region to the end of the waveguide domain uses the cross-section on the boundaries of the near-field region normal to the *x*-direction. Characteristic element sizes, l, within the near-field domain are

bounded as $\frac{\lambda}{1000} < l < \frac{\lambda}{10}$ with a maximum element growth rate of 1.3 and curvature factor of 0.2. Figure 2.8 shows the near-field mesh and a portion of the extruded coarse mesh. The extruded mesh scales with length of the matrix, so the element count is nearly identical for all simulations at approximately 1.9×10^5 elements with approximately 7.6×10^5 DOF.

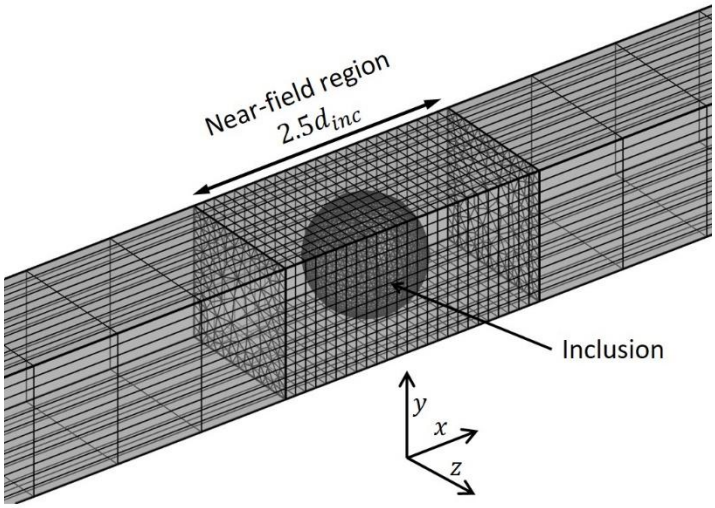

Figure 2.8: Meshing the domain with features of greatly different scale. The inclusion diameter is approximately $1/50$ the full length of the matrix. A fine mesh is used to accurately capture the behavior in and around the inclusion. The rest of the matrix is meshed by extruding the near-field mesh in the x-direction with a much coarser resolution.

Another requirement of the mesh is to maintain accuracy along the Floquet periodic boundary conditions. To avoid interpolation and associated error, the meshes on each periodic boundary are identical by defining the mesh on one face and copying it exactly to the opposite face. Finding the resolution that efficiently yields accurate results for both of these specific mesh regions and the mesh overall is confirmed with a mesh refinement study before accepting results from design evaluations.

Study and Post-processing

Unlike the two-layer Willis material, no analytical homogenization is done in post-processing. Design instances are evaluated by parameterizing all pertinent geometric variables, so they may be changed programmatically. The simulations are executed with the PARDISO solver. Scattered pressure fields are averaged across the probe planes and post-processed to yield reflection, R, transmission, T, and absorption, α, for both incident wave directions at each frequency. Given all constraints, response performance is measured with asymmetry of reflection in decibels, $20\log_{10}(|R_2/R_1|)$, as the performance indicator.

Scattered wave fields are returned by the study and can be plotted in 3D throughout the domain. The study also solves for stress in the inclusion. Figure 2.9 shows the scattered field plot at a single frequency for each direction of incident wave propagation as well as a 2D cross-section of the inclusion with the von Mises stress field visualized. The background pressure wave is 1 Pa for both directions, $p^{in} = 1Pa$. Figure 2.9a&b represents the case where $p_1^{in} = 1Pa$ with $p_2^{in} = 0Pa$, and Figure 2.9c&d represents the case where $p_1^{in} = 0Pa$ with $p_2^{in} = 1Pa$.

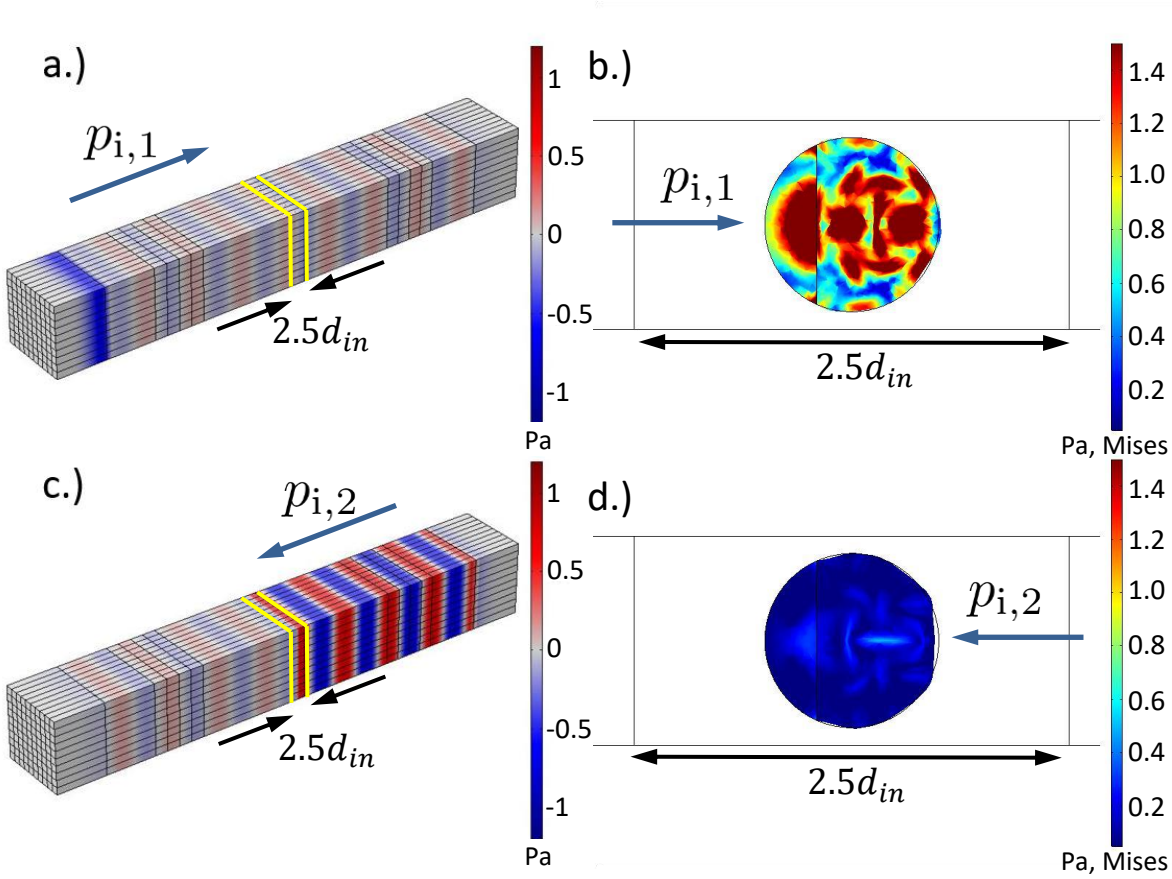

Figure 2.9: Total pressure fields for incident waves propagating along the $\pm x$-direction and the resultant stress field within the inclusions. (a, b) Results for the incident wave traveling from left to right, and (c, d) shows simulation results for incident wave propagating from right to left (c, d). Von Mises stresses are plotted on a cross section of the inclusion.

Scattered fields are post-processed to calculate the asymmetry of reflection. This analysis flow is utilized to evaluate and design AMSs of this type over a broad frequency spectrum and for many combinations of materials and geometric features. Specific materials and geometric features explored and how this analysis is used for design is covered in Chapters 5 and 6.

Chapter 3: Computational Design of AMM

Design engineers are often required to synthesize information, solve intertwining problems, and elegantly present the best designs. The nature of this design process has changed dramatically over the past several decades, driven largely by growth in computational capabilities. By harnessing computational models, an engineer can solve increasingly difficult problems such as design of AMMs, but designing in a computational landscape requires an expanded suite of tools for generating information and learning from it to make better design decisions. In light of this challenge, machine learning (ML) algorithms have emerged as a particularly useful class of tools for engineering design exploration and optimization.

The commonality shared by all the challenges of designing AMMs presented in Section 1.2 is that evaluating an AMM's performance requires enough computational expense to motivate maximizing the information gained from making each evaluation. In other words, a design approach is a valuable tool when it prudently uses expensive system evaluations while providing useful information to the designer. Iterative design is impractical for systems that have complex functional relationships between performance and design features because individual data points provide little information about the design space. Where exhaustive evaluation is prohibitively expensive, another method is necessary to characterize performance in a design space. For AMM design, computationally efficient prediction of a prospective design's performance requires a surrogate model that is capable of creating a mapping between combinations of design features and performance. Supervised ML models do exactly that. They are trained on ground-truth data about the feature-performance relationship so that they can predict output performance from input features in a design space.

Design space mappings generated with supervised ML techniques can be used to inform early-stage design exploration, support optimization, provide reliability assessments, and aid convergence in multiobjective or multilevel problems. However, the accuracy of the mappings can vary based on problem factors such as the number of design variables, presence of discrete variables, multimodality of the underlying response function, and amount of training data available. Additionally, there are several useful ML algorithms available, and each has its own set of algorithmic hyperparameters that significantly affect accuracy and computational expense. This chapter explains the use of machine learning for engineering design exploration and optimization problems by describing how ML models can be used to improve design tasks. Section 3.1 discusses how ML is used for engineering design and specifically for AMMs. Section 3.2 and Section 3.3 present the background on classifiers and metamodels that are implemented and adapted in the research tasks.

3.1 Utility of Computational Design Methods

In engineering design exploration and optimization, supervised ML techniques used for approximating relationships between combinations of design features and performance are broadly separable into metamodeling and classification algorithms. This approximated input/output relationship is also called a mapping of the design space. The difference between metamodels and classifiers is defined by the type of output response they provide. Metamodels map to absolute performance values while classifiers map to distinct groupings known as classes. Classes are problem specific but, for example, could consist of: plant species [67], digits [68], or spam emails [69]. Designers utilizing classification in engineering disciplines are interested in whether designs satisfy one or many performance specifications subject to design constraints. Classifiers that map the

feasible design space are focused on identifying combinations of design variables that satisfy the constraints and meet preferred thresholds for the objective functions. This task is an example of inverse mapping, which is different from forward mapping strategies that are also used pervasively in engineering design. An inverse mapping of the design space seeks to identify the comprehensive set of design variable values that meet a set of performance requirements or thresholds. Metamodeling, on the other hand, produces a forward mapping which accepts unique design variable values as input and predicts their performance.

Selecting an algorithm depends entirely on its suitability for achieving the desired task. Besides creating a metamodel or identifying classes of designs, the characteristics of the task and data affect how suitable an approach may be. Training and testing ML models provides a quantitative measure of model quality, but the expanse of available approaches and variants precludes exhaustive trialing. Furthermore, both metamodels and classifiers may be suitable for a given task in engineering design since they share many aspects of functionality. There is no universally deterministic relationship between task characteristics and ML algorithm selection but broad guidance for algorithm selection is available. Studies and resources such as Sharpe and Wiest *et al*. [34], Fernandez-Delgado *et al.* [70], Borchani *et al*. [71], and the scikit-learn user guide [72] have compared methods on sets of design tasks to provide context and guidance for interested designers. Further adding to the extensive suite of available ML approaches, individual algorithms may be combined to form what are called ensemble methods [73]. The limitlessness of ML model architectures and datasets makes methodology selection for a given problem somewhat of a creative art. Human designers utilizing ML methods must learn how to identify candidate methods for their task and prioritize implementation trials. In practice a limited subset of ML techniques should be selected and then trained and tested to compare performance.

3.1.1 Motivation for Designers

A design engineer may seek to map the design space directly for several reasons. One example is set-based design, which focuses on solving distributed design problems by delaying commitment to a single point solution and preserving a diversity of options for identifying mutually satisfactory cross-disciplinary solutions [74]. In this context, classifiers have been used to solve distributed, multidisciplinary design problems, which are decomposed into interdependent sub-problems that share coupled design variables [75]. A similar set-based approach may be applied to multiscale or multilevel design problems for which it is important to map the input design space for an upper-level sub-problem because it may define the performance constraints of a lower-level sub-problem. Figure 3.1 illustrates an example of mapping a multiscale design problem with three distinct scales, micro-, meso-, and macro-scale. Each scale has a separate set of design variables and satisfactory design instances can be mapped across scales. For these applications, classifiers are used as a means of guiding and potentially improving the efficiency of design exploration. Specifically, they are used to map the boundaries of the feasible design space accurately and efficiently [76] [77] [78] [79]. Mapping the boundaries of the feasible design space accurately can be especially important for improving the efficiency of design exploration because it prevents potentially expensive simulation- or experiment-based exploration of infeasible designs [80]. For highly nonlinear, nonconvex design problems, these spaces can be disjoint and assume arbitrary shapes, making it difficult to capture them with simple techniques such as intervals [75] [77].

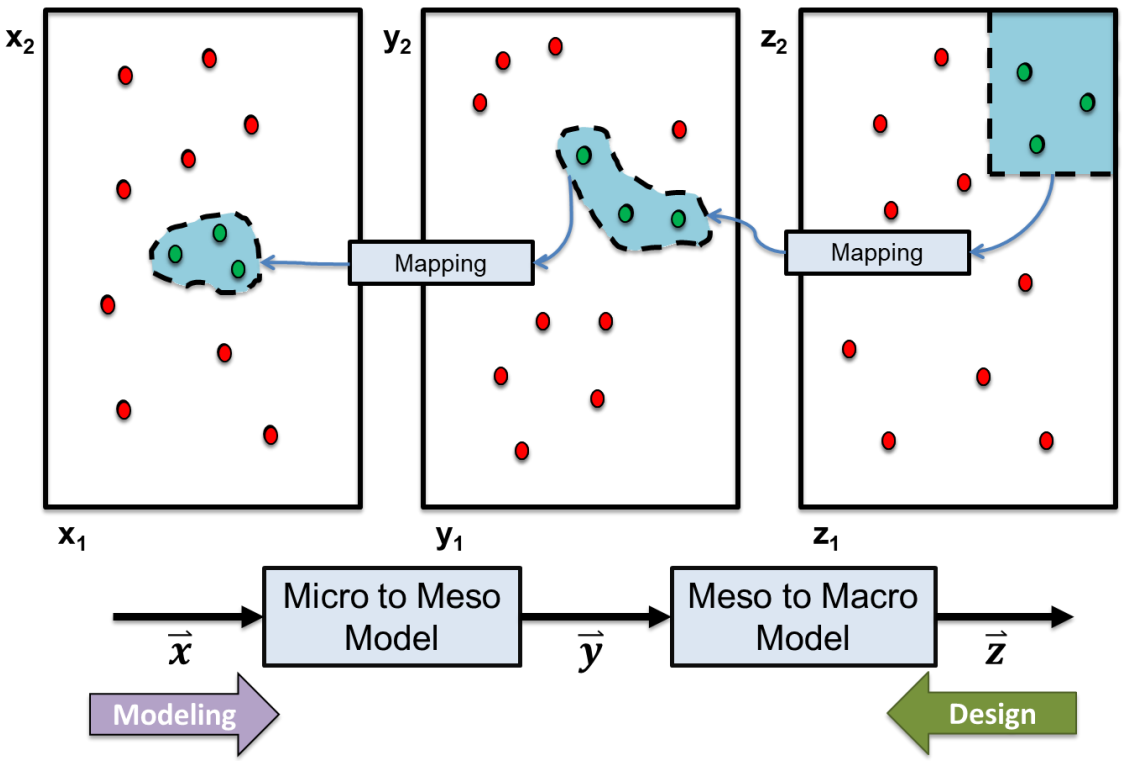

Figure 3.1: Inverse design space mappings allow design instances and regions to be tracked across scales of a multiscale design task whether satisfactory (green) or unsatisfactory (red). This mapping traverses three distinct scales referred to as micro, meso, and macro.

Metamodeling methods are designed to predict outputs that are continuous in nature. Consequently they are not effective for mappings that have discrete outputs and they require algorithmic ensembles to address multilevel and multidisciplinary design tasks. However, metamodels are capable of mapping the feasible region of a design space by applying a constraint or performance threshold to a developed functional mapping. Because of their continuous output, metamodels are able support optimization within a design space [81] in many different ways: global optimization [82] [83], local optimization [84], and model validation optimization [85] [86] [87]. Additionally, metamodeling techniques have been applied to support sample generation within a space [88] [89] [90].

Although output values are continuous, metamodeling methods are subdivided by whether or not they provide gradient information [91]. Metamodels that fit data to a functional form, such as linear regression, generate a polynomial response surface that can be differentiated analytically. On the other hand, metamodels such as decision trees, random forests, and neural networks do not provide differentiable analytical functions and require numerical techniques to obtain gradient information.

While there are myriad applications for metamodels and design space mappings, this work focuses on methods that are relevant to improving the design processes of AMMs. Specifically metamodel-driven sampling optimization, model validation for robust design, and design exploration in high-dimensional spaces.

3.1.2 Model Training and Tuning

Machine learning models are trained to be as accurate as possible so that they may provide trustworthy information to the designer or supplant a more expensive function call. A good fit to the response behavior of the underlying data source is essential for a model to be useful but the user must also be careful to avoid overfitting. Overfitting to data is a common pitfall that is characterized by a model that perfectly matches the training data but is inaccurate for data points in the design space that are not present in the training data [92]. Overfitting is especially problematic when the underlying data is noisy. An overfit model will exhibit a significant accuracy bias toward training data and points in the design space that are extremely close to training data points in a Euclidean sense. Conversely, an underfit model may exhibit very little bias of fit throughout the space but have a very large mean value of error. Both overfit and underfit models generalize poorly throughout the space [93]. For example a one-dimensional polynomial regression model,

$$y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j, \tag{3.1}$$

with multiple polynomial terms of order M will over- or underfit relative to the quantity of data, response behavior, and highest order term M. Figure 3.2. shows models of varying order fit to the same data set which is a sine function with a random noise component, $y = \sin(x) + \epsilon$, where $\epsilon \in \mathrm{E} \sim N(0, \frac{1}{5})$.

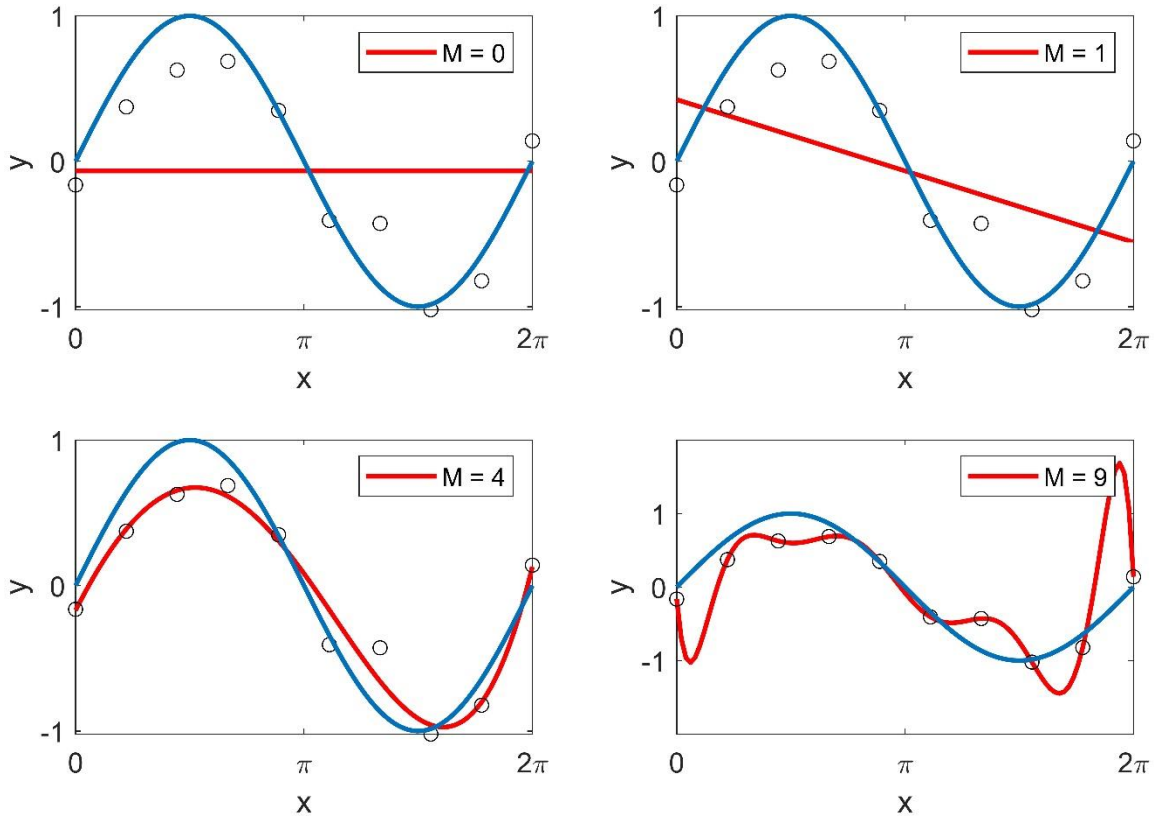

Figure 3.2: The effects of over and underfitting are shown for the polynomial regression on underlying noisy data (black markers) with $y = \sin(x) + \epsilon$, where $\epsilon \in \mathrm{E} \sim N(0, \frac{1}{5})$. The blue curve is a noiseless sine function $y = \sin(x)$ and red curves are polynomial regressions (Equation 3.1) of order M. In this example the $M = 0$ model underfits the data while the $M = 9$ model overfits the data.

This example is a simple way to visualize over- and underfitting but all ML models are subject to similar issues. An ML model's fidelity to the input/output behavior is limited by the quality of the training data, the tuning of the model, and the method by which it is

trained. This issue is mitigated by tuning a model to exhibit high fidelity to the relationship it is trying to approximate, which requires a deliberate procedure to train, test predictive accuracy, and iterate if necessary. There are two steps that are necessary for any statistical or machine learning algorithm to achieve high fidelity: data generation and hyperparameter optimization.

Data Generation

Generating data necessary to train a ML model for simulation-based AMM design requires querying the FEA sufficiently throughout the design space, and there is a wealth of sampling strategies available. Chapter 11 of Bishop 2006 [35] introduces many sampling methods such as random sampling from distributions, Markov Chain Monte Carlo, Gibbs sampling, among others. Modern sampling techniques used in engineering design are space-filling methods and adaptive methods [94] [95] [96]. Space-filling samples are generated using quasi-random sequences that distribute samples quasi-uniformly throughout the space of possible feature settings. They are designed to fill a space of arbitrary dimensionality with low-discrepancy and gained popularity for applications involving numerical integration [97]. The most common space-filling sampling methods are Halton [98], Sobol [99], and Faure [100] sequences, as well as Latin Hypercube sampling [101]. The three named sequences enable any number of samples to be taken incrementally while continuing to fill the space with minimal discrepancy. This is an advantage over sampling on a grid which requires sample size to be doubled when samples are added in order to maintain uniformity throughout a space. The named sequences and Latin Hypercube perform very similarly across the range of dimensionality $1 \leq D \leq 10$ [97] [102].

Because the named sequences are nearly interchangeable in application, the Halton sequence is the method used in this work to generate space-filling samples. The Halton sequence uses coprime numbers as its bases and can be expressed mathematically in multiple ways. In summary of Morokoff and Caflisch's [97] expression of the sequence, $(n)_p = a_k a_{k-1} \dots a_0$ is the base p expansion of any integer n with $0 \leq a_i < p$ and

$$S_p(n) = \frac{a_0}{p} + \frac{a_1}{p^2} + \cdots + \frac{a_k}{p^{k+1}} \tag{3.2}$$

is a one-dimensional uniformly distributed sequence with $0 < S_p(n) < 1$ for all n. The Halton sequence is an s-dimensional generalization of Equation 3.2 expressed as

$$\boldsymbol{x}_n = \left(S_{p_1}(n), \dots, S_{p_s}(n)\right) \tag{3.3}$$

where $(p_1, \dots, p_s)$ are prime integer bases. For example, consider a two-dimensional sequence $\boldsymbol{x}_n = \left(S_{p_1}(n), S_{p_2}(n)\right) \in [0,1]$ with the bases $(p_1, p_2) = (2, 3)$. The first seven terms of the Halton sequence generate as

$$\left(\frac{1}{2}, \frac{1}{3}\right), \left(\frac{1}{4}, \frac{2}{3}\right), \left(\frac{3}{4}, \frac{1}{9}\right), \left(\frac{1}{8}, \frac{4}{9}\right), \left(\frac{5}{8}, \frac{7}{9}\right), \left(\frac{3}{8}, \frac{2}{9}\right), \left(\frac{7}{8}, \frac{5}{9}\right), \dots$$

The sequence continues following this same pattern. Figure 3.3 illustrates how this sequence of points fills a two-dimensional space.

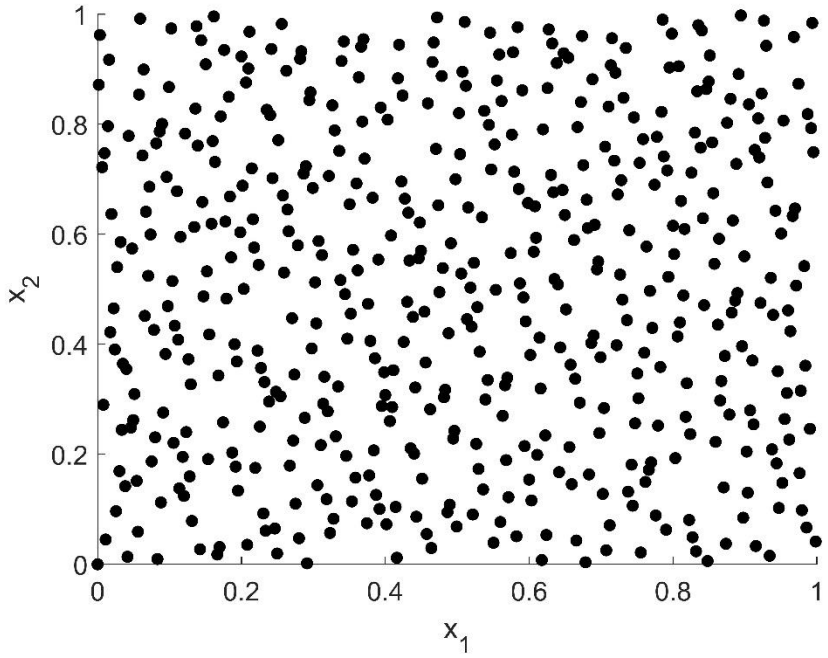

Figure 3.3: Two-dimensional Halton sequence generated in a unit square domain.

Sequences in higher dimensions fill the space in the same way. Issues such as collinearity may arise with certain combinations of prime bases so a practical solution is to scramble the sequence by permuting the coefficients or by scrambling the bases with a reverse-radix scheme [102]. Designs are represented with the Halton sequence by scaling the sequence-generated data points to the range of feature settings that are considered for the design. Taking the inclusion geometry features presented in Section 2.2.2 and Figure 2.6 as an example, the Halton sequence would be transformed such that

$$x_n \in [0,1] \rightarrow r_a \in \left[r_{a,min}, r_{a,max}\right]. \tag{3.4}$$

Data scaling and transformation is important both for generating samples and for pre-conditioning training data for input to an ML model. Generating combinations of features that can be evaluated with FEA or another expensive black-box function must be transformed from a generic range to values that are physically meaningful such as in

Equation 3.4. However, ML model performance may suffer when training data is varied in magnitude and distribution [103] [104]. ML model accuracy improves by transforming the data again before training the model [103]. In this work two transformations are applied: min-max scaling, and standardization. Min-max scaling transforms a sample of a single design feature $\boldsymbol{x} = [x_1, \dots, x_N]$ with length N expressed as a vector into

$$\boldsymbol{x}' = a + \frac{(\boldsymbol{x} - \min(\boldsymbol{x}))(b-a)}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}, \tag{3.5}$$

where (a, b) represent the post-transformation target minimum and maximum values, respectively. Standardization transforms $\boldsymbol{x} \to \boldsymbol{x}'$ so that it has zero mean, $\mu = 0$, and unit variance, $\sigma^2 = 1$:

$$\boldsymbol{x}' = \frac{\boldsymbol{x} - \overline{\boldsymbol{x}}}{\sigma}. \tag{3.6}$$

Min-max scaling is used to transform a space-filling sample for input to FEA simulation while both are applicable as preprocessing steps for ML model training. Which transform works best for preprocessing is algorithm-dependent, and in practice both are worth trying as part of the model tuning and validating processes. In this way, preprocessing data transforms are very similar to model hyperparameters, which also must be tuned to optimize model performance.

Hyperparameter optimization

The tuning process requires searching through, or optimizing, various hyperparameter values to determine the values that lead to the best performance for a specific problem. A hyperparameter of a ML model is a parameter of a model that can be set and affects the learning process. Hyperparameters differ from other parameters of a model because they are not adjusted automatically during the training process. Using polynomial regression (Equation 3.1) as an example, term coefficients $\boldsymbol{w}$ are adjusted as the model is trained (such as by minimizing least squares) and are not hyperparameters.

The quantity of terms M is a hyperparameter. All ML models have hyperparameters and those specific to the models used in this work are discussed in Sections 3.2 and 3.3.

Searching for suitable hyperparameters requires retraining the model for each combination of hyperparameters considered, thus making the tuning effort required to reach acceptable performance an important factor in the overall computational expense of implementing an algorithm. Tuning is commonly performed using a cross-validation scheme [105], in which rotating subsets of the training data are excluded from the training set and used to evaluate the model's performance (chapter 5 of [106] provides more in-depth background discussion of cross-validation). The performance is then aggregated to find the hyperparameter values that yield the best results throughout the entire space. Model performance quantified by an appropriate criterion. For classification the criterion is the relative quantity of correct/incorrect class assignments and how incorrect assignments are made. For metamodeling tasks, the criterion is a measure of the deviation between predictions and true values. More detail on model performance metrics is include in Sections 3.2 and 3.3.

Cross-validation schemes are used to prevent overfitting data on a single test set. Hyperparameter values can be tuned by simply testing a fixed grid of hyperparameter values or a random set of hyperparameter values, or by implementing sequential techniques such as Bayesian optimization to direct the search towards higher performing hyperparameter values at the expense of longer training time. A simple k-fold cross-validation grid search approach is used in this work where $3 \leq k \leq 10$. Some algorithms are more robust to hyperparameters and offer relatively consistent performance across a range of settings, while others are very sensitive to parameter settings.

3.2 MACHINE LEARNING CLASSIFICATION

As described in Section 3.1.2, classification can be used in engineering design exploration and optimization problems to map regions of the design space that satisfy requirements or constraints of interest. These requirements can be formulated mathematically as an inequality constraint as follows:

$$Decide\ c = c_1\ if\ f(x) \leq f_{thresh}\ ;\ else\ decide\ c = c_2 \tag{3.7}$$

where x is a candidate design, f is the performance function of interest, and c_1 and c_2 represent two classes of interest, for example, feasible and infeasible. The true classification designation can be known exactly by evaluating a particular design for a deterministic performance function of interest such as FEA. The performance function is often expensive to evaluate, however, motivating a more efficient alternative for predicting whether a candidate design meets the desired inequality condition. Many classifiers use previously evaluated points to estimate the probability that a new design meets the desired criteria without explicitly evaluating the performance function. A typical probabilistic decision criterion for classification algorithms is shown in Equation 3.8,

$$Decide\ c = c_1\ if\ p(c_1|x) > p(c_2|x)\ ;else\ decide\ c = c_2 \tag{3.8}$$

where $p(c|x)$ is the conditional probability of the class given the candidate design, x.

A number of factors should be considered when implementing a classification approach to support the decision criteria defined in Equations 3.7 and 3.8. The first consideration is the type of classifier to implement. There are two broad categories of classifiers: generative and discriminative [107]. Examples of generative classifiers include naïve Bayes classifiers and Bayesian network classifiers. Support vector machines and neural networks are examples of discriminative classifiers. Discriminative classifiers directly model the conditional probability $p(c|x)$ that a data point, x, is a member of a specific class, c. Generative classifiers instead model the joint probability of data and class

$p(c, x)$ which can then be transformed into the posterior probability of a class by using Bayes' rule $p(c|x) = \frac{p(x|c)\,p(c)}{p(x)}$. In many cases, the direct approach of the discriminative classifiers yields better classification accuracy [107], but not in all cases. The generative classifiers can offer different advantages, such as utilizing the joint probability provided by generative classifiers for sampling combinations of class and data, which may be of interest in sequential sampling approaches.

Computational expense is also an important concern. The computational expense for training classifiers grows with the number of training points and the number of design variables, and the scaling of training time with respect to these factors varies for different algorithm types. Table 3.1 summarizes the worst case theoretical time complexity of a few common classification algorithms where *n* is the number of training points, *m* is the number of variables, *t* is the number of trees in a random forest, *p* is the number of variables randomly sampled at each node of the random forest decision trees, *h* is the number of neurons (assumed to be constant in each layer for simplicity), *k* is the number of layers in a neural network, and *i* is the number of backpropagation iterations [72] [108] [109]. The linear time scaling of Gaussian Naïve Bayes makes it the most efficient algorithm of this group to train. Random forests and support vector machines exhibit exponential scaling with the number of training points, causing their computational expense to increase quickly for large datasets. The efficiency of the fully-connected neural network depends heavily on the number of layers and neurons present in the network, but many modern networks utilize large numbers of parameters that result in computational expense significantly greater than the other algorithms presented here. It is important to note that the computational complexity in Table 3.1 refers to the cost of training the classifiers. Training time varies by problem, ML model, and hyperparameter optimization methods. All of these

classifiers require less, often trivial, computational expense for prediction relative to the computational cost of training them.

Algorithm	Computational Complexity
Gaussian Naïve Bayes	$O(nm)$
SVM	$O(n^2m)$ to $O(n^3m)$
Neural Network (Perceptron)	$O(nmh^ki)$
Random Forest	$O(pn^2t\log(n))$

Table 3.1: Computational complexity for training classifiers

Every classification approach expresses a set of underlying assumptions about the data via its mathematical form and the types of hyperparameters that help define its underlying model. Consequently, classifier performance can vary widely depending on the characteristics of the problem, including the number of variables, variable type (continuous, discrete), the strength of interactions between variables, and the modality of the design space.

3.2.1 Classifier Scoring

A multitude of scoring metrics have been defined to quantify classifier performance [110], but they all seek to express the degree of similarity between the true and predicted classes of each test sample. The simplest statement of the similarity between the predicted and true classes of a sample set is represented by a confusion matrix [106], as shown in Figure 3.4. *P* and *N* are the two classes: *positive* and *negative*. The positive class is typically associated with feasible or promising designs in an engineering design context. A correctly identified member of class *P* is called a true positive and represented by *TP*. *FP* is a false positive, which corresponds to a member of class *N* that is incorrectly

identified as a member of class *P*. It follows that *TN* and *FN* are true negative and false negative predictions, respectively.

		Predicted Class	
		P	N
True Class	P	TP	FN
	N	FP	TN

Figure 3.4: Confusion matrix for organizing classifier predictive performance

Most metrics used to score classifiers use some combination of the entries in the confusion matrix. For many classification tasks, accuracy (ACC), as defined in Equation 3.9, is a useful starting point for evaluating overall classifier performance.

$$ACC = \frac{TP+TN}{(TP+FP+FN+TN)} \tag{3.9}$$

There are some cases for which other classifier scoring metrics are more appropriate. For example, when there are many more instances of one class compared to the other in a data set, the classification problem is imbalanced. In these cases an accuracy metric can be misleading, as a classifier that blindly predicts all instances to belong to the majority class will show high accuracy but have limited predictive value. Specifically for designers, scoring metrics formulated to prioritize prediction of feasible or high-performing solutions may be preferred at the expense of overall accuracy. One such metric that provides information on classification of the positive (*P*) class is true positive rate (*TPR*), also commonly referred to as sensitivity or recall. Maximizing TPR increases the likelihood of identifying a true member of the positive class of interest:

$$TPR = recall = \frac{TP}{P} = \frac{TP}{(TP+FN)}. \tag{3.10}$$

Another metric that may be of interest is precision, or positive predictive value. The precision metric provides a measure of the prediction accuracy specific to the positive class by reporting the proportion of positive predictions that match with true class membership:

$$precision = \frac{TP}{(TP+FP)}. \tag{3.11}$$

The F1 score is a common scoring metric for binary classification that balances precision and recall using a harmonic mean. The F1 score is especially popular for assessing classifier performance in cases with a class imbalance or with greater importance placed on classifier performance for the positive class:

$$F1 = 2 * \frac{precision*recall}{precision+recall}. \tag{3.12}$$

Finally, false negative rate (FNR) can also be utilized to assess whether the classifier is missing potential members of positive class. Minimizing FNR allows designers to maximize their chances of identifying all potential designs of interest.

$$FNR = \frac{FN}{P} = \frac{FN}{(TP+FN)} = 1 - TPR. \tag{3.13}$$

In combination, these metrics capture how well a classifier is identifying regions of the design space containing positive (typically, feasible high performance) designs.

3.2.2 Gaussian Naïve Bayes

Bayesian classifiers, which are based on Bayes' theorem, have been found to perform well on a variety of classification problems [75] [111] [112]. The naïve Bayes (NB) classifier is a well-known representative of Bayesian classifiers, which assumes that all features are class-conditionally independent. They use Bayesian decision theory to

determine the probability that a design belongs to a defined class, according to the formulation of Bayes rule in Equation 3.14:

$$P(c_l|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c_l)P(c_l)}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|c_l)P(c_l)}{\sum_{k=0}^{1} P(\boldsymbol{x}|c_k)P(c_k)}. \quad (3.14)$$

This formulation is for a binary classification where the prior probability of each class is represented by $P(c_l)$, the class conditional probability for a given set of *D* design variables $\boldsymbol{x} = [x_1, x_2, \dots, x_D]$ is represented by $P(\boldsymbol{x}|c_l)$, and the probability that the design belongs to a designated class is called the posterior probability of class membership and is represented by $P(c_l|\boldsymbol{x})$. Class assignment of a given design $\boldsymbol{x}$ is chosen based on the relative magnitude of each class's posterior probability:

$$Decide\ c = c_0 if\ P(c_0|\boldsymbol{x}) > P(c_1|\boldsymbol{x}); else\ c = c_1. \quad (3.15)$$

Prior probabilities, $P(c_l)$, can be formulated in many different ways depending on the expected distributions of each class, but a simple counting prior, as shown in Equation 3.16, is often sufficient. N_l represents the number of samples in class *l*, while *N* is the total number of samples.

$$P(c_l) \cong \frac{N_l+1}{N+2} \quad (3.16)$$

To determine the class conditional probability, $P(\boldsymbol{x}|c_l)$, a kernel density estimate (KDE) is constructed. Kernel functions are centered on each candidate design point, and those functions are aggregated into the KDE. Although many kernel functions can be used for constructing KDEs, the Gaussian kernel is implemented in this work. Using a Gaussian KDE, the class conditional probability can be evaluated with Equation 3.17:

$$P(\boldsymbol{x}|c_l) = \frac{1}{N_l}\sum_{j=1}^{N_l}\prod_{i=1}^{D} \frac{1}{\sigma_{i,l}\sqrt{2\pi}}\, e^{\frac{\left(x_i - \hat{x}_i^j\right)^2}{2\sigma_{i,l}^2}} \quad (3.17)$$

Each Gaussian kernel is assigned a *D*-dimensional standard deviation $\boldsymbol{\sigma}$. The variable x_i represents the design point in indicial notation and $\hat{x}_i^j$ is the data point at the center of the j^{th} kernel in the i^{th} dimension. The standard deviation sets the width of each kernel and assigning the value that yields the best performing KDE is the topic of much research [113] [114] [115] [116], but in this work, it is treated as a hyperparameter and tuned to maximize classification accuracy for the problem at hand. Although the variables in this discussion are described as continuous variables, discrete variables can be accommodated straightforwardly by substituting frequency-based distributions for the continuous distributions that define the class-conditional probabilities.

3.2.3 Bayesian Network Classifiers

Bayesian network classifiers (BNCs) are a generalization of NB where features may be conditionally dependent for a given class c. The word *network* is used because dependency between features, x_i, can be represented as a network or graph [117]. Defined as a graph, the features are nodes and dependency is represented with an edge. BNCs have been shown to be useful in design exploration because they can be used to partition a design space according to the ability of candidate designs to meet specified performance requirements [75]. Effectively, they enable inverse mappings of regions of interest in a design space. Furthermore, their roots in Bayesian statistics enable incorporation of prior expert knowledge and support for adaptive sampling [118].

The general BNC allows for a network structure where conditional dependencies between features may exist. This network structure of dependencies varies from the assumption that all features are dependent on one another to the assumption that no features

are dependent on each other (naïve). Figure 3.5 illustrates these two cases and their forms of class conditional probabilities.

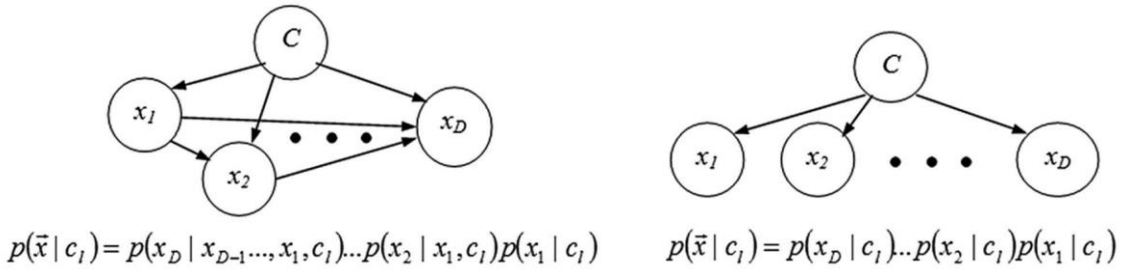

Figure 3.5: Fully connected BNC with feature conditional dependencies (left) and a naïve BNC where all features are conditionally independent (right).

The class conditional probability for the naïve case is much easier to calculate mathematically and therefore incurs less computational expense when training and testing data that is high-dimensional. Due to computational expense, the best network is the one that accurately classifies with minimal conditional dependencies. Selecting a network structure requires information about the underlying data or an optimization process. Knowledge of the underlying data can be obtained by from the original function or by evaluating the correlation of a training data set. Assuming a black-box function, optimization processes are suitable ways to identify an appropriate network structure. Sharpe *et al*. [119] demonstrated this using a genetic algorithm for optimization and evaluating BNC performance based on speed of convergence to a minimal error with respect to the quantity of training data. Regardless of network structure, the generative nature of the BNC makes it a powerful tool that can be extended beyond discrete class identification.

By applying Bayes' rule to the class conditional probabilities, the posterior probability of class membership is calculated separately for each class of interest. For

example, in a binary classification scheme (e.g. high-performance, c_1, versus low-performance, c_0, with respect to specified requirements) a design is evaluated twice to determine $P(c_0|\boldsymbol{x})$ and $P(c_1|\boldsymbol{x})$, and the candidate design is assigned to be a member of the class with the larger posterior probability, according to Equation 3.14. In some cases a designer may wish to bias the assignment toward a certain class based on risk or some *a priori* knowledge. These heuristic risk factors are defined as $\lambda_l \in [0,1]$ and applied as weights on the posterior (default $\lambda_l = 1 \; \forall \; l$). The difference between the posterior probabilities is called the posterior class discriminant (PCD).

$$\begin{aligned} PCD \in [-1,1] &= \lambda_1 P(c_1|\boldsymbol{x}) - \lambda_0 P(c_0|\boldsymbol{x}) \\ &= \frac{\lambda_1 P(\boldsymbol{x}|c_1)P(c_1) - \lambda_0 P(\boldsymbol{x}|c_0)P(c_0)}{P(\boldsymbol{x}|c_1)P(c_1) + P(\boldsymbol{x}|c_0)P(c_0)} \end{aligned} \tag{3.18}$$

The *D*-dimensional hypersurfaces along which $PCD = 0$ represent decision boundaries in the space. An example of posterior probability surfaces for a binary classification in 2D is shown in Figure 3.6. In the figure, green and red points represent instances belonging to different classes. The blue and red surfaces represent the posterior probability of their respective classes throughout the space. The probability of each class is equal ($PCD = 0$) where these surfaces intersect as represented by the black curves in Figure 3.6.

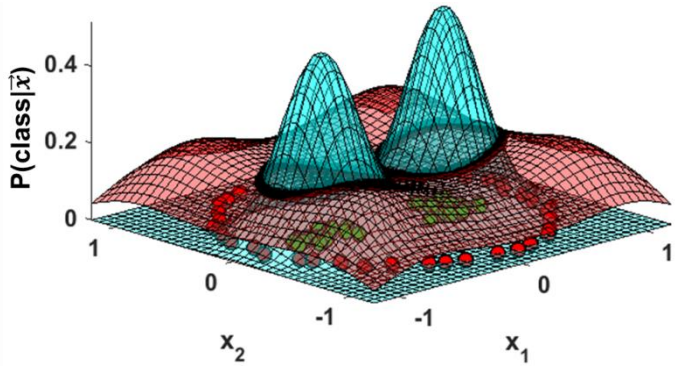

Figure 3.6: Posterior probability surfaces over an example 2D design space. Each surface is generated from the sample points using a kernel-based Bayesian network classifier (BNC) [120].

The hypersurfaces along which $PCD = 0$ are particularly important for accurate classification because they are the decision boundaries between design space regions of different class membership. When the number of training points available for one class is much greater than for another class, the PCD can be skewed in favor of the dominant class, a condition called imbalanced classification, which is addressed in Chapter 4.

3.3 ML METAMODELING

A metamodel approximates a function $f(\boldsymbol{x})$ with another function $g(\boldsymbol{x})$ that is computationally cheaper to evaluate. In ML, $g(\boldsymbol{x})$ need not be an explicit mathematical function. It is often an algorithm or ensemble of algorithms that work together as a metamodel of $f(\boldsymbol{x})$. Because the metamodel, $g(\boldsymbol{x})$, is fit to a limited sample of results from $f(\boldsymbol{x})$, training the metamodel is an optimization problem where error, ϵ, between the two functions is minimized:

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) - \epsilon, \tag{3.19}$$

$$Minimize\ \epsilon = |f(\boldsymbol{x}) - g(\boldsymbol{x})|. \tag{3.20}$$

The base of available metamodeling methods is extremely large [70]. Regression-like methods were used by many scientists in the 18th century and first clearly formulated by Legendre in 1805 using the method of least squares for linear regression [121]. Since then the research area and applicability of metamodeling methods has grown continuously and taken many forms. Metamodeling is a well-covered topic and an exhaustive review of those methods is outside the scope of this work. However, there are some important factors to consider when using metamodeling for the engineering design methods presented in Chapters 4 and 6.

Similar to classification models, metamodels have an underlying functional form and parameters that are optimized so that the metamodel most accurately replicates the response behavior of the expensive model. Simple examples of metamodels include linear regression,

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=1}^{M-1} w_j x_j, \tag{3.21}$$

and the non-linear polynomial regression in Equation 3.1. These two formulations have limitations for predictive accuracy because of their form. As the name implies, linear regression allows only linear combinations of the dependent variables, $\boldsymbol{x}$, and the scalar weight terms, $\boldsymbol{w}$, are tuned. Polynomial regression is non-linear but still restrictive to functions that are linear in the weight terms.

The utility of a metamodel is greater when it can represent severely non-linear and multi-modal responses, so general non-linear approximation models are powerful. Once again there are myriad forms of non-linear approximators. A straightforward extension from linear regression is replacing the dependent variables with non-linear basis functions of those variables,

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}), \tag{3.22}$$

where ϕ_j is a basis function. Any function could be used as a basis function. In fact, the polynomial regression in Equation 3.1 is using a polynomial basis function, and the linear regression in Equation 3.21 is using a linear basis function. Basis functions are pervasive throughout ML and are often used in kernel functions. The most popular example is the radial basis function kernel [35]. Kernel functions are used in support vector regression, Gaussian process models, and kernel ridge regression [35] [122]. Other non-linear approximators that do not use basis functions exist such as the neural network, which is covered in greater detail in Section 3.3.2. Regardless of the underlying form, training metamodels requires minimizing the error, and the computational expense of training

increases as the number of parameters increases. Hyperparameter settings such as choice of basis functions, kernel functions, and other factors also affect training expense.

3.3.1 Metamodel Scoring

Training is carried out by minimizing an error term. The error term may also be called the loss function or scoring metric. For basic least squares, the sum of squared residuals is minimized where the residual at any observation, i, is $r_i = y_i - \hat{y}_i$ with y_i being the observed response and $\hat{y}_i$ being the response predicted by the metamodel:

$$RSS = \sum_{i=1}^{N} r_i^2. \tag{3.23}$$

This metric is viable but because the values are an aggregated square of the error, the quantity may be quite large relative to the individual errors and therefore is not intuitive for a designer. Other scoring metrics have the advantage of better representing the error by more closely matching the scale of the data while remaining convex for versatility in optimization processes [123]. Root mean square error (RMSE) applies an inverse square term to regularize the error magnitude toward the error of a single data point as well as measuring the mean of all observations to provide an average error that would be expected at each one:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} r_i^2}{N}}. \tag{3.24}$$

Mean absolute error (MAE) improves upon RSS by using an absolute value rather than a square and averages over the number of observations, N:

$$MAE = \frac{\sum_{i=1}^{N} |r_i|}{N}. \tag{3.25}$$

Both RMSE and MAE are used in this work and either may lead to stronger model performance depending on the characteristics of the ground-truth data, the metamodel, and the optimization method that minimizes the error term. For this reason, the scoring metric and optimizer used in training are hyperparameters of the ML model.

Optimization processes are used to minimize the loss function programmatically. Optimization is another area of research with a very robust history and catalog of methods available [124], [125]. In application to ML, iterative methods use the data available to search for a minimum using either gradients or finite differences to numerically approximate gradients. A local minimum is found where the gradient of the function is zero, $f'(\boldsymbol{x}) = 0$ and the surrounding function values are all larger in magnitude. Some examples are quasi-Newton methods [126] and gradient descent [127]. Finding global minima in multi-modal functions requires adaptive or stochastic methods such as stochastic gradient descent (SGD) [128], Adam [129], and L-BFGS [130]. These three methods are used as optimizers for training neural networks because they are effective for training with a large number of parameters.

3.3.2 Neural Networks

An artificial neural network (NN) is a common supervised learning method for both regression and classification tasks. Despite their common use as classifiers, NNs are metamodels that have a form more similar to regression models. They employ a network learning structure that consists of three types of layers: an input layer, an output layer, and some number of hidden layer(s) [131] [132]. This basic NN architecture is also referred to as a multilayer perceptron (MLP). The size of the input layer depends on the dimensions of the inputs for the classification or netamodeling problem, while the size of the output layer changes based on the number of different output values or classes of interest. The network architecture, including the number of nodes in each hidden layer and the number of hidden layers, is determined based on the complexity of the problem. A general feed-forward MLP model is shown in Figure 3.7. The output of the k_{th} output node of the network can be represented as

$$y_k = \phi_o \left\{ \alpha_k + \sum_j w_{jk} \phi_h \left(\alpha_j + \sum_i w_{ij} x_i \right) \right\} \tag{3.26}$$

where w and α denote the weights and biases of the neural network, respectively, and ϕ represents the activation function. A *weight* is an attribute of edges of the network while a *bias* is an attribute applied at nodes. Activation functions are task dependent and effectively hyperparameters of the NN which may be applied in hidden layers and the output layer. They are similar to basis functions by enabling a non-linear mapping between inputs and outputs.

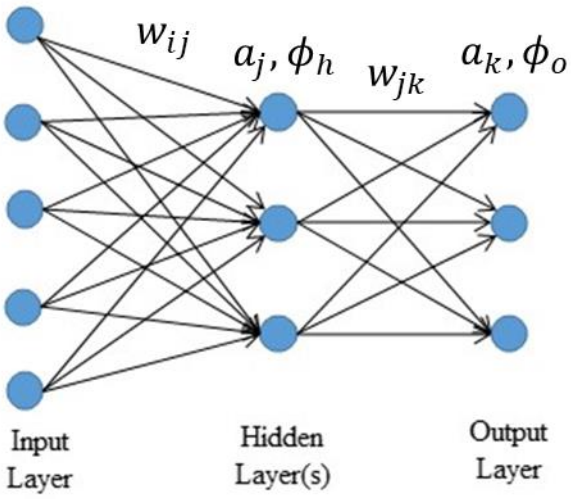

Figure 3.7: A general feed-forward neural network model with a single hidden layer and three output variables. Weights, w, biases, a, and activation functions, ϕ, are attributes of the network layers.

For regression tasks the activation function on the output layer is an identity function, $\phi_o(x) = x$, that returns the input value from the previous layer, and when used in hidden layers, it is essentially a linear regression model defined on a graph. Non-linear activation functions, such as rectified linear units (ReLU) and sigmoid functions can model more complex mappings [133] and are especially useful in classification tasks because the output is thresholded to represent one of two states, *on* or *off*. On/off output states is

terminology specific to NNs but in translation to general classification tasks, it is analogous to class assignment, $c_0 \; OR \; c_1$. Activation functions are required to be differentiable so that they may be trained via backpropagation [134] of gradient information and optimized using the loss functions and algorithms discussed in the previous section.

When training a neural network, the weights and biases are fit to minimize a loss function by using an optimization algorithm. The backpropagation neural network (BNN) is one type of NN that uses backpropagation as a supervised learning technique for the network training [134]. In the backpropagation training, the inputs are propagated to the output layer via the hidden layers, and the errors are backpropagated to the input layer. The propagation of error is called backpropagation because the network is modeled as a directed graph with information flowing from input to output. Errors are reduced iteratively with an optimizer by adjusting the weights and biases that are attributes of the graph. To query the BNN for classification, a new data point to be classified serves as an input to the trained BNN model, and the outputs of the trained BNN are the conditional probabilities of class membership that are interpreted according to the decision criterion of Equation 3.8 to produce the predicted binary class label. Various NN architectures may be trained by backpropagation so the term is excluded from the names of many NNs that utilize it.

Neural networks can be structured in many ways. A fully-connected feed-forward MLP is utilized in Chapter 5 because it performs sufficiently for the classification task at hand. This model and any others with a few hidden layers are shallow NNs. Deep NNs [135] are those with a large number of hidden layers and, often, multiple networks combined with aggregation operations in an ensemble method. Deep learning has gained significant popularity in the last few decades as advances in computing power, such as massively parallel GPU architectures, and software such as TensorFlow [136], have enabled the training of deep neural networks to process large datasets with impressive

results in the machine learning [137] [138] and design communities [139] [140] [141]. However, training deep learning networks requires access to very large amounts of training data to fit thousands of associated network parameters. Deep NNs and deep learning tasks in general make up an entire field of research with many communities focused on the area [142]. A few techniques are especially noteworthy. Convolutional neural networks (CNNs) are network architectures that use mathematical convolutions in hidden layers as well as pooling/aggregation operations. They have gained popularity when working with spatially structured data such as images or topologies [139] [137]. Recurrent neural networks (RNNs) use feedback loops in the network architecture and have been used successfully for sequential tasks such as those with time series data [143] and text strings [144]. Autoencoders (AE) are designed for compression and decompression of high-dimensional input data for dimensionality reduction, feature extraction, and denoising and have been applied successfully for many other tasks as well [145]. Graph neural networks (GNNs) are a class of methods that work with input data that is structured on a graph itself. They are capable of utilizing all of the network architectures described henceforth for operating on the data, as well as many others [146]. In Chapter 6 a GNN architecture is used to learn on spatio-temporal data from an acoustic metamaterial simulation. Further description of that specific implementation is given in that chapter.

This chapter has provided background information on metamodeling and classification concepts that are used for design of AMM and, in particular, to address the research tasks. Bayesian network classifiers are used for design exploration in Chapter 4 and ,because of the generative nature of the algorithm, provide information that is used in an adaptive synthetic sampling technique. Chapter 5 and 6 utilize very different NN architectures. An MLP classifier is employed as part of a manufacturing-aware design

framework in Chapter 5 while the method presented in Chapter 6 uses an ensemble of multiple NNs to enable design in a design space with a variable number of dimensions.

Chapter 4: PCD-informed SMOTE Sampling for Design Exploration[1]

Exploration of a design space is the first step in identifying sets of high-performing solutions to complex engineering problems. Classification algorithms are well-adapted for design space exploration in engineering tasks because they separate a space along boundaries of class membership and map regions of the space as either high- or low-performance. Furthermore, generative classifiers provide information about class member regions beyond just the class assignment. Generative classifiers provide the joint probability distribution of each class and that distribution is essentially a continuous function of the design variables. Each class's distribution may be exploited to improve a design space mapping such as with active learning and other adaptive sampling strategies [147] [78] [76] [79].

For the purpose of exploration, Bayesian network classifiers (BNCs) have been shown to be effective for mapping regions of interest in the design space even when those regions of interest exhibit complex topologies [75]. Bayesian network classifiers are generative and can be tuned for classification performance by adjusting hyperparameters. They can use various kernels with adjustable distributions to form the joint probability distributions and each class's distribution may be scaled heuristically. However, identifying sets of desirable solutions can be difficult with a BNC when attempting to map

[1] The majority of the content in Chapter 4 has been previously published in [255] and the primary author of that paper is also the author of this dissertation. Some additional background has been added for the dissertation and a comparison of the methodology to other similar methods has been included in Section 4.5.1. Carolyn Seepersad contributed to problem conception, supervised the work, and revised the manuscript. Caleb Sieck provided a template finite element model which was adapted by the author to generate data as described in Section 4.4.2. David Shahan and Clint Morris wrote the original MATLAB script for the Bayesian Network Classifier. The text of this chapter has been written by the author and edited by Carolyn Seepersad.

a space in which high-performance designs are spread sparsely (as described in Section 1.2) among a disproportionately large number of low-performance designs, resulting in an imbalanced classifier. By adding synthetic design points into the BNC training set, a designer can rebalance an imbalanced classifier and improve classification accuracy throughout the space. Methods that oversample a less populous class with synthetic points often adopt the acronym SMOTE, for Synthetic Minority Oversampling TEchnique, from the 2002 seminal paper by Chawla *et al.* [148]. Moreover, because the BNC is a generative method, synthetic data can be added near decision boundaries in the design space or in other areas where deeper investigation is beneficial.

In this chapter, a method is developed that creates a design space mapping of class membership probabilities for known training points and utilizes an ensemble of metamodels to interpolate between those points and generate synthetic high-performance points in a design space. For demonstration, this new synthetic oversampling approach is used to design the Willis material AMM with a two-layer heterogeneity detailed in Section 2.2. Specifically, design exploration is improved in a sparse design space that has a combination of discrete and continuous design variables by increasing the efficiency of sampling to create a valuable design space mapping. The effectiveness of the new synthetic oversampling technique, called PCD-informed SMOTE, is compared to other leading SMOTE approaches in literature.

4.1 Classification with Imbalanced Data

In many cases, analysts and designers seek to predict events that occur vary rarely within a dataset. Classic examples include identifying patients with early-stage cancer indicators from imaging data [149] in the medical field, identifying oil slicks from satellite imagery [150] or detecting instances of credit card fraud from a large number of legitimate

transactions [151]. This problem is called imbalanced learning or anomaly detection in the machine learning community and is characterized by a significant class imbalance. An imbalanced learning problem may occur for any number of classes but for convenience is described henceforth as a binary classification in which a minority class is of particular interest and all other outcomes are grouped into a majority class. In the binary framework, a minority class instance is considered a "positive" P result and the majority class instance is considered a "negative" N outcome. With the classes defined in this way TP is a true positive result, FP is a false positive, TN is a true negative, and FN is a false negative. Figure 3.4 provides a two-by-two confusion matrix that further illustrates these classification assignments.

These labels give insight into the classification task but more importantly serve as the basis for more informative evaluation metrics to make comparisons between classifiers. Quite a large number of evaluation metrics have been created for the purpose of comparing classifiers [152]. Some simple yet descriptive metrics include: true positive rate (TPR),

$$TPR = \frac{TP}{P} = \frac{TP}{(TP+FN)}, \tag{4.1}$$

false positive rate (FPR),

$$FPR = \frac{FP}{P} = \frac{FP}{(TP+FN)}, \tag{4.2}$$

and accuracy (ACC),

$$ACC = \frac{TP+TN}{(TP+FP+FN+TN)}. \tag{4.3}$$

These metrics measure the performance of the classifier in different ways and are selected by the designer with consideration of the task at hand. More detail on these metrics is provided in Section 3.2.1.

With a small number of minority class members used for training, classifiers tend to predict that all candidates belong to the majority class. When using a BNC, this occurs because the KDE for evaluating the majority class posterior probability overwhelms that

of the minority class, resulting in diminished minority class regions of the design space. Since almost all instances are members of the majority class, the ACC will be very high, even if every single minority class instance is misclassified [153]. For this reason, the ACC is insufficient for evaluation of imbalanced classification tasks because identification of minority class instances is very important and misclassification can be very costly. In the cancer detection example introduced earlier, for example, misclassification of a minority class instance means that a patient with cancer may not be diagnosed. In a problem of this nature, classifier performance is better described by its ability to identify minority class instances, so TPR and FPR are more meaningful scoring metrics than ACC.

In sparse design spaces, sampling is likely to lead to very few high-performance designs isolated locally among an overwhelming number of low-performance designs. In this case, training a kernel-based classifier yields a KDE that indicates a misleadingly small region of the space holds high-performance (minority class) designs. Figure 4.1 shows the effect of a class imbalance of 50:1 in a 2D design space. In this example space, a region of high performance exists near the middle of the space, but due to sparsity of sampling, only a single high-performance instance exists in the data set. The resulting design space mapping fails to accurately represent the performance regions because of the imbalance. Predictions based on this mapping would likely misclassify any high-performing designs near the single high-performance training point because the posterior probability of class membership for the low-performance points dominates that of the high-performance point(s).

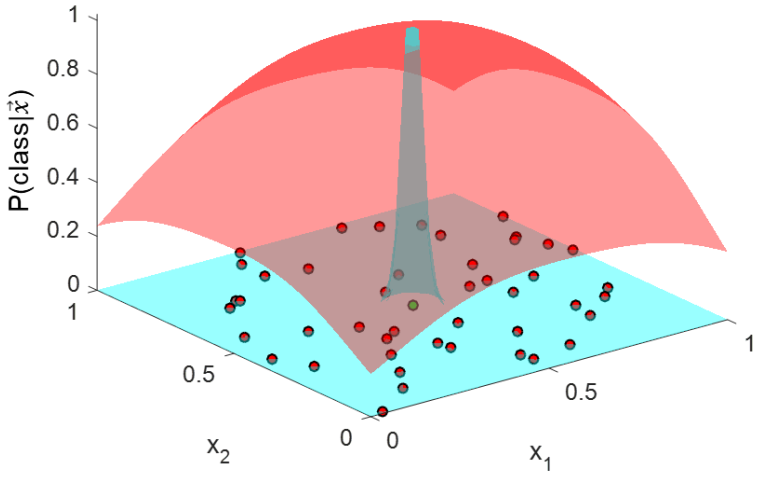

Figure 4.1: High-performance (blue) and low-performance (red) posterior probability surfaces over a sparse 2D design space with a class imbalance of 50:1. The 49 low-performance points are red; the single high-performance point is green. Due to imbalance, this mapping suggests that a misleadingly small region of the space holds high-performance designs.

Due to the challenge of imbalanced classification and its prevalence in machine learning tasks [150] [154] [155] [156] [157], significant research has focused on improving classifier performance under these conditions [148] [153] [158] [159] [160] [161] [162]. The two most general approaches are to train the classifier in a cost-sensitive manner and to restore balance by resampling to either decrease the number of majority instances or increase the number of minority instances in training data [153]. Resampling the training data set can be done in many ways including:

1. Gathering more real samples of the minority class, where real is defined as sampled directly from the underlying simulation
2. Randomly removing majority class samples (random undersampling)

3. Informed undersampling of the majority class using an algorithm to remove samples so minimal definition of the space is lost
4. Removing or altering overlapping instances from the two classes
5. Generating synthetic samples to bolster the minority class

Each method has benefits, but the generation of synthetic minority samples has been shown to be particularly powerful both in a static data set [148] and as an adaptive sampling tool [163].

4.3 SYNTHETIC MINORITY OVERSAMPLING

Imbalanced classification occurs in engineering design tasks when only a small number of high-performance designs exist within a design space that contains a disproportionately large number of low-performance designs. Imbalance is exacerbated in sparse design spaces with a mix of continuous and discrete variables that are poorly suited for gradient-based optimization techniques. In these cases, adapting a SMOTE approach adds capability to improve classifier performance and refine the exploration of sparse high-performance regions within the design space. The motivation for generating synthetic samples is purely cost-driven. Evaluating an expensive model to oversample is viable but if the class membership can be predicted otherwise with sufficient accuracy, synthetic samples provide valuable data while incurring less computational expense. The original SMOTE by Chawla *et al.* [148] works by adding synthetic points, $\boldsymbol{s}_i$, along *D*-dimensional lines between each minority class instance in the feature (design) space and its *K* nearest neighbors where *K* can be adjusted based on the desired breadth of minority oversampling. The synthetic sample is generated on a line by taking the vector difference between a minority class sample, $\boldsymbol{x_i}$, and one of its nearest neighbors, $\boldsymbol{x}_{ki}$, such that the synthetic sample can be defined as

$$\boldsymbol{s}_i = \boldsymbol{x}_i + (\boldsymbol{x}_{ki} - \boldsymbol{x}_i). \tag{4.4}$$

By generating synthetic samples only between known minority class instances, the authors assume they would also be members of the minority class. Of course, validation of this assumption is important when testing the model.

Adaptations of the original SMOTE are numerous [163] [164] [165] [166] [167], but the two most highly cited methods are borderline-SMOTE [163], and ADASYN [164]. Borderline-SMOTE calculates the K nearest neighbors of all minority class points and creates a separate set of the nearest neighbors associated with each minority class point. Within each set, the number of majority class points is counted and then the sets are sorted according to the ratio of majority to minority class nearest neighbors, $K_{maj}: K_{min}$. Synthetic samples are then generated between minority class points and their nearest K_{min} neighbors with preference to minority class points with a large number of neighboring points that belong to the majority class. Consequently, more synthetic minority class samples are generated near the decision boundary than not. The other popular technique, ADASYN, also identifies the ratio $K_{maj}: K_{min}$ for each real sample $\boldsymbol{x}_i$ and generates synthetic samples only between $\boldsymbol{x}_i$ and minority class nearest neighbors, $\boldsymbol{x}_{zi}$. It is expressed as

$$\boldsymbol{s}_i = \boldsymbol{x}_i + (\boldsymbol{x}_{zi} - \boldsymbol{x}_i) \times \lambda \tag{4.5}$$

where $\lambda \in [0,1]$ is a randomly sampled number that affects the placement of $\boldsymbol{s}_i$ along the line. Both of these methods improve upon the original SMOTE but leave opportunity for improving synthetic sample placement within a space.

The following section describes a novel way to generate synthetic samples of the minority class and, by using them to train a BNC, improve the accuracy of the design space mapping. As a result, the model is improved and class prediction of candidate designs

becomes more accurate. This method is particularly advantageous in sparse and/or imbalanced design spaces where the BNC model underestimates the size of high-performance regions in the design space. It uses class membership probability distributions to logically select where to add synthetic training points in a design space and improve definition around decision boundaries.

4.4 PCD-INFORMED SMOTE TO IMPROVE BNC TRAINING

The SMOTE adaptation developed here exploits a feature of the class probability distributions that are produced by generative classifiers. Unless a space is severely discontinuous, imbalance in a sparse training set suggests there is a strong likelihood that more high-performance designs exist near the ones identified in the initial sample. For this reason, it is desirable to add samples and consequently improve sample resolution near decision boundaries. For a BNC, the decision boundary is called the posterior class discriminant (PCD) and defined as

$$\begin{aligned} PCD &= \lambda_1 P(c_1|\vec{x}) - \lambda_0 P(c_0|\vec{x}) \\ &= \frac{\lambda_1 P\left(\vec{x}\middle|c_1\right)P(c_1) - \lambda_0 P\left(\vec{x}\middle|c_0\right)P(c_0)}{P\left(\vec{x}\middle|c_1\right)P(c_1) + P\left(\vec{x}\middle|c_0\right)P(c_0)} \end{aligned} \tag{4.6}$$

as presented in Section 3.2.3. Where $PCD \in [-1,1]$ and $PCD = 0$ indicates the boundary between regions of differing class membership. Because the BNC is a generative classifier, an interval of PCD values can be specified to indicate specific regions around the decision boundary to add synthetic samples. The following section describes the PCD-informed SMOTE methodology in detail. This method improves the classifier's ability to identify true positives and reduce the number of positives that are misclassified as false negatives. The metric for measuring success of this goal is increasing TPR while minimizing adverse effects such as an increase in FPR. This method is particularly advantageous in sparse

and/or imbalanced design spaces where the BNC model underestimates the size of high-performance regions in the design space.

4.4.1 PCD-informed SMOTE Algorithm

To improve the overall classifier performance in a sparse, imbalanced design space we seek to add synthetic samples to the design space mapping near the decision boundaries ($PCD = 0$). As shown in Figure 4.2, the procedure starts with a sampling strategy and an initial design space mapping. Following a standard *k*-fold cross validation (CV) procedure, the BNC classifier is trained using a training set of candidate designs with known performance. Then, the accuracy of the BNC is evaluated with a separate set of test data (also with known performance evaluated from the expensive or black-box model). The BNC classifies the design space into high- and low-performance regions according to performance thresholds specified by the designer. If the high-performance designs are represented sparsely in the training data, and the TPR is unacceptably low, the BNC is a candidate for synthetic sampling. Figure 4.2a illustrates a simplified 2D design space with one discrete variable (x_2) and one continuous variable (x_1) and sparse representation of high-performance designs, shown as green points in the figure. This BNC is a candidate for a synthetic sampling procedure because its TPR is unacceptably low, as indicated by the substantial proportion of high-performance points outside of the decision boundary, which is represented by the solid black line in Figure 4.2a.

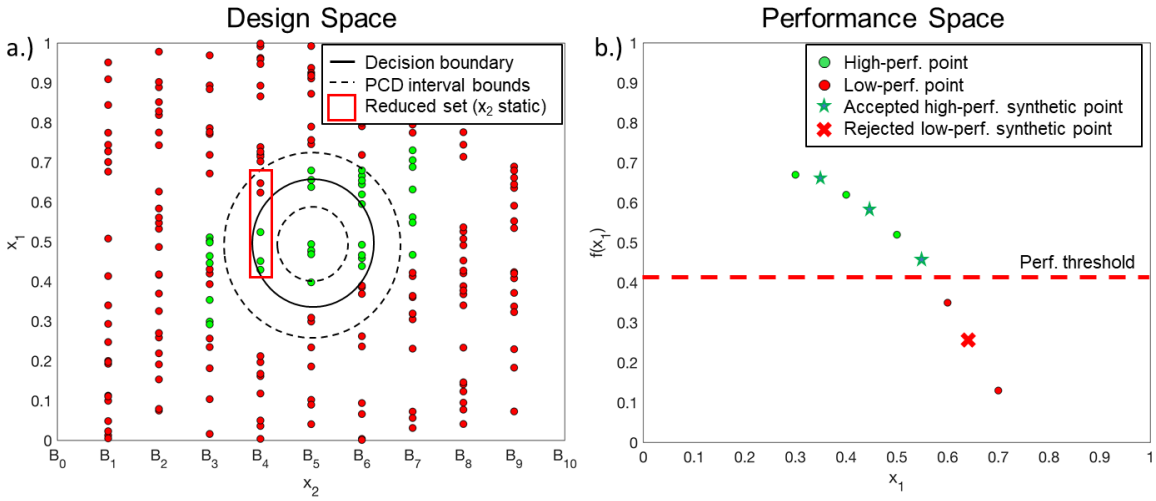

Figure 4.2: (a) A 2D design space with one discrete (x_2) and one continuous variable (x_1). The high-performance region is bounded by the black decision boundary derived from the KBN. As shown, several high-performance (green) points are incorrectly classified as low-performance (outside of the decision boundary), contributing to an undesirably low TPR, so a PCD interval, indicated by the dashed lines, is generated to identify basis points for synthetic sampling. (b) A linear interpolation scheme is used to generate synthetic points for the reduced set outlined by the red box in (a). If the interpolation indicates that the point is high performance, it is added to the training set as a synthetic point, as indicated by the green stars. Otherwise, it is rejected as a candidate synthetic point, as indicated by the red X.

The PCD-informed SMOTE method operates by adding synthetic training points near the decision boundaries to improve the accuracy of the classifier. The procedure begins by identifying candidate designs near a decision boundary, where $PCD = 0$. By specifying an interval of PCD values, the designer selects the design points that are suitable basis points for synthetic sampling. For example, the interval could be specified as $PCD \in [-0.2, 0.2]$ to utilize 20% of the space on either side of the decision boundary. In Figure 4.2a, the decision boundary is represented by the solid black line, and a small interval around the PCD is represented by the dotted black line. Any points within the PCD interval are designated as suitable basis points for synthetic sampling. The size of the PCD interval

is treated as a heuristic or hyperparameter and determines the extent of the design space that is utilized for synthetic sampling. An interval of $PCD \in [-1, 1]$ would encompass the entire space.

With an assigned PCD interval, the next step is to interpolate between the basis points. In this case, one of the variables, x_2, is a discrete variable, which is not amenable to interpolation. Accordingly, an initial set of basis points is selected based on a common value for the discrete variable, x_2, as indicated by the red box in Figure 4.2a. Then, synthetic points are generated by interpolating between the basis points to estimate the response of a candidate synthetic point using a regression or metamodeling technique. In Figure 4.2b, the performance response, $f(x_1)$, is plotted as a function of the continuous variable, x_1. The value of the continuous variable is adjusted to generate candidate synthetic points. The performance, $f(x_1)$, of the candidate synthetic points is evaluated by interpolation between the basis points of known performance. In the example case described here, interpolation is performed via simple linear regression between neighboring points, but any metamodeling method (non-linear regression, kriging, etc.) could be utilized to perform the interpolation. If the interpolated performance of the candidate synthetic point exceeds the performance threshold specified by the designer for the purposes of classification, it is accepted as a synthetic point to be added to the training set, as represented by one of the green stars in Figure 4.2b. If not, it is rejected, as represented by the red X in Figure 4.2b. Then, the process is repeated for all unique values of the discrete variable(s) until all of the basis points within the PCD interval have been considered. At least one continuous variable must be present in the design space. While this illustrative example includes only one continuous variable, multiple continuous variables can be accommodated by the metamodel.

As shown in Figure 4.3, the synthetic training data is merged with the original training data to form a new training data set, and a synthetically enhanced BNC is trained. The accuracy of the synthetically enhanced BNC is evaluated with the same test data utilized to evaluate the accuracy of the original BNC. If the accuracy is still unacceptable, the process may be repeated iteratively.

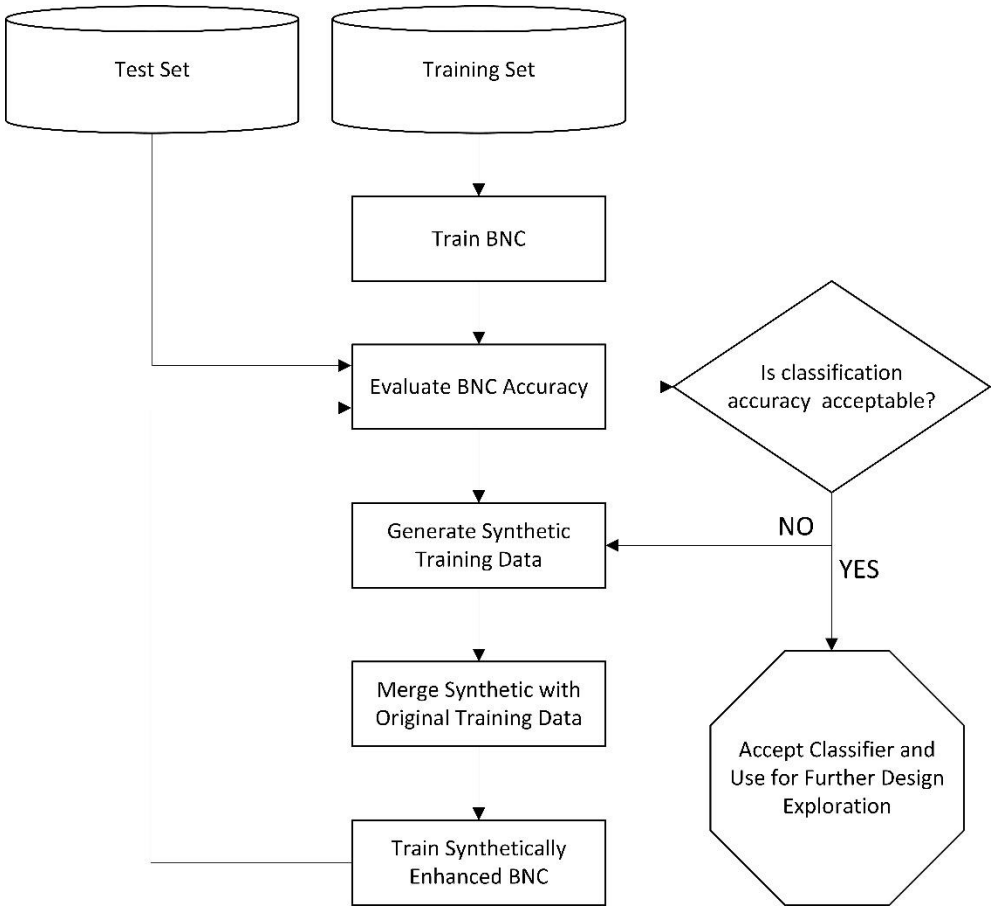

Figure 4.3: Flowchart outlining the strategy of PCD informed SMOTE.

When applying this method to a design problem, it is good practice to select a subset of synthetic points to validate with the underlying expensive model. Although using metamodels to interpolate between basis points is intended to reduce computational

expense, validating a subset of synthetic points helps ensure the accuracy of the synthetically enhanced design space. The appropriate size of the validation subset primarily depends on the accuracy of the metamodel. For example, if a linear regression model is used to generate and evaluate synthetic points that exhibit a highly multimodal response with few basis points, the performance prediction of the synthetic design points could deviate significantly from the ground-truth performance gathered from the underlying expensive model. If that were the case, training the synthetically enhanced model with a CV scheme should indicate it performs poorly and would suggest that the chosen metamodel may not be appropriate. Regardless, validating a subset of synthetic samples reduces the risk of accepting an erroneous synthetically enhanced model. A validation step is performed in the demonstration problem in the next section.

4.4.2 Design of Two-layer Willis Material with PCD-informed SMOTE

To investigate the effectiveness of the PCD informed SMOTE procedure, we consider the task of identifying acoustic non-reciprocity in a two-layer Willis material AMM. The primitive design of the AMM and heterogeneity are described in detail in Chapter 2 so this section focuses on the design of this AMM to a minimum specified acoustic non-reciprocity. Because any two materials may be used in the heterogeneity and the frequency band where non-reciprocal responses may occur is very large, the design space is vast. Finite element analysis is used to evaluate the non-reciprocal performance of prospective designs. Because of the quantity of possible feature combinations and computational expense of FEA, exhaustive exploration of this design space is infeasible. However, the design space can be mapped by limiting the scope of the investigation. The problem is well suited to set-based design and classification of high- and low- performance designs. Additionally, it provides a challenging case of class imbalance due to the nearly

infinite number of candidate designs and the relatively small fraction of those designs that meet reasonable high-performance thresholds.

Data Generation

There are six features stated in Section 2.2 that are considered as design variables for the two-layer Willis material and four performance indicators. Even and odd Willis coupling are the key performance indicators while the impedance ratio between the heterogeneity and background material and the effective wavenumber inside the heterogeneity are used as constraints. All the variables and performance indicators are defined and organized in Table 4.1.

Design Variables		Performance Indicators	
Density of Layer 1	ρ_1	Normalized Even Coupling	$c_o\chi^e/k_0L$
Density of Layer 2	ρ_2	Normalized Odd Coupling	$c_o\chi^o/k_0LkL$
Sound Speed in Layer 1	c_1	Ratio of Effective Impedance to Background Impedance	$\frac{Re(Z_{eff})}{Z_0}$
Sound Speed in Layer 2	c_2		
Normalized Wavenumber in the Fluid	$k_0L = \frac{2\pi fL}{c_0}$	Effective Normalized Wavenumber	kL
Volume Fraction of Inhom.	$VF = \frac{l}{L}$		

Table 4.1: Key terms for use in design of a two-layer Willis material. The design space is six-dimensional and there are four important performance indicators.

Sampling was restricted to consider 324 possible combinations of 18 common materials (i.e. steel, rubber, glass, lead, etc.) of varying properties but equal layer thickness in a background of liquid water. A Halton sequence was used to uniformly sample $VF \in (0.1,\ 0.35)$ at 5 points and $f \in [500Hz,\ 50{,}000Hz]$ at 100 points in increments of 500

Hz. Sieck *et al*., the authors of the paper "Origins of Willis Coupling…" [27], suggested starting with these intervals for VF and f to find useful results [27]. With these sampling increments, the data set includes 162,000 samples.

To meet performance constraints, the effective impedance of the composite, $Re(Z_{eff})$, was constrained to 80%-120% that of water, and the effective wavenumber, kL, was constrained to be less than π, which is considered the upper limit for the dynamic homogenization scheme to produce valid results. After removing all samples violating these constraints, the data set contained 6,750 samples in a 6D design space and was prepared for classification using BNCs. Gathering the data set with $N = 6750$ took approximately 17 hours on a Linux machine with two 12-core Intel Xeon Gold 5118 CPUs and 250 GB of RAM. A reasonable coupling performance threshold was selected to distinguish between high- and low-performance classes. By classifying any sample with $|c_o\chi^e/k_0L| > 0.02$ as high-performance, 10% of samples were classified as high-performance and 90% of samples as low-performance. For a visualization of the 6D design space, the material properties were combined by calculating the difference in characteristic acoustic impedance between the 2 layers of the inhomogeneity $Z_1 - Z_2 = \rho_1 c_1 - \rho_2 c_2$. Figure 4.4 shows this reduced design space in a 3D scatter plot with a color map indicating normalized even coupling, $c_o\chi^e/k_0L$.

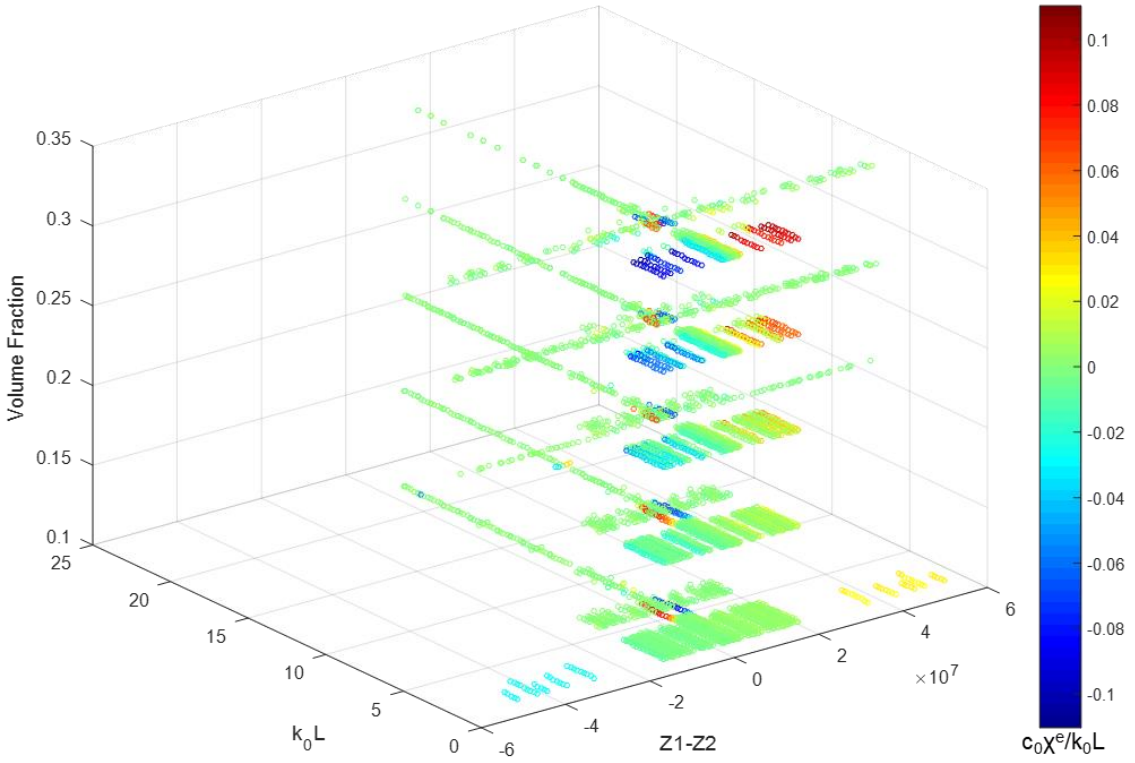

Figure 4.4: All sample points used for the demonstration problem scattered in a design space reduced to 3D by consolidating the material properties into a difference of characteristic acoustic impedances ($Z_1 - Z_2$). All points in this set have effective impedances within 80%-120% of water and effective wavenumbers less than π. The color map represents normalized even coupling values, with absolute values greater than 0.02 indicating high-performance.

Figure 4.5 shows some examples of the normalized even coupling response as a function of the normalized wave number, k_0L, for unique material combinations. The relationships are a subset of mostly high-performance samples that are later used for synthetic sample generation in the synthetically enhanced BNC model.

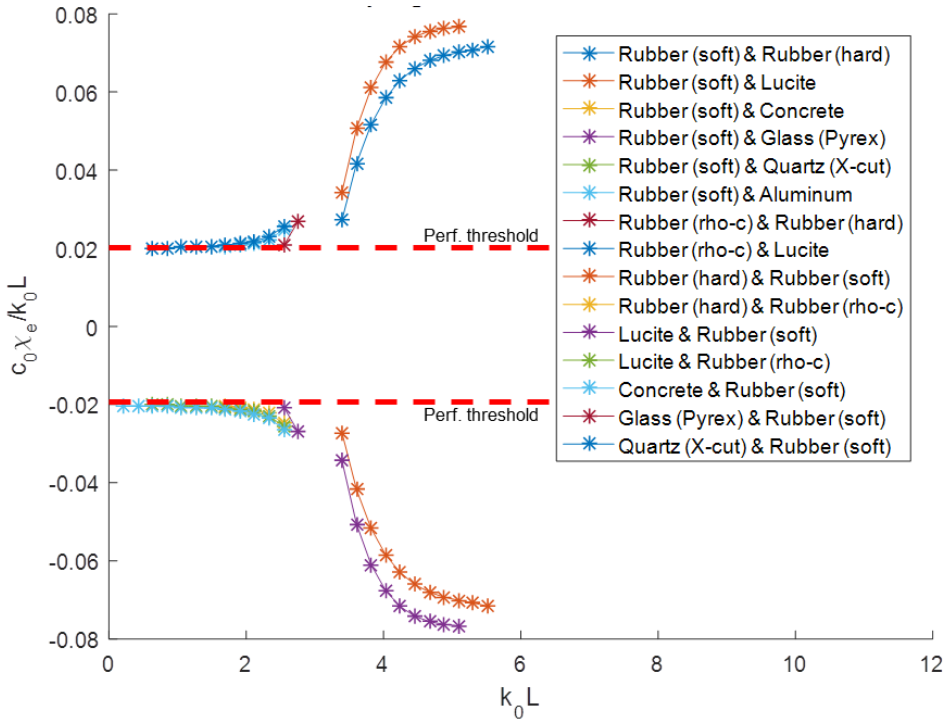

Figure 4.5: Normalized even coupling values as a function of normalized wave number, for 15 unique combinations of materials at a volume fraction of 17%. All responses have effective wavenumber less than $\boldsymbol{\pi}$ and effective impedance within 80%-120% of water.

This entire data set with $N = 6750$ was used to train and cross-validate a naïve Bayesian network classifier both with and without the PCD informed SMOTE method presented in this work.

Training the Classifier

The BNC described in Section 3.2.3 used a Gaussian kernel and was trained and tuned by three-fold cross-validation. Classifier performance is evaluated using the three scoring metrics: ACC, TPR, and FPR. The Gaussian kernel parameter is expressed as

$$\sigma_{i,l} = \frac{\alpha\hat{\sigma}_{i,l}}{N_l^{1/D}}, \tag{4.7}$$

with:

$\sigma_{i,l}$: kernel width parameter (Equation 3.17),

α : heuristic scalar,

$\hat{\sigma}_{i,l}$: st. dev. of design var. i for designs belonging to class l,

N_l : number of samples in class l,

D : number of design vars. (dimensions).

Performance of the classifier was evaluated with the sole hyperparameter, α, set as each member of a linear sequence from 0.01 to 0.5 and 0.001 was included to investigate the effect of a very thin kernel width. The full sequence is given as $\alpha = [0.001, 0.01, 0.064, 0.119, 0.173, 0.228, 0.282, 0.337, 0.391, 0.446, 0.500]$.

The synthetically enhanced model was tuned and compared against the base BNC model by using an identical set of data and with synthetic points added only to the training set. Synthetic points were generated by linearly interpolating along the $c_o\chi^e/k_0L$ vs. k_0L response curves (e.g. Figure 4.5) for every unique combination of heterogeneous materials in the data set using the exact methodology developed in Section 4.4.1. Linear interpolation was used so that the performance of this method can be compared more easily to other SMOTE methods that do the same. The SMOTE method performance comparison is described in Section 4.5.1. Hyperparameters for the PCD-informed SMOTE are: α, PCD interval, and the quantity of synthetic points added between real samples (interpolation layers). Early tuning showed three α settings from the base BNC training yielded the best performance for the synthetically enhanced model so the α space was reduced for efficiency. As a result, a hyperparameter space with 240 unique combinations was evaluated for the synthetically enhanced model. Settings for each hyperparameter of the BNC with PCD-informed SMOTE model are organized in Table 4.2.

Hyperparameter	Settings

Interpolation Layers	[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
PCD Interval	±[0.1, 0.2, 0.5, 0.6, 0.8, 0.9, 0.95, 0.99]
α	[0.391, 0.446, 0.500]

Table 4.2: Hyperparameters for the PCD-informed SMOTE enhanced classifier

Training and testing each classifier model variation takes less than 30 seconds on a Windows machine with an Intel i7-8700 CPU and 32 GB of RAM. The performance of the base BNC and synthetically enhanced BNC are compared in the results section that follows.

4.5 PCD-INFORMED SMOTE TO IMPROVE BNC TRAINING RESULTS

The base BNC model showed the best performance with thin kernels (e.g. at $\alpha = 0.01$). Table 4.3 below summarizes the performance of the base model. Regardless of the α setting, the TPR never rose above about 38%, indicating poor performance identifying the high-performance regions of the design space. Three classifier scoring metrics are used to track potential adverse effects of increasing TPR within the space. For this base model, FPR increased with TPR except for the very thin kernels.

α	TPR	FPR	ACC
0.001	0.379	0.011	0.922
0.010	0.381	0.011	0.923
0.064	0.349	0.014	0.916
0.119	0.206	0.013	0.902
0.173	0.119	0.011	0.894
0.228	0.071	0.011	0.889
0.282	0.035	0.010	0.885
0.337	0.020	0.010	0.884
0.391	0.008	0.008	0.884
0.446	0.001	0.008	0.884
0.500	0.003	0.007	0.885

Table 4.3: Results of tuning the base BNC model by cross-validation with varying hyperparameter α. The green highlighted hyperparameter settings in Table 4.2 indicate the values that lead to strong performance for the synthetically enhanced model.

The BNCs that were enhanced with PCD-informed SMOTE showed a very significant increase in TPR and some increase in FPR. Figure 4.6 shows the achievable trade-off between FPR and TPR graphically for all 240 hyperparameter combinations, as well as the best base BNC models highlighted in red. Detailed performance of 10 synthetically enhanced models with high TPR and the lowest associated FPR is shown in Table 4.4. It is clear that FPR rises sharply for models achieving a TPR greater than 70%. Ideally, TPR is maximized while FPR is minimized, so the best models are considered to be those nearest the top-left corner of Figure 4.6. This trade-off relationship is analogous to a Pareto front in multi-objective optimization [90].

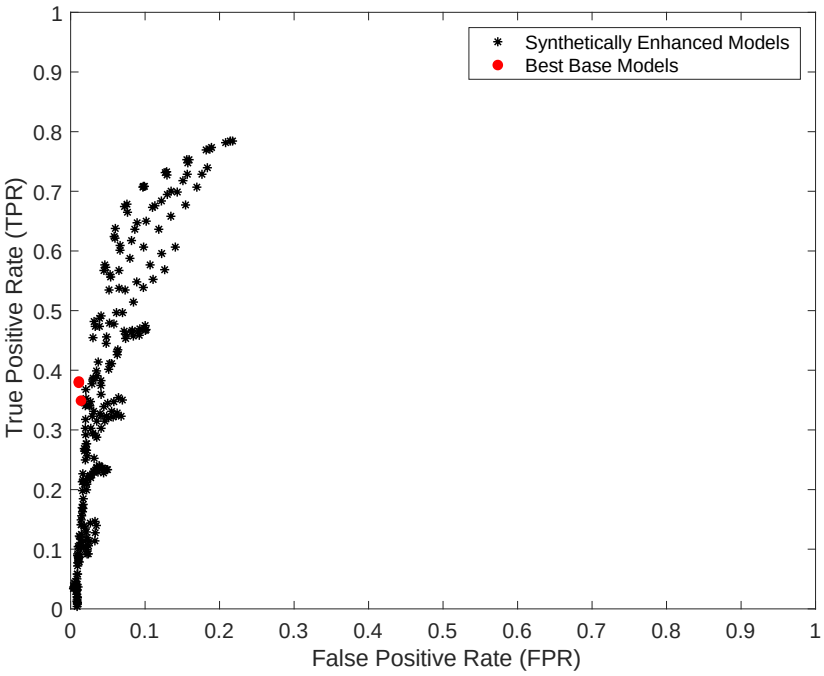

Figure 4.6: Results of tuning the synthetically enhanced model for the two-layer Willis material design. Selecting the optimal model requires a designer to determine an appropriate trade-off between increasing TPR and FPR.

Interpolation Layers	PCD Interval	α	TPR	FPR	ACC
6	0.99	0.500	0.624	0.059	0.907
6	0.99	0.446	0.638	0.060	0.907
7	0.99	0.391	0.665	0.076	0.895
7	0.99	0.500	0.674	0.072	0.900
7	0.99	0.446	0.678	0.076	0.897
8	0.99	0.500	0.708	0.097	0.881
8	0.99	0.446	0.708	0.099	0.880
9	0.99	0.391	0.731	0.127	0.858
9	0.99	0.446	0.733	0.129	0.856
10	0.99	0.391	0.753	0.156	0.834

Table 4.4: Ten best performing BNCs enhanced by PCD-informed SMOTE sorted by TPR.

Increasing FPR is an undesirable result that comes along with expanding the high-performance regions of the design space mapping, but it is difficult to avoid. The ACC is slightly lower for the synthetically enhanced models than the best base model, which means that more false positives have been introduced than false negatives have been removed. However, in the interest of identifying regions that hold minority high-performance class members in design exploration, false positives are of lesser consequence than false negatives. With a higher TPR rate, this classifier is now much more useful for exploring a design space. Note that the large PCD intervals work well for this problem. This is likely a result of two things: smooth relationships between even coupling and wavenumber within each unique material combination and rejection of synthetic points that would belong to the majority class according to the metamodel-predicted coupling response. In a design space with a highly non-linear and multimodal mapping, a thinner PCD interval may be necessary. Overall the performance of this classifier has been improved noticeably by adding synthetic points around the decision boundaries.

Synthetic Sample Validation

A validation step compared the interpolated performance of the synthetic points to the results of the FEA. Plots of $c_o\chi^e/k_0L$ vs. k_0L showed mostly smooth functional relationships that were suitable for linear interpolation, but to be thorough, we evaluated 2811 synthetic design points. Simple but representative hyperparameters were chosen: $Interpolation\ Layers = 1, PCD\ Interval = [-1, 1]$, and $\alpha = 0.5$, along with a performance threshold of $|c_o\chi^e/k_0L| > 0.02$. The mean squared error (MSE) between the interpolated and simulated $c_o\chi^e/k_0L$ values of the synthetic points was $7.4x10^{-6}$. For reference, $c_o\chi^e/k_0L \in [-0.1, 0.1]$. This low error rate demonstrated that the interpolated performance values were very close to the result we would have gotten by incurring the computational expense to evaluate them all with FEA. In this case, it required about 3.5 minutes to evaluate each design point, so evaluating all 2811 points required approximately 164 hours on a PC with an Intel i5 processor and 16 GB of RAM. For comparison, generating the synthetic points on the same machine takes just a few seconds.

4.5.1 Comparison to other SMOTE techniques

PCD-informed SMOTE improves design space exploration for engineering design tasks that have a performance response that is functionally related to at least one continuous design variable. Accuracy of the design space mapping can be increased efficiently by exploiting this relationship when oversampling with synthetic design points. The methodology presented here performs well when compared with other leading SMOTE methods. To benchmark this method, SMOTE, ADASYN, and borderline-SMOTE were also used to enhance the BNC classifier. The best instantiations of each technique maximized TPR more than PCD-informed SMOTE but also showed more bias towards positive classification in general. They increased FPR and reduced ACC more than PCD-

informed SMOTE. Classification performance when using each method to enhance the BNC is displayed in Table 4.5.

Method	α	SMOTE			Borderline-SMOTE			ADASYN		
Score		TPR	FPR	ACC	TPR	FPR	ACC	TPR	FPR	ACC
	0.322	0.957	0.166	0.895	0.962	0.225	0.868	0.952	0.208	0.872
	0.367	0.961	0.176	0.893	0.961	0.232	0.864	0.952	0.214	0.869
	0.411	0.969	0.185	0.892	0.962	0.237	0.863	0.953	0.214	0.869
	0.456	0.971	0.196	0.888	0.964	0.236	0.864	0.954	0.221	0.867
	0.5	0.972	0.204	0.884	0.965	0.237	0.864	0.954	0.223	0.865

Table 4.5: Performance of the three leading SMOTE implementations. Each SMOTE algorithm is used to oversample the design space for the two-layer Willis material design task using the same dataset that was used for PCD-informed SMOTE.

The comparison is only for a single design task so no definitive conclusions can be made about the general effectiveness of PCD-informed SMOTE as compared to other leading techniques but at very least, this method is competitive with those leading methods. Because more information is utilized to place synthetic design points, it could be easily expected that PCD-informed SMOTE would perform more favorably in a design space with an even more challenging response topology.

4.5.2 Results Discussion

Despite the challenging characteristics of the design space, adding synthetic minority class samples to the training set improved the classifier's performance by increasing TPR at a significantly greater rate than the FPR when hyperparameters were well tuned. An ideal synthetic oversampling would not increase FPR at all, but since the

synthetic points are enlarging the high-performance regions of the design map, it is expected that some low-performance designs will lie in those regions and be misclassified during the cross-validation.

One obvious opportunity to expand this work is to apply the method to additional design tasks and fully benchmark it against other under- and over-sampling techniques. Another opportunity is to study the relationship between the PCD bounds and design space characteristics such as sparsity and their effect on the synthetic generation process. It would also be worth testing more sophisticated metamodeling techniques for generating synthetic samples from the exploitable continuous variable relationship. For example, prior knowledge of the response or transfer learning from similar problems could be incorporated to add realistic synthetic points in areas of the space with little definition.

In general, validating the classified performance of synthetic points is of great interest for improving the accuracy of the enhanced classifier, but it must be done in a cost-effective way. There is potential for developing a confidence-based algorithmic scheme to select a validation subset from the synthetic points to optimize validation expense. For example, a sampling scheme could be used to evaluate synthetic points while the synthetic sample is being generated, perhaps using an expected improvement framework to determine which synthetic points are most valuable to validate with the underlying expensive model.

Chapter 5: AM-aware Design of Asymmetric Acoustic Absorbers[2]

Materials that absorb sound waves asymmetrically have long been desired for a variety of acoustic applications. A material that exhibits asymmetric absorption of acoustic fields dissipates energy of an incident sound wave differently depending on the propagation direction. The AMS with asymmetric ellipsoidal inclusions presented in Section 2.3 has been shown by initial experiments to be a promising way to achieve this behavior; it is the focus of this chapter. Multi-material additive manufacturing (AM) processes enable fabrication of these types of metamaterials because they provide a greater variety of options for material and geometry than traditional manufacturing processes. However, these processes place important limits on the resolution of features and introduce variability in the material properties and geometries of fabricated parts. In this chapter a metamodel-driven framework is created to efficiently perform uncertainty analysis and enable robust design of asymmetric absorbers such that they can be reliably manufactured using multi-material AM processes. The method evaluates the probability of building an AMS to specifications given any particular AM machine with its associated accuracy. It also allows exploration of different AM machine accuracy characteristics with negligible additional computational expense. As a result, this method facilitates the search for suitable AM machines and helps a designer identify and propose machine accuracy targets to help guide AM process development.

[2] The majority of the content in Chapter 5 has been previously published in [256] and the primary author of that paper is also the author of this dissertation. Some additional background has been added for the dissertation. Carolyn Seepersad and Michael Haberman contributed to problem conception, supervised the work, and revised the manuscript. AJ Lawrence provided a template finite element model which was adapted by the author to generate data as described in Section 5.4.2. The Zhang laboratory at the University of California – Berkeley built the handmade specimen shown in Figure 5.1. The text of this chapter has been written by the author and edited by both Michael Haberman and Carolyn Seepersad.

Asymmetric absorption has been shown experimentally in a proof-of-concept metamaterial structure that is very similar to the ellipsoidal inclusion AMS, as described in a forthcoming journal article by AJ Lawrence and Michael Haberman. The specimen was fabricated by hand with $d = 1mm$ inclusions encased in a matrix of PDMS. The inclusions contain iron (Fe) and polyethylene glycol (PEG) roughly separated such that the material interface is normal to the direction of incident wave propagation. Unlike the AMS design utilized in this work, the inclusion layer is multiple inclusions deep with varying thickness across the inclusion layer. Figure 5.1 is a photo of the specimen.

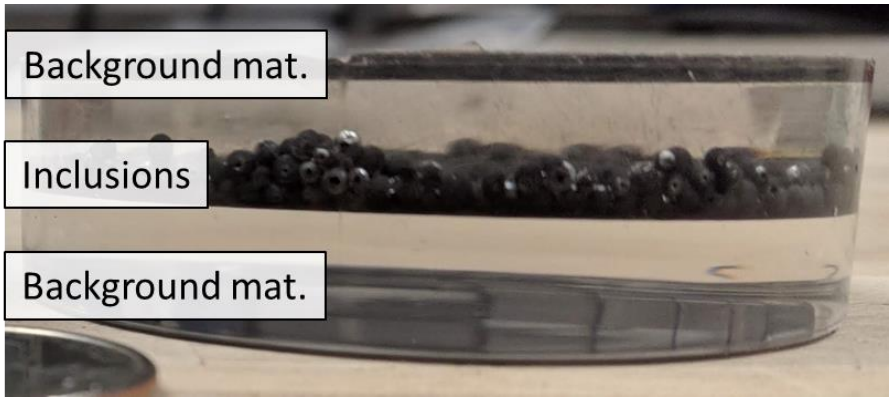

Figure 5.1: Proof-of-concept sample of an asymmetrically absorbing AMM. Inclusions are made by solidifying droplets of an iron-PEG slurry. The inclusions are spread into a layer on a solid matrix of PDMS and then encased in PDMS by curing more PDMS resin on top of the inclusions. Post-fabrication, inclusion materials are separated and aligned by melting the PEG and spinning the AMM in a centrifuge to move the denser iron particles to one side.

While this experiment proved the concept of fabricating an asymmetric absorber, significant design changes are necessary to build them to specific engineering specifications, and the multi-material constitution and size of the inclusions make accurate fabrication challenging.

This chapter presents a manufacturing-aware design methodology for an asymmetrically absorbing metamaterial with a planar layer of multi-material ellipsoidal inclusions. By designing for robust as-built performance, the expected absorptive response of an additively manufactured AMS can be evaluated against inherent process variability, which is of fundamental importance in realizing metamaterial concepts for real-world applications. Furthermore, from the results of the robust design process, a manufacturer gains the information required to appropriately tune—or even develop—an AM machine to meet manufacturing requirements for future AMM and AMS designs. In this work, the robust design process is used to ensure as-manufactured success of a class of asymmetrically absorbing AMS, but is practically extensible to any other AMM or AMS when considering manufacturing variability.

5.1 AM METHODS FOR FABRICATING AMMS

Although there has been considerable theoretical work on the topic of asymmetrically absorbing materials and some proof-of-concept experimental demonstrations, pushing the boundaries of AMM behavior requires manufacturing methods that can produce the novel material arrangements conceived by engineers. In practice, manufacturing capabilities often restrict the way that material arrangements can be realized. To this end, the ongoing development of AM is a paradigm shift that enables the realization of metamaterials that were previously impossible to create [168]. AM enables material to be patterned with greater control and for complex features to be built at small scales. To achieve predictable behavior of complex metamaterial designs, the manufacturing method must be able to fabricate them reliably. For a passive asymmetrically absorbing metamaterial, the behavior is sensitive to geometric variation

because the physical response relies on resonating inclusions to affect the scattered fields [169] [28] [29].

A robust design method for AM must account for manufacturing-induced stochastic variations in geometry and the resulting deviations from target performance of the AMS. By studying the sensitivity of AMS performance to manufacturing variation, the development of a prospective process can directly target the accuracy metrics required to produce an AMS with as-built performance that meets design requirements. Quantifying those metrics, in turn, reduces the need for experimental iterations in process development. The true performance of an as-built AMM is expected to deviate from the nominal performance depending on the sensitivity of the performance to error (manufacturing variability) associated with the as-built inclusion geometry and the magnitude of that error. To quantify performance of the AMS, we express the acoustic response as a stochastic function, $g(\boldsymbol{x}, f)$, having an error component represented as a random variable, $\epsilon(\boldsymbol{z}, \boldsymbol{T})$:

$$g(\boldsymbol{x}, f) = L(\boldsymbol{x}, f) + \epsilon(\boldsymbol{z}, \boldsymbol{T}) \tag{5.1}$$

with:

$\boldsymbol{x}$: design features,

f: frequency,

L: deterministic response,

ϵ: error,

$\boldsymbol{T} \in \boldsymbol{\Omega}$: AM process random variable set,

$\boldsymbol{z} \in \mathbb{R}^3$: 3D position in build chamber.

As defined, the error component depends on the interaction between a process-specific set of random variables, $\boldsymbol{T}$, as well as the position, $\boldsymbol{z}$, in the physical space of the build chamber. Spatial error manifests itself as inaccuracy of size, shape, and location of each material and the interfaces between them. There is an abundance of literature

attempting to quantify error for specific AM processes [170] [171] [172] [173], as well as translating process error to as-built part inaccuracy [174] [175] [176] [177]. Regardless of the source of build accuracy information, a proper robust design workflow must be able to utilize it.

Once the as-built part variability is quantified, the next step is to translate it into expected variation in performance and resulting manufacturability requirements. For the present study, the variation in performance is quantified in terms of the degree of asymmetry of acoustic absorption, $\bar{\alpha} = {}^{\alpha_2}/_{\alpha_1}$. This type of analysis may be performed using a Monte Carlo (MC) approach, but MC methods can be prohibitively expensive when the underlying FEA is computationally expensive. This is especially true when the MC analysis must be repeated iteratively as part of an optimization process. In this work, the computational expense is reduced by replacing the FEA with a classifier-based metamodel of the FEA. The modeling and design for AM are described in detail in Section 5.4.

5.2 MULTI-MATERIAL AM PROCESSES

This work seeks to prescribe the limits on fabrication accuracy for the creation of the Willis-coupled AMS displaying asymmetric absorption detailed in Chapter 2. The accuracy of the manufacturing method and the sensitivity of AMS performance must be considered during the design process in order to be certain that an as-built AMS will meet the as-designed performance criteria. Multi-material AM is an obvious choice for the creation of this type of AMS because of the small size scale as well as the unique material and geometric challenges of building heterogeneous inclusions to specification [178]. Multi-material AM processes enable a wide variety of materials and geometries to be explored to create an AMS demonstrating asymmetric absorption. This work considers combinations of material choices and manufacturing accuracies that are not currently

achievable to demonstrate how robust design can be used to help guide process development toward an AM process that is capable of fabricating similar AMMs and AMSs.

Existing literature on multi-material AM offers a few potential approaches. Direct ink writing is capable of patterning material at the sub-millimeter scale [179], and microfluidic devices are able to deposit inclusions as small as $200\mu m$ in a direct ink writing process [180], but depositing multiple materials within each inclusion is an outstanding challenge. Ultrasonic [181] and magnetic [182] fields can be used to align and arrange inclusions but have not been applied to multi-material inclusions. Aerosol deposition methods can create the type of material interfaces required [183] and have achieved deposition of silver in lines as narrow as $10\mu m$ wide [184] and with very thin layers (on the order of $1\mu m$) [183], so layering enough material for the inclusions in question may be prohibitively time-consuming. Since none of these methods have yet been used to create an AMM or AMS with the architected features motivated by the asymmetric absorber, a robust design approach is sought to help specify the requirements of a suitable manufacturing process.

5.3 Performance Sensitivity and Spatial Variability

The expected variation in geometry or material properties induced by the manufacturing process must be considered during the design process. By doing so, the design task becomes an exercise in manufacturing-aware robust design. Robust design processes seek to identify designs that exhibit high levels of performance despite underlying variation in the design itself and/or the environment in which it functions [175] [185] [186]. These approaches account for randomness in subsystems or manufacturing processes as an integral part of the design process; deterministic design optimization is

insufficient because it does not account for variable production and operating conditions. Utilizing robust design is especially important for design tasks where the effect of variations among fabricated components is significant with respect to expected performance. Robust topology optimization techniques have also been developed to incorporate stochastic variation in material properties and geometry into a topology optimization process [174] [187] [188], but they require the underlying analysis to be amenable to gradient-based topology optimization formulations. The approach presented in this chapter is more general and broadly applicable than topology optimization approaches. It makes use of a unique classification procedure to identify high performance designs and then to identify the subset of those designs that meet performance targets reliably even with manufacturing-induced variability taken into account.

5.3.1 Approximately Planar Scattered Fields

As part of behaving like a metamaterial, the resultant scattered fields from a planar incident field must also be planar. A planar scattered field would be expected from a conventional bulk material in the absence of a metasurface layer. Furthermore, planar fields allow absorption to be expressed in a pseudo-one-dimensional form despite the AMS geometry containing features in three dimensions. This assumption is valid in the metamaterial paradigm where characteristic inclusion sizes, a, and inter-inclusion spacing are subwavelength, i.e. $a \ll \lambda$ and $d_p \ll \lambda$, and the scattered field is probed multiple wavelengths from the scattering surface such that evanescent fields are extinguished. Absorption asymmetry can then be characterized with the ratio

$$\bar{\alpha} = \frac{\alpha_2}{\alpha_1} \tag{5.2}$$

which is a measure of the dependence of acoustic absorption with respect to the direction of incidence. These structures exhibit reciprocal transmission but not reflection, so

$$\bar{R} = \left|\frac{R_2}{R_1}\right| \tag{5.3}$$

will serve as the performance metric for this study. Figures 2.9 and 5.6 show that the scattered field is indeed planar when analyzed at an appropriately low ka value. This scattered wave behavior matches expectations for inclusions with $ka \ll 1$, which is a constraint for the AMS being in the metamaterial regime. If the inclusions are larger with respect to incident wavelength, multiple modes are excited, which makes calculating energy absorption much more difficult. While the performance goal is application specific, one obvious goal would be to maximize asymmetry as a function of design features, namely the excitation frequency, the composition of the heterogeneous inclusions, and the distribution of constituent materials within the inclusion domain.

5.3.2 Narrowband asymmetric behavior

Robust design approaches are especially valuable for systems with response behavior that is highly sensitive to variability in either design features or exogenous factors. The AMS considered here demonstrates narrow bands of asymmetric behavior with respect to frequency. For example, the asymmetric response performance of an AMS with spherical silver and polystyrene inclusions is shown in Figure 5.2.

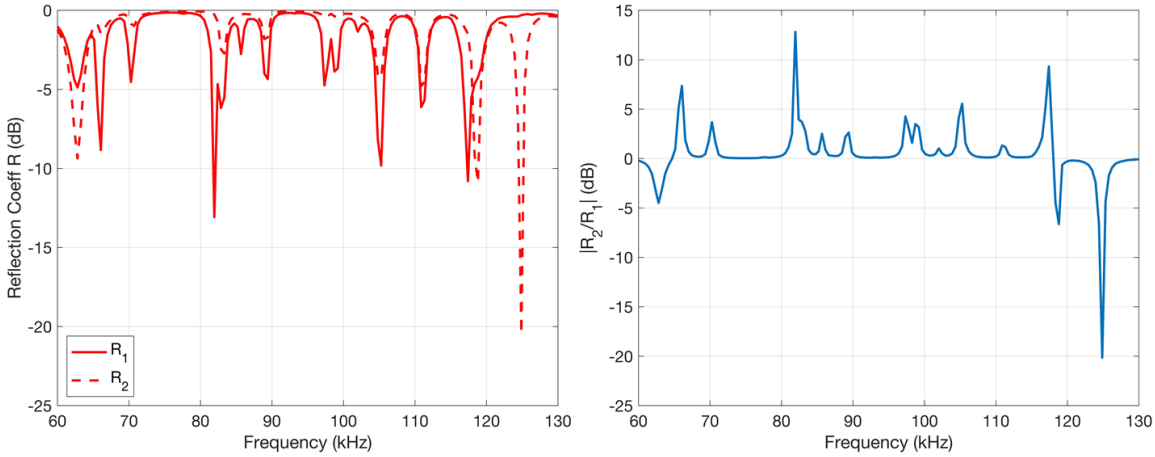

Figure 5.2: Response behavior of the asymmetrically absorbing AMS is narrowband with respect to frequency. The response of the AMS with an inclusion layer made of spherical silver and polystyrene inclusions is plotted as: (left) separate reflection coefficients, R, and (right) the ratio of those coefficients. Both responses are in decibels.

As a result of the narrowband behavior, the design space is somewhat sparse and disjointed. Because the physical behavior is dependent on resonance of the inclusions, the response is also sensitive to inclusion feature variability. The combination of narrowband response and sensitivity to feature variation makes robust design practically necessary when designing this AMS. Sensitivity to feature variability and the effect on the design space is further examined in Section 5.4.2 – *Training Data Generation*.

5.4 AM-AWARE METAMODELING METHODOLOGY

Identifying a robust design and specifying manufacturing requirements for this class of asymmetric absorber requires a deliberate design approach to be computationally feasible without requiring supercomputing capacity. Feasibility is a concern because FEA is necessary to evaluate candidate designs and the potential design space is high-dimensional and vast. Furthermore, consideration of the robustness of any given design to manufacturing variability requires additional sampling near optimally performing designs to quantify the sensitivity of performance to manufacturing variation. In lieu of brute force methods, a progressively refining, coarse-to-fine, design approach is employed here to address this challenge. A visualization of the design process is provided as a flowchart in Figure 5.3. The focus in the present study is on an AMS displaying asymmetric acoustic absorption at a specified design frequency, but the robust design framework presented is applicable for studying the manufacturability of any AMM or AMS that has a quantifiable performance that can be functionally related to geometric features.

Although the design method seeks to minimize computational expense, simulating the behavior of the asymmetric absorber using finite elements is a critical element of the design because the spatial variability of the geometry precludes analytical solutions. A thorough description of the FEA specific to this AMS is provided in Section 2.3.2. The design method uses FEA to gather data to train a classifier-based metamodel that closely replicates the input/output behavior of the FEA. For fabrication processes like AM, which have variable process conditions, estimating the success rate for manufacturing a complex component whose performance is sensitive to geometry variability would require many calls to the FEA, making an uncertainty analysis computationally expensive. Alternatively, a metamodel of the simulation input/response relationship can be queried quickly and for much less computational expense. Metamodels are surrogate models that are trained on data obtained from an underlying physics-based model and approximate the predictions of the underlying model [189] [190] [191]. There are many types of metamodels, and selecting the appropriate one requires information from studies that benchmark techniques based on characteristics of the task [191] [192] [193] or directly comparing a suite of techniques for the task at hand. The metamodel used for asymmetric absorber design is detailed in Section 5.4.2. An accurate metamodel then replaces the FEA for MC simulations to evaluate manufacturability of the asymmetric absorber on a particular machine.

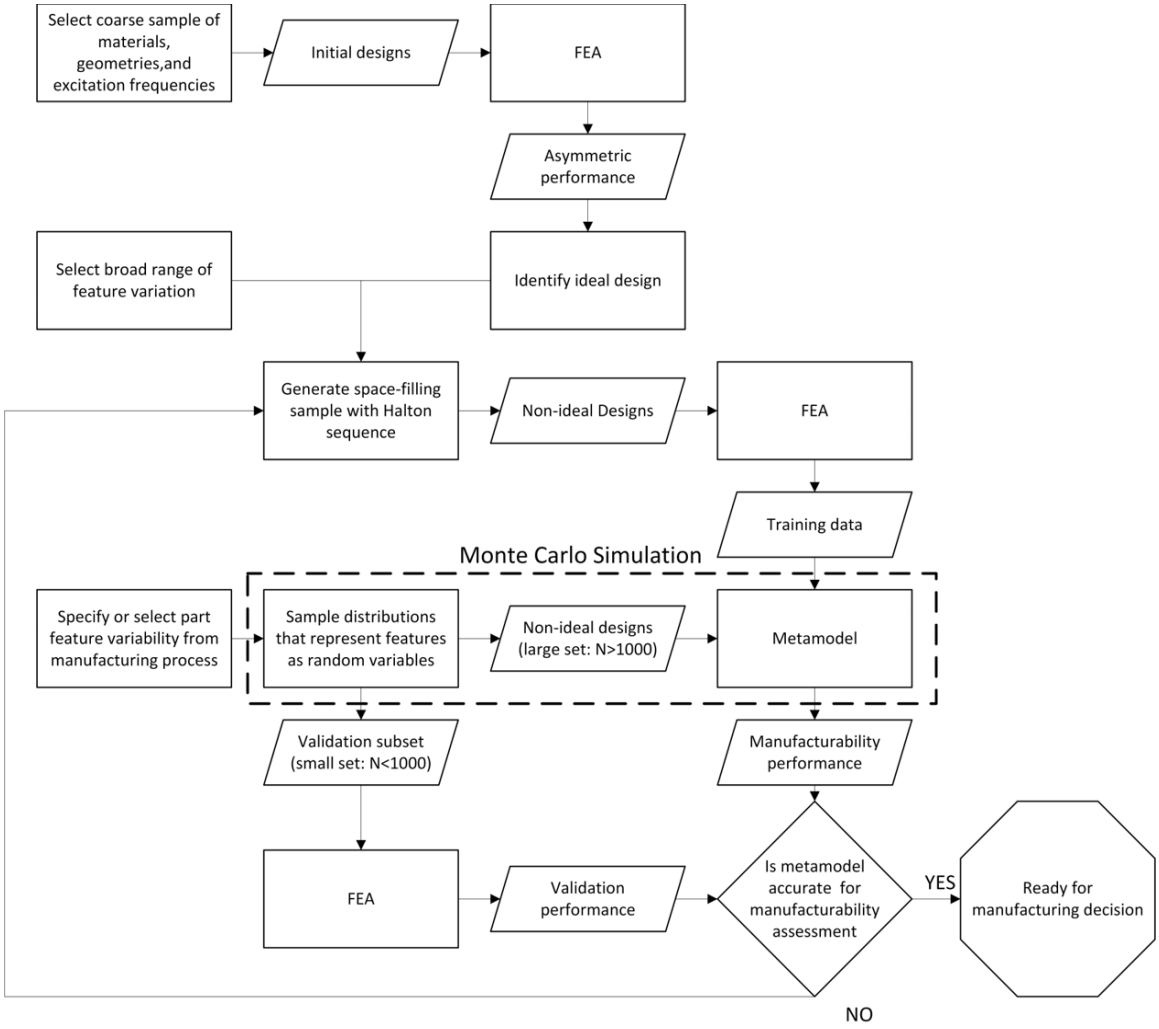

Figure 5.3: Design process for AM of asymmetric absorbers subject to manufacturing variation of part features. The process begins with FEA of initial designs, which are evaluated based on asymmetric absorption to identify ideally performing designs. Designs with feature variation based on a Halton sequence around the ideal design are evaluated to train a metamodel for MC simulations. The MC simulations are a statistical experiment to quantify the number of satisfactory as-built designs that can be expected while trying to manufacture an ideal design.

5.4.1 Target Design Search

The first step of the design methodology is to identify a target design assuming no manufacturing variation. Exploration of the asymmetry with respect to the constraint $ka \ll$

1 is carried out by holding the geometric parameter a constant (for the manufacturing-agnostic design only) with $a = d = 1$ mm and varying frequency in the range of 1 to 160 kHz. Design instances are evaluated by parameterizing all pertinent geometric variables so they may be changed programmatically. The simulations are executed in Comsol Multiphysics with the PARDISO solver [56] [194] as described in Section 2.3.2. Scattered fields are averaged across the probe planes and post-processed to yield reflection, R, transmission, T, and absorption, α. Given all constraints, the manufacturing-agnostic ideal design is chosen from a coarse exploratory sample of feature combinations with asymmetry of reflection, $|R_2/R_1|$, as the performance indicator. All combinations of the features presented in Table 5.1 are included in the initial exploration for a total of 240 unique designs. There are too many features (dimensions) to visualize so a subset of this initial sample is displayed with only three of the candidate metals and with $h = 300\mu m$ in Figure 5.4.

Feature name	Features	Settings
Metal inclusion material	Elastic modulus E Density ρ Poisson ratio ν	Aluminum, Brass, Copper, Iron(cast), Lead, Nickel, Silver, Steel
Viscoelastic inclusion material	Elastic modulus E Density ρ Poisson ratio ν Loss factor η_s	Rubber, PMMA, PMMA 150C, Polystyrene, PTFE
Inclusion geometry	—	Sphere
Characteristic size	Inclusion diameter d	1 mm
Material interface	Distance from inclusion center h	50, 100, 200, 300, 400, 450 µm
Excitation	Frequency f	1 kHz to 160 kHz (30 steps)

Table 5.1: Features included in initial sample of asymmetric AMS absorber designs.

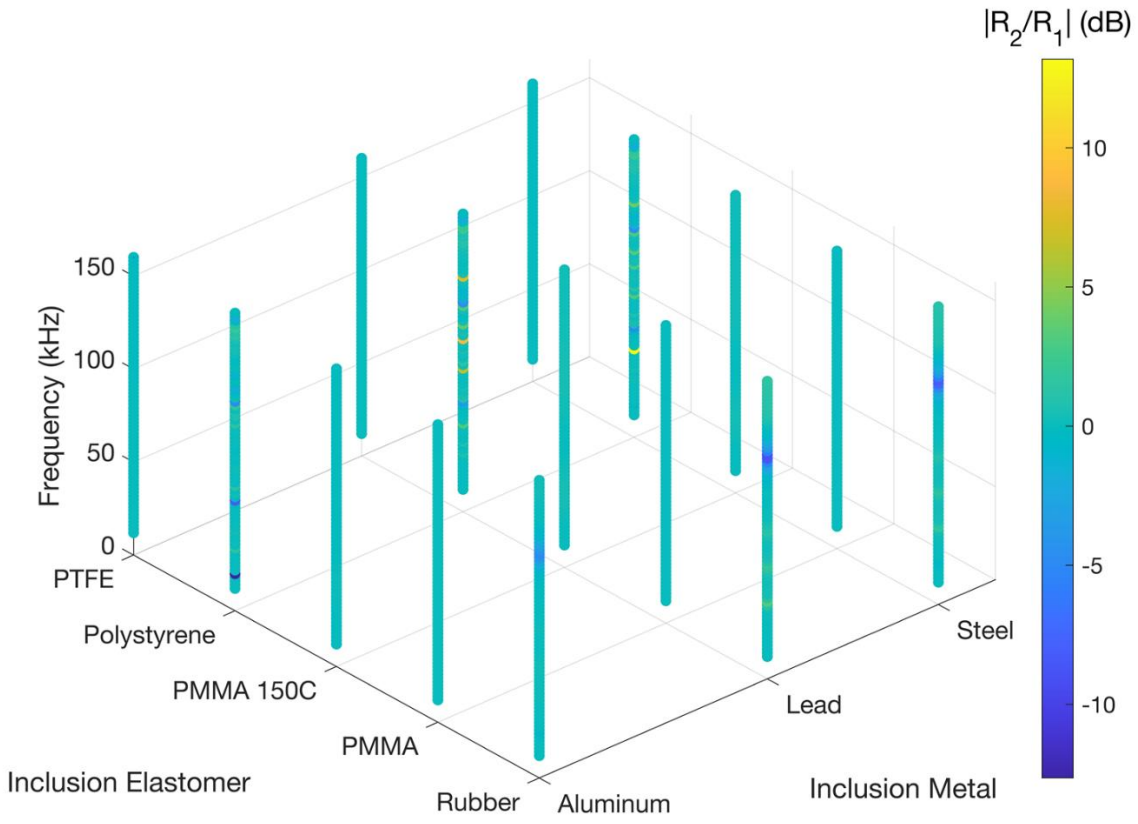

Figure 5.4: Asymmetric response of multiple material combinations where the geometric parameter settings are identical for all of these instances. Each of these design instances is manufacturing agnostic and has a centered spherical inclusion ($[x_a, x_b, x_c, r_a, r_b, r_c] = 0$). The dividing plane between materials is $h = 300\mu m$ for all instances in this plot.

The aggregate results of this initial exploration are used to identify the ideal design. From this sample, the most promising ideal design is made of silver and polystyrene with a material interface at $h = 200\ \mu m$ for excitation frequencies, $f \cong 125\ kHz$. The strong narrowband asymmetry in acoustic absorption of this design is shown in Figure 5.5. Narrow peaks of asymmetric response indicate that this design is sensitive to excitation frequency or, conversely, that performance at the design frequency may be very sensitive to geometric parameters.

The strong asymmetric response of this instance can also be visualized in 3D plots of the total pressure fields in the matrix and 2D cross-sections of stress in the inclusions. Stress inside the inclusion shows strong dependence on direction of incident waves. When propagating in the $+x$ direction, much more acoustic energy is converted to energy of deformation in the heterogeneous solid inclusion and therefore more energy is absorbed by the viscoelastic material. Recall that the AMS is an interface of finite, but subwavelength, thickness separating two half-spaces and that the asymmetric absorption occurs in a narrow band of frequencies around a resonance frequency of the inclusions. Due to the asymmetric material distribution within the inclusions, the mode shape of the resonance is also asymmetric, and thus there are different local impedances for waves incident from each side. Specifically, on one side of the AMS, the resonance leads to very little motion, while on the other side, the local particle velocity and pressure fields better match the characteristic acoustic impedance in the background medium. The asymmetric mode therefore leads to poor impedance matching and strong reflections for plane waves traveling in the $+x$-direction (case 1), while waves traveling along the $-x$-direction (case 2) are associated with a good impedance match and strong acoustic absorption (see Figure 5.5). This behavior is analogous to the resonances in de-tuned Helmholtz resonators reported by Long *et al*. [28] and Merkel *et al*. [29], but the present work exploits material property asymmetry in subwavelength structures to create asymmetric mode shapes and thus asymmetric acoustic absorption with reciprocal transmission. Visualizations of the resulting fields are presented in Figure 5.6.

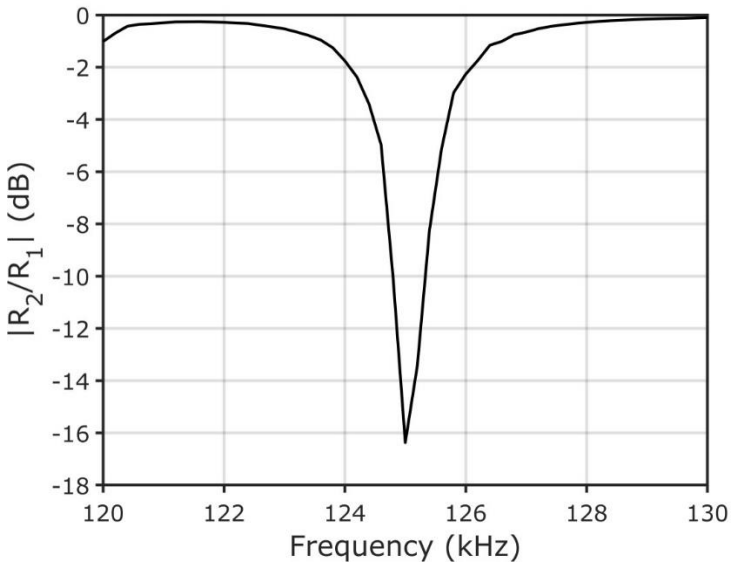

Figure 5.5: Narrowband asymmetric reflection response of an asymmetric AMS absorber made of silver and polystyrene inclusions embedded in a PDMS matrix.

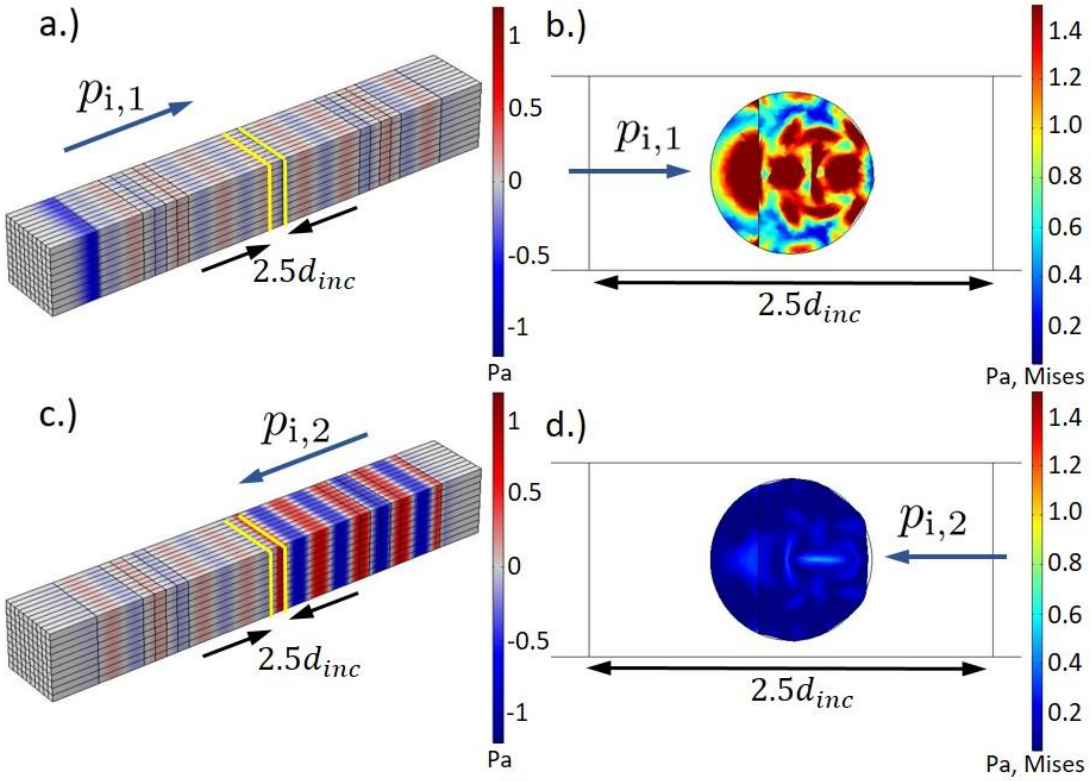

Figure 5.6: Total pressure fields for incident waves propagating along the $\pm x$-direction and the resultant stress field within the inclusions. (a, b) Results for the incident wave traveling from left to right (+x-direction), and (c, d) shows simulation results for incident wave propagating from right to left (-x-direction). Von Mises stresses are plotted on a cross-section of the inclusion.

5.4.2 Metamodeling with MLP Classifier

Like any part whose performance is sensitive to variability, manufacturing an AMS to achieve strong asymmetry in absorption requires tolerance specifications on accuracy. It is impractical to design parts like this one to exact feature sizes without accounting for manufacturing variation, especially in the case of AM, which is well-known to suffer from high part-to-part variation [195]. To account for manufacturing variation, each source of variability can be modeled as a random variable with a specified distribution based on known fabrication techniques. For AM processes, the distribution that describes the variation of any given feature is often determined empirically via a metrology study. However, variation in AM processes is inconsistent across machines, raw material types, process parameter settings, etc. [195]. As a result, the success rate of building a satisfactory asymmetric AMS absorber will be different as each process variable changes. For effective and efficient production of these parts, an estimate of the build success rate given a specific set of variable process conditions is invaluable, such that the build success rate quantifies the likelihood that a fabricated AMM meets a threshold for satisfactory performance.

For application to the asymmetric absorber design, the ability to specify a minimum magnitude of asymmetry is required. To that end, a classifier is a suitable type of metamodel because classifiers separate inputs into classes [192]. In this work, classes are separated in a binary sense based on whether the performance of a given design is satisfactory or unsatisfactory. We specify a performance threshold for acoustic absorption of 6 dB of directional asymmetry of reflection. A shallow neural network (NN) architecture

called a multi-layer perceptron (MLP) classifier is used as the metamodel for this application because it is a general purpose non-linear approximator that has many options for tuning the performance [192]. The same classifier would be applicable to any case for which a performance specification requires a minimum (threshold) value to render the part satisfactory.

Training Data Generation

Generating data necessary to train the MLP classifier requires querying the FEA sufficiently throughout the space of possible design feature settings. For a classifier that maps performance to manufacturing variability, this means starting with the ideal design, estimating the maximum range of manufacturing variability for each feature, and generating a space-filling sample of the variability space. The space-filling sample is generated using a Halton sequence, which is a quasi-random sequence that distributes samples quasi-uniformly throughout the design space and is described in detail in Section 3.1.2. Training the classifier with a space-filling sample improves its accuracy by minimizing the distance between any prospective design instance and a ground-truth point from FEA.

The ideal silver and polystyrene asymmetric absorber identified previously in this section is being designed for robustness to spatial variability of the build process. Using the best reported aerosol deposition raster widths as a reference [183] [184], the range of each spatial variable is estimated to be $\boldsymbol{x} \pm 10\ \mu$m where $\boldsymbol{x} = [x_a, x_b, x_c, r_a, r_b, r_c, h]$. A 7D Halton sequence is used to generate 500 unique structural designs in this space, which are then evaluated at 20 discrete frequencies in a 2 kHz frequency range spanning 124-126 kHz. This frequency range bounds the strong asymmetric response at 125 kHz in the ideal design. As a result of the spatial variability in the sample, peak asymmetry of the structures

subject to manufacturing variability varies in frequency and magnitude, as shown in Figure 5.7. The training data generation details are in Table 5.2. Data is generated by calling the FEA model for each design at each frequency and post-processing the data to calculate the asymmetric response. The computational expense of generating 10,000 data points is quite high as expected. Approximately 100 computing hours are required on a Linux machine with two 12-core Intel Xeon Gold 5118 CPUs and 250 GB of RAM for an evaluation rate of 100 samples/hour via FEA.

Training Data Generation		
Feature	Mean μ (mm)	Range (μm)
x_a	0	±10
x_b	0	±10
x_c	0	±10
r_a	0.5	±10
r_b	0.5	±10
r_c	0.5	±10
h	0.2	±10
Exogenous	N	Range (kHz)
f	20	124-126

Table 5.2: Variable definitions for all variables included in the Halton sequence of training data.

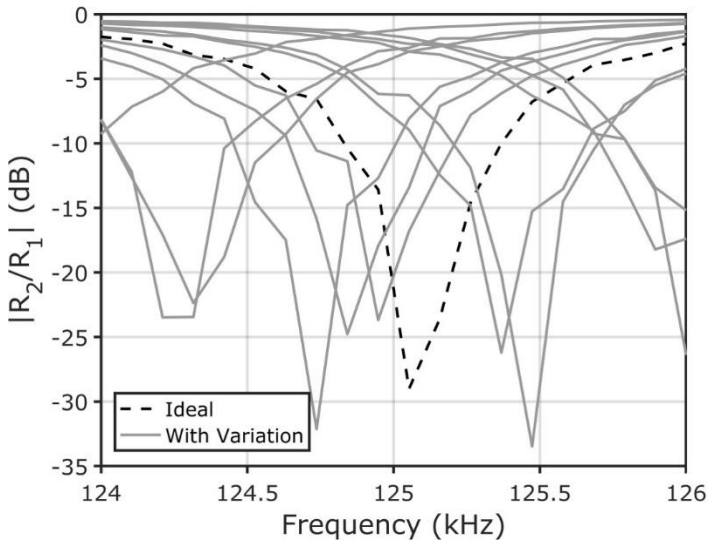

Figure 5.7: Asymmetric reflection of a small (N=10) subset of samples that are subject to manufacturing variation are shown with the ideal design. The frequency at which the peak asymmetry occurs and the magnitude vary with respect to the spatial feature settings.

Training the Classifier

The training data is used to fit the MLP classifier algorithm to accurately represent the FEA outputs in order to reliably classify designs that were not explicitly modeled. The MLP classifier is a network (graph) of connected nodes organized into three types of layers: an input layer, an output layer, and one or more hidden layer as shown in Figure 5.8 [196] [197]. Edges and nodes of the network are represented by weights and biases that are multiplied and summed from layer-to-layer such that each output node is a non-linear function of preceding layers.

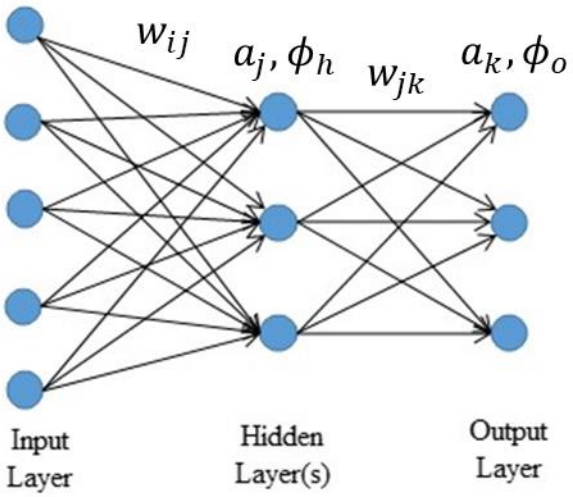

Figure 5.8: An example MLP architecture with five input features, one hidden layer of three nodes, and three output values.

Finally, each output node is scaled by an activation function [198]. The MLP is trained using backpropagation [199] and the mapping between input and output is optimized using a gradient-based optimization algorithm. A more detailed description of the MLP classifier algorithm and its hyperparameters is given in Section 3.3.1.

Data from the FEA is used to train the MLP classifier and optimization of its hyperparameters is performed using a three-fold cross-validation (CV) [200] scheme where prediction accuracy,

$$\mathrm{ACC} = \frac{TP+TN}{N}, \tag{5.4}$$

is maximized. In the expression above *TP* and *TN* represent true positive and negative predictions, respectively, and *N* is the total number of predictions. As part of CV, the predictive quality of a classifier trained on a subset of data is evaluated for comparison to other classifier training iterations. Training and testing the MLP takes less than a minute on a Windows machine with an Intel i7-8700 CPU and 32 GB of RAM. Cross-validation

is the most straightforward way to optimize hyperparameters, and the CV process also helps avoid overfitting by training and testing the MLP on multiple unique slices of the training data. Specified hyperparameter settings are tested for all unique combinations. For this design, the optimized hyperparameters are the hidden layer quantity and size, the activation function, the weight optimizer, and the regularization parameter (λ_w). Because training and testing the model requires very little computational expense, a large hyperparameter space was explored. The hyperparameter search space and optimal settings are displayed in Table 5.3. With optimized parameters, training accuracy of the classifier is repeatably 98%.

Hyperparameter	Settings	Optimized Setting
Hidden layers (# 1st layer, # 2nd layer)	(100,0), (200,0), (500,0), (100,2), (500,2), (100,100), (500,500)	(100,100)
Activation function	Logistic, Rectified linear (ReLu), tanh	ReLu
Weight optimizer	L-BFGS, Stochastic gradient descent (SGD), Adam	Adam
Regularization param. λ_w	Logspace 10^{-4} to $10^{-0.25}$	$10^{-2.04}$

Table 5.3: Hyperparameters for the MLP classifier of an asymmetric absorber. Optimal settings identified with cross-validation are in the rightmost column.

5.4.3 Evaluating Manufacturing Processes with Monte Carlo Simulations

Once the classifier is trained to a high accuracy, it is used to evaluate the success of manufacturing the ideal design on a machine with non-ideal size and position accuracy (i.e. spatial variability). The trained classifier is called 1 million times by the MC simulation to evaluate the expected success rate of parts subject to manufacturing variability. One million calls to the trained classifier takes approximately 2 seconds on the

same machine used to train the classifier. The success rate of this design instance is evaluated for multiple different idealized AM machines where the geometric variables from Table 5.2 are distributed random variables. These random geometric variables are characterized by multivariate truncated Gaussian distributions and multivariate beta distributions. Truncated and beta distributions are chosen to represent centered and biased distributions while being bounded. As a matter of practicality, the distributions must be bounded to prevent the inclusion geometry being built outside the matrix. The biased beta distributions represent a manufacturing process that commonly skews one way (e.g., shrinkage or over-sintering).

The truncated Gaussian distributions are truncated at $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$ for $\mu_x = 0$ mm, $\mu_r = 0.5$ mm, $\mu_h = 0.2$ mm and $\sigma_{x,r,h} \in \{1,3,5,8,10,15,20\}\,\mu$m. The range of standard deviations, σ, was selected such that the lower bound matches the thinnest known layer thickness achieved from aerosol deposition techniques ($1\mu m$) [183], while the upper bound represents double the raster width ($20\mu m$) [184]. The beta distributions are defined as $Beta(2,6)$ and $Beta(6,2)$ with the lower bound (LB) and upper bound (UB) defined as $(\mathrm{LB}, \mathrm{UB}) = (\boldsymbol{\mu}, \boldsymbol{\mu} + 10\mu m)$. Probability density functions (PDFs) of these distributions are shown in Figure 5.9 for a single variable. Figure 5.10 shows $\pm 1\sigma$ truncated Gaussian distributions a variety of σ settings. For a real AM machine, variability of every spatial feature would likely be described by a different probability distribution and any distribution that characterizes a machine's spatial feature variability within the range of training data is applicable. For the sake of this demonstration, the distributions and standard deviations for each variable are assumed to be defined identically (e.g., $x_{a,b,c} \sim N(\mu_x, \sigma)$, $r_{a,b,c} \sim N(\mu_r, \sigma)$, $h \sim N(\mu_h, \sigma)$) and frequency is sampled uniformly $f \sim U(124\ kHz, 126\ kHz)$ for all MC simulations.

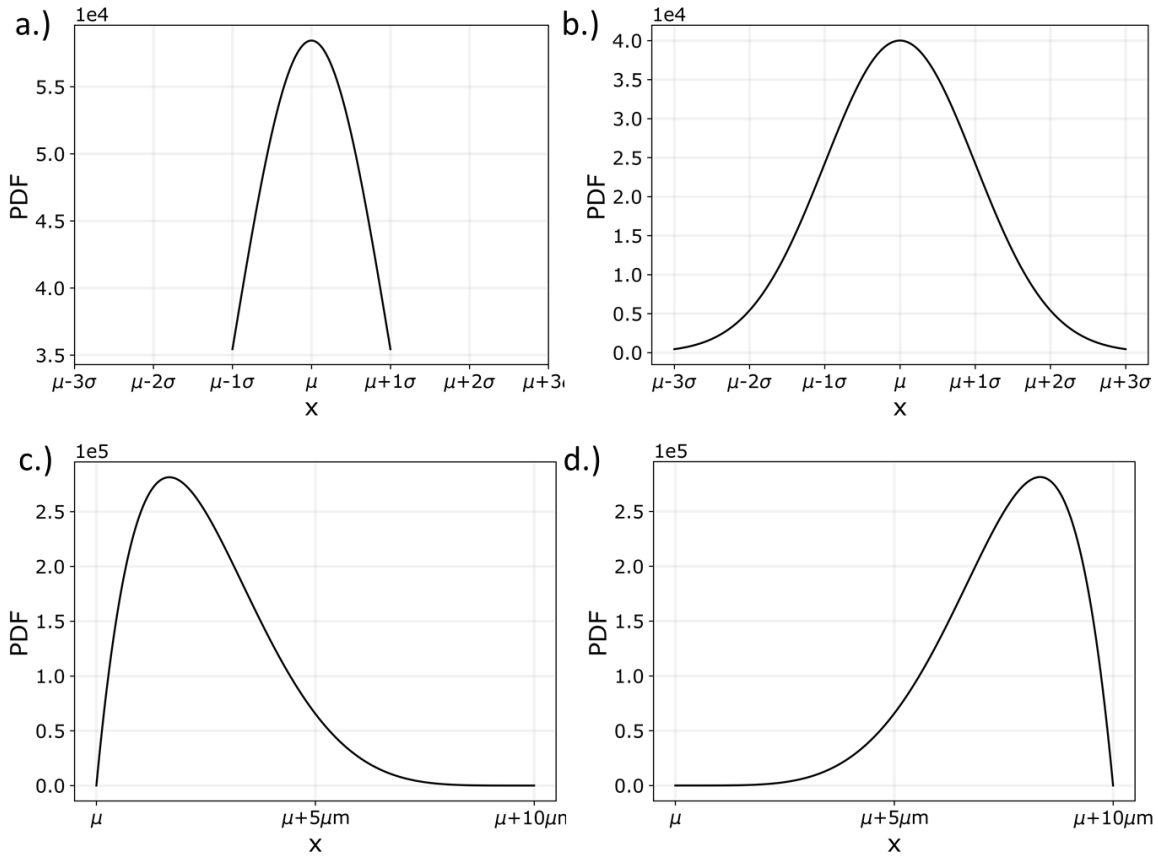

Figure 5.9: PDFs of the distributed geometric variables used for MC simulation. Single variable distributions are shown for ease of visualization. Each distribution represents the idealized manufacturing variability of an AM machine. Distributions are: (a) ±1σ truncated Gaussian, (b) ±3σ truncated Gaussian, (c) Beta(2,6), and (d) Beta(6,2).

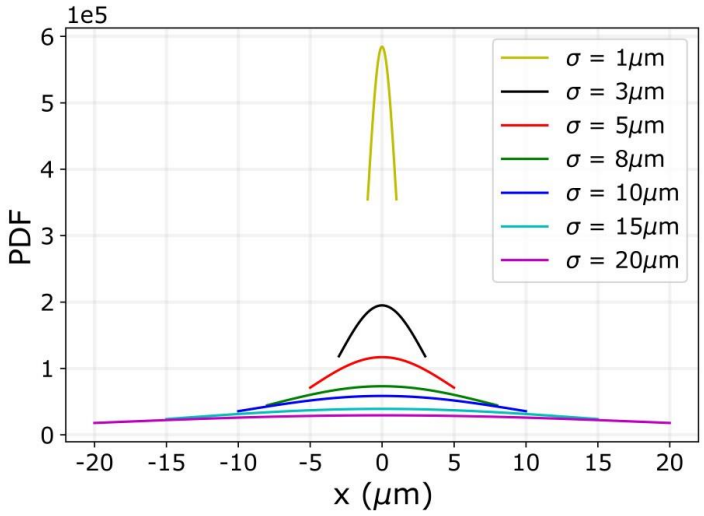

Figure 5.10: PDFs of ±1σ truncated Gaussian distributions used for MC simulation with varying standard deviation (σ) parameters.

Finally, for validation of the metamodel, a subset of 500 designs generated by each MC simulation are evaluated with FEA and compared with the classifier metamodel's predicted performance. The validation step ensures sufficient accuracy of the classifier for a specific set of process conditions.

5.5 AM-Aware Metamodeling Results

Results of the MC simulation using the MLP classifier metamodel provide important information for the prospective manufacturing of this asymmetric absorber. Most clearly, it shows the proportion of as-built parts with dimensions specified by the idealized design that would be expected to achieve a satisfactory magnitude of asymmetry. Conversely, the required manufacturing variability for an AM process can be specified in order to reach a satisfactory as-built success rate. As a result, the manufacturer gains information to balance the risk of part failure with the cost of accuracy. For the example case considered here, the predicted success rates are higher for lower variability, as one

would expect, with a success rate of approximately 40% for a machine with $\sigma = 1\ \mu$m accuracy for all specified geometric variables. Conversely, running a machine with $\sigma = 20\ \mu$m would lead to only about 14% success rate. Success rate results for all of the $\pm 1\ \sigma$ truncated Gaussian distributions used in MC simulations are shown in Figure 5.11(a). Figure 5.11 (b) shows results for $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$ truncated Gaussian distributions. Results of a validation subset generated by evaluating the FEA simulation are also shown in Figure 5.11 labeled as "True."

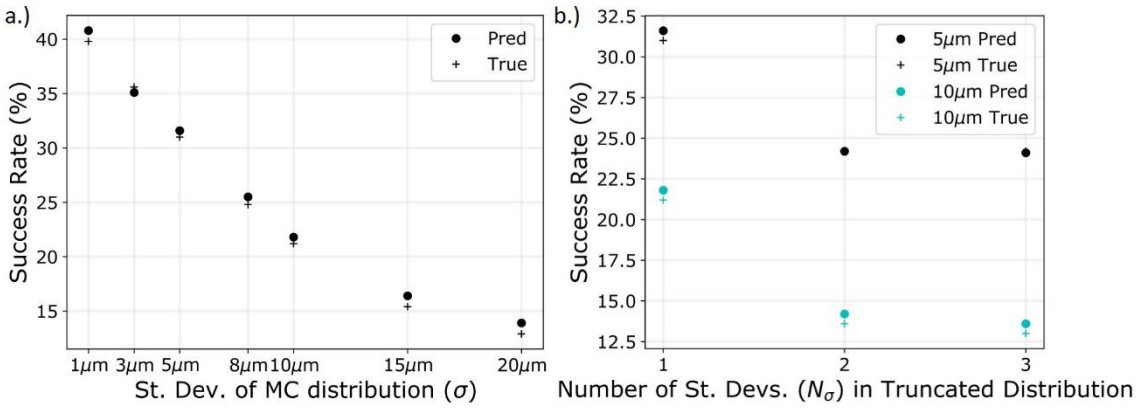

Figure 5.11: Predicted and true (from FE simulation) success rates from the MC simulation process are shown for many idealized AM machines characterized by a variety of different distributions of geometric feature size and position. Plot (a) compares results for a variety of ±1σ truncated Gaussian distributions. Plot (b) compares ±1σ, ±2σ, and ±3σ truncated Gaussian distributions for two different values of σ.

Analysis of validation samples shows that the classifier is ≥95% accurate when samples are generated from the MC simulations with $\pm 1\sigma$ truncated Gaussian distributions, as documented in Figure 5.12(a). As shown in Figure 5.12(b), classifier accuracy for the $\pm 3\sigma$ truncated Gaussian distribution with $\sigma = 10\ \mu$m is very high at 98.2% despite no designs with a feature size $\mu \pm 30\ \mu$m being included in the training set.

Although very few samples are taken from the tail probability near $\sigma = \pm 30\,\mu$m, it is notable that the metamodel is effectively extrapolating in these cases.

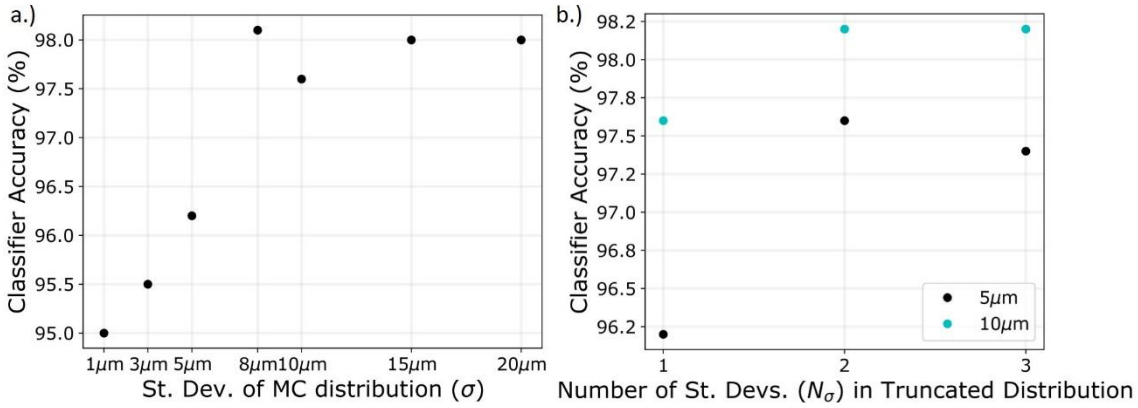

Figure 5.12: The accuracy of the classifier is evaluated relative to MC simulations for several different distributions. Plot (a) compares results for a variety of ±1σ truncated Gaussian distributions. Plot (b) compares ±1σ, ±2σ, and ±3σ truncated Gaussian distributions for two different values of σ.

In contrast to the non-biased Gaussian distributions that concentrate as-built instances near the ideal design, the beta distributions are used in MC simulations to show the effect of bias in a manufacturing process. This case represents a machine with a process parameter maladjusted leading to the mean spatial result being skewed away from the expected mean. When the geometric variables are skewed toward the design setting with $Beta(2,6)$, the manufacturability performance is very similar to the Gaussian distribution results. However, when they are skewed away from the design setting and closer to the upper bound of $10\mu m$ with $Beta(6,2)$ there are no satisfactory as-built instances identified in the validation set. The classifier accuracy is still good for both cases; it even successfully identifies all instances as unsatisfactory when the variables are distributed as $Beta(6,2)$. The results of the skewed MC simulations are documented in Figure 5.13.

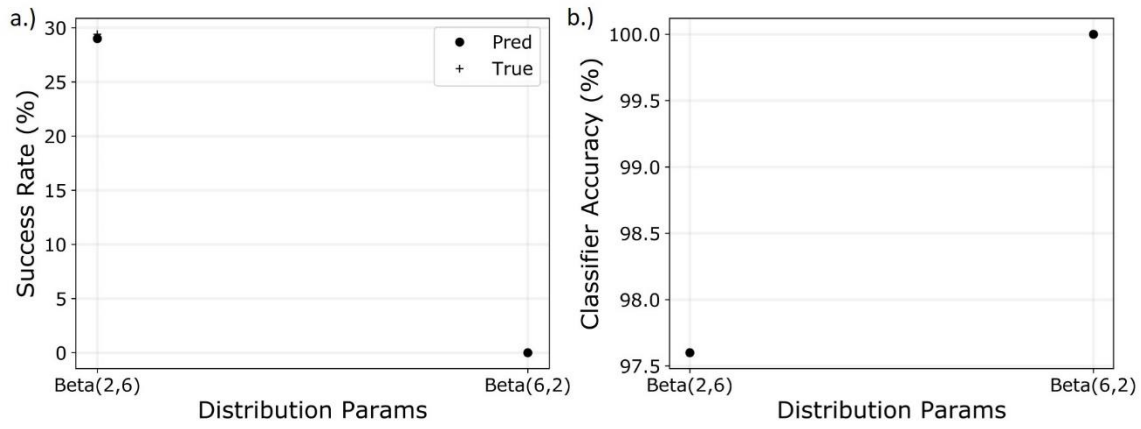

Figure 5.13: (a) Predicted and true (from simulation) success rates from the MC simulations with skewed beta distributions. (b) Classifier accuracy performance for both beta distribution validation sets. Where the geometric variables are distributed as $Beta(6,2)$ there are zero successful as-built instances.

Ideally, the accuracy of a metamodel is the same for any spatial feature distributions tested. The demonstration problem shows very strong accuracies of 95-99% as long as the metamodel is fit to a range of data such that studies require interpolation only. While some extrapolation may be possible, the metamodel will lose accuracy very rapidly with respect to extension beyond its training data range. For this reason, the selection of an appropriate training data range is important. The range $\pm 10\ \mu$m was selected for training in this demonstration because it became clear during the initial design exploration that very few samples were satisfactory for larger geometric feature ranges at the frequencies of interest.

Clearly, the performance of this structural asymmetric absorber is very sensitive to spatial variation during manufacturing. This is evident in that manufacturing success rate is only 40% with a modest standard deviation of $\sigma = \pm 1\ \mu$m on all geometric features. The success rate declines non-linearly as the distributions widen, whether due to more tail probabilities included ($\pm 1\sigma$ to $\pm 3\sigma$ truncations) or larger σ. The success rate decline

flattens as the truncated Gaussian distributions get wider because they are all centered on the mean value of the ideal design. Comparatively, the metamodel predicts zero instances of successful builds for the beta distribution that is skewed away from the mean. This shows the value of unbiased error in manufacturing these metamaterials. While it may be unlikely that all spatial features skew that much from the ideal design dimensions, it demonstrates a worst-case scenario and shows that the metamodel's predictive capability is equally strong in the unsatisfactory regions of the design space.

Results Discussion

This work has demonstrated that the fabrication of an AMS with Willis constitutive behavior, well-matched to water, with asymmetric absorbing capabilities is possible using subwavelength asymmetric inclusions in an elastomeric matrix material. However, for the design to function properly at a design frequency of 125 kHz requires strict levels of accuracy in the fabrication process. The information provided here lends itself to a few options. Most obviously, the size of the structure could be increased. However, increasing the size will necessitate designing for lower frequency incident waves to stay in the metamaterial regime ($ka \ll 1$) and maintain approximately planar scattered fields. Increasing the size of the metamaterial may also limit its applicability as its size and weight increase accordingly. Another option is to develop a rigorous multi-material AM process suitable for building the AMS so it can work at higher frequencies. The methodology developed in this work evaluates the manufacturing process accuracy required to build this type of asymmetric absorber for any desired operating frequency and size scale. At time of writing, aerosol deposition development has achieved $1\mu m$ size resolution [183] but only in the direction normal to the deposition surface. Aerosol deposition looks promising for building inclusions although future work will need to more fully characterize the spatial

variability of the process, as well as address the co-deposition of viscoelastic polymers. Additionally, the layer-wise deposition of many layers is difficult due to thin layer sizes and layer deposition abrading the previous layer.

In addition to spatial parameters, constitutive material properties are a source of manufacturing variability. This study only considered spatial parameters to reduce complexity and improve clarity of the method and demonstration, but in practice material properties are also random variables dependent on AM process parameters. For example, the material properties of PDMS are functionally related to resin viscosity, curing conditions, additives, material aging, etc. Metals produced with AM processes often have material properties that differ from reference values and are usually anisotropic based on build directions. For these reasons, the variability of as built material properties cannot be ignored when building sensitive AMS such as the asymmetric absorber considered here. Fortunately, as the AM industry grows and looks toward the future, many organizations are characterizing their machines and performing exactly the type of variability studies needed to execute a classifier-based robust design with MC simulation [201]. However, there will continue to be a need for design methods that help designers predict and manage the effect of process-induced variability on acoustic material performance.

Although this work focuses on a particular AMS with a single layer of elliptical inclusions, the design methodology presented is applicable to the design of any structure whose performance is sensitive to spatial or material variability associated with AM machines. Reusability of the metamodel is key to reducing the computational expense of evaluating the robustness of manufacturing the part on an AM machine, but it is crucial that the designer understand the limits on reusability. A metamodel can be reused when the variability of the manufacturing process stays within, or very near, the range of the data used to train the metamodel. Any significant changes in geometry will require training of

a separate metamodel. No metamodel is universal, but the method presented here satisfies the need to evaluate many machines given a target design and performance specification.

Chapter 6: Efficient AMM Metamodeling for Design Domains of Variable Size

Passive metamaterials are composed of many small-scale features to cause macro-scale behavior that is otherwise unachievable with conventional materials. Analyzing metamaterials with a wide range of size scales leads to large quantities of data to represent topologies and requires a tremendous number of computations to evaluate them. The most direct way to address this challenge is to employ big-data supercomputing techniques that can represent the entire metamaterial at the scale of its smallest features. Aage *et al.* demonstrated this approach to design a full-scale airplane wing by gradient-based topology optimization over multiple days on an 8000 CPU supercomputer [4]. While computational power is ever increasing, access to these resources is limited, so it is still prudent to develop more efficient design methods. Multilevel BNCs [202] are one such tool that have been demonstrated for design exploration across hierarchical size scales. Another approach is to design a structure with repeated unit cells and exploit symmetry or periodicity relations to reduce the size of the domain. For topology optimization, multi-scale techniques based on homogenization of micro-scale responses [203] and simultaneous optimizations for each discrete scale [204] have been developed. Another approach is to use machine learning (ML) methods to create a metamodel that is capable of representing macro-scale behavior of a metamaterial with a combination of micro-scale components without strict homogenization.

When designing metamaterials, domain size is defined by the dimensionality of the design space, which is related to the complexity of the underlying topology and the number of design variables associated with it. Metamaterial systems with high-dimensional design spaces and behaviors that vary with respect to excitation frequency are challenging to represent with metamodels for several related reasons. Training models for high-

dimensional systems requires large volumes of data, and responses that are sensitive with respect to frequency compound the issue by requiring dense sampling through the spectrum. When the design space dimensionality is variable, efficiency is gained by training a metamodel that is valid for systems with a range of dimensionalities. This process is known as domain adaptation in transfer learning research and enables ML models to accept designs as sets (vectors) of features with variable dimensionality (length) D. This chapter presents a method to efficiently design acoustic metamaterials (AMMs) with many heterogeneous features such that analysis cannot be simplified by reducing the domain to a unit cell, and therefore, homogenization cannot be applied. A metamodel that is agnostic to input dimensionality represents the AMMs as graphs and uses neural network architectures that operate on graphs to update the metamodel from state-to-state. For the purposes of AMM analysis and design, states are discrete frequencies in the spectrum. Encoding the attributed graph before—and decoding after—calling the state update functions enables the update functions to maintain generality with respect to dimensionality. The trained update functions are then applicable to systems with a range of input dimensionalities in the training set. By skewing training sample generation toward lower-dimensional systems that are less computationally expensive to evaluate, the computational expense of gathering training data can be reduced with minimal loss of accuracy in predicting the dynamic behavior of higher-dimensional systems. The method is demonstrated using the asymmetric acoustic absorber described in Section 2.3 where geometric features of each inclusion in the metasurface layer may vary.

6.1 Design of Dynamic Systems with Variable Design Space Dimensionality

Metamodeling of a high-dimensional system is difficult because many samples are required to train the model sufficiently, and acquiring samples is often expensive for

engineering tasks [205]. This trait of metamodels is often referred to as the curse of dimensionality, and many metamodeling techniques have been developed to enhance the trade-off between accuracy and training sample size N. Some strategies include dimension reduction, optimization of coarse and fine models simultaneously, and the use of radial basis functions combined with model reduction, among others [205] [206] [207] [34] [208] [209] [210]. These techniques are best suited when the output of interest is a single-valued or aggregated response (even if a dynamic process acts to transform input to output) [211] [212].

Creating a metamodel that can be queried at any step during a dynamic process is also a challenging pursuit that has inspired some work on the topic [87] [213] [214] [215]. Despite incremental increases in training efficiency, these referenced metamodels can evaluate only inputs with design spaces having the same number of dimensions as the training data. Consequently, a new metamodel must be trained for any change in design space dimensionality, which makes representing combinatorial problems expensive. The metamodeling approach presented in this chapter addresses both of these challenges. Using neural networks (NNs) to operate on graphs, we create metamodels of dynamic systems that are general with respect to length, D, of the input design feature set. The metamodel is also able to predict dynamic responses for combinations and spatial arrangements of design features that are not included in the training set. It is worth noting that, although general for input set length, this metamodel does not extrapolate but rather acts as a function approximator for the interactions among graph elements based on their attributes. Any individual input feature setting in a model query must be within the bounds of that feature's set in the training data for the metamodel to achieve useful predictive capabilities. The following sections introduce how a dynamic system with many features can be represented as a graph and NN architectures that operate on these graphs.

6.2 Representing Physical Structures as Graphs

Many types of structured information can be represented as graphs, G, composed of nodes, V, and connecting edges, E. This includes canonical examples such as the Traveling Salesman and Seven Bridges of Königsberg problems to more recent social and computer networks. It is often natural to represent systems as graphs in engineering disciplines because, in essence, engineers deal with objects (nodes) and interactions (edges). For these purposes, attributed and directed graphs are used to describe the properties and actions of an engineering system. Some systems that are well represented by graphs include truss structures, gravitational interactions, serial manipulator robots, and circuits. The flexibility for grouping features is an advantage of representing systems as graphs and sets the approach apart from other feature-based metamodeling techniques.

An attributed, directed graph, G, has three types of elements: nodes, $\boldsymbol{v}$, edges, $\boldsymbol{e}$, and global features, u. Attributes for each element can be collected into a set and indexed to denote structure and connectivity. Attribute sets are defined as: global attributes, $\boldsymbol{u}$, node attributes, V, and edge attributes, E. The direction of action described by each connecting edge is expressed by organizing each edge into a three-tuple with the index of receiving and sending nodes associated with each edge's attributes. With these definitions, the graph may be formulated as

$$G = (\boldsymbol{u}, V, E), \tag{6.1}$$

with

$$V = \{\boldsymbol{v}_i\}_{i=1:N_m}, \tag{6.2}$$

$$E = \{(\boldsymbol{e}_k, r_k, s_k)\}_{k=1:N_e}, \tag{6.3}$$

where:

N_m: quantity of nodes,

N_e: quantity of edges,

r_k: index of receiving node,

s_k: index of sending node.

A graph defined in this way with four nodes and five edges is illustrated in Figure 6.1.

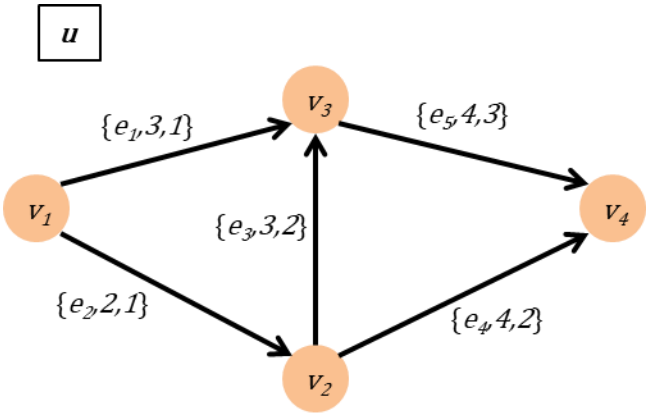

Figure 6.1: A graph with four nodes, v, and five edges, e, has global attributes $\boldsymbol{u}$. Connectivity of nodes and direction of action by edges is denoted by associating sender and receiver indices of nodes to each edge definition.

Engineering systems made of discrete elements are naturally adaptable to a graph structure but continuum systems may also be defined on a graph with some assumptions. Just as a truss can be analyzed with joints as nodes and members as edges, an acoustic metasurface of inclusions can be represented with inclusions as nodes and their interactions as edges. Modeling the asymmetrically absorbing AMS as a graph is detailed in Section 6.4. Updating the attributes of a graph is done with a function of the elements and their attributes. The update function is not constrained to any particular form, but NNs are a flexible type of non-linear approximator and can be trained to represent the relations between graph elements when they are not known *a priori*. However, a NN architecture must be deliberately designed to operate on dynamic systems, graphs, and graphs with variable dimensionality.

6.3 Neural Network Architectures for Dynamic Systems

Although NNs can require significant volumes of training data to provide accurate predictions [34] [94], the increasing availability of training data for many modern engineering problems makes NNs more attractive as metamodels. Simple NN architectures like the multi-layer perceptron (MLP) map inputs to outputs and can feasibly accept a time-step signature or other system state variable as an input but have some shortcomings for modeling dynamic systems. In particular, other input variables are dependent on the state variable and an MLP is incapable of updating inputs based on the output at a previous state.

However, research on deep learning has produced NN architectures that are tailor-made for dynamic systems. The original NN architecture for dynamic data is the recurrent neural network (RNN) [199] by Rumelhart *et al.* in 1986. It is well suited for dynamic systems because an RNN maintains state information across a sequence of calls to the model by passing latent variables with a feedback loop. There have been countless architectures developed on dynamic systems since then [216] and the field has subdivided to address specific functionalities. The following sections provide a brief overview of temporal NN methods and how NNs may be coupled with graph representations to model dynamic systems with variable input dimensionality.

6.3.1 Temporal NN Methods

Research seeking to improve the capabilities of NNs for engineering applications has led to two popular approaches to modeling non-linear dynamic systems. The most obvious and perhaps more difficult approach is to directly approximate parameters of ordinary and partial differential equations (ODEs and PDEs). One such method called Neural Differential Equations [217] performs backpropagation through ODE solvers by solving a second ODE through a reversed time series using adjoint states. The PDE-Net

method fits parameters of PDE models in order to learn the expression of a system's governing PDE by training a feed-forward deep network [218]. This approach implements NNs to model dynamic systems and is well suited for discovering an expression that underlies the nature of a particular system when the general structure of the expression is known.

The other popular approach to modeling dynamic systems with NNs is to operate on graphs. Gori *et al.* [219] introduced graph neural networks (GNNs) as a class of models for learning on graphs. In this formulation NNs such as RNNs are used as functions to update the state of the graph. Others have used convolutional filter layers in NNs to treat graph states analogously to image data. Kipf and Welling [220] demonstrated that a well-designed architecture of sequential convolutional neural networks (CNNs) could model time-series data faster than a similar system using RNNs because of the parallelization improvements gained by not requiring the sequential passing of state information across time steps. Overall, there is a wealth of literature detailing implementation of NNs on different types of graph data; however, most approaches require spatial input features to be constant through time. Zhou *et al.* [221] and Wu *et al.* [146] provide near-comprehensive surveys of GNN methods.

6.3.2 Spatiotemporal NN methods

From an engineering design perspective, interesting work has been done to model systems that change through both space and time using attributed graphs and neural networks. This class of algorithms is called spatiotemporal graph neural networks (STGNNs). STGNNs are designed to operate on graphs with attributes that change in both space and time. The NN based processing core of STGNNs can be formulated with RNNs or CNNs or a combination of both to utilize the strengths of each. Most STGNNs in the

literature have architectures that are tailor-made for specific problems [146]. This class of models has been implemented with CNN cores for traffic forecasting [222] [223] and classification of human actions from kinematic data [224]. For graphs where the existence of node or edge inputs changes, these and many other STGNNs are no longer functional [146]. While custom architectures are appropriate for modeling systems with static feature dimensionality, problem specific architectures lack the ability to generalize across input dimensionalities.

6.3.3 Encoded Graph Neural Networks

An effective way to operate on graphs with variable dimensionality is to use update functions that operate on groups of multiple graph attributes. These update functions can be represented by any trainable non-linear approximator, but NNs with a relatively simple architecture are a good choice for the task of updating all the nodes or all the edges of a system. Many common combinatorial optimization problems have been explored with similar approaches [225] [226] [227]. The breadth of work on GNNs has led to the development of frameworks and codebases to unify the field and spur further development [228] [229] [230]. Sanchez-Gonzales *et al.* demonstrated a GNN with the capability to learn nonlinear physics of multi-bar linkages by encoding input graphs [231]. Their implementation showed that the model predicted motion with good accuracy even for systems with a few more links than the maximum number included in the training data set.

Graph-encoding GNN methods have the ability to represent interactions between graph elements. This strength, along with aggregated node and edge update functions, enables the metamodeling of multiple systems with different input feature dimensionalities with a single model and reduces the computational expense from training each separately.

The limitation is that the relationship between graph elements must be similar regardless of quantity.

The arbitrary function approximators, ϕ, that update global, node, and edge attributes operate on each graph element separately with a single function per type of graph element. Encoding, decoding, and processing input to output are done in separate NN blocks but all operate on the graph elements the same way with

$$\boldsymbol{e}'_k = \phi_b^e\left(\boldsymbol{e}_k, \boldsymbol{v}_{r_k}, \boldsymbol{v}_{s_k}, \boldsymbol{u}\right), \tag{6.4}$$

$$\boldsymbol{v}'_i = \phi_b^v(\bar{\boldsymbol{e}}'_i, \boldsymbol{v}_i, \boldsymbol{u}), \tag{6.5}$$

$$\boldsymbol{u}' = \phi_b^u(\bar{\boldsymbol{e}}', \bar{\boldsymbol{v}}', \boldsymbol{u}), \tag{6.6}$$

where subscript b represents the block index and $\bar{*}$ represents an aggregation (mean or sum). Encoding and decoding the graph with NNs allows for the number of nodes and edges to vary because the input and output size of the processing block remains constant. Figure 6.2 illustrates how data flows between blocks of the GNN architecture for a set of update functions, ϕ_b, $b = 1{:}3$.

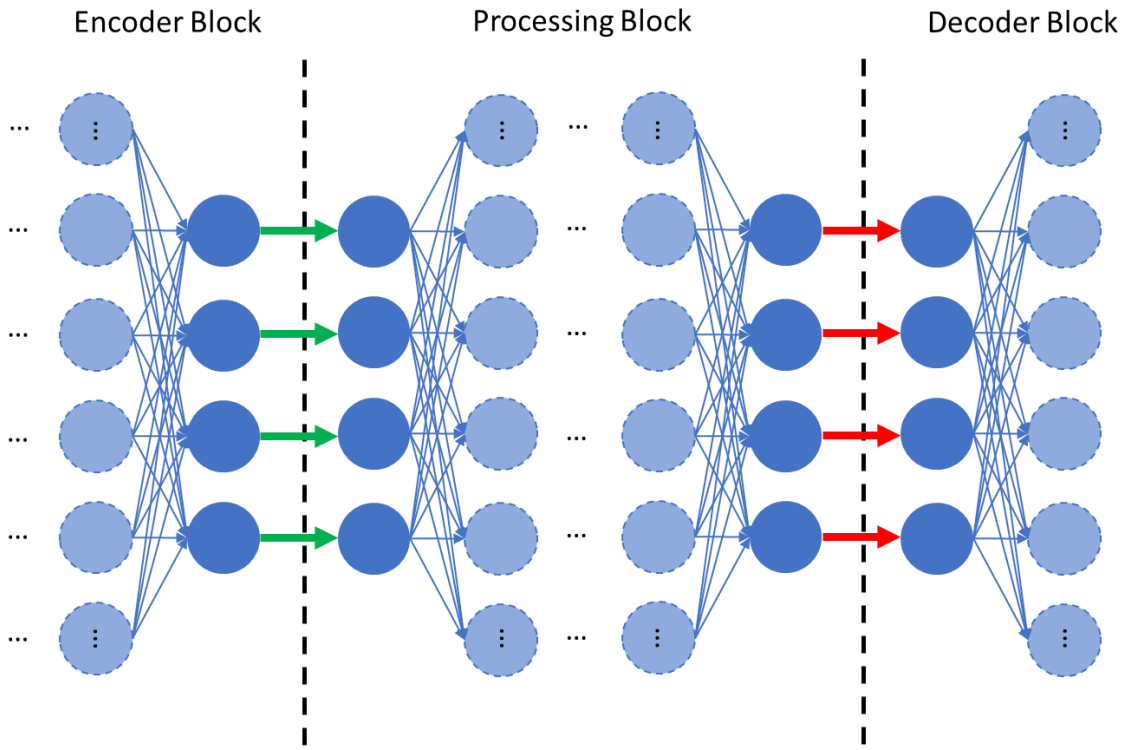

Figure 6.2: The encoder NN block feeds a constant input size to the processing block and the decoder NN block accepts the output of the processing block with a constant size.

Any NN architecture may be used for each block so long as the inputs and outputs of the encoder, processing block(s), and decoder are compatible as shown in Figure 6.2.

This approach has been shown by Battaglia *et al*. to work for a few systems with variable quantities of nodes and edges [228]. This work, however, will extend the concept to AMM design with heterogeneous arrangements of unit cells with variable design space dimensionality.

6.4 AMS Design with GNN Metamodeling Methodology

A metamodel that accepts a design space of variable dimensionality is a valuable tool for designing the directionally asymmetric acoustic absorber introduced in Section 2.3.

Analysis and design of that type of structure has utilized periodic Floquet boundary conditions to exploit in-plane symmetry and reduce the domain of the system to a single unit cell. As a result, every inclusion in the semi-infinite AMS layer has an identical geometry. Because the asymmetry is dependent on resonant behavior of the inclusions, a layer of identical inclusions produces narrowband asymmetry at resonant frequencies of the AMS. However, a heterogeneous arrangement or functionally graded inclusion features across the AMS planar layer may produce broader band asymmetric responses.

Literature on similar resonance-based AMM absorbers has shown that heterogeneous combinations of micro-scale features produce broadband and multi-band absorption [232] [233]. Furthermore, early-stage manual design exploration has identified designs that exhibit broader band asymmetric absorption using a heterogeneous layer of multi-material inclusions in the AMS introduced in Section 2.3. Broader band asymmetric absorption was observed in AMS layers composed of repeating "super cells", which are arrangements of single-inclusion unit cells in which the geometry of each inclusion may vary independently. The super cells also have Floquet periodic conditions on the boundaries that are orthogonal to the inclusion layer plane. Super cells of 16 total unit cells in a square-packed planar arrangement were required to identify broader band performance. Figure 6.3 shows a 4x4 arrangement of inclusions labeled with indexed feature h, which represents the distance of the material dividing plane from center of the inclusion ($x = 0$).

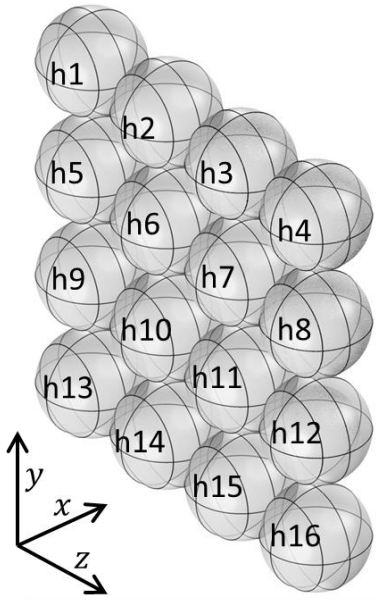

Figure 6.3: A 4x4 arrangement of inclusions labeled with indexed features h in an isometric view.

Super cells are more computationally expensive to evaluate than single unit cells because the number of elements and degrees of freedom is proportional to the number of cells in the geometry of the FEA. With a 4x4 super cell, evaluating the system at a single excitation frequency takes 15 minutes on a Linux machine with two 12-core Intel Xeon Gold 5118 CPUs and 250 GB of RAM. Because of the high expense and need to evaluate many frequencies to generate a response curve with respect to frequency, only h was varied among inclusions in the initial super cell exploration to minimize confounding variable interactions and computational expense. A few super cell arrangements showed broader band performance than what is observed from homogenous AMS layers. Figure 6.4 illustrates some of these examples along with a homogenous layer for reference.

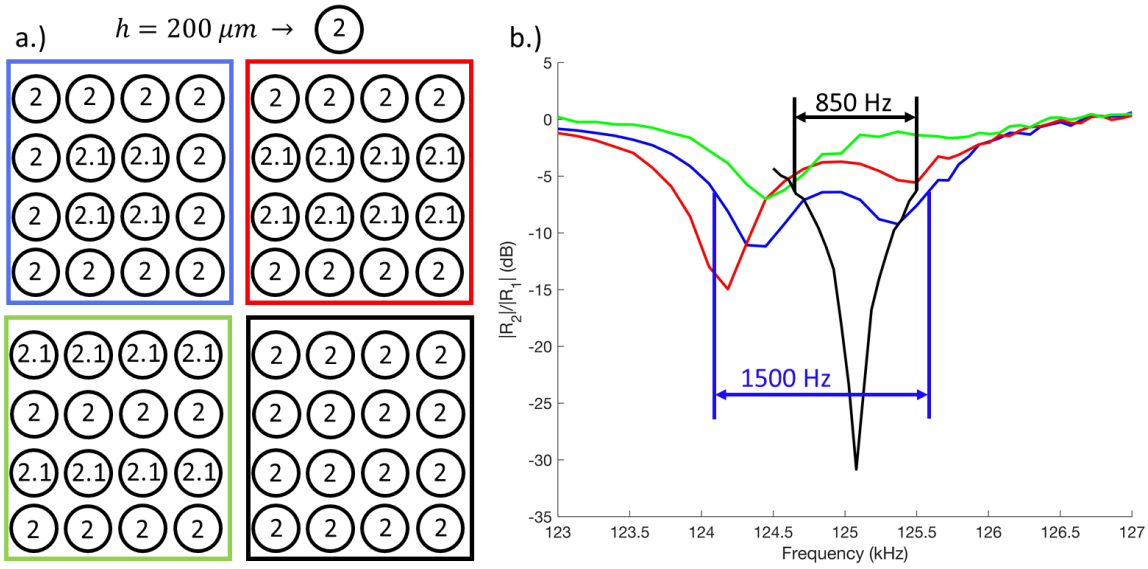

Figure 6.4: Resulting asymmetric response curves with respect to frequency for four different arrangements of inclusions in a 4x4 super cell. One arrangement (blue) exhibits asymmetric behavior spanning a 1500 Hz band at $|R_2/R_1| < -6dB$. A homogenous layer of inclusions (black) produces asymmetric behavior spanning an 850 Hz band at $|R_2/R_1| < -6dB$. The other arrangements also show broader band asymmetric absorption but at a lower magnitude.

These preliminary results motivate metamodeling with GNNs to find broadband asymmetric absorption. However, without broadband behavior in the training set the model cannot predict broadband asymmetry. Within the constraints of the computational power and time available and considering that feature combinations leading to broadband performance are not well-understood, the GNN metamodel is demonstrated using training data from super cells with only one to nine inclusions. The following section describes the NN architecture used in GNNs for the asymmetrically absorbing AMSs before the geometry-specific modeling is explained.

6.4.1 GNN Model Architecture

To encode the graphs and achieve size generalizability, each update function requires three separate blocks of NNs. All update functions, ϕ, are MLPs with 2 hidden layers of 16 latent variables. The encoder and decoder MLPs feed into LayerNorm operators that normalize the distributions of intermediate layers [234]. The processing block uses a message passing neural network (MPNN) [235]. An MPNN is also an MLP with 2 hidden layers of 16 latent variables but operates multiple times sequentially by passing output to input as many times as is specified by the number of processing steps, N_{ps}. This feature allows the GNN to make multiple sequential predictions as the value of the state variable frequency increases through the spectrum. Once trained, it can predict both the magnitude of asymmetry, $\left|\frac{R_2}{R_1}\right|_{j+1}$, as well as frequency, f_{j+1}, of the subsequent state, $j+1$, given an input graph $G_j(\boldsymbol{u}, V, E)$ at state j. A visual workflow of the GNN architecture is shown in Figure 6.5. The workflow uses arrows to indicate information flow between segments of the NN architecture. Solid line arrows represent the linear flow from graph-to-graph for each group of graph elements. Dot-dash line arrows represent information shared amongst NN segments such as aggregated information about the edge values provided as input to the node calculations as shown in Equations 6.4-6.5.

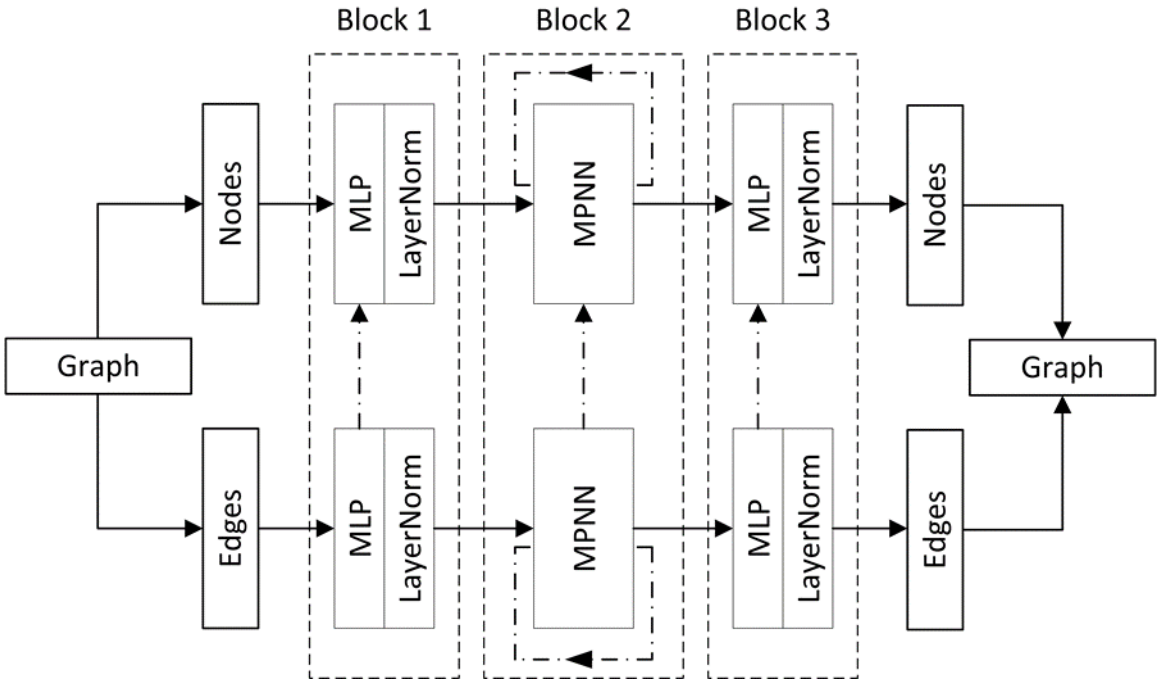

Figure 6.5: Architecture of the multi-block GNN metamodel. The graph is encoded before and after processing with the MPNN. The workflow uses arrows to indicate information flow between segments of the NN architecture. Solid line arrows represent the linear flow from graph-to-graph for each group of graph elements. Dot-dash line arrows represent information shared amongst NN segments. Edge update functions pass an aggregation of their outputs to the node update functions as shown in Equations 6.4-6.5.

The NN architecture is built using Tensorflow 1.15, sonnet 1.35, and graph-nets 1.0.4 software packages. Each of these is available for Python and provides robust functionality for defining multi-block NN architectures, training them, and testing them.

6.4.2 Metamodeling Asymmetrically Absorbing AMS

The AMSs are modeled as graphs with node, edge, and global attributes. Table 6.1 provides the attributes associated with each element. All attributes are detailed in Section 2.3 in Figure 2.6.

Graph Element	Attributes

Node, v	$[x_a, x_b, x_c, r_a, r_b, r_c, h]$
Edge, e	d_p
Global, u	$[f, \lvert R_2/R_1 \rvert]$

Table 6.1: Graph elements and their attributes for the AMS structured as a graph

With these attributes, the graphs are assembled to represent the connectivity of the inclusions in the AMS layer. Figure 6.6 provides illustration of this for a 2x2 super cell. A 3x3 super cell is connected in the same manner but with an expanded graph.

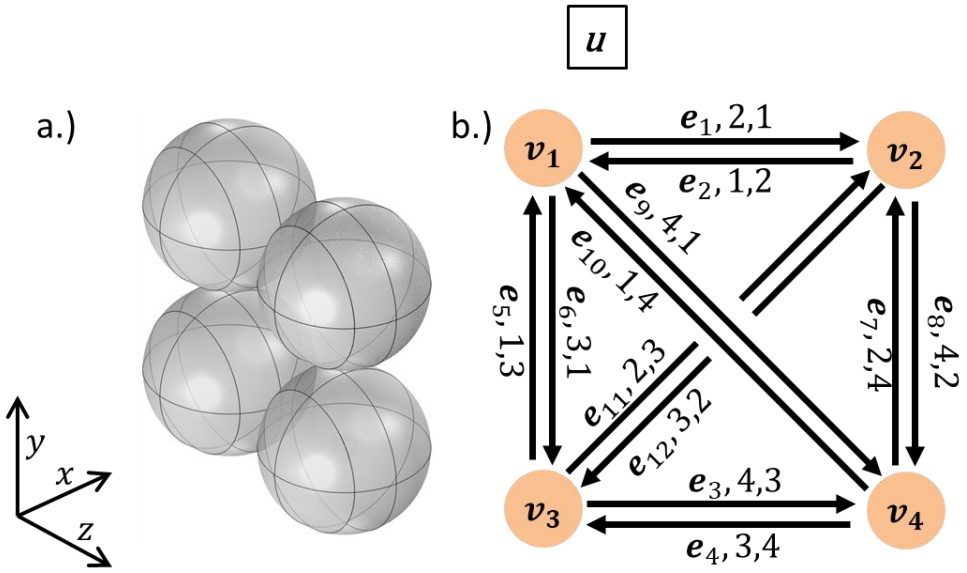

Figure 6.6: A 2x2 super cell of inclusions is defined on a graph. Part (a) illustrates the geometry used in the FEA while part (b) shows the graph structure of the geometry that is used for metamodeling with the GNN.

Generation of this graph is automated with script that organizes the attributes of each element into sets as defined in Equations 6.1-6.3. Assignment of sender and receiver nodes is also automated and requires only the number of inclusions as input. Once graphs are generated for each spatial arrangement in the training set, they are sent as inputs to the NN architecture. The process for operating on the graphs in training and testing the GNN

is covered in the section titled, *Training the GNN Model*, but first the process to generate data will be described.

Data Generation

Two separate GNN models are trained in this work with different training sets to demonstrate the two most advantageous characteristics of GNNs separately. The evaluated model characteristics are: 1) domain transduction and 2) interaction learning. Domain transduction involves predicting outputs for input domain dimensionalities that are present in the training set at a low occurrence rate or are not present in the training set at all. Interaction learning means the model is trained to represent spatial arrangements (graphs) that are not present in the training set so long as the input features are within the ranges of the training set. It is called interaction learning because the model learns interactions between nodes and edges agnostic to any specific connectivity. The two characteristics are considered separately to benchmark the model's predictive capability for each trait independently. The domain transduction model hereafter is referred to as GNN1 and the interaction learning model is referred to as GNN2 for brevity. It is important to note that both GNN1 and GNN2 use the same architecture and both are capable of both goals. Only the data used to train and test the models is different. From a sampling perspective, this strategy also reduces the computational expense to generate data. Table 6.2 provides the time require to generate a single frequency sample for each super cell size using a Linux machine with a two 12-core Intel Xeon Gold 5118 CPUs and 250 GB of RAM.

Super Cell	Minutes per Single Frequency Sample
1x1	0.7
2x2	3.7
3x3	11
4x4	15

Table 6.2: Sampling times for a single frequency of a supercell on a Linux machine with two 12-core Intel Xeon Gold 5118 CPUs and 250 GB of RAM.

The data for the GNN1 model reuses the sampling scheme in Section 5.4.2. In summary, the 7 geometric features of each inclusion, $v_i = [x_{a,i}, x_{b,i}, x_{c,i}, r_{a,i}, r_{b,i}, r_{c,i}, h_i]$, are sampled with a Halton sequence with each feature distributed uniformly over a $\pm 10\mu m$ range. Each design is then evaluated at 20 discrete frequencies from 124 kHz to 126 kHz. The same scheme is used to sample systems that are of size 1x1, 2x2, and 3x3. All inclusions are identical in the AMS layer. Data is split such that 1x1 systems represent $2/3$ of the training data ($N = 500$), and 2x2 systems account for $1/3$ of it ($N = 250$). Only a few 3x3 systems are evaluated, and they are not included in the training set, but rather used to test the generalizability of the model trained only on smaller systems.

Data for GNN2 represents solely 2x2 super cell systems with heterogeneous inclusion features. Initial exploration showed that asymmetric absorption performance was so sensitive to variability in inclusion features that peaks of asymmetric behavior sometimes moved outside the sampling frequency range. As a result, only h was varied for the training data set. The features were set at $[x_{a,i}, x_{b,i}, x_{c,i}, r_{a,i}, r_{b,i}, r_{c,i}] = [0, 0, 0, 500, 500, 500]\mu m$ and $h \in H \sim U(200 - 5, 200 + 5)\mu m$ and each design was sampled over 40 discrete frequencies from 124.5 kHz to 125.5 kHz. One hundred 2x2 super cell designs were sampled ($N = 100$) and Figure 6.7 shows ten of the $|R_2/R_1|$vs. f response curves from the FEA evaluated sample of heterogeneous 2x2 super cells.

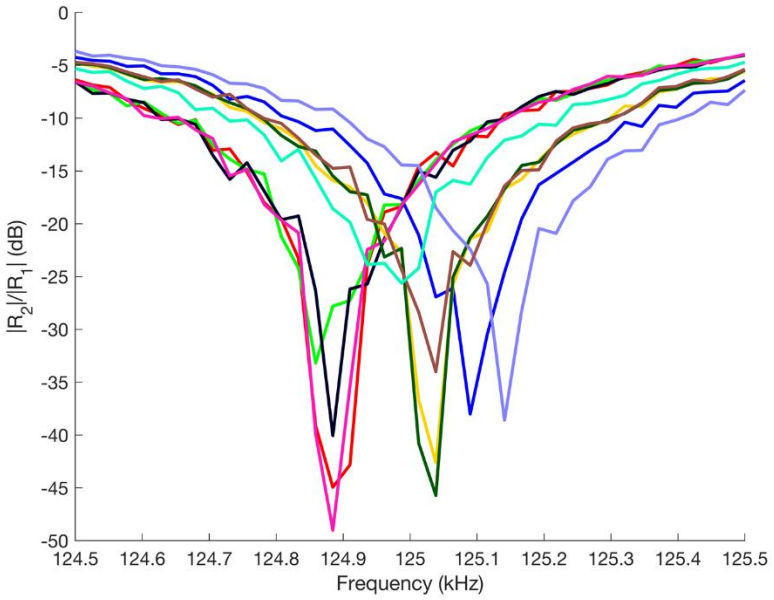

Figure 6.7: Ten asymmetric response curves for 2x2 super cell systems with heterogeneous inclusions. These responses are used to train GNN2 along with other members of the same sampling scheme.

Despite the increased sampling density of $f = 40$, these responses are noisier than the ones for homogenous AMS layers. Noise was considered when training the metamodels.

Training the GNN Metamodel

Both models are trained to update a graph from one frequency state to a subsequent state. All elements of the graph, $G(\boldsymbol{u}, V, E)$, are updated from state-to-state. To minimize error while training, a random state from the training set is selected as the input graph and the target output graph is the next state in the sequence (e.g. $G_{124kHz} \rightarrow G_{125kHz}$). The loss function to be optimized is the root mean square error (RMSE) of the state and magnitude values:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} r_i^2}{N}}, \tag{6.7}$$

$$r_i = \boldsymbol{u_i} - \boldsymbol{u_i}', \tag{6.8}$$

where $\boldsymbol{u_i}$ represents the predicted global variables values f and $|R_2/R_1|$ at one state and $\boldsymbol{u_i}'$ indicates the ground-truth values. Loss is minimized by backpropagation and gradient descent using the Adam algorithm [129] as the optimizer with a learning rate of 1×10^{-6}. Both models are trained over 3.5 million training iterations in batches of graphs per GNN. Table 6.3 contains an ordered presentation of the hyperparameter settings for training.

GNN Model Settings		
Parameter	GNN1	GNN2
Latent space size	16	16
MLP layers	2	2
N_{train}	3,500,000	3,500,00
Training Batch	200	10
Testing Batch	1	1
State per sample	20	40

Table 6.3: Parameters of the MPNNs trained as graph update functions

While training, the training loss is stored at regular intervals to evaluate convergence. Both models reach a very low error rate in about 250,000 training iterations and then decrease only slightly over the rest of training and show some variability. Although the error rate deviates some in late stages of training, its value remains very small and the model does not diverge. Figure 6.8 shows the training loss for both models during the training process.

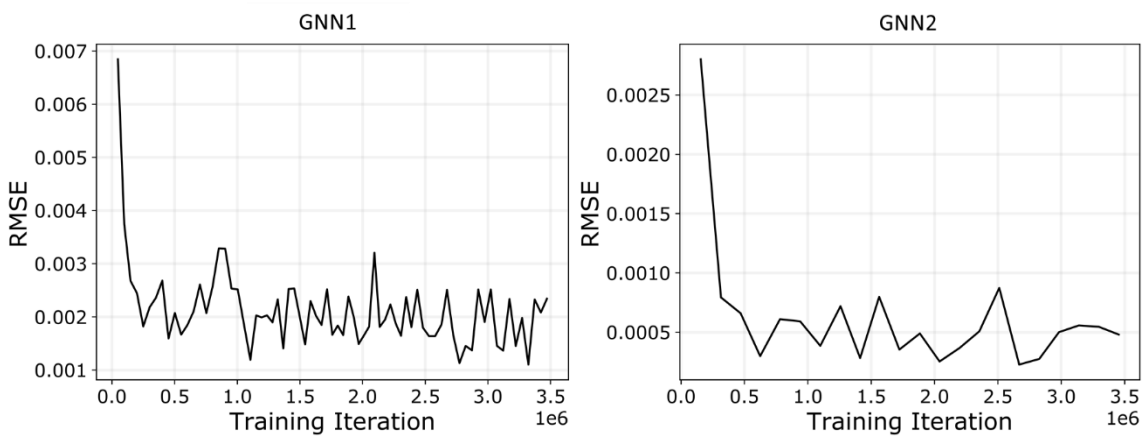

Figure 6.8: Error rate as a function of training iteration for both models, GNN1 and GNN2.

Once trained, the model is tested by making sequential predictions of both state and asymmetric performance produced by specifying the MPNN process as many times as the number of requested states (i.e., frequencies) beyond the current state. For example to predict the output at state 5 given state 1, the MPNN passes 4 output states back to its input sequentially. The AMS geometries used in testing were excluded from the training sets.

6.5 AMS Design with GNN Metamodeling Results

The trained GNNs generate trajectories of the systems' states and magnitudes of asymmetry. Predicted and ground-truth trajectories of all the test systems were stored and error was aggregated by calculating RMSE between the true and predicted values of f and $|R_2/R_1|$. Aggregate error for all predicted trajectories are presented in Table 6.4

	$RMSE_{freq}$ (kHz)	$RMSE_{R_2/R_1}$ (dB)
Domain Transduction GNN1	0.126	1.59
Interaction Learning GNN2	0.373	1.31

Table 6.4: Aggregate RMSE for all test trajectories separated by GNN model.

Test sets for GNN1 included ten 1x1 systems, five 2x2 systems, and three 3x3 systems. For GNN2 ten heterogeneous 2x2 systems were used for testing. In testing of GNN1, error varied amongst dimensionalities of the predicted systems. As expected, systems with dimensionalities that occurred less in the training set showed slightly higher error. Table 6.5 lists aggregated RMSE by system dimensionality.

		$RMSE_{freq}$ (kHz)	$RMSE_{R_2/R_1}$ (dB)
Test System Dimensionality	1x1	0.117	1.43
	2x2	0.124	1.55
	3x3	0.154	1.71

Table 6.5: Aggregate RMSE for test trajectories from GNN1 model separated by dimensionality of the system.

While error was low when aggregated, an important part of designing these AMSs is identifying the peaks of asymmetric behavior. To this end, predicted trajectories are plotted against the ground-truth response curves from FEA and error is measured along the curve. The non-aggregating error measures are absolute error (AE),

$$AE = |true - predicted|, \tag{6.9}$$

and relative percent error (%Error),

$$\%Error = \frac{|true-predicted|}{true} * 100. \tag{6.10}$$

The following figures present examples of the way specific system trajectories compare to their true responses. Predicted and true response curves are plotted for three systems from

the test set (Figures 6.9 and 6.11), and one of those models is plotted along with its non-aggregating error measures, AE, and $\%Error$ (Figures 6.10 and 6.12).

GNN1 Single System Trajectories

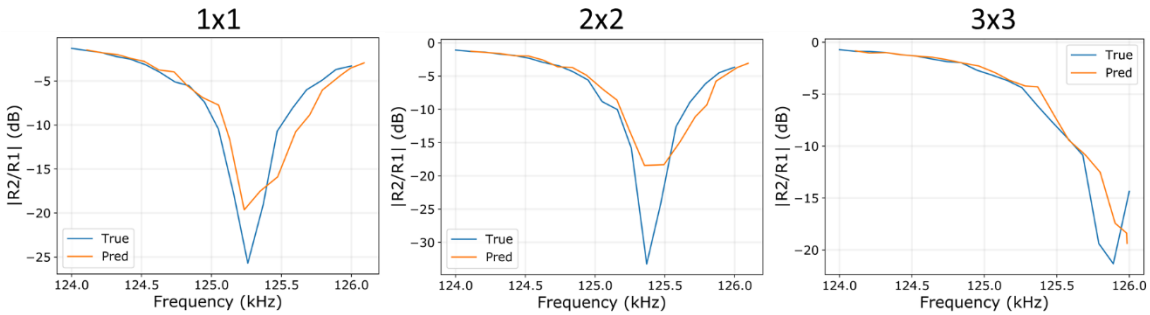

Figure 6.9: True and predicted response curves are plotted for randomly selected models from the GNN1 test set. One curve is provided for each different size of super cell.

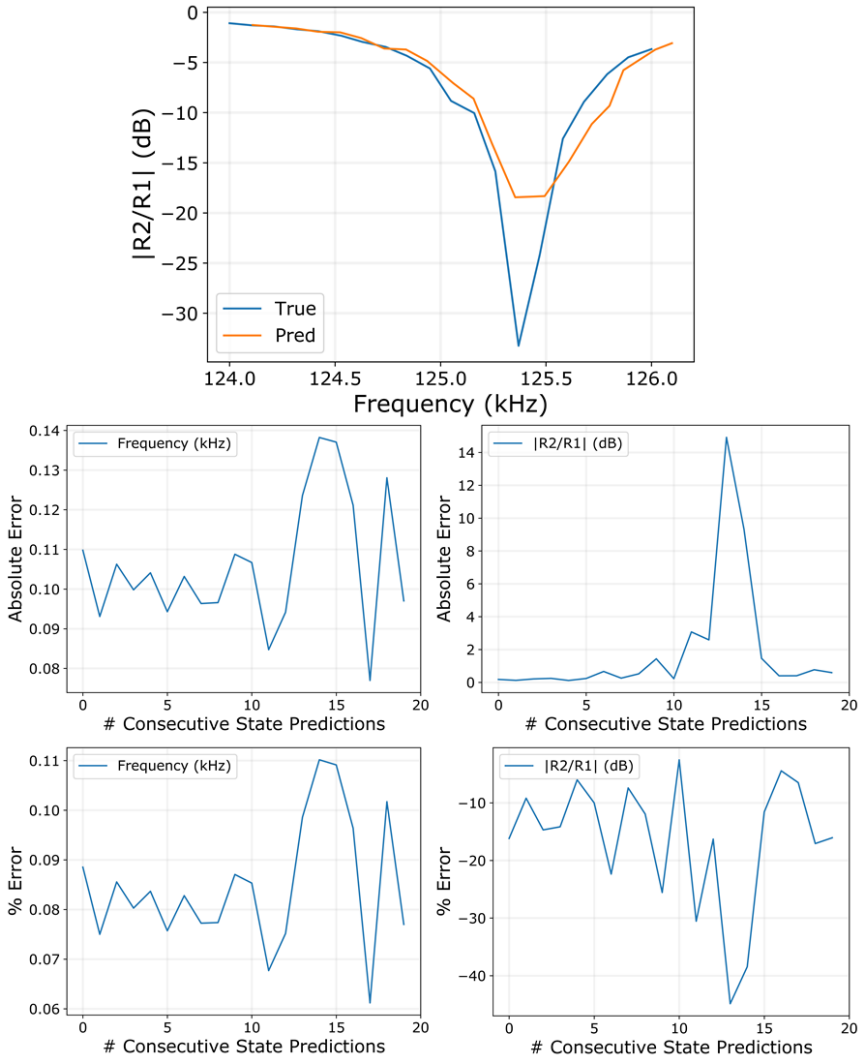

Figure 6.10: A predicted response from the GNN1 test set is plotted along with its non-aggregating error measures.

GNN2 Single System Trajectories

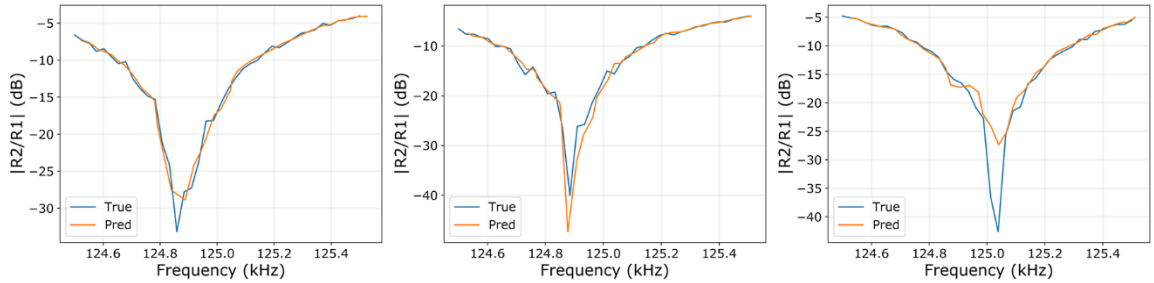

Figure 6.11: True and predicted response curves are plotted for randomly selected models from the GNN2 test set. Each curve represents a system with a different heterogeneous arrangement of inclusions.

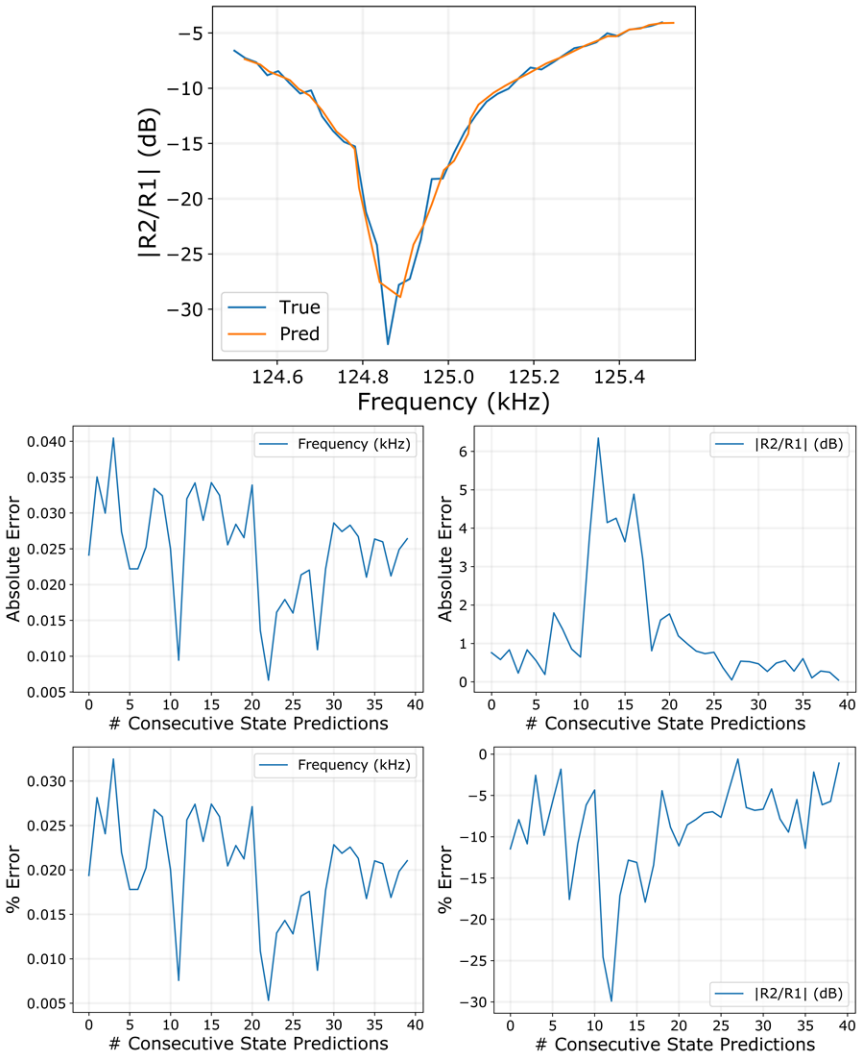

Figure 6.12: A predicted response from the GNN2 test set is plotted along with its non-aggregating error measures.

Results Discussion

Both GNN models are adept at identifying peaks of asymmetric response behavior. For some predictions the error measures indicate significant error around the peaks, but they still predict the peak at the correct frequency. Thresholding a magnitude of asymmetry

that qualifies as a peak of asymmetric behavior would allow a designer to automate peak identification even when using the GNN as the metamodel for evaluating this system. From the aggregate RMSEs in Table 6.4 and inspection of predicted response curves it is clear that the GNN2 predicts the magnitude of peaks with more accuracy than GNN1. This may be because the GNN does not have to generalize between different size domains but also could be explained by the quantity of variable node features in the training set of GNN1. Recall that GNN1 had seven variable design features per node while GNN2 had only one variable design feature per node.

There are ways this GNN metamodel could be improved. In this demonstration only one NN architecture was used. Architecture of the metamodel is essentially a collection of hyperparameters and can be either optimized with a cross-validation scheme or set heuristically based on prior knowledge or analogous implementation in the literature. Additionally, the attributes of nodes and edges could be changed so that they provide more information such as attributes that update dynamically. For example, edges could be defined as dynamically updating distances between each node. For metamodeling of an AMS with resonating inclusions like the one focused on here, relative node motion would likely be a strong indicator of absorptive behavior. This approach was not implemented here because the motion would be very small in magnitude and have to be extracted from the FEA. Because the excitation is steady state at a given frequency in the FEA, the value would be an average distance and be very close to the static distances used here. However, a time-domain analysis would provide time-series inclusion motion data and the GNN methodology is capable of updating across time states as well as frequency states.

Clearly GNNs are valuable tools for metamodeling an AMM system with a variable number of features and learning interactions between heterogeneous features. Training and evaluating a GNN model is more efficient than sole reliance on FEA, especially for design

exploration purposes even though it requires a nontrivial investment in generating training data.. In the AMS design use case presented here, the relative sampling, training, and evaluation costs are illustrative. The full training set for GNN1 required 17 days of computation and each 3x3 test system required 3.7 hours. Based on the speed of convergence shown in Figure 6.8, this much data was likely not necessary but $2/3$ of it was reused from the sampling in Section 5.4 and the data was generated while model development was in progress. Training the GNN took 7 hours and predicting each 3x3 trajectory took 2 seconds. To investigate the training data required, the model was trained again with only $1/4$ of the full data set (4.5 days to generate) and there was no loss of predictive capacity on 3x3 system test trajectories. The upfront FEA and GNN training is equivalent to evaluating 31 3x3 systems directly with FEA in this trial. While the upfront computational investment is substantial, the investment in metamodeling breaks even at 31 3x3 systems so using the GNN metamodel becomes the more efficient choice if a designer expects needing to evaluate more than that. A designer must weigh the cost and benefit of data acquisition and modeling in any computational design task and this method is no exception. Furthermore, data acquisition may be done in parallel with GNN training, testing, and optimizing which would enable early stopping once a sufficient model accuracy is achieved.

The results of this work suggest this implementation is capable of metamodeling AMSs with heterogeneous or functionally graded inclusion layers that exhibit broadband behavior as long as enough compute power is committed to generate a sufficient training set. Furthermore, this implementation is extensible to other metamaterial systems with a large difference in size scales across their domains that are not amenable to strict homogenization.

Chapter 7: Conclusions and Future Work

The body of work documented in this dissertation is focused on computational methods to automate and accelerate design processes for engineering acoustic metamaterials. The research tasks are listed below:

> *Task One: Improve sampling efficiency during design exploration in sparse design spaces with mixed (continuous, discrete, categorical) data types.*
>
> *Task Two: Evaluate the robustness of metamaterial performance to spatial variability induced by manufacturing processes in a computationally efficient way.*
>
> *Task Three: Establish an automated, simulation-based method to efficiently design metamaterials with a variable number of repeating features and functionally graded spatial properties.*

All three of these tasks have been addressed. Chapter 2 provided foundational background on the analysis of acoustic metamaterials (AMMs) including the analytical and numerical methods for the two types of AMMs used to demonstrate each research task. Chapter 3 provided context on computational design methods such as statistical and machine learning (ML) and differentiated between metamodeling and classification methods including brief histories and evaluation techniques for each. The first research task was addressed in Chapter 4 by developing the PCD-informed SMOTE algorithm and using it to improve design exploration in the imbalanced design space of a two-layer Willis material. The second task was satisfied in Chapter 5 by developing a workflow that replaces finite element analysis (FEA) with a metamodel and uses that metamodel to perform Monte Carlo analysis on the manufacturability of a directionally asymmetric absorber for an arbitrary manufacturing process. The third task was addressed in Chapter 6 by designing a graph neural network (GNN) architecture that is compatible with inputs of variable

dimensionality and can be trained to represent asymmetric acoustic absorbers with functionally graded spatial properties.

As a result of completing the three research tasks, contributions to the engineering design community have been made. These contributions are simultaneously applicable for efficient design of AMMs and extensible to other engineering design and computational automation objectives. The following section summarizes the research contributions.

7.1 Summary of Research Contributions

Completion of the research tasks has a tangible impact on AMM design. Each task directly affects a challenge that has limited AMM design thus far. Using products of the first task, metamaterial designers can take advantage of local metamodels that exploit frequency and/or time dependence to efficiently mitigate the design space sparsity caused by high dimensionality, computational expense, and discrete design variables. As a result, metamaterial designs can be discovered during design exploration more efficiently and stricter performance specifications can be satisfied. The process developed in the second task informs the design of reliably manufacturing metamaterials by providing designers a way to quantify the probability that as-built metamaterials will match the expected behavior of a precise design. It can also enable co-development of AMMs and the manufacturing processes necessary to realize them by providing bounds on acceptable spatial variation as goals for manufacturing process innovators. The final task addresses the challenge of designing heterogeneous micro-structures in cases where analytical homogenization is not feasible as a means of reducing computational expense. For these cases retraining metamodels for each unique dimensionality of a feature space is inefficient. This work removes that constraint and motivates and enables transduction between design spaces of differing dimensionality. All three of these tasks contribute to

the overarching goal of enabling data-driven design for applications in which intuition and brute force design methods are infeasible.

Satisfying the three main challenges also enables development beyond the motivating topic of AMM. Any generative design space mapping using a probabilistic discriminating metric can benefit from selective sampling to refine that space. The general idea of generating synthetic samples from a metamodel is extensible anywhere that they can be used to cheaply represent a black box function. Especially when data types are mixed, doing so will reduce the sampling burden and make design less expensive by more efficiently finding the promising regions of a design space. For the same expense minimization reason, network-based metamodels that accommodate design domains of variable size are valuable for numerous other applications. These models provide the most benefit when representing dynamic black box behavior of assemblies of constituent components. Although these techniques are focused on AMM, they are applicable for other classes of metamaterials as well. Perhaps models that are general with respect to spatial and temporal variables would encourage metamaterial development in disciplines that have otherwise ignored them by providing a way to overcome the obvious obstructions to identifying useful designs in a complicated multi-scale domain. Finally, processes that evaluate robustness of design for metamaterial manufacturing will be useful to guide process development toward specific performance goals. Whether developing multi-material processes or otherwise, determining build requirements requires a way of quantifying manufacturing variability's effect on a part's performance.

7.2 Opportunities for Future Work

The research contributions also provide opportunities for extension and applicability in many ways.

7.2.1 Adaptive Sampling for Generative Classifiers

Chapter 4 demonstrated a PCD-informed SMOTE using a BNC classifier for design space mapping. This technique is benchmarked against other leading SMOTE techniques from the literature but only on the two-layer Willis material design problem. While PCD-informed SMOTE performed slightly better than those other techniques on that problem, it would be interesting to benchmark it more broadly. Testing PCD-informed SMOTE and the other techniques on many design tasks with various common characteristics would elucidate its standing among SMOTE methods. If the new method performed well enough it could warrant development of an open-source software implementation and potentially be included in existing ML software distributions.

7.2.2 AMM Designs that are Robustly Manufacturable

The design process for evaluating robustness of as-built AMM designs used guidance from literature to specify manufacturing variability that could be expected when attempting to build the directionally asymmetric acoustic absorber. In particular, the best reported layer thicknesses and raster widths of aerosol deposition were used as the basis for assumptions of spatial feature variability distributions. Gaussian and beta distributions were built to simulate patterns of variability that are common to additive manufacturing processes such as mean-centered and biased variability. The obvious goal is to actually build an AMS and the work in Chapter 5 concluded some reasonable bounds for expected success rate of as-built parts. However, there are two main tasks to be completed before attempting to build. First a machine (or multiple) that is thought to be accurate enough to build the part with a reasonable success rate must be identified and then a metrology study must be conducted to characterize the variability distributions of as-built spatial features with respect to the design. A coordinate measurement machine could be used to create a

distribution of variability for each spatial feature of interest. Once the prospective manufacturing machine is characterized, the analysis presented in Chapter 5 should be conducted using the machine specific distributions to evaluate a real as-built success rate for that part. If satisfactory, a run of AMS parts could be made and tested. Experimental results from this process would help validate and/or improve the numerical analysis of AMS performance.

7.2.3 Graph Neural Networks for Metamaterial Design

The results discussion of Chapter 6 suggested trying different attributes for the nodes and edges of the graph that models the AMS system and optimizing the NN architecture. Both of these efforts have the potential to improve the accuracy of the GNN metamodel. In addition, the clear forward path is to continue work toward identifying broadband asymmetric behavior of this type of structure and attempt to efficiently design for it. There are two main thrusts of this effort. The first is improving the acoustic analysis so that the underlying physics are better understood. Of particular interest is how many hertz must separate the local resonance frequencies of localized homogenous inclusion groupings. Initial exploration showed that small groupings of at least four adjacent inclusions needed to be identical to create the local resonant behavior that leads to macro-scale asymmetric absorption. However, local groupings of inclusions that were too similar in geometry to other local groupings in the AMS layer would cause interference that reduced or negated macro-scale asymmetry. Gaining understanding of this topic would constrain and guide the search for AMSs that exhibit broadband asymmetric absorption. That information is essential to the second thrust, which is acquiring enough compute time to sample the design space of 4x4 and 5x5 super cells with sufficient resolution in the frequency spectrum to identify peaks of asymmetric behavior. Because the minimum super

cell size to identify broadband asymmetric absorption requires a great deal of computation, the GNN methodology presented in Chapter 6 will be invaluable to assist exploration for broadband asymmetry.

7.3 CLOSURE

Computational power is ever increasing but Moore's empirical law of integrated circuit capacity no longer holds. Meanwhile, demand for computational power is accelerating in large part due to the advent of deep learning. Under these circumstances improving the efficiency of computational design is crucial to continue the pace of development of new, interesting, and useful engineering systems. Even if the circumstances change and computational power tracks a new path of greatly accelerating improvement, no resource, including computing power, is ever infinite. For this reason, the pursuit of computational efficiency in design exploration will always be a worthwhile endeavor.

Appendix: Nomenclature

The following table presents nomenclature used in the dissertation. Terms are listed by chapter as they are introduced. Later chapters may utilize previously introduced terms.

Chapter 1	
S	Scattering matrix
p_i^+, p_i^-	Incident pressure waves travelling in the $\pm x$-directions
p_o^+, p_o^-	Outgoing pressure waves travelling in the $\pm x$-directions
R	Reflection coefficient
T	Transmission coefficient
α	Absorption coefficient
$\bar{\alpha}$	Ratio of asymmetric absorption
λ	Wavelength
v	Wave propagation speed
f	Frequency
Chapter 2	
$\boldsymbol{\mu}$	Momentum density
$\boldsymbol{u}$	Particle velocity
ϵ	Volume strain
p	Acoustic pressure
$\underline{\boldsymbol{\rho}}$	Anisotropic mass density
β	Adiabatic compressibility
$\boldsymbol{\chi}^o$	Odd coupling
$\boldsymbol{\chi}^e$	Even coupling
k	Wavenumber
a	Characteristic feature size
ρ	Material density

c	Sound speed in a material
L	Unit cell length in the Willis material
l	Heterogeneity length in the Willis material
Z	Characteristic acoustic impedance
ω	Frequency in radians
$\boldsymbol{n}$	Unit normal vector
l_e	Characteristic element size
p_s	Scattered pressure
α_m	Monopole polarizability
$\underline{\boldsymbol{\alpha}}_d$	Dipole polarizability
$\boldsymbol{\alpha}_c$	Coupled polarizability
$\tilde{\alpha}_m$	Complementary monopole polarizability
$\underline{\tilde{\boldsymbol{\alpha}}}_d$	Complementary dipole polarizability
$\tilde{\boldsymbol{\alpha}}_c$	Complementary coupled polarizability
Λ_m	Monopole fields
$\underline{\boldsymbol{\Lambda}}_d$	Dipole fields
$\boldsymbol{\Lambda}_c^o$	Even coupled fields
$\boldsymbol{\Lambda}_c^e$	Odd coupled fields
d_p	Square-packed separation distance of in-plane inclusions
M	Complex plane wave modulus
$\boldsymbol{F}_v$	Body force
S_{pk}	Second Piola-Kirchhoff stress tensor
$\boldsymbol{F}_A$	Force per unit area
$\boldsymbol{u}_{tt}$	Acceleration
$r_\mathrm{a}, r_\mathrm{b}, r_\mathrm{c}$	Ellipsoidal radii
h	Distance from center of ellipsoid to material interface plane
η_s	Structural loss parameter

E	Young's modulus
ν	Poisson's ratio
Chapter 3	
$y(x, \boldsymbol{w})$	Functional form of single variable polynomial regression model
w_j	Scalar term weight coefficients of polynomial regression model
N	Quantity of design instances
D	Dimensionality
$S_p(n)$	Morokoff and Caflisch's Halton sequence expression
$\boldsymbol{x}_n$	n-length sequence of design points
$\boldsymbol{x}$	Ddimensional array of n design instances
(a, b)	Minimum and maximum values for scaling sequence of numbers
$\boldsymbol{x}'$	Transformed Ddimensional array of n design instances
μ	Mean
σ	Standard deviation
c_l	Class number l assignment of design instances
$p(c\|x)$	Conditional probability of the class given the candidate design, x
P	Member of the positive class
N	Member of the negative class
TP	True positive classifier prediction
FP	False positive classifier prediction
TN	True negative classifier prediction
FN	False negative classifier prediction
ACC	Classifier accuracy
TPR	True positive rate
FPR	False positive rate
FNR	False negative rate
λ_l	Bayesian network classifier heuristic probability weighting
PCD	Posterior class discriminant

ϵ	Error between two functions
ϕ_j	Basis function
RSS	Residual sum of squares
$RMSE$	Root mean square error
MAE	Mean absolute error
f'	Gradient of a function f
α	Bias of neural network
w	Weight of neural network
ϕ	Activation function of neural network
Chapter 4	
$\boldsymbol{s}_i$	Synthetic sample design instance
λ_s	Randomly sampled number used to add noise in ADASYN procedure
$\sigma_{i,l}$	Kernel width parameter
$\hat{\sigma}_{i,l}$	St. dev. of design variable i for designs belonging to class l
α	Kernel width scaling hyperparameter
Chapter 5	
d	Diameter of roughly spherical inclusions
$\boldsymbol{T} \in \boldsymbol{\Omega}$	AM process random variable set
$\boldsymbol{z} \in \mathbb{R}^3$	Spatial 3D position in build chamber
$\bar{R}$	Asymmetric reflection ratio
Chapter 6	
G	Graph representation
V	Set of node attributes in a graph representation
E	Set of edge attributes in a graph representation
$\boldsymbol{u}$	Set of global attributes of an entire graph
$\boldsymbol{v}_i$	Single node's set of attributes
$\boldsymbol{e}_i$	Single edge's set of attributes
N_m	Quantity of nodes

N_e	Quantity of edges
r_k	Index of receiving node
s_k	Index of sending node
$\boldsymbol{e}'_k$	Updated edge attribute set representing the subsequent state
$\boldsymbol{v}'_i$	Updated node attribute set representing the subsequent state
$\boldsymbol{u}'$	Updated global attribute set representing the subsequent state
ϕ_b	Graph element update function
AE	Absolute error measure
$\%Error$	Relative percent error measure

Table A: Nomenclature used in the dissertation

References

[1] K. Bertoldi, V. Vitelli and J. Christensen, "Flexible mechanical metamaterials," *Nat Rev Mater,* vol. 2, p. 17066, 2017.

[2] K.-H. Chang and P.-S. Tang, "Integration of design and manufacturing for structural shape optimization," *Advances in Engineering Software,* vol. 32, pp. 555-567, 2001.

[3] A. Hussein, L. Hao, C. Yan, R. Everson and P. Young, "Advanced lattice support structures for metal additive manufacturing," *J Mat Proc Tech,* vol. 213, pp. 1019-1026, 2013.

[4] N. Aage, E. Andreassen, B. Lazarov and O. Sigmund, "Giga-voxel computational morphogenesis for structural design," *Nature,* vol. 550, pp. 84-86, 2017.

[5] J. U. Surjadi, L. Gao, H. Du, X. Li, X. Xiong, N. X. Fang and Y. Lu, "Mechanical Metamaterials and Their Engineering Applications," *Adv Eng Mat,* vol. 21, p. 1800864, 2019.

[6] A. G. Evans, J. W. Hutchinson, N. A. Fleck, M. F. Ashby and H. Wadley, "The topological design of multifunctionalcellular metals," *Progress in Materials Science,* vol. 46, pp. 309-327, 2001.

[7] C. Liu, Z. Du, W. Zhang, Y. Zhu and X. Guo, "Additive Manufacturing-Oriented Design of Graded Lattice Structures Through Explicit Topology Optimization," *J. Appl. Mech.,* vol. 84, no. 8, p. 081008, 2017.

[8] M. Haberman and M. Guild, "Acoustic Metamaterials," *Physics Today,* vol. 69, no. 6, p. 42, 2016.

[9] S. Cummer, J. Christensen and A. Alu, "Controlling Sound with Acoustic Metamaterials," *Nat Rev Mater,* vol. 1, p. 16001, 2016.

[10] M. H. Haberman and A. N. Norris, "Acoustic Metamaterials," *Acoustics Today,* vol. 12, no. 3, pp. 31-39, 2016.

[11] Y. Wu, M. Yang and P. Sheng, "Perspective: Acoustic metamaterials intransition," *J. Appl. Phys.,* vol. 123, p. 090901, 2018.

[12] B.-I. Popa, L. Zigoneanu and S. A. Cummer, "Tunable active acoustic metamaterials," *Phys. Rev. B,* vol. 88, p. 024303, 2013.

[13] B.-I. Popa, Y. Zhai and H.-S. Kwon, "Broadband sound barriers with bianisotropic metasurfaces," *Nat. Comm.,* vol. 9, p. 5299, 2018.

[14] B. M. Goldsberry, S. P. Wallen and M. H. Haberman, "Non-reciprocal Wave Propagation in Mechanically Modulated Continuous Elastic Metamaterials," *J. Acoust. Soc. Am.,* vol. 146, no. 1, pp. 782-788, 2019.

[15] S. A. Cummer, J. Christensen and A. Alu, "Controlling Sound with Acoustic Metamaterials," *Nat. Rev. Mat.,* vol. 1, p. 16001, 2016.

[16] M. H. Haberman and M. Guild, "Acoustic metamaterials," *Physics Today,* vol. 69, no. 6, p. 42, 2016.

[17] M. Muhlestein, C. Sieck, P. Wilson and M. Haberman, "Experimental evidence of Willis coupling in a one-dimensional effective material element," *Nat Comm,* vol. 8, p. 15625, 2017.

[18] S. Koo, C. Cho, J.-h. Jeong and N. Park, "Acoustic omni meta-atom for decoupled access to all octants of a wave parameter space," *Nat Comm,* vol. 7, p. 13012, 2016.

[19] R. Fleury, D. Sounas, C. Sieck, M. Haberman and A. Alu, "Sound Isolation and Giant Linear Nonreciprocity in a Compact Acoustic Circulator," *Science,* vol. 343, no. 6170, pp. 516-519, 2014.

[20] H. e. a. Nassar, "Nonreciprocity in acoustic and elastic materials," *Nature Review Materials,* vol. 5, pp. 667-685, 2020.

[21] G. Ma and P. Sheng, "Acoustic Metamaterials: From Local Resonances to Broad Horizons," *Sci. Adv.,* vol. 2, no. 2, p. e1501595, 2016.

[22] M. Yang, C. Meng, C. Fu, Y. Li, Z. Yang and P. Sheng, "Subwavelength Total Acoustic Absorption with Degenerate Resonators," *Appl. Phys. Lett.,* vol. 107, p. 104104, 2015.

[23] A. N. Norris, "Acoustic Cloaking," *Acoust. Today,* vol. 11, no. 1, pp. 38-46, 2015.

[24] Y. Chen and L. Wang, "Periodic co-continuous acoustic metamaterials with overlapping locally resonant and Bragg band gaps," *Appl. Phys. Lett.,* vol. 105, p. 191907, 2014.

[25] T. Lee and H. Iizuka, "Bragg scattering based acoustic topological transition controlled by local resonance," *Phys. Rev. B,* vol. 99, p. 064305, 2019.

[26] A. O. Krushynska, V. G. Kouznetsova and M. Geers, "Towards optimal design of locally resonant acoustic metamaterials," *Journal of the Mechanics and Physics of Solids,* vol. 71, pp. 179-196, 2014.

[27] C. F. Sieck, A. Alù and M. R. Haberman, "Origins of Willis Coupling (Bianisotropy) in Acoustic Metamaterials through Source-Driven Homogenization," *Phys. Rev. B.,* vol. 96, no. 10, p. 104303, 2017.

[28] H. Long, Y. Cheng and X. Liu, "Asymmetric absorber with multiband and broadband for low-frequency sound," *Appl. Phys. Lett.,* vol. 111, p. 143502, 2017.

[29] A. Merkel, G. Theocharis, O. Richoux, V. Romero-Garcia and V. Pagneux, "Control of acoustic absorption in one-dimensional scattering by resonant scatterers," *Appl. Phys. Lett.,* vol. 107, p. 244102, 2015.

[30] A. Norris and P. Packo, "Non-symmetric flexural wave scattering and one-way extreme absorption," *The Journal of the Acoustical Society of America,* vol. 146, p. 873, 2019.

[31] B.-I. Popa and S. A. Cummer, "Non-reciprocal and Highly Nonlinear Active Acoustic Metamaterials," *Nat. Comm.,* vol. 5, p. 3398, 2014.

[32] M. D. Guild, C. A. Rohde, M. C. Rothko and C. F. Sieck, "3D printed acoustic metamaterial sound absorbers using functionally-graded sonic crystals," in *Euronoise*, Crete, 2018.

[33] B. Yuan, V. F. Humphrey, J. Wen and X. Wen, "On the coupling of resonance and Bragg scattering effects in three-dimensional locally resonant sonic materials," *Ultrasonics*, vol. 53, no. 7, pp. 1332-1343, 2013.

[34] C. Sharpe, T. J. Wiest, P. Wang and C. C. Seepersad, "A Comparative Evaluation of Supervised Machine Learning Classification Techniques for Engineering Design Applications," *Journal of Mechanical Design*, vol. 141, no. 12, p. 121403, 2019.

[35] C. Bishop, Pattern Recognition and Machine Learning, New York: Springer, 2006.

[36] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[37] B.-I. Popa, L. Zigoneau and S. A. Cummer, "Experimental Acoustic Ground Cloak in Air," *Phys. Rev. Lett.*, vol. 106, p. 253901, 2011.

[38] C. W. Cushing, P. S. Wilson, M. R. Haberman, C. Shen, J. Li, S. A. Cummer, Z. J. Tan, C. Ma, H. Du and N. X. Fang, "Characterization of an underwater metamaterial made of aluminum honeycomb panels at low frequencies," *J. Acoust. Soc. Am.*, vol. 149, p. 1829, 2021.

[39] Y. Chen, M. Zheng, X. Liu, Y. Bi, Z. Sun, P. Xiang, J. Yang and G. Hu, "Broadband solid cloak for underwater acoustics," *Phys. Rev. B*, vol. 95, p. 180104(R), 2017.

[40] X. Su, A. N. Norris, C. W. Cushing, M. R. Haberman and P. S. Wilson, "Broadband focusing of underwater sound using a transparent pentamode lens," *J. Acoust. Soc. Am.*, vol. 141, p. 4408, 2017.

[41] M. Muhlestein, C. Sieck, P. Wilson and M. Haberman, "Experimental evidence of Willis coupling in a one-dimensional effective material element," *Nat. Comm.*, vol. 8, p. 15625, 2017.

[42] S. Koo, C. Cho, J.-h. Jeong and N. Park, "Acoustic omni meta-atom for decoupled access to all octants of a wave parameter space," *Nat. Comm.*, vol. 7, p. 13012, 2016.

[43] J. Li, C. Shen, A. Diaz-Rubio, S. A. Tretyakov and S. A. Cummer, "Systematic design and experimental demonstration of bianisotropic metasurfaces for scattering-free manipulation of acoustic wavefronts," *Nat. Comm.*, vol. 9, p. 1342, 2018.

[44] L. Quan, Y. Ra'di, D. L. Sounas and A. Alu, "Maximum Willis Coupling in Acoustic Scatterers," *Phys. Rev. Lett.*, vol. 120, p. 254301, 2018.

[45] H. Nassar, B. Yousefzadeh, R. Fleury, M. Ruzzene, A. Alu, C. Daraio, A. N. Norris, G. Huang and M. R. Haberman, "Nonreciprocity in acoustic and elastic materials," *Nat. Rev. Mat.*, vol. 5, pp. 667-685, 2020.

[46] J. R. Willis, "The nonlocal influence of density variations in a composite," *International Journal of Solids and Structures Topics in Continuum Mechanics,* vol. 21, no. 7, pp. 805-817, 1985.

[47] J. A. Kong, "Theorems of bianisotropic media," *Proceedings of the IEEE,* vol. 60, no. 9, pp. 1036-1046, 1972.

[48] F. Ma, M. Huang and J. H. Wu, "Ultrathin lightweight plate-type acoustic metamaterials with positive lumped coupling resonant," *J. Appl. Phys.,* vol. 121, p. 015102, 2017.

[49] F. Ma, M. Huang, Y. Xu and J. H. Wu, "Bi-layer plate-type acoustic metamaterials with Willis coupling," *J. Appl. Phys.,* vol. 123, p. 035104, 2018.

[50] M. B. Muhlestein, B. M. Goldsberry, A. N. Norris and M. R. Haberman, "Acoustic scattering from a fluid cylinder with Willis constitutive properties," *Proceedings of the Royal Society A,* vol. 474, no. 2220, p. 20180571, 2018.

[51] Y. Liu, Z. Liang, J. Zhu, L. Xia, O. Mondain-Monval, T. Brunet, A. Alu and J. Li, "Willis Metamaterial on a Structured Beam," *Phys. Rev. X.,* vol. 9, p. 011040, 2019.

[52] M. B. Muhlestein and M. R. Haberman, "A micromechanical approach for homogenization of elastic metamaterials with dynamic microstructure," *Proc. R. Soc. A,* vol. 472, p. 0438, 2016.

[53] M. B. Muhlestein and C. M. Mauck, "Optimization of one-dimensional Willis materials with hidden lumped elements," *Proc. Mtgs. Acoust.,* vol. 39, p. 045003, 2019.

[54] R. W. Pryor, Multiphysics Modeling Using COMSOL: A First Principles Approach, Sudbury, MA: Jones and Bartlett Publishers, 2011.

[55] D. F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The Computer Journal,* vol. 24, no. 2, pp. 167-172, 1981.

[56] O. Schenk and K. Gartner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *Future Generation Computer Systems,* vol. 20, no. 3, pp. 475-487, 2004.

[57] B. Assouar, B. Liang, Y. Wu, Y. Li, J.-C. Cheng and Y. Jing, "Acoustic Metasurfaces," *Nature Reviews Materials,* vol. 3, pp. 460-472, 2018.

[58] D. T. Blackstock, Fundamentals of Physical Acoustics, New York: John Wiley & Sons, 2000.

[59] A. Norris and P. Packo, "Non-symmetric flexural wave scattering and one-way extreme absorption," *true J. Acoust. Soc. Am.,* vol. 146, p. 873, 2019.

[60] C. Hakoda, J. Rose, P. Shokouhi and C. Lissenden, "Using Floquet Periodicity to Easily Calculate Dispersion Curves and Wave Structures of Homogeneous Waveguides," *AIP Conf. Proc.,* vol. 1949, p. 020016, 2018.

[61] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.,* vol. 114, no. 2, pp. 185-200, 1994.

[62] M. Zampolli, A. Tesei and F. B. Jensen, "A computationally efficient finite element model with perfectly matched layers applied to scattering from axially symmetric objects," *J. Acoust. Soc. Am,* vol. 122, p. 172, 2007.

[63] J. Norman, "Ultrasonic Dry Coupling through Tissue," *Canadian Acoustics,* vol. 43, no. 3, 2015.

[64] R. M. Guillermic, M. Lanoy, A. Strybulevych and J. H. Page, "A PDMS-based broadband acoustic impedance matched material for underwater applications," *Ultrasonics,* vol. 94, pp. 152-157, 2019.

[65] K. H. Lee, B. A. Szajewski, Z. Hah, K. J. Parker and A. M. Maniatty, "Modeling shear waves through a viscoelastic medium induced by acoustic radiation force," *International Journal for Numerical Methods in Biomedical Engineering,* vol. 28, no. 6-7, pp. 678-696, 2012.

[66] J. Hyun and H. A. Kium, "Transient level-set topology optimization of a planar acoustic lens working with short-duration pulse," *J. Acoust. Soc. Am.,* vol. 149, p. 3010, 2021.

[67] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Ann. Eugen.* , vol. 7, no. 2, pp. 179-188, 1936.

[68] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 141-142, 2012.

[69] A. K. Sharma and S. Sahni, "A Comparative Study of Classification Algorithms for Spam Email Data Analysis," *Intl. J. Comp. Sci. Engr.,* vol. 3, no. 5, pp. 1890-1895, 2011.

[70] M. Fernandez-Delgado, M. S. Sirsat, E. Cernadas, S. Alawadi, S. Barro and M. Febrero-Bande, "An Extensive Survey of Regression Models," *Neural Networks,* vol. 111, pp. 11-34, 2019.

[71] H. Borchani, G. Varando, C. Bielza and P. Larranaga, "A Survey on Multi-output Regression," *Wiley Interdisciplinary Review: Data Mining and Knowledge Discovery,* vol. 5, no. 5, pp. 216-223, 2015.

[72] Scikit-learn Developers, "Scikit-learn v1.0 User Guide," 2021. [Online]. Available: https://scikit-learn.org/stable/user_guide.html. [Accessed September 2021].

[73] J. Mendes-Moreira, C. Soares, A. M. Jorge and J. F. De Sousa, "Ensemble Approaches for Regression: A Survey," *ACM Computing Surveys,* vol. 45, no. 1, pp. 1-40, 2012.

[74] D. Sobek, A. Ward and J. Liker, "Toyota's Principles of Set-Based Concurrent Engineering," *Sloan Management Review,* vol. 40, no. 2, pp. 67-84, 1999.

[75] D. W. Shahan and C. C. Seepersad, "Bayesian Network Classifiers for Set-Based Collaborative Design," *Journal of Mechanical Design,* vol. 134, no. 7, p. 071001, 2012.

[76] E. Galvan, R. Malak, S. Gibbons and R. Arroyave, "A Constraint Satisfaction Algorithm for the Generalized Inverse Phase Stability Problem," *Journal of Mechanical Design,* vol. 139, no. 1, 2016.

[77] W. Chen and M. Fuge, "Beyond the Known: Detecting Novel Feasible Domains Over an Unbounded Design Space," *J. Mech. Des.,* vol. 139, no. 11, 2017.

[78] E. Galvan and R. Malak, "P3GA: An Algorithm for Technology Characterization," *Journal of Mechanical Design,* vol. 137, no. 1, 2015.

[79] R. Malak and C. Paredis, "Using Support Vector Machines to Formalize the Valid Input Domain of Predictive Models in Systems Design Problems," *Journal of Mechanical Design,* vol. 132, no. 10, 2010.

[80] P. Backlund, D. Shahan and C. Seepersad, "Classifier-guided Sampling for Discrete Variable, Discontinuous Design Space Exploration: Convergence and Computational Performance," *Engineering Optimization,* vol. 47, no. 5, pp. 579-600, 2015.

[81] G. G. Wang and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *J Mech Des,* vol. 129, no. 4, pp. 370-380, 2007.

[82] L. Gu, "“A Comparison of Polynomial Based Regression Models in Vehicle Safety Analysis," in *Proc. 2001 ASME DETC-DAC,* Pittsburgh, PA, 2001.

[83] R. H. Myers and D. Montgomery, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Toronto: Wiley, 1995.

[84] B. Talgorn, C. Audet, S. Le Digabel and M. Kokkolaras, "Locally weighted regression models for surrogate-assisted design optimization," *Optim. and Engr.,* vol. 19, pp. 213-238, 2018.

[85] J. E. Dennis and V. Torczon, "Managing Approximation Models in Optimization," *Multidisc. Des. Optim.: State-of-the-art,* vol. 5, pp. 330-347, 1997.

[86] I. G. Osio and C. H. Amon, "An Engineering Design Methodology with Multistage Bayesian Surrogates and Optimal Sampling," *Res. Eng. Des.,* vol. 8, no. 4, pp. 189-206, 1996.

[87] A. Shokry and A. Espuna, "Sequential dynamic optimization of complex nonlinear processes based on kriging surrogate models," *Procedia Technology,* vol. 15, pp. 376-387, 2014.

[88] C. Jiang, X. Cai, H. Qiu, L. Gao and P. Li, "A Two-stage Support Vector Regression Assisted Sequential Sampling Approach for Global Metamodeling," *Struct. Multidisc. Optim.,* vol. 58, pp. 1657-1672, 2018.

[89] L. Wang, S. Shan and G. G. Wang, "Mode-Pursuing Sampling Method for Global Optimization on Expensive Black-box Functions," *Eng. Optim.*, vol. 36, no. 4, pp. 419-438, 2004.

[90] S. Shan and G. G. Wang, "An Efficient Pareto Set Identification Approach for Multi-objective Optimization on Black-box Functions," *J. Mech. Des.*, vol. 127, pp. 866-874, 2005.

[91] M. D. Morris, T. J. Mitchell and D. Ylvisaker, "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, vol. 35, no. 3, pp. 243-255, 1993.

[92] D. M. Hawkins, "The Problem of Overfitting," *J. Chem. Inf. and Comput. Sci.*, vol. 44, no. 1, pp. 1-12, 2004.

[93] H. Jabbar and R. Z. Khan, "Methods to Avoid Over-fitting and Underfitting in Supervised Machine Learning," *Comput. Sci., Comm., and Instr. Devs.*, pp. 163-172, 2015.

[94] T. W. Simpson, J. D. Poplinski, P. N. Koch and J. K. Allen, "Metamodels for Computer-based Engineering Design: Survey and recommendations," *Engineering with Computers*, vol. 17, pp. 129-150, 2001.

[95] T. W. Simson, D. K. J. Lin and W. Chen, "Sampling Strategies for Computer Experiments: Design and Analysis," *Intl. J. Reliab. and Appl.*, vol. 2, no. 3, pp. 209-240, 2001.

[96] M. Sugiyama, "Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error," *J. Mach. Learn. Res.*, vol. 7, pp. 141-166, 2006.

[97] W. J. Morokoff and R. E. Caflisch, "Quasi-Random Sequences and Their Discrepancies," *SIAM Journal on Scientific Computing*, vol. 15, no. 6, pp. 1251-1279, 1994.

[98] J. H. Halton, "Algorithm 247: Radical-inverse Quasi-random Point Sequence," *Comm. ACM*, vol. 7, no. 12, pp. 701-702, 1964.

[99] I. M. Sobol, "Distribution of points in a cube and approximate evaluation of integrals," *U.S.S.R. Comput. Maths. Math. Phys.*, vol. 7, pp. 86-112, 1967.

[100] H. Faure, "Discrépance de Suites Associées a un Système de Numération (en Dimension s)," *Acta Arithmetica*, vol. 41, pp. 337-351, 1982.

[101] M. D. McKay, R. J. Beckman and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239-245, 1979.

[102] A. Sivakumar, C. R. Bhat and G. Okten, "Simulation Estimation of Mixed Discrete Choice Models with the Use of Randomized Quasi-Monte Carlo Sequences: A Comparative Study," *Journal of the Transportation Research Board*, vol. 1921, no. 1, pp. 112-122, 2005.

[103] A. Zheng and A. Casari, Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, O'Reilly Media, Inc., 2018.

[104] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ICMLR*, pp. 448-456, 2015.

[105] S. Arlot and A. Celisse, "A Survey of Cross-validation Procedures for Model Selection," *Statist. Surv.*, vol. 4, pp. 40-79, 2010.

[106] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013.

[107] A. Y. Ng and M. I. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes," *Advances in neural information processing systems*, pp. 841-848, 2002.

[108] G. Louppe, "Understanding Random Forests from Theory to Practice (PhD Dissertation)," University of Liege, Liege, Wallanio, Belgium, 2014.

[109] C. Elkan, "Boosting and Naive Bayesian Learning," University of California, San Diego, San Diego, California, 1997.

[110] N. Saliya, T. Khoshgotaar and J. Van Hulse, "A Study on the Relationships of Classifier Performance Metrics," in *IEEE International Conference on Tools with Artificial Intelligence*, Newark, New Jersey, USA, 2009.

[111] I. Rish, "An Empirical Study of the Naive Bayes Classifier," in *Proceedings of IJCAI-01 workshop on emprical methods* , 2001.

[112] M. Zhang, J. Pena and V. Robles, "Feature Selection for Multi-label Naive Bayes Classification," *Journal of Information Science*, vol. 179, no. 19, pp. 3218-3229, 2009.

[113] D. W. Scott, Multivariate Density Estimation, New York: John Wiley & Sons Inc., 1992.

[114] A. Perez, P. Larranga and I. Inza, "Bayesian Classifiers Based on Kernel Density Estimation," *International Journal of Approximate Reasoning*, vol. 50, pp. 341-362, 2009.

[115] A. K. Jain and M. D. Ramaswami, "Classifier Design with Parzen Windows," in *Pattern Recognition and Artificial Intelligence*, North-Holland, Elsevier Science Publishers, 1988.

[116] J. S. Siminoff, Smoothing Methods in Statistics, New York: Springer-Verlag, 1996.

[117] B. Bollobas, Modern Graph Theory, New York: Springer Science & Business Media, 2013.

[118] D. W. Shahan and C. C. Seepersad, "Sequential Sampling with Kernel-Based Bayesian Network Classifiers," in *ASME IDETC/CIE Design Automation Conference*, Washington, DC, 2011.

[119] C. Sharpe, C. Morris, B. Goldsberry, C. C. Seepersad and M. R. Haberman, "Bayesian Network Structure Optimization for Improved Design Space Mapping for Design Exploration with Materials Design Applications," in *IDETC Design Automation Conference*, Cleveland,OH, 2017.

[120] C. Morris and C. C. Seepersad, "Identification of High Performance Regions of High-Dimensional Design Spaces with Materials Design Applications," in *IDETC Design Automation Conference*, Cleveland, Ohio, 2017.

[121] A.-M. Legendre, New Methods for the Determinatino of the Orbits of Comets (in French), Paris: F. Didot, 1805.

[122] K. P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.

[123] R. G. Pontius Jr, O. Thontteh and H. Chen, "Components of Information for Multiple Resolution Comparison Between Maps that Share a Real Variable," *Environ. Ecol. Stat.,* vol. 15, pp. 111-142, 2008.

[124] P. E. Gill, W. Murray and M. H. Wright, Practical Optimization, Philadelphia: Society for Industrial and Applied Mathematics, 2019.

[125] T. R. Marler and J. S. Arora, "Survey of Multi-objective Optimization Methods for Engineering," *Struct. Multidisc. Optim.,* vol. 26, no. 6, pp. 369-395, 2004.

[126] D. F. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Math. Comput.,* vol. 24, no. 111, pp. 647-656, 1970.

[127] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv Preprint,* vol. 1609.04747, 2016.

[128] L. Bottou, "Stochastic Gradient Descent Tricks," *Neural Networks: Tricks of the Trade,* pp. 421-436, 2012.

[129] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[130] D. C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *Math. Prog.,* vol. 45, no. 1, pp. 503-528, 1989.

[131] P. Heermann and N. Khazenie, "Classification of Multispectral Remote Sensing Data using a Back-propagation Neural Network," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 30, no. 1, pp. 81-88, 1992.

[132] J. Jiaojiao Li, Q. Du and Y. Li, "An Efficient Radial Basis Function Neural Network for Hyperspectral Remote Sensing Image Classification," *Soft Computing,* vol. 20, no. 12, pp. 4753-4759, 2016.

[133] F. Pedregosa et. al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[134] D. Rumelhart, G. Hinton and R. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[135] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, pp. 436-444, 2015.

[136] M. Abadi et. al., *TensorFlow: Large-scale machine learning on heterogeneous systems,* 2015.

[137] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS,* Lake Tahoe, Nevada, 2012.

[138] I. Goodfellow et. al., "Generative Adversarial Networks," *arXiv:1406.2661,* 2014.

[139] S. Banga et. al., "3D Topology Optimization Using Convolutional Neural Networks," *arXiv:1808.07440,* 2018.

[140] W. Chen, K. Chiu and M. Fuge, "Aerodynamic Design Optimization and Shape Exploration using Generative Adversarial Networks," in *AIAA Scitech Forum,* 2019.

[141] C. Sharpe and C. C. Seepersad, "Topology Design with Conditional Generative Adversarial Networks," in *IDETC-CIE,* Anaheim, California, 2019.

[142] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F. E. Alsaadi, "A Survey of Deep Neural Network Architectures and Their Applications," *Neurocomputing,* vol. 234, pp. 11-26, 2017.

[143] J. T. Connor, R. D. Martin and L. E. Atlas, "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Trans. Neur. Net.,* vol. 5, no. 2, pp. 240-254, 1994.

[144] T. Mikolov, T. Karafiat, L. Burget, J. Cernocky and S. Khudanpur, "Recurrent Neural Network Based Language Model," *Interspeech,* vol. 2, no. 3, pp. 1045-1048, 2010.

[145] J. Zhai, S. Zhang, J. Chen and Q. He, "Autoencoder and its Various Variants," in *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Miyazaki, Japan, 2018.

[146] Z. Wu, S. Pan, F. Chen, G. Long and C. Zhang, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596,* 2019.

[147] W. Chen and M. Fuge, "Beyond the Known: Detecting Novel Feasible Domains Over an Unbounded Design Space," *Journal of Mechanical Design,* vol. 139, no. 11, 2017.

[148] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[149] K. Woods, C. Doss, K. Bowyer, J. Solka, C. Priebe and W. Kegelmeyer, "Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 7, no. 6, pp. 1417-1436, 1993.

[150] M. Kubat, R. Holte and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," *Machine Learning,* vol. 30, no. 2-3, pp. 195-215, 1998.

[151] P. K. Chan, W. Fan, A. L. Prodromidis and S. J. Stolfo, "Distributed Data Mining in Credit Card Fraud Detection," *IEEE Intelligent Systems,* vol. 14, no. 6, pp. 67-74, 1999.

[152] N. Saliya, T. M. Khoshgoftaar and J. Van Hulse, "A Study on the Relationships of Classifier Performance Metrics," in *IEEE International Conference on Tools with Artificial Intelligence*, Newark, 2009.

[153] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, no. 9, pp. 1263-1284, 2009.

[154] T. E. Fawcett and F. Provost, "Adaptive Fraud Detection," *Data Mining and Knowledge Discovery,* vol. 3, no. 1, pp. 291-316, 1997.

[155] N. Japkowicz, C. Myers and M. Gluck, "A Novelty Detection Approach to Classification," in *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, 1995.

[156] C. X. Ling and C. Li, "Data Mining for Direct Marketing: Problems and Solutions," in *Dir*, International Conference on Knowledge Discovery and Data Mining, 1998.

[157] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume and C. Brunk, "Reducing Misclassification Costs," in *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, 1994.

[158] P. Domingos, "Metacost: A General Method for Making Classifiers Cost-sensitive," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, 1999.

[159] A. Estabrooks and N. Japkowicz, "A Mixture-of-Experts Framework for Concept-Learning from Imbalanced Data Sets," in *Proceedings of the 2001 Intelligent Data Analysis Conference*, Cascais, Portugal, 2001.

[160] C. Elkan, "The Foundations of Cost-Sensitive Learning," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[161] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling," in *Proceedings of the Fourteenth International Conference on Machine Learning*, Rome, 1997.

[162] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intelligent Data Analysis,* vol. 6, pp. 429-449, 2002.

[163] H. Han, W. Y. Wang and B. H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *Advances in Intelligent Computing ICIC 2005*, 2005.

[164] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *IEEE International Joint Conference on Neural NetworksIEEE International Joint Conference on Neural Networks*, Hong Kong, 2008.

[165] C. Bunkhumpornpat, K. Sinapiromsaran and L. Chidchanok, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009.

[166] E. Ramentol, Y. Caballero, R. Bello and F. Herrera, "SMOTE-RSB *: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowledge and Information Systems,* vol. 33, no. 2, pp. 245-265, 2012.

[167] J. A. Saez, J. Luengo, J. Stefanowski and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Information Sciences,* vol. 291, pp. 184-203, 2015.

[168] D. W. Rosen, C. C. Seepersad, T. W. Simpson and C. B. Williams, "Design for Additive Manufacturing: A Paradigm Shift in Design, Fabrication, and Qualification," *J. Mech. Des.,* vol. 137, no. 11, p. 110301, 2015.

[169] A. Norris and P. Packo, "Non-symmetric flexural wave scattering and one-way extreme absorption," *J. Acoust. Soc. Am.,* vol. 146, p. 873, 2019.

[170] Y. Jin, S. J. Qin and Q. Huang, "Out-of-plane geometric error prediction for additive manufacturing," in *IEEE International Conference on Automation Science and Engineering (CASE),* Gothenburg, 2015.

[171] H. Geng, J. Xiong and D. Huang, "A prediction model of layer geometrical size in wire and arc additive manufacture using response surface methodology," *Int. J. Adv. Manuf. Technol.,* vol. 93, pp. 175-186, 2017.

[172] Z. Huang, J.-Y. Dantan, A. Etienne, M. Rivette and N. Bonnet, "Geometrical deviation identification and prediction method for additive manufacturing," *Rapid Prototyping Journal,* vol. 24, no. 9, pp. 1524-1538, 2018.

[173] E. Garcia-Plaza, M. A. Caminero-Torija, P. J. Nunez-Lopez and J. M. Chacon-Munoz, "Analysis of PLA Geometric Properties Processed by FFF Additive Manufacturing: Effects of Process Parameters and Plate-Extruder Precision Motion," *Polymers,* vol. 11, no. 10, p. 1581, 2019.

[174] M. Schevenels, B. S. Lazarov and O. Sigmund, "Robust topology optimization accounting for spatially varying manufacturing errors," *Comput. Methods Appl. Mech. Engrg.,* vol. 200, pp. 3613-3627, 2011.

[175] G. I. Schueller and H. A. Jensen, "Computational methods in optimization considering uncertainties – An overview," *Comput. Methods Appl. Mech. and Engrg.,* vol. 198, no. 1, pp. 2-13, 2008.

[176] C. Morris, L. Bekker, M. R. Haberman and C. C. Seepersad, "Design Exploration of Reliably Manufacturable Materials and Structures With Applications to Negative Stiffness Metamaterials and Microstereolithography," *J. Mech. Des,* vol. 140, p. 111415, 2018.

[177] P. Pawlus, R. Reizer and M. Wieczorowski, "A review of methods of random surface topography modeling," *Tribology Intl.,* vol. 152, p. 106530, 2020.

[178] R. Singh, R. Kumar, I. Farina, F. Colangelo, L. Feo and F. Fraternali, "Multi-material additive manufacturing of sustainable innovative materials and structures," *Polymers,* vol. 1, no. 62, p. 11, 2019.

[179] J. A. Lewis, "Direct Ink Writing of 3D Functional Materials," *Adv. Funct. Mater.,* vol. 16, pp. 2193-2204, 2006.

[180] X. Li, J. M. Zhang, X. Yi, Z. Huang, P. Lv and D. Huiling, "Multimaterial Microfluidic 3D Printing of Textured Composites with Liquid Inclusions," *Adv. Sci.,* vol. 6, p. 1800730, 2019.

[181] T. M. Llewellyn-Jones, B. W. Drinkwater and R. S. Trask, "3D printed components with ultrasonically arranged microscale structure," *Smart Mater. Struct.,* vol. 25, p. 02LT01, 2016.

[182] H. A. Khazdozian, L. Li, M. P. Paranthaman, S. K. McCall, M. J. Kramer and I. C. Nlebedim, "Low-Field Alignment of Anisotropic Bonded Magnets for Additive Manufacturing of Permanent Magnet Motors," *JOM,* vol. 71, no. 2, pp. 626-632, 2019.

[183] D. Hanft, J. Exner, M. Schubert, T. Stocker, P. Fuierer and R. Moos, "An Overview of the Aerosol Deposition Method: Process Fundamentals and New Trends in Materials Applications," *J. Ceram. Sci. Tech.,* vol. 6, no. 3, pp. 147-182, 2015.

[184] B. Clifford, D. Beynon, C. Phillips and D. Deganello, "Printed-Sensor-on-Chip devices – Aerosol jet deposition of thin film relative humidity sensors onto packaged integrated circuits," *Sensors and Actuators B,* vol. 255, pp. 1031-1038, 2018.

[185] W. Chen, "A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design," *AIAA J.,* vol. 37, no. 8, pp. 982-989, 1999.

[186] M. Kalsi, K. Hacker and K. Lewis, "A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex System Design," *J. Mech. Des.,* vol. 123, no. 1, pp. 1-10, 1999.

[187] O. Sigmund, "Manufacturing Tolerant Topology Optimization," *Acta. Mech. Sin.,* vol. 25, pp. 227-239, 2009.

[188] S. Chen, W. Chen and S. Lee, "Level Set Based Robust Shape and Topology Optimization Under Random Field Uncertainties," *Struct. Multidiscip. Optim.,* vol. 41, no. 4, pp. 507-524, 2010.

[189] G. G. Wang and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *J. Mech. Des.,* vol. 129, no. 4, pp. 370-380, 2007.

[190] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box

functions," *Structural and Multidisciplinary Optimization,* vol. 41, no. 2, pp. 219-241, 2010.

[191] M. A. Nik, K. Fayazbakhsh, D. Pasini and L. Lessard, "A comparative study of metamodeling methods for the design optimization of variable stiffness composites," *Composite Structures,* vol. 107, pp. 494-501, 2014.

[192] C. Sharpe, T. J. Wiest, P. Wang and C. C. Seepersad, "A Comparative Evaluation of Supervised Machine Learning Classification Techniques for Engineering Design Applications," *J. Mech. Des.,* vol. 141, no. 12, p. 121403, 2019.

[193] F. A. Viana, C. Gogu and R. T. Haftka, "Making the most out of surrogate models: tricks of the trade," in *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference,* Montreal, 2010.

[194] M. Bollhofer, O. Schenk, R. Janalik, S. Hamm and K. Gullapalli, "State-of-The-Art Sparse Direct Solvers," *Parallel Algorithms in Computational Science and Engineering. Modeling and Simulation in Science, Engineering and Technology,* pp. 3-33, 2020.

[195] I. Gibson, D. Rosen, B. Stucker and M. Khorasani, Additive manufacturing technologies, New York: Springer, 2014.

[196] P. Heerman and N. Khazenie, "Classification of Multispectral Remote Sensing Data using a Back-propagation Neural Network," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 30, no. 1, pp. 81-88, 1992.

[197] J. J. Li, Q. Du and Y. Li, "An Efficient Radial Basis Function Neural Network for Hyperspectral Remote Sensing Image Classification," *Soft Computing,* vol. 20, no. 12, p. 47534759, 2016.

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

[199] D. Rumelhart, G. Hinton and R. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[200] G. James, D. Witten and R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013.

[201] R. A. Roach, N. Argibay, K. Allen, D. K. Balch, L. L. Beghini, J. E. Bishop and B. Boyce, "Born Qualified Grand Challenge LDRD Final Report," Sandia National Lab., Albuquerque, NM and Livermore, CA, 2018.

[202] J. Matthews, T. Klatt, C. Morris, C. C. Seepersad and M. R. Haberman, "Hierarchical Design of Negative Stiffness Metamaterials Using a Bayesian Network Classifier," *Journal of Mechanical Design,* vol. 138, no. 4, p. 041404, 2016.

[203] P. B. Nakshatrala, D. A. Tortorelli and K. B. Nakshatrala, "Nonlinear structural design using multiscale topology optimization. Part I: Static formulation," *Comput. Methods Appl. Mech. and Engrg.*, vol. 261, pp. 167-176, 2013.

[204] R. Sivapuram, P. D. Dunning and H. A. Kim, "Simultaneous material and structural optimization by multiscale topology optimization," *Structural and multidisciplinary optimization,* vol. 54, no. 5, pp. 1267-1281, 2016.

[205] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and multidisciplinary optimization,* vol. 41, no. 2, pp. 219-241, 2010.

[206] S. Shan and G. G. Wang, "Metamodeling for High Dimensional Simulation-based Design Problems," *Journal of Mechanical Design,* vol. 132, no. 5, 2010.

[207] P. B. Backlund, D. W. Shahan and C. C. Seepersad, "A comparative study of the scalability of alternative metamodeling techniques," *Engineering Optmization,* vol. 44, no. 7, pp. 767-786, 2012.

[208] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen and F. J. Wu, "Building surrogate models based on detailed and approximate simulations," *Journal of Mechanical Design,* vol. 128, no. 4, pp. 668-677, 2006.

[209] D. Liu and Y. Wang, "Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling," *Journal of Mechanical Design,* vol. 141, no. 12, p. 121403, 2019.

[210] D. Huang, T. T. Allen, W. I. Notz and R. A. Miller, "Sequential kriging optimization using multiple-fidelity evaluations," *Structural and Multidisciplinary Optimization,* vol. 32, no. 5, pp. 369-382, 2006.

[211] A. Forrester, A. Sobester and A. Keane, Engineering design via surrogate modeling: a practical guide, John Wiley & Sons, 2008.

[212] A. Forrester and A. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences,* vol. 45, no. 1-3, pp. 50-79, 2009.

[213] J. Allison and D. Herber, "Special Section on Multidisciplinary Design Optimization: Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal,* vol. 52, no. 4, pp. 691-710, 2014.

[214] J. Allison, T. Guo and Z. Han, "Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization," *Journal of Mechanical Design,* vol. 136, no. 8, p. 081003, 2014.

[215] A. Deshmukh and J. Allison, "Design of dynamic systems using surrogate models of derivative functions," *Journal of Mechanical Design,* vol. 139, no. 10, p. 101402, 2017.

[216] S. Chen and S. A. Billings, "Neural Networks for Nonlinear Dynamic System Modelling and Identification," *Intl. J. Control,* vol. 56, no. 2, pp. 319-346, 1992.

[217] T. Q. Chen, Y. Rubanova, J. Bettencourt and D. Duvenaud, "Neural Ordinary Differential Equations," *Advances in neural information processing systems*, pp. 6571-6583, 2018.

[218] Z. Long, Y. Lu, X. Ma and B. Dong, "PDE-net: Learning PDEs from data," *arXiv preprint arXiv:1710.09668*, 2017.

[219] M. Gori, G. Monfardini and F. Scarselli, "A new model for learning in graph domains," *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 729-734, 2005.

[220] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2017.

[221] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu and L. Wang, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[222] B. Yu, H. Yin and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[223] S. Guo, Y. Lin, N. Feng, C. Song and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019.

[224] S. Yan, Y. Xiong and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Thirty-second AAAI conference on artificial intelligence*, 2018.

[225] I. Bello, H. Pham, Q. V. Le, M. Norouzi and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[226] E. Khalil, H. Dai, Y. Zhang and B. Dilkina, "Learning combinatorial optimization algorithms over graphs," *Advances in Neural Information Processing Systems*, pp. 6348-6358, 2017.

[227] W. Kool, H. Van Hoof and M. Welling, "Attention, learn to solve routing problems!," *arXiv preprint arXiv:1803.08475*, 2018.

[228] P. W. Battaglia, J. B. Hamrick and V. Bapst, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[229] "graph_nets," Deepmind, [Online]. Available: https://github.com/deepmind/graph_nets. [Accessed 24 02 2020].

[230] NYU, NYU Shanghai, AWS Shanghai AI Lab, AWS MXNet Science Team, "Deep Graph Library," [Online]. Available: https://github.com/dmlc/dgl. [Accessed 2020 24 02].

[231] A. Sanchez-Gonzales and et al., "Graph Networks as Learnable Physics Engines for Inference and Control," *arXiv preprint arXiv:1806.01242*, 2018.

[232] C.-L. Ding and X.-P. Zhao, "Multi-band and Broadband Acoustic Metamaterial with Resonant Structures," *J. Phys. D.: Appl. Phys.*, vol. 44, p. 215402, 2011.

[233] Z.-X. Xu, H.-Y. Meng and A. Chen, "Tunable Low-frequency and Broadband Acoustic Metamaterial Absorber," *J. Appl. Phys.*, vol. 129, p. 094502, 2021.

[234] J. L. Ba, J. R. Kiros and G. E. Hinton, "Layer Normalization," *arXiv,* p. 1607.06450, 2016.

[235] J. Gilmer, S. Schoenholz and et al., "Neural Message Passing for Quantum Chemistry," *arXiv:1704.01212v2,* 2017.

[236] D. Shahan and C. Seepersad, "Bayesian Networks for Set-Based Collaborative Design," *Journal of Mechanical Design,* vol. 134, no. 7, 2012.

[237] J. Matthews, T. Klatt, C. Morris, C. Seepersad, M. Haberman and D. Shahan, "Hierarchical Design of Negative Stiffness Metamaterials using a Bayesian Network Classifier," *Journal of Mechanical Design,* vol. 138, no. 4, 2016.

[238] C. Morris, L. Bekker, M. Haberman and C. Seepersad, "Design Exploration of Reliably Manufacturable Materials and Structures with Applications to Negative Stiffness Metamaterials and Microstereolithography," *Journal of Mechanical Design,* vol. 140, no. 11, 2018.

[239] D. Rosen, "A Set-Based Design Method for material-Geometry Structures by Design Space Mapping," in *ASME IDETC*, Boston, Massachusetts, 2015.

[240] A. Basudhar and S. Missoum, "Reliability assessment using probabilistic support vector machines," *J. Reliability and Safety,* vol. 7, no. 2, pp. 156-173, 2013.

[241] A. Basudhar, S. Missoum and A. Sanchez, "Limit State Function Identification Using Support Vector Machines for Discontinuous Responses and Disjoint Failure Domains," *Probabilistic Engineering Mechanics,* vol. 23, no. 1, pp. 1-11, 2008.

[242] J. Patel and S.-K. Choi, "An enhanced classification approach for reliability estimation of structural systems," *Journal of Intelligent Manufacturing,* vol. 25, no. 3, pp. 505-519, 2014.

[243] H. Song, S.-K. Choi, I. Lee, L. Zhao and D. Lamb, "Adaptive virtual support vector machine for reliability analysis of high-dimensional problems," *Struct Multidisc Optim,* vol. 47, no. 4, pp. 479-491, 2013.

[244] Z. Wang and P. Wang, "An Integrated Performance Measure Approach for System Reliability Analysis," *Journal of Mechanical Design,* vol. 137, no. 2, 2015.

[245] Z. Hu and X. Du, "Integration of Statistics- and Physics-Based Methods- A Feasibility Study on Accurate System Reliability Prediction," *Journal of Mechanical Design,* vol. 140, no. 7, 2018.

[246] P. Wang, Z. Wang and A. Almaktoom, "Dynamic Reliability-based Robust Design Optimization with Time-Variant Probabilistic Constraints," *Engineering Optimization,* vol. 46, no. 6, pp. 784-809, 2014.

[247] X. Zhuang and R. Pan, "A Sequential Sampling Strategy to Improve Reliability-Based Design Optimization with Implicit Constraint Functions," *Journal of Mechanical Design,* vol. 134, no. 2, 2012.

[248] Y. Wang, X. Yu and X. Du, "Improved Reliability-based Design Optimization with Support Vector Machine and Its Application in Aircraft Wing Design," *Mathematical Problems in Engineering,* vol. 1, no. 14, 2015.

[249] B. Xiong and H. Tan, "New structural reliability method with focus on important region and based on adaptive support vector machines," *Advances in Mechanical Engineering,* vol. 9, no. 6, pp. 1-12, 2017.

[250] L. M. White, T. K. West and A. J. Brune, "Reliability-based Design of Thermal Protection Systems with Support Vector Machines," *Journal of Spacecraft and Rockets,* 2019.

[251] s.-k. l. developers, "scikit-learn v0.20.3 user guide," 2019. [Online]. Available: https://scikit-learn.org/stable/user_guide.html. [Accessed February 2019].

[252] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge: Cambridge University Press, 2000.

[253] P. Tamilselvan and P. Wang, "Failure Diagnosis using Deep Belief Learning Based Health State Classification," *Reliability Engineering and System Safety,* vol. 115, pp. 124-135, 2013.

[254] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics,* vol. 6, no. 1, pp. 19-26, 1980.

[255] T. Wiest, C. C. Seepersad and M. Haberman, "Design Space Exploration in Sparse, Mixed Continuous/Discrete Spaces via Synthetically Enhanced Classification," in *Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2B: 44th Design Automation Conference*, Quebec City, Quebec, 2018.

[256] T. Wiest, C. C. Seepersad and M. R. Haberman, "Robust Design of an Asymmetrically Absorbing Willis Acoustic Metasurface Subject to Manufacturing-Induced Dimensional Variations," *J. Acoust. Soc. Am.,* vol. 151, no. 1, pp. 216-231, 2022.