NORTHWESTERN UNIVERSITY

Data-Driven and Diversity-Enhanced Design of Heterogeneous Multiscale Structures

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Mechanical Engineering

By

Yu-Chin Chan

EVANSTON, ILLINOIS

September 2022

ABSTRACT

Data-Driven and Diversity-Enhanced Design of Heterogeneous Multiscale Structures

Yu-Chin Chan

The advent of metamaterials – hierarchical structures that manifest properties beyond those found in nature through geometry rather than material composition – inspired new possibilities and research in many fields. In mechanics, periodic metamaterials exhibit behaviors ranging from unprecedented compressibility to extreme stiffness. Numerous geometric classes of metamaterials with these properties have been discovered, such as re-entrant structures, chiral patterns, trusses, and isosurfaces. Nevertheless, periodic mechanical metamaterials are limited in scope and application. Instead, aperiodic systems, or heterogeneous multiscale structures wherein microstructural topologies differ from neighbor-to-neighbor, can achieve fine control over spatially varying properties, leading to even more extraordinary functionalities such as programmable deformation and crashworthiness. However, the challenges of designing these heterogeneous structures grows alongside their geometrical and behavioral complexities. Computational methods like topology optimization have risen to meet some of the hurdles, but they are restricted by the computational costs of multiscale simulations, searching through vast combinatorial

design spaces, and resolving disconnected neighboring microstructures. Recent attention is now shifting to data-driven design, which utilizes datasets of unit cells, usually from only one geometric class, and their pre-computed homogenized properties to alleviate much of the computational burden. While powerful, this new paradigm is held back by often-ignored questions pertaining to the adverse effect that the diversity and quality of a dataset can have on data-driven design representation and synthesis.

We propose a suite of methods that take steps toward releasing the full potential of data-driven methods, improving the diversity of metamaterials datasets to enhance the design of complex heterogeneous multiscale structures. The methods focus on a core hypothesis: that *diverse subsets of unit cells, including those from different geometric classes, form the basis of scalable and general data-driven synthesis frameworks.* To this end, we create an automated data selection method, METASET, that uses similarity metrics and probabilistic Determinantal Point Processes to distill subsets that are jointly diverse in shapes and properties. For the first time in metamaterials design, we validate that by eliminating redundant or biased samples, small yet diverse subsets can boost the connectivity, design performance, and scalability of data-driven search algorithms.

Leveraging the ability of a diverse subset to concisely encompass a wide design space, we then create two data-driven frameworks, which utilize diverse unit cell "basis" classes to seed low dimensional design representations, transforming the design of functionally graded structures into highly efficient and effective approaches. Unlike typical methods, ours require neither pre-defined sets of compatible basis classes nor connectivity constraints to achieve a well-connected design. In the first framework, we propose a new multiclass shape blending scheme, which generates novel microstructures as a blend of diverse basis classes. By taking the blending parameters as low dimensional design variables and optimizing their distribution throughout the global structure, we can guarantee well-connected, functionally graded designs that attain high performance across compliance minimization and programmable shape matching problems, even with a small handful of diverse basis classes. Finally, we propose a generative deep learning-based framework to expand the design freedom, generality, and efficiency even further. Data from truss and isosurface classes, traditionally expressed through very different design variables, are incorporated into a single pipeline, from the simultaneous acquisition of a diverse and high quality multiclass dataset, to the training of a multiclass Wasserstein Generative Adversarial Network that compresses the classes into a unified, 10-dimensional latent representation. With the latent variables as design variables, our framework can optimize and synthesize exceptionally smooth grading between trusses and isosurfaces while achieving lower maximum stress. Thus, we can automate, with high efficiency and generality, the decision of whether to use trusses, isosurfaces, or hybrids of both, in a design. Although our proposed methods, METASET and data-driven synthesis, are demonstrated with linear elastic mechanical applications, they are general in order to welcome the potential of future extensions to additional domains.

Acknowledgements

...there are rare moments when another soul dips near yours,

as stars once a year brush the earth.

- *Circe*, Madeline Miller

My dissertation would not have been possible without the encouragement and guidance of my advisor, Professor Wei Chen. I am extremely grateful for the care, respect, and patience she has shown by allowing me to carve out my own research and career path. I am always inspired and awe-struck by her dedication to her students and research, as well as by her achievements as a pillar in the field of computational design.

I'm immensely grateful to these labmates who were instrumental in guiding my earlier years at Northwestern: Prof. Ramin Bostanabad, Dr. Tianyu Huang, Prof. Faez Ahmed, and Dr. Siyu Tao; and for the Data-Driven Metamaterials Design subgroup, who constantly inspire and challenge me: Liwei Wang, Doksoo Lee, Dr. Daicong Da, and Dr. Wei (Wayne) Chen. A special thanks goes out to Doksoo and Stefan for helping to proofread this dissertation at the last minute. To Akshay, Anton, Yaxin, Umar, Yido, Henry, Tuba, Vispi, and all of the IDEAL family, thank you so much for your camaraderie. I will miss our coffee time, chats, and all-too-short-lived board game nights.

I would also like to thank my colleagues at Siemens Technology, from my fellow summer 2018 interns to my current teammates and collaborators.

Finally, but not least, I thank my parents for their sacrifice and unwavering support throughout my life, and my sister for never failing to make me laugh. We've come a long way.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Possessing a variety of extraordinary properties with far-reaching impact, metamaterials have fascinated researchers for the past few decades. Like composite materials, the macroscopic responses of interest, or effective properties, can be calculated from a representative microstructure; but unlike typical composites, the effective properties of metamaterials are beyond those found in nature and stem from their intentionally hierarchical structure rather than their material composition [4]. These designs are traditionally periodic, with a single repeated microstructure (otherwise known as unit cell or building block), but can also be assembled from multiple microstructures to form aperiodic, *heterogeneous multiscale structures* that attain spatially varying behavior.

The potential applications of multiscale structures are numerous and impactful. The idea of a “metamaterial” originates in the electromagnetic field, in which intriguing properties are achieved when the unit cell periodicity is on a scale smaller than the wavelength of interest [5]. In particular, a wealth of research exists for optical (photonic) metamaterials, which can produce, e.g., negative effective refraction indices, super-resolution lenses and invisibility cloaks [6]. In recent years, additive manufacturing has allowed easier fabrication of geometrically complex structures, and as a result metamaterial systems have gained traction in other fields, including mechanics [5, 7]. Acoustic (phononic) metamaterials are one branch of mechanical metamaterials, and can be used to control wave propagation through gaseous, liquid or solid media [8, 9] or to achieve complete bandgaps that prevent sound waves from any direction [10, 11].

This dissertation focuses on another type of mechanical multiscale structure: ones that undergo mechanical loading. The simplest applications of these are the "ultralight ultrastiff" structures that disobey the typical scaling laws of natural materials, which state that strength degrades significantly as density is reduced. Since the greater surface area and porosity from the hierarchical features allow high performance-to-weight ratios, these are attractive as, for example, lighweight automotive and aerospace designs [4, 12, 13], biomedical implants [14–16], and multifunctional heat exchangers [17, 18]. Reconfigurable and adaptive elastic structures can also be created through pattern transformations and compressive instability, such as those in cellular origami and porous structures under large deformation [4, 19–21]. These multistable designs exhibit high energy absorption during sudden collapses when deformed past the linear elastic regime. Along similar lines, the buckling-induced, layer-by-layer failure mechanism of heterogeneous, *functionally graded structures*, in which the geometries of the microstructures vary smoothly from neighbor to neighbor, have been spotlighted since they can increase energy absorption compared to periodic structures, permitting programmable deformation responses [22–24]. Multiscale structures that can elastically recover their initial shapes after loading have also been demonstrated in Refs. [25–28], and hold significant benefits for impact and blast protection applications, e.g., helmets and armor [29]. Similarly, auxetic metamaterials with negative Poisson's ratio that expand laterally when stretched [30], unlike most natural materials which contract, can also be utilized in damping, energy absorption, responsive devices, tougher-yet-lightweight structures and more [31–33]. A thorough review of mechanical metamaterials designed based on Young's modulus, effective density, Poisson's ratio, and other elastic properties not covered above, can be found in Ref. [4].

1.1. Research Motivation

It is important to note that most metamaterials research thus far involve repeating only one unit cell throughout the entire structure – periodic structures, which can be designed at a single scale (the unit cell). However, aperiodic, *heterogeneous systems* with more than one type of microstructure can attain precise control over spatially varying properties and are useful for applications like soft robotics, customized wearables and energy absorption [34–37]. These are still not as commonplace since their heterogeneity make them more difficult to design and fabricate due to the huge combinatorial design space and geometric frustration when disparate microstructures do not connect to neighboring ones [35, 38]. The scalable and efficient design of heterogeneous multiscale systems is one of the most intriguing and necessary hurdles to surmount in order push the current state-of-the-art into a new era of metamaterials design [39, 40]. This dissertation responds to these challenges, summarized below, from a data-driven perspective.

1.1.1. Challenges of Traditional Design Methods

Just as a variety of functionalities are available for metamaterials and multiscale systems, there is an abundance of design representations for unit cell morphologies – so many that several review papers have been devoted to categorizing them; see Refs. [4, 7, 33, 41, 42]. To facilitate clarity of discussion, an important definition is used throughout this dissertation:

A ***class*** *refers to a group of unit cells that can be generated from the same geometric motif or design parameterization. In other words, unit cells from each class can be created from the same parametric variables or functions, and a design that combines multiple motifs can be called* ***multiclass****.*

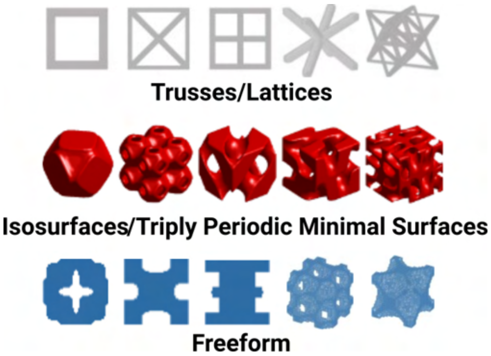


Figure 1.1. Examples of geometric unit cell classes.

A few example classes are shown in Fig.1.1. It is possible for classes to coincide with others: extreme examples are *freeform* topology optimized classes derived from density-based [43] or level-set representations [44], which are general and can describe both simple and counterintuitive shapes, thus engendering the possibility of overlaps. Apart from the freeform classes, two of the most common geometrical families are *trusses* (sometimes called lattices), and *isosurface* or triply periodic minimal surfaces (TPMS). Although the advantages of certain classes have been elucidated empirically, many also have desirable properties similar those of other classes. For example, the truss class is known to exhibit high strength-to-weight ratio, great heat dissipation and low negative Poisson's ratio, while the isosurface class is known for high energy absorption and tailorable bandgaps due to their amenability to functional grading. Both have been used in similar applications as well, such as lightweighting [45, 46], energy absorption [22, 47], and acoustic bandgap tailoring [11, 48]. Although a segment of research has been devoted to comparing the performance of different classes [24, 49, 50], *these differences and overlaps frustrate the choice of which unit cell class to utilize in a multiscale design* [51]. For this reason, the majority of the existing works manually select a specific class for their applications.

To automate some of the difficult choices faced by a designer of metamaterials and multiscale systems, computational design synthesis can generate novel structures. These approaches typically employ: 1) parametric shape optimization of a periodic unit cell [52–54], 2) topology optimization (TO) of one or more microstructures, which are then assembled into pre-defined sub-regions in the global macro-structure [32, 55–57]; or 3) hierarchical and concurrent multiscale TO that optimize the micro- and macro-scale structures simultaneously [58, 59]. In the first approach, the general shape and parameterization of the unit cell must be decided beforehand. Although this allows the domain knowledge of certain desirable features (e.g., honeycomb shapes) to be chosen, *it not only leaves the designer with the question of which class to choose prior to design but also limits the achievable properties.* For the latter two TO methods, the unit cell geometries are generally represented as either pixels in 2D, voxels in 3D, or level-set fields. Although this opens an infinite-dimensional design space to create “freeform” unit cells, designers must still choose an initial shape since TO infamously suffers from local optima that are sensitive to initial guesses. Moreover, *traditional multiscale TO methods have other challenges* stemming from the extremely high dimensional design space, computational costs from multiscale analysis, and complex formulations to ensure manufacturable and physically connected microstructures.

Meanwhile, recent works in the TO field have shown that *multiclass heterogeneous structures*, which incorporate multiple unit cell classes, can obtain greater performance than single-class structures [60–64]. Yet, to ensure well-connected interfaces, the "classes" in most of these existing works are carefully pre-defined or constrained to be similar to each other. Removing this limitation to deploy *unit cell classes that cover more diverse shapes and properties would significantly increase the generality of multiclass design methods*, but the challenges that this would bring in connectivity and efficiency have thus far not been resolved.

1.1.2. Gaps in Data-Driven Design

To design multiclass heterogeneous structures, a promising alternative is a data-driven approach that consists of three components (Fig. 1.2): 1) **data acquisition and representation**, 2) **mechanistic-based learning**, and 3) **multiscale structure synthesis**. The key is a large database of many unit cells and their corresponding properties. Based on how this dataset is utilized, the approaches can be further split into two categories. In "top-down" frameworks, the optimal macrostructure and macroscale properties are first identified. Subsequently, the pre-computed unit cells that match the optimal target properties while remaining well-connected can be efficiently selected through combinatorial optimization. Thus, top-down methods can quickly explore the large design space while improving microstructure connectivity in aperiodic structures [1, 34, 65, 66]. On the other hand, "bottom-up" frameworks directly optimize the geometric parameters of both the macro- and micro-structures simultaneously by closely following traditional TO [3]. Since this approach can leverage well-established TO formulations and does not require

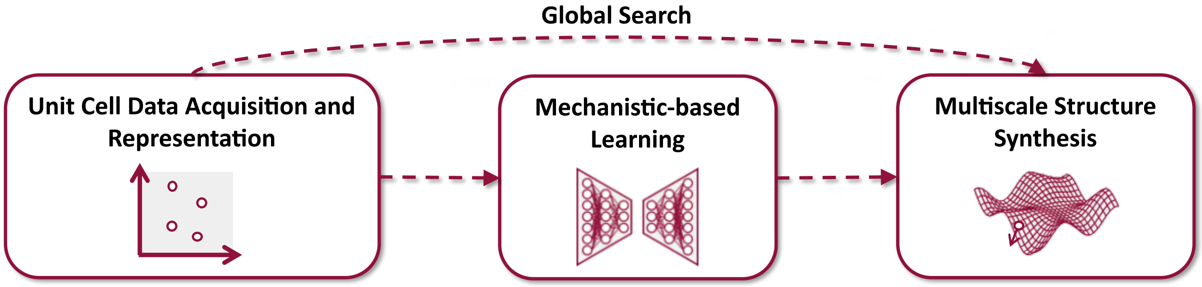

Global Search
Unit Cell Data Acquisition and Representation
Mechanistic-based Learning
Multiscale Structure Synthesis

Figure 1.2. Three components of data-driven heterogeneous multiscale structure design. This thesis addresses challenges in the first (acquisition and representaiton) and last (synthesis), but still uses the second (mechanistic learning).

finding the optimal target properties, it shows more promise in more realistic applications with, e.g., nonlinear mechanics or multiphysics. A few recent bottom-up methods have achieved multiclass designs with little additional computational cost compared to traditional TO [3, 67]. However, it is relatively less studied than top-down approaches.

In both top-down and bottom-up frameworks, generative deep learning models can learn from the datasets to represent unit cells shapes in a compact, low-dimensional latent space for optimization or fast generation of new designs [1, 68–70]. In addition, mechanistic-based models can be trained to map microstructural shapes to their properties, extracting mechanics knowledge from the data to bypass costly homogenization simulations and inform design decisions during synthesis [1, 71–73]. However, the core assumption of existing data-driven works is that an optimal microstructure already exists in, or can be interpolated from, the database. Moreover, the current databases are constrained since they are generated from a single geometry class, which can limit the generality and optimality of the designs. As such, *one critical hurdle to improving the effectiveness of data-driven design is to improve the diversity, i.e., the coverage of a wide* *variety of properties and geometries (such as from multiple classes), of the datasets on which they rely.*

Parallel to the surge of data-driven methods is the push for data-sharing and online communities, which could enable faster acquisition of diverse unit cell data. For example, NanoMine [74], one of several open source databases fueled by NIST's Materials Genome Initiative, is an online data curation system for polymer nanocomposites. Since its creation, it has permitted microstructural design using data-driven optimization [75] as well as deep learning [76, 77]. Building on these successes, it has been extended to MetaMine (together forming MaterialsMine [78]), an online resource of metamaterials and design tools. MetaMine is currently being built to allow the collection and exploration of microstructure data from multiple sources – and therefore multiple geometry classes. It will provide advantageous resources for data-driven metamaterials and multiscale design, but will also raise its own questions such as *how to filter and search through large amounts of data for scalable design*.

While crowd-sourcing is a practical way to gather large quantities of data, different sources of data will indubitably arrive with different levels of quantity and quality. For example, one source may provide thousands of truss unit cells with similar Young's moduli, while another contributes only tens of isosurfaces but covers a wider range of Young's moduli. From a data-driven point of view, this presents a *class imbalance problem*, which can cause machine learning models to inadequately learn under-represented features (e.g., certain shapes for generative models or properties for mechanistic models) with less data [79]. In realistic design applications, this is also problematic since any models learned from imbalanced and low-quality data are likely to be inaccurate, biased towards the over-represented class, or contain too many infeasible unit cells, leading to less optimal designs. *Synthesizing high performance and feasible designs by combining imbalanced data from disparate classes has rarely been addressed in existing literature.*

This thesis endeavors to answer the following research questions that capture the current challenges in data acquisition, representation, and synthesis of heterogeneous multiscale structures under data-driven frameworks:

(1) Scalability: How to distill a smaller yet diverse subset of microstructures from existing datasets to improve the efficiency of data-driven design without sacrificing the coverage of the design space?

(2) Representation: Can a compact and unified design representation merge disparate classes to discover novel microstructures that are not constrained by predefined choices?

(3) Novelty: Is there a method that can efficiently generate novel unit cells that are both hybrids of existing classes and feasible, for use in mechanistic learning and gradient-based design synthesis?

(4) Synthesis: What design frameworks and deep learning techniques can effectively generate heterogeneous multiscale structures with greater potential applicability to real-world problems?

In answer to the questions above, the overarching hypothesis of this dissertation is: *By leveraging smaller yet diverse and balanced subsets of unit cells and geometrical classes as seeds for novel heterogeneous multiscale structures, the scalability and generality of data-driven design synthesis frameworks can be improved.* In pursuit of this, a suite of methods to complement the growing data revolution in metamaterials design, as well

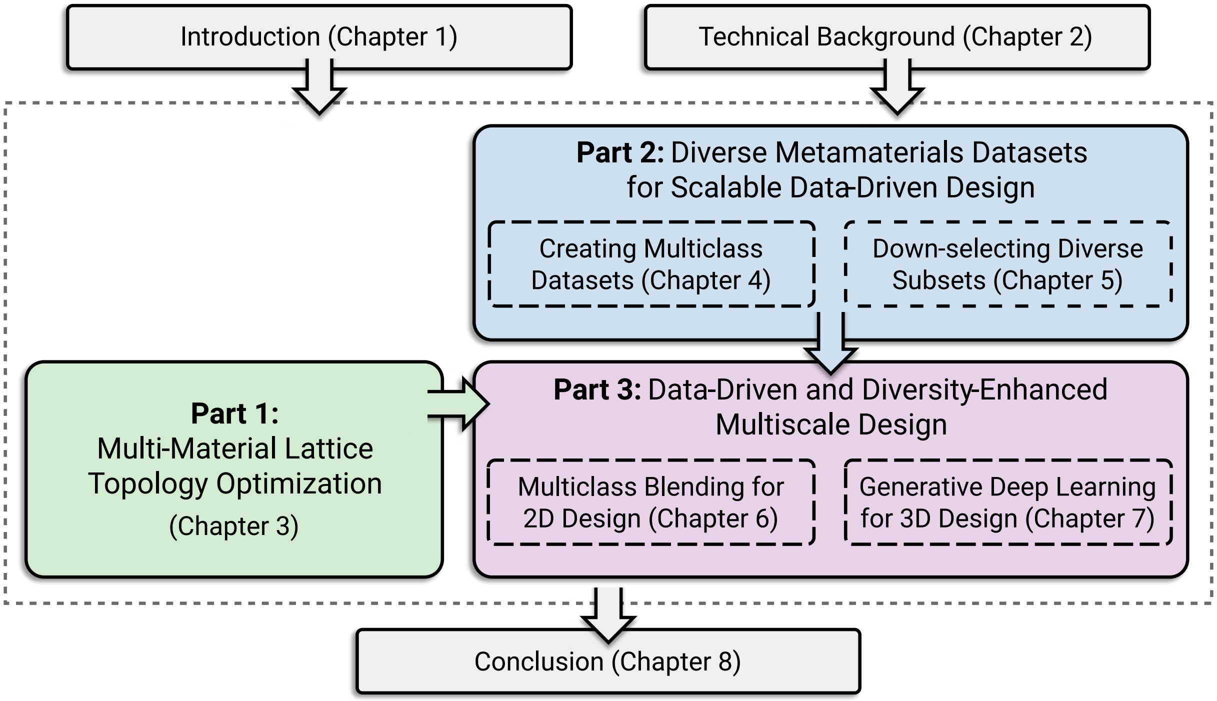

Introduction (Chapter 1)
Technical Background (Chapter 2)
Part 2: Diverse Metamaterials Datasets for Scalable Data-Driven Design
Creating Multiclass Datasets (Chapter 4)
Down-selecting Diverse Subsets (Chapter 5)
Part 1: Multi-Material Lattice Topology Optimization (Chapter 3)
Part 3: Data-Driven and Diversity-Enhanced Multiscale Design
Multiclass Blending for 2D Design (Chapter 6)
Generative Deep Learning for 3D Design (Chapter 7)
Conclusion (Chapter 8)

Figure 1.3. The structure of this dissertation.

as to fill the void in scalable multiscale TO, are proposed. An efficient algorithm is created to down-sample diverse and balanced datasets, removing any redundancies or biases that impede efficient design. Instead of limiting the design representation to one geometry class, multiple classes of unit cells are combined in lower dimensional spaces via a new multiclass shape blending scheme rooted in computer graphics, as well as a deep generative model based on computer vision methods. The initial classes are then leveraged as starting points to efficiently create high performing multiscale designs composed of novel microstructures beyond the existing classes.

1.2. Dissertation Structure

This chapter introduced the background, motivation, and research tasks. In Chapter 2, we introduce technical concepts that form the backbone of this thesis. The remaining chapters document the interconnected research tasks in this dissertation, which are as follows:

- **Chapter 3: Multi-Material Lattice Structure Topology Optimization.**

  Additive manufacturing techniques have recently enabled printing intricate, multi-material lattice structures, or designs which are composed of networks of trusses. However, manufacturing is imperfect; uncertainty arising from deviations in geometry and the source printing material, as well as the actual loading conditions, can affect the performance of the structures. This chapter proposes a robust topology optimization (TO) method to design multi-material lattice structures by performing uncertainty quantification of the design objective under simultaneous material and load uncertainty. To accelerate the process, univariate dimension reduction and Gauss quadrature are employed to evaluate the statistical moments of compliance. We also introduce a new, generalized multi-material interpolation scheme that can be integrated into robust TO and which easily scales to any number of materials. While this chapter does not use data-driven design, it sets the foundation for later tasks with the general multi-material mixing scheme that we extend to multiclass blending in Chapters 6 and 7.

- **Chapter 4: Creating Multiclass Datasets for Data-Driven Design.**

  Data acquisition is a crucial first step in data-driven design. Although several approaches have been developed to speed up data acquisition, they make sacrifices to their coverage in properties or coverage in shapes. Inverse optimization-based data acquisition can cover wide property spaces, but leads to many redundant shapes. Methods driven by sampling only parameters in the shape space are fast in that the properties do not need to be calculated during acquisition, but in turn means low coverage in property space. Hence, coverage in shapes and properties are not one-to-one. In this chapter, we introduce methods to collect a multiclass (truss and isosurface) unit cells dataset based on existing state-of-the-art techniques in literature. The benefits and disadvantages of these approaches are observed and recommendations are made. The experiment motivates the need for the next task, diverse subset selection (Chapter 5).

- **Chapter 5: Shape and Property Diverse Data Selection for Scalable Data-Driven Multiscale Design.**

  An oft-neglected challenge in data-driven design of metamaterials and multiscale structures is the acquisition of datasets that are diverse in both shapes and properties. This joint diversity is necessary to assemble spatially-varying designs that meet arbitrary performance requirements while offering a selection of unit cells that are well connected to their neighbors. However, existing approaches use exhaustive data generation schemes to cover as many properties or shapes as possible. Inevitably, coverage in one space is sacrificed for the other, which may waste computational resources and affect downstream tasks such as mechanistic learning and design synthesis.

  As a solution to imbalanced or biased datasets, we propose METASET, a probabilistic methodology to autonomously down-select small yet diverse subsets of unit cells. To maximize the diversity of subsets, METASET is rooted in Determinantal Point Processes, which models the probability of a diverse subset as the determinant of a similarity matrix containing the pairwise similarities between every data point. A larger determinant value indicates that a set contains less similar items that span a larger volume, and hence has greater diversity. To obtain the similarity matrix, we propose diversity metrics for both shapes and properties based on continuous vector representations (e.g., stiffness tensor components for properties and deep learned latent representations for shapes). In "top-down" data-driven design of shape matching metamaterials, we discover that more data is not always better – small yet diverse subsets can, in fact, improve the design performance, search algorithm efficiency, and connectivity of heterogeneous structures. This algorithm also initializes the data acquisition in Chapters 6 and 7, wherein diverse subsets and classes of microstructures are utilized in design frameworks.

- **Chapter 6: Data-Driven Topology Optimization of Functionally Graded Structures with Multiclass Shape Blending.**

  Functionally graded structures are desirable for the gradual changes in the shapes of neighboring microstructures, which naturally leads to better connectivity and may help reduce the error of the homogenization assumption often used in multiscale design. However, existing TO methods to design them are limited to a small number of pre-defined unit cells.

Bridging the freedom of fully aperiodic structures with the efficiency and smooth interfaces of graded designs, this chapter proposes a “bottom-up” data-driven framework for multiclass functionally graded structures that mixes several basis unit cell classes to create spatially-varying designs with guaranteed feasibility. The major contribution is a new multiclass shape blending scheme that generates smoothly graded microstructures without requiring compatible basis classes or connectivity and feasibility constraints. The scheme transforms the microscale problem into an efficient, low-dimensional one without confining the design to predefined shapes. We perform case studies with sets of five common truss geometries and five diversity-based freeform topologies to highlight the versatility and effectiveness of our framework. We reveal that truss-type classes consistently achieve low compliance, while diverse freeform classes reach satisfactory performance across multiple applications despite being automatically chosen without considering their compatibility.

- **Chapter 7: Generative Deep Learning with Diverse and Quality Multiclass Data as Seeds for Novel Multiscale Structures.**

  The “bottom-up” frameworks have received less attention than their “top-down” counterparts because they are, to some extent, reliant on traditional TO methods – both optimize the geometric parameters of the structures. Despite this, they show the most promise in being extensible to complex, real-world applications. To improve the generality and efficiency of these approaches even further, we propose a generative deep learning-based approach to expand the design space not only tenfold (to 50 basis classes) compared to the previous task but also to incorporate both trusses and isosurfaces with a unified, latent representation. First, we show that METASET can consider multiple classes of unit cells simultaneously using signed distance fields. Furthermore, we introduce a new quality-augmented diversity metric such that a tradeoff between the diversity (to improve coverage in shape and property spaces) and quality (to aid model training and design synthesis) can be tuned during subset selection. We then illustrate that multiclass shape blending can efficiently expand a diverse and high quality subset into a huge shape dataset containing trusses, isosurfaces, and hybrids of both. Subsequently, we demonstrate that a Wasserstein GAN with gradient penalty (WGAN-GP) can map multiclass shapes to a 10-dimensional latent space without mode collapse. The latent representation is harnessed in multiscale TO to optimize the choice of whether trusses, isosurfaces, or hybrids should be used in heterogeneous designs. In several case studies, we validate that smooth grading between neighboring microstructures are guaranteed, thereby achieving lower maximum stress, even when multiple morphology types are present in the design because of the well-trained WGAN-GP.

Finally, we summarize our contributions and conclude our dissertation with recommendations for future works in Chapter 8. Additional information, such as proposed algorithms, are listed in the Appendices.

CHAPTER 2

Technical Background

2.1. Homogenization Theory for Linear Elastic Unit Cells

In contrast to their homogeneous structural counterparts, the superior behaviors of metamaterials are derived from their multiscale architecture rather than their constituent materials. However, multiscale structures naturally include geometrical features that are far smaller and more complex than the macroscopic designs. Directly simulating their performance with conventional finite element analysis (FEA) necessitates fine meshes, and hence often intractable computational expense. However, by assuming that the length scale of the unit cell, d, is well-separated from the scale of the macrostructure, L (i.e., $\epsilon = L/d \ll 1$), one can leverage homogenization theory for periodic materials to reduce a finely-discretized, multiscale problem to a coarser one at the global (macro-) scale. In this *homogenization-based problem*, a unit cell can be designed more efficiently by characterizing it by its homogenized, effective material properties, a volume average of the macroscopic properties assuming the unit cell repeated infinitely, rather than its geometry [80, 81]. Although the separation of scales is violated in heterogeneous multiscale structures where the microstructures are no longer periodic, homogenization is often used to bring the computational analysis into a single scale at the global level, sacrificing some accuracy for much greater efficiency. The homogenization-based approach is used in Chapters 5-7.

To calculate the homogenized properties of unit cells, the most common approach is the classic two-scale asymptotic expansion [82–84], which defines the microscale, or local unit cell, domain as, $\mathbf{y} = (y_1, y_2, y_3) \in Y$, and the macroscale, or global structure, domain as $\mathbf{x} = (x_1, x_2, x_3) \in \Omega$. In general, the volume average of a periodic function, or material property, is $\langle a_{ij} \rangle = \frac{1}{|\Omega|} \int_\Omega a_{ij}(\mathbf{x}, \mathbf{y}) d\Omega$. To derive it, recall the governing equations of linear elasticity:

$$\begin{aligned} \frac{\partial \sigma_{ij}^\epsilon}{\partial x_j^\epsilon} + f_i = 0, &\quad \text{in } \Omega, \\ \sigma_{ij}^\epsilon = E_{ijkl}^\epsilon \varepsilon_{kl}^\epsilon, &\quad \text{in } \Omega, \\ \varepsilon_{kl}^\epsilon = \frac{1}{2}\left(\frac{\partial u_k^\epsilon}{\partial x_l^\epsilon} + \frac{\partial u_l^\epsilon}{\partial x_k^\epsilon} \right), &\quad \text{in } \Omega. \end{aligned} \tag{2.1}$$

The first equation enforces equilibrium under the body force per unit volume, $\mathbf{f}$, the second is Hooke's law, where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\varepsilon}$ is the strain tensor, and $\mathbf{E}$ is the elasticity tensor, which is symmetric such that $E_{ijkl} = E_{jikl} = E_{ijlk} = E_{klij}$; the last describes the strain-displacement relationship, where $\mathbf{u}$ is the displacement. The subscripts indicate Einstein notation, and the superscript ϵ denotes Y-periodicity.

The key to the asymptotic method is to use the elliptical operator,

$$\mathcal{A}^\epsilon = \frac{\partial}{\partial y_i}\left(a_{ij}(\mathbf{y}) \frac{\partial}{\partial y_j} \right) = \epsilon^{-2}\mathcal{A}^1 + \epsilon^{-1}\mathcal{A}^2 + \epsilon^0 \mathcal{A}^3, \tag{2.2}$$

to expand the Y-periodic material properties of the unit cell, $a_{ij}(\mathbf{y}) = a_{ij}(\mathbf{x}, \mathbf{y})$. This allows, after some manipulation, the separation of the two length scales and the replacement of the macroscopic governing equations by homogenized local ones.

To begin, the displacement field is expanded in terms of both length scales as:

$$u_i^\epsilon(\mathbf{x}) = \epsilon^0 u_i^{(0)}(\mathbf{x},\mathbf{y}) + \epsilon^1 u_i^{(1)}(\mathbf{x},\mathbf{y}) + \epsilon^2 u_i^{(2)}(\mathbf{x},\mathbf{y}) + \cdots , \tag{2.3}$$

where $u_i^{(n)}(\mathbf{x},\mathbf{y})$ are Y-periodic functions. Due to the strain-displacement relationship, the strain tensor can also be expanded as

$$\begin{aligned}\varepsilon_{ij}^\epsilon(\mathbf{x}) &= \epsilon^{-1}\varepsilon_{ij}^{(0)}(\mathbf{x},\mathbf{y}) + \epsilon^0\varepsilon_{ij}^{(1)}(\mathbf{x},\mathbf{y}) + \epsilon^1\varepsilon_{ij}^{(2)}(\mathbf{x},\mathbf{y}) + \cdots \\ &= \frac{\epsilon^{-1}}{2}\left(\frac{\partial u_i^{(0)}}{\partial y_j} + \frac{\partial u_j^{(0)}}{\partial y_i}\right) \\ &+ \frac{\epsilon^0}{2}\left(\frac{\partial u_i^{(0)}}{\partial x_j} + \frac{\partial u_j^{(0)}}{\partial x_i} + \frac{\partial u_i^{(1)}}{\partial y_j} + \frac{\partial u_j^{(1)}}{\partial y_i}\right) \\ &+ \frac{\epsilon^1}{2}\left(\frac{\partial u_i^{(1)}}{\partial x_j} + \frac{\partial u_j^{(1)}}{\partial x_i} + \frac{\partial u_i^{(2)}}{\partial y_j} + \frac{\partial u_j^{(2)}}{\partial y_i}\right) + \cdots .\end{aligned} \tag{2.4}$$

Combining Eq. 2.3 with Hooke's law in Eq. 2.1 and the symmetry of $\mathbf{E}$ results in

$$\begin{aligned}\sigma_{ij}^\epsilon(\mathbf{x}) &= \epsilon^{-1}\sigma_{ij}^{(0)}(\mathbf{x},\mathbf{y}) + \epsilon^0\sigma_{ij}^{(1)}(\mathbf{x},\mathbf{y}) + \epsilon^1\sigma_{ij}^{(2)}(\mathbf{x},\mathbf{y}) + \cdots , \\ &= E_{ijkl}(\mathbf{y})\frac{1}{2}\Bigg[\epsilon^{-1}\left(\frac{\partial u_k^{(0)}}{\partial y_l} + \frac{\partial u_l^{(0)}}{\partial y_k}\right) \\ &+ \epsilon^0\left(\frac{\partial u_k^{(0)}}{\partial x_l} + \frac{\partial u_k^{(1)}}{\partial y_l}\right) + \epsilon^1\left(\frac{\partial u_k^{(1)}}{\partial x_l} + \frac{\partial u_k^{(2)}}{\partial y_l}\right) + \cdots\Bigg].\end{aligned} \tag{2.5}$$

Plugging the above into Eq. 2.1 gives:

$$\epsilon^{-2}\frac{\partial\sigma_{ij}^{(0)}}{\partial y_j} + \epsilon^{-1}\left(\frac{\partial\sigma_{ij}^{(0)}}{\partial x_j} + \frac{\partial\sigma_{ij}^{(1)}}{\partial y_j}\right) + \epsilon^0\left(\frac{\partial\sigma_{ij}^{(1)}}{\partial x_j} + \frac{\partial\sigma_{ij}^{(2)}}{\partial y_j} + f_i\right) + \cdots = 0. \tag{2.6}$$

By equations the ϵ^{-2} term to zero, a differential equation with the two length scales in separate terms can be derived as

$$\frac{\partial \sigma_{ij}^{(1)}}{\partial y_j} = \frac{\partial}{\partial y_j}\left[E_{ijkl}(\mathbf{y})\left(\frac{\partial u_k^{(0)}}{\partial x_l} + \frac{\partial u_k^{(1)}}{\partial y_l}\right)\right] = 0, \tag{2.7}$$

where $u_{ij}^{(0)} = u_i^{(0)}(\mathbf{x})$ is only in terms of the macroscale.

The general solution to Eq. 2.7 is

$$u_i^{(1)}(\mathbf{x},\mathbf{y}) = -\chi_i^{kl}(\mathbf{y})\frac{\partial u_k^{(0)}}{\partial x_i}(\mathbf{x}) + \xi_i(\mathbf{x}), \tag{2.8}$$

where $\xi_i(\mathbf{x})$ are constants of integration that can be considered zero in a first-order approximation. The characteristic Y-periodic displacement fields, $\boldsymbol{\chi}_i^{kl}$, are solutions of the auxiliary *local cell problem*:

$$\begin{aligned} &\int_Y E_{ijkl}\frac{\partial \chi_k^{mn}}{\partial y_i}\frac{\partial \nu_i}{\partial y_j}dY = \int_Y E_{ijmn}\frac{\partial \nu_i}{\partial y_j}dY, \quad \forall \nu_i \in Y, \\ &\frac{1}{|Y|}\int_Y \nu_i dY = 0, \end{aligned} \tag{2.9}$$

where $\boldsymbol{\nu}$ are arbitrarily chosen Y-periodic virtual displacement fields. The second equation is imposed so that, by the Fredholm alternative, $\boldsymbol{\chi}_i^{kl}$ will have a unique solution.

Finally, equating the ϵ^0 term in Eq. 2.6 to zero, taking the volume average and applying the divergence theorem leads to the homogenized macroscopic constitutive equation:

$$\bar{\sigma}_{ij} = E_{ijkl}^H \bar{\varepsilon}_{kl}, \tag{2.10}$$

where the *homogenized elasticity tensor*, $\mathbf{E}^H$ is

$$E^H_{ijkl} = \frac{1}{|Y|}\int_Y E_{ijpq}(\mathbf{y})\left(\delta_{pk}\delta_{ql} - \frac{\partial \chi^{kl}_p}{\partial y_q}\right) dY. \tag{2.11}$$

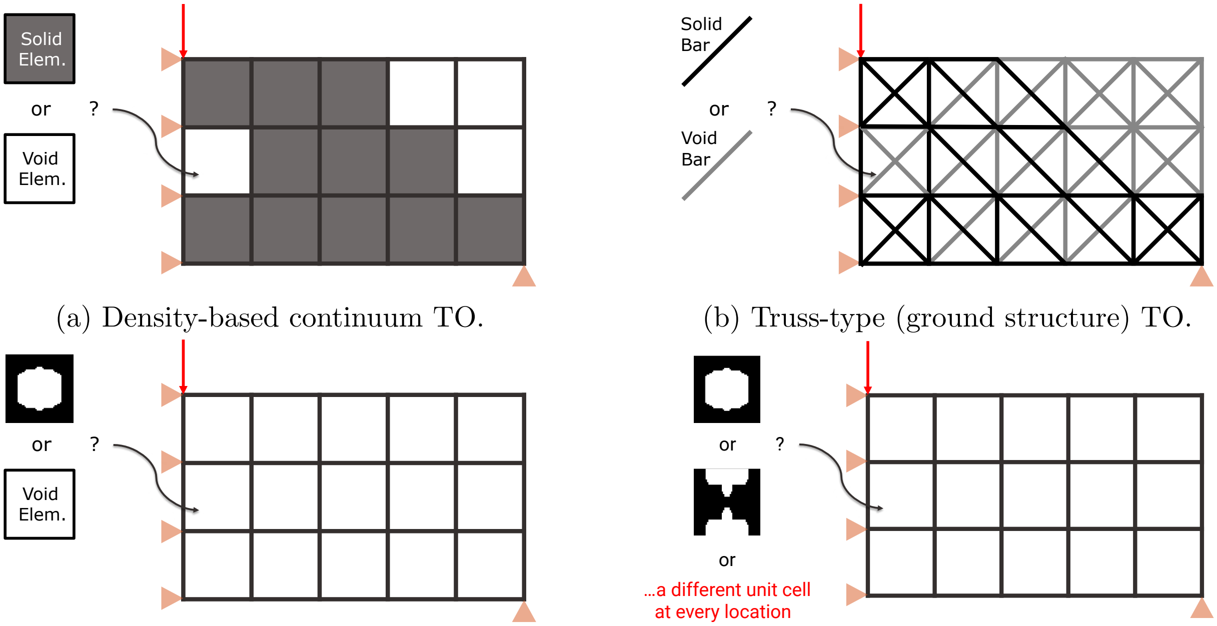
To solve Eq. 2.9 for $\boldsymbol{\chi}^{kl}_i$ and subsequently Eq. 2.11 for $\mathbf{E}^H$, periodic boundary conditions are applied while constraining $k, l = \{1, 2\}$ in 2D and $k, l = \{1, 2, 3\}$ in 3D for linearly independent combinations. Therefore, there are six possible load cases for 3D unit cells: $kl = \{11, 22, 33, 12, 13, 23\}$. In finite element methods, the loads typically correspond to six prescribed macroscopic unit strains:

$$\begin{aligned} &\bar{\varepsilon}^{11} = [1,0,0,0,0,0]^\top, \bar{\varepsilon}^{22} = [0,1,0,0,0,0]^\top, \bar{\varepsilon}^{33} = [0,0,1,0,0,0]^\top, \\ &\bar{\varepsilon}^{23} = [0,0,0,1,0,0]^\top, \bar{\varepsilon}^{13} = [0,0,0,0,1,0]^\top, \bar{\varepsilon}^{12} = [0,0,0,0,0,1]^\top. \end{aligned} \tag{2.12}$$

The above formulations and the work in this research are in the linear elastic domain with small strains. For an energy-based homogenization approach [81, 85], which sometimes used in lieu of the classical asymptotic method, the derivations are similar; an implementation can be found in Ref. [81]. For future extensions with geometric or material nonlinearity, however, a different homogenization approach would have to be applied.

2.2. Traditional Topology Optimization

Over the past few decades, topology optimization (TO) continues to gain momentum and increased applicability to a wide variety of complex design problems. In this section, we give a high level overview of four categories of TO that are relevant to this thesis, in order of increasing complexity: (1) density-based continuum, (2) truss-type (ground



(a) Density-based continuum TO.

(b) Truss-type (ground structure) TO.

(c) Periodic TO.

(d) Heterogeneous (aperiodic) TO.

Figure 2.1. Conceptualization of topology optimization (TO) methods covered in this work.

structure), (3) homogenization-based periodic multiscale, and (3) homogenization-based heterogeneous (aperiodic) multiscale. A conceptual demonstration of each type of design are illustrated in Figs. 2.1.

The classic continuum TO problem can be described as distributing materials within a design domain such that boundary conditions are met and a physics-based objective is optimized. The simplest formulation is that pioneered by Ole Sigmund, the solid isotropic material with penalization (SIMP) scheme (Fig. 2.1a), where the continuous design variables are the artificial, or "pseudo", densities of the elements of design domain, which is typically discretized into quadrilateral 4-node finite elements. For this reason, it is also known as density-based TO. The material property of each element can be either "solid" (homogeneous material) or “void” (removed material) is expressed as a continuous interpolation function that penalizes intermediate values of the densities through an exponent, p. For example, the Young’s modulus of each element e in a design containing one solid material can be described by the standard SIMP model as [43]

$$E_e = E(\rho_e) = \rho_e^p E, \tag{2.13}$$

where $\rho_e \in (0, 1]$ is the artificial density variable of element e such that it is greater than some small number ρ_{min} to avoid singularity when solving the finite element problem, and E is the overall Young’s modulus of the solid material.

Using the so-called modified SIMP scheme, singularity in the stiffness matrix can be circumvented by assigning a small positive number to the void phase as follows [86]:

$$E_e = E_e(\rho_e) = \rho_e^p (E_1 - E_0) + E_0, \tag{2.14}$$

where E_1 is the elastic modulus of the solid material, E_0 is a small value that represents the void phase, and p is the penalty coefficient to discourage intermediate densities within $[0, 1]$.

Using an interpolation scheme, the general TO problem under linear elasticity is expressed formally as

$$
\begin{aligned}
\underset{\boldsymbol{\rho}}{\text{minimize}} \quad & J_0(\boldsymbol{\rho}) \\
\text{subject to} \quad & \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f}, \\
& J_1 = V(\boldsymbol{\rho}) - \bar{V} \leq 0, \\
& 0 \leq \rho_e \leq 1, \quad e = 1, \ldots, N_{el},
\end{aligned}
\tag{2.15}
$$

where N_{el} is the total number of elements in the design domain, $\mathbf{u}$ is the displacement vector, and $\mathbf{f}$ the external load vector. The volume of the structure can be $V = \sum_{e=1}^{N_{el}} \rho_e$ constrained by an upper limit, $\bar{V}$. This volume constraint is frequently prescribed in TO problems where the target structural performance competes with the weight of the structure.

One such problem is compliance minimization, a common testbed for TO methods and equivalent to stiffness maximization, where the objective function is

$$
J_0(\boldsymbol{\rho}) = c(\boldsymbol{\rho}) = \sum_{e=1}^{N_{el}} \mathbf{u}_e^\top E_e \mathbf{k}_e^0 \mathbf{u}_e, \tag{2.16}
$$

where the stiffness in each element, E_e, is calculated using an interpolation scheme such as Eq. 2.14, $\mathbf{u}_e$ are the element displacements, and $\mathbf{k}_e^0$ are the unit local stiffness matrix.

2.2.1. Truss-Type Topology Optimization

The simple idea of adding and removing material within the design domain turns out to be a powerful formulation. Moving beyond continuum structures, the density-based concept can be adapted to design truss-type multiscale structures that are built from networks of bars connected by shared nodes (Fig. 2.1b). Instead of the quadrilateral finite elements, however, the design variables correspond to each bar in a pre-defined network, i.e., a ground structure. Similar to the SIMP method, the ground structure-based approaches gradually removes structurally unnecessary bars until convergence [87]. Multiple materials can also be assigned to different bars [29, 87].

2.2.2. Homogenization-Based Multiscale Topology Optimization

The traditional TO approach for continuum structures can also be extended to homogenization-based multiscale TO in a relatively straightforward manner by assuming that the unit cells are periodic and characterized by their macroscopic effective properties. Therefore, each macroscopic element houses a *unit cell* or *microstructure* rather than solid, homogeneous material (Fig. 2.1c). The design can then be split into two connected optimization problems at the macro- and micro- scales, solved either hierarchically or concurrently.

Taking compliance minimization as an example objective, the homogenized properties can be integrated into the macroscale problem by now assembling the global stiffness matrix, $\mathbf{K}$, in Eq. 2.17 using the homogenized local stiffness of each unit cell. That is, we now have

$$c(\boldsymbol{\rho}^M, \boldsymbol{\rho}^m) = \sum_{e=1}^{N_{el}} \mathbf{u}_e^\top E_e(\boldsymbol{\rho}^M)\mathbf{k}_e^H(\boldsymbol{\rho}^m)\mathbf{u}_e, \tag{2.17}$$

where $\boldsymbol{\rho}^M$ and $\boldsymbol{\rho}^m$ are the density variables at the macro- and micro- scales, respectively, $E_e(\boldsymbol{\rho}^M)$ can be found using a density-based interpolation scheme such as Eq. 2.14, and $\mathbf{k}_e^H(\boldsymbol{\rho}^m)$ is the effective stiffness of the unit cell at macro-element e. For details of MATLAB implementation of homogenization-based, periodic design, the reader is encouraged to consult Refs. [81, 88].

By defining multiple sub-problems at the microscale, i.e., by designing $\boldsymbol{\rho}^{m,j}, \forall j \in [1, 2, \ldots, J]$, where J is the total number of different microstructures and therefore sub-problems, one can also extend multiscale TO to design *heterogeneous* (aperiodic) multiscale structures. However, care must be taken to ensure that these disparate microstructures can connect to each other so that structural integrity is not compromised. Note that here we call the microscale structures "microstructures" rather than "unit cells", since periodicity no longer applies.

With great computational resources, it is also possible to optimize fully aperiodic systems where the microscale topologies can differ at each macro-element (Fig. 2.1d). Using traditional TO approaches, each unit cell becomes a sub-problem, and its homogenized properties must also be re-calculated at every iteration [80]. As one can imagine, this can quickly incur enormous, and potentially intractable, expense. For a review of existing traditional TO methods for heterogeneous designs and their challenges, please refer to those covered in Chapter 6.2.

2.3. Data-Driven Synthesis of Heterogeneous Multiscale Structures

Recent efforts strive to alleviate the cost of heterogeneous design with data-driven frameworks. To this end, two approaches – "top-down" and "bottom-up" – exploit the use of unit cell databases to accelerate the design process without sacrificing much design freedom. Since the geometry and homogenized properties are pre-computed for each sample, i.e., unit cell, the datasets can be leveraged for machine and deep learning, utilized as look-up tables, and are recyclable for multiple applications. At their core, both approaches are still rooted in the homogenization-based TO method.

In the *bottom-up* data-driven framework (Fig. 2.2(c)) the parameters at the microstructural level, e.g., volume fraction and geometric class, are directly used as design variables in a single optimization step. Its general optimization problem is the same as Eq. 2.17 above for a compliance minimization problem. One difference is that, instead of costly, nested homogenization loops at each iteration, the properties $\mathbf{k}_e^H(\boldsymbol{\rho}^m)$ of all microstructures are predicted by a surrogate or machine learning model of the material law. In some cases, the design representation of the microstructures, $\boldsymbol{\rho}^m$, can be simplified into a lower-dimensional space either through parameters specific to a class (e.g., rod thickness for trusses) or through machine and deep learning methods (e.g., latent variables from generative models). These steps greatly expedite the design process and therefore allows a larger number of microstructures, $J \leq N_{el}$, to be introduced into the problem compared to using traditional TO. However, it is still subject to the same connectivity challenges and, as a result, existing bottom-up methods consider only a small number of pre-selected microstructures from a single geometric class, leading to sub-optimal solutions.

On the other hand, *top-down* methods shown in Fig. 2.2(a) design the macro- and micro-structures are designed sequentially. First, the macroscale topology and spatial distribution of homogenized material properties are concurrently optimized using gradient-based methods by taking the properties as the design variables. Assuming that $J = N_{el}$

(a different microstructure at every location), this can be written as:

$$\begin{aligned} \underset{\mathbf{k}^T}{\text{minimize}} \quad & \sum_{e=1}^{N_{el}} \mathbf{u}_e^\top \mathbf{k}_e^T \mathbf{u}_e \\ \text{subject to} \quad & \boldsymbol{K}(\mathbf{k}_e^T)\boldsymbol{u} = \boldsymbol{f}, \\ & -\phi(\mathbf{k}_e^T) \leq 0, \end{aligned} \tag{2.18}$$

where the effective element stiffness tensors $\mathbf{k}^T$ are design variables, and the signed L2 distance field ϕ of the property space is a constraint to limit the properties to those in the unit cells dataset.

Next, the optimized properties serve as targets, $\mathbf{k}_e^T$, and a combinatorial optimization technique can be used to search for existing microstructures in the database whose properties match those values. These chosen microstructures are then "tiled" into the multiscale structure. An example of this combinatorial problem is:

$$\underset{\boldsymbol{l}}{\text{minimize}} \quad \frac{1}{N_{el}} \sum_{e}^{N_{el}} \|\mathbf{k}_e^H(l_e) - \mathbf{k}_e^T\|_2^2 \tag{2.19}$$

where $\mathbf{k}_e^H(l_e)$ are the effective stiffness tensors of the unit cells selected from the dataset, which are represented by their indices $l_e \in \{1, 2, \ldots, G\}$. The total number of cells in the dataset is G while the number to be assembled into the structure is N_{el}. As G and N_{el} increase, the search space also grows. Hence, top-down frameworks should be able to explore such large spaces in a scalable manner. Moreover, like with any other design method for heterogeneous structures, they should also be able to handle compatibility constraints on neighboring cells to ensure well-connected structures.

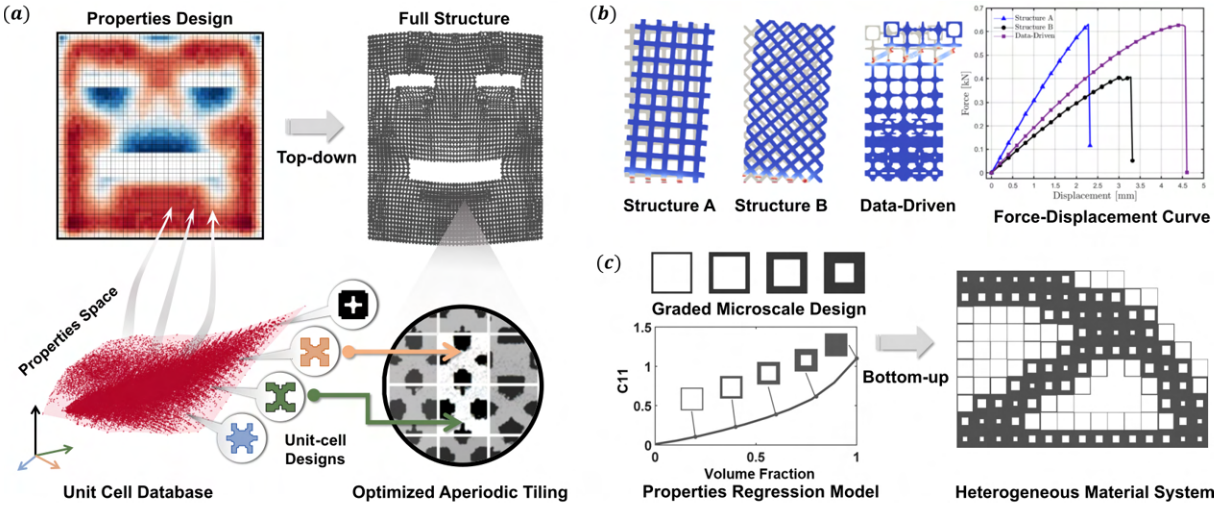


Figure 2.2. Examples of two approaches in data-driven design of heterogeneous systems: (a,b) "top-down", which follows a two-stage (properties design and tiling) process [1, 2], and (c) "bottom-up" [3], which directly optimizes the geometries of each microstructure. Figure credits: Liwei Wang and Dr. Daicong Da.

Despite considerable design freedom and efficiency (relative to traditional methods), top-down methods assume that the heterogeneous microstructures have little impact on the mechanical properties of their neighbors, and have been confined to designs in the realm of linear elasticity. In comparison, bottom-up frameworks show more promise in accommodating a wider range of applications, including those with nonlinear mechanics, since they do not need to first identify target properties, which may not necessarily be scalar. Applications of these two different approaches are exemplified in Fig. 2.2, and reviewed in Chapters 5 (top-down) and 7 (bottom-up).

Part 1

Multimaterial Lattice Structure Topology Optimization

CHAPTER 3

Robust Topology Optimization of Multi-Material Lattice Structures

3.1. Introduction

Enabled by advancements in multi-material additive manufacturing, lightweight lattice structures consisting of networks of strut-based unit cells have gained popularity due to their extraordinary performance and wide array of functions. Because they consist of many structural members and are naturally porous, they hold desirable properties, e.g., low weight and material cost, while exhibiting extraordinary mechanical, thermal or acoustic properties. In recent years, additive manufacturing (AM) has allowed easier manufacture of geometrically complex lattices, and as a result the structures have gained traction as both micro-architectured metamaterials and mesostructures.

At the microscale, for example, lattice metamaterials have been designed for negative Poisson's ratio [31, 89, 90], acoustic manipulation [10, 91] and energy absorption [27]. Micro- and mesoscale lattice structures have also been embedded into global structures to achieve multifunctional heat exchangers [17, 18], and parts with high energy absorption [29, 92–94] and low weight [12, 95]. In particular, lattice structures are exceptionally popular in the biomedical field due to their porosity, which allows biocompability with organic tissue, high performance-to-weight ratio, and ease of customization [14–16]. By exploiting the advancements in multi-material AM, greater performance improvements [29], like graded stiffness [31] and a larger range of stiffness [96], are possible.

This work proposes a density-based robust topology optimization method for multi-material lattice structures under any combination of material and load uncertainties. The method utilizes a new generalized material interpolation scheme for an arbitrary number of materials, and employs univariate dimension reduction and Gauss-type quadrature to quantify and propagate uncertainty. By formulating the objective function as a weighted sum of the mean and standard deviation of compliance, the tradeoff between optimality and robustness can be studied and controlled. Examples of a cantilever beam lattice structure under various material and load uncertainty cases exhibit the efficiency and flexibility of the approach. The accuracy of univariate dimension reduction is validated by comparing the results to the Monte Carlo approach. The general framework established here can form the basis of future works in both continuum and multiscale structural design with and without uncertainty.

3.2. Literature Review

In this chapter, we focus on multi-material lattice structures whose initial ground structures are composed of periodic truss unit cells. Over the course of optimization, as struts, or bars, are either assigned a material or removed altogether, the structure becomes heterogeneous. Due to the large number of struts as well as the additional materials, the design space expands significantly. Thus, to design these structures, computational methods, e.g., topology optimization (TO), which automatically lays out the materials, are highly desirable. While multi-material TO has become more prevalent, most methods are not applicable to lattice structures. The color level-set [97] and multiphase phase-field [98] methods require solving the Hamilton-Jacobi and Cahn-Hilliard partial differential equations, respectively, on a continuous domain. Therefore, they cannot design lattice structures in which each strut is a discrete element. Techniques for discrete elements like the multi-material genetic algorithm [99] and pseudo-sensitivity [100] approaches exist, but are prohibitive for large problems. Zhang et al. [87] used the ground structure method to iteratively remove members, applying a new variable update method to effectively optimize multi-material trusses under multiple constraints.

Similar to Ref. [87], this work takes the strut member as an element and can either remove or assign materials to each. Departing from the above methods, however, a density-based approach is employed by calculating the elastic modulus based on the well-known 3-phase standard solid isotropic material with penalization (SIMP) scheme by Bendsøe and Sigmund [43]. It is also inspired by the modified version of SIMP that replaces the void elastic modulus with a small number to avoid singularity during finite element analysis (FEA) [86]. The formulation in this chapter takes SIMP a step further by allowing multiple materials.

A few others have extended Sigmund's 3-phase interpolation schemes to M solid phases and can be broadly categorized into: 1) schemes that add design variables for each new material, similar to standard and modified SIMP, and 2) schemes that limit the number of design variables in order to reduce cost. Stegmann and Lund [101] employed a model of the first category, initially allowing more design variables but later reducing the number using "patch" variables. They, however, followed the less stable standard SIMP formulation. Gaynor et al. [102] also proposed a first category scheme that, despite considering practical PolyJet manufacturing constraints, unrealistically requires the difference between the elastic modulus of each material to be equal. From the second category, Yin and Ananthasuresh [103] used a peak function to transform one design variable into the modulus of multiple materials. Although it does not increase the design variables, the function contains horizontal slopes that may cause instability, and includes several curve parameters that must be meticulously selected by the user. Another method is Zuo and Saitou's ordered scheme [104], which avoids instability in its power functions but has a complicated formulation and fluctuations in the optimization history.

It should be noted that although using less design variables does reduce the time needed to evaluate the modulus and to update the variables, the vast majority of TO, especially in large problems, is spent in FEA, which depends on the number of elements in the domain. As this number generally remains the same throughout optimization regardless of how many materials exist, it is not necessary to force a more complicated scheme in exchange for less design variables.

Furthermore, the uncertainty in the materials and loads are considered in this work. In AM, uncertainty is inherent in the material (e.g., variability in the powder of laser powder bed fusion) and the build process (which can lead to inconsistent geometry). In lattice structures, the strut dimensions and mechanical properties can vary greatly depending on the processing conditions [105, 106]. Park et al. [107] quantified the variability in the strut geometry by calculating the probabilistic distribution of effective diameters, then deriving the effective elastic modulus.

TO methods that consider geometric [108–110] and material [111, 112] uncertainties exist for single material continuous problems. There is relatively more research for random loads, such as Refs. [113–115]. Nonetheless, the majority of these methods use level-set TO, projection filters to model uncertainty, or Monte Carlo to quantify uncertainty — all of which are difficult or impossible with multi-material lattice structures consisting of a large number of strut elements.

3.3. Contributions

In response to the gaps identified above, we propose a new multi-material interpolation scheme that is generalized for arbitrary numbers of materials, easy to implement, and based on modified SIMP for more stability. The scheme is demonstrated on truss structures here but easily applicable to both continuum and density-based multiscale structures (defined in Chapter 2.2). For example, we extend it to design multiclass heterogeneous structures in Chapter 6) by designating each microstructural geometric class as a "material" in the scheme.

To account for uncertainties, we use the elastic moduli of the materials directly as the uncertain parameters. By doing so, both the uncertainty in the material and the built geometry can be simultaneously considered by prescribing appropriate probability distributions. Load uncertainty that can arise during application is also added for more robust designs. Both material and load influence the part performance, which is used as the objective function in TO; considering their variability can result in different and more robust designs. Rather than Monte Carlo or additional filters, we suggest a weighted sum of the mean and standard deviation of the performance criteria, e.g., compliance, as the objective function, where the statistical moments are calculated using univariate dimension reduction (UDR) [116] and Gauss-type quadrature sampling.

The new generalized multi-material interpolation scheme is proposed in Section 3.4. In Section 3.5, the deterministic formulation of the TO using the scheme is introduced, followed by the robust formulation and details of uncertainty quantification in Section 3.6. Numerical examples of a lattice structure with a global cantilever beam shape under different uncertainty cases is shown and discussed in Section 3.7. Finally, the conclusions are presented in Section 3.8.

3.4. Generalized Multi-Material Interpolation Scheme

The proposed multi-material interpolation scheme follows the SIMP method, where the continuous design variables are the artificial densities of the elements. The material property of each element is expressed as a material interpolation function that penalizes intermediate values of the densities through an exponent, p. For example, the Young's modulus of each element e in a design containing one solid material can be described by the standard SIMP model as [43]

$$E_e = E(\rho_e) = \rho_e^p E, \tag{3.1}$$

where $\rho_e \in (0, 1]$ is the artificial density variable of element e such that it is greater than some small number ρ_{min} to avoid singularity when solving the finite element problem, and E is the overall Young's modulus of the solid material.

Using the so-called modified SIMP scheme, singularity in the stiffness matrix can be circumvented by assigning a small positive number to the void phase as follows [86]:

$$E_e = E_e(\rho_e) = \rho_e^p(E_1 - E_0) + E_0, \tag{3.2}$$

where E_1 is the elastic modulus of the solid material, E_0 is a small value that represents the void phase, and p is the penalty coefficient to discourage intermediate densities within $[0, 1]$.

Gibiansky and Sigmund [117] showed the following 3-phase (two distinct solid materials and one void phase) scheme for composites that includes a second penalization parameter, q:

$$E(\rho_{e\cdot 1}, \rho_{e\cdot 2}) = \rho_{e\cdot 1}^p[(1 - \rho_{e\cdot 2}^q)E_1 + \rho_{e\cdot 2}^q E_2]. \tag{3.3}$$

where E_2 is the modulus of the second solid material, and the design variables are $\rho_{e\cdot 1}, \rho_{e\cdot 2} \in [0, 1]$. This model was tested by Stegmann and Lund in Ref. [101], where the two penalty parameters had to be carefully chosen in order to converge to a global optimum. Moreover, Eq. 3.3 follows the standard SIMP model instead of the more stable modified version in Eq. 3.2.

In response, this work amalgamates Eq. 3.3 and the modified SIMP scheme for three phases, without the inclusion of another penalty parameter, as follows:

$$\begin{aligned} E_e(\rho_{e\cdot 1}, \rho_{e\cdot 2}) &= \rho_{e\cdot 1}^p[\rho_{e\cdot 2}^p E_2 + (1 - \rho_{e\cdot 1}^p)E_1] + (1 - \rho_{e\cdot 1}^p)E_0 \\ &= \rho_{e\cdot 1}^p \rho_{e\cdot 2}^p(E_2 - E_1) + \rho_{e\cdot 1}^p(E_1 - E_0) + E_0, \end{aligned} \tag{3.4}$$

From the second line, it is clear that the first set of variables, $\rho_{e\cdot 1}$, determines the optimal topology of the overall structure, while the second set, $\rho_{e\cdot 2}$, selects the material at each solid element. This idea is demonstrated for a three phase scenario in Table 3.1.

Table 3.1. Expansion of the proposed interpolation scheme for three phases ($M = 2$).

Density Variables	Young's Modulus
$\rho_{e\cdot 1} = 0, \rho_{e\cdot 2} = 0$	E_0
$\rho_{e\cdot 1} = 1, \rho_{e\cdot 2} = 0$	E_1
$\rho_{e\cdot 1} = 1, \rho_{e\cdot 2} = 1$	E_2

This work further proposes extending Eq. 3.4 to a general material interpolation scheme that can handle any number of materials:

$$E_e(\rho_{e\cdot 1}, \rho_{e\cdot 2}, \ldots, \rho_{e\cdot M}) = \sum_{j=1}^{M} \left((E_j - E_{j-1}) \prod_{k=1}^{j} \rho_{e\cdot k}^{p} \right) + E_0, \tag{3.5}$$

where M is the number of distinct, non-void materials. When $M = 1$ and $M = 2$, this scheme simplifies to the established expressions Eqs. 3.2 and 3.4.

Using the proposed Eq. 3.5, M design variables per element are required, but the number of elements in the FEA simulation does not change when more materials are introduced. Since the majority of the computational expense in TO is typically attributed to FEA, the cost of additional design variables due to multiple materials is relatively small. Based on the stable modified SIMP scheme, the proposed formulation is easily integrated into problems with any number of materials.

3.5. Deterministic Topology Optimization

The density-based deterministic optimization problem to minimize compliance is:

$$
\begin{aligned}
\underset{\boldsymbol{\rho}}{\text{minimize}} \quad & J_0 = c(\boldsymbol{\rho}) = \mathbf{u}^\top \mathbf{K}(\boldsymbol{\rho})\mathbf{u} \\
\text{subject to} \quad & \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f}, \\
& J_1 = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\nu_e) - \bar{J}_1 \leq 0, \\
& J_2 = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\rho_{\cdot 2}\nu_e) - \bar{J}_2 \leq 0, \\
& \vdots \\
& J_M = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\rho_{e\cdot 2}\dots\rho_{e\cdot M}\nu_e) - \bar{J}_M \leq 0, \\
& 0 \leq \rho_{e\cdot i} \leq 1, \quad e = 1, \dots, N_{el}, \quad i = 1, \dots, M,
\end{aligned}
\tag{3.6}
$$

where $\boldsymbol{\rho}$ is the vector containing all design variables $\rho_{e\cdot i}$, c is the compliance of the structure, v_e is the volume of element e, $\mathbf{u}$ is the displacement vector, $\mathbf{f}$ the external load vector, and N_{el} is the total number of elements. The stiffness in each element, E_e, is calculated using the proposed interpolation scheme (Eq. 3.5), and the global stiffness matrix is $\mathbf{K}(\boldsymbol{\rho}) = \sum_{e=1}^{N_{el}} E_e \mathbf{k}_e$, with the unit local stiffness matrix $\mathbf{k}_e$. The volume ratios of solid materials in each element are constrained by the equations $J_1, J_2, \dots, J_M$ and the upper bounds of those ratios, $\bar{J}_1, \bar{J}_2, \dots, \bar{J}_M$.

Since the TO problem is solved with gradient-based methods, the derivative of the compliance $c(\rho)$ with respect to the density variables, $\rho_{e\cdot i}$, is derived using the adjoint variable method [43] as

$$\frac{\partial c(\boldsymbol{\rho})}{\partial \rho_{e\cdot i}} = \boldsymbol{u}^\top \frac{\partial \mathbf{K}(\boldsymbol{\rho})}{\partial \rho_{e\cdot i}} \boldsymbol{u} = -p\rho_{e\cdot i}^{p-1} \sum_{j=i}^{M} \Big[(E_j - E_{j-1}) \prod_{k=1,k\neq i}^{j} \rho_{e\cdot k}^{p}\Big] \boldsymbol{u}_e^\top \mathbf{K}_e \boldsymbol{u}_e, \tag{3.7}$$

where $\boldsymbol{u}_e$ is the element displacement vector of element e.

3.6. Robust Topology Optimization

To capture uncertainty mathematically, we introduce the random variables $\boldsymbol{\omega}_m \in \boldsymbol{\Theta}^M$ and $\boldsymbol{\omega}_f \in \boldsymbol{\Theta}^F$, where $\boldsymbol{\Theta}^M$ and $\boldsymbol{\Theta}^F$ are the random sample spaces [118] corresponding to uncertainties in the material properties and loads, respectively. Here M is the number of uncertain solid phases as defined previously, while F is the number of uncertain external loads.

The robust TO is then formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{\rho}}{\text{minimize}} \quad & J_0 = \mu\big(c(\boldsymbol{\rho}, \boldsymbol{\omega}_m, \boldsymbol{\omega}_f)\big) + \beta\sigma\big(c(\boldsymbol{\rho}, \boldsymbol{\omega}_m, \boldsymbol{\omega}_f)\big) \\ \text{subject to} \quad & \mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\omega}_m)\mathbf{u} = \mathbf{f}(\boldsymbol{\omega}_f), \\ & J_1 = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\nu_e) - \bar{J}_1 \leq 0, \\ & J_2 = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\rho_{e\cdot 2}\nu_e) - \bar{J}_2 \leq 0, \\ & \vdots \\ & J_M = \sum_{e=1}^{N_{el}} (\rho_{e\cdot 1}\rho_{e\cdot 2}\dots\rho_{e\cdot M}\nu_e) - \bar{J}_M \leq 0, \\ & 0 \leq \rho_{e\cdot i} \leq 1, \quad e = 1, \dots, N_{el}, \quad i = 1, \dots, M, \end{aligned} \tag{3.8}$$

where $c(\boldsymbol{\rho}, \boldsymbol{\omega}_m, \boldsymbol{\omega}_f)$ is the compliance under uncertainty, characterized by its mean and standard deviation, $\mu(c)$ and $\sigma(c)$. A weighted sum of these two statistical moments using a constant weight, β, results in a multi-objective optimization problem, which allows β to be increased in order to put more emphasis on minimizing the variation in compliance of the final design.

3.6.1. Uncertainty Quantification and Propagation

3.6.1.1. Multi-Material Interpolation Scheme under Uncertainty. When materials uncertainty exists, an extended version of the proposed multi-material scheme (Eq. 3.5) can be used to calculate the Young's modulus of each element, E_e:

$$E_e(\rho_{e\cdot 1}, \rho_{e\cdot 2}, \ldots, \rho_{e\cdot M}) = \sum_{j=1}^{M} \left[(E_j(\omega_j) - E_{j-1}(\omega_{j-1})) \prod_{k=1}^{j} \rho_{e\cdot k}^{p} \right] + E_0, \tag{3.9}$$

where each modulus $E_j(\omega_j)$ is a realization of its corresponding random variable, $\boldsymbol{\omega}_j \in \boldsymbol{\omega}_m$. In the case when there is no materials uncertainty (i.e., there is only load uncertainty, or the problem is deterministic), the above simply reverts to Eq 3.5 due to the lack of $\boldsymbol{\omega}_m$.

3.6.1.2. Calculation of Statistical Moments. The material and load uncertainties are propagated using UDR, which, when combined with Gauss-type quadrature, efficiently reduces the moments of multivariate probability distributions into a weighted sum of univariate functions [116]. For the method to be accurate, all random variables should be chosen such that there are no strong interactions between them [119, 120]. Furthermore, it is assumed that all of the random variables are mutually independent, and for notational simplicity, they are not separated into $\boldsymbol{\omega}_m$ and $\boldsymbol{\omega}_f$. Instead, they are lumped into $\boldsymbol{\omega} =$

$[\boldsymbol{\omega}_m, \boldsymbol{\omega}_f] \in \boldsymbol{\Theta}^Z$, where Z is the total number of random variables. For example, if both material and load uncertainties are considered, $Z = M + F$.

The statistical moments $\mu(c)$ and $\sigma^2(c)$ can be expressed as

$$\mu\big(c(\boldsymbol{\rho}, \boldsymbol{\omega})\big) = \int_{\boldsymbol{\Theta}^Z} c(\boldsymbol{\rho}, \boldsymbol{\omega}) f_{\boldsymbol{\omega}}(\boldsymbol{\omega})\, d\boldsymbol{\omega}, \tag{3.10}$$

$$\sigma^2\big(c(\boldsymbol{\rho}, \boldsymbol{\omega})\big) = \int_{\boldsymbol{\Theta}^Z} \big[c(\boldsymbol{\rho}, \boldsymbol{\omega}) - \mu\big(c(\boldsymbol{\rho}, \boldsymbol{\omega})\big)\big]^2 f_{\boldsymbol{\omega}}(\boldsymbol{\omega})\, d\boldsymbol{\omega}, \tag{3.11}$$

where $f_{\boldsymbol{\omega}}(\boldsymbol{\omega})$ is the joint probability density function of the random variables.

With UDR, the compliance can be approximated by reducing it to a sum of univariate functions. In each function, all random variables are held equal to their mean except for one, ω_j, such that

$$\begin{aligned} c(\boldsymbol{\rho}, \boldsymbol{\omega}) &\cong \sum_{j=1}^{Z} c\big(\boldsymbol{\rho}, \mu(\omega_1), \dots, \omega_j, \dots, \mu(\omega_z)\big) - (Z-1)c\big(\boldsymbol{\rho}, \mu(\omega_1), \dots, \mu(\omega_z)\big) \\ &= \sum_{j=1}^{Z} c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j=\omega_j}) - (Z-1)c(\boldsymbol{\rho}, \boldsymbol{\mu}). \end{aligned} \tag{3.12}$$

Note that the univariate functions $c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j=\omega_j})$ are independent since the random variables are independent, and therefore Eqs. 3.10 and 3.11 can be approximated as follows [121]:

$$\mu\big(c(\boldsymbol{\rho}, \boldsymbol{\omega})\big) = \sum_{j=1}^{Z} \mu\big(c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j=\omega_j})\big) - (Z-1)c(\boldsymbol{\rho}, \boldsymbol{\mu}), \tag{3.13}$$

$$\sigma^2\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big) = \sum_{j=1}^{Z} \sigma^2\big(c(\boldsymbol{\rho},\boldsymbol{\mu}|_{\mu_j=\omega_j})\big). \tag{3.14}$$

Then, utilizing one-dimensional Gauss-type quadrature to calculate the univariate moments in Eqs. 3.13 and 3.11, the final expressions are derived as

$$\mu\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big) \cong \sum_{j=1}^{Z}\left[\sum_{k=1}^{N} w_{j\cdot k}c(\boldsymbol{\rho},\boldsymbol{\mu}|_{\mu_j=l_{j\cdot k}})\right] - (Z-1)c(\boldsymbol{\rho},\boldsymbol{\mu}), \tag{3.15}$$

$$\sigma^2\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big) \cong \sum_{j=1}^{Z}\sum_{k=1}^{N} w_{j\cdot k}\left[c(\boldsymbol{\rho},\boldsymbol{\mu}|_{\mu_j=l_{j\cdot k}}) - \sum_{n=1}^{N} w_{j\cdot n}c(\boldsymbol{\rho},\boldsymbol{\mu}|_{\mu_j=l_{j\cdot n}})\right]^2, \tag{3.16}$$

where N is the number of quadrature nodes, $l_{j\cdot k}$ is the kth node of the random variable ω_j, and $w_{j\cdot k}$ is the corresponding weight.

Using these methods, the statistical moments of compliance can be estimated with $ZN+1$ deterministic finite element simulations, a value that is, as discussed earlier, not dependent on the number of elements in the design domain. In comparison to tensor product quadrature, which is an alternative univariate moment estimation method requiring N^Z FEA evaluations, UDR is much more efficient [108].

3.6.2. Sensitivity Analysis

For gradient-based optimization of the robust problem, the derivative of the weighted objective function with respect to each artificial density variable, $\rho_{e\cdot i}$, is

$$\begin{aligned}\frac{\partial J_0}{\partial \rho_{e\cdot i}} &= \frac{\partial \mu\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}} + \beta \frac{\partial \sigma\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}} \\ &= \frac{\partial \mu\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}} + \frac{\beta}{2\sigma\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)} \frac{\partial \sigma^2\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}},\end{aligned} \tag{3.17}$$

where $e = \{1, 2, \ldots, N_{el}\}$ and $i = \{1, 2, \ldots, M\}$.

As in the previous section, when applying UDR to estimate the sensitivities of the mean and variance, the random variables are fixed at their mean except for ω_j, which is sampled using Gauss-type quadrature. Thus,

$$\frac{\partial \mu\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}} \cong \sum_{j=1}^{Z} \left[\sum_{k=1}^{N} w_{j\cdot k} \frac{\partial c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j = l_{j\cdot k}})}{\partial \rho_{e\cdot i}} \right] - (Z-1) \frac{\partial c(\boldsymbol{\rho},\boldsymbol{\mu})}{\partial \rho_{e\cdot i}}, \tag{3.18}$$

$$\begin{aligned}\frac{\partial \sigma^2\big(c(\boldsymbol{\rho},\boldsymbol{\omega})\big)}{\partial \rho_{e\cdot i}} \cong 2 \sum_{j=1}^{Z} \sum_{k=1}^{N} w_{j\cdot k} &\left[c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j = l_{j\cdot k}}) - \mu\big(c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j = l_{j\cdot k}})\big) \right] \\ &\left[\frac{\partial c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j = l_{j\cdot k}})}{\partial \rho_{e\cdot i}} - \sum_{n=1}^{N} w_{j\cdot n} \frac{\partial c(\boldsymbol{\rho}, \boldsymbol{\mu}|_{\mu_j = l_{j\cdot n}})}{\partial \rho_{e\cdot i}} \right].\end{aligned} \tag{3.19}$$

Similar to Eq. 3.7 but with the addition of random variables, the derivative of compliance under uncertainty is

$$\begin{aligned}\frac{\partial c(\boldsymbol{\rho},\boldsymbol{\omega})}{\partial \rho_{e\cdot i}} &= \boldsymbol{u}^{\top} \frac{\partial \mathbf{K}(\boldsymbol{\rho},\boldsymbol{\omega})}{\partial \rho_{e\cdot i}} \boldsymbol{u} \\ &= -p\rho_{e\cdot i}^{p-1} \sum_{q=i}^{M} \left[\big(E_q(\omega_q) - E_{q-1}(\omega_{q-1})\big) \prod_{k=1,k\neq i}^{q} \rho_{e\cdot k}^{p} \right] \boldsymbol{u}_e^{\top} \mathbf{K}_e \boldsymbol{u}_e,\end{aligned} \tag{3.20}$$

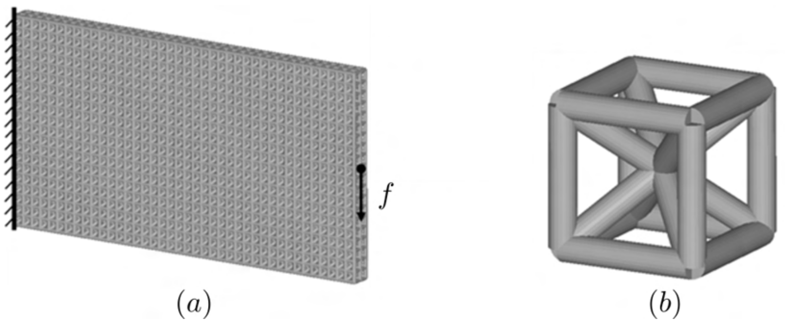


Figure 3.1. (a) Boundary conditions; (b) initial lattice unit cells.

where $\omega_q \in \boldsymbol{\omega}_m$. This derivative can be calculated after solving $\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\omega}_m) = \mathbf{f}(\boldsymbol{\omega}_f)$, which is deterministic due to UDR since $\boldsymbol{\omega}_m$ and $\boldsymbol{\omega}_f$ are either held at the mean or equal to the quadrature node according to Eqs. 3.18 and 3.19.

3.7. Numerical Examples

The robust multi-material formulation was demonstrated with a 4-phase 3D cantilever beam under different combinations of material and load uncertainty, where each strut of the lattice structure is one FEA element with three design variables. In all examples, the initial structure (Fig. 3.1(a)) has dimensions $200mm \times 100mm \times 20mm$ and a total of $19,502$ elements. It is created by repeating the cubic unit cell in Fig. 3.1(b), which is $10mm \times 10mm \times 10mm$ and consists of 20 elements with $2mm$ diameters, and removing overlapping struts. The uniformity of the unit cell is not guaranteed in the optimized result, however, as individual strut elements can be removed if $\rho_{e \cdot 1} = 0$. Hence, our design is initialized as a periodic lattice structure but becomes *heterogeneous* as bars can be assigned as different solid phases or removed altogether.

The FEA at each iteration is performed at the global structure scale in Altair HyperWorks OptiStruct, which has the capability to generate and simulate lattice structures, including differently oriented struts, as part of their lattice optimization module. Each strut is modeled as a 1D `CBEAM` element, and has `PBEAML` properties with `TYPE=ROD`. The material properties and their volume ratio upper limits are listed in Table 3.2, and the boundary conditions are shown in Fig. 3.1(a).

The optimization problems are solved via the method of moving asymptotes (MMA) [122], which is efficient despite the large number of design variables and volume constraints. The design variables $\rho_{e \cdot i}$ are initially set equal to their corresponding volume ratio limits, e.g., $\rho_{e \cdot 1} = \bar{J}_1$. To simplify the calculations, the volume of each element, v_e, was assumed to equal one. In these examples, the generalized multi-material interpolation scheme (Eq. 3.5 or 3.9) does not require high penalization, with $p = 2$ sufficing to drive away most intermediate (gray) densities. However, it is necessary to significantly increase the MMA parameter `asydecr` from the default value to 0.97 in order to decrease oscillations in the optimization history. The `asyincr` parameter is lowered to 1.03 for faster convergence with a slight sacrifice to the compliance, the initial `movelimit` is 1.2, and the rest of the parameters are kept at the default values recommended by Svanberg [123]. The optimization is stopped when the relative difference between the objective function values between two iterations is less than 1e-6.

Table 3.2. Material properties of the 4-phase examples.

Material	Color	Mean Young's Modulus (MPa)	Poisson's Ratio	Volume Constraint
Void	-	1e-8	0.3	-
Soft	Green	500	0.3	10%
Medium	Blue	1000	0.3	10%
Hard	Yellow	2000	0.3	10%

For UDR, all examples use $N = 8$ Gauss quadrature points. Increasing the number of points beyond this does not noticeably improve the accuracy of the estimated statistical moments in comparison to Monte Carlo with $10,000$ samples; instead it escalates the computational cost unnecessarily. In addition, the weight β in the robust objective function J_0 (Eq. 3.8) is varied to generate Pareto optimal fronts of each example, showing the effects of the applied uncertainties. When $\beta = 0$, the objective function reverts to the deterministic one (Eq. 3.6) and the optimal topology in this case is essentially the deterministic one. As β is increased, more emphasis is put on minimizing the standard deviation of compliance, i.e., the uncertainty. Three cases of uncertainty are examined in the following sections: uncertainty in the 1) materials only, 2) load only, and 3) both materials and load.

3.7.1. Materials Uncertainty

In the first case, the Young's moduli of the four material phases are taken as normally distributed random variables. Thus, each modulus is prescribed a mean and standard deviation. The means of the soft, medium and hard materials are listed in Table 3.2; the standard deviations are set to $100MPa$ each, which is equivalent to 20%, 10%, and 5% of the mean values, respectively. The load is deterministic with a magnitude of $1.5N$ (Fig. 3.1(a)). Despite applying a high weight on the standard deviation of compliance, $\beta = 10$, the history of the objective function decreases stably and monotonically overall to 10.2% of the original value in 38 iterations (Fig. 3.2). In Fig. 3.3, the tradeoff between the the mean and standard deviation of compliance can be observed. When β is decreased, the mean decreases while the standard deviation increases, i.e., more optimal but less robust solutions are found. This relationship can also be seen in Fig. 3.4, where the robust solution has more of the medium (blue) and hard (yellow) materials, which were assigned lower relative standard deviations (10% and 5% of their means, respectively) than the soft material (20% of its mean).

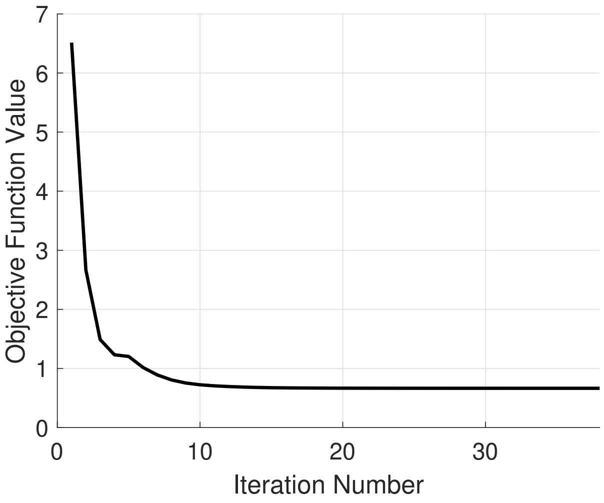


Figure 3.2. Optimization history of the objective function under materials uncertainty when $\beta = 10$.

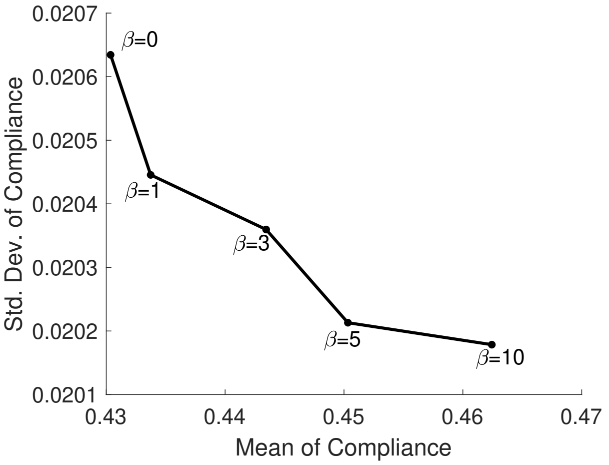


Figure 3.3. Pareto front under materials uncertainty with different objective function weights β.

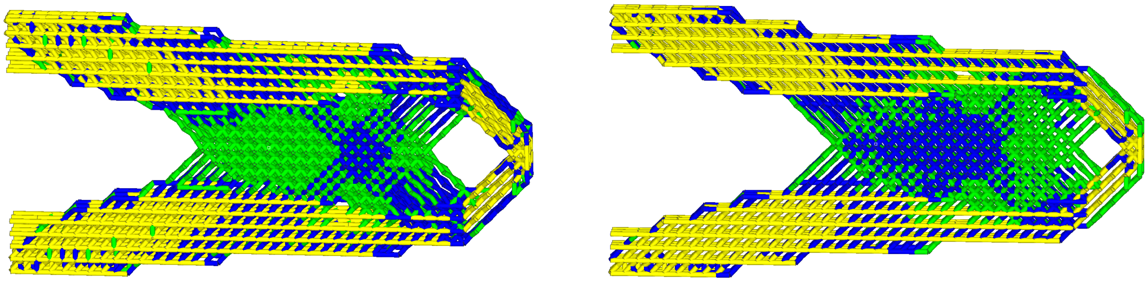


Figure 3.4. Optimal solution under materials uncertainty when (a) $\beta = 0$ (deterministic) and (b) $\beta = 10$ (robust).

To validate the statistical moments that were estimated using UDR, the final topologies were evaluated with $10,000$ Monte Carlo samples and compared in Table 3.3. The study shows that the UDR approximations are accurate within 3% of the Monte Carlo results.

Table 3.3. Mean and standard deviation (Std. Dev.) of the compliance ($N \cdot mm$) results under materials uncertainty using UDR with 8 quadrature nodes and Monte Carlo with $10,000$ samples.

Solution Type	Moment	UDR/Quadrature	Monte Carlo	Difference
Deterministic ($\beta = 0$)	Mean	0.43036	0.43027	0.02%
	Std. Dev.	0.02063	0.02087	1.15%
Robust ($\beta = 10$)	Mean	0.46244	0.46262	0.04%
	Std. Dev.	0.02018	0.02072	2.61%

3.7.2. Load Uncertainty

For the example under load uncertainty, two normally distributed random variables are considered: the magnitude and the angle of the external load applied (Fig. 3.5). The mean of the load magnitude is $1.5N$ and the standard deviation is $0.225N$, or 15% of the mean, while the mean and standard deviation of the angle are 0° and 10°, respectively.

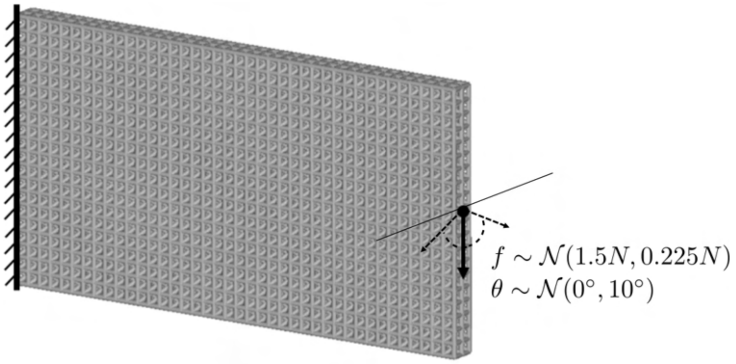

$f \sim \mathcal{N}(1.5N, 0.225N)$
$\theta \sim \mathcal{N}(0°, 10°)$

Figure 3.5. Boundary conditions with the random external load, which varies in both magnitude (f) and angle (θ).

As before, the objective history for $\beta = 10$ is fairly smooth and stable, and the final value is 5.67% of the initial compliance after 37 iterations (Fig. 3.6).

Unlike the case with only materials uncertainty, the robust designs are dramatically different, with long horizontal bars of the hard (yellow) material added in order to brace the cantilever against loads with angles that are not 0° (Figs. 3.8 and 3.9). Consequently, the mean and standard deviation of compliance for the robust solutions are significantly lower than the deterministic one (Fig. 3.7(a) and Table 3.4). Still, the Pareto optimal front considering only robust solutions shows a similar tradeoff as before (Fig. 3.7(b)), and once again the topology with higher b consists of stronger materials (Fig. 3.9). When load uncertainty is considered, however, the standard deviation estimated using UDR is not as accurate as Monte Carlo sampling, with differences higher than 10% (Table 3.4).

3.7.3. Materials and Load Uncertainties

The final case demonstrates the proposed method's ability to capture multiple sources of uncertainty by simultaneously considering material and load uncertainties. This example

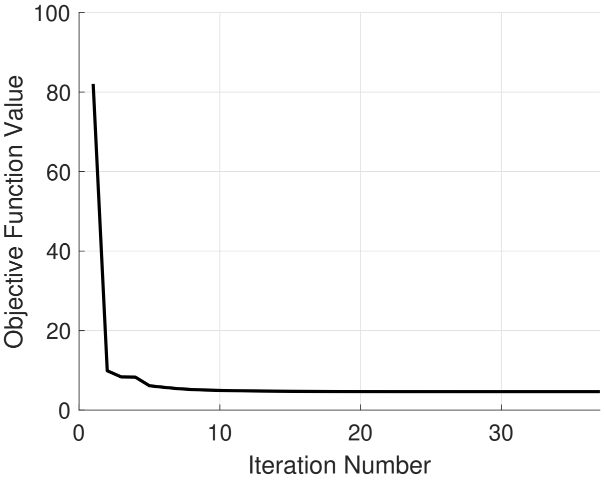
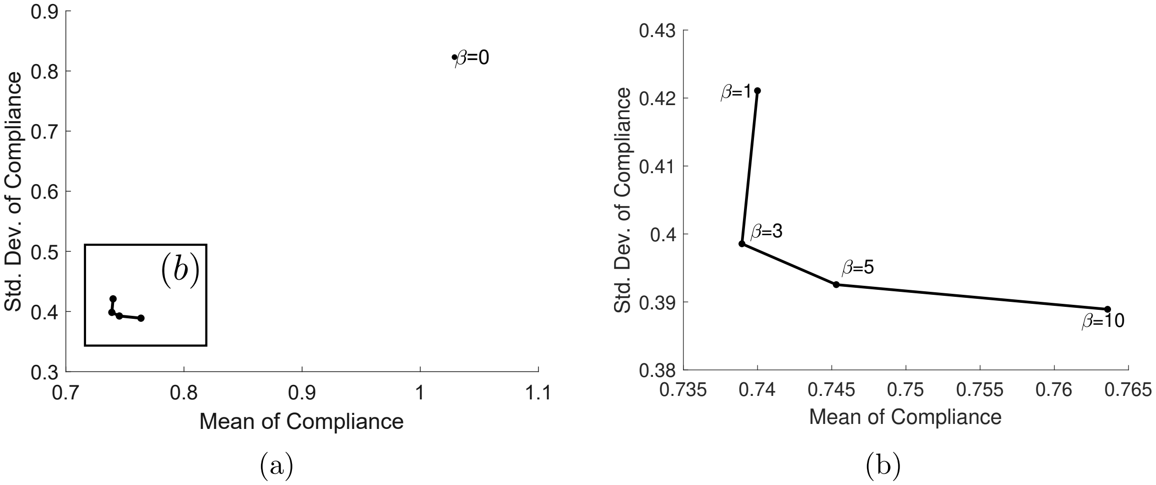


Figure 3.6. Optimization history of the objective function under load uncertainty when $\beta = 10$.



(a)

(b)

Figure 3.7. Pareto front for different objective function weights under load uncertainty (a) including $\beta = 0$ (deterministic), and (b) showing robust solutions only ($\beta = 1$ to $\beta = 10$).

combines the two previous ones: the same normal distributions for the three solid materials as in Sec. 3.7.1, and the same for the load magnitude and angle as in Sec. 3.7.2, are applied.

Again, the optimization when $\beta = 10$ converges well, achieving an objective value that is 5.71% of the initial in 34 iterations (Fig. 3.10). The Pareto frontier (Fig. 3.11) and

Table 3.4. Compliance ($N \cdot mm$) results under load uncertainty using UDR with 8 quadrature nodes and Monte Carlo with 10,000 samples.

Solution Type	Moment	UDR/Quadrature	Monte Carlo	Difference
Deterministic ($\beta = 0$)	Mean	1.02900	1.04167	1.22%
	Std. Dev.	0.83206	0.92341	10.87%
Robust ($\beta = 1$)	Mean	0.74000	0.74949	1.27%
	Std. Dev.	0.42108	0.49094	14.23%
Robust ($\beta = 10$)	Mean	0.76359	0.77067	0.92%
	Std. Dev.	0.38892	0.44047	11.70%

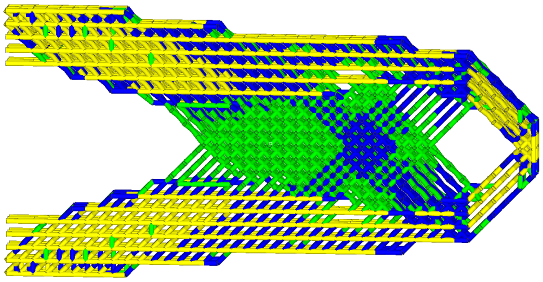
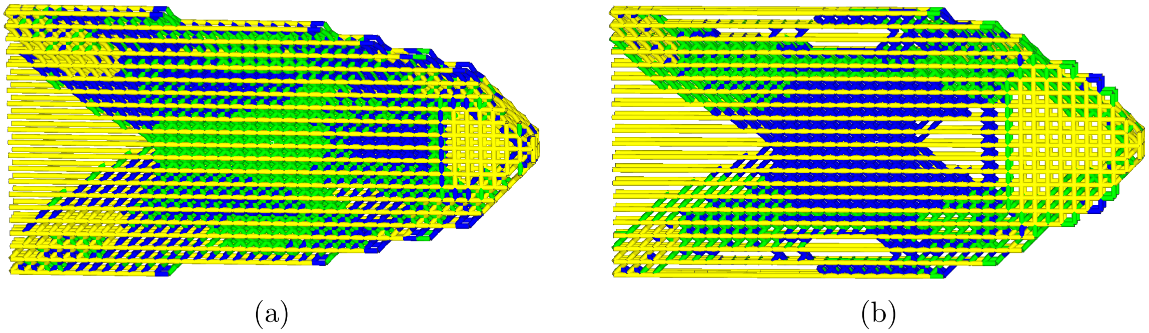


Figure 3.8. Optimal solution under load uncertainty when $\beta = 0$ (deterministic).



(a)

(b)

Figure 3.9. Optimal robust solutions under load uncertainty when (a) $\beta = 1$ and (b) $\beta = 10$.

topologies (Figs. 3.12 and 3.13) are akin to those when only load uncertainty is considered in that the mean and standard deviation of compliance for the robust solutions also decrease due to the formation of the hard material (yellow) bars. With the additional uncertainty in the materials, however, the compliance increases marginally and the accuracy of UDR deteriorates slightly (Table 3.5).

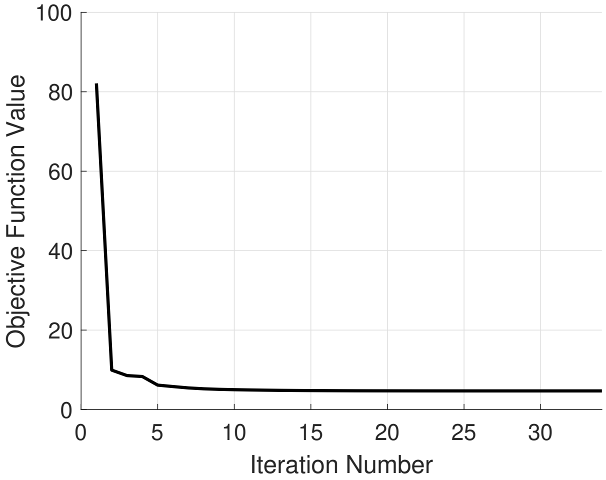
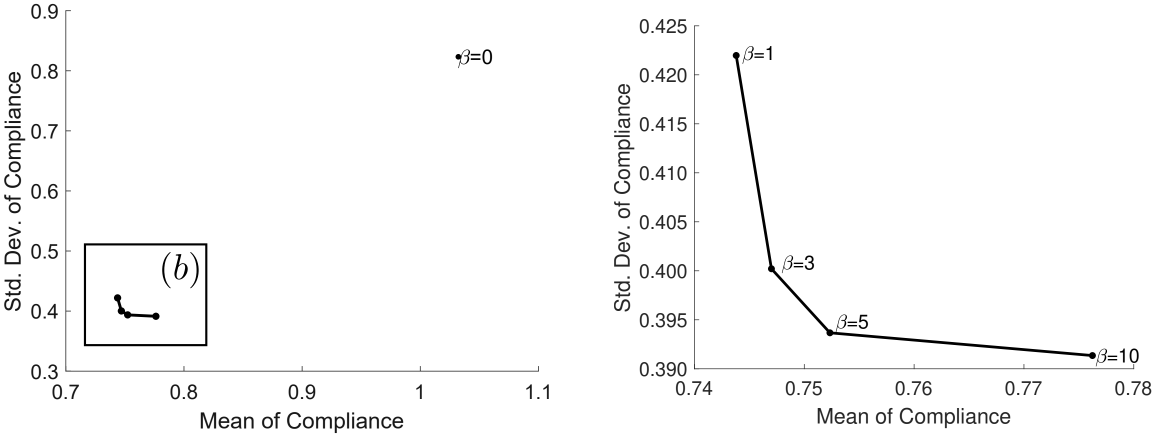


Figure 3.10. Optimization history of the objective function under both materials and load uncertainties when $\beta = 10$.



(a)

(b)

Figure 3.11. Pareto front for different objective function weights under both materials and load uncertainties (a) including $\beta = 0$ (deterministic), and (b) showing robust solutions only ($\beta = 1$ to $\beta = 10$).

Table 3.5. Compliance ($N \cdot mm$) results under material and load uncertainties using UDR with 8 quadrature nodes and Monte Carlo with $10,000$ samples.

Solution Type	Moment	UDR/Quadrature	Monte Carlo	Difference
Deterministic ($\beta = 0$)	Mean	1.03223	1.03533	0.30%
	Std. Dev.	0.82331	0.92008	10.52%
Robust ($\beta = 1$)	Mean	0.74380	0.76006	2.14%
	Std. Dev.	0.42198	0.49447	14.66%
Robust ($\beta = 10$)	Mean	0.77624	0.78349	0.93%
	Std. Dev.	0.39135	0.45541	14.07%

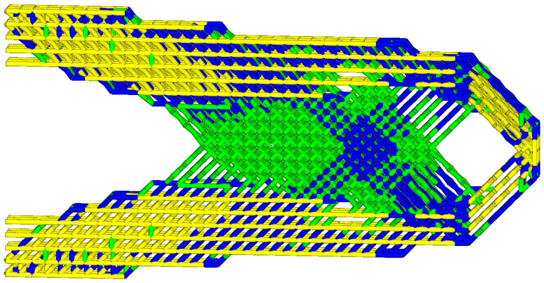
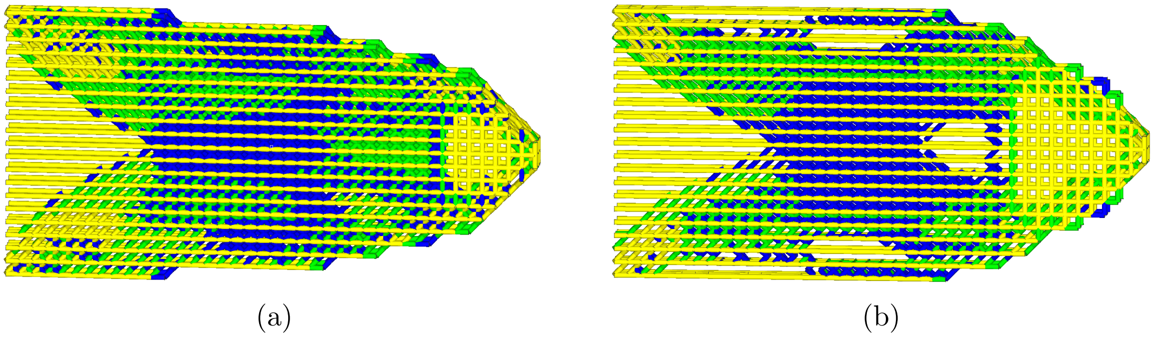


Figure 3.12. Optimal solution under both materials and load uncertainties when $\beta = 0$ (deterministic).



(a)

(b)

Figure 3.13. Optimal robust solutions under both materials and load uncertainties when (a) $\beta = 1$ and (b) $\beta = 10$.

3.8. Conclusions

The work in this chapter proposes a new generalized material interpolation scheme for an arbitrary number of materials, as well as the employment of UDR and Gauss-type quadrature to quantify and propagate any combination of uncertainty in the materials and loads. The multi-material robust TO method is demonstrated with lattice structures, although the methods can be easily transferred to multi-material continuum structures, as well as heterogeneous multiscale designs. The latter in particular has the potential for high impact. Since our multi-material interpolation scheme is not limited to only Young's moduli, but rather general enough to be applied to any continuous value, it can be applied to interpolate between the parameters of different microstructures in a heterogeneous structure.

The numerical examples of 4-phase cantilever beam lattice structures reveal that the method is efficient for large, multi-material problems with multiple sources of uncertainty. The optimization converges stably in under 40 iterations despite nearly $20,000$ elements and up to five random parameters. Using UDR and Gauss-type quadrature significantly cuts down the computational expense of calculating the statistical moments of the objective function, requiring only eight FEA evaluations per iteration rather than, for example, $10,000$ using Monte Carlo simulation. Although there is some loss of accuracy with UDR, the difference compared to the Monte Carlo approach is between 0.02% and 2.14% for the mean of compliance, and 1.13%–14.66% for the standard deviation, which is acceptable in exchange for the computational savings. Furthermore, by setting the objective function as a weighted sum of mean and standard deviation, multiple designs along the Pareto frontier can be generated and compared to observe and regulate the effect of uncertainty on the problem.

With the rise of highly functional, multi-material lattice structures built by AM, this work offers a simple solution to efficiently and robustly design these complex structures. Possible extensions of this work include but are not limited to: applying objectives more complicated than compliance; assigning targeted properties to achieve functionally graded structures; incorporating microstructure design within a multiscale framework that also optimizes the global geometry; and optimizing multi-material conformal lattice-skin structures.

Part 2

Diverse Metamaterials Datasets for Scalable Data-Driven Design

CHAPTER 4

Creating Multiclass Metamaterials Datasets for Mechanical Design

To examine the current state-of-the-art and motivate why attention needs to be placed on the data acquisition and representation step of data-driven design frameworks, this chapter first reviews recent data generation methods in the metamaterials and multiscale structural design field. Then, we illustrate the pitfalls of these methods by demonstrating two common data gathering pipelines: 1) property-driven sampling using inverse topology optimization (TO), and 2) shape-driven sampling by taking advantage of parameterized geometries or functions. The two approaches are applied to two different unit cell design representations, i.e., geometry classes – trusses and isosurfaces. The intention is that these datasets could later be leveraged in a design framework that integrates multiple classes.

4.1. Literature Review

When generating unit cells data, research has primarily been focused on a *property-driven approach* combining conventional TO with sampling strategies and parallel computing to efficiently populate large (e.g., greater than $10,000$ unit cells) databases that cover as broad a range of properties as possible. In most cases, the TO is density-based with the objective of minimizing the error between a unit cell's actual properties and the targets, e.g., the elasticity tensor, Young's modulus, or Poisson's ratio. Schumacher et al.

generated representative unit cells using TO, then converted pixels to continuous signed distance fields to smoothly generate new unit cells and create families of unit cells [34]. In our past work, an initial set of 2D unit cells were optimized using TO, then iteratively morphed using stochastic shape perturbation to reach $88,000$ structures [124]. Similarly, Zhu et al. built massive databases of up to $388,000$ 2D unit cells and $88,000$ 3D unit cells using both TO and random pixel/voxel flipping [66]. Rather than solving an inverse TO problem as in the works above, the target Young's moduli are converted to constraints, and level-set TO is used to maximize the effective conductivity in Ref. [125]. This formulation naturally avoids internal voids, which are undesirable during manufacturing. The traditional TO problems of minimizing mass [126] or maximizing stiffness [127] can also be applied to build libraries of unit cells that are optimal for a specified range of applied stresses. Despite the ability to cover a large property space, starting from TO can be burdensome, especially in 3D. Additionally, optimal solutions may not be found for all of the prescribed targets (e.g., only 36% of the targets were successfully met in Ref. [128], and 60% in Ref. [125]), wasting computational resources.

Under the *shape-driven* umbrella, a possible brute-force alternative is to simply enumerate all possible $0/1$ pixel combinations in a unit cell, such as in Ref. [129], but this is feasible only for small unit cells. Instead, Panetta et al. parameterize lattice unit cells using nodes and connecting edges, and develop several heuristic rules to generate and search through up to $16,221$ connected unit cells [65]. In a related approach, the authors of Ref. [130] exploited crystallographic cubic symmetry operations to quickly build a lattice dataset from a minimal number of initial bars and nodes; the downside is that they only generated 144 structures, though the method could be extended to create more. However, these methods are specific to lattice structures, and cannot be applied to metamaterials with other representations.

Another challenge is that the substantial size of most of these databases, particularly for unit cells that require fine resolutions to preserve complicated features, can consume huge amounts of memory, e.g., trillions of voxels in Refs. [66, 131], and be inefficient to query. Reducing high-dimensional unit cell representations before design exploration and optimization is therefore a crucial step. To this end, the authors of Ref. [131] fitted parameters such as beam thickness and orientation to $15,000$ voxelated unit cells. The parameters were then compressed even further using principal component regression, and the principal directions were utilized to navigate the property space. Alternatively, in our previous work, we reduced the dimension of the unit cells with spectral shape descriptors based on the Laplace-Beltrami (LB) operator, a low dimensional representation popularized by the computer vision community for shape retrieval and analysis. By solving a general eigenvalue problem, a vector known as the LB spectrum, or Shape-DNA, was obtained to characterize a wide variety of freeform shapes [124]. We have also used large datasets and generative deep learning methods, like variational autoencoder (VAE), to learn abstract, low-dimensional latent representations of unit cells [1].

Instead of reducing the dimension of the data, Ref. [132] compressed the size of their database by selecting a representative subset based on material properties, which allowed them to more efficiently fit a data-driven property prediction model. Another approach is to parameterize the property space: The authors in Ref. [66] and Ref. [133] fit function approximations of the database's property space using level-sets and the method of moving least squares, respectively, enabling macrostructure optimization and unit cell mapping in terms of the properties.

It is important to note that *these databases were built on the assumption that the chosen class, i.e., geometric representation, will offer optimal choices for the pre-specified application.* We challenge this assumption and, in later chapters, explore methods to expand the design space by merging multiple classes into the same design framework. As the first step in that goal, this chapter investigates whether state-of-the-art data acquisition methods are truly suitable for data-driven multiclass design through concrete examples.

4.2. Strategies to Acquire Data with Different Representations

To pare down the complexity of creating a bounty of data, we choose two popular classes of mechanical metamaterials that display a variety of desirable properties, yet can be generated with less design variables than in freeform (e.g., voxelated) shapes. These are the truss and isosurface classes. The former is known to exhibit high strength-to-weight ratios and great heat dissipation, and is widespread in the AM community due to ease of manufacturing. The latter contains bio-inspired geometries that can be expressed as analytical functions, and is popular for energy absorption, functionally graded structures and tailorable bandgaps. For thorough reviews on these classes, see Refs. [4, 7, 33, 41, 42]. Existing computational design methods for both rely on the designer's choice to select the exact class *a priori* to multiscale design [24, 36, 50, 134–136]. They are excellent choices to study whether eliminating the disjunction in current metamaterials design by bringing multiple classes into the same design space can 1) reveal the advantages of each class, and

2) merge the beneficial features of different classes to generate creative designs (however, these research questions are explored later in Chapter 7).

One argument against this multiclass approach might be that traditional TO could produce a database of freeform shapes similar to truss or isosurface structures. To answer, previous works have already demonstrated that conventional TO suffers from many computational cons due to high dimensionality, is not easily scalable to complex physics during data generation [1, 34, 66, 124], and may have low manufacturability, hence limiting the potential of data-driven heterogeneous design in impactful, real-world applications.

To calculate the linear elastic effective properties of each unit cell of both classes, we use classical asymptotic homogenization [84] with a constituent material commonly used in computational methods, Young's modulus $E = 200$ MPa and Poisson's ratio $\nu = 0.3$.

4.2.1. Property-Driven Generation of Trusses

Truss, or lattice, structures are represented as "bars-and-nodes": a list of the (x, y, z) coordinates of the nodes, as well as the connections between nodes, and the thickness of the bars. In a ground structure-based approach (Chapter 2.2.1), an initial topology containing many bars is defined. Each bar can be taken away to form a new "child" structure. By assigning a binary 0/1 design variable to each bar, a shape-driven sampling could take all possible combinations of the bars to form a large database. However, to offer a wider variety of child structures, the ground structure needs to contain many initial bars, which also leads to a large number of combinations as well as a high possibility of obtaining infeasible structures with disconnected members. Naively enumerating over all possibilities would be inefficient. In this section, a property-driven, inverse TO approach such as in Refs. [34, 66, 124] is used instead. Its pros and cons are then discussed.

4.2.1.1. Inverse Optimization via Genetic Algorithm. Many existing data-driven works like the ones reviewed above generate large databases that cover a wide range of desired properties via TO. Following this, we generate a truss structure database using discrete genetic algorithm (GA), which evolves a population of candidate children at each iteration until the fitness function is minimized. Different from the existing works, all children created by GA are saved into the database without regard for their optimality. This is akin to the approaches in Refs. [66, 124] that use inverse TO to generate an initial dataset, then use stochastic perturbation to expand the data.

Orthotropic symmetry is used to reduce the design space to only a quarter of the full cubic unit cell. Because of the symmetry, there are 12 independent linear elastic coefficients, $E_1, E_2, E_3, \nu_{12}, \nu_{13}, \nu_{23}, \nu_{21}, \nu_{31}, \nu_{32}, G_{12}, G_{23}, G_{31}$, where E_i are Young's moduli, ν_{ij} are Poisson's ratios, and G_{ij} are shear moduli. For simplicity, only 6 are taken as properties of interest: $E_1, E_2, E_3, \nu_{12}, \nu_{13}, \nu_{23}$. Optimal Latin hypercube sampling (OLHS) [137] is used to optimally select a space-filling set of 70 targets, p^*, in the 6-D property space over the range $E_i \in [10^{(} - 3), E_i^g s], \nu_{ij} \in [-1, 1]$, where E_i^{gs} is the effective Young's modulus of the ground structure in the i-th direction. Since the ground structure contains the maximum number of bars, its modulus value is the theoretical upper bound.

An inverse problem is solved to minimize the Frobenius norm of the difference between the target properties and the actual properties of the unit cell, acting similarly to a least-squares multi-objective function [65]:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} && \frac{1}{2}\|p^* - p^H(\mathbf{x})\|_F^2 \\ &\text{subject to} && \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}, \\ & && \mathbf{x} = x_i \in \{0,1\}, \quad i = 1, 2, \ldots, N, \end{aligned} \tag{4.1}$$

where p^* is the 6-D vector of target properties, $p^H(\mathbf{x})$ is the corresponding vector of the homogenized properties of the unit cell, $\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}$ is the linear elasticity constraint, and $\mathbf{x}$ is the vector of N binary design variables.

After reassembling the full structure by reflecting across the three symmetry planes, numerical homogenization is used to calculate the effective properties. Then, following Refs. [66, 124], 30 new targets are selected in the normal directions of uniform samples on the boundary of the initial property space. This process is visualized in a 3D space in Fig. 4.1.

Two ground structures with different topologies are chosen based on our user experience in an attempt to diversify the database (Fig. 4.2). The first has a truncated cube shape with 13 design variables (bars), and the second is a combination of the classical FCC and BCC trusses with 28 design variables. After 100 total targets were optimized for each ground structure, any structures with disconnected features were removed for a final count of $9,392$ structures. The $[E_1, E_2, \nu_{12}]$ property space of the truss database is shown in Fig. 4.3.

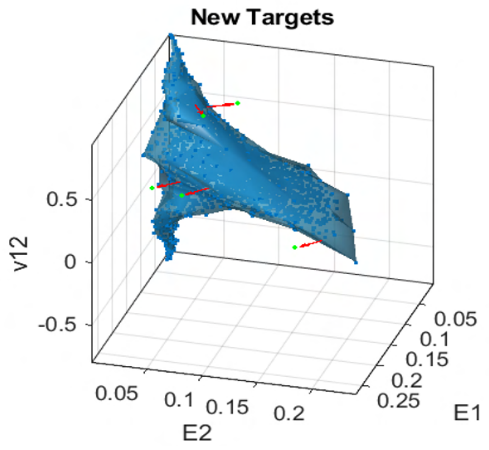
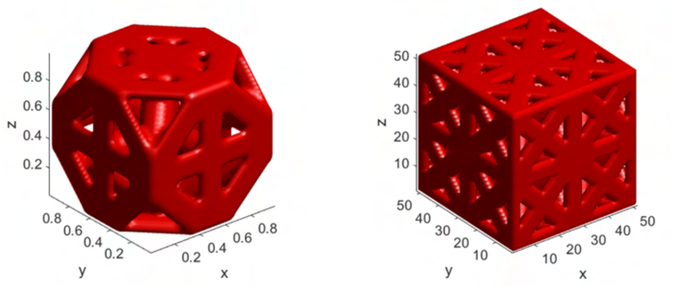


Figure 4.1. Example of sampling on the boundary of an existing property space (blue) to obtain new targets (green dots) in the normal direction (red vectors).



(a) Truncated cube (latTrunCube)

(b) FCC + BCC (latFccBcc)

Figure 4.2. Two ground structures used to generate the truss database.

Despite the ability of GA to rapidly populate the database through the candidate children, it generates many similar structures. This can be seen by the striated grouping of data samples in the property space (Fig. 4.3), and in the examples in Fig. 4.4. In addition, there are still many empty areas in the property space, although the possible properties could be limited by the chosen ground structure. Indeed, this highlights a critical but heretofore overlooked problem in the data acquisition of unit cells: *datasets* *are highly dependent on the heuristics and user bias used to create them.* These shortcomings can be seen in other databases reviewed above, yet the impact of these biased data distributions on the downstream data-driven tasks has not been directly addressed. Rather, the common solution is to sidestep the issue by *continuing to generate as much data as possible until the property no longer expands and the theoretical bounds have been approximated.* An efficient and general method is needed to acquire or compress datasets such that they have high diversity in shapes and properties.

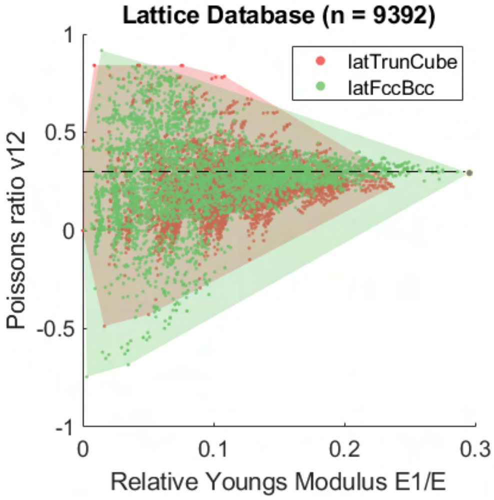


Figure 4.3. The $E_1 - \nu_{12}$ property space of the truss structure database created via GA from two ground structures.

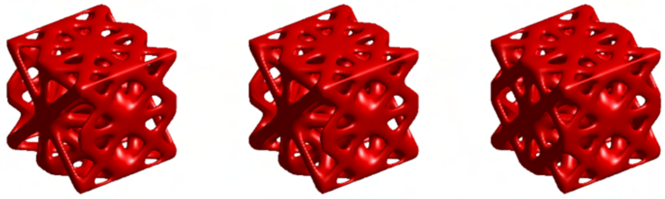


Figure 4.4. Three examples of visually similar FCC + BCC structures found by GA.

4.2.2. Shape-Driven Collection of Isosurfaces

We now investigate the shape-driven sampling approach by collecting a set of isosurfaces. Many structures found in nature, such as block co-polymers and molecules in lipid systems [138], can be described mathematically using minimal surfaces [139]. The classical triply periodic minimal surfaces (TPMS), e.g., Diamond, Primitive and Gyroid, are commonly used to design functionally graded structures in the TO field [22, 36, 50, 134].

The classical TPMS and, in general, isosurface structures can be generated from level set functions, where each coordinate (x, y, z) within the design domain of the unit cell, D, informs whether that point is on the surface of the structure, Γ, inside the solid region, Ω, or in a void region. This is expressed as

$$\begin{cases} f(X,Y,Z) - t < 0 & (X,Y,Z) \in \Omega \\ f(X,Y,Z) - t = 0 & (X,Y,Z) \in \Gamma \\ f(X,Y,Z) - t > 0 & (X,Y,Z) \in D/(\Omega \cup \Gamma) \end{cases}, \tag{4.2}$$

where $X = 2\pi x, Y = 2\pi y, Z = 2\pi z$, and t is the isovalue that controls where the surface is defined. Thus, the solid isosurface structure is

$$f(X,Y,Z) \leq t \tag{4.3}$$

and varying t is equivalent to changing the volume fraction of the structure (Fig. 4.5).

4.2.2.1. Generation via Known Surface Functions. In Ref. [140], von Schnering and Nesper proposed that periodic isosurfaces – specifically, TPMS – can be approximated as

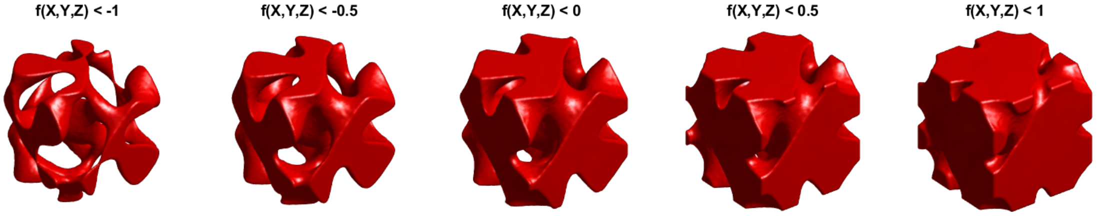


Figure 4.5. Varying the volume fraction of an isosurface structure by setting $t \in [-1, 1]$.

periodic Fourier series of the form:

$$\sum_{hkl} \|F_{(hkl)}\| \cdot \cos\left((2\pi(hx + ky + lz) - \alpha_{(hkl)}\right) = 0, \tag{4.4}$$

where (hkl) are the Miller indices of the reciprocal lattice plane, $\alpha_{(hkl)}$ is the phase shift, and $\|F_{(hkl)}\|$ is the amplitude of the *structure factor*, which, as introduced above, describes the diffraction pattern of incident X-ray beams on a crystal structure [140, 141].

In total, 36 triply periodic level surface functions from Refs. [36, 140, 142] and one quasi-periodic from Ref. [38] can be collected from existing literature to constitute an isosurface database. For each function, a family of unit cells is formed by varying t, which changes both the density and mechanical properties continuously without relying on TO (see the smooth curves formed by most of the families in the property space in Fig. 4.6). The additional parameters in the quasi-periodic function ("qpCallanan"), which are chosen via OLHS over the range $a, b, c \in [-5, 5], \alpha \in \{0, 1, 2, 3\}$, are capable of creating a variety of shapes (Fig. 4.6).

On the other hand, only a few feasible structures can be obtained from the periodic functions, which have only one parameter each – not enough to generate a large database. Moreover, just like the truss database, there can be overlaps in the shapes of the generated structures (Fig. 4.7). Although all of these provide convenient expressions to generate a dataset, they are based on geometrical descriptions only and therefore do not guarantee wide coverage of the property space (Fig. 4.6).

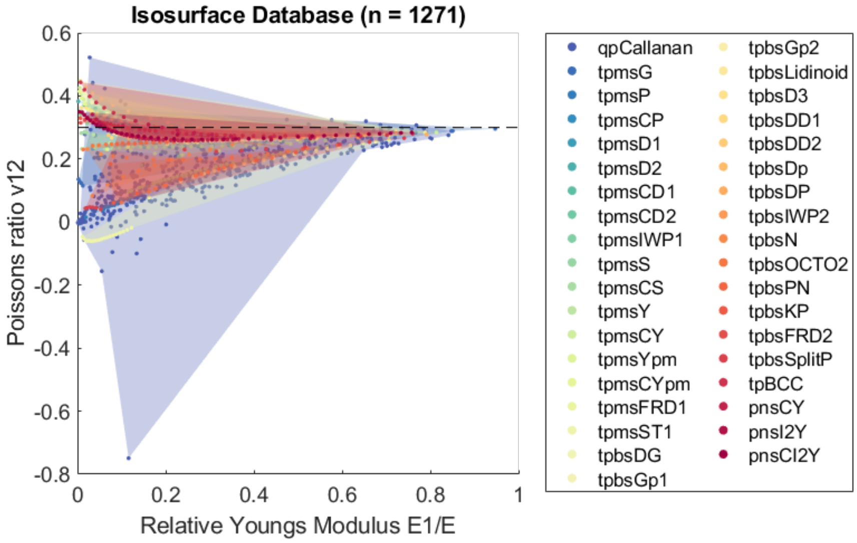


Figure 4.6. The $E_1 - \nu_{12}$ property space of the isosurface database generated by 36 existing periodic and 1 quasi-periodic surface functions.

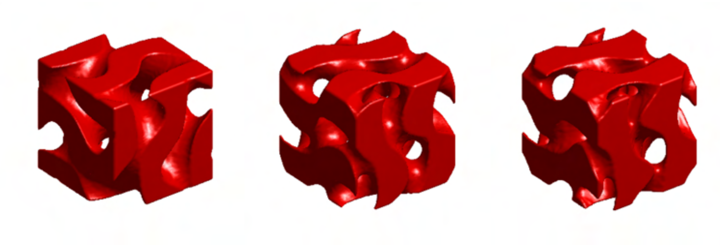


Figure 4.7. Examples from different periodic isosurface functions with similar topology (tpbsDG, tpbsGp1, pnsCI2Y, respectively).

4.3. Imbalanced Datasets and Their Implications for Data-Driven Design

Consider, then, one of the goals of this thesis: to combine all trusses and isosurfaces into a single, multiclass dataset so that they can be synthesized under the same framework.

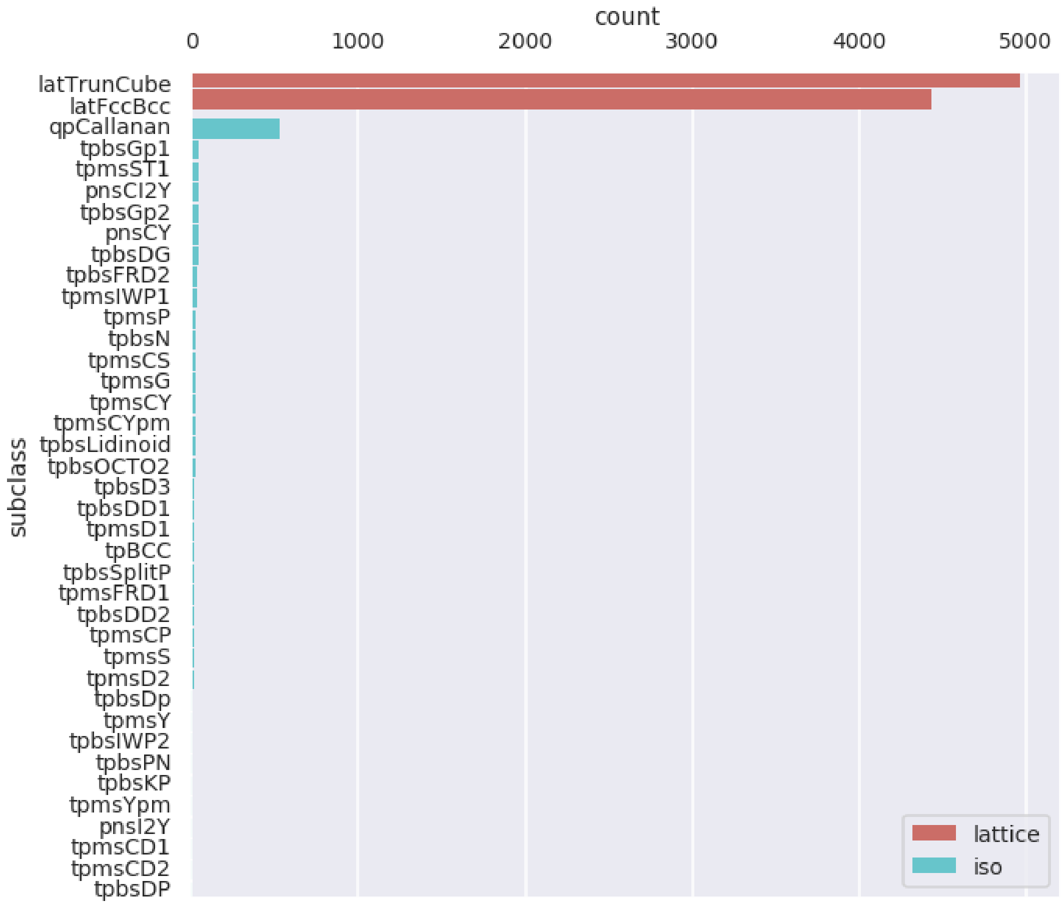


Figure 4.8. Class distribution of the example multiclass database. Here ‘lattice’ refers to truss structures generated by GA, and ‘iso’ refers to isosurface structures generated by known periodic surface functions.

From Fig. 4.8, the discrepancy of the number of samples between the truss and isosurface classes is clear. If we use this combined dataset for the next steps in data-driven design, i.e., mechanistic learning and design synthesis, it is perceivable that the biases and low diversity may affect the accuracy of any predictive models we train, and could lower the chances of finding optimal designs due to lack of valuable data in sparse regions or too many infeasible, redundant data.

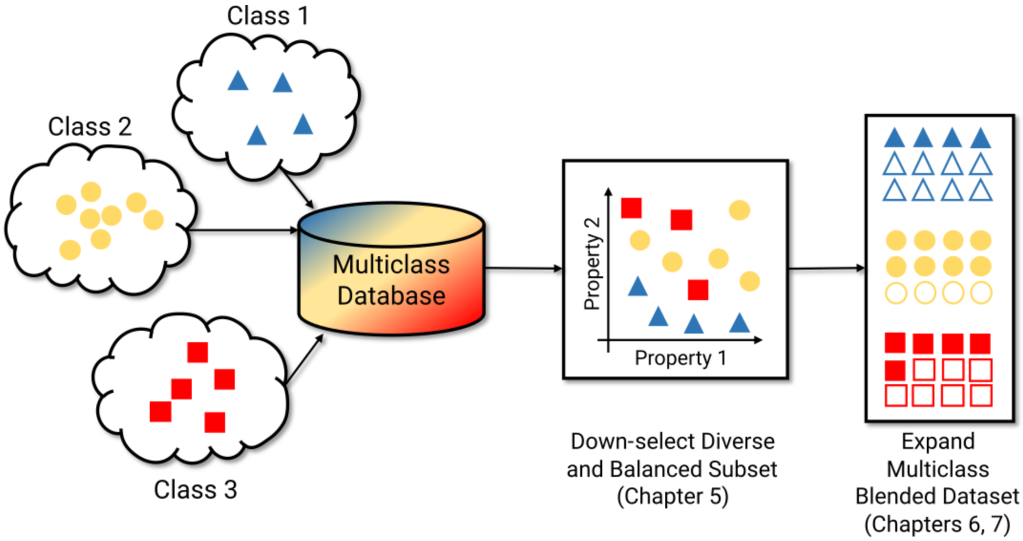

Class 1
Class 2
Class 3
Multiclass Database
Property 2
Property 1
Down-select Diverse and Balanced Subset (Chapter 5)
Expand Multiclass Blended Dataset (Chapters 6, 7)

Figure 4.9. To fill the current gap in data generation for design, methods are needed to pool data from different geometric classes to create a diverse and balanced dataset, then, optionally, to efficiently expand the dataset using methods that can simultaneously handle multiple classes. The shapes with solid colors are existing data, and those with colored borders/white interiors are new data created to expand the datasets.

Although this was a contrived, and perhaps extreme, example, *class imbalance is a critical concern in design scenarios where data is sourced from distinct shape representations, such as in our goal, or when data is collected from multiple sources (e.g., MaterialsMine [78]), or multiple physics domains.* Such imbalance can impede the performance of data-driven models since during training they may not see enough of the smaller class to learn it as well as the more abundant class [79, 143]. Moreover, the redundant shapes can also lead to problems during both model training and the design search process, and diminish the creativity (i.e., novelty) of generated designs. The state of this example multiclass database exemplifies a diversity and quality problem that may occur in data-driven design.

To deal with these inconsistencies, it is necessary to develop a general sampling method that can create balanced, diverse and valuable metamaterials datasets for data-driven design (Fig. 4.9). The hypothesis is that data-driven design methods can more efficiently discover novel and high performance designs by using balanced and diverse training datasets. The next task (Chapter 5) addresses this challenge through a methodology that leverages both shape and property diversity metrics to select smaller yet diverse subsets of metamaterial unit cells. With this method, we provide evidence of the impact that imbalance can have on the scalability and designed performance of multiscale structures by comparing the use of imbalanced datasets against diverse subsets in data-driven design methods.

CHAPTER 5

Shape and Property Diverse Subset Selection for Scalable Data-Driven Metamaterials Design

5.1. Introduction

Metamaterials are drawing increased attention for their ability to achieve a variety of non-intuitive properties that stem from their intentionally hierarchical structures [34]. While they traditionally consist of one unit cell that is repeated everywhere, multiple unit cells (or *microstructures* in this context), can also be assembled to create *aperiodic* mechanical metamaterials with, e.g., spatially-varying or functionally gradient properties [34, 36]. Over the past few years, conventional computational methods have been adapted to design such complex multiscale structures, including topology optimization (TO) of the microstructures within a fixed macroscale structure [57, 126], and hierarchical and concurrent multiscale TO that design both the macrostructure and a pre-specified number of unique microstructures [58, 59, 115]. However, as the desire to attain even more intricate behaviors grows, so too does the complexity of the design process, which must account for the expensive physical simulations and, in aperiodic structures, the vast combinatorial design space and disconnected neighboring microstructures [34, 35].

To combat costly physical simulations and immense, often intractable, geometrical design spaces, data-driven design of mechanical metamaterials is an promising and increasingly popular method. Using a pre-computed dataset of unit cells, a multiscale structure can be quickly filled via combinatorial search algorithms, and machine learning models can be trained to accelerate the process (Fig. 5.1). However, the dependence on data induces a unique challenge: An imbalanced dataset containing more of certain shapes or physical properties can be detrimental to the efficacy of data-driven approaches. In answer, *we posit that a smaller yet diverse set of microstructures leads to scalable search over the design space and unbiased learning.* To select such subsets, we propose METASET, a methodology that 1) uses similarity metrics and positive semi-definite kernels to jointly measure the closeness of microstructures in both shape and property spaces, and 2) incorporates Determinantal Point Processes for efficient subset selection. Moreover, METASET allows the trade-off between shape and property diversity so that subsets can be tuned for various applications. Through the design of 2D metamaterials with target displacement profiles, we demonstrate that smaller, diverse subsets can indeed improve the search process as well as structural performance. By eliminating inherent overlaps in a dataset of 3D unit cells created with symmetry rules, we also illustrate that our flexible method can distill unique subsets regardless of the metric employed. Our diverse subsets are provided publicly for use by any designer.[1]

5.2. Literature Review

Multiscale topology optimization (TO) methods have been used for the design of heterogeneous structural systems, but the scope has been limited thus far. The simplest methods assume a periodic unit cells within the macrostructure, which allows efficient optimization at the cost of sub-optimal designs [57, 144]. For higher design freedom, other multiscale TO methods divide a macro-structure into several subregions of periodic

[1] `https://github.com/lychan110/metaset`

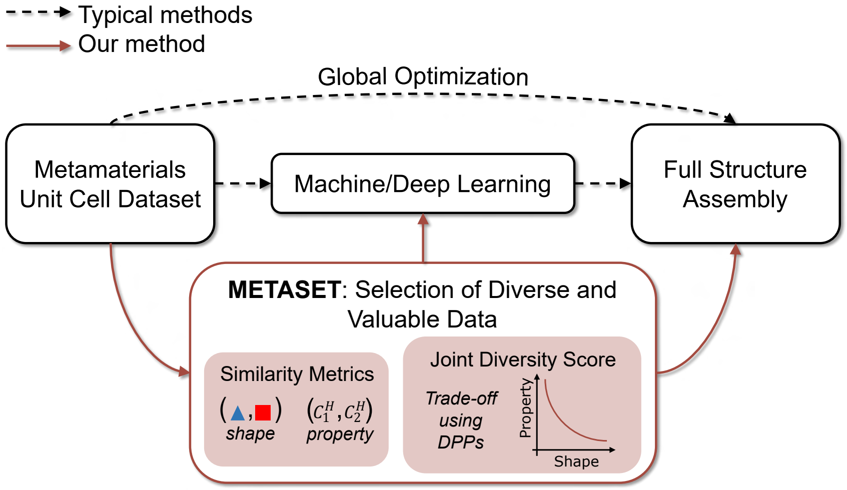


Figure 5.1. A high-level overview of data-driven metamaterials design, and how our proposed method, METASET, fits in. As an example, we show C^H, the homogenized elastic tensor, as the unit cell properties.

designs [59, 115] or assume fully aperiodic microstructure designs [80, 145]. However, existing gradient-based TO techniques have several fundamental limitations that impede their capability of scaling up to more realistic, larger scale designs. These underlying challenges include: (a) "curse of dimensionality" induced from the design space covering multiple scales, (b) combinatorial search associated with high dimensional geometric design, (c) tradeoffs between the higher accuracy of full-scale, physics-based simulations and lower cost of homogenization, and (d) difficulties ensuring manufacturability (e.g., ensuring fabrication constraints are met and/or that neighboring microstructures are connected).

Capitalizing on advances in computing power, data-driven metamaterials design can be a more efficient and therefore enticing solution to the computational challenges above. Its success hinges on pre-computed unit cell libraries or datasets, which can avoid costly on-the-fly physical simulations and multiscale TO in huge design spaces, as well as provide candidate unit cells that are better connected to their neighbors. Fig. 5.1 shows an overview of two common approaches in data-driven design: global optimization methods, and machine learning (ML) based methods. In the first case, combinatorial optimization algorithms can be used to directly search for the set of unit cells that realize a target macroscale behavior while minimizing or constraining the boundary mismatch between neighboring cells [34, 35, 66]. From another perspective, data-driven methods can use the dataset to train ML models that further accelerate design. For example, they have been used to rapidly predict homogenized physical properties as part of the optimization loop [71, 124, 128, 132]. Additionally, deep generative models inspired by the computer vision field (e.g., generative adversarial networks [77] and variational autoencoders [1]) can learn embedded geometric descriptors that act as reduced dimensional design variables, and construct new designs such as optical 2D metamaterials [68, 146] almost instantaneously. Accelerated by data-driven techniques, challenging designs such as spatially-varying displacement profiles and nonlinear behavior that are prohibitively expensive via conventional methods are now tangible.

5.2.1. "Top-Down" Data-Driven Design Frameworks

Through the homogenization-based approach (Chapter 2.2.2), multiscale design methods can achieve efficiency near that of single-scale methods. A pre-computed dataset further expedites heterogeneous design by allowing us to fully explore the unit cells "off-line". That is, we can utilize combinatorial search algorithms to assemble an aperiodic structure from unit cell designs in the dataset without having to perform homogenization on-the-fly. In this chapter, we focus on a particular data-driven approach: a “top-down” design framework that first designs the optimal properties of each microstructure, then tiles the matching unit cells. It is briefly reviewed below.

5.2.1.1. Stage 1: Identification of Target Properties. Assuming that a large database of unit cells has been obtained (using, for example, the techniques summarized in Chapter 4), the first stage of “top-down” multiscale design is to identify the optimal properties of the spatially varying microstructures. In general, these methods can be distinguished by whether the macrostructure topology is fixed *a priori* or allowed to change. For both, employing traditional optimization at the macroscale is often used as a systematic way to find the target property distribution. In existing data-driven works, common optimization objectives for the macroscale structure are to minimize compliance, strains or mass [38, 66, 126], or to achieve target deformation profiles [1, 30, 34, 37, 124].

For fixed topology, a straightforward method is performing combinatorial optimization, trying possible microstructures until the macroscopic objective and connectivity constraints are met [35, 38]. However, this does not scale well to large or complex problems. Choi et al. take the opposite approach and do not perform macroscale optimization at all. For a given topology and its boundary conditions, they simply use the von Mises stress field from one FEA solve as the target distribution [126]. Since their microstructures were each pre-optimized for specific stress cases, the macrostructure tiled with those designs attains a weight, maximum displacement and maximum stress comparable to a conventional TO result, but in a fraction of the time.

Problems with target deformation or displacements typically also fall in the fixed topology category. The Stage 1 macroscale problem can be formulated as minimizing the error between the target and actual displacement fields, with the material properties of each microstructure as the design variables [1, 65, 66]. Although these designs experience large deformations, data-driven methods thus far have ignored the geometric non-linearity in the interest of computational cost. For more accuracy, however, the non-linearity should be considered; this is listed as a "future work" in many papers.

Parameterizing the design variables as the desired properties, e.g., Young's modulus and Poisson's ratio, is also applicable to cases where the macroscale topology is allowed to change. Since it offers an intuitive and direct relation to microstructures characterized by their effective properties, this tactic is used by several works [65, 66, 133]. It is similar to the conventional single-scale TO method, Free Material Optimization, which uses the elasticity tensor as design variables and can therefore also find the property distribution of the microstructures [147, 148].

5.2.1.2. Stage 2: Structure Assembly and Connectivity. After obtaining the target property distribution, the unit cells corresponding to the targets need to be selected and tiled to assemble the macrostructure. The simplest way is to choose the unit cell that matches the targets most closely without worrying about the connectivity. In these cases, the boundary compatibility can be ensured by adding layer around regions of different microstructures [126, 149], or by averaging the node locations at the boundaries for lattice structures [65].

However, an appeal of data-driven methods is that, with a large-enough database, several candidate unit cells can be chosen based on the distance of their properties to the target ones. From these candidates, the optimal set that minimizes the boundary mismatches (e.g., the percentage of pixels/voxels that are different between neighboring microstructures) can then be chosen. This problem has been solved with combinatorial methods [38, 66], a Markov random field graph-based approach [1], and a greedy algorithm using tensor-based error diffusion [129]. For greater efficiency, the NP-hard problem was approximated by the alternating direction method of multipliers in Ref. [34]. In addition, an additional physics-based boundary dissimilarity measure was implemented using the force [34] and stress [1] discrepancies across microstructure boundaries.

In some works, if a unit cell meeting the targets or boundary constraints does not exist, a new one will be directly optimized using the same design representation as was used to generate the database [38, 65], interpolated from existing structures [34, 125, 128], or generated from a deep-learned latent space [1]. This, of course, assumes that the new unit cell is from the same class as those in the dataset.

5.2.2. Challenges and Gaps in Data-Driven Multiscale Design

Although the state-of-the-art shows promise, a challenge is that the efficacy of data-driven methods relies highly on the size and coverage of the datasets. The search space of global optimization methods can quickly explode when the number of unit cells increases. Meanwhile, imbalanced datasets with skewed data distributions can reduce the chance of meeting certain property or compatibility requirements, and hobble the performance of ML models since they may not learn a less frequent property or shape as well [79]. To our best knowledge, these risks have rarely been addressed in the general data-driven design field, much less for metamaterials design. Therefore, due to the importance of the data on downstream design tasks, in this chapter we focus on the first step of data-driven design: dataset selection.

As Chapter 4 illustrated, existing metamaterial datasets are often built using heuristics or the designer's intuition, with the assumption that the unit cells will offer sufficient coverage for the desired application. Many employ TO to inversely design unit cells that meet pre-specified target properties [34, 66, 124], and some expand the dataset by morphing the shapes [34, 124] or randomly flipping pixels or voxels [66]. Alternatively, Panetta et al. developed graph-based rules to create truss-like unit cells [65]. Although these are more feasible than enumerating over all possibilities, bias toward particular properties or shapes can be unintentionally introduced, deteriorating the performance of the design algorithm or the design itself.

Moreover, the point at which to stop generating new unit cells has thus far been heuristic with the same goal in mind: to cover a broad property space. The range of this space is sometimes restricted for specific applications [126], or strict symmetry and manufacturability constraints are implemented to limit the possible shapes [65]. More often, the property space is allowed to grow at will, e.g., TO and shape perturbation are repeated until the change in the density of the property space is less than a given tolerance [66, 124]. While efficient, all of the works to date have only considered coverage in the property space alone, which can produce similar shapes or overlook those that might benefit the design with regards to boundary connectivity. In contrast, our work explores coverage in both property and shape spaces.

Contrary to the dearth in the metamaterials field, improving imbalance arising from data with multiple classes has been extensively researched in computer science. The most relevant to our application are the data preprocessing strategies such as undersampling to remove data from majority classes, oversampling to replicate data from minority classes, or combinations thereof [79, 143]. However, the former can accidentally remove samples with important features, i.e., decrease the diversity, and the latter can lead to model overfitting and increased training overhead [150]. Nor are they made to consider the diversity of data with features that have drastically different representations, like shape and property. The issue of downsampling a metamaterial database was addressed by Chen et al. [132], who compressed the size of their database by selecting the samples that are farthest from each other with respect to properties (not shape), allowing them to more efficiently fit a property prediction model. As far as we know, there is currently no method to assess or select a diverse set of unit cells that can simultaneously cover the shape and property spaces.

A more intriguing computer science example is recommender systems, which rank diverse items such as online products to match users' preferences. These are based on the concept of diminishing marginal utility [151], wherein lower ranking items bestow less additional value onto the users. In design, too, researchers have developed methods to help designers sift through large sets of ideas by ranking them. In particular, to balance diversity against quality of designs, Ahmed et al. introduced the idea of clustering items into groups for subset selection [152] by employing submodular functions that follow the property of diminishing marginal utility. Additionally, Ahmed et al. [153] showed the application of Determinantal Point Processes (DPPs) [154], which model the likelihood of selecting a subset of diverse items as the determinant of a kernel matrix, to the diverse ranking task. The latter, in particular, are elegant probabilistic models that capture the trade-off between competing ideas like quality and diversity. While the goal of maximizing the determinant is similar to the optimality criterion used in generating D-optimal designs [155] in design of experiments, DPPs are not restricted to linear kernels, and have advantages in that calculating marginals, computing certain conditional probabilities and sampling can all be done in polynomial time. This research task shows that DPPs can also be used for coverage in multiple spaces defined over the shapes and properties of unit cells.

5.3. Contributions

We propose METASET, an automated methodology that simultaneously considers the diversity of shape and property to select subsets of unit cells from existing datasets. By doing so, we can achieve scalable data-driven design of metamaterials using smaller yet diverse subsets and eliminate bias in imbalanced datasets to improve any downstream task in the data-driven framework. As a part of METASET, we introduce similarity metrics to efficiently assess the diversity of the shapes and properties of 2D and 3D metamaterials. We also propose that a weighted sum of Determinantal Point Process (DPP) kernels based on the shape and property similarities can measure and allow the maximization of the joint diversity of both spaces. For the first time in data-driven metamaterials design — to our knowledge — we reveal through 2D case studies that diverse subsets can expedite and even enhance the design performance and connectivity of aperiodic metamaterials. Finally, applying METASET to 3D unit cells, we identify diverse families of isosurface unit cells and discover that these extend beyond the ones commonly considered in the design of functionally-graded structures [36, 134].

The components of our methodology are detailed in Sec. 5.4. In our 2D case studies (Sec. 5.5), we explore the effects of diversity and subset size on 2D metamaterial designs with non-intuitive target displacement profiles. In a 3D example (Sec. 5.6), we compare the impact of different shape similarity metrics on diverse unit cell families and demonstrate that METASET can diversify datasets regardless of the chosen metric.

5.4. METASET: An Automated Algorithm for Diverse Subset Selection in Multiple Spaces

The inner workings of METASET consist of three main steps: 1) Defining similarity metrics for metamaterials that quantify the difference between pairs of 2D or 3D shapes and mechanical properties (Sec. 5.4.1); 2) Using a DPP-based submodular objective function to measure the joint coverage of a set of unit cells in shape and property spaces via pairwise similarity kernel matrices (Sec. 5.4.2); 3) Maximizing the joint diversity with an efficient greedy algorithm while allowing trade-off in the two spaces to be tuned to suit the desired application (Sec. 5.4.3). In this section, we describe these components and summarize the methodology with Algorithm 10.

5.4.1. Similarity Metrics for Metamaterials

A diverse metamaterial dataset should ideally contain unit cells that are sufficiently different, i.e., dissimilar, such that they cover the shape and property spaces. To measure the diversity of a set, then, the similarities between the shapes and properties of unit cells first need to be quantified. We do so by defining metrics independently in each space, based on the observation that a set of unit cells dissimilar in shape space is not necessarily also dissimilar in property space, and vice versa. This can be illustrated by a simple example. Say we wish to distill diverse values from x and y, which we assume to be sets of integers: $x = \{0, 1, 2, 4, 5\}$ and $y = \{0, 2, 10, 20, 10\}$. We assume that $y = x * k$, where $k = \{3, 2, 5, 5, 2\}$ is a transformation function. If we were to select three diverse values of x, i.e., the values that most cover its space, we would select $\{0, 2, 5\}$. For y, however, we would choose $\{0, 10, 20\}$ rather than $\{0, 10, 10\}$, the ones corresponding to the diverse x values. Hence, though some relationship between two spaces may exist, e.g., an intrinsic function between shape and property, there is a need to model their coverage separately. This observation is validated in our later design experiments (Sec. 5.5.2), where the correlation coefficient between shape and property coverage shows that no link exists between the two.

5.4.1.1. Property Similarity. Since mechanical properties are generally scalar values that can be expressed as a vector, e.g., by flattening the elastic tensor, we can use any similarity metric between vectors. In this work, we use the Euclidean distance. We note that the properties do not need to be the tensor components; rather, they can be other values of interest such as elastic or shear moduli, or Poisson's ratios. Neither do they need to be limited to scalar mechanical properties. For instance, dynamic acoustic dispersion curves or bandgaps could be considered if the pairwise similarity can be quantified.

5.4.1.2. Shape Similarity. Shape similarity metrics are key in many computer vision and graphics applications, e.g., facial recognition and object retrieval from databases. In these methods, the shapes are usually first represented by structural descriptors extracted from individual shapes [156], or by embedded features learned via data-driven methods such as clustering or deep learning [157, 158]. The distances between features can then be measured in Euclidean [156] or Riemmanian space [159, 160]. Since Riemannian metrics are based on geodesic distances, they are suitable if one needs invariance to deformation, i.e., if one considers a shape to be the same after bending.

For metamaterials, however, we must rule out deformation and rotation invariant metrics since any transformation of a unit cell impacts its properties. Additionally, we seek techniques that are efficient but still able to discriminate fine details and form positive semi-definite similarity matrices for the next step involving DPPs. Thus, we introduce the following Euclidean metrics based on structural features: a descriptor-based distance for 2D, and two point cloud-based metrics for 3D, namely, the Hausdorff distance and embedded cosine similarity utilizing deep learning. While we elected for separate metrics in 2D and 3D by bearing in mind their respective computational efficiencies, shape analysis is a wide and ever-growing topic of research in computer science; many other metrics are available. As we later show in Sec. 5.6.2, METASET selects diverse subsets regardless of the metric used, as long as the requirements for DPPs are met.

2D Descriptor-Based Euclidean Distance: For 2D unit cells, which are typically binary images resulting from TO, we propose using a descriptor-based approach by first extracting division-point-based descriptors [161] to reduce the images into vectors that capture salient features at different levels of granularity. This has been applied to the field of optical character recognition [162, 163]. The binary image of a unit cell is recursively divided into sub-regions that contain an equal number of solid pixels. The coordinates of all division points, i.e., points at the intersection of two division lines between each sub-region, are then obtained as descriptors of the unit cell. This process is repeated until the desired level of detail is captured, constructing a k-d tree of the distribution of solid materials. In our 2D case study (Sec. 5.5), we obtain a sufficient amount of detail by performing the division seven times for each unit cell, resulting in 62 division points that constitute a 124-dimensional shape descriptor.

Using the above method, we can represent each 2D unit cell as a vector, then use the Euclidean norm to find the distance between any pair. However, the input for a DPP is a positive semi-definite similarity matrix, L, so we transform the distance to a similarity metric through a radial basis function kernel with unit bandwidth, i.e., $L_{i,j} = \exp(-0.5\, d(i,j)^2)$, where $d(i,j)$ is the distance between i-th and j-th unit cells. In practice, the choice of an appropriate transformation is equivalent to choosing the right distance metric between items. Our empirical study on other common transformations showed that different choices mainly affect the distribution of similarity values but do not significantly affect the final outcome or the key findings of our work.

3D Hausdorff Distance: As for 3D unit cells, mesh formats such as STL are commonly used so that the metamaterials can be manufactured through additive manufacturing. However, since performing analysis on 3D shapes is undoubtedly more computationally intense due to the curse of dimensionality, we suggest representing each unit cell as points on the surface of the original mesh, i.e., point clouds, which are more efficient for extracting and processing 3D features [164]. This extra conversion can take little computation with well-established sampling methods, e.g., randomly sampling the surface of a mesh with the probability of choosing a point weighted by the area of the triangular faces.

We then use a distance metric commonly utilized to measure the distance between sets of points, the Hausdorff distance. In essence, it computes the difference between two clouds as the maximum of the nearest neighbor distances of each point. This is expressed as [165]:

$$h(A, B) = \max_{a \in A} \big[\min_{b \in B} \|\cdot\| \big], \tag{5.1}$$

where a is a point within cloud A and b is a point in the second cloud B. The notation $\|\cdot\|$ indicates that any distance can be used; for example, we can use the Euclidean norm or the cosine distance between two points. In our implementation, we computed the nearest neighbor Euclidean norms using a GPU-enabled code by Fan et al. [166]. Then, to obtain a symmetric distance, we take the maximum as follows:

$$d_{\mathrm{H}}(A, B) = d_{\mathrm{H}}(B, A) = \max \big[h(A, B), h(B, A) \big]. \tag{5.2}$$

Finally, we convert the pairwise distances into a DPP similarity kernel, L, using the following transformation: $L_{ij} = \frac{1}{1+d(i,j)}$.

3D Embedded Cosine Similarity: Alternatively, the embedded features of the unit cells in a given dataset can be extracted using deep learning models as simple as an autoencoder, a dimension reduction technique that compresses, i.e., encodes, complex shapes into vectors. Once such a model has been trained, an embedding-based shape similarity metric can be defined as the similarity between the vector representations of unit cells, much like the 2D descriptor-based distance earlier.

Here we also leverage point clouds, which are growing as a scalable and powerful representation for 3D deep learning [167]. We utilize a point cloud autoencoder provided by Achlioptas et al. [158] with the Earth Mover's distance as the reconstruction loss. Our 3D dataset (described in Sec. 5.6.1) is split into training, test and validation sets by 70%, 15%, and 15%, respectively, and a grid search is performed to decide the hyperparameters: 64-dimensional embedded vectors for each unit cell, a learning rate of 0.0005 and batch size of 32. After training the model for 120 epochs, we can then take the cosine similarity between the embedded vector representations of any two unit cells as the shape metric. In our 3D experiment (Sec. 5.6.2), we compare the diverse subsets obtained using this embedded feature approach against those using the Hausdorff distance.

5.4.2. Determinantal Point Processes for Joint Diversity in Two Spaces

With a similarity kernel matrix L, we can now measure the diversity of a dataset using Determinantal Point Processes (DPPs), which are models of the likelihood of choosing a diverse set of items. They have been used for set selection in ML, e.g., diverse pose detection and information retrieval [154, 168], and recently in ranking design ideas based on diversity and quality [153]. Viewed as joint distributions over the binary variables that indicate item selection, DPPs capture negative correlations. This means that, intuitively, the determinant of L is related to the volume that the set covers in a continuous space. In other words, the larger the determinant, the more diverse the set.

To model our data, we construct DPPs through L-ensembles [169], using a positive semi-definite matrix L to define a DPP. Hence, given the full unit cells dataset of size N, which we denote as ground set G, DPPs allow us to find the probability of selecting any possible subset M of unit cells as:

$$\mathbb{P}(M) = \frac{det(L_M)}{det(L+I)}, \tag{5.3}$$

where $L_M \equiv [L_{ij}]_{ij \in M}$ is the submatrix of L with entries indexed by elements of the subset M, and I is a $N \times N$ identity matrix. The probability of a set containing two items increases as the similarity between them decreases. Therefore, the most diverse subset of any size has the maximum likelihood $\mathbb{P}(M)$, i.e., the largest determinant. For a fixed subset size, the denominator can be ignored when maximizing the diversity via an algorithm such as the one described in Sec. 5.4.3.

Unlike submodular clustering approaches, DPPs only require the similarity kernel matrix L as an input, and do not explicitly need the data to be clustered or a function that models diversity to be defined. This also makes them more flexible, since we only need to provide a valid similarity kernel, rather than an underlying Euclidean space or clusters.

For METASET, we calculate two different similarity values — one in shape space and another in property space — between any two unit cells. Hence, for all the unit cells combined, we have one kernel matrix corresponding to each of the two spaces. In order to measure the joint coverage in both spaces, we take a weighted sum of the two matrices, thus also allowing the trade-off between diversifying in shape or property space:

$$L = (1-w) \cdot L_P + w \cdot L_S, \tag{5.4}$$

where L, L_P and L_S are, respectively, the joint, property and shape similarity kernels, and w is a weight parameter can be varied between 0 and 1. By adding the two kernels, we assume that the total similarity between two unit cells is the weighted average of how similar they are in the shape and property spaces.

While it is possible to combine two kernel matrices in many ways, we choose this formulation for two reasons. First, the weighted sum of two positive semi-definite matrices is also positive semi-definite, which is a pre-requisite for a DPP kernel. Second, it allows us to control the amount of diversity in both spaces, as well as to frame the later subset selection problem as multi-objective one, using a single tuning parameter w. We conducted multiple experiments on simulated data with easy-to-verify coverage metrics and found that this approach is effective in capturing diversity in both spaces. For brevity, we have not included these experiments here but directly report and discuss the results using joint kernels for metamaterials in Secs. 5.5.2 and 5.6.2.

5.4.3. Algorithm for Optimizing Diversity

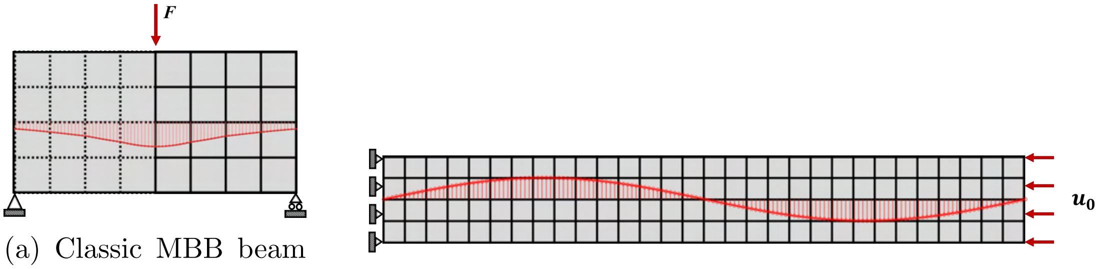
Optimizing the diversity of a subset M in two spaces is an inherently multi-objective problem that can be accomplished by maximizing the log determinant of the joint similarity kernel, i.e., $f = \log[det(L_M)]$. Note that the log determinant of a positive semi-definite matrix is monotonically non-decreasing and submodular. In general, finding the set of items that maximizes a submodular diversity function is NP-Hard. When solving such problems, a well-known limit due to Feige [170] is that any polynomial-time algorithm can only approximate the solution up to $1 - \frac{1}{e} \approx 67\%$ of the optimal.

However, this is where choosing a submodular function f as the objective comes in handy. It turns out that greedily maximizing this function is guaranteed to achieve the optimality bound [170]. We use this property to substantially accelerate diversity optimization using a scalable greedy algorithm [171], which has theoretical approximation guarantees and is widely used in practice. At each step, the algorithm picks an item, i.e., a unit cell, that provides the maximum marginal gain in the objective function (lines 5-8 in Algorithm 10 in Appendix **??**). This makes greedy maximization of diversity the best possible polynomial-time approximation to an otherwise NP-Hard problem.

5.5. METASET in Data-Driven 2D Metamaterials Design

Selecting a diverse and economical dataset prior to design can augment the performance and results of any data-driven algorithm. In this section, we demonstrate that this improvement can be achieved by adding METASET to existing data-driven frameworks with little extra cost (Fig. 5.1) by designing 2D aperiodic mechanical metamaterials that meet desired displacement profiles and constraints on the connectivity of neighboring microstructures. Given a 2D dataset of unit cells from our previous work (briefly described in Sec. 5.5.1), we use METASET to select several subsets with differing sizes and diversity scores (Sec. 5.5.2). By employing these subsets to assemble full structures, we study the effects of subset size and diversity on the search process and final designs (Sec. 5.5.3). To emphasize that our diverse selection methodology is an advantageous addendum to any data-driven method, we perform the designs with two existing approaches — genetic algorithm for an illustrative example, and a two-stage method for a more complex design motivated by practical applications (Sec. 5.5.4).

The design settings, a classic MBB beam and a cantilever, along with the boundary conditions and target displacement profiles (red curves) are shown in Fig. 5.2. The design objective for both is to minimize the mean squared error (MSE) between the target



(a) Classic MBB beam (Sec. 5.5.3)

(b) Cantilever (Sec. 5.5.4)

Figure 5.2. Problem settings of the 2D examples, both of which should achieve the target displacement profiles shown in red.

and achieved displacement profiles. These types of structures, which require spatially varying elastic behavior and therefore benefit from aperiodic configurations and data-driven methods, have been a growing focus in recent research, with applications such as soft robotic grippers and biomedical devices [30, 34]. We deliberately choose these since spatially varying properties are difficult to obtain using conventional methods, particularly when the objective is dependent on the relative spatial distribution of properties rather than an absolute performance value like compliance.

To support this claim, we attempted to benchmark the performance of a conventional TO approach based on the Solid Isotropic Material with Penalization (SIMP) scheme [43] for the MBB problem (Eq. 5.5), whose sensitivities can be derived using adjoint analysis. Each unit cell is discretized into 50 × 50 quadrilateral finite elements, and the density of each element is treated as a design variable, $\rho_e \in [0, 1]$, where the goal is to converge as close to 0 (void) or 1 (solid) as possible. To eliminate mesh dependency, a sensitivity filter with a radius of 2 is applied. For combinations of different penalty factors, $p \in \{1, 3\}$, and volume fraction constraints, $V \in \{0.50, 0.75, 1.0\}$, we minimize the MSE using the method of moving asymptotes (MMA) [122] and the same stopping criteria. All results are infeasible, however, with high MSE ranging from 4.69 to 6137.08 and numerous intermediate densities (more than 95% of the elements). This underscores the need for more advanced approaches like data-driven design, which have successfully achieved target spatially varying behavior [34, 35, 66, 124]. We will leverage two such approaches in the following sections, since the goal of this paper is not to propose new design methods but to select diverse subsets which provide salient advantages to any existing data-driven design framework.

5.5.1. Generation of 2D Unit Cells

In [124, 128], we previously proposed using a combination of TO and stochastic shape perturbation to generate a large dataset of 2D unit cells. To initialize the dataset, we ran density-based TO for each uniformly sampled target property, the components of homogenized elastic tensors (with a constitutive material with Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.3$), and then iteratively perturbed the shape of the unit cells with the most extreme or uncommon properties. By doing so, we created a dataset of 88,000 unit cells that covered a relatively large property space within reasonable computational cost. Note that we did not build this dataset with geometry in mind, leading to many similar shapes. Also, even though we aimed to fill the less populated regions of the property space by perturbing unit cells in those locations, there is a higher concentration of final unit cells with lower property values (the lower left corners in Fig. 5.4), indicating that the dataset is somewhat imbalanced. For details, please see [124].

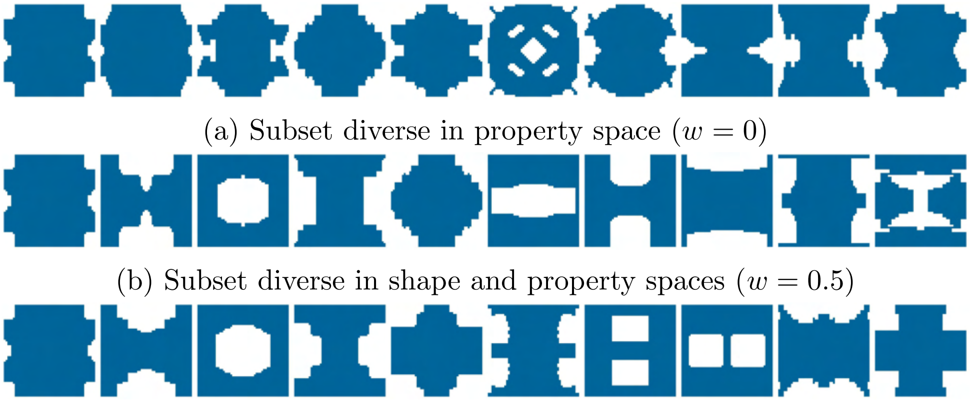
Before applying METASET, we preprocess the data by randomly sampling unit cells from the original dataset that have a volume fraction greater than 0.70, resulting in 17,380 unit cells. This fraction was chosen so that the chosen unit cells are less likely to have very thin features, which makes them more feasible for manufacturing. Additionally, when computing shape diversity, if unit cells occupy very different volume fractions, a diverse subset is more likely to be dominated by flimsy, low density structures, whose shapes have the least probability of overlap with other unit cells. However, as we will show with the design examples, this preprocessing does not impede the chances of designing well-connected structures that met the targets quite well.

5.5.2. Diverse 2D Unit Cells

For the dataset of 17,380 2D unit cells, which we now refer to as the full or ground set G, we calculate the property and shape similarity matrices, L_P and L_S, respectively, as described in Sec. 5.4.1. Taking their weighted sum forms the joint DPP kernel matrix L (Sec. 5.4.2), whose determinant, $det(L_M)$, scores the diversity in both spaces. To explore this, we rank several subsets using the greedy algorithm from Sec. 5.4.3 by varying their sizes, N_M, and kernel weights, w. From the results, we can make three observations:

(1) By increasing w, we shift from ranking a subset based on diversity in the property space alone, to a mixture of both spaces, and to the shape space only. In essence, the trade-off between shape and property diversity can be easily controlled.

(2) The correlation coefficient between the shape and property diversity scores of 1,000 random subsets of size five is 0.0047. Similar near-zero correlation is found for other set sizes too. In addition, the correlation between the shape and property similarity values of 100,000 random pairs of unit cells is -0.0024. Therefore,



(a) Subset diverse in property space ($w = 0$)

(b) Subset diverse in shape and property spaces ($w = 0.5$)

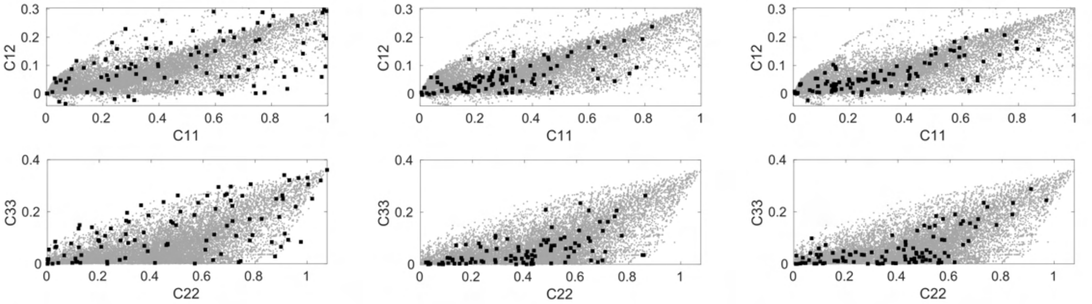
(c) Subset diverse in shape space ($w = 1$)

Figure 5.3. Examples of 2D unit cells from the diverse subsets used in the cantilever and MBB design problems.

our assumption that the joint similarity can be modeled as a weighted sum is appropriate.

(3) By observing the joint diversity score of the subsets as more items, i.e. unit cells, are added, we find that the gains in shape and property diversities saturate at approximately $N_M = 20$. Thus, a very small number of unit cells are sufficient to cover both spaces.

Ten example unit cells from the subsets with $w \in \{0, 0.5, 1\}$ are shown in Fig. 5.3, where the subset optimized for only shape diversity (Fig. 5.3c) displays the most variety of topologies compared to the subset diverse in only properties (Fig. 5.3a). Meanwhile, the balanced subset contains a mixture of unit cells akin to both extreme sets (Fig. 5.3b). This may be counter-intuitive since similar shapes should have similar mechanical properties. However, note that upon close inspection, the property diverse unit cells exhibit tiny features that lead to low effective elastic property values. Such small details in the



(a) Property diverse samples (b) Random samples (c) Shape diverse samples

Figure 5.4. The property space of the 2D unit cell subsets optimized for property and shape diversity, and a randomly sampled set, plotted against the full dataset. We observe that property diverse subsets cover the space well, hence it is more likely to have unit cells near any target property combination.

shape may lead to a larger change according to the physical simulations and the property similarity metric, i.e., the Euclidean norm.

Comparing the properties of the unit cells in diverse subsets to the ground and randomly sampled sets (Fig. 5.4), we can confirm that the property diverse subsets cover all regions of the original property space, even the sparsely populated areas. As expected, the shape diverse subset does not do as well, and the random subset contains tight clusters in certain areas. Along with the observation that the diversity scores as well as the similarity values in the shape and property spaces are essentially uncorrelated, these findings confirm that the formulation of the joint kernel L_M as a weighted linear sum (Eq. 5.4) is effective for controlling the amount of diversity in either space.

Finally, the result that only 20 unit cells is needed to cover the shape and property spaces is quite interesting since a main tenet of data-driven design thus far is that ”more is better” — larger datasets provide more candidates from which we can choose compatible unit cells. So, to explore the impact of the subset size on the data-driven approach, we selected the top 20 as well as top 100 ranking unit cells from each subset to move on to the next step: full structure assembly.

5.5.3. Illustrative Study on the Effects of Size and Diversity

We begin by designing a relatively simple classical example from the TO field, the MBB beam, such that its horizontal centerline conforms to the red curve when loaded with a vertical force F (Fig. 5.2a). Due to the structural symmetry, we only need to design the right half of the beam with 4×4 microstructures, outlined by the solid black lines. The full structure can then be obtained by reflecting over the vertical centerline. Using subsets of unit cells with varying sizes and levels of diversity for metamaterials design using global optimization, we can elucidate 1) the effect of subset size on the search algorithm's efficiency, and 2) the impact of diversity on the final design performance as well as the compatibility of neighboring microstructures. We choose the following diverse subsets using METASET:

- P_{20}: Property diverse subset of size 20
- SP_{20}: Shape and property diverse subset of size 20
- S_{20}: Shape diverse subset of size 20 diverse
- P_{100}: Property diverse subset of size 100
- SP_{100}: Shape and property diverse subset of size 100
- S_{100}: Shape diverse subset of size 100.

In addition, we utilize these sets, which are not diverse and are not selected by our method, as baselines:

- R_{20}: Random subset of size 20

- R_{100}: Random subset of size 100
- G: Full dataset of size 17,380.

To design the MBB beam, we pass each of the datasets to a global optimization method, which for this example is a single objective genetic algorithm. Although the approach is simple, we chose it to focus on illustrating the effects of subset size and diversity on the final results. It also allows us to restrict our design to the discrete choice of unit cells in our subsets, whereas most gradient-based algorithms for data-driven metamaterials design map continuous design variables to the nearest existing, or interpolated, unit cell in dense databases [34, 66].

Specifically, the genetic algorithm is used to select the combination of unit cells from each given dataset that minimizes the MSE between the achieved and target displacement profiles. In addition, since detached neighbours are not desirable, we add a compatibility constraint by requiring that the number of disconnected microstructures, N_{dc}, in the full structure be equal to zero. The optimization problem is formulated as:

$$\begin{aligned}
&\underset{\boldsymbol{l}}{\text{minimize}} && \frac{1}{n}\|\boldsymbol{u}(\boldsymbol{l}) - \boldsymbol{u}_t\|_2^2 \\
&\text{subject to} && \boldsymbol{K}(\boldsymbol{l})\boldsymbol{U} = \boldsymbol{F}, \\
& && N_{dc}(\boldsymbol{l}) = 0, \\
& && l_i \in \{1, 2, \cdots, N_M\}, \quad i = 1, 2, \ldots, N_f,
\end{aligned} \tag{5.5}$$

where $\boldsymbol{u}$ is the displacement of n nodes located on the centerline of the structure, $\boldsymbol{u}_t$ is the discretized target displacements, $\boldsymbol{K}$ is the global stiffness matrix, and $\boldsymbol{U}$ and $\boldsymbol{F}$ are global displacement and loading vectors, respectively. The number of unit cells in the

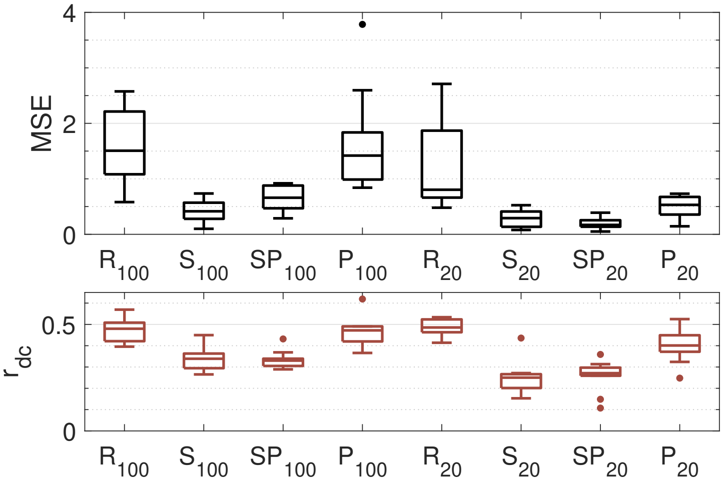


Figure 5.5. The final objective values (MSE) and ratios of disconnectivity (r_{dc}) of 10 runs per subset. Lower values are better. The best overall MSE is obtained by SP_{20} and S_{20}, and the best r_{dc} by S_{20} and SP_{20}.

Table 5.1. Means of the final results for the MBB example, with the lowest values in **bold**.

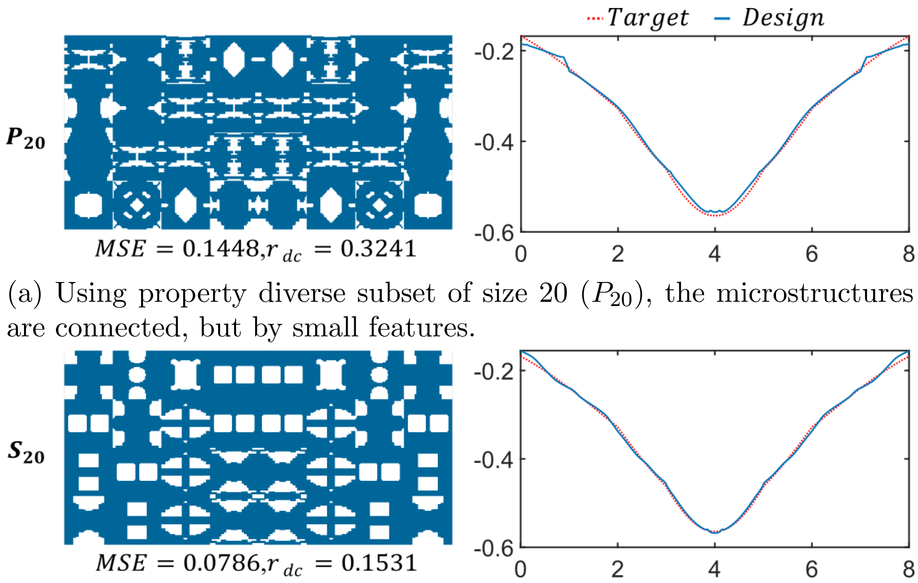
	G	R_{100}	S_{100}	SP_{100}	P_{100}	R_{20}	S_{20}	SP_{20}	P_{20}
MSE	1.3E+18	1.5341	0.4278	0.6454	1.6648	1.2395	0.2865	**0.2017**	0.4926
r_{dc}	0.5184	0.4770	0.3406	0.3347	0.4653	0.4836	**0.2488**	0.2578	0.3996

given dataset is N_M while the number in the full structure is N_f, and $\boldsymbol{l} = [l_1, l_2, \ldots, l_{N_f}]^T$ is a vector of the indices of the chosen unit cells.

Due to the stochasticity of genetic algorithms, we run the optimization ten times for each dataset and report the MSE of the final topologies in Fig. 5.5. In addition, we show a measure of the connectivity of the final structure: the mean ratio of disconnected pixels on the boundaries of touching microstructures, r_{dc}. Similar to N_{dc} in the constraint (Eq. 5.5), a fully compatible structure should have r_{dc} as zero. The averages of these results are also disclosed in Table 5.1.

When given the baseline full dataset, G, the genetic algorithm is overwhelmed and not able to find any designs with satisfactory MSE (see the high values in Table 5.1), even failing to meet the compatibility constraint in one run. This can be attributed to a vast search space since the number of possible unit cell combinations grows exponentially as the size of the dataset increases. A larger set may also contain more redundant shapes or properties that contribute little to diversity, exacerbating the search challenge and possibility of local optima. Conversely, every run using the 20- and 100-item subsets satisfy the design requirements (Fig. 5.5 and Table 5.1). These include the baseline random subsets selected without our method, which obtain reasonable performance and connectivity due to the reduced search space. The values of MSE and r_{dc} using random subsets, however, vary widely. In fact, our results highlight that *smaller yet diverse* subsets more consistently outperform all other sets under the same search algorithm and termination criteria. Notably, the lowest mean MSE is reached by the small SP_{20} and S_{20} sets. Moreover, the best connected structures, i.e., those with lowest r_{dc}, result from the diverse subsets that consider shape, i.e., S_{20} and SP_{20}. We remark that our optimization problem only constrains the number of disconnected microstructures and does not explicitly minimize r_{dc}. Therefore, the shape diverse results naturally attain higher connectivity.

Fig. 5.6 shows the final topologies and optimal displacement profiles of the runs that achieve the minimum MSE for two datasets. As expected from the worse performance and compatibility, the designs using the full dataset G (not pictured) contain disconnected and oddly matched microstructures. In a similar vein, the high r_{dc} for property diverse sets correspond to mediocre connectivity, as shown by the P_{20} result in Fig. 5.6a, where



(a) Using property diverse subset of size 20 (P_{20}), the microstructures are connected, but by small features.

(b) Using shape diverse subset of size 20 (S_{20}), we observe superior connectivity between neighboring microstructures.

Figure 5.6. Final topologies and displacement profiles of the classic MBB beam example with the lowest MSE out of 10 runs using 20-item diverse sets. The full structure after symmetry is shown.

neighbors are linked by tiny features. This can be associated with the observation in Sec. 5.5.2 that METASET tends to include unit cells with small features as it maximizes property diversity, leading to subsets with less compatible unit cells. With shape diverse subsets, however, the final designs possess excellent compatibility, such as in Fig. 5.6b, further enforcing the advantages of shape diversity.

Although our constrained genetic algorithm provides satisfactory designs, we must point out that our goal is not to introduce new design methods; this global method was implemented to showcase the impact of subset size and diversity. While more elegant optimization techniques would be better suited for practical applications, we nevertheless believe that the insights gained from this study — that selecting diverse subsets can accelerate and benefit metamaterial design — can be generalized to other data-driven methods, such as the one in the next section.

5.5.4. Additional Study with a Shape Morphing Metamaterial Structure

In the previous section, a simple example using genetic algorithm demonstrated that data-driven metamaterials design can benefit from small and diverse subsets of unit cells. To validate that this is also true for more sophisticated algorithms and designs, we now test the same hypothesis by combining our diverse subsets with an advanced optimization method we proposed in [124], which is described briefly below. Here we design a cantilever composed of 4 × 30 microstructures to achieve a sine-wave shape when a prescribed displacement boundary condition is imposed (Fig. 5.2b). As opposed to the MBB beam, the spatially varying behavior of the cantilever is designed to deform in opposite directions in the left and right halves, and we expect that different regions in the structure will require distinctly contrasting properties. The prescribed boundary instead of a point load poses an additional challenge. The closest problem to this that has been addressed by traditional TO methods is the compliant mechanism design, which aims to control the ratios between output and input displacements or forces by minimizing the displacement at a particular node. To obtain feasible mechanism designs, however, Deepak et al. found in [172] that it is necessary to assume a force-displacement relationship, i.e., a spring, at that output node. In contrast, our problem minimizes the MSE over all nodes along the centerline. Since adding a spring at each of those would significantly deviate from our problem setting, conventional design methods are not plausible.

Due to the difficulty of this problem, or indeed any realistic metamaterials design, searching over larger datasets to locate compatible unit cells while meeting the desired performance is also expensive or even intractable. In our case, we are only able to use the smaller diverse subsets S_{20}, SP_{20} and P_{20} introduced earlier, as well as baseline random subsets R_{20}. Since there are 120 macro-elements in the cantilever, this still means that there are 120^{20} possible combinations of unit cells for each subset.

For this example, we follow our two-stage optimization framework [124], wherein inverse TO is utilized in the first stage to determine the macroscale property distribution, and combinatorial optimization based on weighted graphs is used in the second stage to assemble unit cells that meet the target properties with compatible boundaries. Specifically, we define the following optimization problem for the first stage:

$$\begin{aligned} &\underset{\boldsymbol{C_e}}{\text{minimize}} && \frac{1}{n}\|\boldsymbol{u}(\boldsymbol{l}) - \boldsymbol{u}_t\|_2^2 \\ &\text{subject to} && \boldsymbol{K}(\boldsymbol{C_e})\boldsymbol{U} = \boldsymbol{F}, \\ & && -\phi(\boldsymbol{C_e}) \leq 0. \end{aligned} \tag{5.6}$$

Compared to the problem solved via genetic algorithm in the previous section (Eq. 5.5), this inverse property design directly uses the element stiffness matrix $\boldsymbol{C_e}$ as design variables, which are constrained by the signed L2 distance field ϕ of the property space of the full subset G. This inverse problem can be efficiently solved with MMA [122].

After obtaining the optimized macro-property distribution, we construct a grid-like weighted graph with each node representing an element in the macrostructure, and with edges connecting neighbouring microstructures. We can then view the assembly problem as selecting an index from the given subset to label each node in the graph. The Euclidean distance to the target property is assigned as the nodal weight during this process, and the ratio of disconnectivity, r_{dc} defined in the last section, is assigned as the edge weight for each pair of neighboring nodes. With this graph, we can use a dual decomposition Markov random field (DD-MRF) method [173] to efficiently find the optimal labels of the graph with the lowest sum of nodal and edge weights, thereby designing a full structure that meets the target properties and is well-connected.

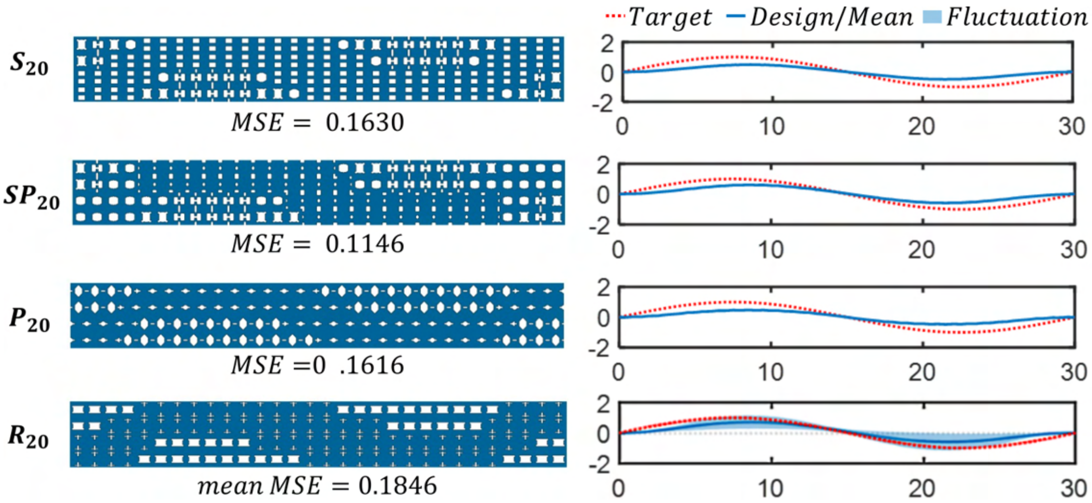


Figure 5.7. Optimized structures using different subsets, and their associated displacement profiles, for the cantilever example.

Since the labeling problem for the graph is a complex combinatorial optimization process where a large candidate set of unit cells equates to an immense search space, a small subset is required for a higher efficiency. As aforementioned, we use three diverse subsets, S_{20}, SP_{20}, P_{20}, and five subsets randomly selected without METASET, R_{20}, each with 20 unit cells as the candidate sets for the second stage. The resulting full structures and their respective MSE values and displacement profiles are shown in Fig. 5.7. We repeat the design using random subsets five times, then plot the mean displacement profile and depict the fluctuation of the results with the shaded area.

By virtue of our weighted graph method, all optimized designs have compatible boundaries. However, the subsets which account for shape diversity, i.e., S_{20} and SP_{20}, include a wider variety of unit cells in the full structure. This can be credited to an observation we made in the previous MBB beam example, that a shape diverse set can provide more compatible pairs, rendering a larger feasible design space for the assembly problem. In addition, we note that although some random subsets can achieve relatively low MSE, this performance is not guaranteed; the mean MSE is still the worst overall. In contrast, the shape and property diverse subset SP_{20} has the lowest MSE value. The reason is that, even with small subsets, shape diversity provides better compatibility while property diversity helps to achieve the target property distribution. This is again in line with our findings that a small yet diverse subset considering shape and properties is a boon for data-driven metamaterials design, and has exciting implications for future works.

5.6. METASET for Discovery of Diverse 3D Unit Cell Classes

Beyond selecting diverse subsets for direct use in design, another advantage of METASET is eliminating inherent bias by optimizing the diversity of a dataset. We demonstrate this with a 3D study, first introducing a new method based on periodic functions to generate families of unit cells with the same underlying structure but varying densities, which although fast creates a great number of overlapping shapes. Our goal in applying METASET to this 3D data is to sift through the overlaps to discover diverse sets of unique isosurface families, which can subsequently be leveraged for data-driven design or ML of, e.g., property prediction or generative models (Fig. 5.1).

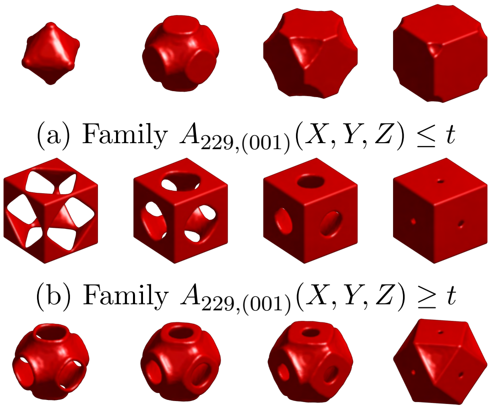
Triply periodic isosurface unit cells, whose symmetries follow those of crystal structures [141], are often used in 3D mechanical metamaterials design due to excellent surface area-to-performance ratios and manufacturability [36]. In addition, their representation as level set functions allows the density of the unit cells to be easily manipulated for functionally-graded structures [36, 134] and tailorable acoustic bandgaps [11]. A level set function $f(x, y, z) = t$ is an implicit representation of geometry where the t-isocontour, i.e., the points where $f = t$, describes the surface of the structure, while the locations where $f < t$ are solid material, and void where $f > t$. Thus, by varying the isovalue t, an entire family of isosurface unit cells with graded densities can be extracted from one level set function.

The most prevalent type of isosurfaces used in metamaterials design is a special subset known as Triply Periodic Minimal Surfaces (TPMS). However, only a few TPMS families have been used since their functions are complex to derive [141]. For example, Maskery et al. use six families in their design work [36], while Li et al. use four [134]. Moreover, it has not been investigated whether these few families cover the gamut of shapes and properties needed for design applications. Suppose a researcher wishes to design a new functionally-graded 3D metamaterial by tuning the densities of isosurface functions, but does not know beforehand which families would best suit their application. Due to the computational expense of design in 3D, they may desire to select a smaller set of families that can then be used in their optimization method. In this section, we present METASET as a procedure to choose those families such that the resultant subset has large coverage over different properties and shapes. In doing so, we also demonstrate that METASET removes bias in datasets by maximizing diversity.

5.6.1. Generation of 3D Unit Cell Families

Before selecting diverse families, we must first generate an initial pool to choose from. Thus, to build a large 3D dataset, we propose a new method to create isosurface families based on the level set functions of crystallographic structure factors, which describe how particles are arranged in a crystal unit cell [174]. In contrast to most unit cell generation methods, our approach here does not set targets in the property space or use TO, and different from TPMS functions, a larger variety of shapes can be found without complex derivations.

In crystallography, structures that are invariant under the same symmetry operations belong to the same space group, of which there are 230 for 3D structures [174]. For the



(a) Family $A_{229,(001)}(X, Y, Z) \leq t$

(b) Family $A_{229,(001)}(X, Y, Z) \geq t$

(c) Family $A^2_{229,(001)}(X, Y, Z) \leq t^2$

Figure 5.8. Examples of unit cells from isosurface families generated by the structure factor for space group No. 229 and $(hkl) = (001)$. The effect of increasing t to create a family is shown from left to right.

purposes of our work, we will focus on the 36 cubic groups, No. 195 through 230, to obtain our level set functions. Experimentally, the space group of a crystal can be determined through, e.g., X-ray techniques, by scattering radiation off a lattice plane denoted by (hkl), and then observing the diffraction pattern. These symmetric patterns have been analytically modeled as *structure factors*, which are periodic functions of the form:

$$f_{group,(hkl)}(X,Y,Z) = A + iB, \tag{5.7}$$

where $A = \cos\big(hX + kY + lZ\big)$, $B = \sin\big(hX + kY + lZ\big)$, $X = 2\pi x$, $Y = 2\pi y$, and $Z = 2\pi z$. The equations of these structure factors are listed in [174] for all space groups and their allowable (hkl).

We can split each structure factor into six isosurface families by separating A and B in Eq. 5.7 (inspired by [141]), and converting them into level set functions as follows:

$$\begin{aligned} &A_{group,(hkl)}(X,Y,Z) \leq t, \\ &A_{group,(hkl)}(X,Y,Z) \geq t, \\ &A^2_{group,(hkl)}(X,Y,Z) \leq t^2, \end{aligned} \tag{5.8}$$

and similarly for $B_{group,(hkl)}$. These, respectively, correspond to setting as solid material the function values that are less than t (Fig. 5.8a), greater than t (Fig. 5.8b), and in between $-t$ and t (leading to a "thin-walled" structure; Fig. 5.8c).

Thus, instead of using the limited TPMS functions, we can use the structure factors of all 36 cubic space groups and their corresponding (hkl) to generate a greater number of isosurface families for data-driven design. To ensure manufacturability, we also identify the feasible density range of each family by prohibiting internal voids and disconnected features, and eliminate families whose feasible range is $\rho_{max} - \rho_{min} < 0.2$. In this way, we quickly created 294 families without performing property-driven optimization. Although efficient, this method also causes an imbalance in geometry, since several structure factors differ only by a coefficient and lead to overlapping families. For example, the equations for space groups No. 195 and 196 listed in [174] are related as $A_{195,(hkl)} = 4 \cdot A_{196,(hkl)}$, and therefore generate the same structures. Next, we demonstrate the prowess of METASET in systematically removing such overlaps when selecting diverse subsets.

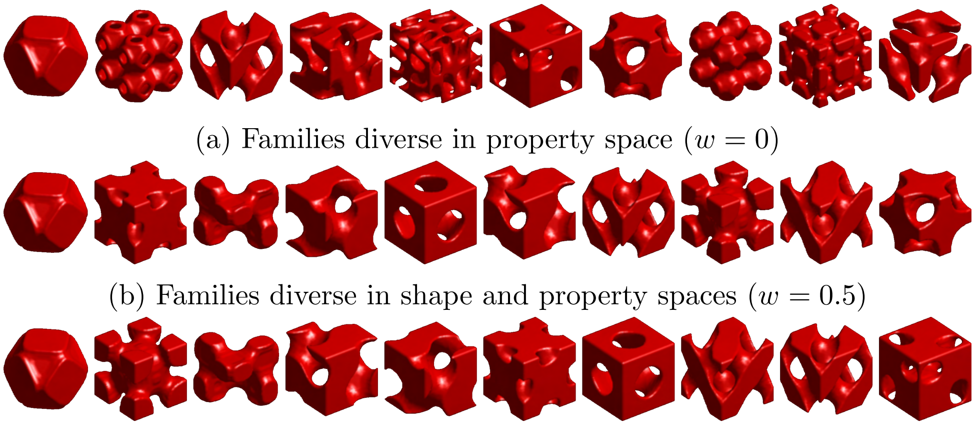
5.6.2. Diverse 3D Classes and Comparison of Shape Similarity Metrics

While applying METASET to discover unique isosurface families, we also test the impact of the two proposed 3D shape similarity metrics (Sec. 5.4.1.2): the Hausdorff distance and the cosine similarity between deep learning-based embeddings. As the families are comprised of a range of densities and thus shapes and properties, we need to capture the similarities of individual unit cells while assessing the similarities between families. Therefore, we generate 100 samples from each family covering the feasible range identified in the previous section, giving 29,400 unit cells total. Each unit cell is represented as a 4096-dimensional point cloud by first converting its level set field into a triangle mesh [175], and then sampling on the triangular faces [176]. We also remove any small disconnected features during post-processing, and find the homogenized elastic tensors of each unit cell using a code modified from [84].

To quantify the similarity between two families, we assume each family is a collection of points, where each point corresponds to a unit cell. This reduces the problem of finding similarity between two families to one between two point sets using the Hausdorff distance (Eq. 5.1). We calculate the similarity between families C and D in two steps: first using one of the 3D metrics to calculate the distance between individual unit cells $c \in C$ and $d \in D$, and then substituting this into the Hausdorff distance to obtain the *inter-familial* distance, $h(C, D)$. Intuitively, this means that the shape similarity between two families is the maximum of the similarities between closest-in-shape pairs of unit cells. In property space, the similarity between families is related to the maximum of the pairwise Euclidean distances between each unit cell's properties. Therefore, rather than simply averaging the features of each family, the inter-familial similarities also consider the diversity of individuals within each family. In short, we apply METASET to our 3D dataset using two approaches to measure shape similarity:

- H-H: Hausdorff distance between unit cells, followed by Hausdorff distance between families
- E-H: embedded cosine similarity between unit cells, followed by Hausdorff distance between families.

Utilizing both of our shape similarity metrics, along with the property metric, we find diverse subsets of 10 isosurface families. In addition, we vary the joint diversity weight w between 0 and 1 (Eq. 5.4). Some example subsets of diverse families are shown in Figs. 5.9 and 5.10, where the median sample from each family are pictured. Like the 2D diverse subsets (Sec. 5.5.2), the property-only and shape-only sets (Figs. 5.9a and 5.9c, respectively) share very few of the same families. Intriguingly, the shape diverse sets obtained from either metric contain families generated from the same space group and (hkl), but different level set forms (Eq. 5.7). For example, with the H-H approach, the fourth and fifth items in Fig. 5.9c have the equations $A_{213,(011)} \geq t$ and $A_{213,(011)} \leq t$. The



(a) Families diverse in property space ($w = 0$)

(b) Families diverse in shape and property spaces ($w = 0.5$)

(c) Families diverse in shape space ($w = 1$)

Figure 5.9. Examples of subsets of 3D isosurface families selected by METASET using the H-H shape metric.

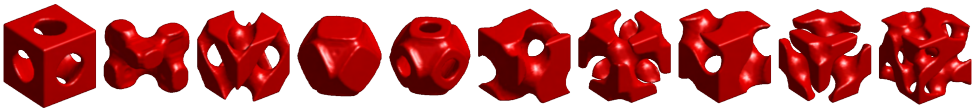


Figure 5.10. Shape diverse subset ($w = 1$) selected by METASET using the embedding-based E-H shape metric.

same families appear with the E-H approach as the eighth and sixth items. One could think of these as completely different shapes with almost no overlaps, which is further validation of the shape diversity chosen by METASET.

Comparing the subsets obtained via either similarity metric (Figs. 5.9c and 5.10), we observe that 6 out of 10 shape diverse families overlap, indicating that the choice of metric does not drastically impact diversification. This is supported by a correlation coefficient of 0.836 between the H-H and E-H shape similarity kernels, L_S, of the shape diverse families. Additionally, we cross-examine these results by applying the E-H approach to score ($det(L_S)$) the shape diverse subset chosen using H-H, and vice versa. As a baseline, we also randomly sample 10,000 sets of 10 families without METASET and measure their diversity with respect to each metric. The results are reported in Table 5.2, where

Table 5.2. Shape diversity scores of subsets of 10 isosurface families, evaluated using either the Hausdorff (H-H) or embedded (E-H) shape metrics. The first two columns are for shape diverse subsets selected by METASET; the last shows the maximum of 10,000 random sets. The highest scores of each row in **bold** indicate that METASET always maximizes the diversity score with respect to the metric used during selection.

	METASET (H-H)	METASET (E-H)	Random Sampling
Score (H-H)	**1.0554E-04**	6.3690E-05	2.8504E-05
Score (E-H)	1.6250E-13	**6.4271E-12**	5.4262E-14

the greatest (most diverse) scores across each row reveal that the greedy algorithm will maximize the diversity score regardless of the similarity metric employed. Moreover, the subsets chosen by METASET have higher diversity than the random ones no matter which metric is used to evaluate the score.

The high diversity of our subsets can also be seen in Fig. 5.11, where their property and shape scores using H-H are plotted against those of the 10,000 random subsets. Here, 99.74% of the random sets (which are representative of the distribution of pairwise similarity values for our dataset) still fall short of the optimized subset with the lowest shape diversity score. This is compelling evidence that 1) the original dataset was severely imbalanced, and 2) METASET is able to combat such bias and select more diverse subsets.

Fig. 5.11 additionally visualizes the trade-off between diversity in the shape and property spaces. Although our greedy algorithm maximizes the joint diversity score, the independent shape and property scores illustrate that, in general, the diversity in one space drops as we select sets that are more diverse in the other. This trade-off might raise a question as to whether a set of families that are quite diverse in property space can have low diversity in shapes, even though similar shapes are expected to possess similar

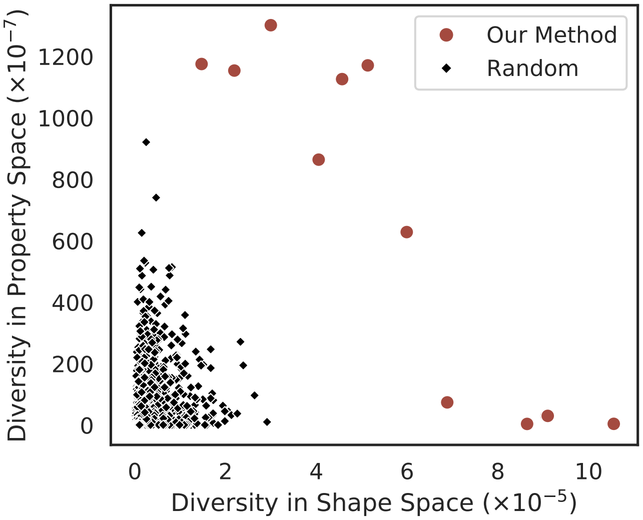


Figure 5.11. Trade-off between diversity in property vs. shape (using the H-H approach) spaces. The minimum diversity in shape space for optimized sets has a diversity score greater than 99.74% of random samples.

properties. Our previous observation emerges as an answer: the sets of families with higher diversity in property space and seemingly “low” diversity in shape space actually have larger shape scores than the majority of the random sets. Therefore, the highest diversity in property space is achieved by a set of families which are also very diverse in shape.

Finally, we note that our diversified sets include isosurface families beyond the common TPMS used in existing metamaterials design, such as the Primitive, Gyroid and Diamond (see [36, 134]). We provide the data of the METASET results publicly so that our diverse families can be employed by any designer in their work as well. For example, these can be directly utilized in existing functionally-graded design methods such as [134]. Data-driven design with diverse isosurface families will be investigated in future works.

5.7. Discussion

Although we illustrated the benefits of METASET with several case studies, there are nevertheless some topics worthy of examining in the future. From our design of 2D aperiodic structures, we saw that shape diverse subsets may increase the chance to find compatible neighboring microstructures, while property diverse sets might enhance problems that require a wider range of target properties at the cost of connectivity. This dependence on shape vs. property diversity extends to ML tasks in the data-driven design framework (Fig. 5.1) as well. To train property prediction models, one may need a property diverse dataset, while for a deep generative model that learns geometric features, a shape diverse set might be more appropriate. Along these lines, it would be interesting to further validate the improved performance of design and ML tasks using our subsets of diverse 3D unit cell families in a future work.

In the 2D examples, we also observed that smaller subsets led to designs with performance closer to the targets; in fact, we found using METASET that only 20 unit cells were enough to form a diverse subset. In most cases, the benefits of reducing the search space, model training time, or storage requirement of the dataset could outweigh any loss of data. However, certain applications such as ML may need large datasets. A key benefit of using a METASET, even for large subset sizes, is that it reduces bias by rank ordering all items in the dataset. The items with the highest redundancy in shape or property (like duplicates) are pushed toward the end of the rank-ordered list, so that ML algorithms trained on any subset will be less biased. While it is not difficult to increase the size of the set, determining how much data is enough is more challenging since this too is contingent on the application and any limits on computational cost. The effectiveness of size and diversity on specific tasks in metamaterials design is an important question for future studies. Fortunately, the ease at which a subset's size as well as the weight of shape and property diversity can be explored is yet another advantage of METASET.

Lastly, we remark that the capability of METASET depends on the choice of similarity metrics as well as the definition of the joint similarity kernel, both of which are avenues of further research. Our 3D study demonstrated that METASET will maximize diversity regardless of the metric adopted. Although the results indicated that different shape similarity metrics can be highly correlated and slightly change the diverse subsets, there are a wealth of other choices that may provide different results. Extending METASET to more complex properties, like dynamic ones, may necessitate new metrics. For the joint DPP kernel, we chose a simple weighted sum to join the shape and property matrices, thereby casting the greedy selection as a multi-objective problem. We found in Sec. 5.5.2 that this was a valid assumption, but other methods to combine kernels while preserving submodularity are also possible. However, swapping these to best suit the application is easily done since the input of the DPPs-based greedy algorithm in METASET is a positive semi-definite similarity kernel that can be obtained from any appropriate metric or definition.

5.8. Conclusions

In this chapter, we proposed a methodology, METASET, that incorporates joint diversity in the shape and property spaces into data selection to improve the downstream tasks in data-driven design. As an enhancement to any existing data-driven framework, METASET is efficient and flexible, allowing the emphasis on either shape or property to be easily traded by measuring and maximizing the joint diversity of subsets through a weighted DPP similarity kernel. To calculate this kernel matrix, we introduced similarity metrics that cater specifically to 2D and 3D metamaterials.

By way of our 2D aperiodic metamaterial design examples, we demonstrated that small yet diverse subsets of unit cells can boost the scalability of search algorithms while leading to designs with greater performance and enhanced boundary compatibility. This revelation shakes a common belief in the field of data-driven mechanical metamaterials design that a larger and denser dataset is required to design well-connected structures while still meeting the target behavior. To our knowledge, this is the first time that such a result has been studied and presented.

In our 3D case study, we not only proposed a new method to generate triply periodic isosurface unit cells using crystallographic structure factors, but also verified that METASET can effectively discover unique unit cell families in order to build diverse, unbiased and economical datasets for design regardless of the shape similarity metric employed. Different from well-known TPMS unit cells, our dataset of families are optimized for shape and property diversity rather than arbitrarily chosen. In future works, we will explore the use these diverse families for data-driven metamaterials design and ML.

Although this paper focused on showcasing METASET through the design of mechanical metamaterials, the methods we proposed are broadly applicable to other metamaterial domains, or indeed any other design problems that need to balance design space against some performance or quality space. In design ideation, our method can be used to select ideas that are functionally different from each other while achieving different performance goals. It can also be integrated with existing multi-objective optimization algorithms as a niching method. To contribute to the growth and capability of data-driven metamaterials design methods and other fields, we have shared diversified subsets of 2D and 3D unit cells, as well as the corresponding equations of isosurface families. These unit cells can be directly plugged into the application of any metamaterials designer.

Part 3

Data-Driven and Diversity-Enhanced Design of Multiscale Structures

CHAPTER 6

Data-Driven Topology Optimization of Functionally Graded Structures with Multiclass Shape Blending

6.1. Introduction

Multiscale mechanical structures present exciting functionalities and unprecedented performance ranging from global objectives like light-weighting, thermal conductivity and energy absorption [2, 177–179] to targeted local behaviors such as shape morphing or pattern reconfiguration for soft robots [30, 37] and active airfoils [28]. To design such complex structures, multiscale topology optimization (TO) has risen to prominence and flourished. While early research focused on periodic microstructures (also known as unit cells or building blocks), two types of *heterogeneous* designs, in which neighboring microstructural topologies differ from each other, now surpass them in terms of performance: fully aperiodic systems [34, 65, 66] and functionally graded structures (FGS) [50]. The two approaches, which are reviewed in Sec. 6.2, compete in terms of design freedom and efficiency.

In this work, we aim to bridge the freedom of aperiodic designs with the efficiency and smooth interfaces of FGS. We propose to inherit the advantages of both through a data-driven framework for multiclass functionally graded structures that mixes several families, i.e., classes, of microstructure topologies to create spatially-varying designs with guaranteed feasibility. The key is a new multiclass shape blending scheme that generates smoothly graded microstructures without requiring compatible classes or connectivity and feasibility constraints. Moreover, it transforms the microscale problem into an efficient, low-dimensional one without confining the design to predefined shapes. Compliance and shape matching examples using common truss geometries and diversity-based freeform topologies demonstrate the versatility of our framework, while studies on the effect of the number and diversity of classes illustrate the effectiveness. The generality of the proposed methods supports future extensions beyond the linear applications presented.

6.2. Literature Review

Considering heterogeneous microstructures in multiscale designs expands the realm of possibilities in terms of design functionality. For example, structures can be designed to deform in desired patterns, forming elastic joints [34] or soft robots [37]. Recent design methods in pursuit of these complex structures can be split into fully aperiodic systems and FGS. Both typically follow a homogenization-based approach where the material properties of each element in the macroscopic structure are replaced by the effective properties of the microstructure at that location. This greatly expedites multiscale design as the performance can be evaluated at the macroscale, albeit with lowered accuracy since heterogeneous structures break the assumption of infinite periodicity [180]. The two approaches differ in that aperiodic systems can be composed of very different microstructures, whereas the change in the topologies and/or volume fractions of neighboring microstructures in FGS are deliberately designed to be continuous. In this section, we review the existing methods for multiscale TO in general, then the methods for heterogeneous structures specifically, including a brief comparison between aperiodic and functionally graded approaches.

6.2.1. General Multiscale Topology Optimization

The current major TO methods to computationally design metamaterials can be categorized into single-scale TO of the unit cell(s), which are then tiled into a predefined macrostructure [32, 55–57], and hierarchical or concurrent multiscale TO that optimize both the unit cell(s) and macrostructure [58, 59]. In hierarchical TO, the design domain are decomposed by the micro- and macro-scales [61, 181]. The unit cell within each macro-element is optimized along with the global structure, which allows spatially varying microstructures. Accordingly, the geometrical descriptors of all of the unit cells are the design variables, which unfortunately can result in millions of design variables [80]. The problem is solved in a two-level scheme, where the outer problem is TO of the global, macro-structure, and the inner problems are the TO of all unit cells. Concurrent TO, on the other hand, limits the number of unique unit cells to a finite number, and evolves the macro- and micro-scale topologies simultaneously. The macro-scale problem can be formulated as a multi-material problem where each "material" corresponds to a different unit cell [115, 149, 182], or as a single-material problem with pre-specified zones for the different unit cells [59]. The optimality of the concurrent method is subject to the human choice of the number of distinct unit cells and, if applicable, the assigned zones.

Maintaining continuous boundaries is a key challenge when multiple unit cells are allowed. Techniques such as varying the volume fraction or size of one unit cell (chosen manually by the designer) to match an optimized density field [134, 183], adding connectivity constraints [59], subdomains [58] and substructuring [184] have been proposed. For the most part, the methods compound the already computationally demanding TO, especially in 3D or high resolutions. Taking advantage of growing computational resources, some have also explored using GPUs to optimize super high resolution unit cells [185], and replacing expensive numerical homogenization with data-driven mesh coarsening [132] and machine learning models [3, 71–73, 128, 186]. Another intriguing work uses deep reinforcement learning for shape optimization of spatially varying unit cells [187]. An “actor” iteratively suggests the size parameters of unit cells and is rewarded based on the improvement in the macroscale compliance. However, it is also limited to unit cells whose parameterizations must be chosen beforehand. Unfortunately, while these may alleviate computational expenses, they are still subject to problems with local optimality and connectivity.

On the other hand, the straightforward, classical density-based TO method without density filters, i.e., allowing any density between 0 and 1, can be used to both design the macroscale topology and decide the volume fraction [134] or size [183, 188] of each microstructure. If the scaling law – the relation of Young’s modulus to volume fraction – of a microstructure class is known, the pseudo-densities can be replaced by the physical densities to quickly obtain FGS [134, 189]. Although relatively straightforward to implement, this approach will inevitably be subjected to the local optima problem of density-based TO.

6.2.2. Topology Optimization of Heterogeneous Structures

Although aperiodic structures allow immense design freedom – a different microstructure at every location – they come at the cost of explosive problem sizes. Two avenues of data-driven methods have emerged to counter this "curse of dimensionality". One assembles microstructures from pre-computed libraries via combinatorial optimization [34, 35, 65, 190], and the other accelerates gradient-based design by creating deep learning (DL) models from massive datasets for dimension reduction and rapid property predictions [1, 72].

These approaches are powerful, but require large overhead costs to build the datasets and models, and need careful strategies to select reasonably compatible neighboring microstructures. Even then, they may not achieve connectivity on par with FGS. Combined with the use of the effective, homogenized properties, low connectivity can result in the manufactured performance deviating greatly from the optimized design. These drawbacks mean that, in their current state, data-driven methods are difficult to scale to large systems with heterogeneity and complex physics, e.g., nonlinear mechanics. Compared to FGS, such approaches are less suitable when stress concentrations or expensive property simulations must be avoided.

In contrast, some works have indicated that the continuous interfaces and more gradual topological change in FGS may be able to mitigate the errors from homogenization [34, 50, 60]. Functional grading can be further categorized into three camps: continuously varying volume fraction [134, 191–193], topology [63, 72], or a hybrid of both [3, 64]. While traditional FGS use the first, recent research is shifting towards the latter two, which have demonstrated that expanding the design space to include multiple topology types can considerably improve the structural performance. Our work belongs to the hybrid one, and to facilitate ease of discussion, we define a *microstructure class* as a family of microstructures that possess the same overall topological concept but vary individually by volume fraction. Hence, the works in the last two categories can be termed *multiclass*.

Within existing multiclass methods, the prevailing strategy is to treat FGS design as a multi-material TO problem by allocating each class to its own region with distinct boundaries. This assumes that the interfaces between classes are perfectly connected. As a result, most approaches pre-define a few mutually compatible classes [3, 64] or fix their connections [61, 62, 186], which reduces computational cost and complexity but can yield suboptimal solutions.

Similar methods in the general multiscale TO field accomplish connected heterogeneous designs without the above simplifications by: (1) sharing finite elements and design variables at interfaces [58], (2) adding connectivity constraints [59, 60], (3) controlling the change in the properties of intermediate microstructures [194], (4) creating geometric gradations during pre- or post-processing [63, 194, 195], and (5) interpolating random field representations of microstructures [72]. Of these, only Refs. [58, 60, 63] concurrently design the macrostructure as well as the distributions of multiple microstructures, and just Ref. [64] also optimized the graded volume fractions. Moreover, many do not scale well with the number of classes.

6.3. Contributions

Merging the advantages of the two heterogeneous approaches, we propose a general TO framework for multiclass FGS that achieves smooth transitions between multiple microstructure classes without additional constraints, even if those classes are not compatible initially. The cornerstone of our approach is a novel multiclass shape blending scheme that generates new microstructures from a small set of predefined basis classes while guaranteeing certain aspects of feasibility. We define a design as *feasible* when the microstructures are not only self-connected (i.e., have no disconnected features) but also well-connected to their neighbors. In addition, we desire the microstructures to meet a minimum feature size, as required in some manufacturing techniques.

Our framework departs from existing heterogeneous design methods in several ways:

- First, we create a continuous and low-dimensional microstructure representation by using the parameters of our shape blending scheme as design variables. This transforms the microscale problem into a parametric one without using costly, geometry-based DL. While we do require predefined basis classes, this by no means restricts our design, as we allow the regions for each class and the boundaries between them to be blurred so that the microstructures anywhere in the FGS can be novel, i.e., not found in the initial basis classes.
- Second, our blending scheme integrates naturally into existing TO methods. In this work, we incorporate it with discrete and gradient-based TO, along with a new penalty that promotes diverse designs. These significantly reduce the cost to concurrently optimize the macrostructure as well as the graded topologies, volume fractions, and distributions of any number of microscale classes. Thus, through blending, our framework features design freedom near that of aperiodic methods while inheriting the efficiency of functionally graded design.

- The blending parameters serve as effective inputs for feedforward neural networks that predict the effective properties of new microstructures and can be re-used in multiple applications, further accelerating design.
- Certain feasibility metrics (e.g., self-connectedness and minimum feature size) can be built into the blending scheme so that they do not need to be included explicitly in the TO problem. Furthermore, the flexibility of using basis classes permits designers to incorporate expert knowledge and eliminates the frustration of choosing compatible classes. If desired, manual selection can be removed altogether by extracting classes from open-source databases, e.g., using diversity metrics [196].

In this chapter, we introduce three crucial components in our approach: (1) multiclass shape blending and global interpolation schemes (Sec. 6.4), (2) neural networks for property prediction (Sec. 6.5), and (3) concurrent multiclass data-driven topology optimization (TO), which ties all of the methods into one framework (Sec. 6.6). With compliance and shape morphing examples, we demonstrate the efficiency and inherent ability of our approach in designing multiscale structures with continuous transitions. The benefits of multiclass FGS are verified by utilizing both common truss-type and diverse freeform (topology-optimized) basis classes, and by comparing our results to designs in literature.

6.4. Multiclass Shape Blending and Smooth Interpolation

6.4.1. Background

Our proposed shape blending scheme is heavily inspired by the computer graphics field, where morphing one geometric model into another has long been studied and utilized in, e.g., animation films and video games [197, 198]. These methods have also supported applications like medical imaging [199] and metal-forming manufacturing simulations [200]. In fact, evolving geometries through surface representations [201] and partial differential equations [202] is the foundation of the popular level set TO method [44], while combining shapes using distance fields is the bedrock of TO algorithms that use geometry projection [203] and Movable Morphing Components [204].

Most closely related to the interest of this paper are blending techniques that use function representations (FReps) of shapes, which parameterize any geometric model as a series of operations (e.g., unions, differences, and intersections) performed on a set of primitives or basis geometries [205]. It is extremely flexible as it allows the bases to be defined by any representation, e.g., meshes or voxels, and any resolution. In our case, we represent our basis microstructure classes as continuous signed distance fields (SDFs), which are implicit function representations similar to level sets. That is, the sign of the field determines whether the material is solid or void at any arbitrary point (x, y) within the microstructure domain, D, as follows:

$$\begin{cases} \mathbf{\Phi}(x,y) > t & (x,y) \in \Omega \\ \mathbf{\Phi}(x,y) = t & (x,y) \in \Gamma \\ \mathbf{\Phi}(x,y) < t & (x,y) \in D/\Omega \end{cases} , \tag{6.1}$$

where $\mathbf{\Phi}$ is the SDF, Ω is the solid region, and Γ is the boundary of the solid structure. The isovalue t controls the isocontour of the field and therefore enables us to tune volume fractions. With this powerful representation, we can not only generate an entire family of microstructures over a continuous range of volume fractions, but also combine multiple SDFs to create novel classes.

Blending operations to mix SDFs have been studied for decades, beginning with the simple set-theoretic operation for the union of the function representations of two shapes, $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ [206]:

$$\mathbf{\Phi}_{union} = \max(\mathbf{\Phi}_1, \mathbf{\Phi}_2). \tag{6.2}$$

Since then, many works have improved metamorphosis using function-based operations [205, 207], optimization [208], and DL-based interpolation [209, 210]. However, these can require intensive user-interaction and computational costs, and are, therefore, intractable or unsuitable for multiscale design.

6.4.2. Multiclass Shape Blending Scheme

Rather than using complex formulations, our proposed multiclass shape blending scheme is a combination of two simple techniques: (1) a weighted sum of basis classes based on cross dissolving, and (2) an activated union with the lower feasible bounds of each basis. The entire scheme is differentiable and efficient, which fits in well with gradient-based optimization algorithms.

Crucial to our scheme are basis microstructure classes, which do not need to be mutually compatible. We denote the SDF representations of arbitrary basis classes as $\mathbf{\Phi}_d^B$ for $d \in [1, \ldots, D]$, where the shapes $\mathbf{\Phi}_d^B$ and total number of classes D can be defined by the user.

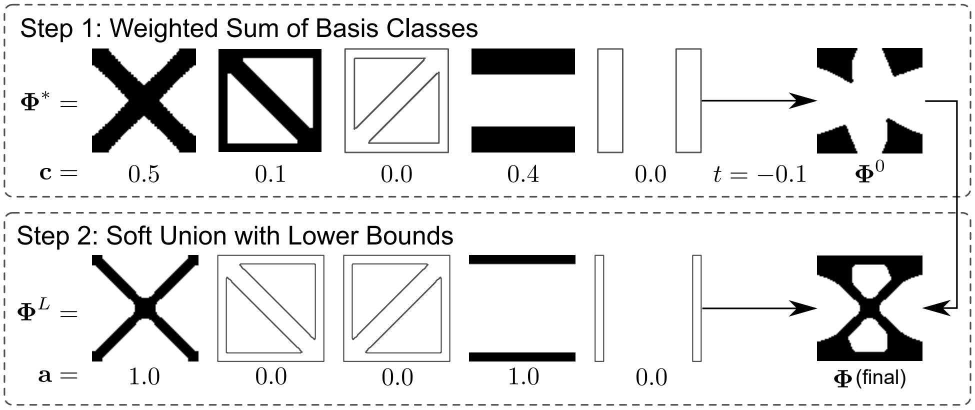


Figure 6.1. Illustration of using multiclass shape blending scheme to generate a microstructure from truss basis classes. Outlined shapes represent classes whose weights are zero. Step 1 (top row; Eq. 6.3) may cause broken shapes, but Step 2 (bottom row; Eq. 6.4) ensures the final microstructure, $\mathbf{\Phi}$, is feasible.

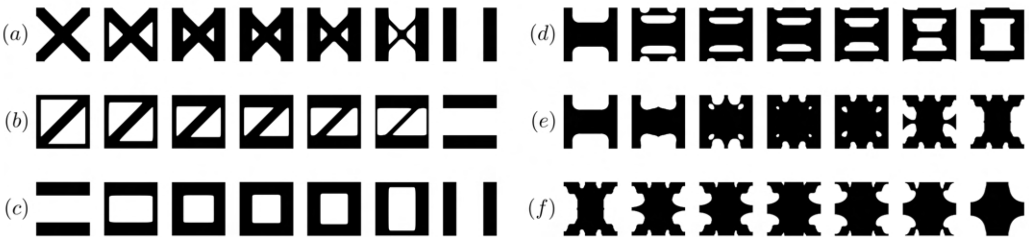


Figure 6.2. Examples of blending pairs of basis classes. The first and last microstructures are from basis classes, and those in between are produced by linearly interpolating the weights in the blending scheme. Between truss classes (left column): (a) 1 and 5, (b) 2 and 4, and (c) 4 and 5. Between freeform classes (right column): (d) 2 and 3, (e) 2 and 4, and (f) 4 and 5. The classes themselves are depicted in Fig. 6.4.

Prior to blending, we normalize the D basis classes so that they can be mixed fairly, and also find their lower feasible bounds. The following pre-processing steps only need to be performed once per set of bases:

(1) Choose a common volume fraction, v^*, and find the representative SDF for each basis, $\mathbf{\Phi}_d^* = \mathbf{\Phi}_d^B + t_d^*$, such that it has v^*. We use the well-known bisection algorithm.

(2) Find the SDF of the lower feasible bound of each basis, $\mathbf{\Phi}_d^L = \mathbf{\Phi}_d^B + t_d^L$, using any desired feasibility metric. In this work, we set a minimum feature size of 4 pixels for a 50×50 microstructure.

After this one-time process, we can use $\mathbf{\Phi}_d^*$ and $\mathbf{\Phi}_d^L$ repeatedly for our proposed multiclass blending. The first step of the scheme is based on cross dissolving [198], which can be simply defined as a linear interpolation between the source and target geometries. For example, between two SDFs, the blending operation is $\mathbf{\Phi} = (1-c)\mathbf{\Phi}_1^* + c\mathbf{\Phi}_2^*$, for $c \in [0,1]$. This can induce a double exposure effect where traces of both shapes co-exist in the blended result for middling values of c. While that causes unnatural morphing of, e.g., human faces, it organically achieves connected transitions between neighboring microstructures.

We express multiclass interpolation as a weighted sum of the bases:

$$\mathbf{\Phi}^0 = \sum_d^D c_d \mathbf{\Phi}_d^* + t, \tag{6.3}$$

where $c_d \in [0,1]$. Although it effectively creates new classes of microstructures, this interpolation is agnostic to important geometrical features and can lead to broken shapes that have disconnected or thin features (see the top row of Fig. 6.1).

Therefore, to guarantee that the blended microstructures are sufficiently connected and feasible, we propose an additional step that enforces a lower feasible bound on blending. It also acts as an implicit constraint for simple manufacturing considerations, e.g., minimum feature sizes or volume fractions. This second step is an activated soft-max function, a continuous and differentiable extension of the set-theoretic union above (Eq. 6.2):

$$\boldsymbol{\Phi} = \frac{1}{\beta_2} \log \Big[\exp\left(\beta_2 \boldsymbol{\Phi}^0\right) + \sum_{d}^{D} a_d \exp\left(\beta_2 \boldsymbol{\Phi}_d^L\right) \Big], \tag{6.4}$$

where a_d are the activated weight parameters using the Heaviside function:

$$a_d = H(c_d, \beta_2, \eta_2) = \frac{\tanh\left(\beta_2 \eta_2\right) + \tanh\left(\beta_2 (c_d - \eta_2)\right)}{\tanh\left(\beta_2 \eta_2\right) + \tanh\left(\beta_2 (1 - \eta_2)\right)}. \tag{6.5}$$

By setting the threshold $\eta_2 > 0$, the activation guarantees that the minimum feasible bound of at least one – but not all, or else low volume fractions would be difficult to attain – of the bases are present in each blended result. That is, in this step, only weights that are greater than η_2 are activated to equal one (shaded shapes in the second panel of Fig. 6.1) while others are suppressed to zero (outlined shapes in Fig. 6.1). We find that setting η_2 to the 75th percentile of the weights, $\mathbf{c}$, works quite well in promoting connected transitions.

To further visualize blending between pairs of basis classes, we illustrate several examples in Fig. 6.2. In particular, we select pairs that do not appeared in our design results later (Sec. 6.7). No matter how simple, complex or incompatible the basis classes, our multiclass blending scheme is able to provide feasible, i.e., well-connected, intermediate microstructures. This is the benefit of (1) the interpolation between the SDFs of basis classes, and (2) the imposition of the lower feasible bounds of each basis through Eqs. 6.3 and 6.4.

With the two-step shape blending scheme, the representation of all possible blended microstructures, including those in the original basis classes, can be compactly expressed as the weight parameters. We can, therefore, formulate the microscale design variables as $\mathbf{c}^{(m)}$ for $m \in [1, M]$ desired optimal classes, transforming the typically high-dimensional optimization problem into a simple and efficient parametric one that can still generate a wide range of microstructures.

A final note regarding microstructure design is that, in practice, Eq. 6.3 allows each $c_d^{(m)} \in [0, 1]$ and can lead to $\sum_{d=1}^{D} c_d^{(m)} = 0$, which results in completely solid microstructures and occasional numerical issues. Moreover, it causes redundancy in the design space since taking $\mathbf{\Phi}^0 \geq 0$ to obtain the solid topology cancels out the least common denominator of the weights. There are numerous ways to enforce the sum of $c_d^{(m)}$ to be equal to one, such as multi-material interpolation schemes in TO. We extend one such scheme from our previous work [211] to normalize the basis class weights as follows:

$$\tilde{\mathbf{c}}^{(m)} = \mathbf{z}^{(1)} + \sum_{j=1}^{D-1} \left[\left(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)} \right) \prod_{k=1}^{j} c_k^{(m)} \right], \tag{6.6}$$

where $\mathbf{z}^{(j)}$ are constant one-hot encoded vectors for each basis such that $z_i^{(j)}$ equals 1 for $i = j$ and 0 for all $i \neq j$. Subsequently, c_d in Eqs. 6.3 and 6.5 are replaced with $\tilde{c}_d$. While our shape blending and design methods work well without Eq. 6.6, for increased stability in the optimization process, and the added bonus of reducing the microscale design variables, $\mathbf{c}^{(m)}$, to size $[1 \times D - 1]$, we use it in the remaining discussions.

6.4.3. Integration with Multiscale Design

In the context of multiscale design, the subscript e is added to denote individual microstructures, which each resides in one macroscopic quadrilateral 4-node finite element. Instead of directly optimizing D weights at each microstructure, we reduce the number of design variables by optimizing M new classes and interpolating them throughout the global structure using the distribution fields $\boldsymbol{\xi}^{(p)}$. An example with $M = 2$ is portrayed by the red and blue classes in the middle panel of Fig. 6.3. For demonstration, we artificially create the values of the macro- and micro-scale design variables in the middle panel, but during design, these values are optimized concurrently.

Similar to multi-material TO (and Eq. 6.6), we require that the sum of the distributions at each element equals one. Thus, to obtain each microstructure, e, we can globally interpolate the optimal class weights, $\tilde{\mathbf{c}}^{(m)}$, with:

$$\hat{\mathbf{c}}_e = \tilde{\mathbf{c}}^{(1)} + \sum_{j=1}^{M-1} \left[\big(\tilde{\mathbf{c}}^{(j+1)} - \tilde{\mathbf{c}}^{(j)}\big) \prod_{k=1}^{j} \hat{\xi}_e^{(k)} \right], \tag{6.7}$$

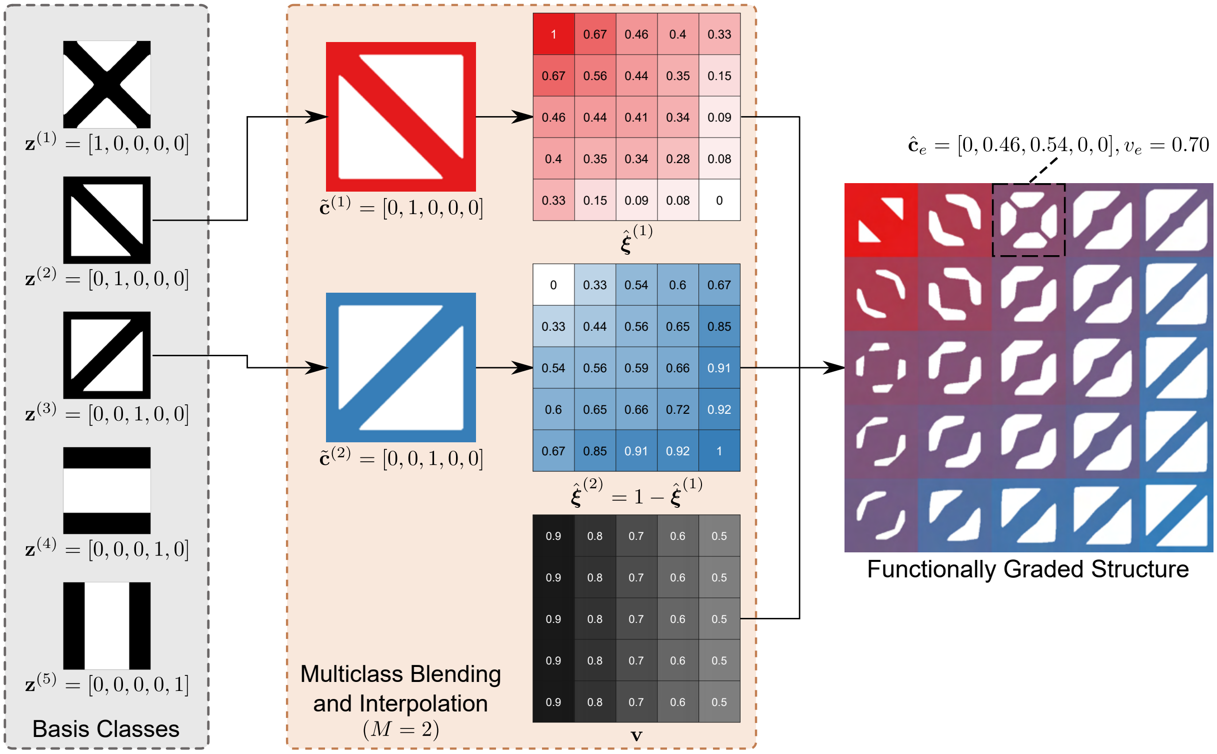
where $\hat{\boldsymbol{\xi}}$ are the smoothed distribution fields after applying the radial filter commonly found in TO methods [86] to encourage functional grading.

Therefore, combining Eqs. 6.3 through 6.7, our final multiclass shape blending scheme for a microstructure at element e is:

$$\boldsymbol{\Phi}_e = \frac{1}{\beta_2} \log \Big\{ \exp \big[\beta_2 \big(\sum_{d}^{D} \hat{c}_{e \cdot d} \boldsymbol{\Phi}_d^* + t_e \big) \big] + \sum_{d}^{D} a_{e \cdot d} \exp \big(\beta_2 \boldsymbol{\Phi}_d^L \big) \Big\}, \tag{6.8}$$

where $a_{e \cdot d} = H(\hat{c}_{e \cdot d})$ and t_e is found using the bisection algorithm to match a given or optimized volume fraction, v_e. This replaces Eq. 6.4 during optimization.

The process of blending and global interpolation to assemble a FGS is demonstrated in Fig. 6.3. For this, we use five truss-type basis classes and artificially create $M = 2$ new classes, $\tilde{\mathbf{c}}^{(m)}$, their distribution fields, $\hat{\boldsymbol{\xi}}^{(m)}$, and volume fractions, $\mathbf{v}$. In practice, these parameters would be optimized. For each microstructure, e, in the FGS, its globally


$\mathbf{z}^{(1)} = [1, 0, 0, 0, 0]$
$\mathbf{z}^{(2)} = [0, 1, 0, 0, 0]$
$\mathbf{z}^{(3)} = [0, 0, 1, 0, 0]$
$\mathbf{z}^{(4)} = [0, 0, 0, 1, 0]$
$\mathbf{z}^{(5)} = [0, 0, 0, 0, 1]$
Basis Classes
$\tilde{\mathbf{c}}^{(1)} = [0, 1, 0, 0, 0]$
$\tilde{\mathbf{c}}^{(2)} = [0, 0, 1, 0, 0]$
Multiclass Blending
and Interpolation
$(M = 2)$
$\hat{\boldsymbol{\xi}}^{(1)}$
$\hat{\boldsymbol{\xi}}^{(2)} = 1 - \hat{\boldsymbol{\xi}}^{(1)}$
$\mathbf{v}$
$\hat{\mathbf{c}}_e = [0, 0.46, 0.54, 0, 0], v_e = 0.70$
Functionally Graded Structure

1	0.67	0.46	0.4	0.33
0.67	0.56	0.44	0.35	0.15
0.46	0.44	0.41	0.34	0.09
0.4	0.35	0.34	0.28	0.08
0.33	0.15	0.09	0.08	0

0	0.33	0.54	0.6	0.67
0.33	0.44	0.56	0.65	0.85
0.54	0.56	0.59	0.66	0.91
0.6	0.65	0.66	0.72	0.92
0.67	0.85	0.91	0.92	1

0.9	0.8	0.7	0.6	0.5
0.9	0.8	0.7	0.6	0.5
0.9	0.8	0.7	0.6	0.5
0.9	0.8	0.7	0.6	0.5
0.9	0.8	0.7	0.6	0.5

Figure 6.3. Demonstration of the integration of multiclass shape blending and global interpolation. The basis classes are defined (left). Examples of $M = 2$ new classes, $\tilde{\mathbf{c}}^{(m)}$, their distributions, $\hat{\boldsymbol{\xi}}^{(m)}$, and volume fractions, $\mathbf{v}$, are given (middle). Each microstructure in the FGS (right) is generated by using Eq. 6.8.

interpolated weights, $\hat{\mathbf{c}}_e$, are found via Eq. 6.7, and its SDF generated by the blending scheme in Eq. 6.8.

6.5. Property Prediction with Neural Networks

The continuous and low-dimensional microstructure representation lends itself well to one of the simplest deep learning (DL) techniques: regression with neural networks. More specifically, we can create feedforward neural networks with three or fewer hidden layers that predict the components of a microstructure's effective stiffness tensor, $\mathbf{C}_e^H$, given the scalar values of the blending weights and volume fraction as inputs. That is, $\mathbf{C}_e^H = NN(\hat{\mathbf{c}}_e, \hat{v}_e)$, where $\hat{\mathbf{c}}_e$ and $\hat{v}_e$ are the interpolated class and filtered (Sec. 6.6) volume fraction design variables, respectively.

After each hidden layer, we use a tanh activation function. For training, we use the mean squared error (MSE) loss and the Levenberg-Marquardt optimizer [212]. To include all possible microstructures, such as those where only a few basis classes have non-zero weights, we use an optimal sliced Latin hypercube method [213] to sample the weights c_d, first creating combinatorial "slices", then 20 space-filling samples for each "slice". To cover volume fractions, we also sample 15 microstructures from each resulting SDF (i.e., each set of weights) over $t \in [-1, 1]$. As an example, we obtain a total of $22,575$ microstructures for $D = 5$ basis classes. The effective stiffness tensors of each are calculated using an energy-based homogenization method [180]. We set aside $70\%, 15\%, 15\%$ of the data for training, validation and testing.

Once the model is trained, we can use backpropagation [214] to analytically derive the gradients of $\mathbf{C}_e^H$ with respect to the design variables, $\hat{\mathbf{c}}_e$ and $\hat{v}_e$. This, together with the rapid predictions that bypass the cost of homogenization, allows the neural networks to significantly boost the efficiency of design.

6.6. Concurrent Multiclass Data-Driven Topology Optimization

One challenge in creating a concurrent functionally graded design framework that produces realistic results while remaining as general as possible is the different feasible ranges of arbitrary microstructure classes. Consider the five truss basis classes we used to demonstrate the blending scheme in Sec. 6.4. For a prescribed minimum feature size of 4 pixels, the first basis has a minimum feasible volume fraction of 0.2, while the second has a minimum of 0.4. The question that arises is: when different $v_{min} > 0$ are possible for each element e, how can we design the distribution of volume fractions while also optimizing a clearly defined macrostructure where some microstructures are allowed to be void (i.e., $v_e = 0$)?

Regarding this, most existing FGS research have either elected to ignore the macroscale design altogether [134, 215], or adopted a hybridized method that splits the macro- and micro-scale designs into two optimization problems [62, 186, 191, 193]. None of these works incorporate multiclass designs where the basis topologies can be drastically different, however.

We propose to overcome this hurdle by merging parametric and non-parametric methods in a framework that utilizes evolutionary TO (BESO [216]) to optimize the discrete global structure, $\mathbf{x}$, and gradient-based TO solved by the method of moving asymptotes (MMA) [122] to concurrently design the coefficients of M new classes $\mathbf{c}$, their distributions $\boldsymbol{\xi}$, and volume fractions $\mathbf{v}$. The approach is similar to the latter group above, but unlike many, we evolve the designs at both scales in the same iteration. This combination allows arbitrary sets of basis microstructures to be used rather than strict constraints or careful handpicking, and distinguishes our framework in terms of generality and efficiency.

Thus, the general optimization problem can be formulated as:

$$
\begin{aligned}
\underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi},\mathbf{x}}{\text{minimize}} \quad & f = f_{perf}(\mathbf{c},\mathbf{v},\boldsymbol{\xi},\mathbf{x}) + k f_{div}(\mathbf{c}), \\
\text{subject to} \quad & \mathbf{KU} = \mathbf{F}, \\
& g_j \leq 0, \\
& 0 \leq c_d^{(m)} \leq 1, \\
& 0 \leq v_{min} \leq v_e \leq v_{max}, \\
& 0 \leq \xi_e^{(p)} \leq 1, \\
& x_e \in \{x_{min}, 1\}, \\
& j \in [1, N_{con}], \quad e \in [1, N_{el}], \\
& m \in [1, M], \quad d \in [1, D-1], \quad p \in [1, M-1],
\end{aligned}
\tag{6.9}
$$

where f_{perf} is an application-dependent measure of design performance, f_{div} is a penalty on low class diversity (Sec. 6.6.1), N_{con} is the number of constraints g_j (if any), N_{el} is the number of macroscopic elements or microstructures, and M and D are the numbers of new (to be optimized) and basis (fixed) classes, respectively. A small number, $x_{min} = 1\text{e}{-9}$, is used to indicate void microstructures to avoid numerical issues. The minimum volume fraction, v_{min}, is dependent on the chosen set of basis classes, i.e., $\min(volume(\boldsymbol{\Phi}_d^L))$, whereas the upper bound, v_{max}, is 0.95.

We employ the traditional radial averaging filter [86] on our global-level design variables to avoid mesh dependency, resulting in the smoothed fields $\hat{\mathbf{v}}$, $\hat{\boldsymbol{\xi}}^{(p)}$, and $\hat{\mathbf{x}}$. This additionally enforces the interface between optimal classes to be functionally graded, and that the macrostructure has a minimum feature size of r_{min}.

To update the designs, we use the default algorithms for BESO [216] and MMA [122]. The only difference is that, to connect the two scales, the sensitivity number for BESO are dependent on the predicted effective properties of the microstructures. The derivations for this and all other sensitivities in our problem are shown in Sec. 6.6.3.

In total, then, our multiclass FGS design framework has $(D-1)M + (M+1)N_{el}$ variables, where $M \leq D \ll N_{el}$. For the $M = 2$ MBB beam example later (Fig. 6.5a), our method has $1,928$ design variables. Perhaps the most comparable methods are the multilattice approach [63] uses $3,200$ variables for the same problem without optimizing the graded volume fractions, while the latent variable multiclass method [3], which uses $1,920$ variables and includes functionally graded volumes; both, however, require predefined classes with manually defined connections. It is important to note that although it is possible for other methods to contain less design variables, they make simplifications that we do not.

6.6.1. Penalty to Encourage Convergence to Diverse Classes

Our method can enable high design freedom even with a low-dimensional microscale representation since blending allows the basis classes to mix continuously at both scales. Depending on the chosen optimizer, managing such complexity in two-scale design can be a challenge, one that is also encountered by existing multiscale methods. However, we propose that a cost-effective penalty on the objective function can aid the optimizer (in our case, MMA) without resorting to user-defined restrictions on the design space.

We introduce a penalty on low diversity between the M new classes, encouraging the microscale design variables, $\mathbf{c}^{(m)}$, to converge to values away from each other, so that the FGS is more likely to include different basis classes:

$$f_{div} = -\log\left[\det(L_{ij}(\mathbf{c}^{(i)}, \mathbf{c}^{(j)}))\right], \tag{6.10}$$

where $L_{ij} = \exp(-0.5\ \|\mathbf{c}^{(i)} - \mathbf{c}^{(j)}\|_2^2)$ and $i, j \in [1, M]$.

This is based on determinantal point processes (DPPs), which measure the diversity of a set of items (e.g., the classes $\mathbf{c}^{(m)}$ here) using a similarity matrix L_{ij}, where its elements are the similarities between the i-th and j-th pairs of data. The diversity can then be defined as the determinant of L_{ij}. A larger determinant value indicates that a set contains less similar items, spans a larger volume, and hence has greater diversity. DPPs have been successfully applied to create or extract diverse metamaterials, ideation sketches and airfoils from large datasets [152, 153, 196, 217]. More thorough coverage of DPPs, as well as a deeper dive into the benefits of diversity for data-driven multiscale design, can be found in the previous chapter (Chapter 5).

Here we apply the concept of diversity to penalize microstructure classes that are similar to each other. Intuitively, minimizing f_{div} is equivalent to maximizing the diversity of the new classes. Since the value of f_{div} approaches zero as classes become more diverse, i.e., the values of $c_d^{(i)}$ and $c_d^{(j)}$ grow farther apart, diversity serves as a natural penalty function. It needs only a weight k so that its value, typically within $[0, 1]$ after the first few iterations, can compete with the structural performance, f_{perf}. Moreover, it acts similarly to an L_2 regularizer that smooths the objective function as k increases, which may avoid sensitivity to initializations and help find an optimum faster in some non-convex or highly nonlinear problems. Indeed, we find in our case studies that adding the penalty help our optimizers to find more optimal solutions (Sec. 6.7.2).

6.6.2. Volume Relaxation and Adaptive Target Volume

If any volume constraints are defined in the design problem, we must have a way to obtain the continuous gradients of volume with respect to the parameters of the shape blending scheme. To achieve this, we can approximate the filtered volume fraction of a microstructure, $\hat{v}_e$, by transforming its SDF into a relaxed grayscale field, similar to that of density-based TO, using the Sigmoid function

$$S(\mathbf{\Phi}, \beta_1) = \frac{1}{1 + \exp\left(-\beta_1 \mathbf{\Phi}\right)}, \tag{6.11}$$

where β_1 is a fixed parameter to control the strength of relaxation. Thereafter, the approximate volume is

$$\hat{v}_e^a = \frac{1}{n_{el}} \sum_{u=1}^{n_{el}} S(\Phi_{e \cdot u}, \beta_1), \tag{6.12}$$

where n_{el} is the number of elements in the discretized SDF, and the sensitivity of $\partial V_{Global} / \partial \hat{\mathbf{c}}_e$ is straightforward to calculate (see Sec. 6.6.3).

In addition, low volumes are often a goal in multiscale design to take advantage of the porosity of the microstructures. By immediately applying a strict volume fraction constraint, however, it is possible to encounter infeasible and broken structures early in the optimization process [218, 219]. To avoid this, and to ensure that our macro- (BESO) and micro-scale (MMA) designs evolve at approximately the same rate, we use an adaptive scheme to lower the target volumes every 10 iterations. This is outlined in Algorithm **??**, and the pseudo-code of the complete concurrent optimization framework is shown in Algorithm 3 (Appendix A).

6.6.3. Sensitivity Analysis

The derivations of the sensitivity analysis for gradient-based topology optimization (TO) are detailed in this section. All gradients were verified using the finite difference method.

For the global multiclass shape blending scheme (Eq. 6.8), which is used to obtain the microscale topology at e, the gradient is

$$\frac{\partial \mathbf{\Phi}_e}{\partial \hat{c}_{e \cdot d}} = \frac{1}{\beta_2 \mathbf{\Phi}_d^I} \Big[\beta_2 \exp\left(\beta_2 \mathbf{\Phi}_e^0\right) \odot \mathbf{\Phi}_d^* + \frac{\partial H(\hat{c}_{e \cdot d}, \beta_2, \eta_2)}{\partial \hat{c}_{e \cdot d}} \exp\left(\beta_2 \mathbf{\Phi}_d^L\right) \Big], \tag{6.13}$$

where

$$\mathbf{\Phi}_e^I = \exp\left(\beta_2 \mathbf{\Phi}_e^0\right) + \sum_d^D a_d \exp\left(\beta_2 \mathbf{\Phi}_d^L\right), \tag{6.14}$$

$$\mathbf{\Phi}_e^0 = \sum_d^D \hat{c}_{e \cdot d} \mathbf{\Phi}_d^* + t_e, \tag{6.15}$$

$$\frac{\partial H(\hat{c}_{e \cdot d}, \beta_2, \eta_2)}{\partial \hat{c}_{e \cdot d}} = \frac{\beta_2 \left[1 - \tanh^2(\beta_2(\hat{c}_{e \cdot d} - \eta_2))\right]}{\tanh(\beta_2 \eta_2) + \tanh(\beta_2(1 - \eta_2))}, \tag{6.16}$$

and $\odot$ indicates element-wise multiplication.

The sensitivity for the the class interpolation schemes (Eqs. 6.6 and 6.7) with respect to the class design variables is as follows:

$$\frac{\partial \hat{c}_{e\cdot d}}{\partial c_d^{(m)}} = \left\{ \frac{\partial \tilde{c}_d^{(1)}}{\partial \tilde{c}_d^{(m)}} + \sum_{j=1}^{M-1} \left[\frac{\partial\left(\tilde{c}_d^{(j+1)} - \tilde{c}_d^{(j)}\right)}{\partial \tilde{c}_d^{(m)}} \prod_{k=1}^{j} \hat{\xi}_e^{(k)} \right] \right\} \frac{\partial \tilde{c}_d^{(m)}}{\partial c_d^{(m)}}, \tag{6.17}$$

where

$$\frac{\partial \tilde{c}_d^{(m)}}{\partial c_d^{(m)}} = \sum_{j=d}^{D-1} \left[\left(z_d^{(j+1)} - z_d^{(j)} \right) \prod_{k=1, k\neq d}^{j} c_k^{(m)} \right]. \tag{6.18}$$

With respect to the macroscale distribution fields, it is:

$$\frac{\partial \hat{c}_{e\cdot d}}{\partial \xi_e^{(m)}} = \sum_{j=m}^{M-1} \left[\left(\tilde{c}_d^{(j+1)} - \tilde{c}_d^{(j)} \right) \prod_{k=1, k\neq m}^{j} \hat{\xi}_e^{(k)} \right] \frac{\hat{\xi}_e^{(m)}}{\xi_e^{(m)}}. \tag{6.19}$$

We also need the sensitivities of the radial filters. This follows the traditional TO methods closely [86], and are defined by

$$\begin{aligned} \frac{\partial \hat{v}_e}{\partial v_e} &= \frac{w_e}{\sum_{i\in S_e} w_i}, \\ \frac{\partial \hat{\xi}_e^{(m)}}{\partial \xi_e^{(m)}} &= \frac{w_e}{\sum_{i\in S_e} w_i}. \end{aligned} \tag{6.20}$$

S_e is the set of elements neighboring e, and the weighting function w_i is

$$w_i = \max(0, r_{min} - \|(\cdot)_i - (\cdot)_e\|_2), \tag{6.21}$$

where $(\cdot)$ is the appropriate variable, v or $\xi^{(m)}$, and r_{min} is the filter radius.

Using these, we can obtain the other sensitivities. For compliance (Eq. 6.34), the adjoint method [43] and chain rule allow us to derive the following with respect to the design variables:

$$\frac{\partial f_c}{\partial c_{e \cdot d}} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{c}_{e \cdot d}} \mathbf{u}_n\right] \frac{\partial \hat{c}_{e \cdot d}}{\partial c_{e \cdot d}}, \tag{6.22}$$

$$\frac{\partial f_c}{\partial v_e} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{v}_e} \mathbf{u}_n\right] \frac{\partial \hat{v}_e}{\partial v_e}, \tag{6.23}$$

and

$$\frac{\partial f_c}{\partial \xi_e^{(m)}} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{\xi}_e^{(m)}} \mathbf{u}_n\right] \frac{\hat{\xi}_e^{(m)}}{\xi_e^{(m)}}. \tag{6.24}$$

The derivatives of the element effective stiffness matrices are

$$\frac{\partial \mathbf{k}_e}{\partial \hat{c}_{e \cdot d}} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \frac{\partial \mathbf{C}_e}{\partial \hat{c}_{e \cdot d}}, \tag{6.25}$$

$$\frac{\partial \mathbf{k}_e}{\partial \hat{v}_e} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \frac{\partial \mathbf{C}_e}{\partial \hat{v}_e}, \tag{6.26}$$

and

$$\frac{\partial \mathbf{k}_e}{\partial \hat{\xi}_e^{(m)}} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \left[\sum_{d=1}^{D} \frac{\partial \mathbf{C}_e}{\partial \hat{c}_{e \cdot d}} \frac{\partial \hat{c}_{e \cdot d}}{\partial \hat{\xi}_e^{(m)}}\right]. \tag{6.27}$$

Here, due to our data-driven framework, the gradients of the effective properties $\partial \mathbf{C}_e / \partial \hat{c}_{e \cdot d}$ and $\partial \mathbf{C}_e / \partial \hat{v}_e$ are obtained by backpropagating through the layers of the fully connected neural network [214].

To obtain the sensitivities of the global volume fraction constraint, we can use Eq. 6.13 and the Sigmoid function (Eq. 6.11 in Sec. 6.6.2), which gives us a continuous approximation of a volume fraction ($\hat{v}_e^a$ from Eq. 6.12). First, the derivative of the Sigmoid applied to a microstructure's SDF ($\mathbf{\Phi}_e$) is

$$\frac{\partial S(\mathbf{\Phi}_e, \beta_1)}{\partial \mathbf{\Phi}_e} = \beta_1 S(\mathbf{\Phi}_e, \beta_1)\big(1 - S(\mathbf{\Phi}_e, \beta_1)\big). \tag{6.28}$$

Therefore, the sensitivity of the global volume fraction can be decomposed as

$$\begin{aligned} \frac{\partial V_{Global}}{\partial \hat{c}_{e \cdot d}} &= \frac{\partial V_{Global}}{\partial \hat{v}_e^a} \frac{\partial \hat{v}_e^a}{\partial S} \frac{\partial S}{\partial \mathbf{\Phi}_e} \frac{\partial \mathbf{\Phi}_e}{\partial \hat{c}_{e \cdot d}}, \\ &= \frac{x_e}{N_{el}} \left(\frac{1}{n_{el}} \sum_{u=1}^{n_{el}} \frac{\partial S}{\partial \mathbf{\Phi}_e} \right) \frac{\partial \mathbf{\Phi}_e}{\partial \hat{c}_{e \cdot d}}, \end{aligned} \tag{6.29}$$

and the sensitivities of the constraint itself with respect to the design variables are:

$$\frac{\partial g_1}{\partial c_{e \cdot d}} = \frac{1}{V_{Global}^*} \frac{\partial V_{Global}}{\partial \hat{c}_{e \cdot d}} \frac{\partial \hat{c}_{e \cdot d}}{\partial c_{e \cdot d}}, \tag{6.30}$$

$$\frac{\partial g_1}{\partial v_e} = \frac{1}{V_{Global}^*}, \tag{6.31}$$

and

$$\frac{\partial g_1}{\partial \xi_e^{(m)}} = \frac{1}{V_{Global}^*} \frac{\partial V_{Global}}{\partial \hat{c}_{e \cdot d}} \frac{\partial \hat{c}_{e \cdot d}}{\xi_e^{(m)}}. \tag{6.32}$$

We note that these approximations of the microstructural volume fractions do introduce some error into the sensitivities but, in our experience, are minor and not detrimental to the designs.

Since we use the default algorithm for BESO developed by Ref. [216] to update $\mathbf{x}$, the filtering of the macroscale sensitivity numbers for the compliance problems can be found in the original paper. The only difference is that, instead of the element stiffness for a homogeneous material, we use the effective stiffness predicted by the data-driven models. Thus, our sensitivity numbers are modified to be

$$\alpha_e = \mathbf{u}_e^T \mathbf{k}_e^H(\hat{\mathbf{c}}_e, \hat{v}_e)\mathbf{u}_e. \tag{6.33}$$

For the low-diversity penalty function, the sensitivity can be derived using chain rule by taking the gradient of the log-determinant [220], then of the Gaussian kernel and Euclidean distance between the class design variables, $\mathbf{c}^{(m)}$.

Finally, for the derivations of the first stage of the shape matching problem (Eq. 6.35), in which the target properties for each unit cell are optimized, we refer the reader to [1]. For the second stage where the classes and their distributions are optimized to match the target properties (Eq. 6.36), we do not write out the sensitivities here; however, their derivations are straightforward to calculate by following the same steps above, substituting compliance for the mean squared error (MSE) between target and designed effective properties.

6.7. Illustrative Examples

Through several linear elastic problems, we test the ability of our framework to achieve functional grading that is both smooth and feasible. Namely, we design two compliance and two target displacement, i.e., shape morphing, examples. For each case, we study the effects of two sets of basis classes with different morphology types and initial mutual compatibility, as well as the number of new optimal classes, M. In all datasets and problems, the constitutive material has a Young's modulus of 1 and a Poisson's ratio of 0.3. The basis classes and their associated DL models are shown in Sec. 6.7.1, the compliance studies in Sec. 6.7.2, and finally, the shape morphing examples in Sec. 6.7.3.

6.7.1. Basis Classes and Neural Networks

To illustrate the framework across a range of microstructure morphologies commonly found in literature, we use two sets of basis classes: one consisting of truss-type microstructures, and one of topology-optimized freeform shapes. Moreover, to show that human bias can be removed from the design without sacrificing much performance, we compare handpicking the truss basis to automatically selecting the freeform basis using diversity metrics.

6.7.1.1. Handpicked Simple Trusses. Some of the most popular microstructures in the TO field belong to truss-type classes, which possess both simple definitions and satisfactory performance [3, 64, 65, 211]. As such, they are fitting basis classes to validate the proposed framework. For our examples, we choose a set of bases that can, when combined in various ways, cover nearly all of the common truss morphologies in literature. Departing from other methods, we dial up the difficulty by defining the last two bases so that they are broken; to obtain feasible designs, these classes need to be either well-connected to their neighbors or blended with other bases that have self-connectedness. The five classes have already been shown several times in Figs. 6.1-6.3, as well as in Fig. 6.4a and b.

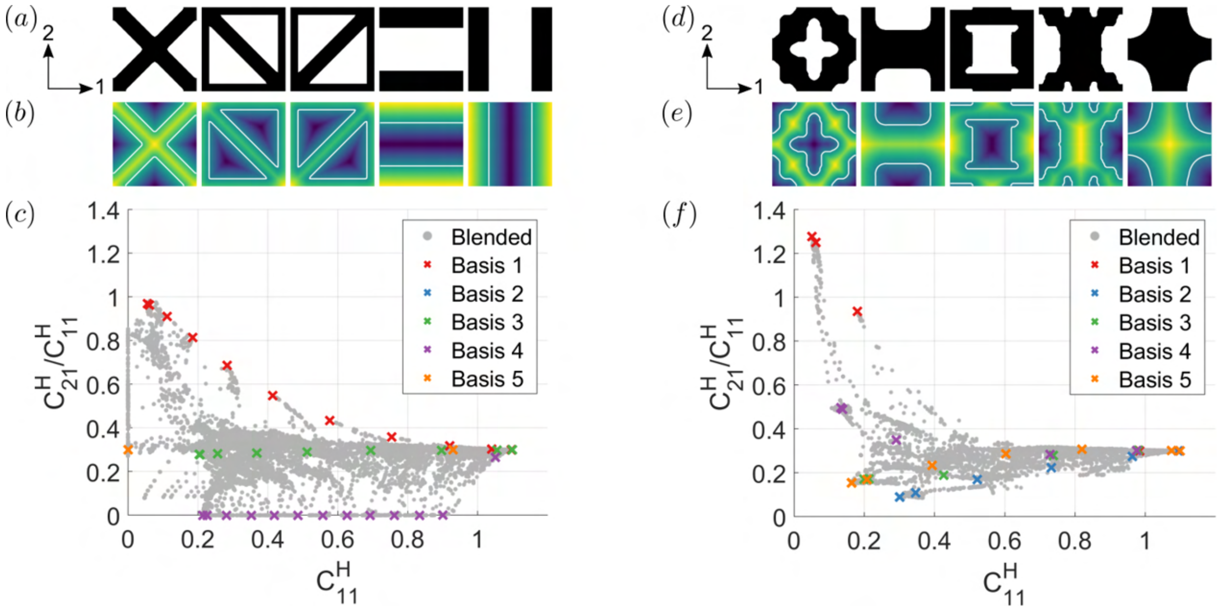


Figure 6.4. (a) Truss basis classes represented as (b) SDFs ($\mathbf{\Phi}^*_d$), and (c) the property space of 22,575 blended microstructures by sampling $\{\mathbf{c}, v\}$. (d) Freeform basis classes, (e) their SDFs, and (f) property space.

6.7.1.2. Shape and Property Diverse Freeform Subsets. We also assess the efficacy of our blending and interpolation schemes under even more challenging circumstances by defining a set of freeform basis classes with complex shapes derived from TO (Fig. 6.4d). They present interesting and highly illustrative case studies as their compatibility with each other is quite low. If used directly in design without our shape blending scheme, the feasibility of the final design would be extremely challenging to guarantee.

We collect these freeform classes from a different perspective, one where a designer has little prior knowledge and wishes to avoid using costly inverse optimization to find the basis microstructures. Thus, the five freeform classes are chosen by leveraging the open-source 2D metamaterials dataset [124] and the automated diverse subset selection method [196] from our previous works. The method utilizes the DPPs introduced earlier

(Sec. 6.6.1) to maximize the shape and property diversity of a subset of microstructures. By automatically covering a wide range of shapes and properties, we hypothesize that diverse basis classes can provide a high return on investment, attaining competent or even superior performance across multiple applications with less effort during the selection of bases.

We filter out any 50×50 microstructures with minimum feature sizes less than 4 pixels prior to applying our subset selection method. This eliminates some of the most complex shapes that provide little benefit for functional grading (e.g., microstructures with thin features that would limit the range of feasible volume fractions). To convert the selected binary microstructures into continuous SDF representations, we use the fast marching method [221]. The shape and property diverse freeform basis classes shown in Fig. 6.4d and e.

6.7.1.3. Property Prediction Models. As discussed in Sec. 6.5, we obtain training datasets of $22,575$ microstructures using sliced Latin hypercube sampling for each set of basis classes. The respective property spaces are depicted in Fig. 6.4c and f, where samples generated from blending are in gray and those from the original basis classes are denoted as different color crosses.

We observe that the blended microstructures are able to interpolate between – and in some cases, extend slightly beyond – basis classes in order to cover the property space. Although some sparse areas still exist due to the lower feasible bounds that we impose, this shows that blending is a powerful technique to create a large design space even with small sets of basis classes. It is also clear from these figures that the first basis (red) from both sets possess the highest stiffness in diagonal directions and ratios of C^H_{21} to C^H_{11}

(equivalent to the effective Poisson's ratio), as opposed to truss bases 4 (purple) and 5 (orange), and freeform basis 2 (blue), which are stiffest in uniaxial directions.

Table 6.1. Neural network architectures and accuracies.

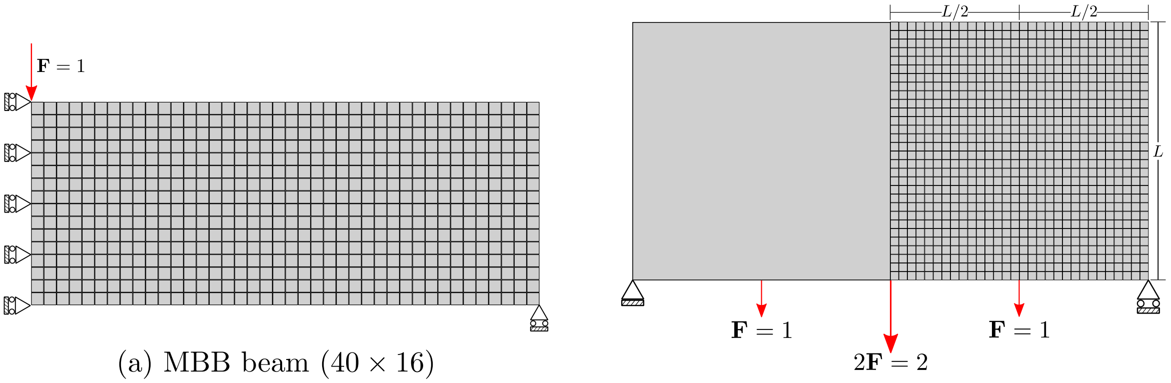
	n_{resp}	n_{node}	Train R^2 (MSE)	Val. R^2 (MSE)	Test R^2 (MSE)
Truss	6	{16,16,12}	0.9983 (2.34e-4)	0.9984 (2.80e-4)	0.9983 (2.46e-4)
Freeform	4	{12,12,6}	0.9991 (2.91e-4)	0.9990 (2.90e-4)	0.9991 (3.01e-4)

While the freeform classes are orthotropic, some of the truss classes are not. The number of responses, n_{resp}, in the neural networks of each set are adjusted accordingly, i.e., 6 elastic tensor components for trusses and 4 for freeform. Table **??** lists the details of the ML model architectures, where n_{node} indicates the neurons of each hidden layer, and the R^2 and MSE metrics of the trained models.

Since they only need to be built once, the same models are used for all examples throughout the paper. The one-time expense of creating our data and models is reasonable for our 2D problems. However, we note that our results show overlapping properties in the dataset (Fig. 6.4c and f) and high R^2 values above 0.99 and low MSE (Table **??**), suggesting that it may not have been necessary to use as many samples as we did. There is great potential to develop adaptive sampling algorithms that better balance accuracy and efficiency, particularly for 3D and complex applications. This direction is left for future works.

6.7.2. Compliance Minimization

We begin with compliance minimization examples, the first of which is the classic MBB beam. The boundary conditions are depicted in 6.5a, and we follow the same set-up as in Refs. [80] and [3] in order to compare our results with those of existing methods. That is, the MBB beam is discretized into 40 × 16 microstructures and an ambitious global volume fraction limit is set as $V^*_{Global} = 0.36$. For the second problem, we pursue a 60×30 bridge structure loaded in three places, as shown in Fig. 6.5b. Due to symmetry, we can cut its size by half into 30 × 30. The target global volume there is $V^*_{Global} = 0.50$.



(a) MBB beam (40 × 16)

(b) Bridge (30 × 30)

Figure 6.5. Problem settings of the compliance minimization examples.

The two compliance problems can be formulated as:

$$
\begin{aligned}
&\underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi},\mathbf{x}}{\text{minimize}} \quad && f = f_c + k f_{div}(\mathbf{c}), \\
&\text{subject to} && \mathbf{K}\mathbf{U} = \mathbf{F}, \\
& && g_1 = V_{Global}/V^*_{Global} - 1 \leq 0, \\
& && g_2 = V_{BESO}/V^*_{BESO} - 1 \leq 0,
\end{aligned}
\tag{6.34}
$$

where $f_c = \sum_{e=1}^{N_{el}} \mathbf{u}_e^T \mathbf{k}_e^H(\hat{\mathbf{c}}_e, \hat{v}_e)\mathbf{u}_e$ is the compliance with element displacements $\mathbf{u}_e$ and effective stiffness matrices $\mathbf{k}_e^H$, which are predicted via the DL models. The global and macroscale volumes are $V_{Global} = \sum_{e=1}^{N_{el}} x_e \hat{v}_e / N_{el}$ and $V_{BESO} = \sum_{e=1}^{N_{el}} x_e / N_{el}$, respectively, and the bounds on the design variables are the same as described in Eq. 6.9. The sensitivities of this optimization problem are detailed in Sec. 6.6.3.

For both examples, we initialize the volume constraints as $V^*_{Global,0} = 0.95, V^*_{BESO,0} = \sqrt{V^*_{Global}}$, the volumes $\mathbf{v}$ as 0.95, the class weights so that $\tilde{\mathbf{c}}^{(m)}$ are the same, and all distribution fields, $\boldsymbol{\xi}^{(p)}$, so that the classes are distributed equally. The penalty parameter (Sec. 6.6.1) is set so that $kf_{div} = 10$ during later iterations. We find that keeping the penalty around this value improves both convergence and design performance.

The fixed parameters in the shape blending scheme are $\beta_1 = 64$ and $\beta_2 = 32$. As we suggested in Sec. 6.4.2, the threshold η_2 is adapts to the design and is equal to the $75th$-percentile of the class weights. The radii of all filters on the design variables are the same, $r_{min} = 3.0$, matching Ref. [80]. For BESO, the evolutionary rate is $ER = 0.05$ in both problems. Otherwise, all other parameters are kept at the default values [122, 216]. Our convergence criteria are when the change in design or the mean change in the objective over 10 iterations are less than 0.01, or when the number of iterations reaches 200. We also use early-stopping if the target V^*_{Global} has been met but the objective has not improved in 20 iterations.

In the following sections, we present our results in figures with the same layout: The left sides illustrate how the optimized classes are created via blending by breaking them down into the individual basis classes. For ease of interpretation, we show $\tilde{c}_d^{(m)}$ from Eq. 6.6, which correspond directly to the weights used during blending (Eq. 6.8), instead of the design variables $c_d^{(m)}$. The right-most sides show the optimal multiclass FGS and their final compliance.

For fair comparison between our results and homogenization-based methods in literature, and between the two types of basis morphologies, all compliance values stated in the main paper are calculated using numerical homogenization. For further validation, Tables 6.5 and 6.6 in Sec. 6.7.2.4 also reports the compliance obtained from the neural networks and fine mesh analysis.

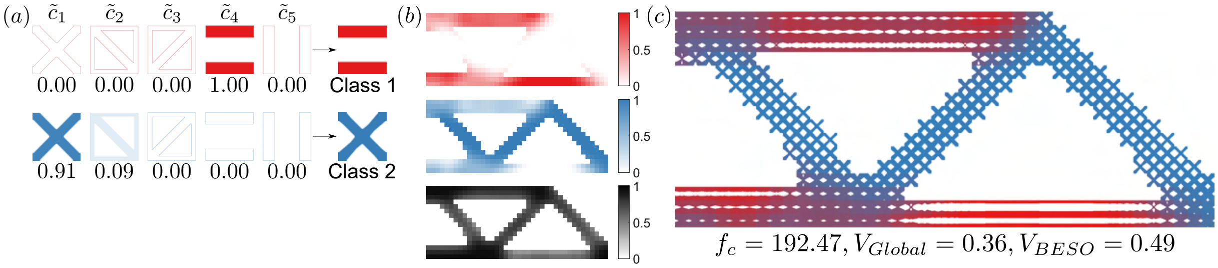


Figure 6.6. Truss MBB, 2-class result. (a) Optimal new classes each drawn in a different color. Left of arrows: optimal weights listed under each basis. Lighter colors indicate low weights while outlined shapes represent weights that are zero. Right of arrows: representative topologies of new classes. (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, and (c) multiclass FGS.

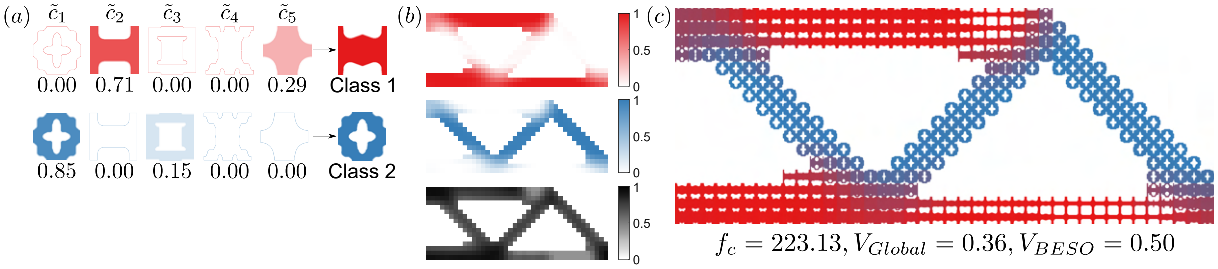


Figure 6.7. Freeform MBB, 2-class result: optimal (a) weights and representatives of new classes, (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, (c) multiclass FGS.

6.7.2.1. 2-Class Results with Different Basis Classes.

We first consider the results using $M = 2$ new classes and both sets of basis classes. The optimal designs are shown in Figs. 6.6 and 6.7 for the MBB beam with truss and freeform bases, respectively, and in Figs. 6.8 and 6.9 for the bridge. The interpretation of the figures is described in the previous section.

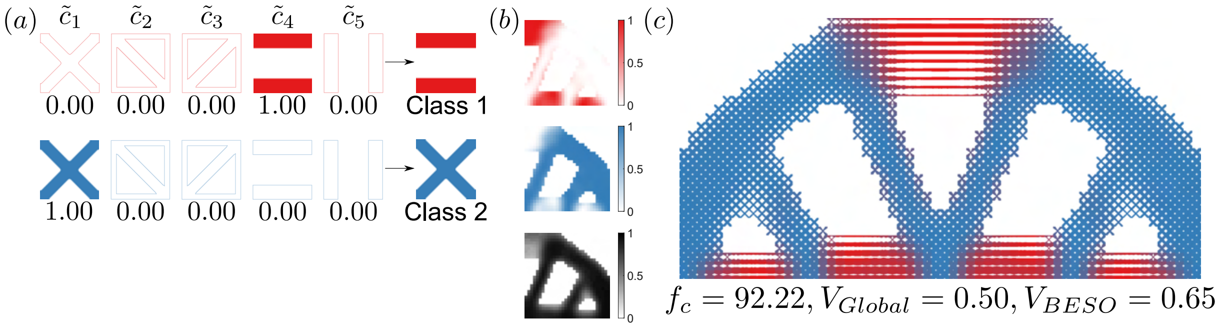


Figure 6.8. Truss bridge, 2-class result: optimal (a) weights and representatives of new classes, (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, (c) multiclass FGS reflected over the symmetry line.

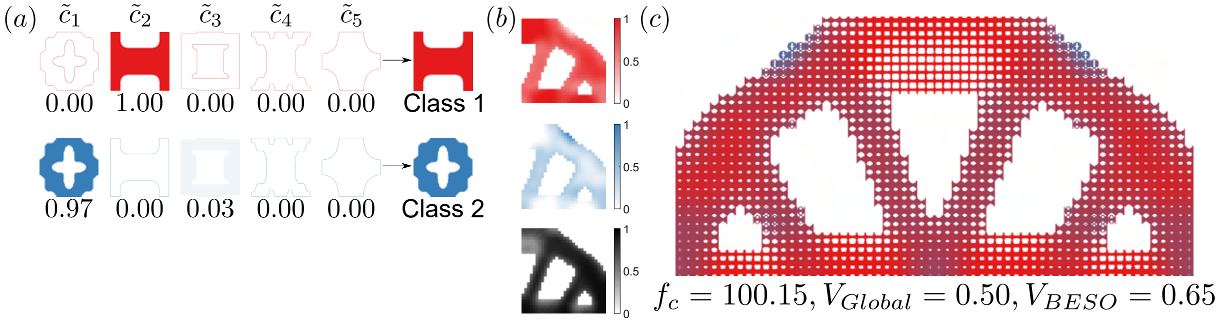


Figure 6.9. Freeform bridge, 2-class result: optimal (a) weights and representatives of new classes, (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, (c) multiclass FGS reflected over the symmetry line.

From the 2-class compliance results, we can see several benefits of integrating multiclass shape blending into FGS design:

(1) The combination of the blending scheme and the radial filters on the distribution fields creates smooth transitions between classes. Topological functional grading is guaranteed and does not depend on the mutual compatibility of the basis classes. Although connections may not be ideal for our freeform bases, which have

low initial connectivity and more complex features, neighboring microstructures change continuously and are at least connected through the imposed lower feasible bounds (Fig. 6.7).

(2) Because of the two-step blending scheme, the microstructures at the interfaces of optimized classes are a union of the classes being mixed there, and the minimum feature sizes of all microstructures match our prescribed lower limit of 4 pixels.

(3) The macroscale distributions, $\hat{\boldsymbol{\xi}}^{(p)}$, can be either dominated solely by one class or contain mixtures of multiple classes. For example, the diagonal struts in the truss MBB (Fig. 6.6a) consist predominantly of the second new class (blue), while the horizontal bars contain both (red and blue), presumably to stiffen the design at those locations. On the contrary, the two new classes in the freeform bridge intermingle throughout nearly the entire structure (Fig. 6.9).

(4) Optimizing $c_d^{(m)}$ can automatically determine if an existing basis class is sufficient to achieve low compliance, or if a novel class needs to be created by fusing several bases. For example, in the freeform MBB result (Fig. 6.7), the mixture of the second and fifth bases stiffens the microstructures, improving the global compliance of the FGS.

(5) Due to BESO, the global macrostructures are clearly defined and change based on the basis classes and optimal microstructures, showing that the hybrid framework works well.

Further observations can be made regarding the framework's ability to adapt to spatially-varying stress distributions. The first and fourth truss-type bases, and the first and second for freeform, are the most popular classes by far, agreeing with our observation in Sec. 6.7.1.3 that these are among the strongest classes in diagonal and uniaxial directions. In both beam and bridge examples, these classes are designed such that the load-bearing features of the blended microstructures intuitively match the load paths.

In particular, our truss-type MBB beam result is akin to those in existing multiscale works with the same design domain. Ref. [80] performed an exhaustive two-scale TO that optimized every microstructure, resulting in horizontal (uniaxial) and diagonal (anisotropic) features that are oriented with stress directions, and a compliance of $f_c = 190$. Meanwhile, Ref. [3] proposed a multiclass design with rectangular trusses on the horizontal macro-bars, X's on the diagonal macro-struts, and a compliance of $f_c = 214.02$. Our framework can be thought of as bridging these two methods. This is indeed reflected in our 2-class truss result, which achieves a compliance value between the two existing works, $f_c = 192.47$, and has similar microstructures.

In terms of performance, truss basis classes outshine the freeform basis in both problems. We theorize that our freeform bases do not perform as well for two reasons. (1) They were automatically chosen to maximize diversity, i.e., coverage, in shapes and properties, which undoubtedly can skip microstructures with properties that are more optimal for these specific problems. (2) They contain complex, thin features that force their lower feasible bounds to have high volume fractions ($v_{min} = 0.28$), which clashes with the low target volume of 0.36 in the MBB problem.

Despite these disadvantages, however, the compliance attained by using freeform basis classes is decent across both examples and near those of existing works, confirming the versatility of diverse bases and our design framework. A deeper look into the MBB

Table 6.2. Single-class MBB beam results using one freeform basis class each.

Basis	1	2	3	4	5
f_c	476.26	247.57	384.89	399.29	290.75

example also reveals that the multiclass freeform FGS (Fig. 6.7) still surpasses single-class designs that vary only in volume fraction (Table 6.2).

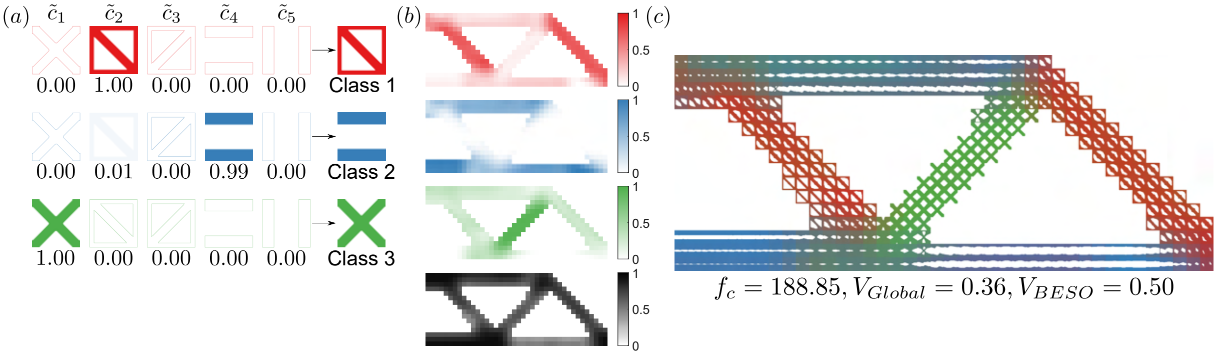

(a)
$\tilde{c}_1$ $\tilde{c}_2$ $\tilde{c}_3$ $\tilde{c}_4$ $\tilde{c}_5$
0.00 1.00 0.00 0.00 0.00 Class 1
0.00 0.01 0.00 0.99 0.00 Class 2
1.00 0.00 0.00 0.00 0.00 Class 3
(b)
1 0.5 0
(c)
$f_c = 188.85, V_{Global} = 0.36, V_{BESO} = 0.50$

Figure 6.10. Truss MBB, 3-class result: optimal (a) weights and representatives of new classes, (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\boldsymbol{\xi}}^{(3)}, \hat{\mathbf{v}}$ from top to bottom, (c) multiclass FGS.

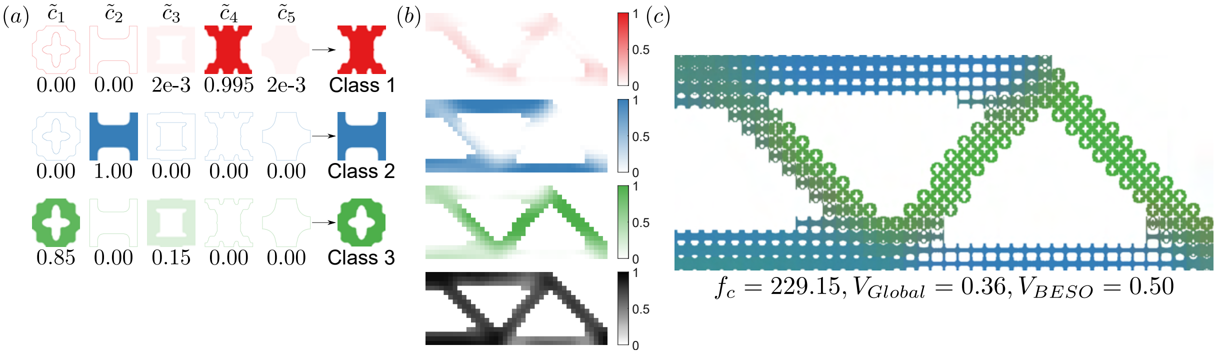
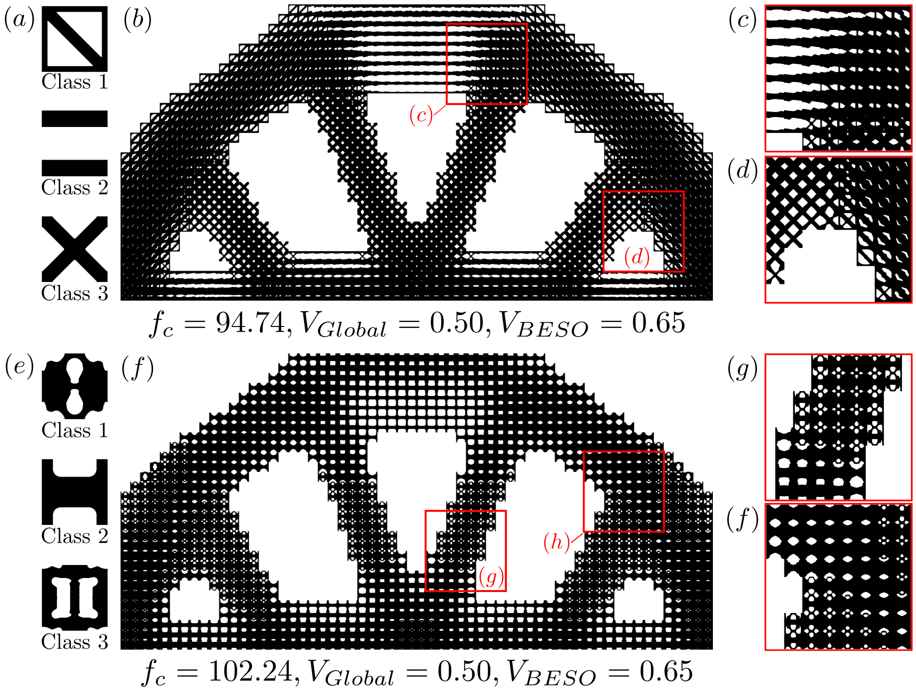

(a)
$\tilde{c}_1$ $\tilde{c}_2$ $\tilde{c}_3$ $\tilde{c}_4$ $\tilde{c}_5$
0.00 0.00 2e-3 0.995 2e-3 Class 1
0.00 1.00 0.00 0.00 0.00 Class 2
0.85 0.00 0.15 0.00 0.00 Class 3
(b)
1 0.5 0
(c)
$f_c = 229.15, V_{Global} = 0.36, V_{BESO} = 0.50$

Figure 6.11. Freeform MBB, 3-class result: optimal (a) weights and representatives of new classes, (b) $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\boldsymbol{\xi}}^{(3)}, \hat{\mathbf{v}}$ from top to bottom, (c) multiclass FGS.

6.7.2.2. Effect of the Number of New Classes. With the effectiveness of the proposed framework established, we now study whether increasing the number of new classes to $M = 3$ can impact the designs and their performances. The problem definitions remain the same as before. For the MBB beam, the 3-class results are given in Figs. 6.10 and 6.11 with the truss and freeform bases, respectively. The bridge results for both sets of bases are combined in Fig. 6.12, where they are shown in black-and-white and with zoomed-in views of the functionally graded topologies and volume fractions.



$f_c = 94.74, V_{Global} = 0.50, V_{BESO} = 0.65$

$f_c = 102.24, V_{Global} = 0.50, V_{BESO} = 0.65$

Figure 6.12. Results of 3-class bridge in black-and-white. For truss bases: (a) optimal classes, (b) FGS, and (c,d) zoom-in views. For freeform bases: (e) optimal classes, (f) FGS, and (g,h) zoom-in views.

The most notable result is the 3-class truss MBB, which achieves a compliance even lower than the fully optimized design of Xia et al. at $f_c = 188.85$. More apparent than in the 2-class result above, it exhibits directional load-bearing features (Fig. 6.10) such as the left-to-right diagonal microstructures (red) and the near-uniaxial microstructures (blue). In addition, the X-shaped class (green) appears mainly in the middle macro-strut. Overall, our result distinguishes itself from those in literature with the mix of isotropic and anisotropic microstructures that are all well-connected. A similar blend of directional and uniaxial classes can be found in our 3-class truss bridge (Fig. 6.12a-d).

With the exception of the truss-type MBB, however, all 3-class results could not overtake the compliance of their 2-class counterparts. We suspect this is because, for simple compliance problems, additional classes are not necessary to achieve optimal performance. Our hypothesis is supported by the new classes of the $M = 2$ examples: for the most part, they are each monopolized by just one basis class. This suggests that, in most cases, only two basis classes are needed throughout the entire FGS; if more are required, they can be incorporated into a single optimal class by adjusting the values of $\mathbf{c}^{(m)}$ without increasing M, like in the 2-class freeform MBB (Fig. 6.7). Another reason could be that we force the third class to be different from the others through the penalty on low diversity, which can lead to the addition of a sub-optimal class. This scenario may have occurred in the 3-class freeform MBB, where the first new class (red) is hardly present in the FGS. Nevertheless, the penalty does not significantly worsen the compliance and can, in some cases improve it. An intriguing possibility that bears further investigation is whether a larger variety of microstructures are needed in problems with finer discretization or more complex objectives.

Finally, we compare the computational efficiency of our proposed framework against others, with the caveat that each method was run on different computers. The 40×16 MBB design is reported to require 200 hours in [80], and 5 minutes in the data-driven

Table 6.3. MBB beam results without penalty on low class diversity ($k = 0$). The dominant weights of each new class, $\tilde{\mathbf{c}}_m$, are in **bold**.

Basis	M	$\tilde{\mathbf{c}}_m$	f_c	V_{Global}	V_{BESO}
Truss	2	$\tilde{\mathbf{c}}_1 = [0.00, \mathbf{0.39}, 0.00, 0.27, 0.33]$ $\tilde{\mathbf{c}}_2 = [0.00, 0.05, 0.11, \mathbf{0.84}, 0.00]$	205.77	0.36	0.50
	3	$\tilde{\mathbf{c}}_1 = [0.00, \mathbf{0.36}, 0.22, 0.12, 0.30]$ $\tilde{\mathbf{c}}_2 = [0.12, 0.12, 0.09, \mathbf{0.66}, 0.01]$ $\tilde{\mathbf{c}}_3 = [0.23, 0.08, 0.16, \mathbf{0.53}, 0.01]$	212.83	0.36	0.54
Freeform	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.09, 0.00, 0.00, \mathbf{0.91}]$ $\tilde{\mathbf{c}}_2 = [0.04, 0.14, 0.39, 0.01, \mathbf{0.43}]$	252.02	0.36	0.51
	3	$\tilde{\mathbf{c}}_1 = [0.01, 0.00, \mathbf{0.53}, 0.00, 0.46]$ $\tilde{\mathbf{c}}_2 = [0.00, 0.45, 0.00, 0.00, \mathbf{0.55}]$ $\tilde{\mathbf{c}}_3 = [0.00, 0.00, 0.00, 0.00, \mathbf{1.00}]$	248.84	0.36	0.50

method of [3]. For our proposed method, the same design using $M = 3$ new classes takes under 12 minutes. However, we note that the majority of this time is consumed by the bisection algorithm (Sec. 6.4.2), which ensures that the microstructures have the optimized volume fractions when converted from SDFs (Eq. 6.11). This aspect of our blending scheme could be improved in future works.

6.7.2.3. Effect of the Low-Diversity Penalty. The function that penalizes new classes with low diversity (Eq. 6.10, Sec. 6.10) can affect performance, although whether that effect is positive or negative depends on the problem or basis classes. In this section, we show concrete examples why the penalty is still recommended by running the same compliance problems without the penalty, i.e., by setting $k = 0$. The results are listed in Tables 6.3 and 6.4. Immediately, we can see that although the truss basis classes can still achieve satisfactory compliance values lower than the existing baselines [3, 80], none of these can beat our results above.

Table 6.4. Bridge results without penalty on low diversity ($k = 0$). The dominant weights of each new class, $\tilde{\mathbf{c}}_m$, are in **bold**.

Basis	M	$\tilde{\mathbf{c}}_m$	f_c	V_{Global}	V_{BESO}
Truss	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.22, 0.00, \mathbf{0.73}, 0.05]$	96.59	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.41, 0.00, 0.09, \mathbf{0.51}]$			
	3	$\tilde{\mathbf{c}}_1 = [0.00, 0.18, 0.00, \mathbf{0.82}, 0.00]$	96.34	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.28, 0.00, 0.21, \mathbf{0.51}]$			
		$\tilde{\mathbf{c}}_3 = [0.00, 0.45, 0.00, 0.00, \mathbf{0.54}]$			
Freeform	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.00, 0.13, \mathbf{0.87}, 0.00]$	105.90	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.03, 0.00, 0.06, \mathbf{0.91}, 0.00]$			
	3	$\tilde{\mathbf{c}}_1 = [0.00, 0.00, 0.10, \mathbf{0.90}, 0.00]$	105.46	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.00, 0.09, \mathbf{0.91}, 0.00]$			
		$\tilde{\mathbf{c}}_3 = [0.02, 0.00, 0.02, \mathbf{0.96}, 0.00]$			

In the tables, we write the highest weight values of each new class, $\tilde{\mathbf{c}}_m$, in bold. From this, we observe that each result is often overshadowed by one basis class (see the bold values in the same column). For the 3-class designs in particular, the second and third new classes are always dominated by the same basis, confirming our earlier suspicion that $M = 2$ is enough to produce optimal results. We also note that there are numerous low values of $\tilde{\mathbf{c}}_m$, signifying that multiple basis classes are being blended into the FGS without improving the design performance. These results additionally imply that greater diversity amongst the microstructure classes improves performance. A more meticulous study on the impact of diversity on the generality and performance of design methods is an intriguing path for future works.

Furthermore, it takes significantly longer for the class design variables to converge without penalization. In the 2-class truss bridge example without the penalty, they often fluctuate and need more than 100 iterations to start converging, whereas they are already converged in under 30 iterations with penalization. These studies validate the benefits that our proposed low-diversity penalty function supply to MMA, helping it to stabilize, escape local minima, and find more optimal solutions. It is possible that another optimizer, such as globally-convergent MMA [222], could alleviate these issues but such an investigation is outside the current scope of this work.

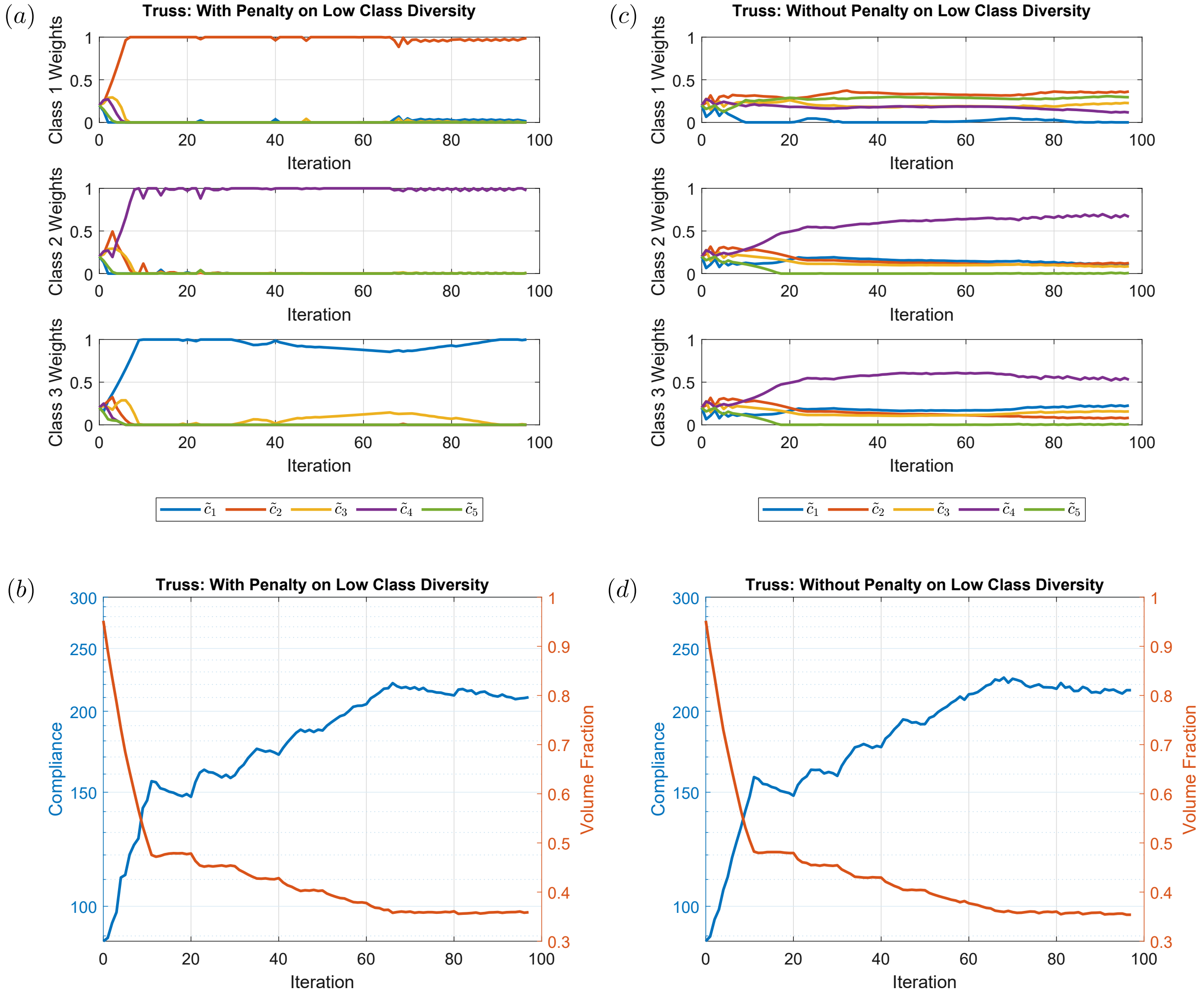


Figure 6.13. For the 3-class truss MBB example: Convergence plots of the class design variables, objective and global volume fraction with the proposed penalty (a,b) and without (c,d).

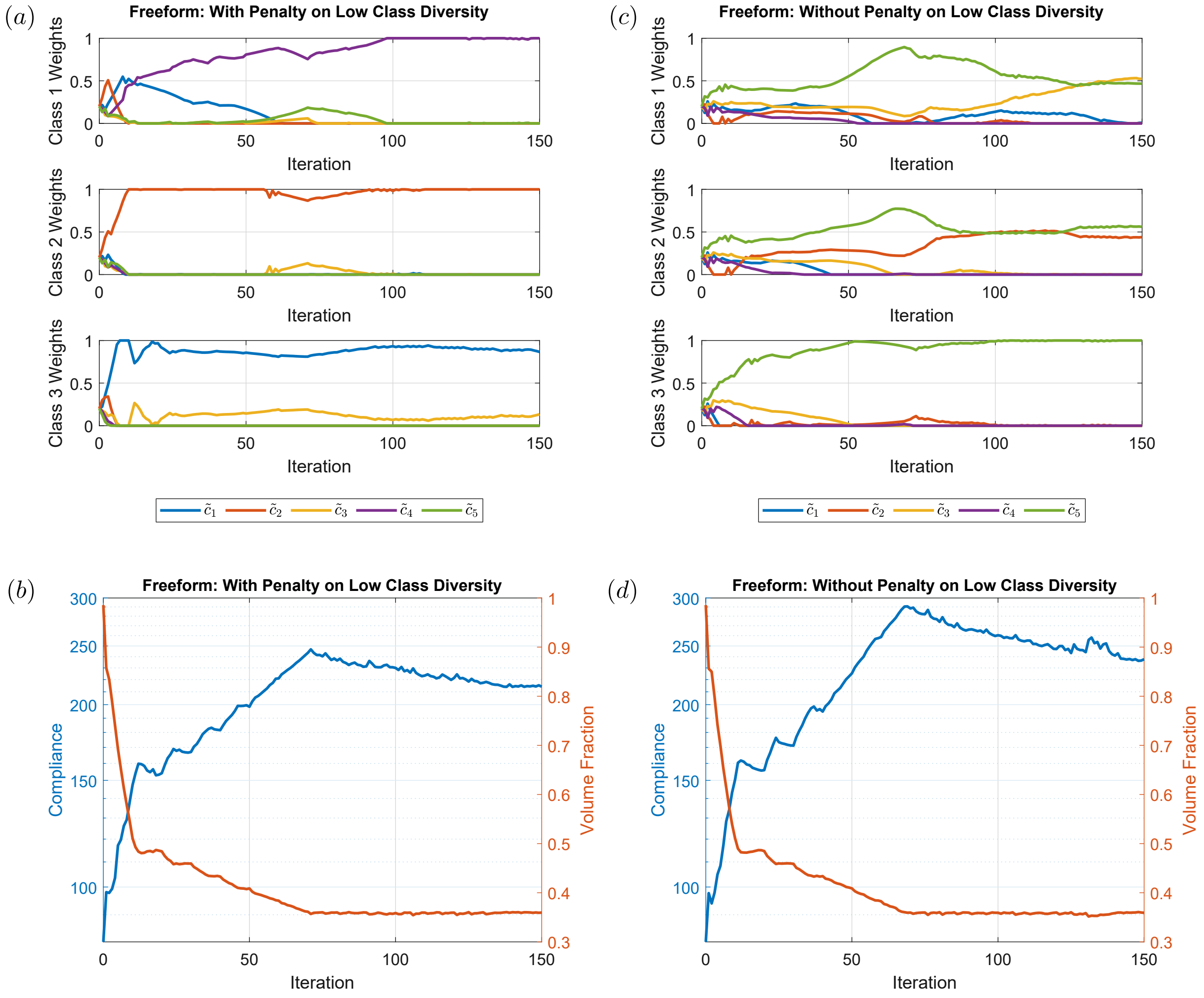


Figure 6.14. For the 3-class freeform MBB example: Convergence plots of the class design variables, objective and global volume fraction with the proposed penalty (a,b) and without (c,d).

To demonstrate the effect of the low-diversity penalty (Sec. 6.6.1) on convergence, we compare the 3-class MBB examples using both the freeform and truss basis classes (Sec. 6.7.2.2). This problem had the most difficulty converging to low compliance values. The histories of the class design variables (which are the only variables used to compute the penalty), as well as the compliance objective and global volume constraint, are plotted in Figs. 6.13 and 6.14.

The penalty on low class diversity has a significant impact on the convergence behavior of the class design variables. They confirm our statement above that applying the penalty can help the weights of the class design variables to converge more quickly, as well as alleviate possible local minima (e.g., compare Figs. 6.13a and c). For the objective values, we first note that the fluctuations here are mainly due to our volume lowering scheme and the use of BESO, which is known to exhibit this behavior [216]. From these plots, we observe that the penalty does not greatly impact the convergence overall but does help reach lower compliance values.

6.7.2.4. Compliance Validation. In the previous sections, we presented the compliance as calculated by numerical homogenization to directly compare our results to those in literature. In Tables 6.5 and 6.6, we additionally report the compliance of our designs performed using: (1) neural networks (same as during design), (2) numerical homogenization (as shown in the main paper), and (3) fine mesh analysis using the multigrid method [223] (for validation).

For MBB in particular, the homogenization-based compliance agree reasonably well with the fine mesh analysis. In general, the numerical homogenization-based results, $f_{c,Hom}$, are slightly closer to the fine analysis values, $f_{c,Fine}$, than the ones using the neural networks, $f_{c,NN}$, which is expected. However, the differences between $f_{c,NN}$ and $f_{c,Hom}$ are not large, validating the use of our predictive models to accelerate the design process.

Table 6.5. Comparison of the compliance of our MBB results in Sec. 6.7.2.1 calculated by: neural network models $f_{c,NN}$, numerical homogenization $f_{c,Hom}$, fine mesh analysis $f_{c,Fine}$. For the homogenization-based values, we also report the percent error from $f_{c,Fine}$ in parentheses.

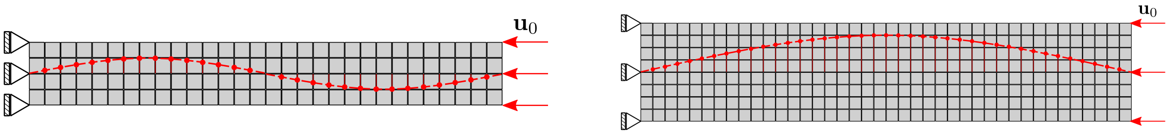
Basis	M	$f_{c,NN}$	$f_{c,Hom}$	$f_{c,Fine}$
Truss	2	187.41 (-13.44%)	192.47 (-11.10%)	216.50
	3	199.74 (-4.56%)	188.85 (-9.76%)	209.28
Freeform	2	210.52 (-20.64%)	223.13 (-15.89%)	265.28
	3	212.68 (-22.55%)	229.15 (-16.55%)	274.60

Table 6.6. Compliance of our bridge results in Sec. 6.7.2.2 calculated three different ways, including the percent error from $f_{c,Fine}$ in parentheses.

Basis	M	$f_{c,NN}$	$f_{c,Hom}$	$f_{c,Fine}$
Truss	2	90.00 (-34.92%)	92.22 (-33.31%)	138.29
	3	92.59 (-31.70%)	94.74 (-30.12%)	135.57
Freeform	2	95.65 (-33.58%)	100.15 (-30.46%)	144.01
	3	93.12 (-36.51%)	102.24 (-30.29%)	146.66

6.7.3. Shape Matching

Heterogeneous structures show great potential for applications where a specific deformation pattern is desired upon actuation, e.g., in form-fitting wearables and soft robots [28, 30, 37]. Motivated by these applications, we optimize two shape matching structures: (1) the target sine-wave deformation profile shown in Fig. 6.15a, and (2) the bump profile in Fig. 6.15b. The first is a cantilever discretized into 30×4 microstructures, fixed at its left side and loaded with displacement boundary conditions on the right – the same example we tested in [196]. The second is similar, but discretized into 40×8. Like the compliance examples, we will also use the truss and freeform basis classes with $M = \{2, 3\}$.



(a) Target sine wave (30 × 4)

(b) Target bump profile (40 × 8)

Figure 6.15. Problem settings of the shape matching examples.

In Chapter 5 and past works [1, 196], we found that these target displacement problems are similar to compliant mechanism design and most effectively solved via a two-stage "top-down" approach that first utilizes inverse TO to find the target effective properties for each microstructure. Departing from previous works, we use the proposed multiclass blending framework in the second stage to optimize the new classes, $\mathbf{c}^{(m)}$, their distributions, $\boldsymbol{\xi}^{(p)}$, and the volume fractions, $\mathbf{v}$, until the target properties are matched.

To find the target properties, i.e., the effective stiffness matrices $\mathbf{C}_t^H$, that achieve a desired displacement profile, the first stage follows the method in Ref. [124] with following problem:

$$\begin{aligned} \underset{\mathbf{C}_t^H}{\text{minimize}} \quad & \frac{1}{n}\|\mathbf{u}-\mathbf{u}_t\|_2^2, \\ \text{subject to} \quad & \mathbf{KU}=\mathbf{F}, \\ & -\phi(\mathbf{C}_t^H) \leq 0, \end{aligned} \tag{6.35}$$

where $\mathbf{u}$ is the displacement vector of n nodes located on the horizontal centerline of the structure, $\mathbf{u}_t$ is the vector of target displacements of the same nodes, $\mathbf{K}$ is the global stiffness matrix, and $\mathbf{U}$ and $\mathbf{F}$ are global displacement and loading vectors, respectively. This inverse problem uses the stiffness matrices of each microstructure as design variables. To ensure that these are within the bounds attainable by shape blending, they are constrained by the signed L_2 distance field ϕ of the properties of the training data for the neural networks (Sec. 6.5 and Fig. 6.4c and f).

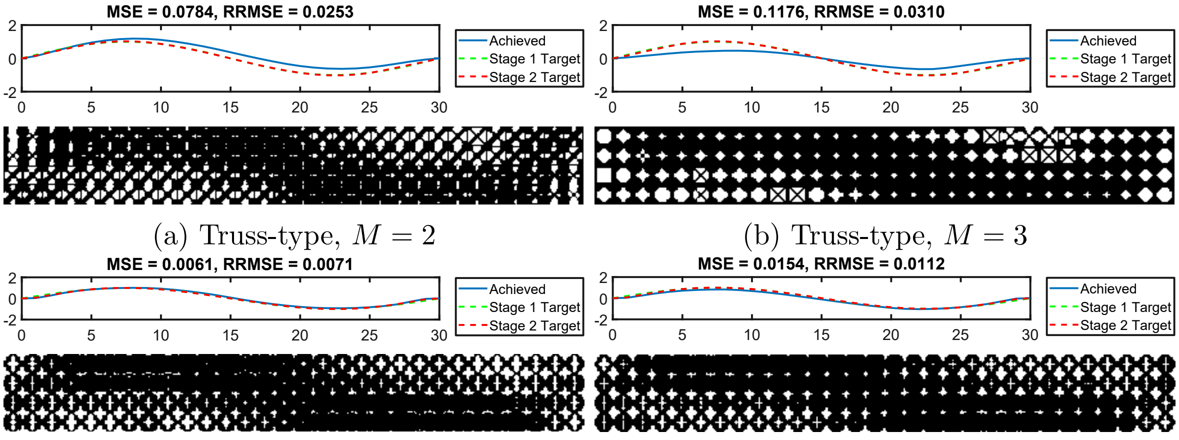
After this, the multiclass FGS is optimized to meet the effective property targets by leveraging our proposed blending scheme. Since we do not aim for a target volume here, there is no global volume constraint and the macrostructure defined by $\mathbf{x}$ remains fixed. The second stage of optimization is thus:

$$\begin{aligned} &\underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{N_{el}}\|\mathbf{C}^H(\mathbf{c},\mathbf{v},\boldsymbol{\xi}) - \mathbf{C}_t^H\|_2^2 + kf_{div}(\mathbf{c}), \\ &\text{subject to} \quad \mathbf{KU} = \mathbf{F}, \end{aligned} \tag{6.36}$$

where the bounds on the design variables are the same as in the previous examples. Due to the omission of $\mathbf{x}$, it can be solved with just MMA. For the 2-class examples, the penalization parameter is set such that $kf_{div} = 4$ in the final iterations, while it is $kf_{div} = 9$ for the 3-class studies. The filter radii are $r_{min} = 1.2$ and $r_{min} = 2.5$ for the sine and bump problems, respectively. The volume fractions are initialized at $\mathbf{v} = 0.5$ in every case, and all other parameters are the same as in the compliance examples.

The final multiclass FGS and their achieved displacement profiles (solid blue) are gathered in Fig. 6.16 for the sine wave and Fig. 6.17 for the bump problem. In the plots of the displacements, we also show the initial target profile, $\mathbf{u}_t$, used in stage one in dashed green lines, along with profile realized by the optimized properties (dashed red), which serves as an indirect target profile in the second stage.

We performed the same sine wave study in Chapter 5, but with a combinatorial method for aperiodic designs, i.e., without functional grading. There, the lowest MSE that we achieved was 0.1146, which most of our proposed multiclass FGS surpass. Interestingly,



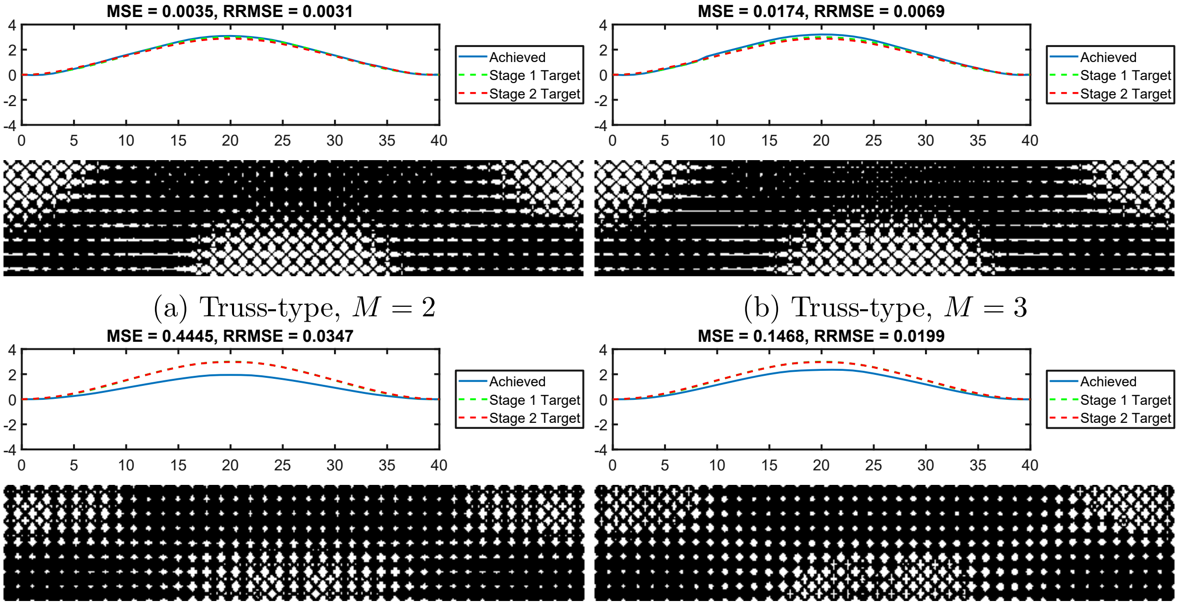
(a) Truss-type, $M = 2$

(b) Truss-type, $M = 3$

(c) Freeform, $M = 2$

(d) Freeform, $M = 3$

Figure 6.16. Results of the target sine wave problem.



(a) Truss-type, $M = 2$

(b) Truss-type, $M = 3$

(c) Freeform, $M = 2$

(d) Freeform, $M = 3$

Figure 6.17. Results of the target bump shape example.

the freeform basis classes perform considerably better than the truss-types. The freeform designs are composed mostly of the first basis, which we noted in Sec. 6.7.1.3 has one of the greatest effective Poisson's ratios, as well as the fifth freeform basis, which has both high stiffness and medium Poisson's ratio.

Conversely, the truss-type FGS match the target bump profile more closely than the freeform ones by utilizing the fourth (horizontal) truss basis. By inspecting the target properties for each problem, we find that the sine wave requires middling values of both Poisson's ratio and stiffness throughout the FGS, which the freeform classes provide more easily, whereas the bump profile needs distinct regions of either large x-directional stiffness or high Poisson's ratio, which the first and fourth truss classes meet exceptionally well (see Fig. 6.4c and f). This observation portends a possible extension of our work where the most efficient basis classes can be chosen to match the distribution of principal macroscale stresses for specific problems, similar to the clustering methods in Refs. [224, 225].

Another intriguing note is that, by blending the last two truss basis classes, we can form square microstructures that are not found in the original set (Fig. 6.16b). Moreover, combining those two with the 'X' basis creates microstructures with star-shaped voids that are not strictly trusses. These are direct results of the weighted sum of SDFs in our proposed blending scheme (Eq. 6.3), which can non-intuitively morph the basis classes to achieve optimal performance.

6.8. Discussion

Our examples exemplify the advantages that multiclass shape blending can bring to functionally graded design, but there are some finer points of discussion to unravel. The first is whether the blending scheme, as we have formulated it, is viable. In particular, we address whether the first step of our scheme is justified, since it can lead to complex or broken microstructures, and whether the second step, which requires pre-processing and could accumulate additional cost, is scalable to high numbers of basis classes, D, and other feasibility metrics.

Regarding the first step (Eq. 6.3), it is undeniable that a naive weighted sum can compound the complexity of the features in the blended microstructure, such as some low quality features like jagged edges or small holes. In our experience, this can be avoided in several ways. (1) Higher resolution can be employed at the microscale, which does not affect the dimensionality of the blending parameters. We use a relatively low 50×50 pixels since this is the native resolution of the open-source freeform dataset [124]. (2) The basis classes could be simplified to suit the user's desires. This is also an advantage afforded by defining fixed bases, so that they can be swapped as needed. (3) Alternatives to the weighted sum are possible, and we have tested numerous. For example, the lower bounds, $\mathbf{\Phi}_d^L$, in the soft-max union (Eq. 6.4) can be replaced with the weighted SDFs, $\tilde{c}_d \mathbf{\Phi}_d^*$. In fact, we found that the weighted soft-max generates clean microstructures that better preserve the original morphologies of the basis classes, as it is essentially a union rather than an interpolation. However, it reduces the achievable shape and property spaces significantly. The simplicity, generality for different design problems, and ability to produce more diverse microstructures is why we elected for the weighted sum presented in this work; there is much room, however, to explore the benefits of alternatives in future works.

During the second step (Eq. 6.4), any costs from obtaining the lower feasible bounds of each basis class, which only needs to be done once, and performing the soft-max union are minimal compared to evaluating the global performance. Therefore, the number of basis classes, D, does not impede efficiency. However, we do observe two bottlenecks in our current implementation: estimating the isovalue, t_e, of each microstructure to match its given volume fraction, v_e, and generating training datasets. For the former, we theorize that the mapping between isovalue and volume is simple enough [36] to be surrogated by a data-driven model, e.g., incorporated into our neural networks as a response, $[\mathbf{C}_e^H, t_e] = NN(\hat{\mathbf{c}}_e, \hat{v}_e)$. This challenge could also be avoided by optimizing the isovalue field, $\mathbf{t}$, instead of $\mathbf{v}$. To answer the latter challenge of the scalability of ML with respect to D, we plan to explore more efficient, adaptive data sampling techniques in the future, such as a task-aware framework [226].

A final question is whether multiclass blending truly creates smooth interfaces between neighboring microstructures. With our current method, we focus on simplicity and ease of use, and can guarantee connectivity but not perfect interfaces. This could be mollified by employing our current framework to problems with fine discretization (which is likely in realistic applications) so that larger radii can be used in the smoothing filters (Sec. 6.6). This also raises an interesting application of our blending method: upscaling a low-resolution design to a finer one. Moreover, there is great potential for further improvement without drastically changing our methods, such as using alternative smoothing filters, extending nonlinear interpolation schemes (to replace Eqs. 6.6 and 6.7), or constraining the distribution fields to change gradually. Nevertheless, we believe that the formulations proposed herein achieve a balance of simplicity, creativity in terms of microstructure designs, and high performance, as validated by the illustrative examples.

6.9. Conclusions

In this chapter, we proposed a novel multiclass shape blending scheme that provides a low-dimensional representation of microstructures for both design and DL, and a data-driven multiscale design method that utilizes a hybrid of TO algorithms along with a new penalty on low diversity designs. By integrating these, we created a multiclass FGS design framework that encapsulates the freedom of fully aperiodic structures while featuring efficiency greater than that of typical multiscale methods. The key is the ability of shape blending to blur the lines between classes, creating graded designs with novel microstructures beyond the initial basis classes. Even with classes that are not mutually compatible or have complex features, continuous transitions between neighboring microstructures are guaranteed.

Furthermore, feasibility constraints are incorporated into the scheme to ensure that they are naturally met. In this work, we use a simple measure – minimum feature size. However, defining the lower feasible bounds of each basis class outside of the optimization process means there is potential for future works to incorporate other feasibility – or even quality – metrics, such as those without cheap or tractable gradients.

We demonstrated these advantages through compliance and shape matching examples, in which blending empowered our FGS to surpass designs in literature. Our results revealed that truss-type classes consistently achieve low compliance, and that diverse freeform classes reach satisfactory performance across multiple applications despite being automatically chosen without considering their compatibility. We also discovered that more is not always better when it comes to classes. By encouraging the design to converge to a smaller number of diverse classes, as few as two can be blended to obtain optimal designs. This outcome merits deeper exploration in the future on how diversity metrics can benefit structural design.

Our framework is general in that it is not tied to the specific DL and TO methods shown in this work. It is also not limited to our 2D classes, since multiclass blending is independent of the topology, representation, dimension and resolution of the basis shapes. This modularity is an especially welcome feature as more advanced prediction models and TO algorithms emerge to solve complex multiphysics and nonlinear mechanics problems, including 3D ones. Beyond the examples presented, our framework can be extended to sought-after functionalities like thermo-elasticity, fracture resistance and energy absorption, and adapted to applications such as customized user products and architectural design. We believe these are all exciting avenues for future works enabled by multiclass shape blending.

CHAPTER 7

Generative Deep Learning with Diverse Multiclass Data as Seeds for Novel Multiscale Structures

7.1. Introduction

The intersection of structural design, a high dimensional problem, and deep learning (DL), a technology to create mappings between shapes and properties in low dimensional spaces, has recently enticed a burst of data-driven design research. In particular, heterogeneous multiscale structures, which are composed of neighboring microstructures that differ from one another (i.e., aperiodic), benefit highly from this paradigm shift. With applications in light-weighting, thermal conductivity, energy absorption [2, 177–179], soft robots [30, 37], active airfoils [28] and more, they are highly desirable. To design heterogeneous structures, however, conventional design approaches such as topology optimization (TO) suffer from several fundamental limitations that impede their capability of scaling up to two or more scales. These underlying challenges include: (a) the "curse of dimensionality" induced from the nested multiscale design space, (b) difficulties ensuring manufacturability due to geometric frustration between disconnected neighboring unit cells, and (c) tradeoffs between the accuracy of full-scale physics-based simulations and the lower cost of homogenization in critical nonlinear applications.

To meet the first challenge, data-driven approaches exploit large databases of unit cells on which DL models can be trained to predict the homogenized properties near-instantaneously (typically with models such as Gaussian processes [3] or multilayer perceptrons [71–73]) and/or to learn lower dimensional latent design spaces through generative models like variational autoencoders (VAEs) [1] and generative adversarial networks (GANs) [70]. Subsequently, design synthesis algorithms can assemble the full multiscale structure by leveraging these models. During this step, expensive microscale simulations can be replaced with the aforementioned predictive models, thereby reducing the multiscale design to an equivalent single-scale problem (the so-called homogenization-based approach) and resolving the "curse of dimensionality". The learned latent representations can further aid in accelerating the design by transforming the microscale design into a parametric one. Recent works have demonstrated that DL models can empower data-driven methods to quickly explore complex design spaces and synthesize novel multiscale systems [1, 3, 70, 72, 73], achieving designs otherwise intractable via traditional approaches.

Regarding the second and third challenges, data-driven approaches split down two paths. On one side are the "top-down" frameworks [1, 34, 65, 66, 70, 72, 73], which include two steps: 1) optimize the macroscale topology and spatial distribution of homogenized material properties, and 2) retrieve the unit cells from either the existing dataset or generate new ones from a trained model. To ensure that the neighboring microstructures are well connected, strategies such as combinatorial search algorithms with compatibility constraints [1], affine unit cell transformations [72], or smoothed transition layers [70] are necessary. Although top-down methods can have considerable design freedom and efficiency when trained on huge, diverse datasets (as in Ref. [1]), they have thus far been confined to the realm of linear elasticity since obtaining optimal properties in complex, e.g., nonlinear, applications is not trivial.

The opposite is “bottom-up” frameworks, which are rooted in conventional TO and directly uses the geometric parameters at both the unit cell and macroscale levels as design variables [3, 227]. Thus, rather than designing the macro- and micro-scales sequentially, the two can be concurrently optimized. These approaches show promise in mitigating the hurdles that their top-down compatriots face when accommodating nonlinear behaviors. However, they are not without their own challenges. More specifically, they are limited by similar restrictions as traditional multiscale TO, which, as mentioned above, suffers from high-dimensionality and geometric frustration. To alleviate these, the heterogeneous design is often limited to only a few pre-defined unit cell types, which leads to sub-optimal solutions. For a review of conventional TO of heterogeneous structures, we refer the reader to Sec. 6.2. Nevertheless, data-driven bottom-up methods should theoretically be able to break out of the computational bottleneck and exceed conventional TO by leveraging the capability of DL models. Due to the low number of bottom-up works, however, their potential has yet to be realized.

This work fills a gap in the bottom-up scene by presenting a data-driven framework for 3D heterogeneous multiscale structures that not only has remarkable design freedom but also high efficiency. Enabling more general and efficient synthesis, we propose a bottom-up framework that uses diverse and quality multiclass unit cell data as seeds for novel and realistic designs. Our method integrates a generative model that learns a unified design space and is capable of generating novel unit cells in between existing classes. By performing multiscale TO in this low-dimensional multiclass space, designs with smoothly graded neighboring microstructures can be created, combining the advantages of multiple classes by, for example, optimally placing trusses, isosurfaces, or hybrids of both. We show that our graded hybrid multiclass designs can achieve not only lower compliance but also lower stress concentrations in linear elastic problems. An extension to a stress-constrained design, which our method solves with ease, highlights the potential of bottom-up frameworks in even more advanced problems with nonlinearity or even multiphysics. Our work is a step toward bringing data-driven design of heterogeneous structures into real-world applications.

7.2. Literature Review

Over the past few years, the engineering community has become more immersed in using deep learning (DL) and computer vision techniques to accelerate design and discovery. In data-driven topology optimization (TO) of multiscale structures specifically, significant speedup to design has been achieved by training a regression model to replace the costly homogenization analysis of unit cells [3, 71–73, 128, 186]. However, through this alone, *few have demonstrated that the data-driven approach can design structures that conventional TO methods cannot when given enough computing power.* Instead, recent research have begun to harness pre-computed unit cell datasets in an emergent approach: generative deep learning (DL). Generative models learn an approximate probability distribution of high dimensional real data in a low dimensional latent space, and a mapping from said latent space back to the real data space. By sampling the probabilistic latent space, new yet realistic data can be generated. For design, generative models proffer a double-edged sword: it can both reduce the dimension of the design space via the latent variables and produce a wide variety of novel designs. These alluring neural network-based models, e.g., multilayer perceptrons, VAEs and GANs [228–237], can be trained to synthesize designs by learning from a dataset of shapes and have, understandably, caught the attention of the TO community. To understand the state-of-the-art and challenges of generative learning-based multiscale design, we provide a brief history of the rise data-driven design in TO, from single-scale to multiscale methods, then review the gaps in the status quo with regard to "bottom-up" design.

7.2.1. Deep Learning for Topology Optimization: A Brief History

DL for TO first appeared in 2017 in the work of Sosnovik and Oseledets. A 2D fully convolutional neural network (CNN) was trained to skip the final iterations of conventional TO, when the topology does not change by much [238]. In other words, it predicts the converged solution of a partially optimized topology. It was extended to 3D not long after [239]. Note that neither of these models are generative.

Unfortunately, this approach is quite impractical as TO must still be run for several iterations before the DL model can be used; it merely reduces TO iterations by post-processing an intermediate result. Also, each sample in the training dataset is obtained by solving a traditional TO problem. In other words, generality is low since, when the design objectives and boundary conditions change, the entire dataset needs to be regenerated and the model re-trained. *These generalization issues, along with physical and manufacturing feasibility, are reoccurring challenges in DL for TO.*

In an attempt to increase the generality of the DL model, another early work included the load and boundary conditions as inputs to their CNN encoder-decoder [240]. It also added a conditional GAN (cGAN) that upscales the low resolution output of the CNN to a higher resolution. Meanwhile, Li et al. trained a GAN to generate low resolution solutions, then also used another GAN in the upscaling stage [241]. Rather than training a secondary model to improve the quality of the solution, some works skip any post-processing [242], or employ a threshold filter for binarization [243]. Unlike Sosnovik's method, no optimization is required to predict near-optimal solutions for any of these works—hence the term "design without any iterations". However, to produce a solution for the design objective, e.g., compliance minimization, the training data are generated by solving TO problems with the same objective. Furthermore, they suffer from solutions with blurry or disconnected members that are neither physically feasible nor manufacturable.

Another problem that can arise in GAN-based methods is *mode collapse*, in which the generator fails to generate diverse samples from all modes in the training dataset's probability distribution. This is due to the competition between the two networks that are trained simultaneously, the discriminator and generator. Since the generator tries to create synthetic samples that can fool the discriminator into thinking that they are real, it can become focused on learning to generate very realistic data from one mode while ignoring other modes. For example, in Ref. [244], the authors trained a cGAN that was conditioned on the volume fraction, meaning that they should be able to generate structures with a chosen volume fraction. However, their model could only produce structures with a volume fraction close to 0.4 no matter what value was requested. A modified architecture and different training strategy was able to alleviate the mode collapse challenge [243], but nevertheless, this work demonstrates the difficulty in training GANs.

Even though the generative models above propose intriguing replacements to the traditional TO method, their practicality is difficult to prove since the models are limited to one objective function and a small set of boundary conditions. They struggle to achieve manufacturing feasible solutions even for simple problems such as compliance minimization, which can be easily solved by TO without the overhead of generating a database. While the early works above focus on single-scale TO, these challenges in generality, feasibility, and mode collapse carry over to multiscale design as well. The framework proposed in this chapter will addresses them through new data acquisition and design synthesis techniques.

7.2.2. "Bottom-Up" Data-Driven Design: Optimization in Latent Space

Unlike the above, an arguably more reusable generative model, which can be trained only once and applied to a broad array of problems, is one that is not restricted to a specific objective. One promising avenue to achieve this is to perform design in a latent space learned by a generative model, essentially turning TO into parametric optimization where the design variables are an abstract representation of the shapes, i.e., the latent variables. Of course, *this assumes that a solution to new objectives or target properties can be found and reconstructed from the latent space, and therefore depends on the coverage of the generative model, which is in turn contingent on the diversity and quality of the training data.*

A few works have touched upon this challenge in designing single-scale structures. For example, in Ref. [245], data was collected by running a single objective TO for each sample, similar to the works in the previous section. Instead of directly obtaining a solution, a style transfer VAE was used to optimize multi-objective problems more complex than the ones solved for data generation. However, the resolution and quality of the reconstructed structures were less than appealing.

Rather than applying TO to generate training data, some works demonstrated far more flexible and reusable generative models by populating the database using parametric or stochastic shape variation without optimization, or by sourcing their data from online repositories. Utilizing an open source dataset of real airfoils, Chen et al. trained a modified version of InfoGANs that allowed them to maximize the lift-to-drag ratio [246]. Without new data or re-training, their generative models can be used again to optimize airfoils for other objectives. Later, the authors extended their approach to include both performance-based and shape diversity-based metrics in their GAN model's loss function, allowing them to generate high-performance, new airfoils outside of the space of the training data [217]. In Ref. [247], training shapes were generated by randomly varying the number, locations, and dimensions of holes. Doing so enabled the authors to use VAE models to inversely design initial designs that can achieve the desired final shapes for two different applications, diffusion-induced morphology changes in microfabrication and mask synthesis in photolithography.

All of the works above, with the exception of Ref. [239] and the newer Refs. [69, 70, 248], are in 2D due to the drastic increase in cost of 3D problems. Besides these, one of the first 3D generative model-based design methods is that of Zhang et al., who used airplane CAD models from the ModelNet40 database to train a variation shape learner (VSL), a hierarchical VAE that learns latent variables which encode both global and local features [249]. By leveraging the VSL, they designed 3D gliders that can hit a target when launched at a given speed and angle. Moreover, the VSL-based optimization was able to suggest a diverse set of generated designs that meet targets similar to the performance of the training data. For targets outside of the performance of the training data, their model was still able to generate designs that were close to the criteria, demonstrating the potential of DL to explore beyond the original design space.

With respect to metamaterials, unit cells were inversely designed to meet target optical responses in Ref. [68]. Their data was generated starting from three geometry classes that are commonly used in the photonic field: cross, split ring and h-shapes. A diverse dataset was created by randomly changing the length, width, offset and rotational angles of the shapes, as well as by random shape distortion. In addition, a VAE and a semi-supervised forward model that predicts the optical properties for latent variable inputs were simultaneously trained. Most interestingly, for certain target optical responses, the optimized shapes did not belong explicitly to any of the classes in the training set—rather, they were either mixtures of the original classes or new shapes beyond those existing. This result has momentous implications for the generative design and discovery of new metamaterials.

Since Ma et al.'s work, attention on generative models in metamaterials and multiscale structural design have risen anew. Notable new research are Refs. [1, 69, 70, 248]. Yet, none of these are "bottom-up" frameworks that utilize the latent vectors of the trained models as design variables. Rather, most use inverse ("top-down") design methods to first identify the optimal mechanical properties of each microstructure, e.g., the homogenized stiffness tensors, displacements, or volume fractions, then generate the microstructural topologies that suit those targets. Such approaches are more flexible in terms of ensuring the connectivity of designs, from restricting the training data to pre-defined, compatible microstructures [248] and a few function-based isosurfaces [70], to using constrained combinatorial search [1] and model-based post-processing [69].

We argue that "bottom-up" approaches are more promising in the long term for real-world applications, where the problems are no longer linear and static but instead nonlinear, and possibly dynamic or multiphysics. In such cases, it becomes a challenge to find the optimal properties of each microstructure, whereas "bottom-up" can leverage well-established methods in the conventional TO field, e.g., Refs. [80, 177, 250]. Moreover, it has been suggested that gradual geometric changes may alleviate the error from the homogenization assumption and improve manufacturability [50, 60]. Recent advances in TO [60, 61, 63, 194, 195, 251] can be borrowed to control the connectivity of heterogeneous structures more effectively than in the "top-down" approaches. However, integrating these techniques in "bottom-up" generative model-based methods has, to our best knowledge, not yet been addressed in literature. Therefore, it is critical to develop new methods that can leverage the generative and efficient capabilities of deep learning models in a "bottom-up" manner.

7.3. Contributions

In previous works, we showed that a smaller yet diverse subset of unit cells improves the scalability of data-driven algorithms as well as the performance of heterogeneous designs (Chapter 5 [196]), and that multiclass, functionally graded structures provide an advantageous trade-off betweeen design performance and efficiency (Chapter 6 [252]). However, those core ideas have not yet been pushed to more challenging but necessary goals: a) ensuring that the datasets contain unit cells that are beneficial to functional grading in terms of geometry, b) increasing the applicability of the same DL models to general design cases, and c) performing multiclass, functionally graded design synthesis in 3D.

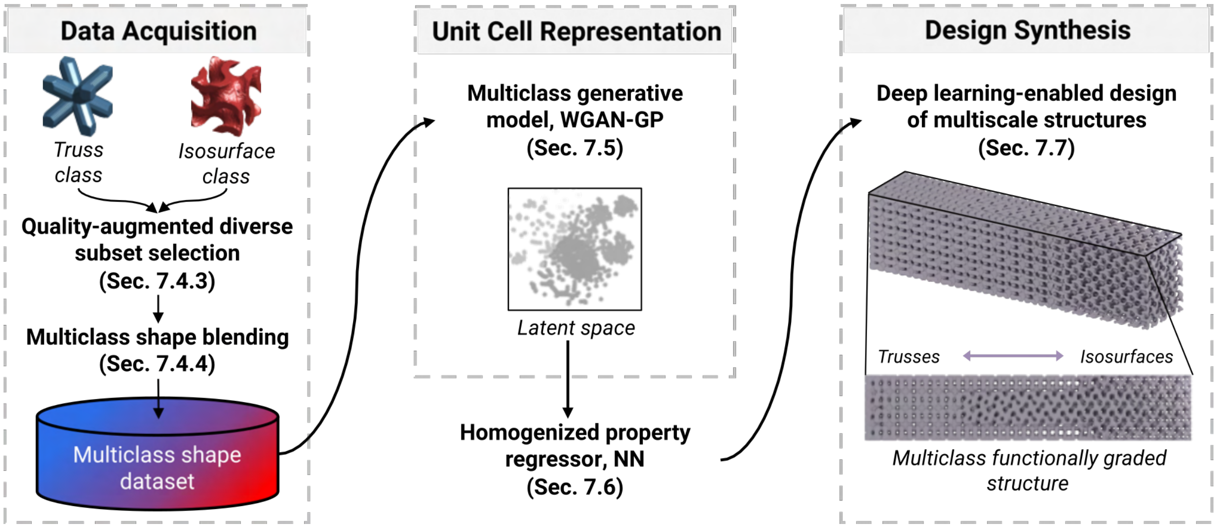

Data Acquisition
Truss class
Isosurface class
Quality-augmented diverse subset selection (Sec. 7.4.3)
Multiclass shape blending (Sec. 7.4.4)
Multiclass shape dataset
Unit Cell Representation
Multiclass generative model, WGAN-GP (Sec. 7.5)
Latent space
Homogenized property regressor, NN (Sec. 7.6)
Design Synthesis
Deep learning-enabled design of multiscale structures (Sec. 7.7)
Trusses
Isosurfaces
Multiclass functionally graded structure

Figure 7.1. Concept of the proposed framework, which addresses challenges in multiclass data acquisition, unit cell representation, and multiscale design synthesis.

Our solution to the first challenge is to acquire a *quality-augmented diverse subset* by extending our probabilistic selection algorithm, METASET (Chapter 5 [196]), to include *quality metrics* on top of similarity metrics. The quality measures we devise can ease the training of our generative model and increase the feasibility of the functionally graded structures we achieve. To integrate morphology types that traditionally have very distinct representations (e.g., bar-and-node trusses and level set function-based isosurfaces) into one data acquisition step, any unit cell can be converted into a signed distance field (SDF), a discretized grid whose elements indicate the distance from that location to the nearest surface, and the similarity between any two SDFs can be calculated based on, e.g., mean absolute error, to form a *class-agnostic diversity metric*.

To address the last two challenges, multiple disjointed design representations are consolidated under our framework by creating a 3D multiclass dataset and training a generative DL model to create a *unified latent design representation*. Given the subset of diverse and quality seed (i.e., basis) classes above, a large shape dataset of unit cell geometries can be quickly created by using multiclass shape blending (Chapter 6 [252]), bridging the truss and isosurface classes through continuous interpolation. This blended dataset allows common generative models found in open source codes to be directly used to learn the multiclass shape space. Leveraging the latent shape representation as design variables, a neural network can be trained to predict the effective properties of newly synthesized unit cells, and concurrent multiscale TO can be harnessed to design both the macrostructure and heterogeneous microstructures.

With this framework, shown conceptually in Fig. 7.1, functional grading between neighboring microstructures is guaranteed even when multiple morphology types are present in the design. In volume- and stress-constrained compliance minimization case studies, we discover that the typical strategy in literature to directly optimize the latent variables of each microstructure is not sufficient to provide a smoothly graded design when searching through an expansive, multiclass shape space. We therefore propose that the multi-material interpolation scheme from Chapter 3 can be extended to be a multiclass scheme. It becomes a simple yet highly effective method to perform *optimal multiclass interpolation in latent space*, not only speeding up the optimization runtime but also guiding the shapes of neighboring microstructures to change continuously, even between disparate classes. Furthermore, by guaranteeing superior connectivity, we can create designs with lower maximum von Mises stress.

7.4. Creation of Multiclass Blended Dataset

To demonstrate our multiclass framework, we choose truss- and isosurface-type classes because the former is popular in literature for high strength-to-weight ratio [253], while the latter is well-known to exhibit high energy absorption due to their amenability to functional grading [51]. Our approach is driven by a generative model that learns on shape data only. However, it is critical that our training dataset includes unit cells that are not only diverse in shapes and properties (to increase generality) but also high quality, i.e., beneficial to the performance of the design (to decrease infeasible design regions that waste resources). In addition, we wish to be able to generate new unit cells that are interpolations of their neighboring microstructures, since this will facilitate smooth grading. Therefore, our data acquisition strategy follows three stages. First, we focus on efficiently collecting a large set of truss and isosurface shapes, which is accomplished by leveraging crystallographic rules that have already enumerated all possible symmetries in a cubic domain. Second, we extend our previous work, a probabilistic subset selection algorithm called METASET (Chapter 5 [196]), to distill a high quality subset of 50 trusses and isosurfaces that are also diverse in shapes and properties. Finally, we leverage a multiclass shape blending scheme with guaranteed self-connectivity (Chapter 6 [252]), to quickly interpolate each pair of basis classes, acquiring a shape dataset of trusses, isosurfaces, and hybrids of both. The unit cells of the final training dataset are represented as $64 \times 64 \times 64$ signed distance fields (SDFs), and all properties are evaluated assuming a constitutive material with a Young's modulus of 1 and a Poisson's ratio of 0.3.

7.4.1. Acquisition of Isosurface Unit Cells

In Chapter 5 [196], we proposed that crystallographic structure factors, which describe how particles are arranged in a crystal unit cell, can be leveraged to quickly gather isosurface classes. These structure factors are similar, and in some cases, identical to the classical triply periodic minimal surfaces (TPMS) found in many multiscale designs [36, 134, 141]. All possible arrangements of crystal structures can be described by these equations, and have been derived in Ref. [174]. We generated a total of 294 cubic structure factors; however, as we noted in our past work [196], this dataset is imbalanced and contains many redundant shapes that, if used directly, could decrease the efficiency of data-driven design. Hence, we wish to increase the diversity of isosurfaces in our final dataset.

7.4.2. Acquisition of Truss Unit Cells

Similar to our work using crystallographic structure factors to derive isosurfaces, Lumpe and Stankovic [253] generated thousands of truss unit cells by treating the bonds between the atoms of a molecule as structural bars. Their collection of crystal-like trusses is provided publicly as lists of nodal positions and bar connectivity. Since their dataset contains all possible symmetry groups, e.g., triclinic, we extracted only the $3,270$ cubic ones. Many of these, however, are very intricate structures with hundreds of bars. Such trusses add little value to our dataset in terms of both design (they are limited to a small range of volume fractions, i.e., radii of their bars) and DL (complex geometries could make our model harder to train well). For this reason, we need to ensure that only the truss classes with high geometric quality are present in the final dataset.

7.4.3. Quality-Augmented METASET for Multiple Design Representations

Compiling the two initial isosurface and truss datasets above gives us an imbalanced, low quality and relatively small training set for generative DL. We now seek to improve its diversity, quality and size efficiently.

7.4.3.1. Measuring Similarity Between Unit Cells with Different Representations. Intuitively, a diverse set contains items that are not similar to each other; to select a subset with maximum diversity, then, one can minimize the similarity between each pair of items. This is the core concept of METASET, our approach to autonomously select diverse subsets (more details to follow in Sec. 7.4.3.3).

Clearly, to perform subset selection, one first needs to quantify the similarity between any two unit cells, which can be accomplished using:

$$S_{ij} = \exp(-0.5\ d(i,j)^2), \tag{7.1}$$

where $d(i,j)$ is the distance between the i-th and j-th unit cells.

The existing approach in Chapter 5 [196] calculated $d(i,j)$ using the Haussdorff distance and the latent vectors of an autoencoder trained on the ground dataset in question. However, Haussdorff distance is relatively expensive in 3D, and the autoencoder is out of the question here, since we cannot ensure its accuracy when trained on the initial, imbalanced, low-quality multiclass set. Nor do we want to add another model to train in this pipeline. Rather, we propose a simple solution: convert all trusses and isosurfaces into signed distance field (SDF) representations. We use geometry projection [203], which is used in TO methods, to handle the conversion from truss bars and nodes to SDFs. Since isosurfaces are already described as level set surfaces, there is no need to convert them. However, we regularize the SDFs of all unit cells using the fast marching method [221] and set their isovalues so that they all have volume fractions of $v = 0.4$, such that they can be compared on equal footing.

Subsequently, our class-agnostic distance metric is the mean absolute error between SDFs:

$$d(i, j) = \sum_{n}^{N} |\mathbf{\Phi}_i - \mathbf{\Phi}_j|, \tag{7.2}$$

where $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_j$ are the SDFs of the i-th and j-th unit cells, respectively, and N is the total number of SDF values in each shape.

As for similarity in the property space, no special treatment is necessary since the homogenized stiffness tensor of all unit cells, regardless of their traditional representation, can be calculated the same way. We use the energy-based homogenization method [88], then take property similarity using Eq. 7.1 and the Euclidean distance between the vectorized stiffness tensors of the i-th and j-th unit cells. Note that we only calculate the property once for each class for the unit cell with volume fraction equal to $v = 0.4$.

Following the approach in METASET, we measure the joint coverage in both spaces by taking a weighted sum of the two similarity matrices, allowing the trade-off between diversifying in shape or property space to be tuned:

$$S = (1 - w) \cdot S_P + w \cdot S_S, \tag{7.3}$$

where S, S_P and S_S are the kernels for the joint, property and shape similarities, respectively, and $w \in [0, 1]$ is a weight parameter.

7.4.3.2. Quantifying Geometry-based Quality. To design successful heterogeneous structures, we propose a few quality metrics based on geometry, which will later be incorporated into METASET (Sec. 7.4.3.3) to automate the selection of high quality unit cell classes. We base these metrics on ease of calculation and quantifiable features that will improve our target designs and 3D generative models, but could be applicable other problems outside our studies.

For functionally graded structures, it is advantageous to incorporate unit cell classes with a wide range of feasible volume fractions, which we define as the range over which a unit cell is neither broken nor containing internal voids. Once represented as SDFs, it is relatively straightforward to find a class's feasible range by using the bisection algorithm. In other words, we can find the range by evaluating whether the solid elements of a unit cell $\mathbf{\Phi} - t \leq 0$ is feasible or not over isovalues t between $[0, 1]$. The first quality metric is then

$$q_i^{(1)} = v_{fr}, \tag{7.4}$$

where v_{fr} is the feasible range of unit cell i.

To eliminate overly complex trusses, we penalize those with high number of bars, N_{bar}. However, initial experiments indicated that some complexity is needed to cover wider property spaces. For example, unit cells with higher average connectivity index, Z_a, which corresponds to the average number of bars connected to a node, have been shown to be more rigid [253]. Therefore, we force these two measures (both provided by the open source dataset) to compete during subset selection by introducing two more quality metrics:

$$\begin{aligned} q_i^{(2)} &= 1 - N_{bar}/N_{bar,max}, \\ q_i^{(3)} &= Z_a/Z_{a,max}. \end{aligned} \tag{7.5}$$

For isosurface classes, only the first is used ($q_i = q_i^{(1)}$), while the mean of all three is used for trusses ($q_i = (q_i^{(1)} + q_i^{(2)} + q_i^{(3)})/3$). The vector of these values for all unit cells will be known as the quality vector.

7.4.3.3. METASET: Subset Selection Algorithm. With all of the classes converted to SDFs, from which we can calculate the similarity kernels and quality vectors, we can now down-select a smaller yet diverse and high quality subset. Briefly, we introduce how to select subsets of unit cells based on diversity only using our prior work, METASET (Chapter 5 [196]). Earlier, we mentioned that diversity measures the volume that a set occupies in a continuous space. Determinantal Point Processes (DPPs) allow us to capture this quantitatively by modeling the probability of selecting any subset of items, M, as related to the determinant of a positive semi-definite similarity kernel matrix, S:

$$\mathbb{P}(M) = \frac{det(S_M)}{det(S+I)}, \tag{7.6}$$

where $S_M \equiv [S_{ij}]_{ij \in M}$ is the submatrix of S with entries indexed by elements of the subset M, and I is an $N \times N$ identity matrix. Intuitively, then, a larger determinant of S_M means that the subset covers a larger volume and is hence more diverse.

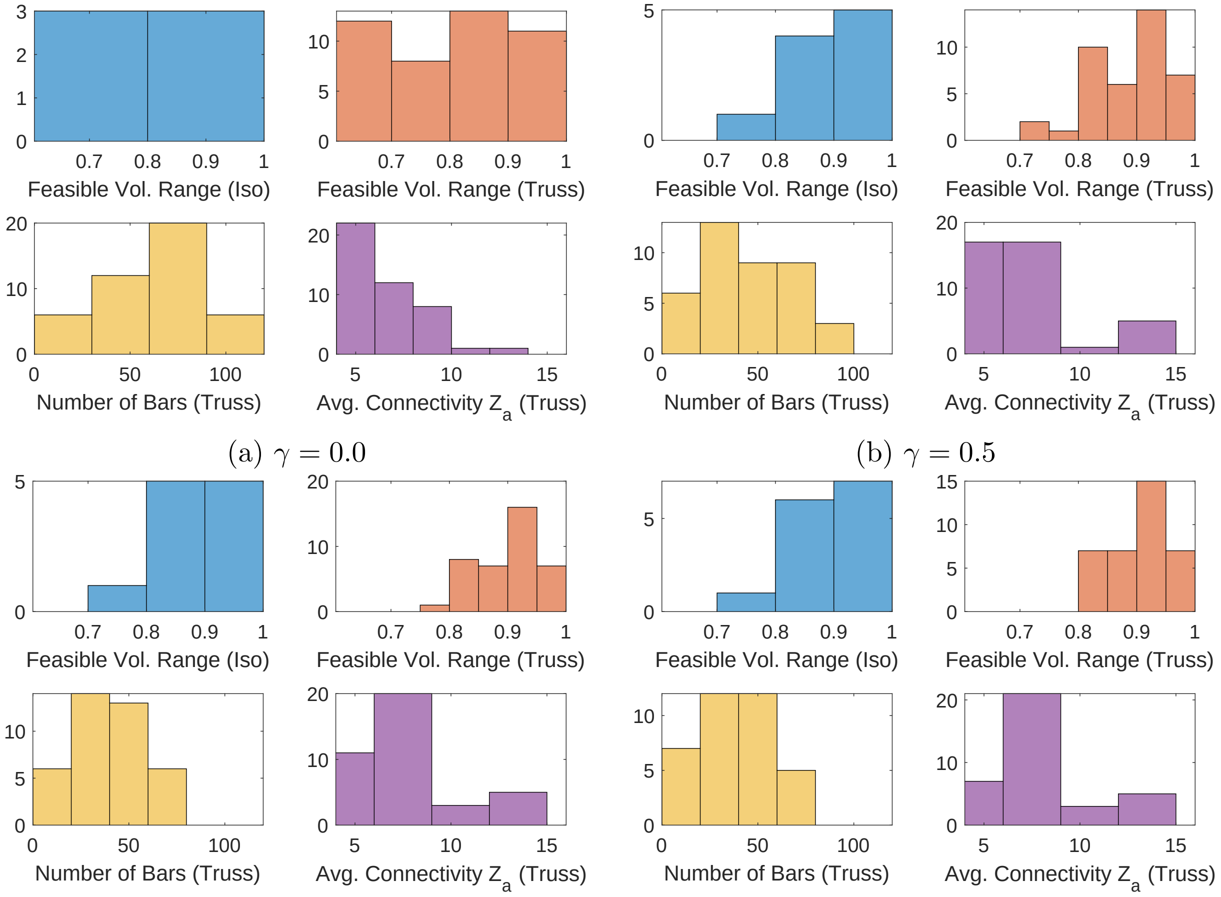
Another benefit of using DPPs is that the log determinant of the positive semi-definite matrix joint similarity kernel, i.e., $f = \log[det(S_M)]$, is submodular. This means that greedy algorithms such as the one used in METASET can optimize this objective with the best possible polynomial-time efficiency. We encourage the reader to refer to Refs. [153, 154, 196] for more theoretical details on DPPs, and Refs. [153, 196] for the greedy algorithm.

This work extends METASET to include quality on top of diversity during subset selection. The decomposition of a DPP kernel into diversity and quality parts can be written as [154, 217]:

$$L_{ij} = S_{ij}\big(q_i q_j\big)^{\gamma}, \tag{7.7}$$

where L_{ij} is the quality-augmented diversity kernel and $\gamma \geq 0$ is a parameter that allows the weight of quality to be tuned. At $\gamma = 0$, L_{ij} is equivalent to including only the similarity kernel, S_{ij}, and at high values of γ, the algorithm will focus primarily on selecting high quality data. Thus, by greedily maximizing $f = \log[det(L_M)]$ using the METASET algorithm, we can select a subset is that both diverse and high quality with respect to the metrics we have defined.

7.4.3.4. Results: Diverse and Quality Multiclass Seeds. Starting with the ground dataset of 294 isosurfaces and $3,270$ trusses, we utilize the quality-augmented METASET algorithm to select subsets that will then serve as our seeds, or basis classes, for novel



(a) $\gamma = 0.0$

(b) $\gamma = 0.5$

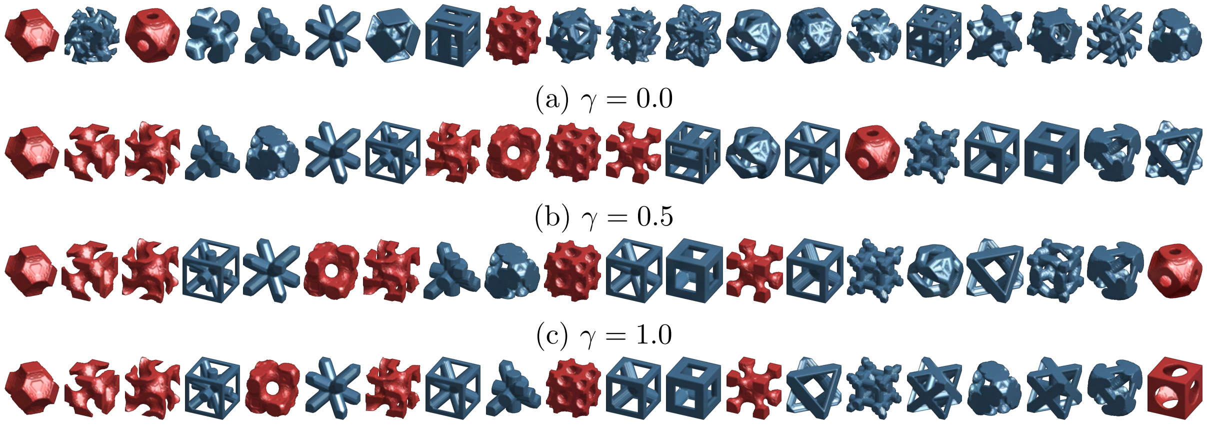
(c) $\gamma = 1.0$

(d) $\gamma = 2.0$

Figure 7.2. Histograms of the geometric features of basis classes selected by quality-augmented METASET. At $\gamma = 0$, the algorithm focuses only on shape and property diversity. As γ increases, METASET selects more high quality classes. This can be seen in the shift of the feasible volume fraction ranges toward the right, while number of bars and Z_a move left.

multiclass designs. We investigate the effect of the tuning parameter on quality, γ. For all experiments, we fix the parameter on joint diversity to be $w = 0.5$, which weights shape and property equally. In Chapter 5 [196], we discovered that diversity in properties can aid in design performance while shape diversity can lead to superior connectivity. We also use subsets of 50 basis classes, since this large size both ensures that we will have many choices for the later design synthesis, i.e., large design freedom. It is also a size that has never before been attempted in bottom-up data-driven approaches, and hence will provide an appropriate litmus test for the capability of our proposed framework.

Fig. 7.2 validates that, as γ increases, METASET puts more and more emphasis on selecting high quality classes, i.e., those with high quality metric values. We observe that the feasible volume fraction ranges of both trusses and isosurfaces shift towards the right, or higher ranges, which we defined as valuable. Meanwhile, the number of bars and average nodal connectivity, Z_a, of trusses shift to the left, decreasing as desired. We affirm this visually as well. In Fig. 7.3, we show the top 20 classes selected for different γ values. Once again, as γ increases, complex trusses are pushed out of the top rankings, while more isosurfaces, which tend to have higher feasible volume ranges, are included.



(a) $\gamma = 0.0$

(b) $\gamma = 0.5$

(c) $\gamma = 1.0$

(d) $\gamma = 2.0$

Figure 7.3. The top 20 unit cell classes chosen with different quality tuning parameter, γ. Red are isosurfaces while blue are trusses.

In addition, we plot the tradeoff between diversity and quality as γ changes in Fig. 7.5. We make an important note: although it appears that when $\gamma > 0.5$, the subsets seem to be of high quality and low diversity, it is deceptive because we have chosen a large subset

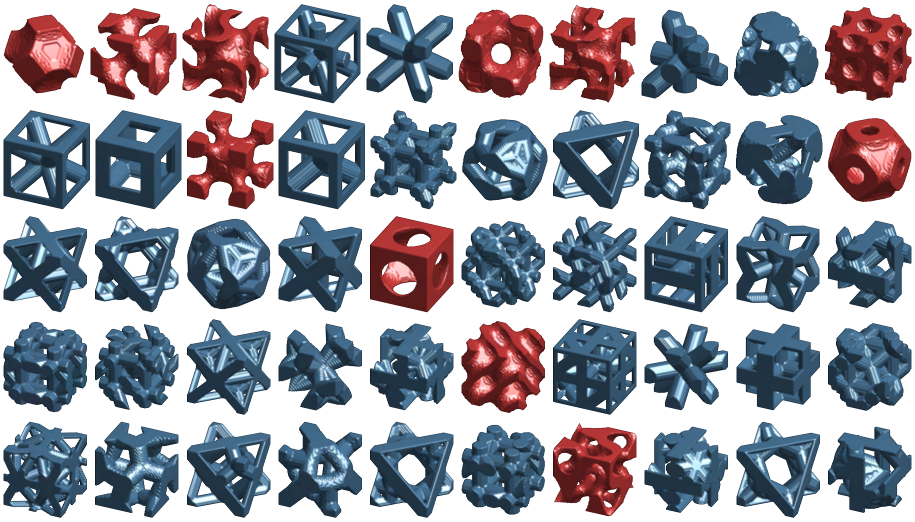


Figure 7.4. The final quality-augmented diverse subset of basis unit cell classes (size 50, $k = 0.5$, $\gamma = 1.0$). Red are isosurfaces while blue are trusses.

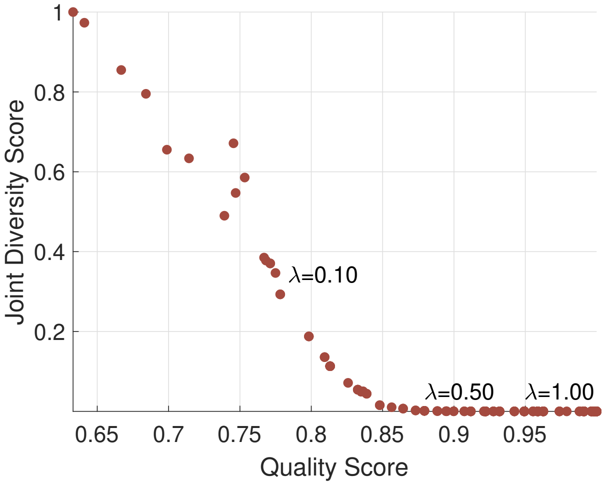


Figure 7.5. The trade-off between the joint ($k = 0.5$) diversity and quality scores as the tuning parameter γ is increased from 0.0 to 1.0 in equal intervals. The scores are normalized by the maximum value over all subsets. The number of items in each subset is 50.

size of 50, for which the diversity score is unavoidably low (it was found in Refs. [153, 196] that diversity diminishes quickly as the number of items increases). However, this does not mean that the subsets have no diversity: see Figs. 7.3 and 7.4 for the shapes and Fig. 7.11a later for the properties.

Given that high quality subsets with larger γ values are able to better suppress the undesirable, complex trusses, we choose to use $\gamma = 1.0$ for the rest of our work. The final subset of 50 truss and isosurface basis classes are illustrated in Fig. 7.4. Despite the fact that we did not focus entirely on shape diversity (e.g., use $k = 0, \gamma = 0$), we can confirm that there is still a satisfactory variety of shapes.

7.4.4. Dataset Expansion with Multiclass Shape Blending

Using the 50 basis classes, we can now expand the shape dataset into a large one suitable for multiclass generative learning. To do this efficiently, we leverage the shape blending scheme from Chapter 6 [252]. The scheme performs a weighted union of multiple SDFs, generating new unit cells while guaranteeing feasibility (i.e., self-connectedness) and smooth grading between other unit cells with similar weights. It is modified slightly for this work to blend only two classes at a time, and is written as:

$$\mathbf{\Phi} = \frac{1}{\beta}\log\Big\{\exp\big[\beta\big((1-c)\mathbf{\Phi}_i^* + c\mathbf{\Phi}_j^*\big)\big] + (1-a)\exp\big(\beta\mathbf{\Phi}_i^L\big) + a\exp\big(\beta\mathbf{\Phi}_j^L\big)\Big\} - t, \tag{7.8}$$

where $\{i, j\}$ are any two pairs of basis classes, c is the weight parameter, $\mathbf{\Phi}^*$ is the SDF of a class at volume fraction $v = 0.4$, $\mathbf{\Phi}^L$ is the SDF at the lower feasible bound of a class, which is enforced to maintain the feasibility of the generated shapes, and t is the isovalue. Both $\mathbf{\Phi}^*$ and $\mathbf{\Phi}^L$ have already been found earlier using the bisection algorithm. Since we do not need the blending scheme to be differentiable (we will not use it directly during design synthesis), we set β to a high value of 128, and define $a = (c \geq 0.75)$ as a binary parameter that "activates" the lower bounds of at least one class.

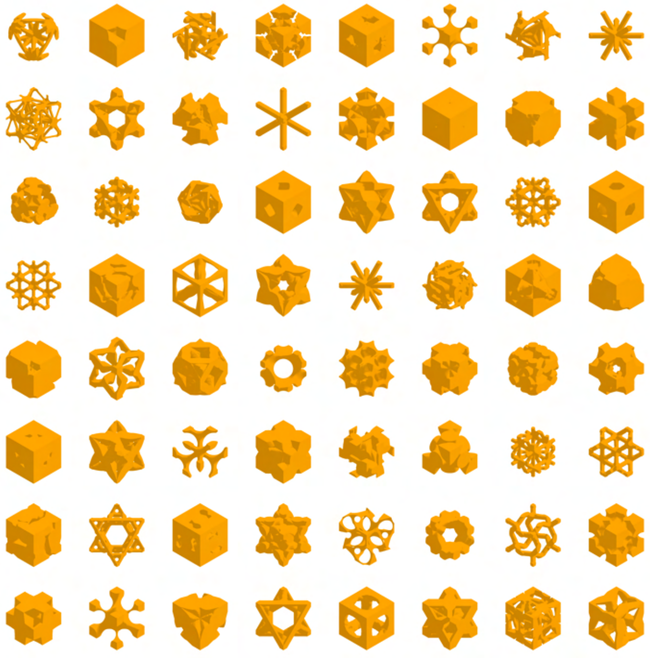


Figure 7.6. Random unit cells from the training dataset created by blending the basis classes in the quality-augmented diverse subset.

We blend only pairs of basis classes $\{i, j\}$ and sample the parameters of Eq. 7.8 relatively sparsely for two reasons. One is to create the training dataset as efficiently as possible; therefore, we avoid sampling the entire combinatorial space of 50 classes. Another is we assume that WGAN-GP should be able to fill in any gaps once it is trained. For each pair, we sample just 5 equally spaced blending weights, c, between $(0, 1)$, and for each value of c we sample 10 values of t to obtain unit cells of different volume fractions. With this process, we obtain a training set of $61,700$ multiclass blended unit cells covering trusses, isosurfaces, and hybrids. Random shapes from this set are plotted in Fig. 7.6.

7.5. Multiclass Generative Model

Following the creation of a large dataset of multiclass blended shapes, we use generative models to compress the shape design representation, and to allow the synthesis of novel unit cells interpolated from the seed basis classes. We hypothesize that the blended training data will guide the models to learn interpolated hybrids without much time-consuming customization and tuning. In this work, we implement the Wasserstein generative adversarial network with gradient penalty (WGAN-GP) [254], a stabler variation of GANs that can also aid in avoiding mode collapse. To underscore its generative capability, we also compare it visually against a variational autoencoder (VAE). For both, a latent dimension of 10-D are used, which decreases the design dimensionality considerably compared to the combinatorial space of 50 basis classes.

7.5.1. Wasserstein Generative Adversarial Network with Gradient Penalty

The typical vanilla GAN [255] contains two networks. One is a generator G that generates a sample $\tilde{\boldsymbol{X}} \in \mathbb{R}^D$ when given a noise vector $\boldsymbol{z} \in \mathbb{R}^{d_{\boldsymbol{z}}}$ drawn from a prior distribution $P_{\boldsymbol{z}}$. Therefore, $\tilde{\boldsymbol{X}} = G(\boldsymbol{z})$. Second is a discriminator D, which is a classifier that discerns whether any given samples are from the real distribution (i.e., training data) or generated by $G(\boldsymbol{z})$ (i.e., fake). Training of both components is performed as a minimax optimization, where D minimizes the classification error and G maximizes the chance of tricking D into thinking that a generated sample is real. The objective is written as

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{X} \sim P_{\text{data}}}[\log D(\boldsymbol{X})] + \mathbb{E}_{\boldsymbol{z} \sim P_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z})))], \tag{7.9}$$

where P_{data} is the training data distribution.

Over the course of training, the networks learn a highly nonlinear mapping from $\boldsymbol{z}$ to $\tilde{\boldsymbol{X}}$, which can also be considered nonlinear dimension reduction if the dimensionality of $\boldsymbol{z}$ is much lower than that of D. A well-trained generator, G, can transform any random latent (noise) vector $\boldsymbol{z} \sim P_{\boldsymbol{z}}$ to a new and realistic sample. Thus, the latent vector $\boldsymbol{z}$ can serve as a compact representation of the generated shape, $\tilde{\boldsymbol{X}}$, and can be leveraged as design variables.

However, vanilla GAN's training process is notoriously unstable since minimizing the divergence between the training data's and generated samples' distributions is not continuous with respect to the generator's parameters [254, 256]. To sidestep this problem, we can utilize a Wassersterin GAN with gradient penalty (WGAN-GP) [254]. Instead of a discriminator that classifies synthetic data as real or fake, WGAN-GP uses a critic, which predicts the Earth Mover's (Wasserstein-1) distance between the distributions of the training and generated samples. Under mild assumptions, the Earth Mover's distance is continuous and differentiable almost everywhere, making the objective:

$$\min_{G} \max_{D \in \mathcal{D}} W(D, G) = \mathbb{E}_{\boldsymbol{X} \sim P_{\text{data}}}[D(\boldsymbol{X})] - \mathbb{E}_{\boldsymbol{z} \sim P_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))], \tag{7.10}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions. In order to satisfy 1-Lipschitz continuity, the gradient should be less than or equal to one everywhere. This is accomplished in WGAN-GP by constraining the gradient norm of the critic's output with respect to its input. The following gradient penalty term is added to the objective:

$$R_1(D) = \mathbb{E}_{\hat{\boldsymbol{X}} \sim P_{\hat{\boldsymbol{X}}}} \left[\left(\left\| \triangledown_{\hat{\boldsymbol{X}}} D(\hat{\boldsymbol{X}}) \right\|_2 - 1 \right)^2 \right], \tag{7.11}$$

where $\hat{\boldsymbol{X}}$ is sampled uniformly along straight lines between pairs of points sampled from the data and generator distributions. The final objective of WGAN-GP is then:

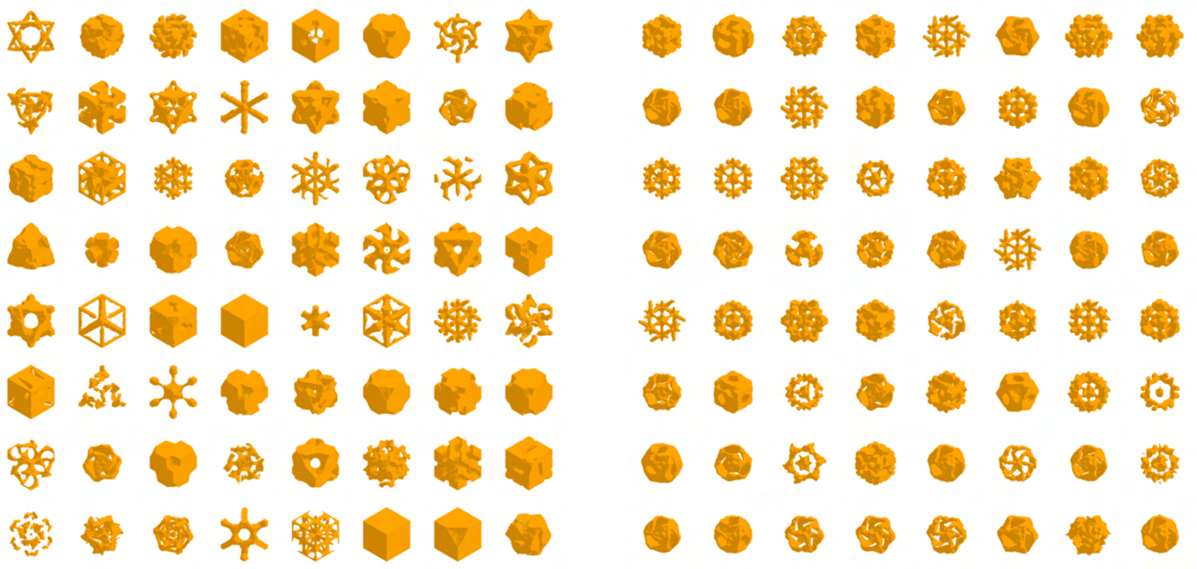
$$\min_{G} \max_{D} W(D, G) - \lambda R_1(D), \tag{7.12}$$

where λ is the weight of the gradient penalty term.

7.5.2. Architecture and Training Details

The architecture of our multiclass WGAN-GP is depicted in Fig. 7.9. The generator takes in the 10-dimensional latent vector and, through several deconvolutional layers, scales it up to a $64 \times 64 \times 64$ SDF – a synthetic unit cell. Meanwhile, the critic passes a unit cell SDF through convolutional layers. We use leaky ReLu activations as indicated in Fig. 7.9, and each has a negative slope coefficient of 0.2. Each batch normalization has a momentum of 0.9. The model was trained using TensorFlow for 50,000 iterations with a batch size of 16 and an Adam optimizer with $\alpha = $ 1e-4, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. We keep the generator constant while updating the critic through the following schedule: in each iteration, the generator is trained once while the critic is trained five times. This allows the critic a chance to learn the generator's "flaws".

For comparison, we also train a VAE. In this case, the encoder has six convolutional layers and one fully connected final layer, and the decoder contains two fully connected layers followed by seven deconvolutional layers. Each leaky ReLu activation in the VAE has a negative slope coefficient of 0.2, while each batch normalization has a momentum of 0.9. The model was trained for 50,000 iterations with a batch size of 16. RMSProp was used as the optimizer with a learning rate of 1e-3.



(a) WGAN-GP

(b) VAE

Figure 7.7. Random unit cells synthesized by two generative models: (a) WGAN-GP and (b) VAE. WGAN-GP is more capable of generating unit cells that resemble "real" shapes (i.e., training data).

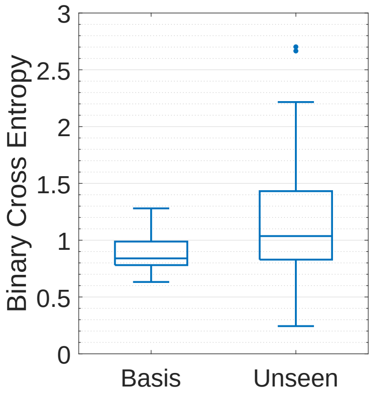
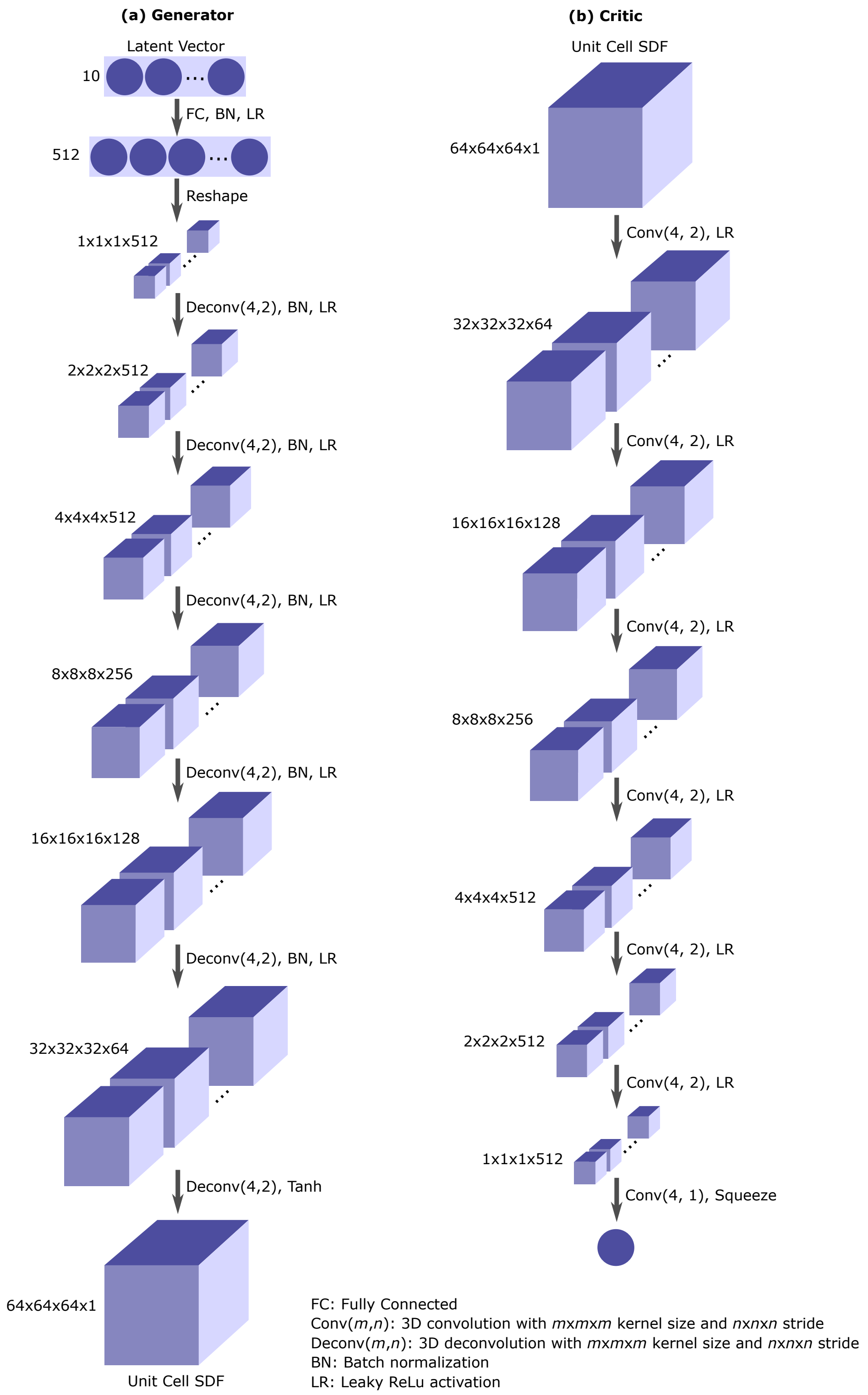


Figure 7.8. Binary cross entropy of recovered basis and unseen blended unit cells. A value of around 1.0 or less is a great fit in our experiments.

7.5.3. Shape Space Coverage

Once both models are trained, we examine which one, WGAN-GP or VAE, can best capture the multiclass space, i.e., generate realistic unit cells that most resemble the



FC: Fully Connected
Conv(*m*,*n*): 3D convolution with *mxmxm* kernel size and *nxnxn* stride
Deconv(*m*,*n*): 3D deconvolution with *mxmxm* kernel size and *nxnxn* stride
BN: Batch normalization
LR: Leaky ReLu activation

Figure 7.9. Architecture of the WGAN-GP model.

training data. Fig. 7.7 compares random synthetic unit cells generated by each model. It is quite clear that WGAN-GP is able to create a wider variety of unit cell geometries over an array of different volume fractions, while VAE mainly sticks to two archetypes with similar volume fractions. We can assume that VAE failed to learn the underlying multi-modal distribution while WGAN-GP successfully escaped mode collapse. By a wide margin, WGAN-GP is the best model to use in our design framework.

We also test the ability of WGAN-GP to recover the 50 basis classes, as well as 50 unseen blended unit cells, i.e., those that were not present in the training set. To do this, we first generate $16,000$ random synthetic data by sampling $\boldsymbol{z} \in [0,1]$ uniformly. A random subset of these were shown in Fig. 7.7a. To create the unseen shapes, we use a space-filling design of experiments method called optimal Latin hypercube [137] to sample the indices of pairs of basis classes to blend, $\{i,j\}$, the blending weight c, and the isovalue t. The values of c and t are different from those used for the training data.

For each of the basis and unseen shapes, we search through the $16,000$ synthetic dataset to find the generated shape that minimizes the binary cross entropy, a metric typically used in classification problems to predict the probability of a new shape being the same as a target one. In our experiments, we found that an entropy value of around 1.0 indicates that the synthetic and target shapes match very well. The minimum entropy values are plotted in Fig. 7.8, where we see that the majority of the basis classes are well represented in the synthetic dataset, and that the unseen unit cells can also be recovered faithfully. These results validate our earlier assumption that there is little mode collapse, and that WGAN-GP can generate novel and realistic shapes beyond our training data.

7.5.4. Post-Processing of Generated Unit Cells

Even with the generative capability of WGAN-GP, we note that it was trained in an unsupervised manner on only shape information. As the model has no quantitative knowledge of what is physically feasible, it does not know when its generated shapes are unreasonable from a mechanics and design perspective. Two major issues are internal voids in generated unit cells with high volume fractions, and broken or disconnected features in those with low volumes. A few examples of this can be seen in Fig. 7.7a.

To alleviate this, we implement a post-processing scheme to "correct" these infeasibilities. It is straightforward since our unit cells are SDFs and therefore trivial to manipulate. For generated designs with interior voids (Fig. 7.5.4(a)), we locate the interior voids using the open-source Python function, `skimage.measure.label`, then fill in those regions by setting the SDF values to be a low negative number (e.g., -64). To resolve broken features (Fig. 7.5.4(b)), we raise the volume fraction of the unit cell by increasing the isovalue t until it is connected. In total, $3,143$ out of the $16,000$ (19.6%) needed to be post-processed, although most were minor adjustments where either only 10's of voxels were filled or the isovalue was increased by less than 0.02 (noting that the range of the values of each SDF $\mathbf{\Phi}$ is ± 64).

7.6. Property Regressor

An essential component of bottom-up data-driven design frameworks is the surrogate model of the microscale simulations. We detail here our neural network for the prediction of unit cell properties. After post-processing the randomly generated synthetic dataset (Sec. 7.5.3) to ensure all $16,000$ unit cells are feasible, we calculate their homogenized

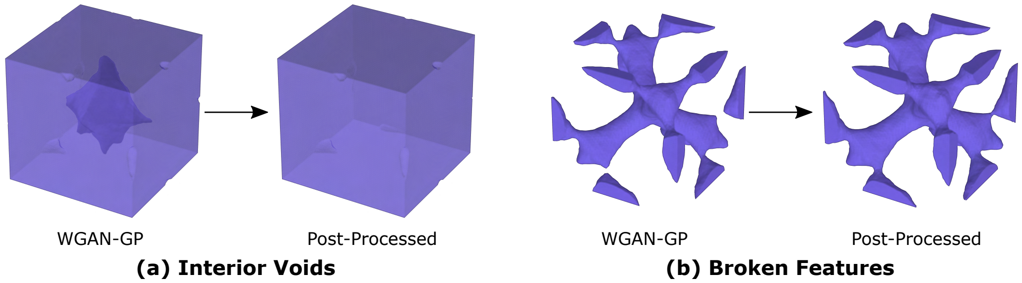


Figure 7.10. Demonstration of the post-processing scheme for two scenarios: (a) unit cells with internal voids, and (b) broken features.

stiffness tensors using the energy-based method [88]. This dataset is able cover a wide property space; a project view is depicted in Fig. 7.11b. The figure also plots the properties of the basis classes, which are shown more closely in Fig. 7.11a. Recalling that we only calculated the properties of unit cells in the basis classes with $v = 0.4$, we observe that our method to create multiclass blended data based only on shapes (Sec. 7.4.4) has also expanded the properties to a large degree. We also note that, since we learn the properties of the processed shapes, we will also need to perform the post-processing scheme (Sec. 7.5.4) after the design synthesis is finished.

Our neural network regressor, implemented in TensorFlow, contains seven fully connected layers with 512 nodes, an L2 regularizer with scale 0.001 and a Relu activation for each. The inputs are the latent vectors of a unit cell, which is also concatenated back into in the fourth layer to improve the prediction accuracy. The final layer outputs the components of the 3D homogenized stiffness tensor, as well as the volume fraction. In other words, our property model is $[\boldsymbol{C}_e, v_e] = NN(z_e)$ for unit cell e.

We find that predicting only the orthotropic components of the homogenized tensors, i.e., $C11, C21, C31, C22, C32, C33, C44, C55$, and $C66$, in addition to the volume fraction,

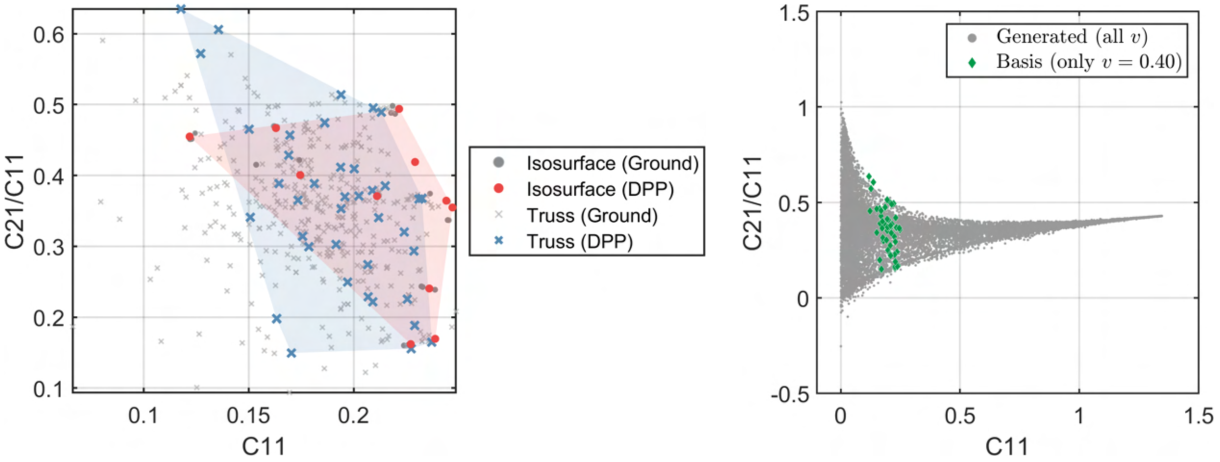
v_e, improves the accuracy of the network compared to regressing all 21 components. This agrees with our observations that over 94% of the unit cells generated by WGAN-GP have trivial values close to zero at the other tensor components. It also follows from the fact that our truss and isosurface basis classes (Sec. 7.4) are orthotropic, indicating that any fully anisotropic unit cells are due to small artifacts or noise in the predicted shapes. The R-squared and mean squared errors (MSE) of the trained property regressor for each prediction response are listed in Table 7.1, which indicate good fit.

Table 7.1. Accuracy metrics of the property regressor.

Property	Train R^2 (MSE)	Test R^2 (MSE)
$C11$	0.995 (7.54e-4)	0.960 (5.61e-3)
$C21$	0.994 (1.35e-4)	0.953 (1.03e-3)
$C31$	0.994 (1.41e-4)	0.954 (1.03e-3)
$C22$	0.995 (7.56e-4)	0.960 (5.59e-3)
$C32$	0.994 (1.40e-4)	0.953 (1.03e-3)
$C33$	0.995 (7.59e-4)	0.960 (5.59e-3)
$C44$	0.995 (6.50e-5)	0.961 (4.99e-4)
$C55$	0.995 (6.50e-5)	0.961 (4.97e-4)
$C66$	0.995 (6.60e-5)	0.961 (5.00e-4)
v	0.995 (4.19e-4)	0.955 (3.71e-3)

7.7. Deep Learning-Enabled Design of Multiclass Functionally Graded Structures

Finally, we can integrate the latent variables, $\boldsymbol{z}$, and the property regressor, NN, with design synthesis. Our proposed bottom-up framework follows conventional TO methodology except for one major exception: since the volume fractions of the unit cells, v, are outputs of the regressor, we cannot use them as design variables like in typical density-based approaches. However, we can still indirectly control volume by optimizing $\boldsymbol{z}$. Therefore,



(a) Final quality-augmented diverse subset (red circles for isosurfaces, blue crosses for trusses). Ground datasets are shown in smaller, gray markers.

(b) $16,000$ random unit cells from the trained WGAN-GP model, covering all possible volume fractions, v.

Figure 7.11. The projected property spaces of the quality-augmented diverse basis subset, and of WGAN-GP generated unit cells.

if we *directly optimize the latent variables of each microstructure*, we can formulate our homogenization-based compliance minimization problem as:

$$\begin{aligned}
&\underset{\boldsymbol{z}}{\text{minimize}} && f = \frac{1}{2}\sum_{e=1}^{N_{el}} x_e \mathbf{u}_e^T \mathbf{k}_e^H(\hat{z}_e)\mathbf{u}_e, \\
&\text{subject to} && \mathbf{K}(\hat{\boldsymbol{z}})\mathbf{U} = \mathbf{F}, \\
&(7.13) && g_1 = V_{Global}(\hat{\boldsymbol{z}})/V^*_{Global} - 1 \leq 0, \\
& && g_2 = v^{ks}_{max}(\hat{\boldsymbol{z}})/v^*_{max} - 1 \leq 0, \\
& && 0 \leq \boldsymbol{z} \leq 1,
\end{aligned}$$

where N_{el} is the total number of microstructures, x_e is an interpolated variable that allows void microstructures (Eq. 7.14), $\mathbf{u}_e$ are element displacements of unit cell e, and $\mathbf{k}_e^H$ are homogenized stiffness matrices obtained from the regressor. To encourage gradual changes in neighboring unit cell shapes, we apply a smoothing filters [86] with radius r_{min} on each dimension of the latent variables to obtain $\hat{\boldsymbol{z}}$. Since this is well-established in TO, we do not spell it out here for brevity. Note that the inputs of the regressor become the filtered variables, i.e., $[\boldsymbol{C}_e, v_e] = NN(\hat{z}_e)$. This problem can be solved via the method of moving asymptotes (MMA) [122].

The global volume fraction of the structure, including the macrostructure and microstructures, is $V_{Global} = \sum_{e=1}^{N_{el}} x_e v_e(\hat{z}_e)/N_{el}$. To simultaneously design the macrostructure, we include x_e, a variable similar to the pseudo-density in SIMP-based methods [43]. The difference is that x_e is not a design variable but rather a result of the following filter:

$$x_e(\hat{z}_e) = \left[\frac{\tanh\left(\beta_1 v^*_{min}\right) + \tanh\left(\beta_1 (v_e(\hat{z}_e) - v^*_{min})\right)}{\tanh\left(\beta_1 v^*_{min}\right) + \tanh\left(\beta_1 (1 - v^*_{min})\right)} \right]^p, \tag{7.14}$$

where p is a penalty parameter, β_1 controls the strength of the filter, and v^*_{min} is the lower bound on microstructure volume fractions. Those familiar with TO will recognize that this is the agglomeration of an approximate Heaviside projection filter and the penalty from SIMP. In short, Eq. 7.14 converts any microstructure with volume fraction, v_e, less than v^*_{min} into an empty or void unit cell by pushing $x_e \rightarrow 0$. Due to linear elasticity, we can also artificially force void unit cells to have low stiffness by using $x_e \mathbf{k}^H_e$, as seen above in Eq. 7.13.

An upper bound on the volume fractions of individual microstructures is enforced via a constraint with KS aggregation so that it is differentiable. Hence,

$$v^{ks}_{max}(\hat{\boldsymbol{z}}) = \frac{1}{k_s} \ln \sum_{e}^{N_{el}} e^{k_s v_e(\hat{z}_e)}, \tag{7.15}$$

where k_s is the aggregation parameter.

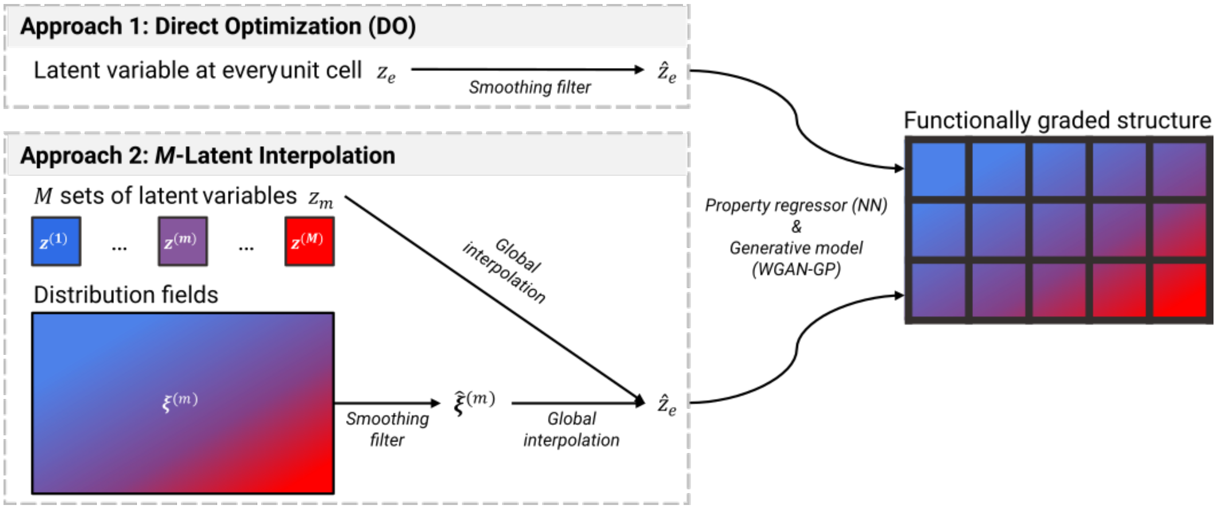

Approach 1: Direct Optimization (DO)
Latent variable at every unit cell z_e
Smoothing filter
$\hat{z}_e$
Approach 2: M-Latent Interpolation
M sets of latent variables z_m
$z^{(1)}$
...
$z^{(m)}$
...
$z^{(M)}$
Distribution fields
$\xi^{(m)}$
Smoothing filter
$\hat{\xi}^{(m)}$
Global interpolation
Global interpolation
$\hat{z}_e$
Property regressor (NN) & Generative model (WGAN-GP)
Functionally graded structure

Figure 7.12. The two proposed multiscale design synthesis approaches. Here global interpolation refers to Eq. 7.16.

However, the direct optimization (DO) approach (top of Fig. 7.12) does not necessarily guarantee that the structures are as functionally graded and well-connected as desired since WGAN-GP does not regularize the latent space to be well-structured (i.e., similar latent vectors may not correspond to similar shapes). To address this, we propose another method for synthesis: *M-latent interpolation* (bottom of Fig. 7.12), wherein a small number, $M \ll N_{el}$, of sets of latent variables, $\boldsymbol{z}^{(m)}, m \in [1, \cdots, M]$, are designed, then interpolated globally into the macrostructure. The sets of distribution fields, $\boldsymbol{\xi}^{(m)}$, are added to optimize the mixture of the M sets of optimal latent variables.

We also introduce a function to map $\boldsymbol{z}^{(m)}$ to the global structure. In previous works, we proposed a multi-material interpolation scheme (Chapter 3 [211]), which was then successfully applied to interpolate multiple unit cells' SDFs in Chapter 6 [252]. Differing slightly from before, we will interpolate latent variables rather than the materials or shapes. The scheme to obtain the globally interpolated latent vectors at each microstructure, e, is as follows:

$$\hat{z}_e = \sum_{j=1}^{M} \left(\left(z^{(j)} - z^{(j-1)}\right) \prod_{k=1}^{j} \hat{\xi}_e^{(k)} \right). \tag{7.16}$$

In contrast to the DO approach above, the radial smoothing filter is only performed on $\hat{\boldsymbol{\xi}}^{(m)}$, not $\boldsymbol{z}$. Otherwise, the optimization problem is exactly the same as Eq. 7.13. With this M-latent interpolation method, we can achieve much greater control over the functional grading of microstructures by deriving each of their latent vectors, $\hat{z}_e$, as a blend of their neighbors' vectors.

To summarize, we propose two design synthesis approaches (Fig. 7.12): direct optimization (DO) of the latent variable of every unit cell, and M-latent interpolation, which designs a small set of latent variables that are then mapped into the global structure. The former requires $10N_{el}$ design variables while multiclass design needs $MN_{el} + 10M$, where 10 is the dimensionality of the latent space. Hence, the number of variables in multiclass design only supersedes DO when $M \geq 10$.

7.7.0.1. Sensitivity Analysis. Since our problem formulation follows traditional density-based TO closely, the sensitivities for the compliance objective can be found using the adjoint method [43]. The sensitivities of the radial filters and KS aggregation functions are also well-known derivations. Examples can be found in Ref. [86] and Ref. [257], respectively. Different from traditional methods, in our data-driven framework the gradients of the homogenized stiffness matrices, $(\partial \mathbf{k}_e^H / \partial \boldsymbol{C}_e)(\partial \boldsymbol{C}_e / \partial \hat{z}_e)$, and of the volume fractions, $\partial v_e / \partial \hat{z}_e$, are obtained from the neural network through automatic differentiation in TensorFlow.

In the global interpolation approach, the additional sensitivities of Eq. 7.16 for $m \in [1, M]$ are as follows:

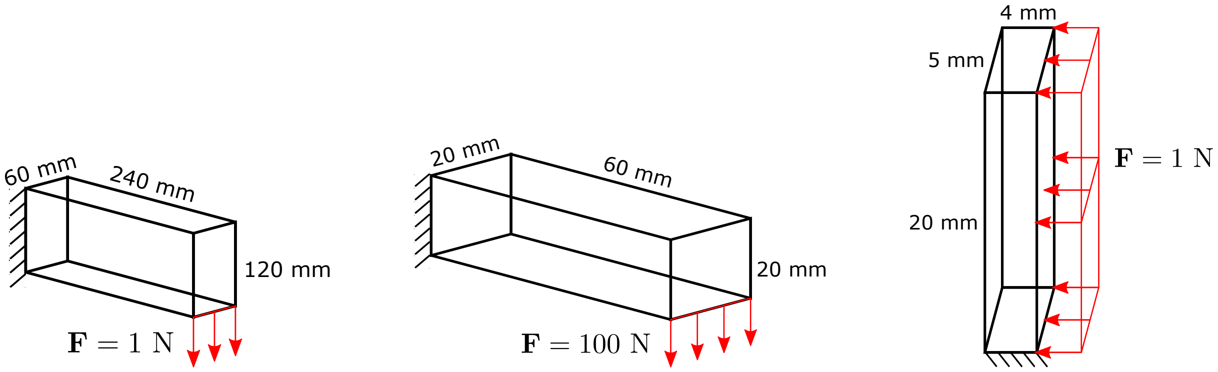
$$\frac{\partial \hat{z}_e}{\partial z^{(m)}} = \sum_{j=1}^{M} \left[\frac{\partial \left(z^{(j)} - z^{(j-1)} \right)}{\partial z^{(m)}} \prod_{k=1}^{j} \hat{\xi}_e^{(k)} \right], \tag{7.17}$$

$$\frac{\partial \hat{z}_e}{\partial \xi_e^{(m)}} = \left\{ \sum_{j=m}^{M} \left[\left(z^{(j)} - z^{(j-1)} \right) \prod_{k=1, k \neq m}^{j} \hat{\xi}_e^{(k)} \right] \right\} \frac{\partial \hat{\xi}_e^{(m)}}{\partial \xi_e^{(m)}}, \tag{7.18}$$

where the term $\partial \hat{\xi}_e^{(m)} / \partial \xi_e^{(m)}$ is the sensitivity of the radial filter [86].

7.8. Illustrative Examples

We use compliance minimization problems to benchmark and validate our framework. First, we follow the small cantilever case study with 12 × 6 × 3 unit cells from Ref. [50], where the compliance (i.e., strain energy) of different strategies for functional grading were compared. Its boundary conditions are illustrated in Fig. 7.13a. We then perform the mid-sized, 18 × 6 × 6, cantilever study in Ref. [134] (Fig. 7.13b). Finally, we demonstrate the potential of our methods with a slightly more complex stress-constrained 4×5×20 actuator design (Fig. 7.13c), inspired by Ref. [258]. As stated in Sec. 7.4, the constitutive material has a unit Young's modulus and a Poisson's ratio of 0.3. We implement our synthesis framework using the Python libraries, `OpenMDAO` [259] and `ParOpt` [260], and utilize the method of moving asymptotes (MMA) [122] for gradient-based optimization. To evaluate compliance, we modify the source codes from Ref. [257] to enable homogenization-based analysis.



(a) Small cantilever
(12 × 6 × 3)

(b) Mid-sized cantilever
(18 × 6 × 6)

(c) Actuator
(4 × 5 × 20)

Figure 7.13. Boundary conditions of the design examples.

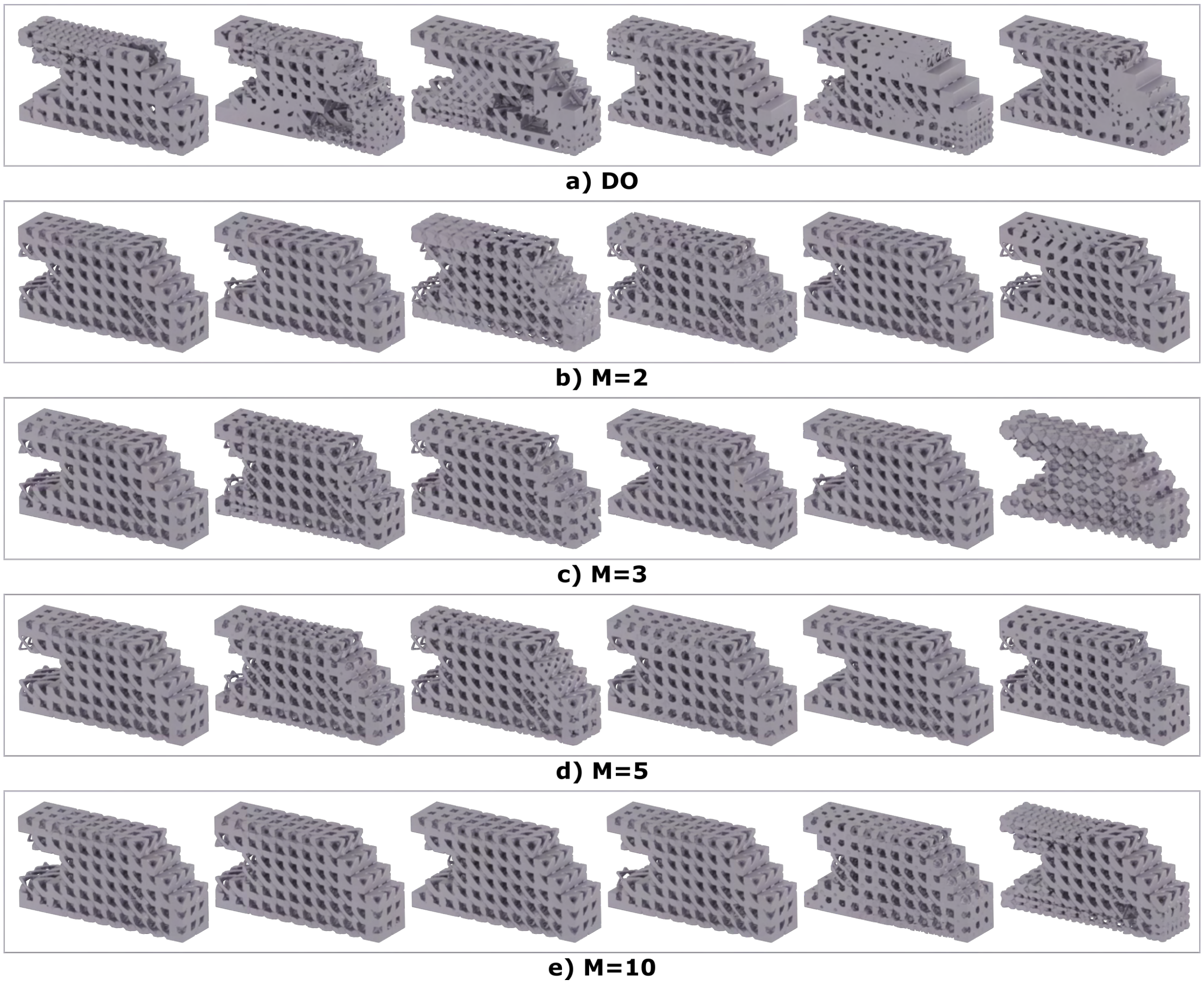
In each study, we apply our proposed synthesis methods, DO and M-latent interpolation with $M = \{2, 3, 5, 10\}$. For each of these configurations, we also test different starting solutions of the latent variables: one fixed start where all $\boldsymbol{z} = 0.5$, and 10 random initializations of $\boldsymbol{z}$. For the small cantilever, $V^*_{Global} = 0.5$ and $r_{min} = 2.0$, whereas they are 0.3 and 3.0 for the mid-size cantilever, and 0.6 and 3.0 for the actuator. For all problems, other parameters, when applicable, are $v^*_{min} = 0.2$, $v^*_{max} = 0.8$, $\beta_1 = 32$, $k_s = 40$, and $p = 3.0$.

7.8.1. Compliance Minimization Results

Since WGAN-GP generates unit cells as SDFs, it is straightforward to convert them into smooth, triangulated meshes of the optimized designs for analysis and visualization. For each of the DO and M-latent interpolation methods, we render six random results of the small cantilever in Fig. 7.14, and three of the mid-sized cantilever in Fig. 7.17. The rendered designs corroborate our earlier theory that the DO approach produces structures that are not as smoothly graded as those from the M-interpolation approach. Although close inspection reveals that the neighboring microstructures in the DO results are indeed connected to their immediate neighbors, the variation of morphology types across the global structure can be abrupt. We believe this is due to: 1) the linear smoothing filter being ineffective when used directly on latent variables, 2) the unregularized latent space of WGAN-GP, as mentioned in Sec. 7.7, and 3) the huge design space encapsulated by WGAN-GP, which allows the design to include too many types of geometries to its detriment. Although the DO results are not quantitatively bad (in fact, their compliance values are generally lower than those of M-interpolated designs), the larger geometric changes from microstructure to microstructure does indicate that the design freedom needs to be more restrained so that smoother functional grading can be achieved.

Our second approach, M-latent interpolation, solves this issue without significantly limiting the achievable design performance. For the 11 total runs using each synthesis approach, we plot the compliance, the post-processed global volume fractions, and the actual maximum von Mises stresses (without KS aggregation, and using homogenization-based analysis) in Figs. 7.15 and 7.16 for the small and mid-sized cantilever, respectively. The average compliance over all initializations of the design variables decreases as M increases, until the values approach or even surpass those of the DO method.

One of the most compelling findings is that, with the M-interpolated approach, the optimized designs achieve not only more geometrically pleasing functional grading of the microstructure shapes but also considerably lower maximum stresses, as seen in Figs. 7.15c and 7.16c. Moreover, as M increases, the results of different initializations converge to a



a) DO

b) M=2

c) M=3

d) M=5

e) M=10

Figure 7.14. Select results of the small cantilever example using random initializations of the latent variables, $\boldsymbol{z}$. (a) Results from the DO approach, which may cause abrupt geometric changes. (b-e) Results by M-latent interpolation, which exhibit superior functional grading. As M increases, the designs converge with respect to geometry, despite different initializations.

similar structure, which one may assume to be locally optimal. Hence, via globally interpolating (Eq. 7.16) M sets of latent variables, we can create designs with less geometric variance than the DO method, even when $M = 10$.

We also validate these observations by re-calculating the homogenized stiffness tensors of each generated microstructure using analytical homogenization [88] rather than the property regressor neural network, then performing homogenization-based finite element analysis of the assembled designs. For each of the proposed design synthesis approaches (DO and M-latent interpolation), six runs with different initializations of the latent design variables, $\boldsymbol{z}$, are compared in Tables 7.2 and 7.3. Overall, the predicted compliance and maximum von Mises stresses match well with the analytical values, ranging from absolute errors of 0.5% to about 13%, indicating that the accuracy of the property regressor is sufficient. We confirm the trend that, while the DO approach can reach lower compliance values, the M-interpolation method achieves lower von Mises stresses since it encourages more gradual changes in microstructure topologies. In future works, we plan to expand this validation by also performing full-scale analysis without the homogenization assumption.

Table 7.2. Validation of the compliance for the small cantilever using the property regressor predictions versus analytical homogenization by comparing 6 initializations from each synthesis approach. Predicted values agree well with analytical results, and the DO synthesis approach tends to achieve lower compliance than M-interpolation with $M = 2$.

	Direct Optimization (DO)			Latent Interpolation ($M = 2$)		
Initialization	Predicted	Analytical	% Error	Predicted	Analytical	% Error
Fixed ($\boldsymbol{z} = 0.5$)	0.7332	0.8072	9.2	0.7865	0.8724	9.9
Random 1	0.7420	0.7982	7.0	0.7838	0.8369	6.3
Random 2	0.7623	0.8505	10.4	0.8693	0.8596	1.1
Random 3	0.7623	0.8254	7.6	0.7731	0.7689	0.5
Random 4	0.7541	0.8189	7.9	0.7758	0.8826	12.1
Random 5	0.7742	0.8921	13.2	0.7713	0.8745	11.8
Mean	**0.7547**	**0.8321**	**9.2**	**0.7933**	**0.8492**	**7.0**

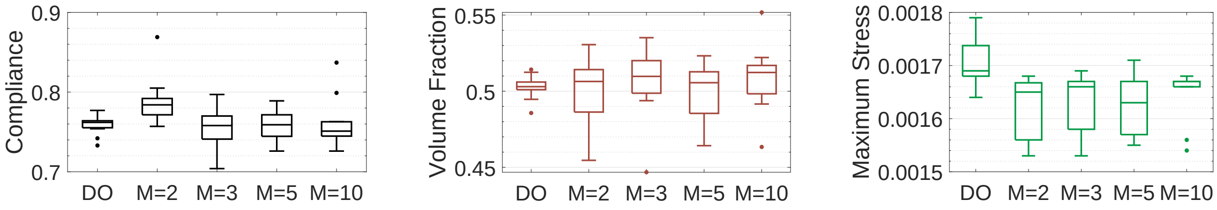
Table 7.3. Validation of the maximum stress for the small cantilever using the property regressor predictions versus analytical homogenization. Predicted and analytical values are close, and, as expected, the M-interpolation approach (using $M = 2$) achieves lower stresses than the DO method.

	Direct Optimization (DO)			Latent Interpolation ($M = 2$)		
Initialization	Predicted	Analytical	% Error	Predicted	Analytical	% Error
Fixed ($\boldsymbol{z} = 0.5$)	1.68e-3	1.70e-3	1.6	1.56e-3	1.58e-3	1.6
Random 1	1.69e-3	1.75e-3	3.6	1.54e-3	1.53e-3	0.1
Random 2	1.74e-3	1.83e-3	4.7	1.67e-3	1.69e-3	1.5
Random 3	1.69e-3	1.77e-3	4.5	1.68e-3	1.63e-3	3.0
Random 4	1.68e-3	1.67e-3	0.5	1.58e-3	1.61e-3	2.1
Random 5	1.64e-3	1.82e-3	10.1	1.53e-3	1.55e-3	1.6
Mean	**1.69e-3**	**1.76e-3**	**4.2**	**1.59e-3**	**1.60e-3**	**1.7**

However, our framework does not have as much success meeting the low global volume constraint of 0.3 in the mid-sized cantilever example. Fig. 7.16b shows that, although the average volume fraction over the 11 runs with different initializations is around 0.31, the 25th percentiles (i.e., lower quantiles) are greater than than the constraint limit. This could arise from a few sources, including: 1) it is possible that WGAN-GP does not generate many feasible low volume fraction shapes, 2) our regressor (Sec. 7.6) is not as accurate for those unit cells, and 3) it may be difficult to control both the macrostructure and unit cell volumes when using the filter in Eq. 7.14. These areas need more detailed investigation or improvements in future works.

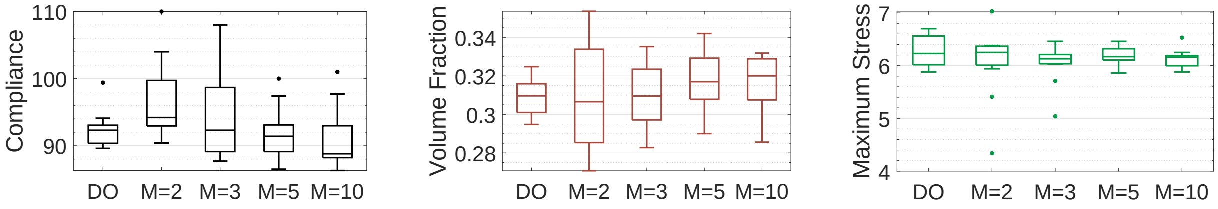
7.8.2. Stress-Constrained Actuator Design

In Ref. [258], a two-material soft actuator was designed out of bioprinted hydrogels by minimizing the compliance of a beam load as shown in Fig. 7.13c. The distributed load mimics the pressure that the structure experiences on one face when electrically actuated [258].



(a) Compliance (b) Global volume fraction (c) Maximum stress

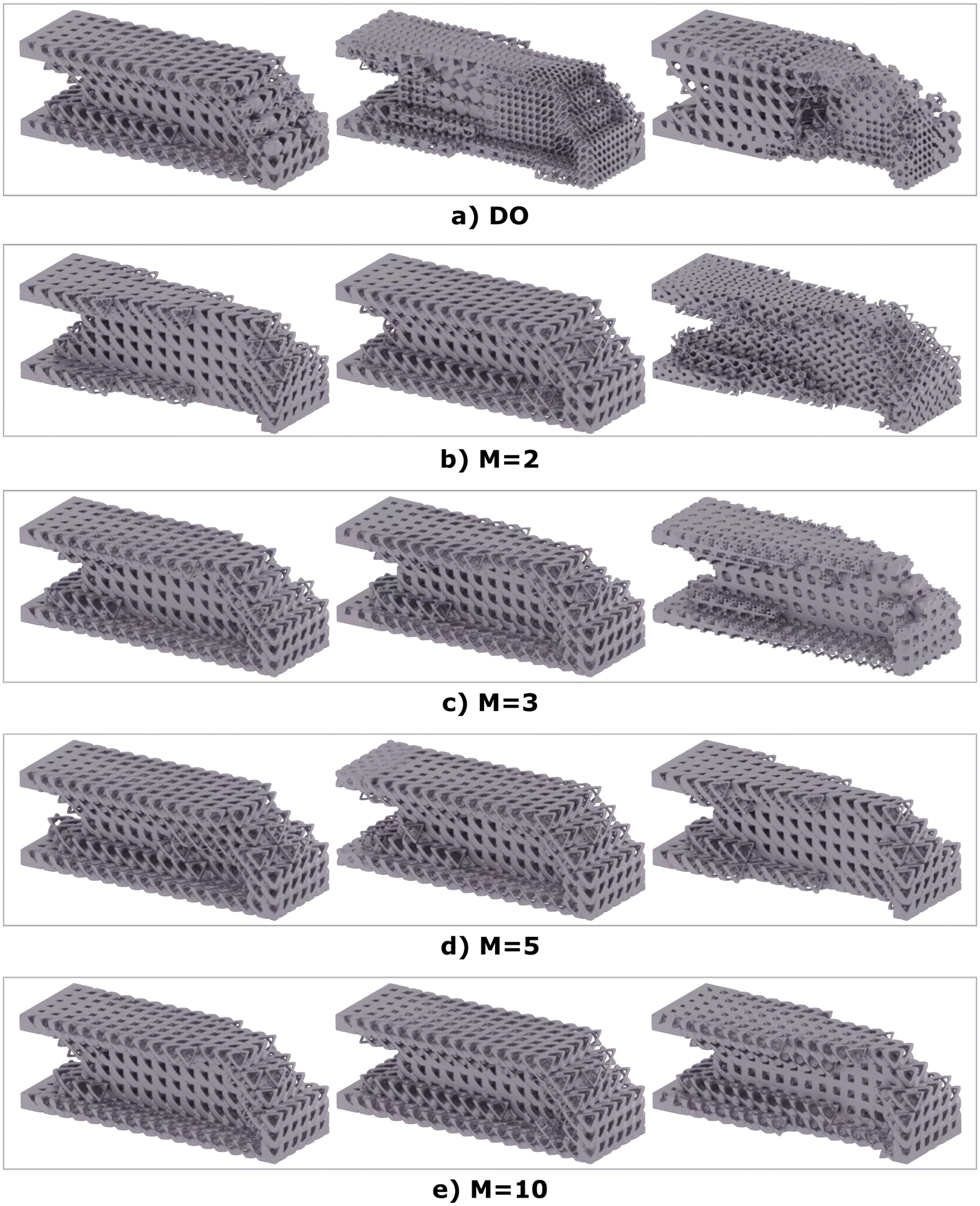
Figure 7.15. Comparison of the small cantilever results from 1 fixed and 10 random initializations. The volume constraint is $V^*_{Global} = 0.5$. Using M-interpolation, compliance decreases as M increases, and stresses are generally lower than those from the DO method.



(a) Compliance (b) Global volume fraction (c) Maximum stress

Figure 7.16. Comparison of the mid-sized cantilever results from 1 fixed and 10 random initializations. The volume constraint is $V^*_{Global} = 0.3$.

In the referenced work, the authors stated that the nonlinearity of stress-based design could cause convergence issues and, therefore, they utilized volume-constrained compliance minimization to optimize the actuator. In contrast, we show that stress constraints can be included on top of several other volume constraints without difficulty.



a) DO

b) M=2

c) M=3

d) M=5

e) M=10

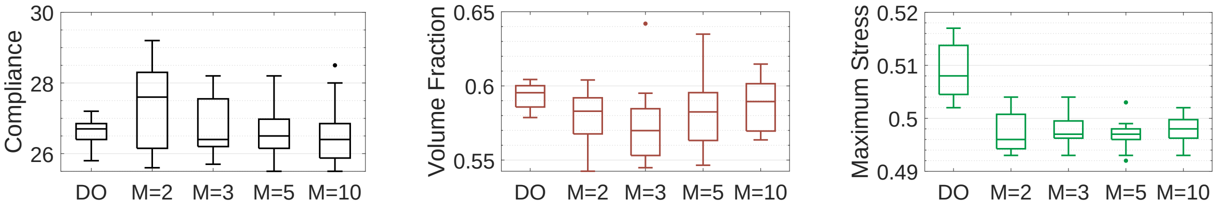
Figure 7.17. Random results of the mid-sized cantilever example using different initializations of the design variables. The M-latent interpolation approach (b-e) achieves smoother geometrical grading than DO (a).

Our stress-constrained compliance minimization problem is defined as:

$$
\begin{aligned}
&\underset{\boldsymbol{z}}{\text{minimize}} && f = \frac{1}{2}\sum_{e=1}^{N_{el}} x_e \mathbf{u}_e^T \mathbf{k}_e^H(\hat{z}_e)\mathbf{u}_e, \\
&\text{subject to} && \mathbf{K}(\hat{\boldsymbol{z}})\mathbf{U} = \mathbf{F}, \\
& && g_1 = V_{Global}(\hat{\boldsymbol{z}})/V_{Global}^* - 1 \leq 0, \\
& && g_2 = v_{max}^{ks}(\hat{\boldsymbol{z}})/v_{max}^* - 1 \leq 0, \\
& && g_3 = 1 - v_{min}^{ks}(\hat{\boldsymbol{z}})/v_{min}^* \leq 0, \\
& && g_4 = \sigma_{max}^{ks}(\hat{\boldsymbol{z}})/\sigma_{max}^* - 1 \leq 0, \\
& && 0 \leq \boldsymbol{z} \leq 1.
\end{aligned}
\tag{7.19}
$$

We fix the macrostructure of the actuator, hence neglecting x_e and Eq. 7.14 while adding a lower bound on the allowable volume fractions, v_{min}^*, using KS aggregation, v_{min}^{ks}. The stress constraint is g_4, where $\sigma_{max}^{ks}(\hat{\boldsymbol{z}})$ is the aggregated maximum von Mises stress (using homogenization-based analysis), and σ_{max}^* is the stress upper limit. To calculate the von Mises stresses and their sensitivities using homogenization-based analysis, we follow Ref. [2].

Despite the additional complexity of a stress constraint, we see similar trends as in the earlier compliance examples. In Fig. 7.18, we can also observe that compliance decreases with greater numbers of sets of latent variables, M. With a higher value of V_{Global}^* (0.6) and without concurrent macroscale design, nearly all of the designs satisfy the volume constraint. Furthermore, the actuator results underscore the importance of encouraging smooth grading in heterogeneous structures: all structures using the DO method fail to meet the stress constraint, while the majority of the M-interpolated designs achieve stresses below the allowable limit.



(a) Compliance (b) Global volume fraction (c) Maximum stress

Figure 7.18. Comparison of the stress-constrained actuator results from 1 fixed and 10 random initializations. The volume constraint is $V^*_{Global} = 0.6$, and the constraint on KS aggregated stress is $\sigma^*_{max} = 0.4$ (which corresponds to the actual value of approximately 0.5 without KS aggregation, plotted).

In Fig. 7.20, we highlight an impressive advantage of our proposed multiclass framework: *by performing design in the multiclass latent space, we can discover optimal and smooth interpolations between different morphological classes of microstructures.* For this actuator case study, our framework creates designs with low compliance and stresses by placing trusses near the bottom of the structure and isosurfaces at the top. At intermediate locations, the WGAN-GP model generates new hybrids of the two classes such that all neighboring microstructures are well-connected. This is achieved without any user intervention, eliminating the biases of pre-defined microstructure classes while expanding the freedom of functionally graded designs. It is also done without explicit connectivity constraints.

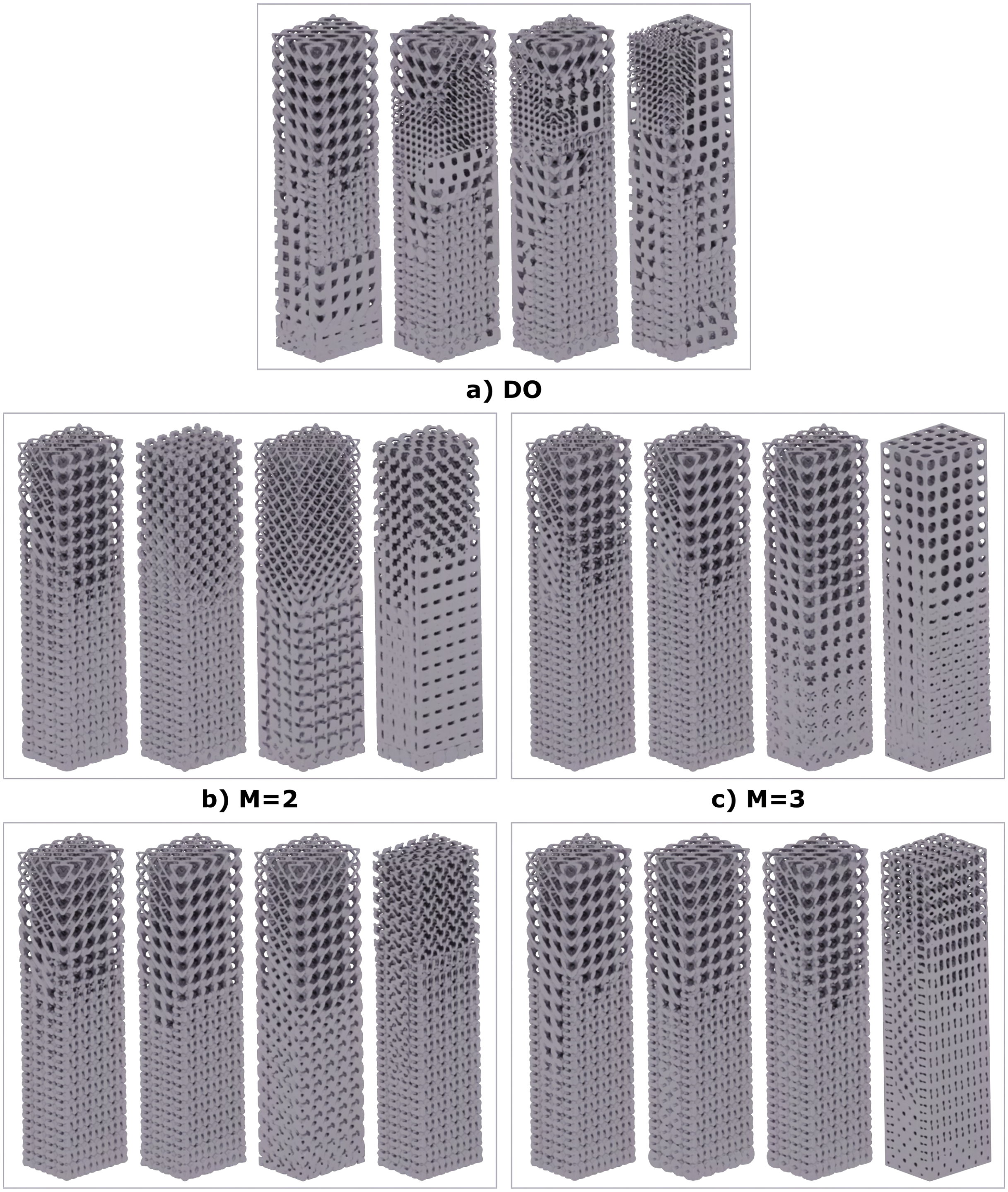
Finally, we compare the computational cost of each synthesis method by averaging the runtimes over 11 different initializations. On a PC desktop with an 8-core Intel i7-6900K and one NVIDIA GeForce GTX 1050 Ti GPU, the stress-constrained design using the DO method takes an average of 8.0 minutes. Meanwhile, the M-latent interpolation approach takes an average of 7.72 minutes (3.8% faster than DO) for $M = 2$, 7.95 for $M = 3$, 7.97 for $M = 5$, and 8.21 for $M = 10$ (2.4% slower than DO). Not surprisingly, using $M = 10$ increases the cost over DO since it involves more design variables and slightly more computations due to interpolation scheme (Eq. 7.16). However, for $M < 10$, we find that the design process is slightly faster than DO while achieving similar compliance and lower maximum stress (see Table 7.4). For a large-scale problem, we would expect the multiclass design with global interpolation to be even more advantageous.

Table 7.4. For the actuator design, comparison of the mean compliance, von Mises stress, and runtime using different synthesis approaches.

	Mean Compliance	Mean Max. Stress	Mean Runtime (min)
DO	26.6	0.509	8.02
$M = 2$	27.3	0.497	7.72
$M = 3$	26.7	0.498	7.95
$M = 5$	26.7	0.497	7.97
$M = 10$	26.6	0.498	8.21

7.9. Discussion

Although our case studies are inspiring and illustrate the potential of the proposed framework, we point to a few areas of improvement that could be addressed in future works. One is the entangled latent space due to the fact that GANs do not regularize the random distribution of the learned space. We noted that this could be a cause of the coarseness of the grading in the DO results (Sec. 7.8.1). In Ref. [1], it was suggested that training a VAE and a property regressor simultaneously can force the latent space to be mechanically meaningful. However, as we showed in Sec. 7.5.3, a vanilla VAE struggles to perform well on our multiclass dataset. To pursue this route, it may be



a) DO

b) M=2

c) M=3

d) M=5

e) M=10

Figure 7.19. Random results of the actuator example using different initializations of the design variables. (a) Results using the DO synthesis approach. (b-e) Results via the M-latent interpolation approach.

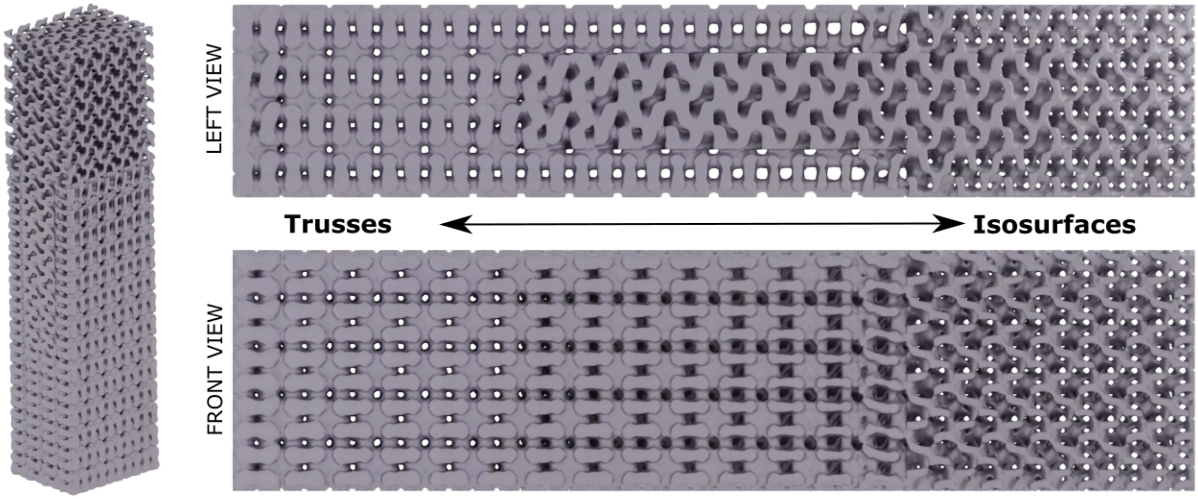


Figure 7.20. Interpolation between trusses and isosurfaces occurs naturally in the latent space of WGAN-GP to create multiclass, functionally graded structures. This particular design was performed using the latent interpolation synthesis approach with $M = 5$.

necessary to employ advanced forms of VAEs that can learn multimodal distributions, such as Wasserstein autoencoders [261].

Second is the lack of physically meaningful constraints when generating our shapes, leading to infeasible unit cells (Sec. 7.5.4) and a lack of synthesized shapes with low volume fractions (Sec. 7.8.1). It is possible to include physics-based constraints as diversity or performance-based quality metrics during the training of GANs, like in Ref. [217]. These would require developing metrics based on the manufacturability of designs, e.g., overhang, minimum feature size, surface area, etc., or the anisotropy, stiffness, Poisson's ratio, etc., of a unit cell. Such quantities may be useful “flags” to steer the model into generating unit cells that are both diverse and feasible. Another solution is to incorporate rigorous topological constraints, such as peristent homology [262], into the training of the generative model to guarantee the continuity of the generated unit cells. Alternatively, quality indicators could be integrated into design synthesis itself via methods utilizing Bayesian approaches [55, 263] or reinforcement learning [187]. These venture outside the realm of conventional TO and may have their own challenges when applied to heterogeneous structures, but are worth future exploration.

Improving the concurrent design of the two scales, macro and micro, is also a valuable direction for future works. In this work, we utilized the filter in Eq. 7.14, which removes entire unit cells and results in dangling features and macrostructures with jagged edges. TO methods that produce smooth contours, such as conformal mapping [57, 182], dehomogenization with function-based mapping [251], approximate level set extraction [264] and cut elements [219], could instead be integrated into our framework to achieve aesthetically pleasing designs.

7.10. Conclusions

To break through the low scalability and inadequate generality of designing heterogeneous multiscale structures via topology optimization methods, we proposed a data-driven framework that merges diverse classes of unit cell geometries under a unified latent representation, thus enabling efficient synthesis of novel, smoothly graded and multiclass designs. Unlike existing works that are restricted to only one traditional class (e.g., trusses, isosurfaces, voxels, or level sets) and limited to a small, pre-defined set of unit cells, our framework synthesizes novel microstructures that are hybrids of the initial seed (i.e., basis) classes. Therefore, we can reach an unprecedented level of efficiency and design freedom in functionally graded design. For example, compared to our own previous work in Chapter 6, we expanded the number of basis classes from 5 to 50, all while decreasing the optimization runtime from 12 minutes for a 2D example with 40×16 (640) unit cells to less than 10 minutes for 3D mid-sized cantilever with $18 \times 6 \times 6$ (648) microstructures. This is accomplished by the three major components in our work: 1) quality-augmented diverse subset selection and data expansion via multiclass blending, 2) generative deep learning via a WGAN-GP, and 3) multiclass design synthesis with global mapping of small sets of latent variables (M-latent interpolation).

A critical step in data-driven design is the acquisition of a dataset that is well-suited to the desired design problems. However, creating a dataset that is both diverse and high quality, so that it does not skew the later deep learning and design synthesis tasks, is difficult under finite computational resources. For this, we presented a probabilistic subset selection method that can automatically choose unit cells classes that cover diverse shapes and properties while also being high quality in terms of their ability to aid deep learning and functionally graded design. The method can simultaneously consider multiple classes by first transforming the shapes into SDFs. We then proposed that a multiclass shape blending scheme can quickly expand a subset of 50 seed classes into a large dataset containing over 61k trusses, isosurfaces, and hybrids of both. Even though two major morphology types, trusses and isosurfaces, were demonstrated here, the proposed methods can easily accommodate a wide range of arbitrary unit cell classes, including those selected by a designer's expertise, once they are converted into SDFs.

For the generative learning in this work, we tested both VAE and a stabler form of GAN, the WGAN-GP, and found that the latter is capable of generating highly realistic truss and isosurface unit cells, as well as any interpolated hybrids between those two classes, hence avoiding mode collapse. Since WGAN-GP maps the high dimensional unit cells into a 10-dimensional latent space, it provides a compact yet expressive design representation that enables us to automate the choice of whether trusses, isosurfaces, or hybrids should be used in structural design. This resolves one of the most overlooked limitations of existing state-of-the-art: the resounding impact that a user's *a priori* bias or choice of initial seed designs can have on the final structure.

Finally, we introduced a multiscale design synthesis approach that optimizes multiple sets of latent variables, i.e., a small number of microstructures, which are then interpolated into the global structure. Compared to optimizing the latent variables of every microstructure, these interpolated designs achieve similar design performance while reducing stress concentrations and requiring less computational cost. As verified by the compliance minimization case studies, including an actuator design with stress constraints, the proposed methods show promise for increasing the manufactuability and practicality of data-driven design in real-world applications. Overall, our framework encompasses a wide design space that merges multiple classes of unit cells and is capable of generating well-connected heterogeneous structures with desirable performance while remaining highly efficient.

Part 4

Conclusions and Appendices

CHAPTER 8

Conclusions

The paradigm shift from traditional optimization methods to large-scale, data-driven frameworks in the heterogeneous metamaterials and multiscale topology optimization (TO) fields have unlocked the possibility of impactful designs like strong-yet-light vehicles, energy absorbing components, strain cloaking devices, soft robots, and more. Open source data sharing communities like MaterialsMine are growing alongside them. Despite the rise in interest, data-driven methods are, for the most part, still confined to linear elastic problems, whereas real-world applications involve highly complex and computationally intensive simulations with nonlinearities, dynamics or multiphysics. Before data-driven approaches are viable in these scenarios, it is critical to improve their scalability and generality.

Data-driven frameworks for heterogeneous structures share one major goal: to efficiently assemble a design with high performance and connectivity. They are composed of three main elements: data acquisition and representation, mechanistic-based learning, and multiscale structure synthesis. The three are deeply interconnected – the quality of a dataset influences the coverage of a generative model and the prediction accuracy of a surrogate model, both of which then impact the effectiveness of the synthesis algorithm. However, many existing methods focus on the last two steps without regard for the compounding effect of the dataset itself. In addition, connectivity is enforced by either limiting the designs to a small set of pre-defined geometry classes or by using costly combinatorial methods; these strategies are suboptimal or inefficient.

This dissertation addresses the challenges in the acquisition, representation, and synthesis steps of data-driven design for multiscale heterogeneous structures. The fundamental hypothesis underpinning our work is that a diverse subset of unit cells can: 1) alleviate the burden of search and optimization algorithms by eliminating redundant or biased samples, hence increasing scalability, 2) still cover the largest possible space of both shapes and properties, therefore heightening generality, and 3) serve as seeds for generating novel structure with enhanced design freedom and performance. The dissertation has accomplished this through the contributions summarized below:

- In Chapter 3, we proposed a multi-material interpolation scheme, which generalizes to a function that can map any continuous representation of unit cell classes into a global macrostructure, i.e., perform multiclass interpolation.
- We shone a light on the pressing need to improve data acquisition in Chapter 4 by presenting two independent methods based on existing state-of-the-art. We collected truss data via property-driven sampling, and isosurface data through a shape-driven approach, which demonstrated how diversity in property space does not guarantee diversity in shapes, and vice versa. Although these two acquisition methods can be employed to collect data in other works, we strongly encourage them to be supplemented by the contribution of the next task: diverse subset selection in joint spaces.
- Given any existing dataset, we proposed METASET in Chapter 5, an automated down-selection method to distill unit cell subsets that are diverse in both shapes

and properties. Using probabilistic Determinantal Point Processes (DPPs), we introduced diversity metrics that quantify the similarities between any continuous representations of shapes or properties. For the first time in metamaterials design, we showcase "top-down" data-driven design examples that affirm our hypothesis: small yet diverse subsets significantly improve the scalability of design algorithms, as well as the connectivity and performance of heterogeneous structures.

- By improving the diversity within our datasets, we are also able to increase the generality of design representations and synthesis methods. In Chapter 6, we presented a "bottom-up" data-driven framework that hinges on the idea that functionally graded structures need not be restricted to a pre-defined set of basis unit cell classes. Furthermore, these initial basis classes no longer need to be mutually compatible. We achieved this through a new multiclass shape blending scheme, which allows the creation of novel microstructures beyond the basis set while guaranteeing that they are feasible in terms of self-connectivity and a minimum feature size. By integrating the blending scheme with the multiclass interpolation function from Chapter 3, we are able to re-parameterize the unit cells into the low-dimensional blending parameters, and to train simple property prediction neural networks. All of these parts are integrated into a data-driven multiscale TO framework that retains nearly the same efficiency as a single-scale TO method. No additional constraints are necessary to guarantee that the optimal structures are functionally graded and well-connected. Moreover, we show that generality can be linked to diversity: even a handful of diverse basis classes can achieve high performance across a variety of problems. The framework of this particular chapter is recommended for applications that need lower overhead cost (the neural network is simple and quick to train) at some loss of generality (a small number of basis classes are linearly interpolated for design).

- Finally, all of the proposed methods above culminate in Chapter 7's generative learning-enabled framework. Our framework here aims to cover a design space far larger than that of the previous task, or indeed any other functionally graded design method, by merging different classes of unit cells into the same pipeline. Towards this end, we first demonstrated that METASET from Chapter 5 can be extended to include class-agnostic similarity metrics as well as geometric quality metrics, such that a diverse and high quality set of 50 trusses and isosurfaces can be simultaneously and autonomously selected. We proposed that a generative model, WGAN-GP, can be trained to unify trusses, isosurfaces, and any hybrids of the two, into the same latent space for functionally graded design. This can be accomplished by rapidly collecting a large shape training dataset using the multiclass shape blending scheme of Chapter 6. Thus, while our shape blending scheme is directly utilized in the synthesis step of the previous chapter to linearly interpolate new microstructures, it is embedded into the data acquisition step here to guide, but not restrict, the novel designs synthesized by the WGAN-GP model. With the latent variables learned by WGAN-GP as design variables, we finally propose a design synthesis method that achieves exceptionally smooth grading between multiple classes – and, subsequently, lower stress concentrations – by utilizing the multiclass interpolation scheme (Chapter 3) to map small sets of latent variables into the global structure. This approach is highly general, efficient, and extensible to a wide variety of applications, including nonlinear ones, due to its roots in conventional TO formulations and the inclusion of a large number of diverse and high quality basis classes. By blurring the boundaries between traditional geometric classes, our framework encompasses a wider design space of heterogeneous multiscale structures with, to our knowledge, significantly lower computational cost than the state-of-the-art.

8.1. Future Directions

The contributions of this dissertation have taken a step toward realizing the data-driven design of heterogeneous structures for real-world applications. There is still much room for future works to build upon this work, including:

- **Adaptive data expansion with diversity metrics.** Our work in Chapter 5, METASET, is a *down-sampling* approach that requires a pre-existing ground dataset that contains both shapes and properties. In applications where the physics are, e.g., dynamic, this is often infeasible. Using the theoretical core of METASET, the opposite can be achieved to intelligently build a diverse and high quality dataset when starting from little or no property data. By integrating data acquisition with model training, one could also answer the question: "How much data is enough?"
- **Improved quality metrics.** In this thesis, relatively simple measures of feasibility and quality are used, e.g., minimum feature size and volume fraction range. There is a need to develop quality metrics that further improve the effectiveness of synthesis as well as the feasibility or manufacturability of the designs. For

manufacturing in particular, one could take inspiration from those used in TO, such as the overhang constraints for additive manufacturing.

- **Finer control over multiclass shape blending.** An advantage of our shape blending scheme is that it needs no constraints or inner optimization loops when integrated into TO. It naturally creates feasible new unit cells and functional grading by using a weighted sum and enforcing a lower feasible bound. However, in some cases, close control over the blending might be desired. For example, our method does not guarantee that the mechanical properties of neighboring microstructures are also functionally graded. Future works may build upon our work by developing constrained shape blending.
- **Multiclass generative deep learning with physics-based and geometric constraints.** As a community governed by physics, it follows that our deep learning models should be too. In our work, we proposed generative deep learning based on shapes only. However, we also showed some flaws to this approach: namely that the generated designs may not be feasible from a geometric or mechanical perspective. By incorporating more rigorous definitions of "feasibility" and encouraging the generative models to learn feasible designs, the effectiveness of data-driven methods can be improved even further. To accomplish this, innovative work at the intersection of computational design and computer science is a necessary but highly promising future work.
- **Integration of data-driven frameworks with state-of-the-art topology optimization.** TO methods are bounded by computational efficiency and the

curse of dimensionality, while data-driven frameworks lie mostly stagnant in linear elastic problems. More effort needs to be undertaken to merge the advantages of both through “bottom-up” data-driven frameworks. By holding hands, data-driven techniques could take inspiration from TO methods that include geometric and/or material nonlinearity, dynamic, or multiphysics formulations. Such a task would also require the development of accurate physics-based neural networks that can predict these complex physics, but could lift data-driven methods up as an unmatched choice for realistic applications.

Emphasizing the impact of data diversity and quality on the scalability and generality of data-driven design, this dissertation has addressed pertinent challenges in data acquisition, representation, and design synthesis. Through diverse subset selection, multiclass shape blending, and general synthesis frameworks, we have pushed the boundaries of computational design for heterogeneous multiscale structures. However, we have also raised critical new research questions to explore, and look forward to future works that unleash the potential of our work to general and truly impactful, real-world applications.

APPENDIX A

Algorithms

A.1. Chapter 5

Algorithm 1: METASET algorithm. After calculating the similarity kernels, a polynomial-time greedy maximization of the gain on the weighted combination of diversity in shape and property spaces is performed. The output is a subset of unit cells such that the joint diversity is maximized.

```
Data: Ground set $G$ of size $N$ of all unit cells;
Result: Subset $M$ of size $N_M$;
Calculate shape and property similarity kernels, $L_S$ and $L_P$;
Calculate joint similarity kernel $L$;
Find subset $M$;
    $M \leftarrow \emptyset$;
while $|M| \neq N_M$ do
    Pick an item $G_i$ that maximizes $\delta f(M \cup i)$;
    $M = M \cup \{G_i\}$;
    $G = G - G_i$;
end while
return $M$;      ▷ Use as input to downstream task such as data-driven
  design or machine learning
```

A.2. Chapter 6

Algorithm 2: Adaptive scheme to decrease volume fraction limits during concurrent multiscale design. i denotes the number of the current iteraton.

```
Require: $V^*_{Global,i-1}, V^*_{BESO,i-1}, V_{Global}, V_{BESO}$;
if ($i \mod 10) = 0$ and $V^*_{Global,i-1} > V^*_{Global}$ then
    $V^*_{Global,i} \leftarrow \min(V_{Global}, V^*_{Global,i-1}) - 0.025$;
end if
if ($i \mod 10) = 0$ and $V_{BESO} \leq V^*_{BESO,i-1}$ then
    $V^*_{BESO,i} \leftarrow V^*_{BESO,i-1} - 0.005$;
end if
return $V^*_{Global,i}, V^*_{BESO,i}$;                    ▷ Updated volume constraints
```

Algorithm 3: Concurrent design framework for multiclass functionally graded structures. If there are no volume constraints, ignore Line 9. If the macrostructure is fixed, ignore Line 11.

```
   Initialize: design variables $\mathbf{c}, \mathbf{v}, \boldsymbol{\xi}, \mathbf{x}$;
               volume constraints $V^*_{Global,0}, V^*_{BESO,0}$;
               weight on low-diversity penalty $k$;
 1 while change in design > tol do
 2     $i \leftarrow i + 1$ ;                                  ▷ Iteration counter
 3     for each macro-element $e$ do
 4         obtain $\hat{\mathbf{c}}_e$ (Eqs. 6.6, 6.7);
 5         find $\mathbf{\Phi}_e$ and $t_e$ so that microstructure has volume $\hat{v}_e$ (Eq. 6.8);
 6         approximate $\hat{v}^a_e$ (Eqs. 6.11, 6.12);
 7         predict effective stiffness $\mathbf{C}^H_e = NN(\bar{\mathbf{c}}_e, \hat{v}_e)$ and obtain $\mathbf{k}_e$;
 8     end for
 9     update volume fraction constraint limits (Algorithm 2);
10     compute objective, constraints and sensitivities (Sec. 6.6.3);
11     update macroscale design $\mathbf{x}$ with BESO;
12     update other variables $\mathbf{c}, \mathbf{v}, \boldsymbol{\xi}$ with MMA;
13 end while
14 return optimal multiclass functionally graded design;
```

APPENDIX B

Related Publications

Publications related to this dissertation have been published in journals and peer-reviewed conferences. They are:

Chapter 3

- Shintani, K., Chan, Y.-C. and Chen, W., 2017, June. Robust multi-material topology optimization for lattice structure under material uncertainties. In World Congress of Structural and Multidisciplinary Optimisation (pp. 1110-1123).
- Chan, Y.-C., Shintani, K. and Chen, W., 2019. Robust topology optimization of multi-material lattice structures under material and load uncertainties. Frontiers of Mechanical Engineering, 14(2), pp.141-152.

Chapter 5

- Chan, Y.-C., Ahmed, F., Wang, L. and Chen, W., 2020, August. METASET: An automated data selection method for scalable data-driven design of metamaterials. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 84003, p. V11AT11A021).
- Chan, Y.-C., Ahmed, F., Wang, L. and Chen, W., 2021. METASET: Exploring shape and property spaces for data-driven metamaterials design. Journal of Mechanical Design, 143(3).

Chapter 6

- Chan, Y.-C., Da, D., Wang, L. and Chen, W., 2022. Remixing functionally graded structures: data-driven topology optimization with multiclass shape blending. Structural and Multidisciplinary Optimization, 65(5), pp.1-22.

Chapter 7

- In preparation.

Additional Related Works

- Bostanabad, R., Chan, Y.-C., Wang, L., Zhu, P. and Chen, W., 2019. Globally approximate gaussian processes for big data with application to data-driven metamaterials design. Journal of Mechanical Design, 141(11).
- Wang, L., Chan, Y.-C., Liu, Z., Zhu, P. and Chen, W., 2020. Data-driven metamaterial design with Laplace-Beltrami spectrum as "shape-DNA". Structural and multidisciplinary optimization, 61(6), pp.2613-2628.
- Wang, L., Chan, Y.-C., Ahmed, F., Liu, Z., Zhu, P. and Chen, W., 2020. Deep generative modeling for mechanistic-based learning and design of metamaterial systems. Computer Methods in Applied Mechanics and Engineering, 372, p.113377.
- Da, D., Chan, Y.-C., Wang, L. and Chen, W., 2022. Data-driven and topological design of structural metamaterials for fracture resistance. Extreme Mechanics Letters, 50, p.101528.
- Lee, D., Chan, Y.-C., Wang, L., van Beek, A. and Chen, W., 2022. t-METASET: Task-Aware Generation of Metamaterial Datasets by Diversity-Based Active Learning. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (in print). preprint arXiv:2202.10565.

References

[1] Liwei Wang, Yu-Chin Chan, Faez Ahmed, Zhao Liu, Ping Zhu, and Wei Chen. Deep generative modeling for mechanistic-based learning and design of metamaterial systems. *Computer Methods in Applied Mechanics and Engineering*, 372:113377, December 2020. doi: 10.1016/j.cma.2020.113377.

[2] Daicong Da, Yu-Chin Chan, Liwei Wang, and Wei Chen. Data-driven and topological design of structural metamaterials for fracture resistance. *Extreme Mechanics Letters*, page 101528, November 2021. doi: 10.1016/j.eml.2021.101528.

[3] Liwei Wang, Siyu Tao, Ping Zhu, and Wei Chen. Data-driven topology optimization with multiclass microstructures using latent variable gaussian process. *Journal of Mechanical Design*, 143(3), November 2020. doi: 10.1115/1.4048628.

[4] Xianglong Yu, Ji Zhou, Haiyi Liang, Zhengyi Jiang, and Lingling Wu. Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review. *Progress in Materials Science*, 94:114–173, May 2018. doi: 10.1016/j.pmatsci.2017.12.003.

[5] Muamer Kadic, Tiemo Bückmann, Robert Schittny, and Martin Wegener. Metamaterials beyond electromagnetism. *Reports on Progress in Physics*, 76(12):126501, November 2013. doi: 10.1088/0034-4885/76/12/126501.

[6] Zuojia Wang, Feng Cheng, Thomas Winsor, and Yongmin Liu. Optical chiral metamaterials: a review of the fundamentals, fabrication methods and applications.

Nanotechnology, 27(41):412001, September 2016. doi: 10.1088/0957-4484/27/41/412001.

[7] Aamer Nazir, Kalayu Mekonen Abate, Ajeet Kumar, and Jeng-Ywan Jeng. A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures. *The International Journal of Advanced Manufacturing Technology*, 104(9-12):3489–3510, July 2019. doi: 10.1007/s00170-019-04085-3.

[8] Min Yang and Ping Sheng. Sound absorption structures: From porous media to acoustic metamaterials. *Annual Review of Materials Research*, 47(1):83–114, July 2017. doi: 10.1146/annurev-matsci-070616-124032.

[9] Steven A. Cummer, Johan Christensen, and Andrea Alù. Controlling sound with acoustic metamaterials. *Nature Reviews Materials*, 1(3), February 2016. doi: 10.1038/natrevmats.2016.1.

[10] Osama R. Bilal, David Ballagi, and Chiara Daraio. Architected lattices for simultaneous broadband attenuation of airborne sound and mechanical vibrations in all directions. *Physical Review Applied*, 10(5), November 2018. doi: 10.1103/physrevapplied.10.054060.

[11] Diab W. Abueidda, Iwona Jasiuk, and Nahil A. Sobh. Acoustic band gaps and elastic stiffness of PMMA cellular solids based on triply periodic minimal surfaces. *Materials &amp Design*, 145:20–27, May 2018. doi: 10.1016/j.matdes.2018.02.032.

[12] Seung Ki Moon, Yu En Tan, Jihong Hwang, and Yong-Jin Yoon. Application of 3d printing technology for designing light-weight unmanned aerial vehicle wing structures. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 1(3):223–228, July 2014. doi: 10.1007/s40684-014-0028-x.

[13] Alessandro Spadoni and Massimo Ruzzene. Numerical and experimental analysis of the static compliance of chiral truss-core airfoils. *Journal of Mechanics of Materials and Structures*, 2(5):965–981, July 2007. doi: 10.2140/jomms.2007.2.965.

[14] L. E. Murr, S. M. Gaytan, F. Medina, H. Lopez, E. Martinez, B. I. Machado, D. H. Hernandez, L. Martinez, M. I. Lopez, R. B. Wicker, and J. Bracke. Next-generation biomedical implants using additive manufacturing of complex, cellular and functional mesh arrays. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1917):1999–2032, April 2010. doi: 10.1098/rsta.2010.0010.

[15] J. Parthasarathy, B. Starly, and S. Raman. A design for the additive manufacture of functionally graded porous structures with tailored mechanical properties for biomedical applications. *Journal of Manufacturing Processes*, 13, 2011. doi: 10.1016/j.jmapro.2011.01.004.

[16] X. Z. Zhang, M. Leary, and H. P. Tang. Selective electron beam manufactured ti-6al-4v lattice structures for orthopedic implant applications: Current status and outstanding challenges. *Current Opinion in Solid State and Materials Science*, 22, 2018. doi: 10.1016/j.cossms.2018.05.002.

[17] K. J. Maloney, K. D. Fink, and T. A. Schaedler. Multifunctional heat exchangers derived from three-dimensional micro-lattice structures. *International Journal of Heat and Mass Transfer*, 55, 2012. doi: 10.1016/j.ijheatmasstransfer.2012.01.011.

[18] K. N. Son, J. A. Weibel, and V. Kumaresan. Design of multifunctional lattice-frame materials for compact heat exchangers. *International Journal of Heat and Mass Transfer*, 115, 2017. doi: 10.1016/j.ijheatmasstransfer.2017.07.073.

[19] Johannes T.B. Overvelde and Katia Bertoldi. Relating pore shape to the non-linear response of periodic elastomeric structures. *Journal of the Mechanics and Physics of Solids*, 64:351–366, March 2014. doi: 10.1016/j.jmps.2013.11.014.

[20] Bastiaan Florijn, Corentin Coulais, and Martin van Hecke. Programmable mechanical metamaterials. *Physical Review Letters*, 113(17), October 2014. doi: 10.1103/physrevlett.113.175503.

[21] Giorgio Oliveri and Johannes T.B. Overvelde. Inverse design of mechanical metamaterials that undergo buckling. *Advanced Functional Materials*, 30(12):1909033, feb 2020. doi: 10.1002/adfm.201909033.

[22] Ian Maskery, Alexandra Hussey, Ajit Panesar, Adedeji Aremu, Christopher Tuck, Ian Ashcroft, and Richard Hague. An investigation into reinforced and functionally graded lattice structures. *Journal of Cellular Plastics*, 53(2):151–165, July 2016. doi: 10.1177/0021955x16639035.

[23] I. Maskery, N.T. Aboulkhair, A.O. Aremu, C.J. Tuck, I.A. Ashcroft, R.D. Wildman, and R.J.M. Hague. A mechanical property evaluation of graded density al-si10-mg lattice structures manufactured by selective laser melting. *Materials Science and Engineering: A*, 670:264–274, July 2016. doi: 10.1016/j.msea.2016.06.013.

[24] János Plocher and Ajit Panesar. Effect of density and unit cell size grading on the stiffness and energy absorption of short fibre-reinforced functionally graded lattice structures. *Additive Manufacturing*, 33:101171, May 2020. doi: 10.1016/j.addma.2020.101171.

[25] Simon R.G. Bates, Ian R. Farrow, and Richard S. Trask. 3d printed polyurethane honeycombs for repeated tailored energy absorption. *Materials &amp Design*, 112:

172–183, December 2016. doi: 10.1016/j.matdes.2016.08.062.

[26] Dixon M Correa, Timothy Klatt, Sergio Cortes, Michael Haberman, Desiderio Kovar, and Carolyn Seepersad. Negative stiffness honeycombs for recoverable shock isolation. *Rapid Prototyping Journal*, 21(2):193–200, March 2015. doi: 10.1108/rpj-12-2014-0182.

[27] S. Shan, S. H. Kang, and J. R. Raney. Multistable architected materials for trapping elastic strain energy. *Advanced Materials*, 27, 2015. doi: 10.1002/adma.201501708.

[28] Thomas S. Lumpe and Kristina Shea. Computational design of 3d-printed active lattice structures for reversible shape morphing. *Journal of Materials Research*, May 2021. doi: 10.1557/s43578-021-00225-2.

[29] T. Stankovic, J. Mueller, and P. Egan. A generalized optimality criteria method for optimization of additively manufactured multimaterial lattice structures. *Journal of Mechanical Design*, 137, 2015. doi: 10.1115/1.4030995.

[30] M. J. Mirzaali, S. Janbaz, M. Strano, L. Vergani, and A. A. Zadpoor. Shape-matching soft mechanical metamaterials. *Scientific Reports*, 8(1), January 2018. doi: 10.1038/s41598-018-19381-3.

[31] D. Chen and X. Zheng. Multi-material additive manufacturing of metamaterials with giant, tailorable negative poisson's ratios. *Scientific Reports*, 8, 2018. doi: 10.1038/s41598-018-26980-7.

[32] Anders Clausen, Fengwen Wang, Jakob S. Jensen, Ole Sigmund, and Jennifer A. Lewis. Topology optimized architectures with programmable poisson's ratio over large deformations. *Advanced Materials*, 27(37):5523–5527, August 2015. doi: 10.1002/adma.201502485.

[33] Krishna Kumar Saxena, Raj Das, and Emilio P. Calius. Three decades of auxetics research - materials with negative poisson's ratio: A review. *Advanced Engineering Materials*, 18(11):1847–1870, June 2016. doi: 10.1002/adem.201600053.

[34] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015. doi: 10.1145/2766926.

[35] Corentin Coulais, Eial Teomy, Koen de Reus, Yair Shokef, and Martin van Hecke. Combinatorial design of textured mechanical metamaterials. *Nature*, 535(7613): 529–532, 2016. doi: 10.1038/nature18960.

[36] I. Maskery, A.O. Aremu, L. Parry, R.D. Wildman, C.J. Tuck, and I.A. Ashcroft. Effective design and simulation of surface-based lattice structures featuring volume fraction and cell type grading. *Materials & Design*, 155:220–232, 2018. doi: 10.1016/j.matdes.2018.05.058.

[37] J. William Boley, Wim M. van Rees, Charles Lissandrello, Mark N. Horenstein, Ryan L. Truby, Arda Kotikian, Jennifer A. Lewis, and L. Mahadevan. Shape-shifting structured lattices via multimaterial 4d printing. *Proceedings of the National Academy of Sciences*, 116(42):20856–20862, October 2019. doi: 10.1073/pnas.1908806116.

[38] Jesse Callanan, Oladapo Ogunbodede, Maulikkumar Dhameliya, Jun Wang, and Rahul Rai. Hierarchical combinatorial design and optimization of quasi-periodic metamaterial structures. In *Volume 2B: 44th Design Automation Conference*. American Society of Mechanical Engineers, 2018. doi: 10.1115/detc2018-85914.

[39] Dennis M. Kochmann, Jonathan B. Hopkins, and Lorenzo Valdevit. Multiscale modeling and optimization of the mechanics of hierarchical metamaterials. *MRS Bulletin*, 44(10):773–781, October 2019. doi: 10.1557/mrs.2019.228.

[40] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2(11), October 2017. doi: 10.1038/natrevmats.2017.66.

[41] Francesco Tamburrino, Serena Graziosi, and Monica Bordegoni. The design process of additively manufactured mesoscale lattice structures: A review. *Journal of Computing and Information Science in Engineering*, 18(4), July 2018. doi: 10.1115/1.4040131.

[42] Gianpaolo Savio, Stefano Rosso, Roberto Meneghello, and Gianmaria Concheri. Geometric modeling of cellular materials for additive manufacturing in biomedical field: A review. *Applied Bionics and Biomechanics*, 2018:1–14, 2018. doi: 10.1155/2018/1654782.

[43] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, Berlin, 2004. doi: 10.1007/978-3-662-05086-6.

[44] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, March 2013. doi: 10.1007/s00158-013-0912-y.

[45] Xiaoyu Zheng, Howon Lee, Todd H. Weisgraber, Maxim Shusteff, Joshua DeOtte, Eric B. Duoss, Joshua D. Kuntz, Monika M. Biener, Qi Ge, Julie A. Jackson, Sergei O. Kucheyev, Nicholas X. Fang, and Christopher M. Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344(6190):1373–1377, June 2014. doi:

10.1126/science.1252291.

[46] Oraib Al-Ketan, Rashid K. Abu Al-Rub, and Reza Rowshan. The effect of architecture on the mechanical properties of cellular structures based on the IWP minimal surface. *Journal of Materials Research*, 33(3):343–359, January 2018. doi: 10.1557/jmr.2018.1.

[47] Diab W. Abueidda, Mete Bakir, Rashid K. Abu Al-Rub, Jörgen S. Bergström, Nahil A. Sobh, and Iwona Jasiuk. Mechanical properties of 3d printed polymeric cellular materials with triply periodic minimal surface architectures. *Materials &amp Design*, 122:255–267, May 2017. doi: 10.1016/j.matdes.2017.03.018.

[48] Sebastian Krödel and Chiara Daraio. Microlattice metamaterials for tailoring ultrasonic transmission with elastoacoustic hybridization. *Physical Review Applied*, 6 (6), December 2016. doi: 10.1103/physrevapplied.6.064005.

[49] Anton Jansson and Lars Pejryd. In-situ computed tomography investigation of the compression behaviour of strut, and periodic surface lattices. In *9th Conference on Industrial Computed Tomography*, 2019. URL `https://www.ndt.net/search/docs.php3?id=23646`.

[50] Ajit Panesar, Meisam Abdi, Duncan Hickman, and Ian Ashcroft. Strategies for functionally graded lattice structures derived using topology optimisation for additive manufacturing. *Additive Manufacturing*, 19:81–94, January 2018. doi: 10.1016/j.addma.2017.11.008.

[51] I. Maskery, L. Sturm, A.O. Aremu, A. Panesar, C.B. Williams, C.J. Tuck, R.D. Wildman, I.A. Ashcroft, and R.J.M. Hague. Insights into the mechanical properties of several triply periodic minimal surface lattice structures made by polymer additive

manufacturing. *Polymer*, 152:62–71, September 2018. doi: 10.1016/j.polymer.2017.11.049.

[52] Oliver Weeger, Bharath Narayanan, and Martin L. Dunn. Isogeometric shape optimization of nonlinear, curved 3d beams and beam structures. *Computer Methods in Applied Mechanics and Engineering*, 345:26–51, March 2019. doi: 10.1016/j.cma.2018.10.038.

[53] Ashish Gupta, George Allen, and Jarek Rossignac. QUADOR: QUADric-of-revolution beams for lattices. *Computer-Aided Design*, 102:160–170, September 2018. doi: 10.1016/j.cad.2018.04.015.

[54] Osama Abdeljaber, Onur Avci, and Daniel J. Inman. Genetic algorithm use for internally resonating lattice optimization: Case of a beam-like metastructure. In *Conference Proceedings of the Society for Experimental Mechanics Series*, pages 289–295. Springer International Publishing, 2016. doi: 10.1007/978-3-319-29751-4_29.

[55] Conner Sharpe, Carolyn Conner Seepersad, Seth Watts, and Dan Tortorelli. Design of mechanical metamaterials via constrained bayesian optimization. In *Volume 2A: 44th Design Automation Conference*. American Society of Mechanical Engineers, August 2018. doi: 10.1115/detc2018-85270.

[56] Hongming Zong, Hongying Zhang, Yiqiang Wang, Michael Yu Wang, and Jerry Y.H. Fuh. On two-step design of microstructure with desired poisson's ratio for AM. *Materials & Design*, 159:90–102, 2018. doi: 10.1016/j.matdes.2018.08.032.

[57] Panagiotis Vogiatzis, Ming Ma, Shikui Chen, and Xianfeng David Gu. Computational design and additive manufacturing of periodic conformal metasurfaces by synthesizing topology optimization with conformal mapping. *Computer Methods in*

Applied Mechanics and Engineering, 328:477–497, 2018. doi: 10.1016/j.cma.2017.09.012.

[58] Hui Liu, Hongming Zong, Ye Tian, Qingping Ma, and Michael Yu Wang. A novel subdomain level set method for structural topology optimization and its application in graded cellular structure design. *Structural and Multidisciplinary Optimization*, 60(6):2221–2247, June 2019. doi: 10.1007/s00158-019-02318-3.

[59] Zongliang Du, Xiao-Yi Zhou, Renato Picelli, and H. Alicia Kim. Connecting microstructures for multiscale topology optimization with connectivity index constraints. *Journal of Mechanical Design*, 140(11), 2018. doi: 10.1115/1.4041176.

[60] Eric Garner, Helena M.A. Kolken, Charlie C.L. Wang, Amir A. Zadpoor, and Jun Wu. Compatibility in microstructural optimization for additive manufacturing. *Additive Manufacturing*, 26:65–75, March 2019. doi: 10.1016/j.addma.2018.12.007.

[61] Jiadong Deng, Claus B. W. Pedersen, and Wei Chen. Connected morphable components-based multiscale topology optimization. *Frontiers of Mechanical Engineering*, 14(2):129–140, January 2019. doi: 10.1007/s11465-019-0532-3.

[62] Sheng Chu, Liang Gao, Mi Xiao, and Yan Zhang. Multiscale topology optimization for coated structures with multifarious-microstructural infill. *Structural and Multidisciplinary Optimization*, 61(4):1473–1494, December 2019. doi: 10.1007/s00158-019-02428-y.

[63] E. D. Sanders, A. Pereira, and G. H. Paulino. Optimal and continuous multilattice embedding. *Science Advances*, 7(16):eabf4838, April 2021. doi: 10.1126/sciadv.abf4838.

[64] Yunfeng Luo, Jingyu Hu, and Shutian Liu. Self-connected multi-domain topology optimization of structures with multiple dissimilar microstructures. *Structural and Multidisciplinary Optimization*, 64(1):125–140, February 2021. doi: 10.1007/s00158-021-02865-8.

[65] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015. doi: 10.1145/2766937.

[66] Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. Two-scale topology optimization with microstructures. *ACM Transactions on Graphics*, 36(4):1, 2017. doi: 10.1145/3072959.3095815.

[67] Yan Zhang, Mi Xiao, Liang Gao, Jie Gao, and Hao Li. Multiscale topology optimization for minimizing frequency responses of cellular composites with connectable graded microstructures. 135:106369, January 2020. doi: 10.1016/j.ymssp.2019.106369.

[68] Wei Ma, Feng Cheng, Yihao Xu, Qinlong Wen, and Yongmin Liu. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Advanced Materials*, 31(35):1901111, 2019. doi: 10.1002/adma.201901111.

[69] Darshil Patel, Dustin Bielecki, Rahul Rai, and Gary Dargush. Improving connectivity and accelerating multiscale topology optimization using deep neural network techniques. *Structural and Multidisciplinary Optimization*, 65(4), March 2022. doi: 10.1007/s00158-022-03223-y.

[70] Jun Wang, Wei (Wayne) Chen, Daicong Da, Mark Fuge, and Rahul Rai. IH-GAN: A conditional generative model for implicit surface-based inverse design of cellular structures. *Computer Methods in Applied Mechanics and Engineering*, 396:115060, June 2022. doi: 10.1016/j.cma.2022.115060.

[71] Daniel A. White, William J. Arrighi, Jun Kudo, and Seth E. Watts. Multiscale topology optimization using neural network surrogate models. *Computer Methods in Applied Mechanics and Engineering*, 346:1118–1135, 2019. doi: 10.1016/j.cma.2018.09.007.

[72] Siddhant Kumar, Stephanie Tan, Li Zheng, and Dennis M. Kochmann. Inverse-designed spinodoid metamaterials. *npj Computational Materials*, 6(1), June 2020. doi: 10.1038/s41524-020-0341-6.

[73] Jan-Hendrik Bastek, Siddhant Kumar, Bastian Telgen, Raphaël N. Glaesener, and Dennis M. Kochmann. Inverting the structure–property map of truss metamaterials by deep learning. *Proceedings of the National Academy of Sciences*, 119(1), December 2021. doi: 10.1073/pnas.2111505119.

[74] He Zhao, Xiaolin Li, Yichi Zhang, Linda S. Schadler, Wei Chen, and L. Catherine Brinson. Perspective: NanoMine: A material genome approach for polymer nanocomposites analysis and design. *APL Materials*, 4(5):053204, March 2016. doi: 10.1063/1.4943679.

[75] Akshay Iyer, Yichi Zhang, Aditya Prasad, Siyu Tao, Yixing Wang, Linda Schadler, L. Catherine Brinson, and Wei Chen. Data-centric mixed-variable bayesian optimization for materials design. In *Volume 2A: 45th Design Automation Conference*. American Society of Mechanical Engineers, August 2019. doi: 10.1115/

detc2019-98222.

[76] Xiaolin Li, Zijiang Yang, L. Catherine Brinson, Alok Choudhary, Ankit Agrawal, and Wei Chen. A deep adversarial learning methodology for designing microstructural material systems. In *Volume 2B: 44th Design Automation Conference*. American Society of Mechanical Engineers, August 2018. doi: 10.1115/detc2018-85633.

[77] Zijiang Yang, Xiaolin Li, L. Catherine Brinson, Alok N. Choudhary, Wei Chen, and Ankit Agrawal. Microstructural materials design via deep adversarial learning methodology. *Journal of Mechanical Design*, 140(11), October 2018. doi: 10.1115/1.4041371.

[78] Deborah McGuinness, Cate Brinson, Wei Chen, Chiara Daraio, Cynthia Rudin, Linda Schadler, Rebecca Cowan, Jamie McCusker, Samuel Stouffer, Neha Keshan, et al. Materialsmine: An open-source, user-friendly materials data resource guided by fair principles. 2022.

[79] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017. doi: 10.1016/j.eswa.2016.12.035.

[80] Liang Xia and Piotr Breitkopf. Concurrent topology optimization design of material and structure within fe2 nonlinear multiscale analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 278:524–542, August 2014. doi: 10.1016/j.cma.2014.05.022.

[81] Liang Xia and Piotr Breitkopf. Design of materials using topology optimization and energy-based homogenization approach in matlab. *Structural and Multidisciplinary*

Optimization, 52(6):1229–1241, July 2015. doi: 10.1007/s00158-015-1294-0.

[82] B. Hassani and E. Hinton. A review of homogenization and topology optimization i—homogenization theory for media with periodic structure. *Computers &amp Structures*, 69(6):707–717, December 1998. doi: 10.1016/s0045-7949(98)00131-x.

[83] J. Pinho da Cruz, J.A. Oliveira, and F. Teixeira-Dias. Asymptotic homogenisation in linear elasticity. part i: Mathematical formulation and finite element modelling. *Computational Materials Science*, 45(4):1073–1080, June 2009. doi: 10.1016/j.commatsci.2009.02.025.

[84] Guoying Dong, Yunlong Tang, and Yaoyao Fiona Zhao. A 149 line homogenization code for three-dimensional cellular materials written in matlab. *Journal of Engineering Materials and Technology*, 141(1), 2018. doi: 10.1115/1.4040555.

[85] Weihong Zhang, Gaoming Dai, Fengwen Wang, Shiping Sun, and Hicham Bassir. Using strain energy-based prediction of effective elastic properties in topology optimization of material microstructures. *Acta Mechanica Sinica*, 23(1):77–89, January 2007. doi: 10.1007/s10409-006-0045-2.

[86] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, January 2007. doi: 10.1007/s00158-006-0087-x.

[87] X. S. Zhang, G. H. Paulino, and A. S. Ramos. Multi-material topology optimization with multiple volume constraints: A general approach applied to ground structures with material nonlinearity. *Structural and Multidisciplinary Optimization*, 57, 2018. doi: 10.1007/s00158-017-1768-3.

[88] Jie Gao, Zhen Luo, Liang Xia, and Liang Gao. Concurrent topology optimization of multiscale composite structures in matlab. *Structural and Multidisciplinary Optimization*, 60(6):2621–2651, August 2019. doi: 10.1007/s00158-019-02323-6.

[89] Y. Chen, T. Li, and F. Scarpa. Lattice metamaterials with mechanically tunable poisson's ratio for vibration control. *Physical Review Applied*, 7, 2017. doi: 10.1103/PhysRevApplied.7.024012.

[90] S. Yuan, F. Shen, and J. Bai. 3d soft auxetic lattice structures fabricated by selective laser sintering: Tpu powder evaluation and process optimization. *Materials & Design*, 120, 2017. doi: 10.1016/j.matdes.2017.01.098.

[91] K. Yu, N. X. Fang, and G. Huang. Magnetoactive acoustic metamaterials. *Advanced Materials*, 30, 2018. doi: 10.1002/adma.201706348.

[92] J. Brennan-Craddock, D. Brackett, and R. Wildman. The design of impact absorbing structures for additive manufacture. *Journal of Physics: Conference Series*, 382, 2012.

[93] Z. Ozdemir, A. Tyas, and R. Goodall. Energy absorption in lattice structures in dynamics: Nonlinear fe simulations. *International Journal of Impact Engineering*, 102, 2017. doi: 10.1016/j.ijimpeng.2016.11.016.

[94] F. N. Habib, P. Iovenitti, and S. H. Masood. Fabrication of polymeric lattice structures for optimum energy absorption using multi jet fusion technology. *Materials & Design*, 155, 2018. doi: 10.1016/j.matdes.2018.05.059.

[95] P. Terriault and V. Brailovski. Modeling and simulation of large, conformal, porosity-graded and lightweight lattice structures made by additive manufacturing. *Finite Elements in Analysis and Design*, 138, 2018. doi: 10.1016/j.finel.2017.09.005.

[96] Krishna Kumar Saxena, Raj Das, and Emilio P Calius. 3d printable multimaterial cellular auxetics with tunable stiffness, 2017.

[97] M. Y. Wang and X. Wang. “color” level sets: A multi-phase method for structural topology optimization with multiple materials. *Computer Methods in Applied Mechanics and Engineering*, 193, 2004. doi: 10.1016/j.cma.2003.10.008.

[98] S. Zhou and M. Y. Wang. Multimaterial structural topology optimization with a generalized cahn-hilliard model of multiphase transition. *Structural and Multidisciplinary Optimization*, 38, 2007.

[99] Jonathan D. Hiller and Hod Lipson. Multi material topological optimization of structures and mechanisms. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO '09*. ACM Press, 2009. doi: 10.1145/1569901.1570105.

[100] A. Ramani. Multi-material topology optimization with strength constraints. *Structural and Multidisciplinary Optimization*, 43, 2011. doi: 10.1007/s00158-010-0581-z.

[101] J. Stegmann and E. Lund. Discrete material optimization of general composite shell structures. *International Journal for Numerical Methods in Engineering*, 62, 2005. doi: 10.1002/nme.1259.

[102] A. Gaynor, N. A. Meisel, and C. B. Williams. Multiple-material topology optimization of compliant mechanisms created via polyjet 3d printing. *Journal of Manufacturing Science and Engineering*, 136, 2014. doi: 10.1115/1.4028439.

[103] L. Yin and G. K. Ananthasuresh. Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. *Structural and Multidisciplinary Optimization*, 23, 2001. doi: 10.1007/s00158-001-0165-z.

[104] W. Zuo and K. Saitou. Multi-material topology optimization using ordered simp interpolation. *Structural and Multidisciplinary Optimization*, 55, 2017. doi: 10.1007/s00158-016-1513-3.

[105] S. L. Sing, F. E. Wiria, and W. Y. Yeong. Selective laser melting of lattice structures: A statistical approach to manufacturability and mechanical behavior. *Robotics and Computer-Integrated Manufacturing*, 49, 2018. doi: 10.1016/j.rcim.2017.06.006.

[106] S. M. Ahmadi, R. Hedayati, and R. K. Ashok Kumar Jain. Effects of laser processing parameters on the mechanical properties, topology, and microstructure of additively manufactured porous metallic biomaterials: A vector-based approach. *Materials & Design*, 134, 2017. doi: 10.1016/j.matdes.2017.08.046.

[107] S. I. Park, D. W. Rosen, and S. Choi. Effective mechanical properties of lattice material fabricated by material extrusion additive manufacturing. *Additive Manufacturing*, 1–4, 2014. doi: 10.1016/j.addma.2014.07.002.

[108] S. Chen and W. Chen. A new level-set based approach to shape and topology optimization under geometric uncertainty. *Structural and Multidisciplinary Optimization*, 44, 2011. doi: 10.1007/s00158-011-0660-9.

[109] M. Jansen, G. Lombaert, and M. Diehl. Robust topology optimization accounting for misplacement of material. *Structural and Multidisciplinary Optimization*, 47, 2013. doi: 10.1007/s00158-012-0835-z.

[110] M. Schevenels, B. S. Lazarov, and O. Sigmund. Robust topology optimization accounting for spatially varying manufacturing errors. *Computer Methods in Applied Mechanics and Engineering*, 200, 2011. doi: 10.1016/j.cma.2011.08.006.

[111] M. Jalalpour and M. Tootkaboni. An efficient approach to reliability-based topology optimization for continua under material uncertainty. *Structural and Multidisciplinary Optimization*, 53, 2016. doi: 10.1007/s00158-015-1360-7.

[112] N. Changizi and M. Jalalpour. Robust topology optimization of frame structures under geometric or material properties uncertainties. *Structural and Multidisciplinary Optimization*, 56, 2017. doi: 10.1007/s00158-017-1686-4.

[113] F. Alvarez and M. Carrasco. Minimization of the expected compliance as an alternative approach to multiload truss optimization. *Structural and Multidisciplinary Optimization*, 29, 2005. doi: 10.1007/s00158-004-0488-7.

[114] P. D. Dunning, H. A. Kim, and G. Mullineux. Introducing loading uncertainty in topology optimization. *AIAA Journal*, 49, 2011. doi: 10.2514/1.J050670.

[115] Jiadong Deng and Wei Chen. Concurrent topology optimization of multiscale structures with multiple porous materials under random field loading uncertainty. *Structural and Multidisciplinary Optimization*, 56(1):1–19, 2017. doi: 10.1007/s00158-017-1689-1.

[116] S. Rahman and H. Xu. A univariate dimension-reduction method for multidimensional integration in stochastic mechanics. *Probabilistic Engineering Mechanics*, 19, 2004. doi: 10.1016/j.probengmech.2004.04.003.

[117] L. V. Gibiansky and O. Sigmund. Multiphase composites with extremal bulk modulus. *Journal of the Mechanics and Physics of Solids*, 48, 2000. doi: 10.1016/S0022-5096(99)00043-5.

[118] X. Yin, S. Lee, and W. Chen. Efficient random field uncertainty propagation in design using multiscale analysis. *Journal of Mechanical Design*, 131, 2009. doi:

10.1115/1.3042159.

[119] S. H. Lee and W. Chen. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization*, 37, 2009. doi: 10.1007/s00158-008-0234-7.

[120] S. H. Lee, W. Chen, and B. M. Kwak. Robust design with arbitrary distributions using gauss-type quadrature formula. *Structural and Multidisciplinary Optimization*, 39, 2009. doi: 10.1007/s00158-008-0328-2.

[121] Y. G. Zhao and T. Ono. Moment methods for structural reliability. *Structural Safety*, 23, 2001. doi: 10.1016/S0167-4730(00)00027-8.

[122] Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2): 359–373, February 1987. doi: 10.1002/nme.1620240207.

[123] K. Svanberg. Mma and gcmma—two methods for nonlinear optimization. 2007.

[124] Liwei Wang, Yu-Chin Chan, Zhao Liu, Ping Zhu, and Wei Chen. Data-driven metamaterial design with laplace-beltrami spectrum as “shape-DNA”. *Structural and Multidisciplinary Optimization*, 2020. doi: 10.1007/s00158-020-02523-5.

[125] V.J. Challis, A.D. Cramer, and A.P. Roberts. An optimised family of anisotropic microstructures with application to functionally graded materials. *International Journal of Solids and Structures*, 171:17–29, October 2019. doi: 10.1016/j.ijsolstr.2019.05.009.

[126] Haejoon Choi, Adrian Matias Chung Baek, and Namhum Kim. Design of non-periodic lattice structures by allocating pre-optimized building blocks. In *Volume 1: 39th Computers and Information in Engineering Conference*. American Society

of Mechanical Engineers, 2019. doi: 10.1115/detc2019-98204.

[127] A. Ferrer, J.C. Cante, J.A. Hernández, and J. Oliver. Two-scale topology optimization in computational material design: An integrated approach. *International Journal for Numerical Methods in Engineering*, 114(3):232–254, January 2018. doi: 10.1002/nme.5742.

[128] Ramin Bostanabad, Yu-Chin Chan, Liwei Wang, Ping Zhu, and Wei Chen. Globally approximate gaussian processes for big data with application to data-driven metamaterials design. *Journal of Mechanical Design*, 141(11), 2019. doi: 10.1115/1.4044257.

[129] Yuen-Shan Leung, Tsz-Ho Kwok, Huachao Mao, and Yong Chen. Digital material design using tensor-based error diffusion for additive manufacturing. *Computer-Aided Design*, 114:224–235, September 2019. doi: 10.1016/j.cad.2019.05.031.

[130] P. Lohmuller, J. Favre, S. Kenzari, B. Piotrowski, L. Peltier, and P. Laheurte. Architectural effect on 3d elastic properties and anisotropy of cubic lattice structures. *Materials &amp Design*, 182:108059, November 2019. doi: 10.1016/j.matdes.2019.108059.

[131] Desai Chen, Mélina Skouras, Bo Zhu, and Wojciech Matusik. Computational discovery of extremal microstructure families. *Science Advances*, 4(1), January 2018. doi: 10.1126/sciadv.aao7005.

[132] Desai Chen, David I. W. Levin, Shinjiro Sueda, and Wojciech Matusik. Data-driven finite elements for geometry and material design. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015. doi: 10.1145/2766889.

[133] Xingchen Liu. Distributed design of two-scale structures with unit cells. In *Volume 2A: 45th Design Automation Conference*. American Society of Mechanical Engineers, August 2019. doi: 10.1115/detc2019-97672.

[134] Dawei Li, Ning Dai, Yunlong Tang, Guoying Dong, and Yaoyao Fiona Zhao. Design and optimization of graded cellular structures with triply periodic level surface-based topological shapes. *Journal of Mechanical Design*, 141(7), 2019. doi: 10.1115/1.4042617.

[135] Yunlong Tang, Guoying Dong, Qinxue Zhou, and Yaoyao Fiona Zhao. Lattice structure design and optimization with additive manufacturing constraints. *IEEE Transactions on Automation Science and Engineering*, 15(4):1546–1562, October 2018. doi: 10.1109/tase.2017.2685643.

[136] Recep M. Gorguluarslan, Umesh N. Gandhi, Raghuram Mandapati, and Seung-Kyum Choi. Design and fabrication of periodic lattice-based cellular structures. *Computer-Aided Design and Applications*, 13(1):50–62, August 2015. doi: 10.1080/16864360.2015.1059194.

[137] Ruichen Jin, Wei Chen, and Agus Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. In *Volume 2: 29th Design Automation Conference, Parts A and B*. ASMEDC, January 2003. doi: 10.1115/detc2003/dac-48760.

[138] Karsten Grosse-Brauckmann. Triply periodic minimal and constant mean curvature surfaces. *Interface Focus*, 2(5):582–588, March 2012. doi: 10.1098/rsfs.2011.0096.

[139] Hermann Karcher. The triply periodic minimal surfaces of alan schoen and their constant mean curvature companions. *Manuscripta Mathematica*, 64(3):291–357,

September 1989. doi: 10.1007/bf01165824.

[140] H. G. von Schnering and R. Nesper. Nodal surfaces of fourier series: Fundamental invariants of structured matter. *Zeitschrift für Physik B Condensed Matter*, 83(3): 407–412, October 1991. doi: 10.1007/bf01313411.

[141] Meinhard Wohlgemuth, Nataliya Yufa, James Hoffman, and Edwin L. Thomas. Triply periodic bicontinuous cubic microdomain morphologies by symmetries. *Macromolecules*, 34(17):6083–6089, 2001. doi: 10.1021/ma0019499.

[142] James T. Hoffman. Table of surfaces, 2004. URL `http://www.msri.org/publications/sgp/jim/papers/morphbysymmetry/table/index.html`.

[143] Yazhou Zhang. *DEEP GENERATIVE MODEL FOR MULTI-CLASS IMBALANCED LEARNING*. PhD thesis.

[144] X. Yan, X. Huang, Y. Zha, and Y.M. Xie. Concurrent topology optimization of structures and their composite microstructures. *Computers &amp Structures*, 133: 103–110, March 2014. doi: 10.1016/j.compstruc.2013.12.001.

[145] Liang Xia and Piotr Breitkopf. Multiscale structural topology optimization with an approximate constitutive model for local material microstructure. *Computer Methods in Applied Mechanics and Engineering*, 286:147–167, April 2015. doi: 10.1016/j.cma.2014.12.018.

[146] Zhaocheng Liu, Dayu Zhu, Sean P. Rodrigues, Kyu-Tae Lee, and Wenshan Cai. Generative model for the inverse design of metasurfaces. *Nano Letters*, 18(10): 6570–6576, 2018. doi: 10.1021/acs.nanolett.8b03171.

[147] Alemseged Gebrehiwot Weldeyesus and Mathias Stolpe. A primal-dual interior point method for large-scale free material optimization. *Computational Optimization and*

Applications, 61(2):409–435, December 2014. doi: 10.1007/s10589-014-9720-6.

[148] Xingtong Yang and Ming Li. Free isotropic material optimization via second order cone programming. *Computer-Aided Design*, 115:52–63, October 2019. doi: 10.1016/j.cad.2019.05.002.

[149] Panagiotis Vogiatzis, Shikui Chen, Xianfeng David Gu, Ching-Hung Chuang, Hongyi Xu, and Na Lei. Multi-material topology optimization of structures infilled with conformal metamaterials. In *Volume 2B: 44th Design Automation Conference*. American Society of Mechanical Engineers, August 2018. doi: 10.1115/detc2018-85663.

[150] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2):1–50, 2016. doi: 10.1145/2907070.

[151] Clyde H. Coombs and George S. Avrunin. Single-peaked functions and the theory of preference. *Psychological Review*, 84(2):216–230, 1977. doi: 10.1037/0033-295x.84.2.216.

[152] Faez Ahmed, Mark Fuge, and Lev D Gorbunov. Discovering diverse, high quality design ideas from a large corpus. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2016.

[153] Faez Ahmed and Mark Fuge. Ranking ideas for diversity and quality. *Journal of Mechanical Design*, 140(1), 2017. doi: 10.1115/1.4038070.

[154] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[155] P Fernandes de Aguiar, B Bourguignon, MS Khots, DL Massart, and R Phan-Than-Luu. D-optimal designs. *Chemometrics and intelligent laboratory systems*, 30(2): 199–210, 1995.

[156] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Computing Surveys (CSUR)*, 37(4):345–387, 2005. doi: 10.1145/1118890.1118893.

[157] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu. A survey on data-driven 3d shape descriptors. *Computer Graphics Forum*, 38(1):356–393, 2018. doi: 10.1111/cgf.13536.

[158] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[159] E. Sharon and D. Mumford. 2d-shape analysis using conformal mapping. *International Journal of Computer Vision*, 70(1):55–75, 2006. doi: 10.1007/s11263-006-6121-z.

[160] Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2246–2259, 2015. doi: 10.1109/tpami.2015.2408346.

[161] Georgios Vamvakas, Basilis Gatos, and Stavros J Perantonis. Handwritten character recognition through two-stage foreground sub-sampling. *Pattern Recognition*, 43(8):

2807–2816, 2010.

[162] Nibaran Das, Jagan Mohan Reddy, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. A statistical–topological feature combination for recognition of handwritten numerals. *Applied Soft Computing*, 12 (8):2486–2495, 2012.

[163] Ritesh Sarkhel, Nibaran Das, Aritra Das, Mahantapas Kundu, and Mita Nasipuri. A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognition*, 71: 78–93, 2017.

[164] Leif Kobbelt and Mario Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004. doi: 10.1016/j.cag.2004.08.009.

[165] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. doi: 10.1109/34.232073.

[166] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[167] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. doi: 10.1109/tpami.2020.3005434.

[168] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1193–1200, 2011.

[169] Alexei Borodin. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.

[170] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

[171] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical Programming*, 14(1):265–294, 1978.

[172] Sangamesh R Deepak, M Dinesh, Deepak K Sahu, and GK Ananthasuresh. A comparative study of the formulations and benchmark problems for the topology optimization of compliant mechanisms. *Journal of Mechanisms and Robotics*, 1(1), 2009.

[173] N Komodakis, N Paragios, and G Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531–552, March 2011. doi: 10.1109/tpami.2010.108.

[174] U. Shmueli, editor. *International Tables for Crystallography: Volume B, Reciprocal space*, chapter 1.4, pages 114–174. International Union of Crystallography, 2010. doi: 10.1107/97809553602060000108.

[175] Panagiotis Vogiatzis, Shikui Chen, and Chi Zhou. An open source framework for integrated additive manufacturing and level-set-based topology optimization. *Journal of Computing and Information Science in Engineering*, 17(4), 2017. doi:

10.1115/1.4037738.

[176] Yanir Kleiman. Github repository, sample_mesh. `https://github.com/hexygen/sample_mesh`, 2017.

[177] Jun Wu, Ole Sigmund, and Jeroen P. Groen. Topology optimization of multi-scale structures: a review. *Structural and Multidisciplinary Optimization*, 63(3):1455–1480, March 2021. doi: 10.1007/s00158-021-02881-8.

[178] Daicong Da, Xiangyang Cui, Kai Long, Yong Cai, and Guangyao Li. Multiscale concurrent topology optimization of structures and microscopic multi-phase materials for thermal conductivity. *Engineering Computations*, 36(1):126–146, February 2019. doi: 10.1108/ec-01-2018-0007.

[179] Jiao Jia, Daicong Da, Jianxing Hu, and Sha Yin. Crashworthiness design of periodic cellular structures using topology optimization. *Composite Structures*, 271:114164, September 2021. doi: 10.1016/j.compstruct.2021.114164.

[180] Erik Andreassen and Casper Schousboe Andreasen. How to determine composite material properties using numerical homogenization. *Computational Materials Science*, 83:488–495, February 2014. doi: 10.1016/j.commatsci.2013.09.006.

[181] H. Rodrigues, J.M. Guedes, and M.P. Bendsoe. Hierarchical optimization of material and structure. *Structural and Multidisciplinary Optimization*, 24(1):1–10, August 2002. doi: 10.1007/s00158-002-0209-z.

[182] Long Jiang, Xianfeng David Gu, and Shikui Chen. Generative design of bionic structures via concurrent multiscale topology optimization and conformal geometry method. *Journal of Mechanical Design*, 143(1), July 2020. doi: 10.1115/1.4047345.

[183] Yafeng Han and Wen Feng Lu. A novel design method for nonuniform lattice structures based on topology optimization. *Journal of Mechanical Design*, 140(9), July 2018. doi: 10.1115/1.4040546.

[184] Junjian Fu, Liang Xia, Liang Gao, Mi Xiao, and Hao Li. Topology optimization of periodic structures with substructuring. *Journal of Mechanical Design*, 141(7), March 2019. doi: 10.1115/1.4042616.

[185] Vivien J. Challis, Anthony P. Roberts, and Joseph F. Grotowski. High resolution topology optimization using graphics processing units (GPUs). *Structural and Multidisciplinary Optimization*, 49(2):315–325, August 2013. doi: 10.1007/s00158-013-0980-z.

[186] Yan Zhang, Hao Li, Mi Xiao, Liang Gao, Sheng Chu, and Jinhao Zhang. Concurrent topology optimization for cellular structures with nonuniform microstructures based on the kriging metamodel. *Structural and Multidisciplinary Optimization*, 59(4):1273–1299, November 2018. doi: 10.1007/s00158-018-2130-0.

[187] Prakhar Jaiswal. *Geometric Reasoning and Machine Learning: A Set of Design and Manufacturing Problems*. PhD thesis, 2019.

[188] Mahmoud Alzahrani, Seung-Kyum Choi, and David W. Rosen. Design of truss-like cellular structures using relative density mapping method. *Materials &amp Design*, 85:349–360, November 2015. doi: 10.1016/j.matdes.2015.06.180.

[189] Oraib Al-Ketan, Rashid K. Abu Al-Rub, and Reza Rowshan. Mechanical properties of a new type of architected interpenetrating phase composite materials. *Advanced Materials Technologies*, 2(2):1600235, December 2016. doi: 10.1002/admt.201600235.

[190] A. Ferrer, J.C. Cante, J.A. Hernández, and J. Oliver. Two-scale topology optimization in computational material design: An integrated approach. *International Journal for Numerical Methods in Engineering*, 114(3):232–254, 2018. doi: 10.1002/nme.5742.

[191] Yiqiang Wang, Lei Zhang, Stephen Daynes, Hongying Zhang, Stefanie Feih, and Michael Yu Wang. Design of graded lattice structure with optimized mesostructures for additive manufacturing. *Materials & Design*, 142:114–123, March 2018. doi: 10.1016/j.matdes.2018.01.011.

[192] Hongming Zong, Hui Liu, Qingping Ma, Ye Tian, Mingdong Zhou, and Michael Yu Wang. VCUT level set method for topology optimization of functionally graded cellular structures. *Computer Methods in Applied Mechanics and Engineering*, 354: 487–505, September 2019. doi: 10.1016/j.cma.2019.05.029.

[193] M. Jansen and O. Pierard. A hybrid density/level set formulation for topology optimization of functionally graded lattice structures. *Computers & Structures*, 231:106205, April 2020. doi: 10.1016/j.compstruc.2020.106205.

[194] Xiao-Yi Zhou, Zongliang Du, and H. Alicia Kim. A level set shape metamorphosis with mechanical constraints for geometrically graded microstructures. *Structural and Multidisciplinary Optimization*, 60(1):1–16, May 2019. doi: 10.1007/s00158-019-02293-9.

[195] S M Tareq Zobaer and Alok Sutradhar. An energy-based method for interface connectivity of incompatible microstructures through parametric modeling. *Computer Methods in Applied Mechanics and Engineering*, 370:113278, October 2020. doi: 10.1016/j.cma.2020.113278.

[196] Yu-Chin Chan, Faez Ahmed, Liwei Wang, and Wei Chen. METASET: Exploring shape and property spaces for data-driven metamaterials design. *Journal of Mechanical Design*, 143(3), November 2020. doi: 10.1115/1.4048629.

[197] Mathieu Sanchez, Oleg Fryazinov, Valery Adzhiev, Peter Comninos, and Alexander Pasko. Space-time transfinite interpolation of volumetric material properties. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):278–288, February 2015. doi: 10.1109/tvcg.2014.2356196.

[198] Anjum I Rohra and Ramesh K. Kulkarni. Survey on recent trends in image morphing techniques. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, November 2019. doi: 10.1109/icssit46314.2019.8987879.

[199] J. Carballido-Gamio, J.S. Bauer, Keh-Yang Lee, S. Krause, and S. Majumdar. Combined image processing techniques for characterization of MRI cartilage of the knee. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2005. doi: 10.1109/iembs.2005.1617116.

[200] Anoop Ebey Thomas, Simon Guevelou, Edmondo Di Pasquale, Anne Chambard, Jean-Louis Duval, Francisco Chinesta, Victor Limousin, Xavier Delgerie, and Emmanuel Leroy. Shape parametrization & morphing in sheet-metal forming. *Procedia Manufacturing*, 47:702–706, 2020. doi: 10.1016/j.promfg.2020.04.216.

[201] D.E. Breen and R.T. Whitaker. A level-set approach for the metamorphosis of solid models. 7(2):173–192, 2001. doi: 10.1109/2945.928169.

[202] Morten Bojsen-Hansen, Hao Li, and Chris Wojtan. Tracking surfaces with evolving topology. *ACM Transactions on Graphics*, 31(4):1–10, August 2012. doi: 10.1145/2185520.2185549.

[203] Hollis Smith and Julián A. Norato. A MATLAB code for topology optimization using the geometry projection method. *Structural and Multidisciplinary Optimization*, 62(3):1579–1594, March 2020. doi: 10.1007/s00158-020-02552-0.

[204] Weisheng Zhang, Jie Yuan, Jian Zhang, and Xu Guo. A new topology optimization approach based on moving morphable components (MMC) and the ersatz material model. *Structural and Multidisciplinary Optimization*, 53(6):1243–1260, December 2015. doi: 10.1007/s00158-015-1372-3.

[205] Mathieu Sanchez. *Distance based heterogeneous volume modelling.* PhD thesis, Bournemouth University, 2015.

[206] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157–160, February 1973. doi: 10.1093/comjnl/16.2.157.

[207] Vadim Shapiro. Semi-analytic geometry with r-functions. *Acta Numerica*, 16:239–303, April 2007. doi: 10.1017/s096249290631001x.

[208] M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-free shape correspondence by deformation. *Computer Graphics Forum*, 38(5):1–12, August 2019. doi: 10.1111/cgf.13785.

[209] Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder image interpolation by shaping the latent space, 2020.

[210] Jiaqi Gu, Zeju Li, Yuanyuan Wang, Haowei Yang, Zhongwei Qiao, and Jinhua Yu. Deep generative adversarial networks for thin-section infant MR image reconstruction. *IEEE Access*, 7:68290–68304, 2019. doi: 10.1109/access.2019.2918926.

[211] Yu-Chin Chan, Kohei Shintani, and Wei Chen. Robust topology optimization of multi-material lattice structures under material and load uncertainties. *Frontiers of Mechanical Engineering*, 14(2):141–152, January 2019. doi: 10.1007/s11465-019-0531-4.

[212] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In *Lecture Notes in Mathematics*, pages 105–116. Springer Berlin Heidelberg, 1978. doi: 10.1007/bfb0067700.

[213] Shan Ba, William R. Myers, and William A. Brenneman. Optimal sliced latin hypercube designs. *Technometrics*, 57(4):479–487, October 2015. doi: 10.1080/00401706.2014.957867.

[214] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

[215] Sourav Das and Alok Sutradhar. Multi-physics topology optimization of functionally graded controllable porous structures: Application to heat dissipating problems. *Materials & Design*, 193:108775, August 2020. doi: 10.1016/j.matdes.2020.108775.

[216] X. Huang and Y.M. Xie. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43(14):1039–1049, October 2007. doi: 10.1016/j.finel.2007.06.006.

[217] Wei Chen and Faez Ahmed. PaDGAN: Learning to generate high-quality novel designs. 143(3), November 2020. doi: 10.1115/1.4048626.

[218] Asger Nyman Christiansen, Morten Nobel-Jørgensen, Niels Aage, Ole Sigmund, and Jakob Andreas Bærentzen. Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399, August

2013. doi: 10.1007/s00158-013-0983-9.

[219] Casper Schousboe Andreasen, Martin Ohrt Elingaard, and Niels Aage. Level set topology and shape optimization by density methods using cut elements with length scale control. 62(2):685–707, March 2020. doi: 10.1007/s00158-020-02527-1.

[220] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, March 2004. doi: 10.1017/cbo9780511804441.

[221] scikit-fmm: the fast marching method for python. `https://github.com/scikit-fmm/scikit-fmm`, 2021.

[222] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12 (2):555–573, January 2002. doi: 10.1137/s1052623499362822.

[223] Oded Amir, Niels Aage, and Boyan S. Lazarov. On multigrid-CG for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49(5):815–829, November 2013. doi: 10.1007/s00158-013-1015-5.

[224] Liang Xu and Gengdong Cheng. Two-scale concurrent topology optimization with multiple micro materials based on principal stress orientation. *Structural and Multidisciplinary Optimization*, 57(5):2093–2107, February 2018. doi: 10.1007/s00158-018-1916-4.

[225] Zheng Qiu, Quhao Li, Shutian Liu, and Rui Xu. Clustering-based concurrent topology optimization with macrostructure, components, and materials. *Structural and Multidisciplinary Optimization*, 63(3):1243–1263, November 2020. doi: 10.1007/s00158-020-02755-5.

[226] Doksoo Lee, Yu-Chin Chan, Wei, Chen, Liwei Wang, Anton van Beek, and Wei Chen. t-metaset: Task-aware generation of metamaterial datasets by diversity-based active learning. *arXiv preprint arXiv:2202.10565*, 2022.

[227] Liwei Wang, Anton van Beek, Daicong Da, Yu-Chin Chan, Ping Zhu, and Wei Chen. Data-driven multiscale design of cellular composites with multiclass microstructures for natural frequency maximization. *Composite Structures*, 280:114949, January 2022. doi: 10.1016/j.compstruct.2021.114949.

[228] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

[229] Hao Wang, Nadav Schor, Ruizhen Hu, Haibin Huang, Daniel Cohen-Or, and Hui Huang. Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[230] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1936–1944, 2018.

[231] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017.

[232] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning*, pages 40–49, 2018.

[233] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D Kulkarni, and Joshua B Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1511–1519, 2017.

[234] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, and Yaron Lipman. Multi-chart generative surface modeling. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

[235] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6040–6049, 2017.

[236] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[237] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[238] Ivan Sosnovik and Ivan Oseledets. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4):215–223, August 2019. doi: 10.1515/rnam-2019-0018.

[239] Saurabh Banga, Harsh Gehani, Sanket Bhilare, Sagar Patel, and Levent Kara. 3d topology optimization using convolutional neural networks, 2018.

[240] Yonggyun Yu, Taeil Hur, Jaeho Jung, and In Gwun Jang. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*, 59(3):787–799, October 2018. doi: 10.1007/s00158-018-2101-5.

[241] Baotong Li, Congjia Huang, Xin Li, Shuai Zheng, and Jun Hong. Non-iterative structural topology optimization using deep learning. *Computer-Aided Design*, 115: 172–180, October 2019. doi: 10.1016/j.cad.2019.05.038.

[242] Yiquan Zhang, Bo Peng, Xiaoyi Zhou, Cheng Xiang, and Dalei Wang. A deep convolutional neural network for topology optimization with strong generalization ability, 2019.

[243] M.-H. Herman Shen and Liang Chen. A new cgan technique for constrained topology design optimization, 2019.

[244] Sharad Rawat and M. H. Herman Shen. A novel topology optimization approach using conditional deep learning, 2019.

[245] Tinghao Guo, Danny J. Lohan, Ruijin Cang, Max Yi Ren, and James T. Allison. An indirect design representation for topology optimization using variational autoencoder and style transfer. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, January 2018. doi: 10.2514/6.2018-0804.

[246] Wei Chen, Kevin Chiu, and Mark Fuge. Aerodynamic design optimization and shape exploration using generative adversarial networks. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, January 2019. doi: 10.2514/6.2019-2351.

[247] Yujie Zhang and Wenjing Ye. Deep learning–based inverse method for layout design. *Structural and Multidisciplinary Optimization*, 60(2):527–536, February 2019. doi: 10.1007/s00158-019-02222-w.

[248] Aaditya Chandrasekhar, Saketh Sridhara, and Krishnan Suresh. Gm-tounn: Graded multiscale topology optimization using neural networks, 2022.

[249] Wentai Zhang, Zhangsihao Yang, Haoliang Jiang, Suyash Nigam, Soji Yamakawa, Tomotake Furuhata, Kenji Shimada, and Levent Burak Kara. 3d shape synthesis for conceptual design and optimization using variational autoencoders. In *Volume 2A: 45th Design Automation Conference*. American Society of Mechanical Engineers, August 2019. doi: 10.1115/detc2019-98525.

[250] T. Buhl, C.B.W. Pedersen, and O. Sigmund. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):93–104, April 2000. doi: 10.1007/s001580050089.

[251] Liwei Wang, Zhao Liu, Daicong Da, Yu-Chin Chan, Wei Chen, and Ping Zhu. Generalized de-homogenization via sawtooth-function-based mapping and its demonstration on data-driven frequency response optimization. *Computer Methods in Applied Mechanics and Engineering*, 395:114967, may 2022. doi: 10.1016/j.cma.2022.114967.

[252] Yu-Chin Chan, Daicong Da, Liwei Wang, and Wei Chen. Remixing functionally graded structures: data-driven topology optimization with multiclass shape blending. *Structural and Multidisciplinary Optimization*, 65(5), April 2022. doi: 10.1007/s00158-022-03224-x.

[253] Thomas S. Lumpe and Tino Stankovic. Exploring the property space of periodic cellular structures based on crystal networks. *Proceedings of the National Academy*

of Sciences, 118(7), February 2021. doi: 10.1073/pnas.2003504118.

[254] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[255] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[256] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[257] Graeme Kennedy and Yicong Fu. Topology optimization benchmark problems for assessing the performance of optimization algorithms. In *AIAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, jan 2021. doi: 10.2514/6.2021-1357.

[258] Ali Zolfagharian, Martin Denk, Abbas Z. Kouzani, Bodaghi Bodaghi, Saeid Nahavandi, and Akif Kaynak. Effects of topology optimization in multimaterial 3d bioprinting of soft actuators. *International Journal of Bioprinting*, 6(2), April 2020. doi: 10.18063/ijb.v6i2.260.

[259] Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi: 10.1007/s00158-019-02211-z.

[260] Ting Wei Chin, Mark K. Leader, and Graeme J. Kennedy. A scalable framework for large-scale 3d multimaterial topology optimization with octree-based mesh adaptation. *Advances in Engineering Software*, 135, 2019. doi: 10.1016/j.advengsoft.2019.05.004.

[261] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders, 2019.

[262] Depeng Gao, Jinhao Chen, Zhetong Dong, and Hongwei Lin. Connectivity-guaranteed porous synthesis in free form model by persistent homology. *Computers Graphics*, 106:33–44, 2022. doi: https://doi.org/10.1016/j.cag.2022.05.018.

[263] Jordan Matthews, Timothy Klatt, Clinton Morris, Carolyn C. Seepersad, Michael Haberman, and David Shahan. Hierarchical design of negative stiffness metamaterials using a bayesian network classifier1. *Journal of Mechanical Design*, 138(4), March 2016. doi: 10.1115/1.4032774.

[264] Daicong Da, Liang Xia, Guangyao Li, and Xiaodong Huang. Evolutionary topology optimization of continuum structures with smooth boundary representation. *Structural and Multidisciplinary Optimization*, 57(6):2143–2159, November 2017. doi: 10.1007/s00158-017-1846-6.