2024

Machine learning powered insights into metamaterial prediction and design

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

MACHINE LEARNING POWERED INSIGHTS INTO METAMATERIAL PREDICTION AND DESIGN

by

LINGXIAO YUAN

B.S., Xi'an Jiaotong University, 2017
M.S., Xi'an Jiaotong University, 2019

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2024

Approved by

First Reader

Harold S. Park, PhD
Professor of Mechanical Engineering
Professor of Materials Science and Engineering

Second Reader

Emma Lejeune, PhD
Assistant Professor of Mechanical Engineering

Third Reader

Douglas P. Holmes, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Materials Science and Engineering

Fourth Reader

Paul E. Barbone, PhD
Professor of Mechanical Engineering
Professor of Materials Science and Engineering

When you are studying any matter, or considering any philosophy, ask yourself only what are the facts and what is the truth that the facts bear out. Never let yourself be diverted, either by what you wish to believe, or by what you think would have beneficent social effects if it were believed. But look only, and solely, at what are the facts.

Bertrand Russell

Acknowledgments

This thesis was made possible with the assistance and support from many individuals.

First and foremost, I would like to thank my advisors Prof. Harold Park and Prof. Emma Lejeune for their consistent guidance and support. Prof. Park has been encouraging me to explore topics that resonate with my interests, patiently guiding me to become an independent and self-directed researcher, and providing ample support whenever I sought help during difficult times. Prof. Lejeune's expertise in Machine Learning, insightful advice, and dedication toward rigorous research have been greatly supporting me along the way in completing this dissertation. I also thank Prof. Lejeune for fostering a collaborative and vibrant research community, which has been invaluable to me.

I am thankful to Prof. Douglas Holmes and Prof. Paul Barbone for their willingness to serve on my Committee, and to Prof. Kamil Ekinci for agreeing to chair my defense. I thank my labmates Jeffrey, Hiba, Saeed, Quan, Sara, Erfan, and Jeong-Ho who have made this journey enjoyable and precious. Additionally, I appreciate the support and assistance provided by MechE staff Emery, Anna, and Patty.

Lastly, I would like to thank my family, especially my parents, my brother, and my grandparents, for always being there for me. I would also like to thank my fiancé for his unwavering support and companionship, without him I would have completed this dissertation ahead of schedule.

MACHINE LEARNING POWERED INSIGHTS INTO METAMATERIAL PREDICTION AND DESIGN

LINGXIAO YUAN

Boston University, College of Engineering, 2024

Major Professors: Harold S. Park, PhD
Professor of Mechanical Engineering
Professor of Materials Science and Engineering

Emma Lejeune, PhD
Assistant Professor of Mechanical Engineering

ABSTRACT

There has been significant recent interest in the mechanics community to apply machine learning methods to design and predict the properties of metamaterials. Metamaterials are distinguished by their programmability, enabling the achievement of novel functionalities typically absent in conventional materials. Understanding metamaterials necessitates unraveling the intricate nonlinear relationship between design choices and mechanical properties. Machine learning (ML) stands out from traditional approaches by its capability to accurately approximate nonlinear relationships and rapidly predict properties across an extensive number of unexplored materials. While ML methods have enabled many breakthroughs, at least two critical questions remain unanswered. The first pertains to whether the assumptions underlying ML models remain validated when applied to real-world mechanics problems. The second question is whether ML models can facilitate the design of novel metamaterials with limited prior information. This thesis is aimed at addressing these questions.

The first study aims to address the limitations of traditional ML models. These models assume that the training (observed) data and testing (unseen) data are independent and

identically distributed (i.i.d). However, when applied to real-world mechanics problems with unknown test environments, these standard ML approaches can be very sensitive to data distribution shifts, and can break down when evaluated on test datasets that violate the i.i.d. assumption. In contrast, out-of-distribution (OOD) generalization approaches assume that the data contained in test environments are allowed to shift. To date, multiple methods have been proposed to improve the OOD generalization of ML methods. However, most of these OOD generalization methods have been focused on classification problems, driven in part by the lack of benchmark datasets available for OOD regression problems. Thus, the efficiency of these OOD generalization methods on regression problems, which are typically more relevant to mechanics research of metamaterials than classification problems, is unknown. To address this, a fundamental study of OOD generalization methods for regression problems in mechanics has been performed. Specifically, three OOD generalization problems are identified: covariate shift, mechanism shift, and sampling bias. For each problem, two benchmark examples are created that extend the Mechanical MNIST dataset collection, and the performance of popular OOD generalization methods on these mechanics-specific regression problems is investigated. The numerical experiments show that in most cases, while the OOD algorithms perform better compared to traditional ML methods on these OOD generalization problems, there is a compelling need to develop more robust OOD methods that can generalize the notion of invariance across multiple OOD scenarios. Overall, it is expected that this study, as well as the associated open access benchmark datasets, will enable further development of OOD methods for mechanics specific regression problems.

The second study aims to design chiral metamaterials that can either violate reciprocity, or exhibit elastic asymmetry or odd elasticity. While these properties are highly desirable to enable mechanical metamaterials to exhibit novel wave propagation phenomena, it remains an open question as to how to design passive structures that exhibit both significant non-reciprocity and elastic asymmetry. In this study, several design spaces are defined for

chiral metamaterials leveraging specific design parameters, including the ligament contact angles, the ligament shape, and the circle radius. Having defined the design spaces, machine learning approaches, and specifically Bayesian optimization, are leveraged to determine optimally performing designs within each design space satisfying maximal non-reciprocity or stiffness asymmetry. Finally, multi-objective optimization by determining the Pareto optimum is performed to find chiral metamaterials that simultaneously exhibit high non-reciprocity and stiffness asymmetry. The analysis of the underlying mechanisms reveals that chiral metamaterials that can display multiple different contact states under loading in different directions are able to simultaneously exhibit both high non-reciprocity and stiffness asymmetry. Overall, this study demonstrates the effectiveness of employing ML to bring insights to a novel domain with limited prior information, and more generally will pave the way for metamaterials with unique properties and functionality in directing and guiding mechanical wave energy.

Contents

List of Tables

List of Figures

List of Abbreviations

EI	.....	Expected Improvement
ERM	.....	Empirical Risk Minimization
FEA	.....	Finite Element Analysis
FEM	.....	Finite Element Method
GP	.....	Gaussian Process
IGA	.....	Inter Gradient Alignment
i.i.d.	.....	independent and identically distributed
IRM	.....	Invariant Risk Minimization
MIP	.....	Maximal Invariant Predictor
ML	.....	Machine Learning
MLP	.....	Multilayer Perceptron
OOD	.....	Out-of-Distribution
PCA	.....	Principal Component Analysis
ReLU	.....	Rectified Linear Unit
REx	.....	Risk Extrapolation
RMSE	.....	Root Mean Squared Error
UCB	.....	Upper Confidence Bound

Chapter 1

Introduction

1.1 Background

Traditionally, the mechanical behavior of macroscale solid metamaterials is obtained through laboratory experiments, or by simulation methods such as the finite element method (FEM) (Lee et al., 2021; Svolos et al., 2022; Liu et al., 2021c; Süli, 2012), while the mechanics of smaller length scale structures can be obtained using molecular dynamics simulations (Park et al., 2006; Bian and Nicola, 2021). These methods are generally accurate and reliable, but can be extremely time-consuming and computationally expensive if there exists a large parameter space to search for an optimal design that maximizes specific functionalities and properties. Both human intuition and expert knowledge in mechanics can accelerate this process, but expert knowledge is expensive per se. Thus, to make the metamaterial design and analysis process more efficient and broadly accessible, there has been significant recent interest in the application of machine learning (ML) methods (Guo et al., 2021; Alber et al., 2019; Peng et al., 2021; Huang et al., 2021; Bock et al., 2019; Morgan and Jacobs, 2020; Pilania, 2021). There are several compelling reasons why ML methods have become a viable approach for addressing mechanics problems of metamaterials. First, by learning from an existing collection of data samples, ML methods have shown the ability to accurately predict metamaterial properties(Hadash et al., 2018; Mohammadzadeh and Lejeune, 2022; Kim et al., 2022; Zhang and Garikipati, 2020; Mianroodi et al., 2021; Saha et al., 2021; Prachaseree and Lejeune, 2022; Mozaffar et al., 2019; Fuhg et al., 2022; Wang et al., 2021; Chen et al., 2019; Su et al., 2021; Chen et al., 2021a). Second, because it takes

only seconds for a trained ML model to predict the target properties of thousands of new samples, this significant savings in computational expense enables the rapid and efficient exploration of large design spaces in search of new materials, material designs, and structural designs (Chen and Gu, 2020; Hanakata et al., 2020; Sanchez-Lengeling and Aspuru-Guzik, 2018; Kollmann et al., 2020; Forte et al., 2022; Liu et al., 2020; Challapalli et al., 2021; Gongora et al., 2020; Ni and Gao, 2021; Kobeissi et al., 2022). Third, as a data-driven method that makes predictions by statistically learning the correlation between input feature vectors and the target outputs, ML methods do not require any preliminary expert knowledge for either the learning or prediction processes. Fourth, the accuracy and reliability of ML models can be continuously enhanced by adding more data through modern techniques such as active learning (Chen and Gu, 2020; Gongora et al., 2020; Liang et al., 2021; Liu et al., 2021d) and reinforcement learning (François-Lavet et al., 2018; Sui et al., 2021; Wang and Sun, 2019; Wang et al., 2019). Finally, these ML methods are often transferable such that a ML model trained on a large dataset often only requires minimal additional data when being adapted to a new yet related mechanics problem (Lejeune and Zhao, 2021; Lu et al., 2020; Goswami et al., 2022).

Despite these significant benefits, at least two critical questions remain unanswered before the potential of ML methods can fully be realized in the context of mechanics problems of metamaterials. The first pertains to whether the assumptions underlying ML models remain validated when applied to real-world mechanics problems. The second question is whether ML models can facilitate the design of novel metamaterials with limited prior information. These critical challenges and the corresponding solutions proposed are elucidated in the subsequent sections.

1.1.1 Out of Distribution Generalization

One of the most important problems of ML models is that the excellent performance on test data is established on the assumption that the training and testing data are independent

and identically distributed (i.i.d.) (Shen et al., 2021; Sagawa et al., 2019; Guo et al., 2020). However, the data distribution of test environments in practical mechanics problems is usually unknown, and in general the data distribution of test environments and the training environments is not guaranteed to be i.i.d. for practical applications. Generating the ML model to test environments for which the data distribution is not identical to the data distribution of the training data that the ML is trained on is known as an out-of-distribution (OOD) generalization problem. While the performance of ML methods on i.i.d. generalization problems is typically excellent given a sufficiently large training dataset, ML models have been shown to be very brittle when applied to OOD problems even for easy-to-learn computer vision tasks (Arjovsky et al., 2019; Sagawa et al., 2020; Nagarajan et al., 2020). For example, after learning from a training dataset with the majority of the datapoints being blue-ish and a minority of the datapoints being green-ish, the accuracy of ML models on a cat vs. dog classification task drops by more than 20% on a test dataset in which all datapoints are green-ish(Nagarajan et al., 2020). Similarly, when most of the cow pictures in the training dataset on which the ML model is trained are collected from common contexts like pastures and grassy fields, the ML model fails to recognize the cows in uncommon contexts like waves and beaches (Beery et al., 2018).

These results, and the results of many other studies (Kurakin et al., 2018; Geirhos et al., 2018; Tsipras et al., 2018; Ye et al., 2021; Hu et al., 2021; Izmailov et al., 2021; Goodfellow et al., 2014), suggest that when unseen test data and training data are not i.i.d., the predictions of a ML model built on this assumption should be called into question. Because material design through data-driven methods requires exploring large unseen domains through learning on limited available data, the i.i.d. assumption will often be violated in this context (Forte et al., 2022). With poor knowledge of the data distribution of the unseen data, the poor generalization of ML models on OOD problems can lead to erroneous predictions on test data from unseen distributions and result in unreliable

predictions, and in the context of material design, propose designs that are far from optimal. For example, when designing composite materials with high strength and stiffness, Kim et al. (2021) showed that when the ML model is trained on training data with lower strength and stiffness and used to predict data with higher strength and stiffness, the prediction accuracy drops dramatically, resulting in being trapped in local minima for forward material design. As one line of research, the detection of OOD data is of interest because it is related to the reliability of ML model predictions (DeVries and Taylor, 2018; Ming et al., 2022; Yang et al., 2021; Berger et al., 2021). For example, methods following a Bayesian approach that measure the uncertainty of ML predictions have been applied to OOD detection (Wang and Aitchison, 2021; Henning et al., 2021; Xie et al., 2022b,a). In addition to detecting OOD data, designing ML models that generalize better to OOD samples is an important new direction for applying ML methods to problems in mechanics.

Recently, to improve the robustness and generalizability of ML models, researchers have designed algorithms that embed the prior known physics into ML models when the underlying physics of the training and testing environments are similar and partially known. For example, the well-known approach of physics-informed neural networks (PINNs) embeds partial differential equations (PDEs) into neural networks during the learning process (Raissi et al., 2019, 2020; Cuomo et al., 2022; Cai et al., 2021). In addition, equivariant neural networks embed known structural symmetries into neural networks, such that the neural networks are robust towards the changes caused by permutation, translation or rotation (Fuchs et al., 2020; Satorras et al., 2021; Smidt et al., 2021; Cohen et al., 2019). Another approach, sparse identification of nonlinear dynamics (SINDy), assumes that a dynamical system can be described by a parsimonious model with few key items and parameters, which can be identified by sparse regression. Proponents of this approach argue that parsimonious models are able to maintain accuracy while preventing over-fitting. Essentially, reducing the complexity of the model is a strategy to improve its generalizability (Brunton et al., 2016;

Champion et al., 2019; Quade et al., 2018). Overall, the methods listed here approach OOD problems by leveraging prior physical knowledge of the system and extracting governing equations from the data. While these approaches are quite powerful, formulating predictive models in this manner is not necessarily suitable for all problems. For example, these approaches often do not extend naturally for making predictions on unseen heterogeneous domains. Alternatively, other recent works in mechanics have focused on methods that improve the robustness of ML models to new domains by active or transfer learning (Kim et al., 2021; Lookman et al., 2019; Lejeune and Zhao, 2021), which updates the current learned model by further collecting a few representative data points from the new test domain. This field is referred as domain adaptation (Wang and Deng, 2018; Csurka, 2017; Farahani et al., 2021), which assumes that the data from the test domain is available such that a few data can be collected from the test domain based on some available prior knowledge of the test data distribution (Shen et al., 2021).

In contrast, OOD generalization, which is also known as domain generalization (Zhou et al., 2021; Shen et al., 2021), considers a broader situation in which the ML model will be generalized to domains that are unseen and unknown. This unknown distribution of future target datasets prevents us from augmenting the training dataset in a manner that is informed by the characteristics of the new target domain. To improve OOD generalization without collecting new data, ML models are encouraged to learn the causal correlation between inputs and outputs that will not change from training to test environments (Peters et al., 2016; Schölkopf, 2022; Bühlmann, 2020; Weichwald and Peters, 2021), i.e, stable correlations. This is because the failure of the traditional ML models, i.e. Empirical Risk Minimization (ERM) (Sagawa et al., 2020; Nagarajan et al., 2020), on OOD generalization problems can be due to data being mixed and shuffled before training, which makes it difficult for ML models to learn whether a feature of the data is stable (referred to as "invariant") or unstable (referred to as "spurious") when the data distribution shifts from one

environment to another (Arjovsky et al., 2019). Inspired by this, recently many methods have been proposed to improve OOD generalization by not shuffling data collected from different environments but instead learning the stable features across these environments (or domains) (Zhou et al., 2021; Shen et al., 2021). Specifically, these OOD generalization methods consider data collected from multiple environments, while setting two goals for a ML model trained on these environments. First, the ML model is trained to perform well on all training environments, which can be achieved by ERM by minimizing the loss between the predicted value and the ground truth. Meanwhile, the ML model is also trained to perform consistently across all environments, which is achieved by adding a penalty term to the loss function (Arjovsky et al., 2019; Koyama and Yamaguchi, 2020; Krueger et al., 2021; Mahajan et al., 2021). Furthermore, there are many other works that prevent a ML model from learning spurious features by different approaches like distributionally robust optimization (Sagawa et al., 2019; Duchi and Namkoong, 2018), game-theoretic ML (Ahuja et al., 2020), and selective rationalization techniques (Chang et al., 2020). Compared to ERM, researchers have shown that these methods perform better on OOD classification tasks like recognizing digits in the colored MNIST dataset (Arjovsky et al., 2019), in which the color of a digit is the spurious and unstable feature that varies across training and testing datasets.

While many methods have been proposed to improve OOD generalization, applying these advances to problems in mechanics is not straightforward for multiple reasons. First, the vast majority of the OOD generalization algorithms have been applied to classification, and not regression problems (Krueger et al., 2021; Koyama and Yamaguchi, 2020; Piratla et al., 2020; Gulrajani and Lopez-Paz, 2020). As a result, the performance of these approaches on regression problems remains poorly understood. Second, there are only a few distribution shift benchmark datasets for validating the efficiency of these methods. To address this, Koh et al. (Koh et al., 2021) presented WILDS as a benchmark dataset which is a collection

of ten distribution shift datasets from different fields like tumor identification and genetic perturbation classification. However, all ten datasets in WILDS are classification problems, which is again an issue for advancing methods relevant to mechanics as regression problems dominate the mechanics field. At present, due to the unavailability of benchmark regression datasets with data distribution shifts, the efficiency of OOD algorithms on regression problems is usually evaluated on synthetic data of artificially constructed toy problems which do not fully represent the challenges associated with practical regression problems (Liu et al., 2021a; Kuang et al., 2020; Tripuraneni et al., 2021). Therefore, to address this gap in the literature, this thesis aims to perform a fundamental study of OOD generalization methods for regression problems in mechanics. Specifically, three OOD generalization problems are identified: covariate shift, mechanism shift, and sampling bias, and two benchmark datasets are created for each problem based on extensions to the Mechanical MNIST dataset collection, with access information given in Section 6.3. The experiment results show that for most cases, while the OOD algorithms outperform traditional ML methods on OOD generalization problems in mechanics, there is a compelling need to develop more robust OOD methods that can generalize notions of invariance across multiple OOD scenarios. Overall, the insights of this study, as well as the benchmark datasets developed, are expected to facilitate further development of OOD methods for regression problems.

1.1.2 Nonreciprocal and Asymmetric Chiral Metamaterials

There has been significant recent interest in the topics of reciprocity (Nassar et al., 2020b), and elastic asymmetry within the broader mechanics community. Reciprocity implies that if we push a structure on one side (X), the other side (Y) will move by a certain amount. If we push the opposite side (Y) with the same force, side X will move the same amount. This idea has been codified through the well-known Maxwell-Betti reciprocity, which can be written as

$$F_X u_{Y\to X} = F_Y u_{X\to Y} \tag{1.1}$$

There is significant interest in finding structures that can break reciprocity, either statically (Coulais et al., 2017; Shaat, 2020), or dynamically (Wang et al., 2023; Nassar et al., 2017; Trainiti and Ruzzene, 2016; Wang et al., 2018; Goldsberry et al., 2019; Fang et al., 2021; Lu and Norris, 2021, 2022; Patil et al., 2022; Wallen et al., 2018; Kuznetsova et al., 2017; Attarzadeh et al., 2020). The major motivation for this has been related to non-reciprocal wave propagation, in which structures enable wave propagation in one direction, but support different levels of wave propagation in other directions. Most approaches to accomplishing this have involved active structures, in which the elastic stiffness and/or density can be actively modulated in space and time to enable non-reciprocal wave propagation (Nassar et al., 2017; Trainiti and Ruzzene, 2016; Wang et al., 2018). In contrast, passive approaches to generating non-reciprocal structures have been less studied, due to challenges in creating passive structures that violate reciprocity (Wang et al., 2023; Goldsberry et al., 2019; Fang et al., 2021; Lu and Norris, 2022; Attarzadeh et al., 2020). This has generally been accomplished using passive structures that exhibit a nonlinearity in which the elastic stiffness of a structure is bilinear, or different depending on the direction of loading (Wang et al., 2023; Lu and Norris, 2022; Goldsberry et al., 2019).

In addition to non-reciprocal elasticity, there has recently emerged significant interest in creating structures that exhibit asymmetric elasticity (Scheibner et al., 2020; Chen et al., 2021b; Tan et al., 2022; Yin and Liu, 2023; Zhang et al., 2020a). This interest has emerged because the mechanical behavior of linear elastic, isotropic solids is typically described by a free energy function, which carries the implication that its elasticity tensor is symmetric. However, Scheibner et al. recently proposed the notion of odd elasticity, for those linear elastic isotropic solids whose mechanical behavior cannot be described by a free energy function (Scheibner et al., 2020). As a result, odd elastic solids have a non-symmetric elasticity tensor, where the mechanical response to different loads (in contrast to the difference in mechanical response to different directions for non-reciprocal elasticity) is not the same.

For example, a solid extension could induce shear, while the same shear would induce a different amount of extension. It has been shown that such odd elastic solids could induce interesting dynamic phenomena, including non-Hermitian wave propagation (Chen et al., 2021b). However, similar to non-reciprocal solids, it has been considerably easier to achieve asymmetric elasticity using active, rather than passive solids.

Recently, Shaat and Park (2023) proposed a chiral metamaterial that exhibits both non-reciprocal and asymmetric elasticity. While chiral metamaterials have been widely studied over the past decade (Wu et al., 2019; Fernandez-Corbaton et al., 2019; Liu and Hu, 2016; Liu et al., 2012; Chen et al., 2013; Nassar et al., 2020a; Shaat and Park, 2023), the mechanism enabling this behavior is that of contact, in which the ligament connecting two rigid circles is initially in a state of contact with both circles. Non-reciprocal (i.e. directional) elasticity is realized because while the ligament remains in contact with both circles under tension, and thus is stiff, it loses contact with the circles under compression, and is thus elastically soft. Similarly, asymmetric elasticity results because in certain directions of loading, the ligament remains in contact, while in other directions of loading it does not. This, in conjunction with the chirality that couples different deformation modes, enables asymmetric elasticity to occur.

An important open question that remains is whether passive solids with tunable mechanical properties that are both non-reciprocal and asymmetric can be created, and further what the mechanisms are that would allow such a combination of properties to be realized. Because the design space for the chiral metamaterial is relatively large, encompassing potential factors such as circle diameter, ligament contact area, and ligament geometry, one approach to realizing these properties is by utilizing a machine learning model to learn the combination of factors enabling these properties to be realized using starting with the base two circles, one ligament system.

To conduct the chiral metamaterial optimization, several design spaces for chiral meta-

materials leveraging specific design parameters are defined, including the ligament contact angles, the ligament shape, and the circle radius. Having defined the design spaces, machine learning approaches, and specifically Bayesian optimization, are leveraged to determine optimally performing designs within each design space satisfying maximal non-reciprocity or stiffness asymmetry. Finally, multi-objective optimization is performed by determining the Pareto optimum and obtaining chiral metamaterials that simultaneously exhibit high non-reciprocity and stiffness asymmetry. The analysis of underlying mechanisms reveals that chiral metamaterials that can display different contact states under loading in different directions can simultaneously exhibit both high non-reciprocity and stiffness asymmetry. Overall, this work demonstrates the effectiveness of employing ML to bring insights to a novel domain with limited prior information, and more generally will pave the way for metamaterials with unique properties and functionality in directing and guiding mechanical wave energy.

1.2 Outlines of the Thesis

This thesis is organized as follows:

Chapter 2 introduces three different kinds of OOD regression problems under the scope of mechanics problems for heterogeneous metamaterials: Covariate shift, Mechanism shift, and sampling bias. Built upon the Mechanical MNIST and Mechanical EMNIST benchmark dataset (Lejeune, 2020), two corresponding benchmark examples are created for each type of OOD problem.

Chapter 3 evaluates the performance of different ML algorithms on the benchmark dataset created in Chapter 2. Specifically, Section 3.1 provides a review of the ERM method and introduces three popular OOD generalization algorithms (invariant risk minimization, or IRM (Arjovsky et al., 2019), risk extrapolation, or REx (Krueger et al., 2021), and inter-gradient-alignment, or IGA (Koyama and Yamaguchi, 2020)) for solving OOD problems. In

Section 3.4, OOD generalization performance of ERM and the OOD algorithms on the OOD problems on metamaterial prediction is demonstrated and discussed. Finally, Section 3.5 gives the main conclusions of this study.

Chapter 4 delves into the non-reciprocity and asymmetry characteristics of chiral metamaterials, analyzing their behavior through Finite Element Simulation. The objective function for optimization is explicitly defined, aiming to get chiral metamaterial designs that exhibit either non-reciprocity and asymmetry, or both. Furthermore, the design parameters of chiral metamaterial are broaden and four distinct design spaces are proposed, enabling comprehensive exploration of optimal chiral designs.

In Chapter 5, the chiral metamaterial is characterized by its high-level geometric features, and data augmentation technique is utilized to increase the training datasets. The Bayesian optimization is employed to seek optimal chiral metamaterials for both single and multi-objective optimization. Subsequently, the identified optimal designs are presented and the relationship between the geometry, the contact mechanisms, and the non-reciprocity and asymmetry properties of chiral metamaterials are revealed. Finally, the main conclusions of this study are drawn in Section 5.4.

Chapter 2

Mechanical MNIST – Data Distribution Shift

2.1 Mechanical MNIST Collection

The original MNIST dataset (LeCun et al., 1998), the inspiration for Mechanical MNIST, is a benchmark image dataset of labeled handwritten digits (0-9) with consistent preprocessing and formatting. Each image in the dataset is a digit represented by a 28×28 pixel bitmap. For the original MNIST dataset, the standard goal is to train a ML model to classify the correct digit based on the input pixel bitmap (Kussul and Baidyk, 2004; Cireşan et al., 2010; An et al., 2020). Inspired by the original MNIST dataset, the Mechanical MNIST dataset was created as a benchmark dataset specifically for problems in mechanics (Lejeune, 2020). In Mechanical MNIST, the images are no longer bitmaps without physical meaning, but instead represent the elastic modulus distribution of a heterogeneous block of material. In Mechanical MNIST, to ensure the elastic modulus values for the image bitmaps have non-zero values and lie within a reasonable range, the 28×28 MNIST image bitmaps are transformed into a map of elastic moduli following the equation:

$$E = \frac{b}{255.0} * (s-1) + 1 \tag{2.1}$$

where b is the corresponding value of the grayscale bitmap in the range $0-255$ and s is set to 100 for the original iteration of the dataset. After the transformation defined by eqn. 2.1, the elastic modulus distribution of the heterogeneous material lies in the range from 1 to s.

In the original contribution to the Mechanical MNIST dataset collection, four different

mechanical processes (Confined Compression, Shear, Equibiaxial Extension and Uniaxial Extension) are considered. In all cases, these mechanical processes are simulated via the Finite Element Method (FEM) using the open source software FEniCS (Logg and Wells, 2010; Logg et al., 2012). This study focus on the Equibiaxial Extension load case as detailed in the work Lejeune (2020). Note that this dataset is publicly accessible under an open source license, with access information given in Section 6.3. For the ML problems discussed in this study, the feature vectors X will contain the modulus distribution of each sample, and the target interest y will contain the change in strain energy after the sample is subject to equibiaxial extension, and thus y is a scalar value for each sample. Specifically, the finite element simulation is performed with a dimensionless study and a compressible Neo-Hookean constitutive law. In addition, the Poisson's ratio was set to 0.3 throughout the sample, and the applied extension displacement is $d = 7.0$ on each boundary, which corresponds to 50% of the square domain side length. Given this basic problem definition, the Mechanical MNIST Dataset Collection is sampled and extended to be suitable for OOD experiments. The details of this work are given in the following Sections.

2.2 OOD Experiment 1: Covariate Shift

2.2.1 Problem Definition

Covariate shift is one of the most common OOD shifts. It happens when only the marginal distribution $P(X)$ changes from the training environment to the testing environment ($P_{tr}(X) \neq P_{te}(X)$), while the conditional distribution $P(y|X)$ remains unchanged($P_{tr}(y|X) = P_{te}(y|X)$) (Shen et al., 2021; Zhou et al., 2021). In simpler terms, covariate shift occurs when the input distribution changes while the output distribution does not. Real world covariate shift datasets are readily available for classification problems. For example, in the colored MNIST dataset (Arjovsky et al., 2019), the same digit appears in different colors between the testing and training environments. In the PACS (Li et al., 2017) dataset, dog images

are collected from Photo, Art, and Cartoon environments for the training dataset but from a Sketch environment for the testing dataset. In the Waterbird dataset (Sagawa et al., 2019), waterbird photographs are mostly taken under a water background in the training dataset, but mostly taken with a land background in the testing dataset.

In contrast to classification problems, covariate shift in regression problems have been much less studied due to the limited availability of relevant regression focused benchmark datasets. This lack of benchmark data motivated us to introduce an open source benchmark dataset to investigate mechanics relevant problems with covariate shift. The introduction of this dataset is in the next Section.

2.2.2 Dataset Description

As shown in eqn. 2.1, the Mechanical MNIST dataset input patterns are transformed from the original MNIST dataset through an environmental control factor s. The elastic modulus E of blocks that make up the digit are close to s while the elastic modulus E of the blocks in the background is set to 1. When s is much larger than 1 (e.g., $s = 100$), the blocks in the digit area are much stiffer than the soft blocks in the background. Thus the digit will barely deform and almost all deformation during equibiaxial extension loading occurs in the blocks in the background. As a result of this large stiffness mismatch, the effect of changing the environmental factor s within the range $s = 50 - 100$ on the total change in strain energy of the domain is small. Therefore, altering s within this range is an appropriate strategy for inducing a covariate shift as defined in Section 2.2.1. Following this strategy, the details of two training environments and two test environments is summarized below:

- Training Environment 1: $s = 100$, data size = 2500 (2000 for training, 500 for validation)
- Training Environment 2: $s = 90$, data size = 2500 (2000 for training, 500 for validation)

- Testing Environment 1: $s = 75$, data size = 2000
- Testing Environment 2: $s = 50$, data size = 2000

The input and output distributions of training and testing environments are shown in Fig. 2·1. In Fig. 2·1a, one representative example is sampled from each of the four environments. For each example, the elastic modulus distribution and the corresponding deformation after equibiaxial extension are presented. Fig. 2·1b is a visualization of the input parameter space of each environment with Principal Component Analysis (PCA). Specifically, PCA is employed to reduce the high-dimensional input feature space (784 parameters) to the first principal component and plot the distribution of this one-dimensional variable in Fig. 2·1b with both a histogram and a redundant boxplot to aid in visualization. Note that because the unseen environments are considered unknown, the PCA model is fit exclusively on the data from the training environment. Fig. 2·1c is a visualization of the output strain energy via the same approach. In these plots, the training and testing environments are distinguished by red and grey hues respectively. As Fig. 2·1b-c shows, in this covariate shift problem, the input feature distribution shifts substantially between the different environments while the distribution of the strain energy output barely changes. In addition, it is worth noting that the difference between the two training environments $s = 100$ and $s = 90$ is substantially smaller than the difference between training environments and testing environments, which comprises $s = 75$ and $s = 50$.

2.3 OOD Experiment 2: Mechanism Shift

2.3.1 Problem Definition

The notion of concept drift has emerged in ML to describe the change in relationship between input and output data over time. Standard examples of concept drift include not accounting for the season when predicting temperature, or changes in how people use mobile phones over time (Tsymbal, 2004; Lu et al., 2018). The elements underpinning the concept

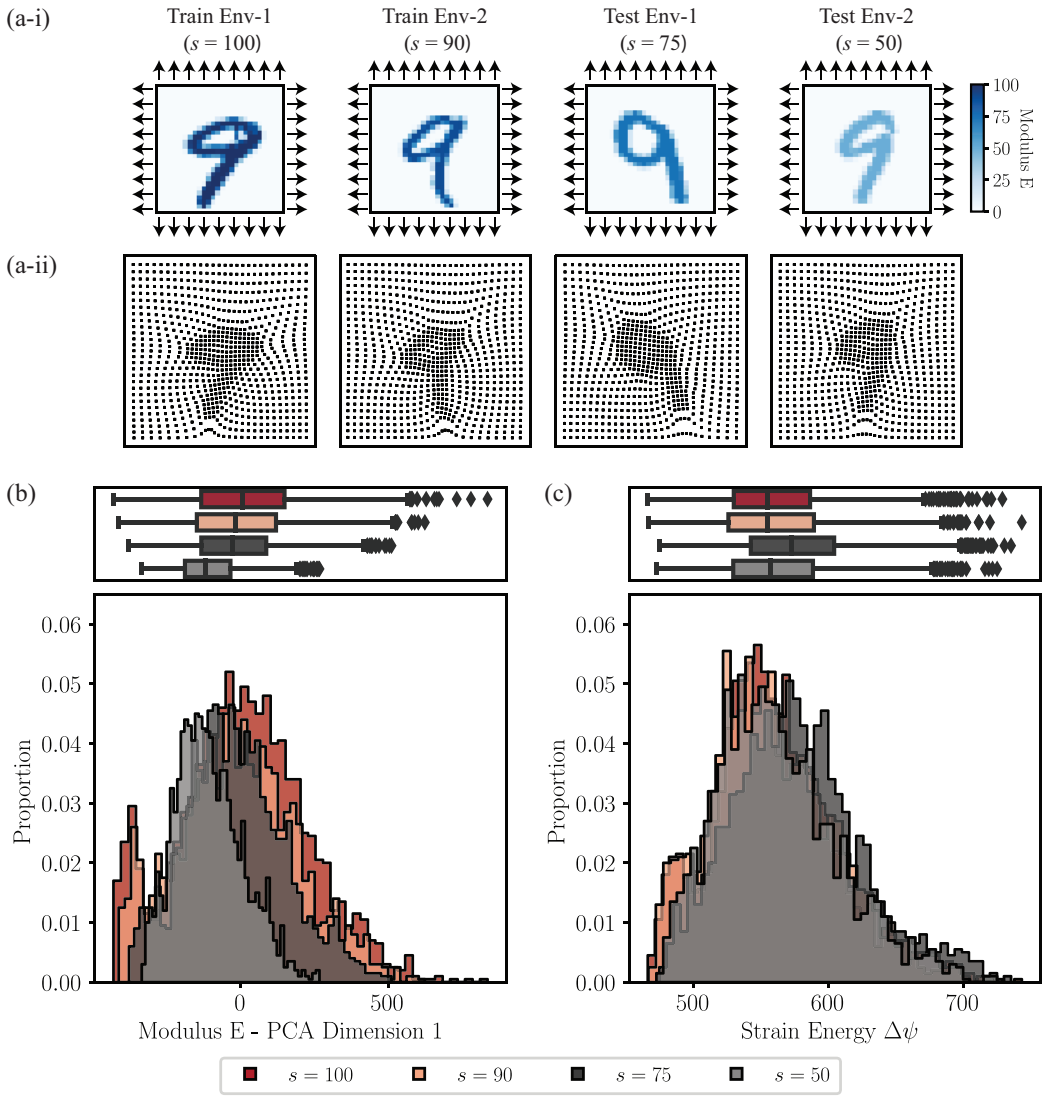


Figure 2·1: Illustration of the covariate shift dataset. (a-i) Equibiaxial extension boundary conditions and elastic modulus distribution for a representative example from each environment. (a-ii) Deformation of each example in (a-i) after the completion of the equibiaxial extension simulations. (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

drift, or change in the underlying data distribution $P(X, y)$, are referred to as the hidden context.

While primarily used in other areas, this notion is also applicable to mechanics and physics, in which the governing equations do not change over time, but can give inaccurate or physically unreasonable outputs when the control variable increases or decreases significantly. For example, continuum mechanics can break down if the representative length scale is small enough to be comparable to the atomic lattice spacing, or classical mechanics can break down when velocities approach the speed of light. To distinguish these examples from the term concept drift in which the data shift is typically caused by a time-dependent shift of the hidden context, this phenomenon is referred to as a mechanism shift in this work. In brief, a mechanism shift is when the data shift is caused by a change in the underlying mechanisms that control the mapping between the input and the output. The introduction of the mechanism shift dataset is in the next Section.

2.3.2 Dataset Description

According to the discussion in Section 2.2.2, changing the value of s when s is large ($s = 50 - 100$) has little effect on the distribution of the strain energy. Both the training data and the testing data in Section 2.2.2 follow this same underlying deformation mechanism where the embedded digit behaves approximately as a rigid body embedded in a soft surrounding matrix. In this Section, the test environments in which s is smaller ($10 - 25$), i.e, the elastic mismatch between the digit and the background matrix is smaller, are considered. With a lower stiffness mismatch, equibiaxial extension loading will cause the digit to deform, and thus changing s will influence the distribution of the final change in strain energy. Following this strategy, the training environments are kept the same as in Section 2.2.2 while introducing two new test environments and summarizing these new test environments along with the previous training environments below:

- Training Environment 1: $s = 100$, data size = 2500 (2000 for training, 500 for validation)

- Training Environment 2: $s = 90$, data size = 2500 (2000 for training, 500 for validation)
- Testing Environment 1: $s = 25$, data size = 2000
- Testing Environment 2: $s = 10$, data size = 2000

The input and output distributions of training and testing environments are shown in Fig. 2·2. As shown in Fig. 2·2(a-i) and (a-ii), compared to the representative examples from the training environments ($s = 100$, $s = 90$), the deformation of the blocks near the digit is much smoother for the representative examples from the test environments ($s = 25$, $s = 10$). Due to this deformation mechanism change, the distribution of the strain energy $\Delta\psi$ also shifts, as shown in Fig. 2·2(c). Therefore, in contrast to the scenario posed in Section 2.2.2, the scenario posed here contains both a shift in the input distribution and a shift in the output distribution. The bi-leveled distribution shift caused by the Mechanism shift makes it a larger challenge for the OOD generalization.

2.4 OOD Experiment 3: Sampling Bias

2.4.1 Problem Definition

Sampling bias happens when the data that is selected for training a ML model is not representative of the entire data pool that will be explored once the model is deployed. The presence of sampling bias makes it more difficult to find causal effects in the dataset by potentially introducing spurious correlations between input features and output properties, thus causing failures in OOD generalization. Sampling bias can exist in any empirical research if the underlying causal relationship is unknown (Winship and Mare, 1992; Winship and Morgan, 1999). For example, although ImageNet (Deng et al., 2009) has a very large sampling size with more than 14 million images, CNNs trained on ImageNet show bias by recognizing an animal by its texture rather than by its shape (Geirhos et al., 2018).

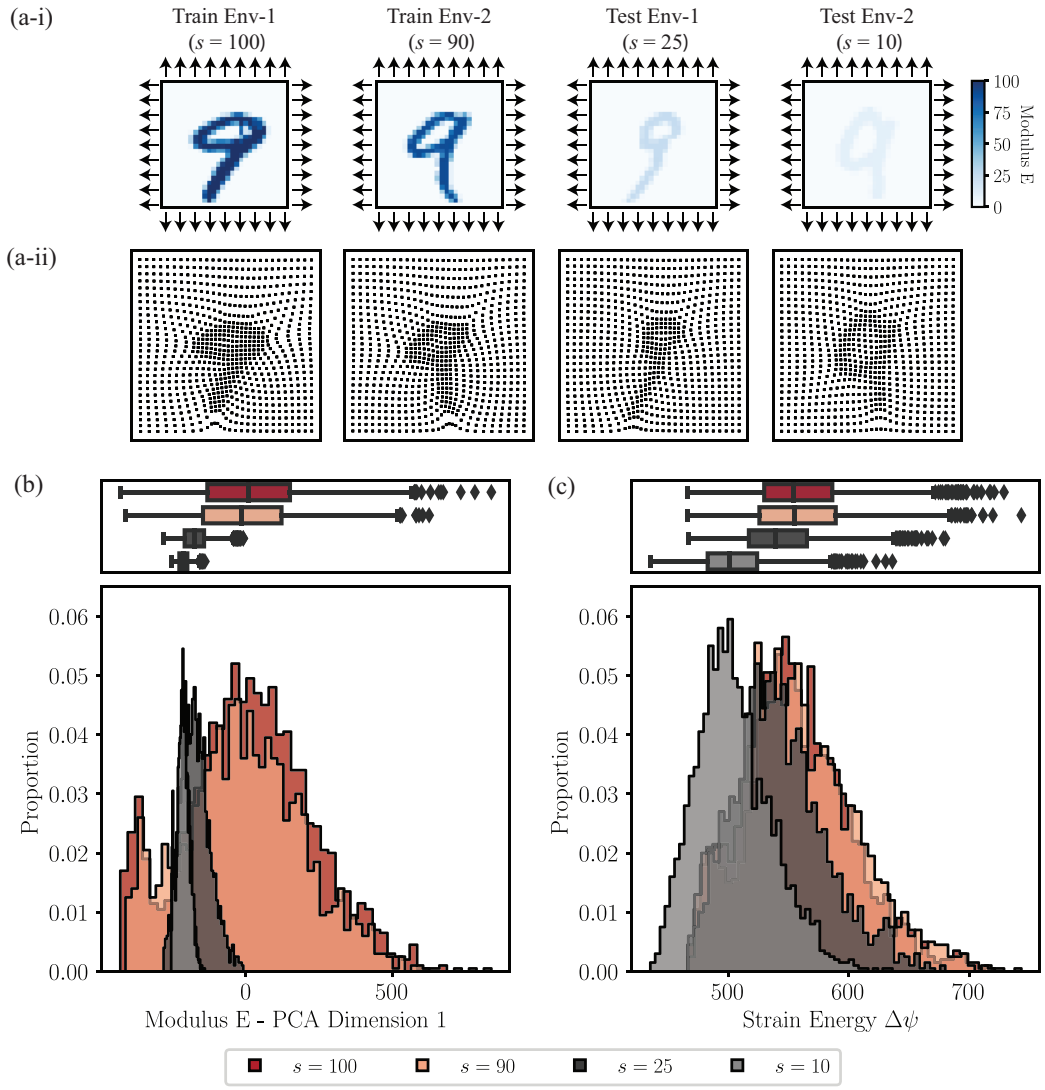


Figure 2·2: Illustration of the mechanism shift dataset. (a-i) Equibiaxial extension boundary conditions and elastic modulus distribution for a representative example from each environment. (a-ii) Deformation of each example in (a-i) after the completion of the equibiaxial extension simulations. (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

Studying the OOD problems caused by sampling bias is critical for material design by data-driven methods. This is because collected samples usually contain some form of bias, and thus predictions made by ML models may be a function of a spurious relationship that will not exist in the testing environment. To study the OOD shift caused by sampling bias, Kuang et al. (2020) introduced a selection mechanism in which data points are selected by a selection probability determined by spurious relationships between the dataset inputs and outputs. Liu et al. (2021b) later applied state-of-the-art OOD algorithms on synthetic data generated by this selection mechanism and showed that they outperform traditional ML methods that do not account for the presence of sampling bias. Although it is convenient to perform experiments on synthetic data where spurious features directly augment the input feature vector, real-world problems often contain sampling biases that are far more subtle. Thus, additional benchmark datasets to examine the performance of ML models in the face of sampling bias remain an important need that the broader field is currently lacking. To provide a benchmark data resource for OOD studies on sample selection bias, a sampling bias dataset within the Mechanical MNSIT dataset collection is introduced in the next Section.

2.4.2 Dataset Description

The input for the Mechanical MNIST dataset is the elastic modulus map for each material domain, which contains no inherent spurious features as every component of the input vector is the Young's modulus E of a block domain and thus contributes to the final change in strain energy of the domain. However, the dataset does potentially contain implicit spurious features embedded in the input vectors that can incorrectly be regarded as causal of the output by a ML model during the training process. In other words, certain characteristics of the input distribution may be spuriously correlated with the simulation output. And, more critically, defining spurious features and intentionally selecting them when establishing training and testing environments can be achieved on the Mechanical MNIST dataset. To

induce spurious correlations, the sum of the modulus of each block domain $V_i = \sum_{j=1}^{784} E_j$ is defined as an implicit spurious feature. Then, data points are sampled from the original Mechanical MNIST distribution ($s = 100$) through a biased selection mechanism. Each point in the original distribution has a selection probability P_i that is designed to induce sampling bias in a controlled fashion through the distribution control parameters r. To establish P_i, the selection probability of data point i, the sum of modulus V_i, and the final change in strain energy y_i are considered. The biased selection mechanism, inspired by the mechanism defined in Kuang et al. (2020), then takes the form:

$$P_i = |r|^{-5|\tilde{y}_i - \text{sign}(r)\tilde{V}_i|} \tag{2.2}$$

where $|r| \geq 1$ and $\text{sign}(r)$ is a sign function with $\text{sign}(r) = 1$ if $r > 0$ and $\text{sign}(r) = -1$ if $r \leq 0$. In addition, the implicit spurious feature V_i and the output distribution of y are normalized through the functions:

$$\tilde{V}_i = \frac{V_i - V_{\text{mean}}}{V_{\text{std}}} \qquad \text{and} \qquad \tilde{y}_i = \frac{y_i - y_{\text{mean}}}{y_{\text{std}}} \tag{2.3}$$

where $V_{\text{mean}} = 10925.34$ and $V_{\text{std}} = 3360.66$ for the sum of modulus implicit feature V of the Mechanical MNIST training dataset, and $y_{\text{mean}} = 567.52$ and $y_{\text{std}} = 47.35$ corresponds to the output change in strain energy y of the Mechanical MNIST training dataset. Parameter r in this selection mechanism controls the direction and the strength of the spurious correlation: positive spurious correlation occurs when $r > 1$, and negative spurious correlation occurs when $r < -1$. Specifically, when $r > 1$, the data points with the implicit spurious feature $\tilde{V}_i$ more highly correlated to their final change in strain energy $\tilde{y}_i$ will have a larger chance of being selected; when $r < -1$, the data points with the negative value of its implicit spurious feature $\tilde{V}_i$ closer to their final change in strain energy $\tilde{y}_i$ will have a larger chance of being selected. In addition, larger $|r|$ means larger selection bias. Note that for ease of implementation of the sampling process, the final selection probability of each data

point is normalized as $\tilde{P}_i = P_i / \sum_{j=1}^{n} P_j$ where n is the total number of data points. This ensures that the sum of the probability of all data points is equal to 1, which is statistically meaningful. Based on this selection mechanism, two training environments and three testing environments dictated by the selection parameters are created and the details are as below:

- Training Environment 1: $r = 15$, data size = 9800 (7840 for training, 1960 for validation)
- Training Environment 2: $r = -2$, data size = 200 (160 for training, 40 for validation)
- Testing Environment 1: $r = -5$, data size = 2000
- Testing Environment 2: $r = -10$, data size = 2000
- Testing Environment 3: $r = 1$, data size = 2000

Fig. 2·3a illustrates the relationship between the implicit spurious feature $\tilde{V}_i$ and the output $\tilde{y}_i$ for each environment. The color of each plot background represents the logarithmic selection probability of each area calculated by eqn. 2.3. Note that the selection probability is normalized through being divided by 10000, which is the total number of pixel points that construct the background of the image. In training environment 1 where $r > 1$, data with positive spurious correlations will be overrepresented in the environment, thus $\tilde{y} \approx \tilde{V}$ throughout the dataset. In training environment 2, testing environment 1, and testing environment 2 where $r < -1$, data points with a negative spurious correlation will be overrepresented in the environment, thus data points around $\tilde{y} = -\tilde{V}$ have a larger probability to be selected. In testing environment 3 where $r = 1$, the selection probability is identical for every data point (i.e., $P_i = 1/n$), thus testing environment 3 is a representative sampling of the entire Mechanical MNIST dataset. Fig. 2·3b-c presents the data input and output distributions following the format in Fig. 2·1-2·2b and c. As shown in Fig. 2·3a, the implicit spurious features and the output strain energy are strongly correlated in the training dataset

but not in the testing datasets. Furthermore, it is worth noting that note that in Fig. 2·3b, though the data in training environment 1 is from a biased selection, in terms of the first principle component of PCA, the biased selection data distribution of training environment 1 (mean = -1.38, standard deviation = 216.65) and testing environment 3 (representative Mechanical MNIST dataset, mean = 65.06, standard deviation=211.18) are within one standard deviation of each other, which demonstrates that it can be non-trivial to detect the presence of implicit spurious features.

2.5 Mechanical MNIST – EMNIST Letters Collection

The original Extended MNIST (i.e., EMNIST) dataset is a benchmark image dataset following the same conversion paradigm used to create the MNIST dataset, except that each image in the original EMNIST Letters dataset is a letter (a-z, A-Z) rather than a digit (0-9) (Cohen et al., 2017). Identical to the MNIST dataset, each input domain is represented by a 28×28 pixel bitmap. The Mechanical MNIST – EMNIST Letters dataset is created following the same process as described in Section 2.1 for Mechanical MNIST. Specifically, the Mechanical MNIST – EMNIST Letters dataset is created by transforming the 28×28 image bitmaps to a map of the elastic modulus for each input pattern in EMNIST Letters through eqn. 2.1. Furthermore, a covariate shift dataset is created through the same process described in Section 2.2, a mechanism shift dataset is created through the same process described in Section 2.3, and a sampling bias dataset is created through the same process described in Section 2.4. Note that when creating the sampling bias dataset for Mechanical MNIST – EMNIST Letters through the selection probability defined in eqn. 2.3, $V_{\text{mean}} = 14151.59$ and $V_{\text{std}} = 4045.36$ are the mean and standard deviation of the spurious feature V of the Mechanical MNIST – EMNIST Letters training dataset, and $y_{\text{mean}} = 708.50$ and $y_{\text{std}} = 88.24$ corresponds to the mean and the standard deviation of the output change in strain energy y of the Mechanical MNIST – EMNIST Letters training dataset.

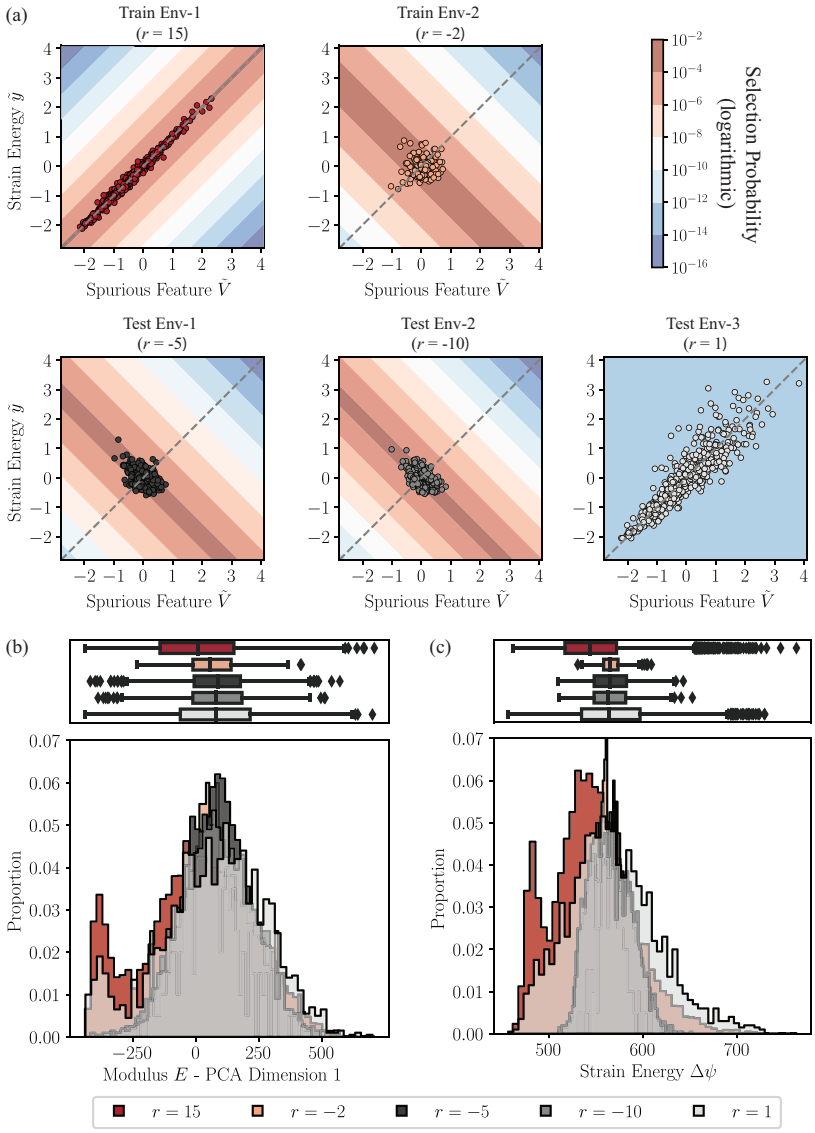


Figure 2·3: Illustration of the sampling bias dataset. (a) Relationship between the spurious feature and the target (the total change in strain energy) for each environment. Note that only 500 randomly selected data points for each environment (with the exception of the 200 available data points for the training environment 2) are shown to aid in visualization. The color of each plot background represents the logarithmic selection probability of each area (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

Section 2.1-2.4 presented the different OOD training and test environments created based on the Mechanical MNIST dataset in Fig. 2·1-2·3. Here, Fig. 2·4-2·6 shows the distribution of the EMNIST Letters data. Specifically, Fig. 2·4 illustrates the OOD data distribution for covariate shift, Fig. 2·5 illustrates the OOD data distribution for mechanism shift, and Fig. 2·6 illustrates the OOD data distribution for sampling bias.

In the next Chapter, a comprehensive study of the OOD generalization problems in mechanics based on the benchmark datasets created in this Chapter will be conducted.

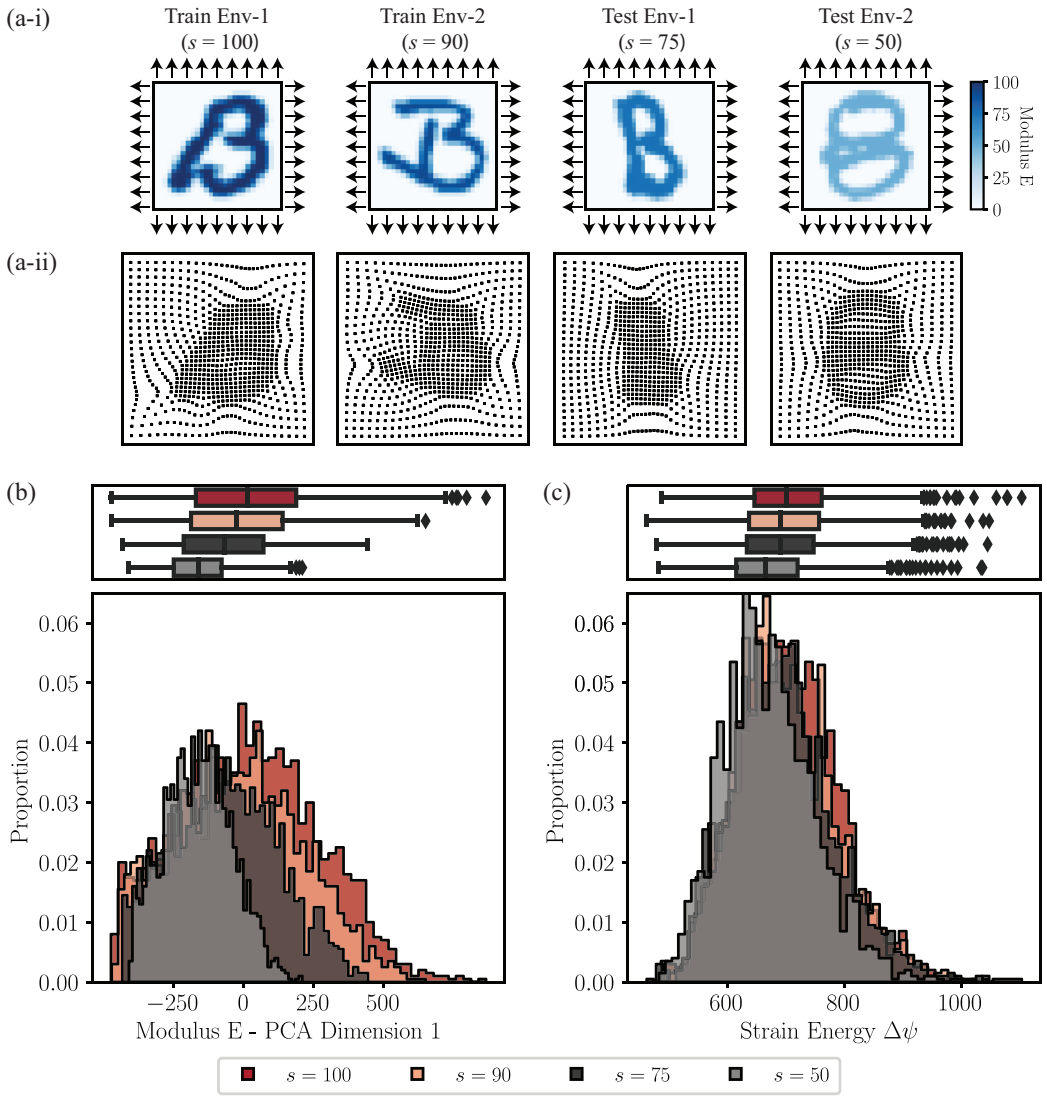


Figure 2·4: Illustration of the EMNIST covariate shift dataset. (a-i) Equibiaxial extension boundary conditions and elastic modulus distribution for a representative example from each environment. (a-ii) Deformation of each example in (a-i) after the completion of the equibiaxial extension simulations. (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

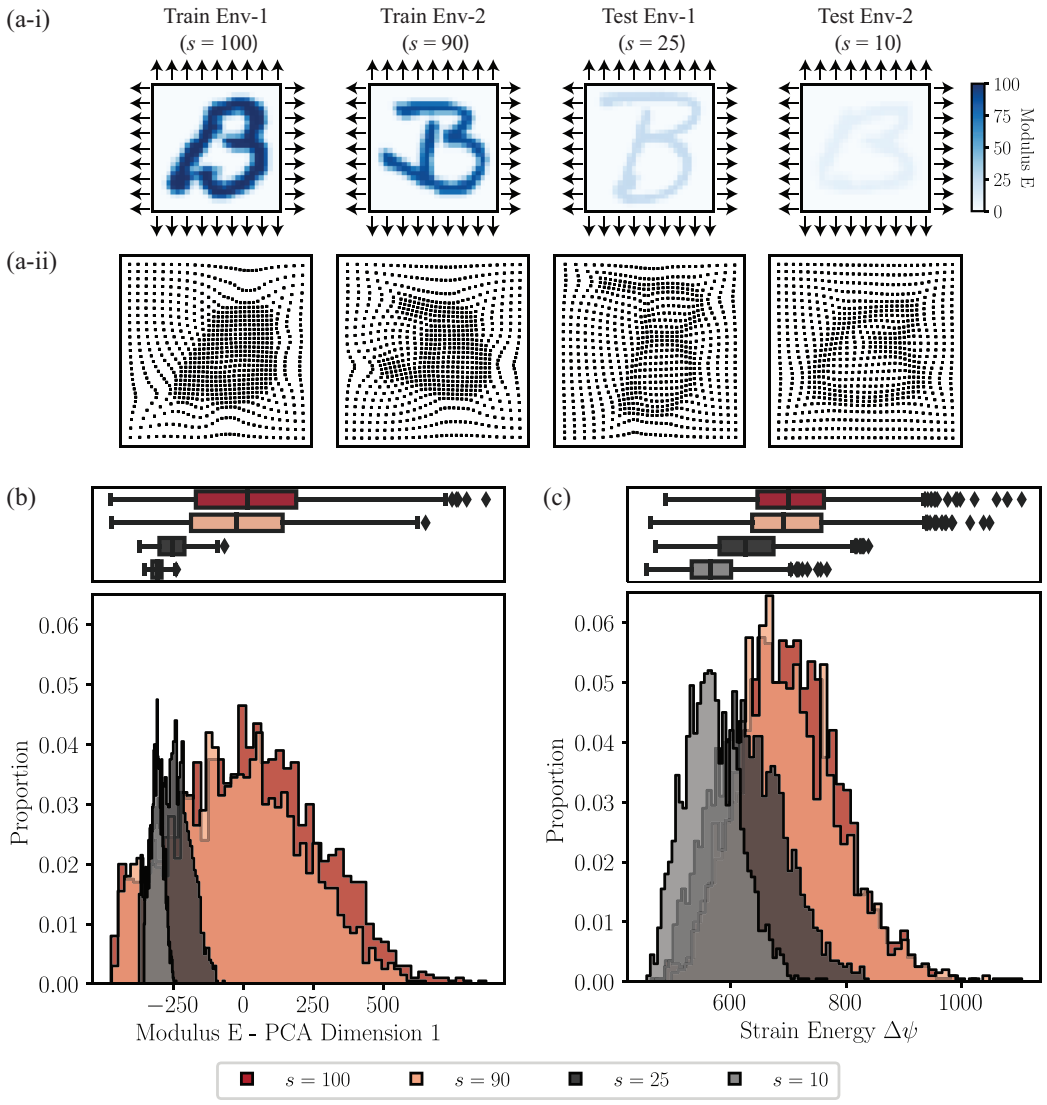


Figure 2·5: Illustration of the EMNIST mechanism shift dataset. (a-i) Equibiaxial extension boundary conditions and elastic modulus distribution for a representative example from each environment. (a-ii) Deformation of each example in (a-i) after the completion of the equibiaxial extension simulations. (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

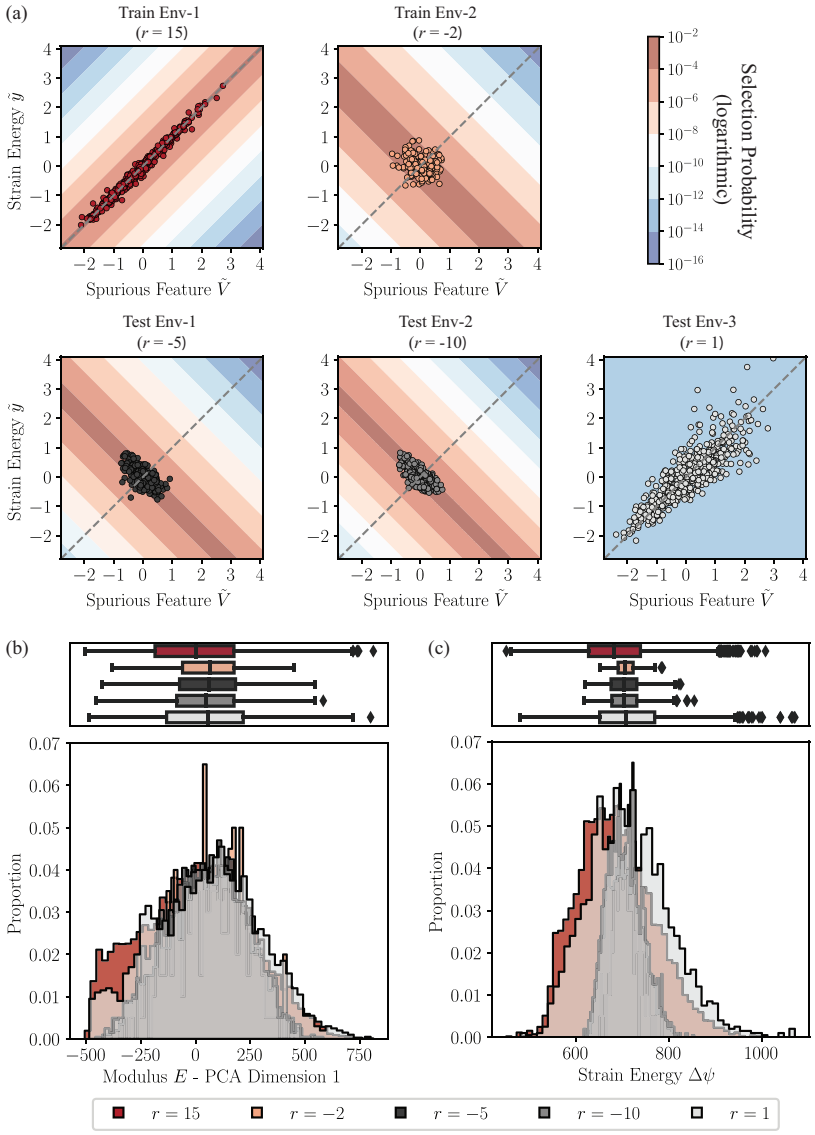


Figure 2·6: Illustration of the EMNIST sampling bias datasets. (a) Relationship between the spurious feature and the target (the total change in strain energy) for each environment. Note that only the 500 randomly selected data points for each environment (with the exception of the 200 available data points for training environment 2) are shown to aid in visualization. The color of each plot background represents the selection probability of each area (b) Input distribution of all environments described by the coefficient of the first principle component obtained through PCA performed on the input elastic modulus distribution of training data. (c) Output distribution of all environments defined as the total change in strain energy of the domain. Note that in (b-c) the histograms and boxplots are two ways of showing the same data.

Chapter 3

OOD Generalization Algorithms and Performance

3.1 Algorithms and Notation

For the supervised learning problems that we will pursue in this study, the input space is defined as $X \subseteq \mathbb{R}^{784}$ and the output space is defined as $y \subseteq \mathbb{R}^1$. The goal of machine learning models is to find a predictor $f_\theta(X)$ that minimizes the loss function $l(f_\theta(X), y)$. Traditionally, the training (observed) data and test (unseen) data are assumed to be independent and identically distributed (i.i.d.). Thus, the optimal predictor can be obtained by minimizing the loss of the model on the training data. This method is referred to as Empirical Risk Minimization (ERM). However, out-of-distribution (OOD) generalization problems deviate from the i.i.d. assumption underlying ERM. In other words, for OOD problems we cannot assume that the training data and testing data are i.i.d. This means that the distribution of unseen data is allowed to shift, which is practically relevant to real world problems. Specifically, OOD generalization approaches acknowledge that the observed training data are collected from different environments $\mathcal{E}^{\text{train}} = \{e_1, ...e_m\}$, which are only a subset of all environments $\mathcal{E}^{\text{all}}$. For an OOD generalization method to perform well, this implies that the predictor $f_\theta(X)$ obtained through learning from the training environments should perform well across all unseen environments that are under consideration. This goal can be expressed by minimizing the worst-case risk defined as:

$$R^{\mathrm{OOD}}(\theta) = \max_{e \in \mathcal{E}^{\mathrm{all}}} R^e(\theta) \tag{3.1}$$

where θ represents the parameters of the predictor (e.g., weights for a neural network). To minimize the OOD risk defined in eqn. 3.1, Arjovsky et al. (Arjovsky et al., 2019) argued that ML models should learn the causation that truly defines the outcome. In defining causation, we distinguish it from correlation, where correlation can be either spurious or causal. A ML model that learns a spurious correlation between input and output can make accurate predictions in training environments, but not in test environments. This is the main reason why a ML model driven by ERM fails in OOD generalization. In contrast, a causal correlation is one that is invariant, and thus does not change across different environments. As a result, a ML model that learns the causation would perform consistently well across all environments, which is the ultimate goal of OOD generalization.

Because causation does not change across environments, Arjovsky et al. (Arjovsky et al., 2019) promoted "invariance" as the main feature of causation and determined that ML models that perform well in OOD generalization should find a predictor that learns the invariant correlation across all environments. We call a predictor that reaches this goal an invariant predictor. In addition to the general requirement for a predictor that it should perform well on the whole training data set collected from all environments, the invariant predictor is also required to exhibit a second quality, referred to as "invariance," which is defined as having consistent performance across every environment.

While the ERM method achieves the general requirement for a predictor, how to get an invariant predictor that exhibits invariance across different environments is the main challenge for developing OOD generalization methods. In the remainder of this Section, we first introduce ERM as a baseline method, and then introduce three additional widely applied OOD algorithms that aim to find an invariant predictor.

3.1.1 Empirical Risk Minimization (ERM)

The Empirical Risk Minimization (ERM) approach assumes that the training data and the test data are i.i.d. Thus, the optimal predictor f_θ can be found by minimizing the average risk of all training environments. The risk form of ERM is defined as:

$$R^{\mathrm{ERM}} = \sum_{e=1}^{m} R_e(\theta) \tag{3.2}$$

where e denotes each training environment. In brief, ERM makes no special acknowledgments that the problem at hand is an OOD problem. Thus, ERM is typically used to define a baseline prediction.

3.1.2 Invariant Risk Minimization (IRM)

Motivated by the idea that causation often does not happen explicitly through input variables of a dataset (e.g., in computer vision the collection of correlated pixels that define a recognizable object are often not explicitly definable), Arjovsky et al. (2019) proposed the Invariant Risk Minimization (IRM) algorithm to search for an invariant predictor through finding a data representation for which the optimal predictor is the same for all environments. Formally, they define a data representation $\Phi : \mathcal{X} \to \mathcal{H}$ that elicits an invariant predictor $w \circ \Phi$ across environments $\mathcal{E}^{\mathrm{train}}$ if there is a classifier w for $\bar{w} : \mathcal{H} \to \mathcal{Y}$, that is simultaneously optimal for all environments, i.e., $w \in \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{Y}} R_e(\bar{w} \circ \Phi),$ for all $e \in \mathcal{E}^{\mathrm{train}}$. Note that the authors of IRM method used the term "classifier" to denote the last layer $\bar{w}$ for both classification and regression problems. Though in this work only regression problems in mechanics are considered, the term "classifier" is still adopted in order to keep it consistent with the original IRM paper (Arjovsky et al., 2019). The mathematical form of IRM is given as:

$$\min_{\substack{\Phi:\mathcal{X}\to\mathcal{H}\\ w:\mathcal{H}\to\mathcal{Y}}} \sum_{e=1}^{m} R_e(w\circ\Phi) \text{ subject to } w\in \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{Y}} R_e(\bar{w}\circ\Phi), \text{ for all } e\in\mathcal{E}^{\text{train}}. \tag{3.3}$$

To help interpret the terms in eqn. 3.3 in the context of neural networks, the corresponding components are schematically illustrated in Fig. 3·1. In this example, training data was collected from m environments. The input and output data of each training environment is defined as $\{X_{e_i}, y_{e_i}\}$. The data representation Φ that transforms X_{e_i} to H_{e_i} is the same across all environments. Determining the associated weights of Φ, w_{e_i}, is the goal of the IRM algorithm. The optimal w_{e_i} for each training environment is found by minimizing the loss $l(\hat{y_{e_i}}, y_{e_i})$ between $\hat{y_{e_i}} = w_{e_i}\circ\Phi(X_{e_i})$ and y_{e_i}, i.e., $R_i(w_{e_i}\circ\Phi)$. Thus, in this example the goal of eqn. 3.3 is to search for a data representation Φ where w_{e_i} is identical across all training environments, i.e., $w_{e_1} = w_{e_2} = \ldots = w_{e_m}$.

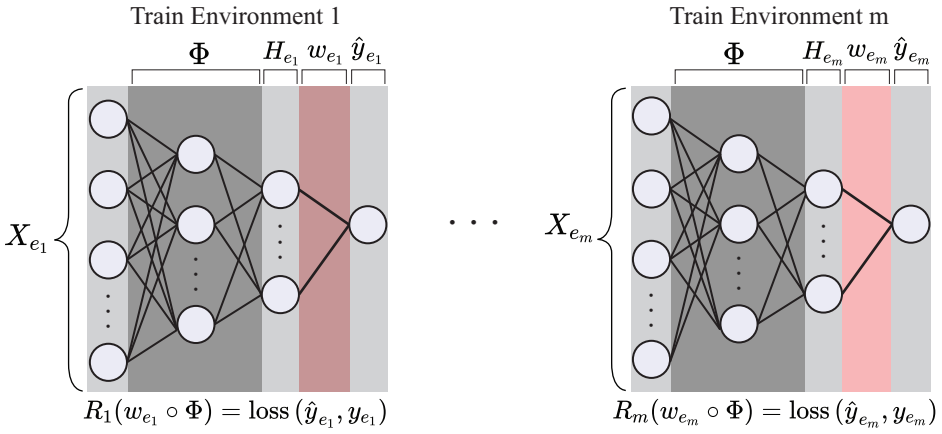


Figure 3·1: Example illustration for IRM.

Because solving the bi-level optimization problem defined in eqn. 3.3 is very challenging, Arjovsky et al. (2019) simplified the optimization problem by assuming that the optimal classifier w is a linear and fixed vector $w = w^*$. Thus, the risk given by the practical version of IRM is:

$$R^{\text{IRM}} = \sum_{e=1}^{m} R_e(w^* \circ \Phi) + \lambda \cdot \left\| \nabla_{w|w=w^*} R_e(w \circ \Phi) \right\|^2 . \tag{3.4}$$

For this version of IRM, the goal becomes to find a data representation Φ such that the optimal w_{e_i} of each training environments is w^*. The first item of eqn. 3.4 measures the predictive power of the predictor $w^* \circ \Phi$ on the training data environments. The second term is the gradient penalty that measures the optimality of the choice $w = w^*$ for all training environments. Because the practical form assumes that the optimal w for all training environments is $w = w^*$, the gradient of $R_e(w \circ \Phi)$ should reach its minimum at $w = w^*$, i.e., the gradient of $R_e(w \circ \Phi)$ with respect to w should be zero at $w = w^*$. More discussion about the relationship between the original objective eqn. 3.3 and the practical objective eqn. 3.4 can be found in Arjovsky et al. (2019). Then, Arjovsky et al. (2019) set w^* to be a fixed scalar $w^* = 1.0$ such that the practical form of IRM is given by:

$$R^{\text{IRM}} = \sum_{e=1}^{m} R_e(\Phi) + \lambda \cdot \left\| \nabla_{w|w=1.0} R_e(w \cdot \Phi) \right\|^2 . \tag{3.5}$$

This practical version of IRM is composed of two terms with a penalty weight λ that controls the balance between the terms. The first term is identical to the ERM term defined in eqn. 3.2. The second term is the gradient penalty that measures the optimality of the pragmatic choice $w = 1$ for all training environments. The evaluation of IRM method on out of distribution tasks through its practical form eqn. 3.5 is in Section 3.4.

3.1.3 Risk Extrapolation (REx)

Inspired by IRM, Krueger et al. (2021) aim to find an invariant predictor through finding an "equipredictive" representation Φ which they defined as a data representation with the property that the distribution $P_e(y|\Phi)$ is equal for $e \in \mathcal{E}^{\text{all}}$. In order to help build a geometric intuition of REx, they emphasized that their method is an extension to another OOD generalization method, Distributionally Robust Optimization (DRO) (Sagawa et al.,

2019), the objective of which is a risk interpolation defined as:

$$R^{\text{RI}} = \max_{\substack{\sum_e \lambda_e = 1 \\ \lambda_e \geq 0}} \sum_{e=1}^{m} \lambda_e R_e(\theta) \tag{3.6}$$

The REx method instead considers that the coefficient of the risk from each environment can be negative such that it allows us to extrapolate to more extreme variations. Since the final goal of OOD generalization is to minimize the maximal risk (or the worst case risk) across all environments, they call their method as minimax Risk Extrapolation (MM-REx), the objective of which is defined as:

$$R^{\text{MM-REx}} = \max_{\substack{\sum_e \lambda_e = 1 \\ \lambda_e \geq \lambda_{\text{min}}}} \sum_{e=1}^{m} \lambda_e R_e(\theta) \tag{3.7}$$

where λ_{min} is a hyperparameter that controls how much to extrapolate. To minimize eqn. 3.7, the optimal solution is obtained at $R_1 = R_2 = \ldots = R_m$. In other words, REx encourages the equality of risks from different environments, which can also be achieved by minimizing the variance of risks across training environments. In practice, they found the optimization landscape is smoother by using the variance of risks to be the penalty term of the risk objective. Therefore their practical form of the risk extrapolation is given by:

$$R^{\text{REx}} = \sum_{e=1}^{m} \mathcal{R}_e(\theta) + \lambda \text{Var}(\{\mathcal{R}_1(\theta), \ldots, \mathcal{R}_m(\theta)\}). \tag{3.8}$$

Similar to eqn. 3.5, the form of REx is composed of an ERM term and a penalty term where a penalty weight λ controls the balance between the two terms. In eqn. 3.8, the penalty term measures the variance of risks across training environments. By directly focusing on the invariance of risks, the creators of REx argue that it provides robustness to covariate shift where IRM can easily fail. The veracity of this claim will be evaluated in Section 3.4 through its practical form eqn. 3.8.

3.1.4 Inter Gradient Alignment (IGA)

Inspired by information theory, Koyama and Yamaguchi (2020) attempts to find a data representation Φ that maximizes the mutual information between Φ and the output interest y. In statistics, the mutual information measures the mutual dependence between two variables. It becomes zero if Φ and y are independent, and becomes largest if Φ is a deterministic function of y. They called the invariant predictor found by this method as the maximal invariant predictor (MIP). To seek the MIP, they proposed Inter Gradient Alignment (IGA) that forces the risk gradient of each training environment to align with each other, i.e., $\nabla_\theta R_1 = \nabla_\theta R_2 = \ldots = \nabla_\theta R_m$. This was implemented by minimizing the variance of the gradient of risks across all environments. Thus the risk form of the IGA algorithm is defined as:

$$R^{\text{IGA}} = \sum_{e=1}^{m} \mathcal{R}_e(\theta) + \lambda \,\text{trace}\left(\text{Var}(\{\nabla_\theta R_1(\theta), \ldots, \nabla_\theta R_m(\theta)\})\right). \tag{3.9}$$

Like with eqn. 3.5 and eqn. 3.8, the first term of the IGA equation is an ERM term. In eqn. 3.9, the second term is the trace of the variance of risk gradient, and λ is a penalty weight control parameter that controls the balance between the two terms. The mathematical relationship between MIP objective and IGA algorithms can be found in Koyama and Yamaguchi (2020)

3.1.5 Limitations

The risk functions of the three algorithms designed for OOD generalization problems all follow the same form. The first term is an ERM term that corresponds to the general quality of a predictor, i.e., low error on training data. The second term penalizes the variance of the predictor across all training environments $\mathcal{E}_{\text{train}}$, which aims to reach the second quality of an invariant predictor defined in Section 3.1, i.e., invariance across all environments $\mathcal{E}^{\text{all}}$. To make sure the invariant predictor for $\mathcal{E}^{\text{train}}$ is also the invariant predictor for

$\mathcal{E}^{\text{all}}$, the invariance across $\mathcal{E}^{\text{train}}$ should imply the invariance across $\mathcal{E}^{\text{all}}$ (Arjovsky et al., 2019). However, this is difficult to realize for real world data. On the other hand, based on the different assumptions underlying each of these algorithms, they may optimally target different types of OOD problems, though unfortunately there exists no guide for the optimal ODD problem for a given OOD algorithm. In Section 3.4 these algorithms will be tested on the different types of OOD problems defined in Chapter 2 to have a better understanding on how they perform on different OOD problems specific to mechanics data.

3.2 Machine Learning Models

This Section will briefly introduce the two base models used to test the four algorithms introduced in Section 3.1 on the three OOD datasets introduced in Chapter 2. The first model is a Multilayer Perceptron (MLP) Model, while the second model is a LeNet (LeCun et al., 1998) model which was originally designed for digit classification on the MNIST dataset. In comparison to the original LeNet model that was designed for classification problems, the activation (Softmax $\rightarrow$ ReLu) and pooling (Avepooling $\rightarrow$ Maxpooling) layers of the LeNet model were modified to better serve regression problems from the Mechanical MNIST dataset. Here, Fig. 3·2 shows the schematic illustrations of the MLP model and the modified LeNet model. The MLP model is a feedforward Neural Network composed of forward fully connected layers and ReLU activation layers. The LeNet model is a Convolutional Neural Network composed of convolutional layers, maxpooling layers and ReLU activation layers (LeCun et al., 1998).

For the OOD datasets targeting covariate shift and mechanism shift, the true unscaled modulus values are used as model inputs because the shifts in the output feature strain energy are controlled by the scale of the modulus. Thus, rescaling the input could impede the OOD algorithms from learning invariant features between the different environments. For the OOD dataset targeting sampling bias where $s = 100$ is applied for all data, the inputs

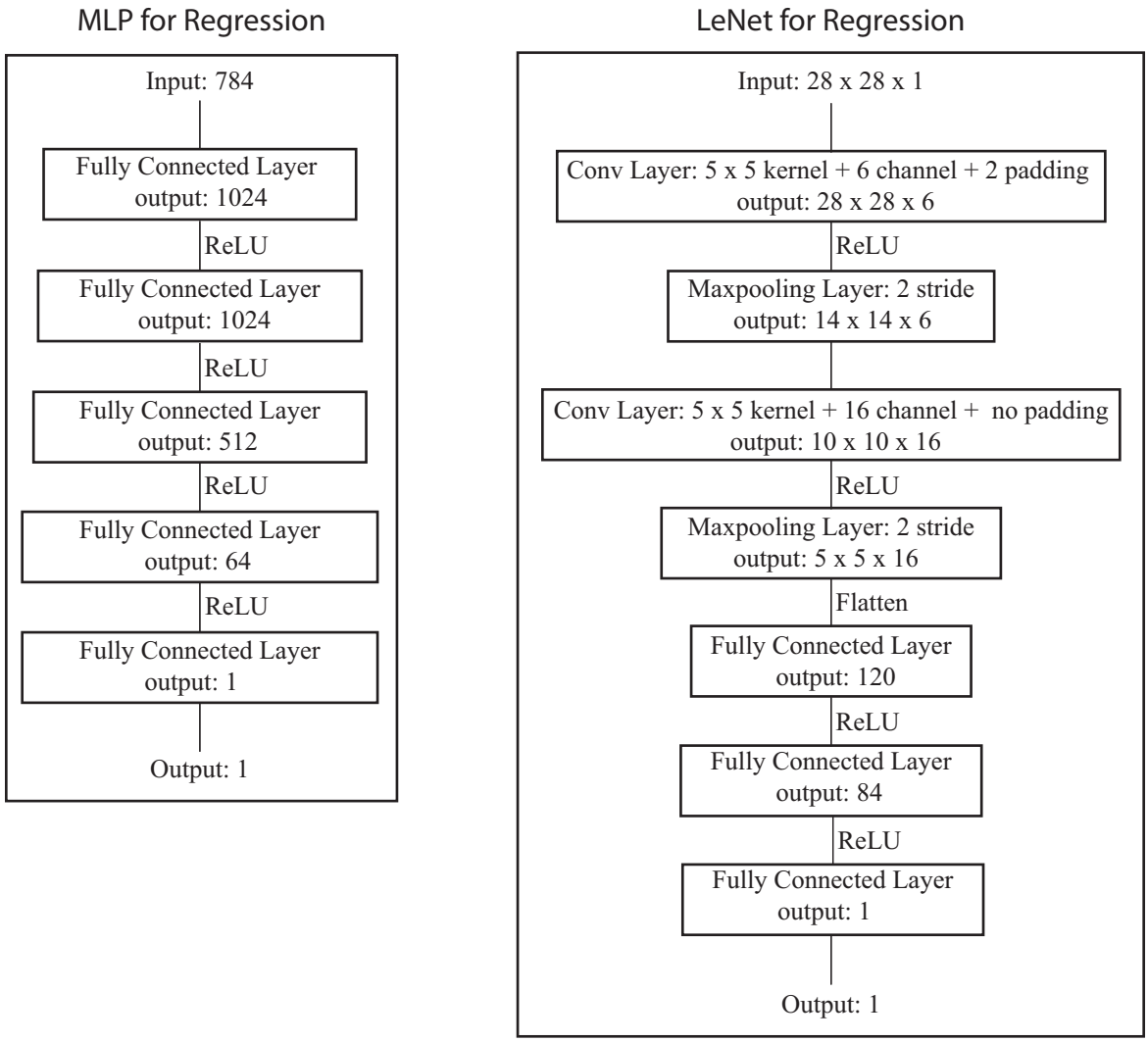


Figure 3·2: Illustration of model structures of the MLP model and the modified LeNet introduced in Section 3.2.

for the models are modulus values that were scaled to be within $0 \sim 1$ by dividing by 100 to help achieve faster model convergence.

3.3 Hyperparameter Tuning and Performance Evaluation Metrics

3.3.1 Model Selection and Hyperparameter Tuning

Although OOD generalization problems are very important for the real world applicability of many ML techniques, and many promising algorithms have recently been proposed (Shen et al., 2021), one major issue is that there does not exist a standardized approach to select models for such problems (Gulrajani and Lopez-Paz, 2020). In ML frameworks, it is imperative that performance is ultimately evaluated on unseen test data (Gulrajani and Lopez-Paz, 2020). For OOD approaches, the delineation between test and training datasets remains essential, yet the violation of the i.i.d. assumption between the test and training datasets adds another layer of complication. Because each test environment can be qualitatively and quantitatively different, it is likely that performance will vary widely across different test environments for an identical set of model hyperparameters. Although existing OOD algorithms have shown robustness on shifted test datasets in comparison to the ERM method, these performance enhancements are obtained by tuning the hyperparameters of the OOD algorithm (i.e., tuning the penalty weight term in the risk function) on test environments (Krueger et al., 2021). In this work, special care is taken to define test environments that are truly "unseen" i.e., are not used in any way to select or tune model hyperparameters. To accomplish this, the training datasets are split into training (80%) and validation datasets (20%) while the hyperparameters are tuned on the validation datasets alone. For the OOD algorithms introduced in Section 3.1, the similar implementation process in (Arjovsky et al., 2019) is used. Specifically, there are two main hyperparameters to tune: the penalty weight λ, and the anneal step t, after which we add the penalty (invariant term) to the loss function. The anneal step is determined by the step after which the validation error stops decreasing for ERM. During the training process, the penalty weight is set to $10^{-4}\lambda$ before the anneal step and set to be λ afterwards. The final optimal penalty weight is then selected as the weight that reaches the lowest mean validation error for three models

trained with different random initializations. Note that if the penalty weight is too small, the penalty term has little influence on the training process such that the training history of the OOD algorithms will be similar to ERM, and so we neglect such small penalty weights during the hyperparameter tuning process. One example of this is shown in Fig. 3·3, which compares the training history of the modified LeNet using ERM and IRM algorithm on the covariate shift dataset defined in Section 2.2. In Fig. 3·3(a), the penalty weight of IRM before the anneal step (15000) is 10^{-10}, which is small enough such that the training history is almost the same as ERM. The penalty weight changes to $\lambda = 10^{-6}$ from the anneal step where the training error suddenly increases at step $t = 15000$. However, in Fig. 3·3, the penalty weight $\lambda = 10^{-7}$ is too small, as the training error after the anneal step still follows the track of ERM, which means that the penalty term does not work. Thus, we will ignore the value $\lambda = 10^{-7}$ regardless of the magnitude of the validation error for the model trained with this penalty weight.

The final selection of hyperparameters for all approaches is given in Appendix A.1.

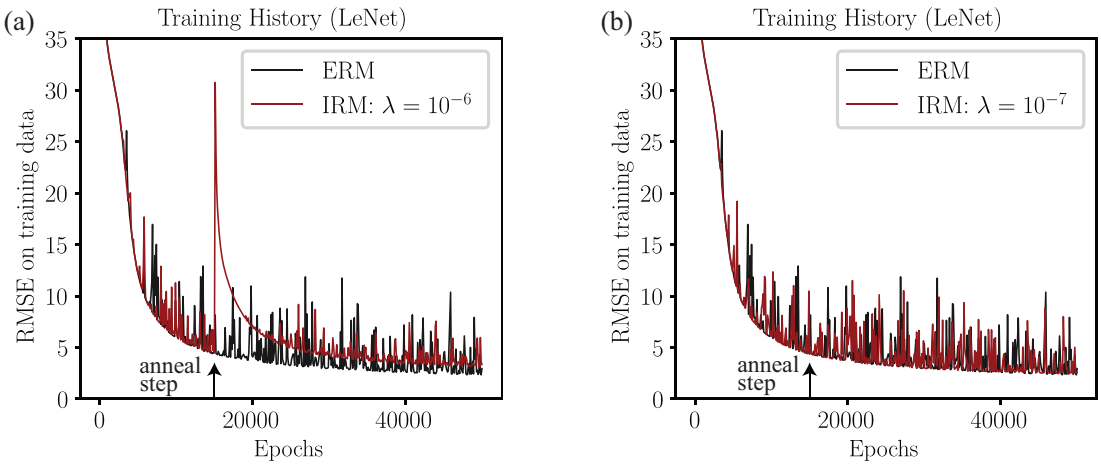


Figure 3·3: Example of how penalty weight influences the training error history of a modified LeNet. The penalty weight is formally introduced after the anneal step (15000). (a) training error history of ERM and IRM with penalty weight is equal to 10^{-6} (b) training error history of ERM and IRM with penalty weight is equal to 10^{-7}.

3.3.2 Evaluation Metrics

Two models described in Section 3.2 were trained 15 times each for each of the four algorithms with initializations defined by seeds from $k = [1,2,3\ldots15]$. The results are reported based on many initializations to ensure that the conclusions are not based on outlier results obtained due to the randomness of the training process. Note that for each approach, if the lowest training error obtained from an initialization is too large, where large is defined as three times higher than the lowest training error obtained from at least ten other initializations, the model will be re-trained with a different initialization not used in the initial group of 15. After that, the performance of each model on all test environments is evaluated by calculating the root mean squared error (RMSE) between the predicted change in strain energy $\hat{y}$ and the ground truth y. The effectiveness of each algorithm in every test environment is evaluated by two metrics. The first is the root mean squared error of the aggregate mean prediction across all 15 models with different initialization, which is defined as:

$$\overline{RMSE} = \sqrt{\left(y - \frac{1}{15}\sum_{k=1}^{15}\hat{y}_k\right)^2}. \tag{3.10}$$

The second is the average of the root mean squared error of all 15 models, which is defined as:

$$\overline{RMSE}' = \frac{1}{15}\sum_{k=1}^{15}\sqrt{\left(y - \hat{y}_k\right)^2}. \tag{3.11}$$

The RMSE of aggregated mean prediction defined in eqn. 3.10 is used to evaluate the four methods in Section 3.1 and the results are discussed in the following Section 3.4. The average of RMSE defined in eqn. 3.11 is represented in Appendix A.2.

3.4 Results and Discussion

This section examines the performance of all four algorithms introduced in Section 3.1 on the OOD datasets defined in Chapter 2. Section 3.4.1 presents the results for the OOD problem

caused by covariate shift. Section 3.4.2 presents the results for the OOD problem caused by mechanism shift. Section 3.4.3 presents the results for the OOD problem caused by sampling bias. Finally, Section 3.4.4 concludes common findings across all investigations.

3.4.1 ML Model Performance on the Covariate Shift Dataset

This Section evaluates the four different ML methods introduced in Section 3.1 on the two covariate shift datasets from Mechanical MNIST and Mechanical MNIST-EMNIST Letters. The problem definition and details of creating the covariate shift datasets were introduced in Section 2.2. For all four algorithms (ERM, IRM, REx, and IGA), the MLP model and the modified LeNet model are trained separately (see model architectures in Section 3.2). The performance of the MLP and LeNet model on the training, validation, and testing environments of each covariate dataset are shown in Fig. 3·4 for each algorithm. Specifically, Fig. 3·4a shows the performance results for the covariate shift dataset from Mechanical MNIST and Fig. 3·4b shows the performance results for the covariate shift dataset from Mechanical MNIST-EMNIST Letters. In each Figure, the root mean square error (RMSE) of the aggregate mean prediction described in eqn. 3.10 is shown. The performance of the baseline ERM method is illustrated with black markers, and the performance of the other three OOD algorithms is represented in red hues. For all algorithms, the RMSE of aggregate mean prediction is plotted for the training environment, the validation environment, and both test environments.

Fig. 3·4a-i shows the results of training the MLP model with all four algorithms on the covariate shift dataset from Mechanical MNIST. Here, the ERM approach achieves a very low error on the training data (RMSE 2.01) with a slightly higher error on the validation data (RMSE 20.85). For context, in Fig. 3·4a, the training dataset has mean 559.12, and standard deviation 45.99, and the validation dataset has mean 559.35, and standard deviation 45.74. For a typical ML modeling approach driven by ERM (i.e., traditional ML method), the model performance on the validation dataset is assumed to be very close to the performance

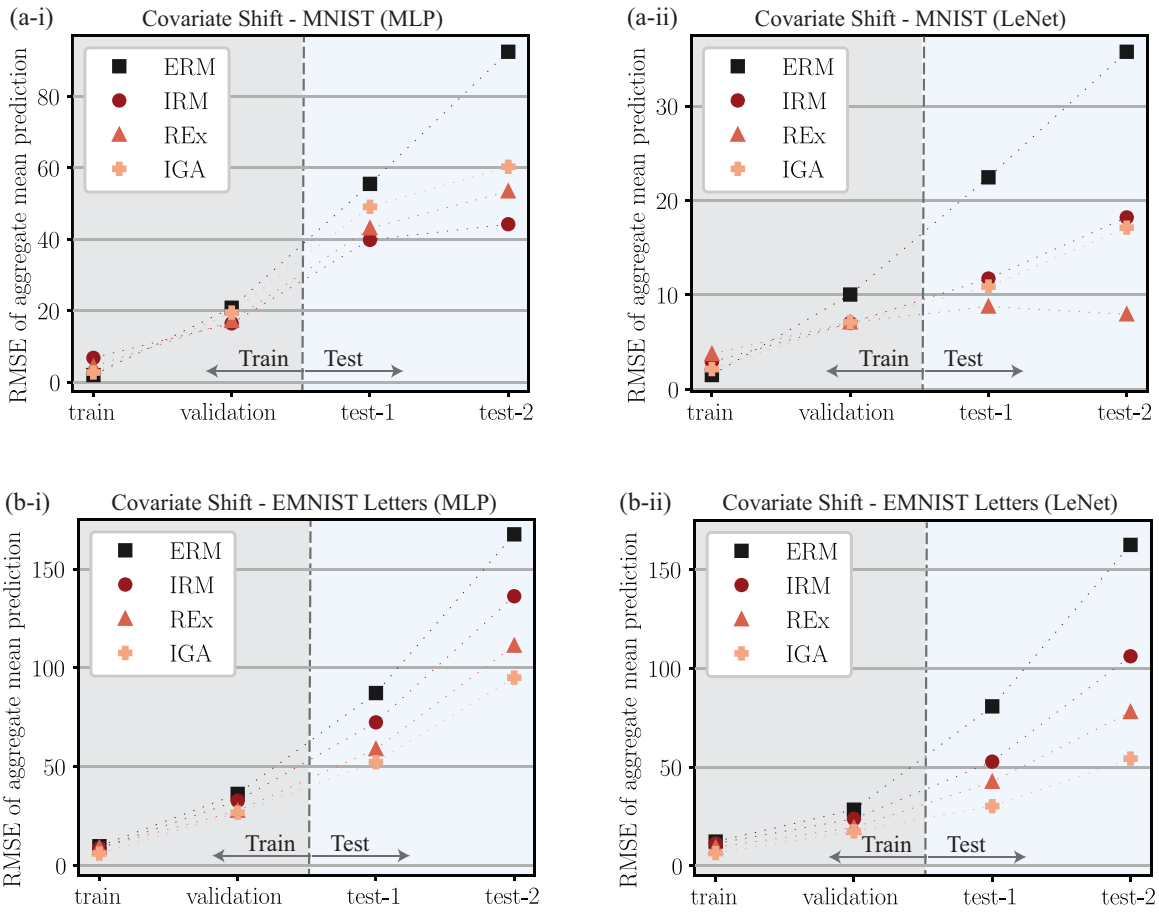


Figure 3·4: The performance of four algorithms (ERM, IRM, REx, IGA) on the covariate shift datasets defined in Section 2.2. The RMSE is calculated using eqn. 3.10. (a) The performance of a MLP model(a-i) and a modified LeNet model(a-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST Collection. (b) The performance of a MLP model(b-i) and a modified LeNet model(b-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST - EMNIST Letters Collection. Additional visualization of these results can be found in Appendix A.2, which shows a comparison of prediction vs. ground truth, and Appendix A.2 which shows representative samples of different error levels.

of the model on unseen test datasets. However, it is worth noting that compared to the validation data, the performance of ERM on both test environments with covariate shift is substantially worse. Specifically, the test error of test environment 1 is RMSE 55.53, and the test error of test environment 2 is RMSE 92.52. For the MLP trained by the three

OOD algorithms, the test error on both test environments 1 and 2 is lower than ERM. In addition to plotting these values in Fig. 3·4, they are listed in Table A.3 in Appendix A.2. Overall, the lowest test error is obtained by IRM with a validation error of RMSE 16.48, and test error of RMSE 39.85 on test environment 1 and RMSE 44.23 on test environment 2. Through the same evaluation process as the MLP model, the results on the covariate shift dataset from Mechanical MNIST obtained by a modified LeNet model are shown in Fig. 3·4a-ii. Overall, the performance of all methods improves because of the deeper and more powerful Convolutional Neural Network model architecture. However, consistent with the MLP results, the methods designed for the OOD problems outperform the ERM algorithm. In this case, the best performance is obtained by REx with the training and test error consistently below RMSE 10, with the error on test environment 1 (RMSE 8.77) being almost a third of ERM (RMSE 22.46), while the error on test environment 2 (RMSE 7.94) is more than four times smaller than ERM (RMSE 35.75).

The results of repeating this process on the covariate shift dataset from Mechanical MNIST - EMNIST Letters are shown in Fig. 3·4b. For context, in Fig. 3·4b, the training dataset has mean 703.01, and standard deviation 86.26, and the validation dataset has mean 706.98, and standard deviation 88.45. For the evaluation results on MLP in Fig. 3·4b-i, the ERM method obtained a low error on the validation dataset (RMSE 36.21) but behaved poorly on test error with RMSE 87.24 on test environment 1 and RMSE 167.57 on test environment 2. For the evaluation results on modified LeNet in Fig. 3·4b-ii, the ERM method again obtained a low error on validation dataset (RMSE 28.47) but a high test error with RMSE 80.77 on test environment 1 and RMSE 162.48 on test environment 2. Note that for ERM, although the validation error decreased by 21.38% by using a deeper convolutional neural network (LeNet) compared to the simple MLP model, the decrease of test error by using LeNet is very small, as the error drops by only 7.42% for test environment 1 and 3.04% for test environment 2. In contrast to ERM, the OOD algorithms still performed

better on test environments for both the MLP model as shown in Fig. 3·4b-i and the LeNet model as shown in Fig. 3·4b-ii. The best performance is obtained by IGA with RMSE 52.24 on test environment 1 and RMSE 95.04 on test environment 2 for the MLP model, while its RMSE was 30.27 on test environment 1 and 54.33 on test environment 2 for the modified LeNet model. And compared to ERM, the performance of all OOD algorithms was improved by using a deeper Neural Network. For example, compared to the MLP model, by implementing IGA on the modified LeNet model, the test error decreased by 42.06% on test environment 1 and 42.83% on test environment 2. More statistics about the data and the evaluation results are given in Table A.4 in Appendix A.2. Further discussion of these results is also given in Section 3.4.4.

3.4.2 ML Model Performance on the Mechanism Shift Dataset

The performance of two ML models is evaluated on the two mechanism shift datasets from both Mechanical MNIST and Mechanical MNIST-EMNIST Letters using the same method described in Section 3.4.1. The problem definition and processing details of the mechanism shift datasets were introduced in Section 2.3. Since the training environments in this Section are the same as in Section 3.4.1, the two ML models (MLP and modified LeNet) already trained in Section 3.4.1 are used to test the two new test environments with mechanism shift. For all algorithms introduced in Section 3.1, the aggregate mean prediction RMSE for the training environment, the validation environment, and both test environments are plotted in Fig. 3·5. Specifically, Fig. 3·5a shows the performance results for the mechanism shift dataset from Mechanical MNIST and Fig. 3·5b shows the performance results for the mechanism shift dataset from Mechanical MNIST - EMNIST Letters. Similar to the results for the covariate shift datasets in Section 3.4.1, for both Mechanical MNIST and Mechanical MNIST - EMNIST Letters, the three OOD algorithms perform better than the baseline ERM method on both test environments for both the MLP and LeNet models.

For the mechanism shift data from Mechanical MNIST, Fig. 3·5a-i shows the perfor-

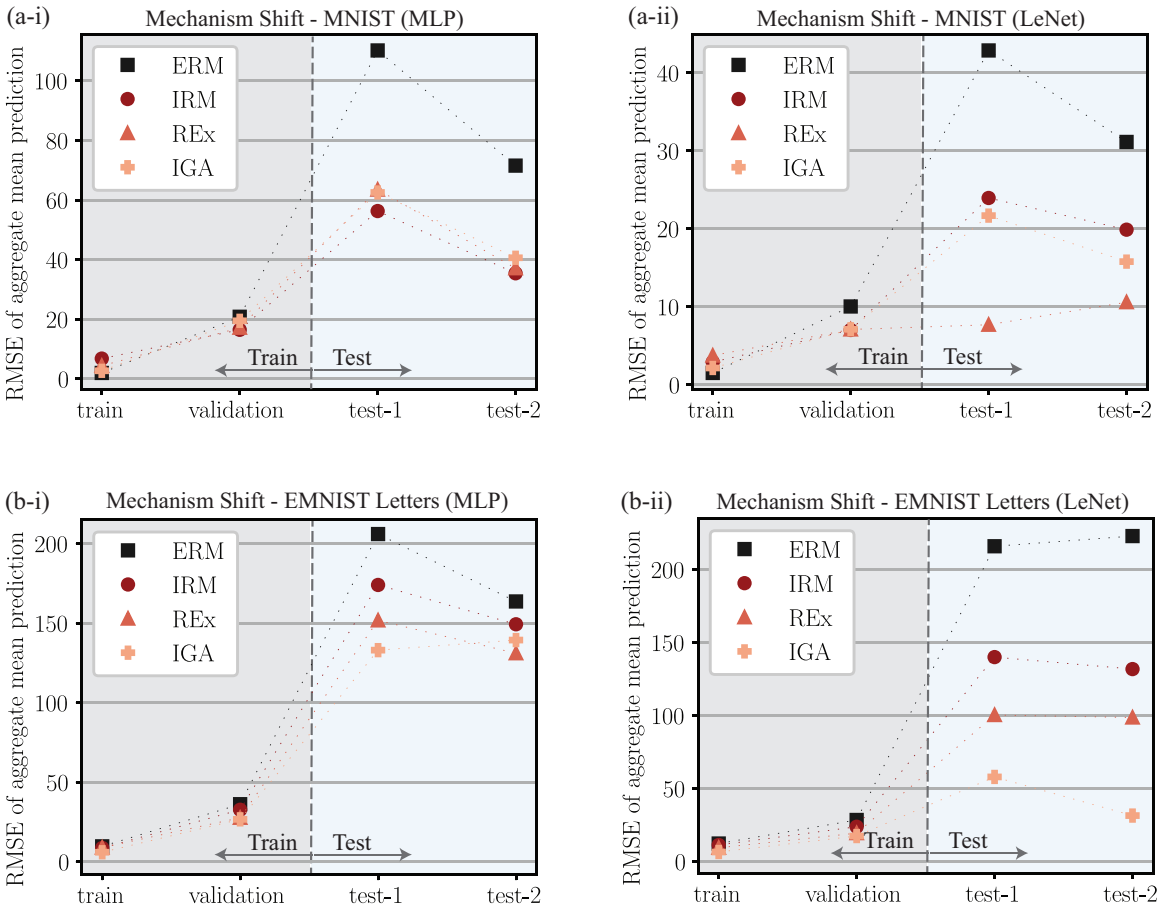


Figure 3·5: The performance of four algorithms (ERM, IRM,REx, IGA) on mechanism shift data defined in Section 2.3. The RMSE is calculated using eqn. 3.10. (a) The performance of a MLP model(a-i) and a modified LeNet model(a-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST Collection. (b) The performance of a MLP model(b-i) and a modified LeNet model(b-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST - EMNIST Letters Collection. Additional visualization of these results can be found in Appendix A.2, which shows a comparison of prediction vs. ground truth, and representative samples of different error levels.

mance of the four algorithms using the MLP model. The performance of the three OOD algorithms for a MLP model is similar with the IRM method achieving the lowest test error for both environments 1 (RMSE 56.31) and 2 (RMSE 35.43). In contrast, the test error achieved by ERM is RMSE 110.19 for test environment 1 and RMSE 71.55 for test

environment 2. In Fig. 3·5a-ii, which shows the performance of the four algorithms on the modified LeNet model, the test error of ERM drops significantly to RMSE 42.88 for test environment 1 and RMSE 31.15 for test environment 2. However, these numbers are still much larger than the test error obtained through the OOD algorithms. Specifically, REx achieves the lowest test error for both environments 1 (RMSE 7.70) and 2 (RMSE 10.59). Since the mechanism shift is a more challenging phenomenon to capture than the covariate shift due to a shift in both the input and output data, it is worth noting that for the ERM, the test error of test environment 1 in this Section is larger than the test error of the two test environments shown in Section 3.4.1, while the RMSE of REx is similar to that seen in the covariate shift example.

For mechanism shift data from the Mechanical MNIST - EMNIST Letters Collection, Fig. 3·5b-i shows the performance of the four algorithms using the MLP model. Compared to ERM which gets RMSE 206.02 on test environment 1 and 163.66 on test environment 2, the decrease in test error obtained by the OOD algorithms is still lower where IGA obtained the lowest error (RMSE 133.05) on test environment 1 while REx obtained the lowest error (RMSE 131.10) on test environment 2. Fig. 3·5b-ii shows the performance of the four algorithms on LeNet model. The performance of all OOD algorithms improved significantly with IGA obtaining the lowest error on both environments (RMSE 57.96 for test environment 1, and RMSE 31.49 for test environment 2) while ERM still obtained very high test error (RMSE 215.98 on test environment 1 and RMSE 222.95 on test environment 2). Further discussion of these results is given in Section 3.4.4.

Furthermore, the test error of test environment 2 in this Section drops slightly compared to the test environment 1 error for both mechanism shift datasets. Note that for mechanism shift datasets from both Mechanical MNIST and Mechanical MNIST - EMNIST Letters, the output standard deviation of test environment 2 is slightly smaller than the output standard deviation of the test environment 1 (see Table A.5 and Table A.6), which can cause test

environment 2 to get a smaller RMSE. Taking the mechanism shift dataset from Mechanical MNIST Collection as an example, the output standard deviation of test environment 2 is 31.37 while the output standard deviation of the test environment 1 is 37.94, as given in Table A.5.

3.4.3 ML Model Performance on the Sampling Bias Dataset

In this Section, the performance of four algorithms described in Section 3.1 is evaluated on OOD problems caused by sampling bias as described in Section 2.4. Similar to Section 3.4.1 and Section 3.4.2, the RMSE of aggregate mean prediction for the training environment, the validation environment, and all test environments are plotted on Fig. 3·6. The performance of the four algorithms on the sampling bias data from Mechanical MNIST is shown in Fig. 3·6a. The performance of ERM and IRM on all environments is very close for both the MLP and LeNet models. The REx method obtained lower test error than ERM for the MLP model, but unlike the covariate shift and mechanism shift examples, achieves no improvement for a LeNet model. The IGA method is the only algorithm that consistently performs better than ERM on test environments, though the improvements are relatively small. Specifically, for the MLP model, ERM achieved a RMSE of 29.54 on test environment 1, a RMSE of 26.63 on test environment 2, and a RMSE of 29.21 on test environment 3, while IGA achieved a RMSE of 16.25 on test environment 1, a RMSE of 15.91 on test environment 2 and a RMSE of 18.24 on test environment 3. For the LeNet model, ERM achieved a RMSE of 10.97 on test environment 1, a RMSE of 10.89 on test environment 2, and a RMSE of 13.22 on test environment 3. In contrast, IGA achieved a RMSE of 10.28 on test environment 1, a RMSE of 10.08 on test environment 2, and a RMSE of 11.74 on test environment 3. Although IGA performs better than ERM on test environments for a LeNet model, it is worth noting that the training and validation error (RMSE 6.21) of IGA was slightly larger than the validation error (RMSE 5.41) for ERM. Critically, this demonstrates that a lower training or validation error does not necessarily mean a lower test error if there is a strong spurious correlation

present in the training environments.

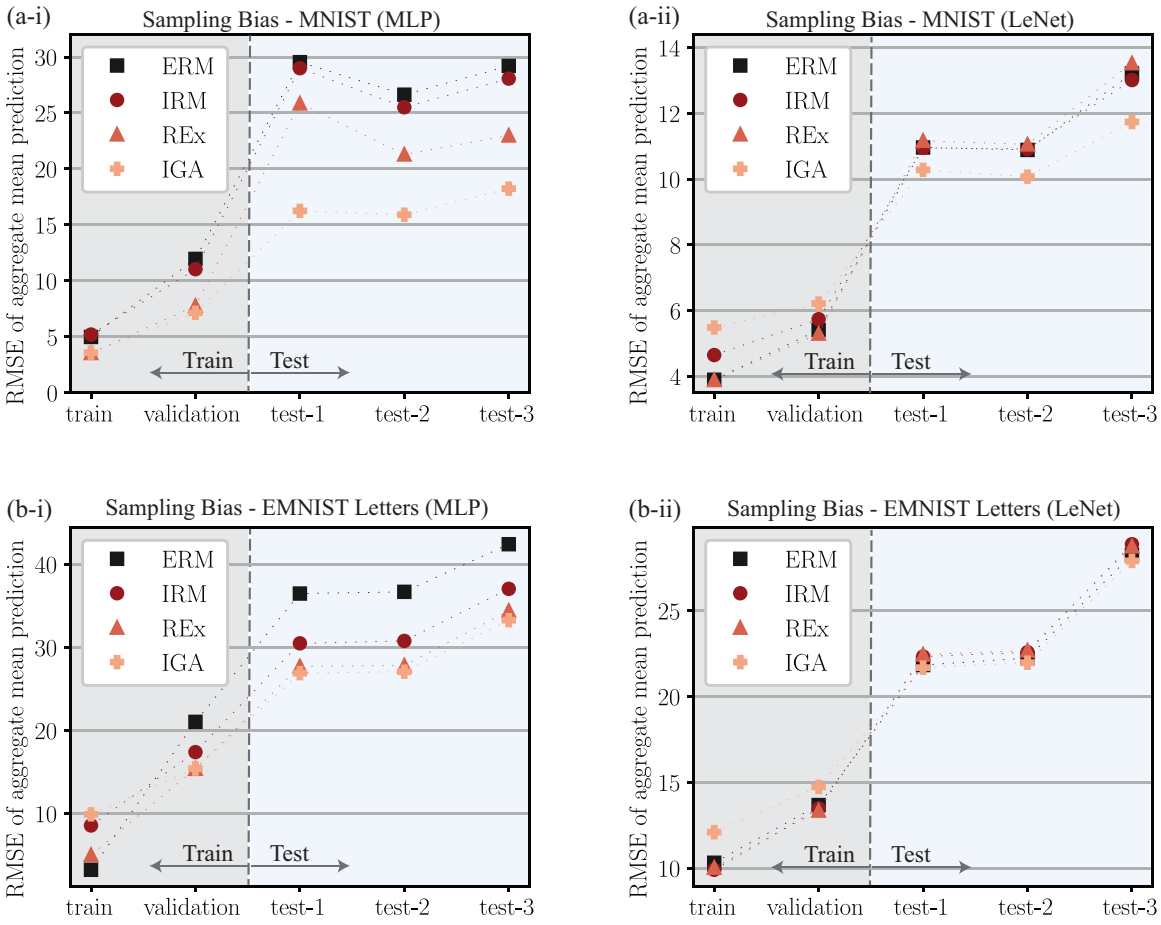


Figure 3·6: The performance of four algorithms (ERM, IRM,REx, IGA) on the sampling bias data defined in Section 2.4. The RMSE is calculated using eqn. 3.10. (a) The performance of a MLP model(a-i) and a modified LeNet model(a-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST Collection. (b) The performance of a MLP model(b-i) and a modified LeNet model(b-ii) trained by four algorithms on training, validation, and testing data from Mechanical MNIST - EMNIST Letters Collection. Additional visualization of these results can be found in Appendix A.2, which shows a comparison of prediction vs. ground truth, and representative samples of different error levels.

The performance of the four algorithms on the sampling bias data from Mechanical MNIST - EMNIST Letters is shown in Fig. 3·6b. For the performance on the MLP model in Fig. 3·6b-i, all three OOD algorithms obtained lower test error than the ERM, with IGA

obtaining the lowest test error where the RMSE is 26.91 for test environment 1, 27.08 for test environment 2, and 33.28 for test environment 3. In contrast, ERM obtained a RMSE 36.48 for test environment 1, 36.69 for test environment 2, and 42.42 for test environment 3. However for the performance on the LeNet model, as shown in Fig. 3·6b-i, the performance of the three OOD algorithms is similar to ERM, with no improvement observed for the three test environments. Further discussion of these results is in the next Section.

3.4.4 Common Findings Across All Datasets

Sections 3.4.1-3.4.3 show the results of all the OOD experiments that have been conducted. Across all experiments, the error on OOD test data using the traditional ERM method (not designed with OOD problems in mind) is much higher than the error that would be predicted based on the validation data. This indicates that models trained by ERM are vulnerable to suffering poor performance when exposed to OOD test data. Thus, even if a ML model obtains a low error for validation data that is i.i.d. with respect to the training data, the test error can be very high when it is applied to practical test situations when the test data distribution may be shifted. In contrast, it is found that the OOD generalization methods were effective in reducing the test error on OOD test data, in particular for the covariate shift and mechanism shift challenge problems. However, the effectiveness of the OOD algorithms in decreasing the test error on sampling bias data was small, in particular when using a deeper ML (LeNet) model. This makes the sampling bias problem that introduces spurious correlations the most challenging out of the three OOD problems in mechanics.

Secondly, although the OOD generalization methods try to find an invariant predictor enabling consistent performance across all environments, the test error is still much larger in comparison to the training and validation errors for all of the OOD experiments conducted in this work. One potential reason is that the hyperparameters are tuned on the validation dataset, which is still collected from the training environments. Thus there is bias in the model selection in terms of hyperparameter tuning. This means that while the model

selection method of tuning the network hyperparameters on the validation dataset can guarantee that the penalty weight selected is within a reasonable regime (i.e., sufficiently large such that it does impact the training process, but not too large such that it jeopardizes convergence), the selected penalty weight is not guaranteed to be the optimal penalty weight for each OOD dataset or for each algorithm. As a result of this, there is variability in performance for each OOD algorithm on the different OOD problems. Specifically, it is not found that one OOD algorithm can consistently outperform others.

Another possible reason why these OOD algorithms cannot obtain a test error as low as the training error is that the "invariance" term introduced by these algorithms is not sufficient for a ML model to learn all factors contributing to the invariance across environments. Thus, while some biased solutions can be avoided during the training process, the penalty introduced by these algorithms is not formulated such that a ML model will find the causal solution exclusively. This implies that there may need to be a re-evaluation of how invariance is defined in this context in order to make the OOD methods more generalizable for problems in mechanics. Specifically, assumptions on the " invariance" of each OOD generalization method should be more clear on how they are related to each type of OOD problem. This development would be critical to confidently solving OOD problems in mechanics, like predicting the mechanical behavior of materials.

Third, it is found that the test error for the covariate shift and mechanism shift datasets from the Mechanical MNIST - EMNIST Letters dataset is much larger than from the standard Mechanical MNIST examples. This can be attributed to the larger mean and the standard deviation of the covariate dataset from Mechanical MNIST - EMNIST Letters than the covariate shift dataset from Mechanical MNIST. For example, for the covariate shift data from the Mechanical MNIST Collection, the standard deviation of the target property, i.e., the change in strain energy, is 48.17 in test environment 1 and 44.90 in test environment 2, while for the covariate shift data from Mechanical EMNIST - Letters, the standard deviation

of the same target property is 86.30 in test environment 1 and 79.58 in test environment 2, both of which are about twice the standard deviation value in covariate shift data from Mechanical MNIST Collection. As a result, the RMSE of results on Mechanical MNIST - EMNIST Letters is also about twice larger than the RMSE of results on Mechanical MNIST Collection. More statistics of each dataset and the details of evaluation results can be found in Appendix A.2, Table A.3, and Table A.4. In addition, the data distribution the original MNIST and EMNIST Letters are not the same, which does not assure that the model trained on these two datasets can get the same performance.

Fourth, one objective of this work is to assess these OOD algorithms not only by their performance on different types of OOD problems, but also by their demand for computational resources and feasibility of implementation. To this end, it is worth noting that the penalty term of REx is the only one that does not require the computation of the gradient of risks during the training process. Thus its applicability is not restricted by computational capacity and can be broadly applied to deeper and more complex ML models. Since the performance of REx on OOD test data was consistently better using the deeper LeNet model, this may imply that REx is a better choice than the other two OOD algorithms for large and complex datasets that require deeper ML architectures for good performance. In contrast, the IGA method is more computationally intensive because it requires computing the risk towards all the weights of a Neural Network. However, the performance of IGA on the three OOD mechanics problems is overall the best as it outperforms the other two OOD algorithms in most scenarios. Thus, IGA may be the most appropriate choice when the problem of interest involves a small dataset without requiring a complex ML model. In contrast to IGA, the IRM method needs to compute the gradient of risk for only the weight of the last layer of a Neural Network, which means it requires less computational capacity than IGA. However, because its performance was generally worse than REx and IGA for the three OOD mechanics problems, IGA and REx appear to be better choices for the OOD problems

studied here.

Finally, in contrast to the performance of these OOD algorithms on classification problems in which they were shown to have similar accuracy on test data and training data (Arjovsky et al., 2019; Krueger et al., 2021; Koyama and Yamaguchi, 2020), the test error obtained by these OOD algorithms for the regression problems in mechanics is still significantly higher than their validation error. This is reasonable, because as compared to a classification problem with binary outputs of either 1 (belong to this class) or 0 (does not belong to this class) in both training and test datasets, the output of regression problems is usually continuous and can shift to very different value ranges than the values in the training dataset. This characteristic of regression problems makes capturing shifts in the data more challenging than in classification. Therefore, as mechanics problems typically require regression, and because of the many different types of OOD problems that are possible, this may require further development of OOD methods that can robustly handle multiple types of OOD shifts in order to solve OOD regression problems in mechanics.

3.5 Conclusion

In Chapter 2 and Chapter 3, a systematic investigation of OOD generalization problems in mechanics was conducted by identifying three new challenge problems with test data distribution shifts: covariate shift, mechanism shift, and sampling bias. To study these problems, two OOD benchmark datasets for each of the three challenge problems based on the Mechanical MNIST and Mechanical MNIST – EMNIST datasets were created. Following that, two types of ML models, a multilayer perceptron (MLP) and a convolutional neural network (modified LeNet), were independently trained on these datasets with four different risk minimization algorithms. One algorithm, the classical Empirical Risk Minimization (ERM) served as a baseline algorithm for comparison with three other popular methods specially designed for solving OOD generalization problems, Invariant Risk Minimization

(IRM), Risk Extrapolation (REx) and Inter Gradient Alignment (IGA). Through evaluating the performance of these algorithms, it is found that OOD generalization methods typically outperform ERM by achieving a lower predicting error on OOD test data while still maintaining good performance on training data. And, all algorithms tended to work better when paired with the more complex LeNet model than when paired with the simpler MLP. However, no algorithm had consistent top performance across all six OOD generalization problem datasets. And, while these OOD generalization algorithms have been reported in the literature to achieve near consistent performance on training and test environments for OOD classification problems (Arjovsky et al., 2019; Krueger et al., 2021; Koyama and Yamaguchi, 2020), for the OOD regression problems in mechanics covered in this study, there are only a few cases where these OOD generalization algorithms achieved comparable performance on both training and testing environments. In most cases, the error in the test environments was much higher than the training and validation errors. These results suggest that there is a need for methods to make these ML models more robust so that they can generalize invariance to multiple OOD scenarios.

It is worth noting that beyond the three kinds of OOD problems considered in this study (i.e., covariate shift, mechanism shift, and sampling bias), there are additional factors that can cause poor generalization of ML models for problems in mechanics. For example, OOD problems also exist if the scope of the training data does not cover the whole landscape of the data distribution, i.e. the scenario where new physics emerges due to the shifts of the landscape in the testing environments. Examples of this include different flow behavior for low and high Reynolds numbers (Smits et al., 2011), or linear vs. nonlinear mechanical response of soft tissues in the small and large strain regimes (Fung, 2013). In these situations, it is difficult for traditional ML methods to predict the new physics and handle the OOD problem simply by learning from the training data. Because the OOD methods explored in this study have only been tested on covariate shift, mechanism shift, and sampling bias,

OOD problems driven by the emergence of fundamentally different physical regimes are beyond the scope of this study, and represent a promising avenue for future work.

Overall, this study provides a critical first evaluation of OOD ML methods in mechanics. Based on this work, it can be anticipated that the benchmark regression datasets created for OOD problems in mechanics can accelerate the study of OOD generalization problems in the context of regression. And, this work of examining OOD generalization problems in mechanics is a critical step towards applying ML methods to practical problems in real world mechanics. Looking forward, defining OOD generalization methods that are specific to problems in mechanics will be an important direction for future research. Furthermore, it is very important to establish a standard way of selecting hyperparameters when developing new OOD generalization methods. Finally, note that in this study the environment label of each dataset is assumed known. Future investigation should consider alternative methods that have been proposed to divide the training data into different environments as a preprocess step (Creager et al., 2021; Liu et al., 2021b). Broadly speaking, the methods and results presented in this study are a starting point for future work in OOD generalization tasks in mechanics.

Chapter 4

Chiral Metamaterials

Shaat and Park (2023) introduced a chiral metamaterial that exhibits nonreciprocal and asymmetric elasticity due to changes in the internal contact mechanism when forces are applied from different directions. The fundamental unit of this chiral metamaterial consists of two rigid circles connected by an elastic ligament. The surface of the rigid circle is frictionless. In Shaat and Park (2023), this building block is characterized by a straight ligament that has a fixed contact angle with the rigid circles, as shown in Fig. 4·1. Note that the initial contact is stress-free so there is no contact pressure between the ligament and the circle. In this work, an analysis will be conducted to show that the response to force from different directions is significantly influenced by the contact between the ligament and the rigid circles. To harness the potential of the ligament-circle contact mechanism and facilitate programmability in the material's reciprocity and stiffness asymmetry, the chiral metamaterial design space is expanded by introducing additional design variations in ligament shape, contact angle, and circle radius. This Chapter lays out the details of this mechanical system.

4.1 Stiffness Definition

Fig. 4·2 illustrates a representative structure undergoing four different types of deformation: extension, compression, anti-clockwise rotation, and clockwise rotation. Finite Element Analysis (FEA) (Liu et al., 2021c; Szabó and Babuška, 2021; Yang et al., 2022; Yuan et al., 2020, 2019) using the commercial software ABAQUS (Smith, 2009) was conducted

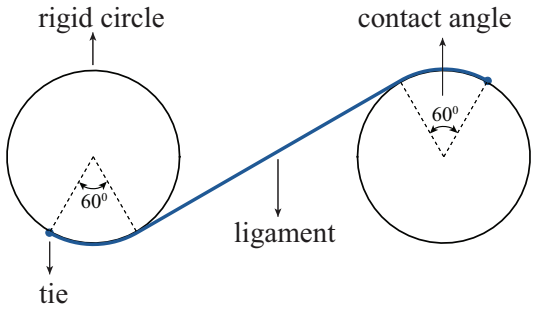


Figure 4·1: Illustration of the chiral metamaterial in Shaat and Park (2023). The ligament was tied to two rigid circles at the ends. The ligament shape was fixed and the contact angle was 60^0 on both sides.

to obtain the force-displacement response for each deformation (see Appendix A.3 for more details). For extension, a displacement u_x^+ to the positive x direction was applied to the right rigid circle. From reaction forces computed via FEA, we can get two stiffness values $k_{xx}^+ = F_x/u_x^+$, where F_x is the reaction force in the x direction when the displacement is in the x direction, and $k_{yx}^+ = F_y/u_x^+$, where F_y is the reaction force in the y direction when the displacement is in the x direction. For compression, a displacement u_x^- in the negative x direction was applied. Similarly, we can get the stiffness values $k_{xx}^- = F_x/u_x^-$ and $k_{yx}^- = F_y/u_x^-$. To distinguish the stiffness values obtained from the deformation of different directions, the symbol "+" is used when the displacement is in the positive direction, and the symbol "−" is used when the displacement is in the negative direction. For the anti-clockwise rotation, the displacement u_y^+ was applied and two stiffness values $k_{xy}^+ = F_x/u_y^+$ and $k_{yy}^+ = F_y/u_y^+$ are calculated. For the clockwise rotation, the displacement u_y^- was applied and the two stiffness values $k_{xy}^- = F_x/u_y^-$ and $k_{yy}^- = F_y/u_y^-$ are calculated. Considering all of these displacement directions, the eight stiffness values of the chiral unit are summarized using an array $K = [k_{xx}^-, k_{xy}^-, k_{yx}^-, k_{yy}^-, k_{xx}^+, k_{xy}^+, k_{yx}^+, k_{yy}^+]$. The material is described using the stiffness matrix:

$$K = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{bmatrix} \tag{4.1}$$

where the values of each element k_{ij} can be either k_{ij}^{-} or k_{ij}^{+}. For typical materials, the relationship $k_{ij}^{-} = k_{ij}^{+}$ and $k_{ij} = k_{ji}$ holds due to the Maxwell–Betti theorem (Viesca and Rice, 2011; Coulais et al., 2017). When the stiffness values are different in opposite directions of the same axis, i.e. $k_{ij}^{-} \neq k_{ij}^{+}$, the reciprocity of linearity breaks, and the material behaves like a bilinear spring (Lu and Norris, 2022). When the stiffness matrix is asymmetric, i.e. $k_{ij} \neq k_{ji}$, the reciprocity of the stiffness matrix is broken, similar to the odd elasticity described recently (Scheibner et al., 2020). Both scenarios lead to nonreciprocal effects in wave propagation. To distinguish between these cases, we refer to $k_{ij}^{-} \neq k_{ij}^{+}$ as "non-reciprocity" and $k_{ij} \neq k_{ji}$ as "asymmetry". In this work, the goal is to design a chiral metamaterial to break symmetry and reciprocity to the largest extent, e.g., $k_{ij}^{-} >> k_{ij}^{+}$ or $k_{ij} >> k_{ji}$.

4.2 Contact Mechanism

This Section investigates how loads applied from various directions can induce distinct contact modes between the rigid circle and the elastic ligament. An analogy can be drawn between these diverse contact modes and varying the boundary conditions of an elastic beam where adding and removing support conditions can dramatically change the mechanical state of the structure, resulting in asymmetrical properties under different loading modes. To further elucidate this concept, Fig. 4·3 presents an example of an identical structure subjected to forces from two opposing directions – compressive and extensional loading. In Fig. 4·3, ligament deformation was obtained through finite element simulation and magnified by a factor of 10 to aid in visualization.

Fig. 4·3(a-i) shows the deformed structure under compressive load. The left circle remains fixed while the right circle is subject to a displacement load u_x^{-} towards the left. Following the deformation, the left circle and the ligament are detached while the right circle and the ligament have a small area of contact. As the circle is frictionless, the ligament in the

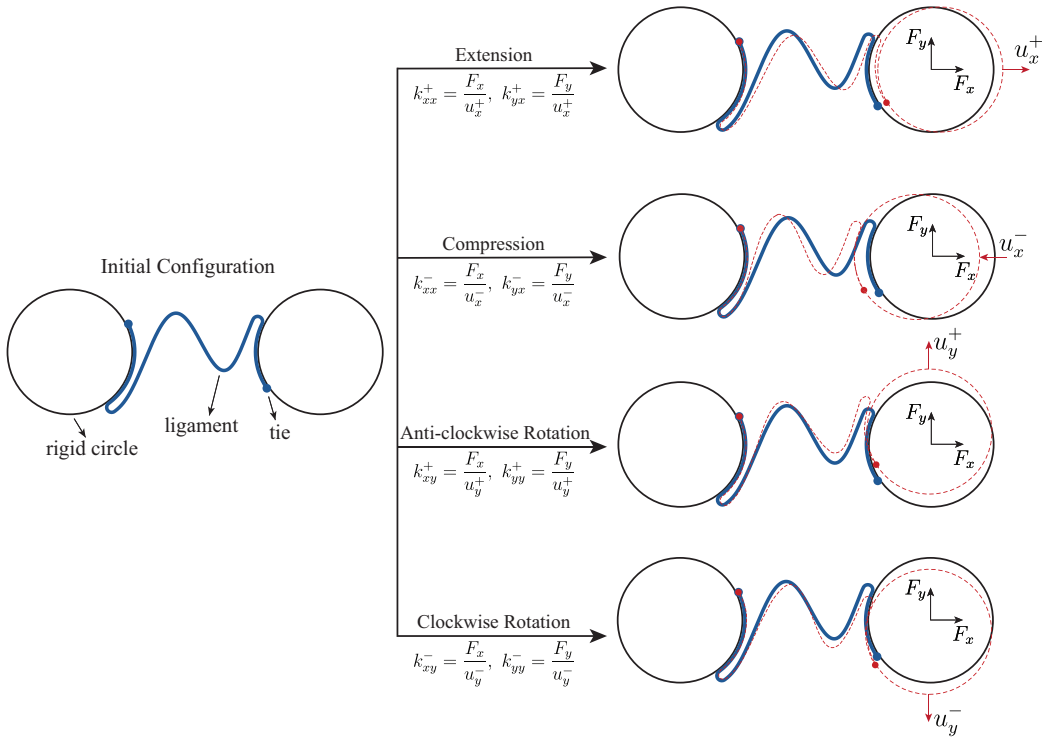


Figure 4·2: Illustration of the chiral metamaterial undergoing displacement from four directions. The reaction force of the right rigid circles in the x and y directions are denoted as F_x and F_y. The stiffness values are calculated using the formula $k_{ij} = F_i/u_j$ when u_j is nonzero. A superscript "+" is added to the displacement and stiffness symbol when the displacement is positive, and "−" when the displacement is negative.

contact area only experiences pressure from the perpendicular direction. Consequently, the mechanics of the structure can be equivalent to a beam fixed at both ends with a frictionless roller in the middle, as illustrated in Fig. 4·3(b-i). In this context, the bending moment around the roller support does not change since roller supports do not contribute to bending moments. Specifically, if a roller support is positioned at the midpoint of a beam, the bending moment will attain its maximum value and undergo an abrupt transition. To validate this, Fig. 4·3(c-i) shows the evolution of the bending moment and axial force along the beam from the left end to the right end. Notably, the bending moment around the roller support (point A) reaches a maximum, aligning well with the proposed model in Fig. 4·3(b-i) that

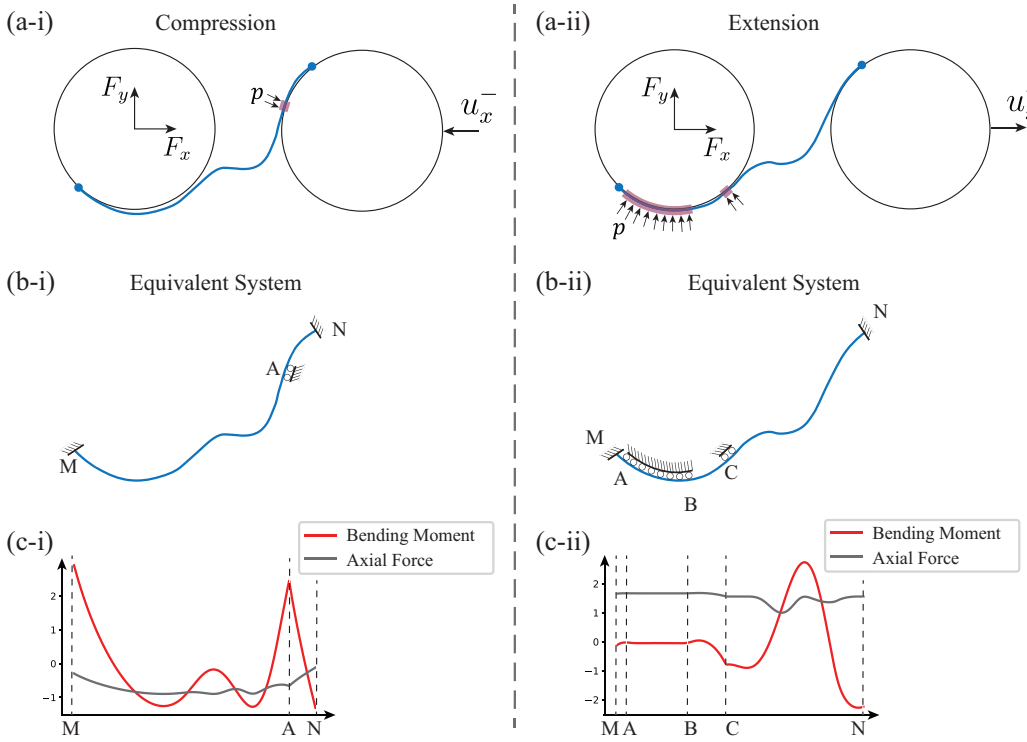


Figure 4·3: Mechanical model of the chiral structure under compression and extension. (a) Chiral structure under (i) compression load and (ii) extension load. The contact area of the deformed structure is highlighted with red color. (b) Equivalent mechanical model illustrating the contact mechanism with roller supporters substituting the contact area. (c) Distribution of Bending Moment and Axial Force along the beam.

describes the mechanism of the chiral structure.

Fig. 4·3(a-ii) shows the deformed structure under an extension load. The left circle is fixed and the right circle is subject to a displacement load u_x^+ towards the right. Following the deformation, the right circle and the ligament are detached while the left circle and the ligament establish two contact areas. Similarly, the Fig. 4·3(b-ii) presents an equivalent model of Fig. 4·3(a-ii). The two contact areas, one from point A to point B, and another around point C, are also equivalent to roller supports. Fig. 4·3(c-ii) shows the evolution of the bending moment and axial force along the beam for the extension load. The bending moment remains constant in the area of rollers.

While the ligament is linearly elastic, variations in boundary conditions occur under loads from different directions, leading to varying stiffnesses in different directions. The goal of this work is to maximize these discrepancies to achieve extreme mechanical behaviors. The details of these objectives are elaborated in the following sections.

4.3 Strain Energy Components

The previous Section 4.2 provides a qualitative analysis of the contact modes of the chiral structures. Specifically, the non-reciprocity and asymmetry of chiral metamaterials are produced by the variation in contact modes leading to different boundary conditions of the equivalent beam models. In this Section, a quantitative examination of the stiffness disparities will be conducted. Specifically, the mechanics of the deformation under different loading directions will be analyzed to understand how stiffness is either enhanced or weakened depending on the direction of applied loads.

Fig. 4·4(a) shows a representative design where the values of k_{xx}^{-} and k_{xx}^{+} are different. Regardless of the directions, when the only nonzero displacement is u_x, the stiffness k_{xx} of the structure is determined by evaluating the total change in strain energy by the equation below

$$E_{strain} = \frac{1}{2} k_{xx} u_x^2 \tag{4.2}$$

Following the finite element analysis of this structure, the strain energy is 0.007764 after compression and 0.1064 after extension, with the u_x set constant as -0.08 for compression and $+0.08$ for extension. Through eqn. 4.2, the stiffnesses $k_{xx}^{-} = 2.42$ for compression and $k_{xx}^{+} = 33.25$ for extension can be derived. It is evident that the strain energy during extension significantly surpasses that of compression, resulting in markedly higher stiffness during extension. Considering that bending and stretching energies are the primary components of strain energy for a linear elastic beam, the stiffness disparities can be investigated by

analyzing the stretching and bending energy distribution along the elastic ligament in Fig. 4·4(b)(c). During compression, the stretching energy is negligible compared to the bending energy, whereas during extension, the stretching energy dominates over the bending energy. This disparity indicates that ligament deformation is predominantly bending during compression, whereas stretching dominates during extension. Given the inherent difficulty in stretching a beam compared to bending it, the stretching energy is generally much larger than the bending energy, leading to substantially greater resistance to external forces during extension. As a result, the stiffness value in extension is much larger than that of compression for this structure. By manipulating the geometry of the chiral structures, the stretching and bending behavior under different loads can be controlled, allowing the discovery of optimal chiral structure designs capable of exhibiting extreme non-reciprocity and asymmetry.

4.4 Optimization Objectives

As briefly introduced in Section 4.1, this work aims to enhance the nonreciprocity and asymmetry of the chiral metamaterial. This section provides additional details on the formalization of these objectives. Furthermore, the scope of the investigation will be extended to multi-objective optimization, allowing for the identification of optimal designs that simultaneously exhibit both nonreciprocity and elastic asymmetry.

4.4.1 Non-Reciprocity

Following the definition in Section 4.1, enhancing the non-reciprocity of metamaterials equals maximizing the variance of stiffness values along opposite directions, i.e. $k_{ij}^{-} \neq k_{ij}^{+}$. To achieve this, this study aims to maximize the ratio between the absolute values of the stiffness. The optimization task includes eight objectives to be maximized, outlined below:

① $f_1 = |\frac{k_{xx}^{-}}{k_{xx}^{+}}|$, ② $f_2 = |\frac{k_{xx}^{+}}{k_{xx}^{-}}|$, ③ $f_3 = |\frac{k_{xy}^{-}}{k_{xy}^{+}}|$, ④ $f_4 = |\frac{k_{xy}^{+}}{k_{xy}^{-}}|$,

⑤ $f_5 = |\frac{k_{yx}^{-}}{k_{yx}^{+}}|$, ⑥ $f_6 = |\frac{k_{yx}^{+}}{k_{yx}^{-}}|$, ⑦ $f_7 = |\frac{k_{yy}^{-}}{k_{yy}^{+}}|$, ⑧ $f_8 = |\frac{k_{yy}^{+}}{k_{yy}^{-}}|$.

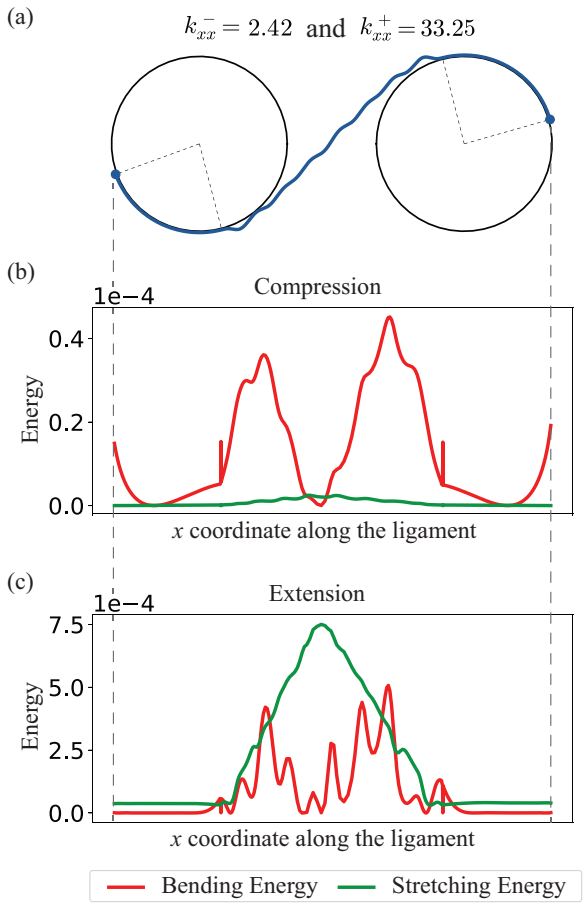


Figure 4·4: Bending and Stretching Energy distribution along the elastic ligament. (a) The schematic of the chiral structure. The design has a smaller stiffness k_{xx}^{-} under compression loads, and a larger stiffness k_{xx}^{+} under extension loads. (b) The bending and stretching distribution along the ligament under compression load. (c) The bending and stretching distribution along the ligament under extension load. The three figures (a)(b)(c) share the same x coordinates.

Note that when the value of these objectives is equal to 1, i.e. the stiffness in opposite directions is the same, the corresponding structure exhibits reciprocity. Additionally, to maximize the difference between each pair of stiffness k_{ij}^{-} and k_{ij}^{+}, we can either design a structure that k_{ij}^{-} is larger k_{ij}^{+}, or k_{ij}^{+} is larger k_{ij}^{-}. For instance, to maximize the disparity between k_{xx}^{-} and k_{xx}^{+}, objective f_1 aims to identify an optimal design where k_{xx}^{-} significantly exceeds k_{xx}^{+}, whereas objective f_2 seeks an optimal design where k_{xx}^{+} significantly exceeds k_{xx}^{-}.

Consequently, when searching for optimal designs that maximize non-reciprocity objective f_1 to f_8, only designs with objective values greater than 1 are considered. In summary, this goal can be formalized as below:

$$\max f_i, \text{ where } f_i > 1 \text{ and } i \in \{1, 2, \ldots, 8\} \tag{4.3}$$

4.4.2 Elastic Asymmetry

Following the definition in Section 4.1, the asymmetry of chiral metamaterial is characterized by an asymmetric stiffness matrix, i.e. $k_{ij} \neq k_{ji}$. Similarly to the approach outlined in Section 4.4.1, the goal is to maximize the ratio between the stiffness values. The eight objectives for asymmetry optimization are summarized below:

① $g_1 = |\frac{k^-_{xy}}{k^-_{yx}}|$, ② $g_2 = |\frac{k^-_{xy}}{k^+_{yx}}|$, ③ $g_3 = |\frac{k^+_{xy}}{k^-_{yx}}|$, ④ $g_4 = |\frac{k^+_{xy}}{k^+_{yx}}|$,
⑤ $g_5 = |\frac{k^-_{yx}}{k^-_{xy}}|$, ⑦ $g_6 = |\frac{k^+_{yx}}{k^-_{xy}}|$, ⑥ $g_7 = |\frac{k^-_{yx}}{k^+_{xy}}|$, ⑧ $g_8 = |\frac{k^+_{yx}}{k^+_{xy}}|$.

Similar to eqn. 4.3, a more general form of these objectives is as below:

$$\max g_i, \text{ where } g_i > 1 \text{ and } i \in \{1, 2, \ldots, 8\} \tag{4.4}$$

4.4.3 Multi-Objective

In Sections 4.4.1 and 4.4.2, a single objective was employed to assess the performance of the chiral structures. However, optimizing a single objective guarantees design optimality in only one dimension. For multiple competing objectives, maximizing a single dimension may compromise the performance concerning another objective. Thus, discovering a chiral metamaterial that exhibits multiple novel functionalities involves striving to optimize the material to display both non-reciprocity and elastic asymmetry.

The problem of searching for optimal designs that excel in multi-objectives is broadly known as determining the Pareto front (Schulz et al., 2018a; Riquelme et al., 2015; Li et al., 2024). The design solutions to the multi-objective optimization are referred to as

Pareto optimal. Specifically, a Pareto optimal design is one for which any adjustment cannot improve all the objectives simultaneously. In other words, for any Pareto optimal design, there is no alternative design that surpasses it in all performance functions. The set containing all the Pareto optimal designs is the Pareto set. The corresponding performance of the Pareto set in a performance space is the Pareto front. To facilitate understanding these concepts in the context of chiral metamaterials optimization, Fig. 4·5 illustrates a schematic performance map of design solutions. Each data point on the map is randomly generated and depicts performances f_i and g_i of a design solution, where f_i, representing an objective of non-reciprocity in Section 4.4.1, and asymmetry g_i, representing an objective of asymmetry in Section 4.4.2. In the problem setting of this study, the objective values of the optimal designs must be larger than 1. Consequently, the designs not satisfying this constraint will not be considered and are denoted as 'Archive'. The design solutions colored red are the Pareto optimal, as no other feasible solutions can achieve larger values for both f_i and g_i. All other feasible solutions are colored grey. The goal is to identify the Pareto optimal designs and the corresponding Pareto front to enhance both non-reciprocity and asymmetry. This goal is expressed mathematically as below:

$$\max\{f_i, g_j\}, \text{ where } f_i, g_j > 1, \; i, j \in \{1, 2, \ldots, 8\} \tag{4.5}$$

4.5 Design Spaces

As illustrated in Fig. 4·6(a), the programmable chiral structure consists of two rigid circles with the same radius R. The distance between the two circles is L, which has a constant value 20. An elastic ligament initiates its connection with the left circle at an angle θ_0 and forms a contact angle θ_1. Similarly, the ligament establishes a connection with the right circle from an angle θ_{00} and forms a contact angle θ_{11}. Positive values for the contact angle indicate an anti-clockwise orientation from the connecting point and vice versa for negative

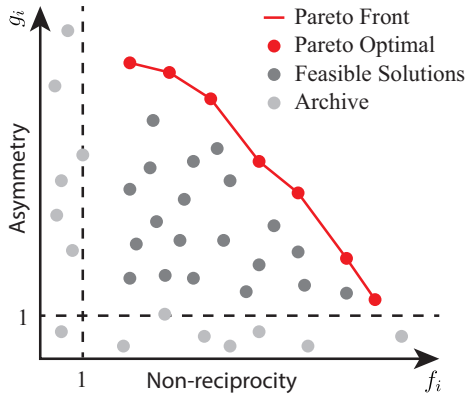


Figure 4·5: Illustration of Pareto front for the multi-objective optimization of chiral metamaterial.

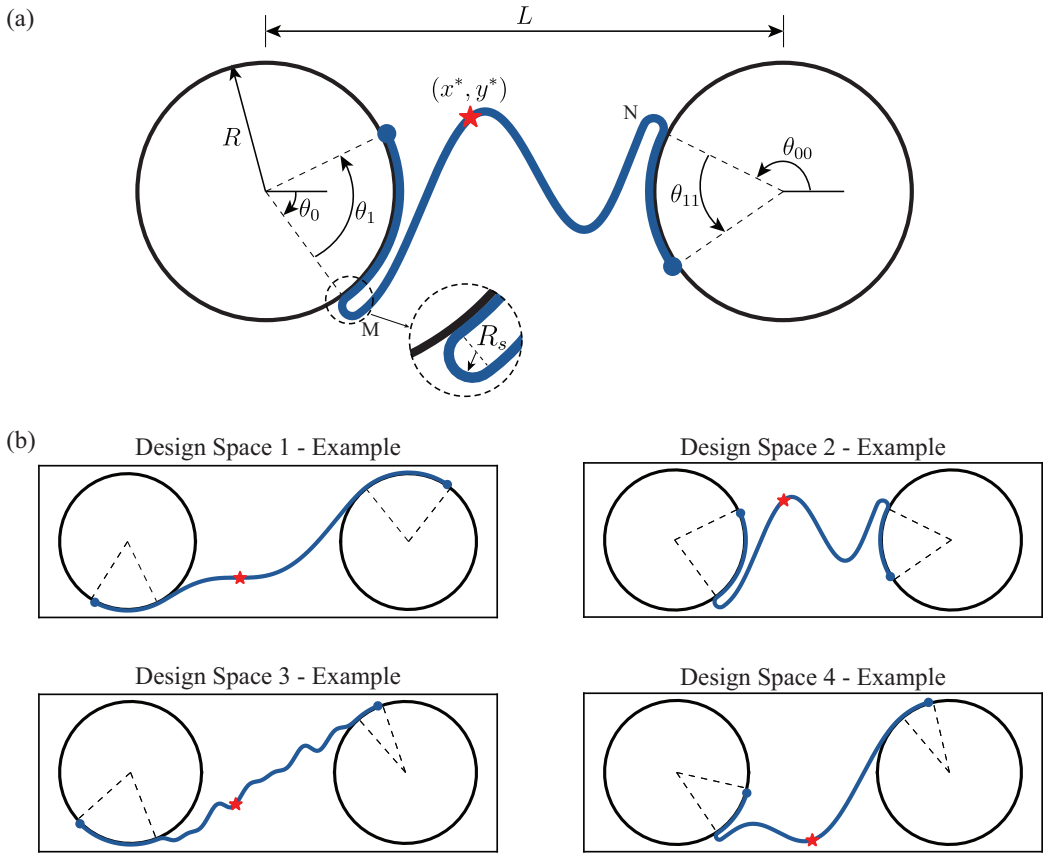


Figure 4·6: Illustration of (a) the design parameters and (b) the design spaces of the chiral metamaterial.

values. For the part of the ligament that is not in contact with the circle, there is a half circle with radius R_s at the transition point for both sides. The ending point for the small half circle

on the ligament is denoted as points M and N. The middle part from M to N is continuous and each point is sampled from the following function:

$$h(x) = \sum_{i=0}^{n-1} a_i h_i(x) = a_0 h_0(x) + a_1 h_1(x) + \cdots + a_{n-1} h_{n-1}(x) \tag{4.6}$$

where x is the Cartesian coordinates with the midpoint of the two circles as the origin of the coordinates, and $h_0(x), h_1(x), \ldots, h_{n-1}(x)$ are continuous and pre-selected functions. To ensure the smoothness of the ligament, we require the the zeroth, first, and second derivatives of the $h(x)$ to be continuous at the two ends M and N. The coordinates of M are represented as (x_M, y_M) and N as (x_N, y_N). Consequently, there are six known constraints on $h(x)$:

$$\begin{cases} h(x_M) = y_M \\ h^{'}(x_M) = y_M^{'} \\ h^{''}(x_M) = y_M^{''} \\ h(x_N) = y_N \\ h^{'}(x_N) = y_N^{'} \\ h^{''}(x_N) = y_N^{''} \end{cases} \tag{4.7}$$

where $y_M, y_M^{'}$, and $y_M^{''}$ denote the zeroth, first, and second derivatives at point M, and $y_N, y_N^{'}$, and $y_N^{''}$ denote the zeroth, first, and second derivatives at point N. Given that the midpoint of the two circles is the origin, the center of the left circle is at $(-L/2, 0)$, and the center of the right circle is at $(L/2, 0)$. Thus, we have:

$$\begin{cases} x_M = -L/2 + (R+2R_s)\cdot\cos(\theta_0) \\ y_M = (R+2R_s)\cdot\sin(\theta_0) \\ y_M^{'} = -1/\tan(\theta_0) \\ y_M^{''} = -R^2/y_M^3 \\ x_N = L/2 + (R+2R_s)\cdot\cos(\theta_{00}) \\ y_N = (R+2R_s)\cdot\sin(\theta_{00}) \\ y_N^{'} = -1/\tan(\theta_{00}) \\ y_N^{''} = -R^2/y_N^3 \end{cases} \tag{4.8}$$

Consider that the number of coefficients to be determined for $h(x)$ is n and there are only 6 boundary conditions in eqn. 4.7. By solving eqn. 4.6 with eqn. 4.7, $h(x)$ has a unique solution when n is equal to 6 and multiple solutions when n exceeds 6. In this work, the parameter n is set to 7 to add one more degree of freedom to the shape of the ligament by fixing a randomly selected point (x^*, y^*) on the ligament, denoted as a red star point in Fig. 4·6(a). Then one more constraint is introduced:

$$h(x^*) = y^* \tag{4.9}$$

The coefficients $a_0, a_1, \ldots, a_6$ of $h(x)$ can be identified by eqn. 4.7 and eqn. 4.9.

In summary, there are 8 design parameters $X = [R, \theta_0, \theta_1, \theta_{00}, \theta_{11}, R_s, x^*, y^*]$. Notably, R_s can take the value of zero, and the subfunction components $h_0(x), h_1(x), \ldots, h_6(x)$ of the ligament can be either polynomial or trigonometric. To search for optimal designs for the objectives defined in Section 4.4, exploration is conducted across four different design spaces, as illustrated in Fig. 4·6(b). The range of the design parameters $[R, \theta_0, \theta_1, \theta_{00}, \theta_{11}]$ of each design spaces are presented in Table 4.1. The details of the value of R_s and the sampling function $h(x)$ for the ligament in each design space are summarized below:

1. Design Space 1
    - $R_s = 0$ for both left and right side
    - $h(x) = \sum_{i=0}^{n-1} a_i x^i$
2. Design Space 2
    - $0.1 \leq R_s \leq R/10$ for both left and right side
    - $h(x) = \sum_{i=0}^{n-1} a_i x^i$
3. Design Space 3
    - $R_s = 0$ for both left and right side
    - $h(x) = a_0 + a_1 x + a_2 sin(x) + a_3 sin(x) + a_4 x^4 + a_5 x^5 + a_6 x^6$.
4. Design Space 4
    - $R_s = 0$ for right side and $0.1 \leq R_s \leq R/10$ for left side
    - $h(x) = \sum_{i=0}^{n-1} a_i x^i$

As discussed in Section 4.2, diverse contact mechanisms can induce varying non-reciprocity and asymmetry properties. The choice of the four design spaces shown in Fig. 4·6(b) aims to encompass a broad range of contact mechanisms. In design space 1, the ligament and the right circle are prone to establish contact when it is subjected to extension. In design space 2, the ligament and the right circle tend to make contact under compression. Design space 3 introduces alterations to the subfunctions of the ligament shape compared to design space 1, allowing the investigation of the impact of the ligament shape on the properties of the chiral metamaterial. In design space 4, the left circle and the ligament tend to contact under compression, while the right circle and the ligament tend to contact under extension loads.

Table 4.1: Design parameter ranges within each design space.

	R	θ_0	θ_1	θ_{00}	θ_{11}
Design Space 1	[3,7]	[-90, -20]	[-90, -20]	[90, 160]	[-90, -20]
Design Space 2	[3,7]	[-90, 0]	[20, 90]	[90,180]	[20,90]
Design Space 3	[3,7]	[-90, -20]	[-90, -20]	[90, 160]	[-90, -20]
Design Space 4	[3,7]	[-90, 0]	[20, 90]	[90,180]	[-90,-20]

Chapter 5

Non-Reciprocal and Asymmetric Elastic Chiral Metamaterials Optimization

Machine learning (ML) methods have been extensively applied to complex engineering challenges, demonstrating their efficiency in uncovering intricate relationships within data (Lee et al., 2023; Ha et al., 2023; Buehler, 2023b; Yuan et al., 2022; Arzani et al., 2023; Nguyen and Lejeune, 2024). Through sufficient training on diverse datasets, an effectively trained ML model exhibits rapid and accurate predictions on unseen data, drastically reducing the time required to characterize material properties by orders of magnitude compared to conventional experimental or simulation methods. This capacity enables it to efficiently explore a large pool of candidates in the search for optimal designs. In the context of chiral metamaterial design, the design space is practically infinite because the design parameters governing the initial contact angle and the shape of the ligament shape are continuous. Consequently, ML methods are employed to guide the exploration of nonreciprocal and asymmetric stiffness chiral metamaterials. The following parts of this section elaborate on the approach to data collection, the details of the ML model, and the data-driven optimization methods utilized to discover optimal designs.

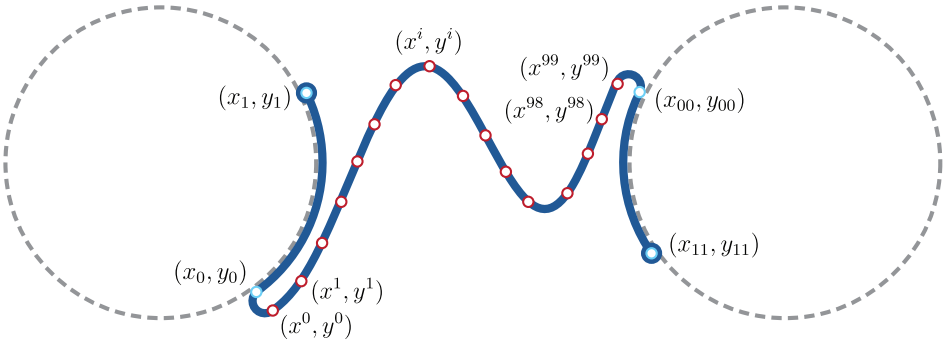


Figure 5·1: Illustration of the feature representation of a chiral metamaterial

5.1 Data Collection

5.1.1 Data Representation

In Section 4.5, each design space was outlined using eight design parameters, denoted as $X = [R, \theta_0, \theta_1, \theta_{00}, \theta_{11}, R_s, x^*, y^*]$. However, due to the diversity in the subfunctions of $h(x)$ that define the ligament shape, these parameters cannot fully characterize the geometry of the chiral structure across all design spaces. Additionally, given the efficacy of machine learning methods in pattern recognition (Bishop, 2006), an alternative way to better represent the chiral structure is by its shape and geometric features. As Fig. 5·1 shows, the points where the ligament and the rigid circles initiate connection are noted as (x_0, y_0) and (x_{00}, y_{00}), and coordinates of the two ends of the ligament as (x_1, y_1) and (x_{11}, y_{11}). Furthermore, 100 points along the x direction are uniformly sampled from the elastic ligament. Therefore, the coordinates of 104 points are employed to uniquely describe the geometry of each chiral structure. The 208 features of a single chiral structure are summarized as feature vector X as follows:

$$X = [x^0, x^1, \ldots, x^{99}, y^0, y^1, \ldots, x^{99}, x_1, y_1, x_0, y_0, x_{00}, y_{00}, x_{11}, y_{11}] \tag{5.1}$$

The second step is to label each chiral structure with its mechanical properties by running a finite element simulation. As the goal of this study is to maximize the nonreciprocity and

asymmetry defined in Section 4.4, the label of each material will be the objectives $f_{1:8}$ and $g_{1:8}$ defined in Section 4.4.1 and Section 4.4.2. To get this, it is necessary to first calculate the stiffness values for each chiral structure. Fig. 5·2 provides an example of how the stiffness values of each structure are obtained. Each row shows the contact region of the structure under the loads from four different directions and the corresponding force-displacement response. The stiffness, i.e. the slope of the response curve, is obtained by fitting the finite element simulation results using least squares.

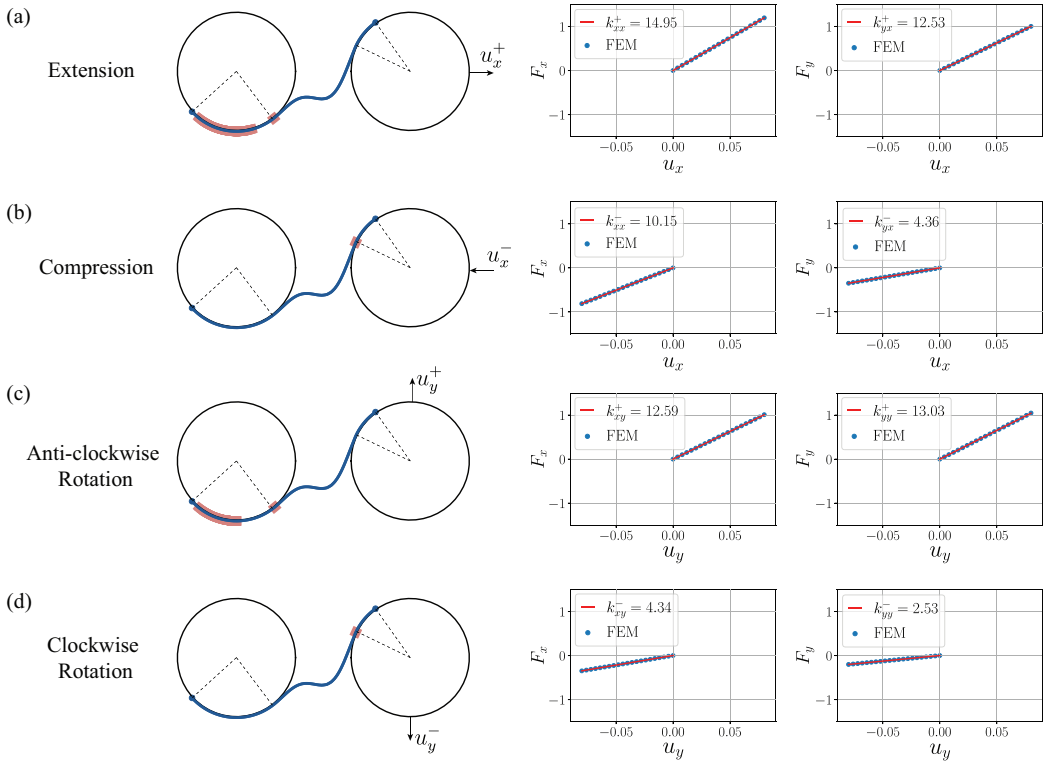


Figure 5·2: Illustration of a representative chiral structure subjected to loads from four different directions and the corresponding force-displacement response obtained from finite element (FEM) simulation. The contact area is highlighted in pink. Stiffness is determined as the coefficient of linear fitting applied to the FEM data using the least squares method.

In Fig. 5·2(a)(b), due to the different contact regimes in compression and extension, the stiffness values $k_{xx}^+ = 14.95$ and $k_{yx}^+ = 12.53$ are obtained for extension, and $k_{xx}^- = 10.15$ and $k_{yx}^- = 4.36$ are obtained for compression. In Fig. 5·2(c) where the structure is subject to an

anti-clockwise rotation load, the stiffness values $k_{yy}^{+} = 13.03$ and $k_{xy}^{+} = 12.59$ are obtained. Notably, the contact regime under the anti-clockwise rotation load in Fig. 5·2(c) closely resembles that of extension in Fig. 5·2(a), which results in the values of $k_{xy}^{+} = 12.59$ and $k_{yx}^{+} = 12.53$ being nearly equal. In Fig. 5·2(d) where the structure is subject to a clockwise rotation load, the stiffness values $k_{yy}^{-} = 2.53$ and $k_{xy}^{-} = 4.34$ are obtained. Similarly, the contact mode of clockwise rotation in Fig. 5·2(d) closely resembles the compression mode in Fig. 5·2(b), and $k_{xy}^{-} = 4.34$ is nearly equal to $k_{yx}^{-} = 4.36$. These observations align with the discussion of the contact mechanism in Section 4.2, indicating that the contact modes play a pivotal role in governing the non-reciprocity and asymmetry properties of the chiral structures. Subsequently, we can identify the non-reciprocity and asymmetry objectives $f_{1:8}$ and $g_{1:8}$ defined in Section 4.4.1 and Section 4.4.2 for this design in Fig. 5·2. The values are listed below:

$$
\begin{aligned}
&f_1 = |\tfrac{k_{xx}^{-}}{k_{xx}^{+}}| = |10.15/14.95| = 0.67, \quad f_2 = |\tfrac{k_{xx}^{+}}{k_{xx}^{-}}| = |14.95/10.15| = 1.47,\\
&f_3 = |\tfrac{k_{xy}^{-}}{k_{xy}^{+}}| = |4.34/12.59| = 0.34, \quad f_4 = |\tfrac{k_{xy}^{+}}{k_{xy}^{-}}| = |12.59/4.34| = 2.90,\\
&f_5 = |\tfrac{k_{yx}^{-}}{k_{yx}^{+}}| = |4.36/12.53| = 0.34, \quad f_6 = |\tfrac{k_{yx}^{+}}{k_{yx}^{-}}| = |12.53/4.36| = 2.87,\\
&f_7 = |\tfrac{k_{yy}^{-}}{k_{yy}^{+}}| = |2.53/13.03| = 0.19, \quad f_8 = |\tfrac{k_{yy}^{+}}{k_{yy}^{-}}| = |13.03/2.53| = 5.15,\\
&g_1 = |\tfrac{k_{xy}^{-}}{k_{yx}^{-}}| = |4.34/4.36| = 0.99, \quad g_2 = |\tfrac{k_{xy}^{-}}{k_{yx}^{+}}| = |4.34/12.53| = 0.34,\\
&g_3 = |\tfrac{k_{xy}^{+}}{k_{yx}^{-}}| = |12.59/4.36| = 2.88, \quad g_4 = |\tfrac{k_{xy}^{+}}{k_{yx}^{+}}| = |12.59/12.53| = 1.00,\\
&g_5 = |\tfrac{k_{yx}^{-}}{k_{xy}^{-}}| = |4.36/4.34| = 1.00, \quad g_6 = |\tfrac{k_{yx}^{+}}{k_{xy}^{-}}| = |12.53/4.34| = 2.88,\\
&g_7 = |\tfrac{k_{yx}^{-}}{k_{xy}^{+}}| = |4.36/12.59| = 0.34, \quad g_8 = |\tfrac{k_{yx}^{+}}{k_{xy}^{+}}| = |12.53/12.59| = 0.99.
\end{aligned}
$$

5.1.2 Data Augmentation

Data augmentation is a technique used to increase the diversity and the size of the training dataset. New training samples are generated by applying transformations or modifications to existing data, such as flipping, rotating, and scaling (Shorten and Khoshgoftaar, 2019; Rebuffi et al., 2021). Given the computational expense of acquiring mechanical properties via finite element simulation, data augmentation becomes essential for chiral metamaterial

design. Across all design spaces, the stiffness properties of a chiral material remain invariant under a 180-degree rotation, as Fig. 5·3 shows. Leveraging this property, the rotation transformation is applied to each of the chiral metamaterial data points and thus the size of the dataset is doubled.

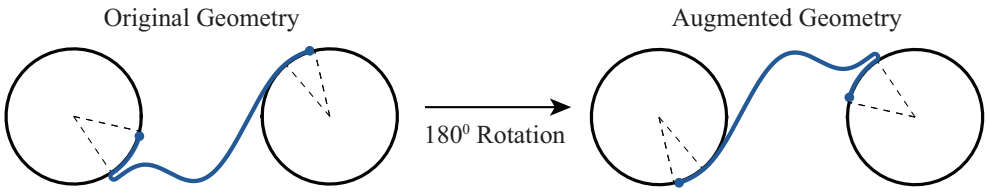


Figure 5·3: Illustration of the rotation transformation for data augmentation. The augmented geometry is obtained by rotating the original geometry by 180 degrees. The original and the augmented structure have the same properties in terms of their stiffness values in all directions.

5.2 Bayesian Optimization

During the search for optimal chiral structures, obtaining properties such as non-reciprocity and asymmetry by FEM is a challenging and computationally expensive task. Moreover, the physics of the relationship between the design and the properties of the chiral structure is nonlinear, which makes it challenging to obtain analytic solutions for the optimization objective. To address these challenges, we can leverage a data-driven approach - Bayesian Optimization (Polyzos et al., 2023). This method is a model-based optimization technique that excels in efficiently determining optimal designs for objective functions that are black-box and costly to evaluate. In the framework of material design, the goal of Bayesian Optimization is to address the optimization problem presented below:

$$x^* = arg\max_{x \in \mathcal{X}} f(x) \tag{5.2}$$

where $\mathcal{X}$ compromise all feasible designs, and $f(x)$ represents the objective function to be maximized. For each design point x that belongs to $\mathcal{X}$, the goal is to identify the optimal

point x^* at which the objective $f(x)$ attains its maximum value. To achieve this, we must select a probabilistic surrogate model $f(x)$ and an acquisition function $\alpha(x)$. The surrogate model $f(x)$ evaluates the objective value of each design point, and the acquisition function $\alpha(x)$ determines which point to query next. The query process involves obtaining the ground truth property for the selected point through experiments or simulation tools.

To conduct Bayesian Optimization, we need to start with an observed data collection $D_t = \{(x_i, y_i)\}_{i=1}^{t}$, where y_i denotes the measured objective value corresponding to point x_i. The first step is to fit the surrogate model with the dataset D_t. The second step is selecting the next point x_{t+1} that maximizes the acquisition function, querying the value of the y_{t+1}, and adding (x_{t+1}, y_{t+1}) to the observed dataset. The final solution to eqn. 5.2 is obtained by repeating these two steps until reaching a maximum iteration limit N. Below is a pseudo-code outlining the process of the Optimization Routine using Bayesian Optimization:

Algorithm 1 Optimization Routine

Require: initial observations D_0, maximum iterations N, surrogate model $f(x)$, acquisition function $\alpha(x)$

for $t = 0, 1, 2, \ldots, N-1$ **do**
    fit the model $f(x)$ with D_t
    calculate the acquisition function $\alpha(x)$ with $f(x)$
    select the next point $x_{t+1} \leftarrow arg\max_x \alpha(x)$
    perform experiment or simulation to evaluate the objective values y_{t+1} at point x_{t+1}
    $D_{t+1} \leftarrow \{D_t, (x_{t+1}, y_{t+1})\}$
end for
return best y and the corresponding x in D_N

When employing Bayesian Optimization for chiral metamaterial design, the process initiates with simulations for 5 randomly selected designs within each of the four design spaces, establishing an initial dataset comprising a total of 20 points. Following this, a pool of $400,000$ design candidates is generated, with $100,000$ in each design space, without conducting FEM simulation. Subsequently, Bayesian Optimization is carried out for 10 iterations. In each iteration, the top 10 points in the design pool with the highest values

determined by the acquisition function are selected for running FEM simulations using Abaqus. This approach of acquiring ten points rather than one in each iteration is adopted due to the simulation setup, where multiple simulation jobs are submitted as a batch in the computing center. Therefore, querying 10 points per iteration takes the same amount of time as querying a single point. Moreover, expanding the dataset faster enables us to get a more accurate surrogate model, expediting the discovery of optimal designs.

5.2.1 Surrogate Model

The Gaussian Process (GP) model (Schulz et al., 2018b; Deringer et al., 2021; Wang, 2023) is commonly favored as a surrogate model choice in Bayesian Optimization due to its capability to estimate prediction uncertainty directly. However, its efficacy diminishes notably in high-dimensional problems (Shahriari et al., 2015; Binois and Wycoff, 2022). Additionally, the intuition behind GP begins with kernel-encoded prior assumptions about the dataset distribution, then obtains a posterior distribution function given the observed data (Marrel and Iooss, 2024; Schulz et al., 2018b). However, in the case of intricate chiral material systems, where prior dataset knowledge is absent and the feature dimension is high, extra challenges arise when selecting appropriate kernels for GP regression models. In contrast, deep learning models like Multilayer Perceptron (MLP) are efficient at capturing black-box, unknown nonlinear relationships within high-dimensional data. Nonetheless, MLP models do not inherently provide uncertainty estimates for predictions. To address this, ensemble learning methods (Ganaie et al., 2022; Mienye and Sun, 2022; Mohammadzadeh et al., 2023) are employed to train a single MLP model multiple times with different seeds on observed data and utilize the variance across model predictions as an indication of uncertainty.

The MLP model is designed with four layers comprising 1024, 1024, 64, and 1 neurons, respectively. Following each hidden layer, Rectified Linear Unit (ReLU) activation functions were applied to introduce non-linearity into the model. The training was performed using a

batch size of 16 over 250 epochs, with a learning rate set to 0.001. The MLP model is trained for $k = 15$ times, each utilizing a different seed for training-validation dataset splitting and weight initialization. The prediction of objective values and the uncertain estimation are calculated below:

$$\begin{cases} \mu(x) = \frac{1}{k}\sum_1^k \hat{y}_i \\ \sigma(x) = \sqrt{\frac{1}{k}\sum_1^k (\hat{y}_i - \mu(x))^2} \end{cases} \tag{5.3}$$

where $\mu(x)$ is the averaged prediction across all MLP models, $\sigma(\hat{y})$ is the standard deviation of the predictions, and $\hat{y}_i$ is the predicted value from the ith model.

5.2.2 Acquisition Function

In the framework of Bayesian Optimization, various acquisition functions are available for selection, such as the probability of improvement (PI) (Kushner, 1964; Ruan et al., 2020), entropy search (ES) (Hennig and Schuler, 2012; Wang and Jegelka, 2017), expected improvement (EI) (Zhan and Xing, 2020; Qin et al., 2017), and upper confidence bound (UCB) (Carpentier et al., 2011; Simchi-Levi and Wang, 2023). This study focuses on employing two specific methods: EI and UCB.

Expected Improvement

The Expected Improvement (EI) method is a versatile choice for many optimization scenarios. It is efficiently designed to trade-off between the global search (exploration) and local minimization (exploitation) (Brochu et al., 2010). The acquisition function of EI is:

$$\alpha^{\mathrm{EI}}(x) = \mathrm{E}[\max(0, f(x) - f_{\max})] \tag{5.4}$$

where $f_{\max}$ is the best performance observed so far. The acquisition function $\alpha^{\mathrm{EI}}(x)$ calculates the expectation of the improvement for each unseen point. In essence, the equation evaluates each point by how much performance it can enhance compared to the current best

point. If the potential improvement surpasses the best point, the expected improvement is the amount of improvement. If the potential improvement cannot surpass the best point, the expected improvement is zero. Assuming the target y follows a Gaussian distribution $\mathcal{N}(\mu(x),\sigma(x))$, the formulation of EI can be written explicitly as (Jones et al., 1998; Brochu et al., 2010)

$$\alpha^{\mathrm{EI}}(x) = (\mu(x) - f_{\max} - \xi)\Phi(\lambda) + \sigma(x)\phi(\lambda) \tag{5.5}$$

where $\lambda = (\mu(x) - f_{\max} - \xi)/\sigma(x)$. $\Phi(\lambda)$ and $\phi(\lambda)$ are the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution respectively. The first term in eqn. 5.5 emphasizes exploitation by favoring points with higher expected objective values, while the second term promotes exploration by favoring points with greater uncertainty. Thus, eqn. 5.5 strikes a balance between exploitation and exploration controlled by the hyperparameter ξ, with higher ξ leading to more exploration. In this study, the value ξ is set to 0.002 to achieve a balanced exploration-exploitation strategy.

Upper Confidence Bound

The Upper Confidence Bound (UCB) calculated the upper bound for the prediction by adding the uncertainty to the estimation, offering a straightforward approach to balancing exploration and exploitation. The acquisition function for UCB is defined as (Shahriari et al., 2015; Cox and John, 1992):

$$\alpha^{\mathrm{UCB}}(x) = \mu(x) + \beta\sigma(x) \tag{5.6}$$

Similar to EI, the first component of the equation favors exploitation, and the second component for exploration. The hyperparameter β serves to balance the weights between exploitation and exploration, with a higher value leading to more exploration.

In this study, the method EI is applied to single objective optimization considering that the algorithm has already proven efficient in material discovery in many other studies (Gongora

et al., 2020; Zhang et al., 2020b; Kotthoff et al., 2021). Although the EI acquisition function form of eqn. 5.5 is typically associated with the Gaussian Process Model, the acquisition function eqn. 5.5 is adopted in single objective optimization with the assumption that the observed data are sampled from a Gaussian Distribution with mean and variance derived from the predictions of ensembled MLPs, i.e. $y \sim \mathcal{N}(\mu, \sigma)$, as formalized in eqn. 5.3. This assumption is grounded in the evidence that MLPs (or Neural Networks) are well-calibrated models (Niculescu-Mizil and Caruana, 2005), thus allowing us to treat the mean and variance of prediction as the posterior distribution of objectives. While the detailed exploration of uncertainty estimation of deep ensembled regression models is not the primary emphasis of this study, readers interested in a more comprehensive and rigorous discussion on this topic can refer to the literature (Lakshminarayanan et al., 2017; Nix and Weigend, 1994; Abdar et al., 2021).

However, in the scenario of multi-objective optimization, fewer valid candidates meet the criterion due to the increased number of objectives to optimize and all objectives have to be larger than 1 (refer to Section 4.4.3). The EI method is unable to filter out invalid designs in this context. Hence, the Upper Confidence Bound (UCB) acquisition method is employed to guide the multi-objective optimization. The acquisition value calculated by UCB acts as an upper bound for the objective value, enabling us to disregard candidates whose objective values fall below 1. The hyperparameter β is set to 3 to increase the upper confidence bound for potential optimal designs, thereby allowing more valid candidates to be considered.

5.2.3 Pareto Front

Section 4.4.3 outlines the approach to addressing the multi-objective optimization challenge in chiral metamaterial design by searching for the Pareto front across all designs. Here this section elaborates on the methodology of discovering the Pareto front and its utilization within Bayesian Optimization. During each iteration of multi-objective Bayesian optimization, the acquisition value of each unseen data point is determined using the Upper

Confidence Bound (UCB) method. Provided that the goal is to optimize a total of m objectives simultaneously, the acquisition value of points i, calculated by eqn. 5.6, is denoted as $\alpha^i = [\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i]$, where α_j^i represents the acquisition value for the unseen data point i in terms of maximizing objective j. Therefore, if there are n unseen points in the pool of design candidates, a matrix $A_{n\times m}$ with row i representing α^i, where $i \in \{1,2,\cdots,n\}$ can be obtained. The next points to be acquired will be the design points whose performances form the Pareto front of $A_{n\times m}$. The Pareto front is identified through an efficient algorithm available in the Python package **artemis-ml** (QUVA-Lab, 2017). The algorithm iteratively eliminates the points that are dominated by at least one other point until only non-dominated points remain, thus yielding the set of Pareto front. The algorithm can be summarized as follows:

Algorithm 2 Find Pareto Front

Require: Acquisition value $A_{n\times m}$ for m objectives of unseen points $1,2,\cdots n$

choose the first point $i_1 = 1$
while $i_1 <= n$ **do**
    for $i_2 = 1,2,\ldots,n$ **do**
        remove point i_2 if $c_j^{i_2} < c_j^{i_1}$ for all $j \in \{1,2,\cdots,m\}$ ▷ i_2 is removed because it is dominated by i_1
    end for
    update n as the size of the rest points
    re-index the rest points as $1,2,\cdots,n$ without changing the order
    $i_1 \leftarrow$ the index of the next point on the order
end while
return the rest points (i.e. nondominated points)

5.3 Results and Discussion

This Section presents the optimal designs discovered using the Bayesian Optimization detailed in Section 5.2. Specifically, Section 5.3.1 presents the results and analysis for the optimal designs that exhibit extreme non-reciprocity. Section 5.3.2 presents the results and analysis for the optimal designs that exhibit extreme elastic asymmetry. Finally, Section 5.3.3

presents optimal designs that were discovered by maximizing both elastic asymmetry and non-reciprocity.

5.3.1 Non-Reciprocity Optimization

For the eight non-reciprocity objectives defined in Section 4.4.1, the search for the optimal design that maximizes each objective is conducted using the Bayesian Optimization methods detailed in Section 5.2. Each objective is maximized individually through separate Bayesian Optimization runs. The designs achieving the highest objective value after optimization are shown in Fig. 5·4. Below is a summary of the optimal performance for each objective.

$$f_1 = |\frac{k_{xx}^-}{k_{xx}^+}| = |39.66/3.11| = 12.75, \quad f_2 = |\frac{k_{xx}^+}{k_{xx}^-}| = |33.24/2.43| = 13.68,$$
$$f_3 = |\frac{k_{xy}^-}{k_{xy}^+}| = |19.70/(-0.04)| = 492.50, \quad f_4 = |\frac{k_{xy}^+}{k_{xy}^-}| = |32.09/1.34| = 23.95,$$
$$f_5 = |\frac{k_{yx}^-}{k_{yx}^+}| = |5.16/0.01| = 516.00, \quad f_6 = |\frac{k_{yx}^+}{k_{yx}^-}| = |26.00/0.97| = 26.80,$$
$$f_7 = |\frac{k_{yy}^-}{k_{yy}^+}| = |22.22/1.19| = 18.67, \quad f_8 = |\frac{k_{yy}^+}{k_{yy}^-}| = |30.93/0.71| = 43.56,$$

It is clear that the optimal designs for objectives f_1, f_3, f_5, f_7 in Fig. 5·4(a)(c)(e)(g) share some similarities, and are all discovered from design space 2. Referring to the stiffness definition illustrated in Fig. 4·2, these objectives necessitate either that the stiffness values for compression $(-x)$ are greater than those for extension $(+x)$, or the stiffness values for clockwise rotation $(-y)$ are greater than those for anti-clockwise rotation $(+y)$. By looking into the deformation process of these structures under various loads, it is found that during compression $(-x)$ and clockwise rotation $(-y)$, contact is established between the ligament and the rigid circle. Conversely, during extension and anti-clockwise rotation, the ligament and the rigid circle detach, resulting in no contact forces during deformation. This mechanism can be understood through intuitively imaging the deformation process of these geometries. Here, the optimal structure for objective $f_1 = |k_{xx}^-/k_{xx}^+|$ is selected as an example and its contact behavior during deformation is shown in Fig. 5·5(a). Notably, there exists a substantial contact area when the structure is subject to compression load (see Fig. 5·5(a-i)) and no contact area for the extension load (see Fig. 5·5(a-ii)). The optimal

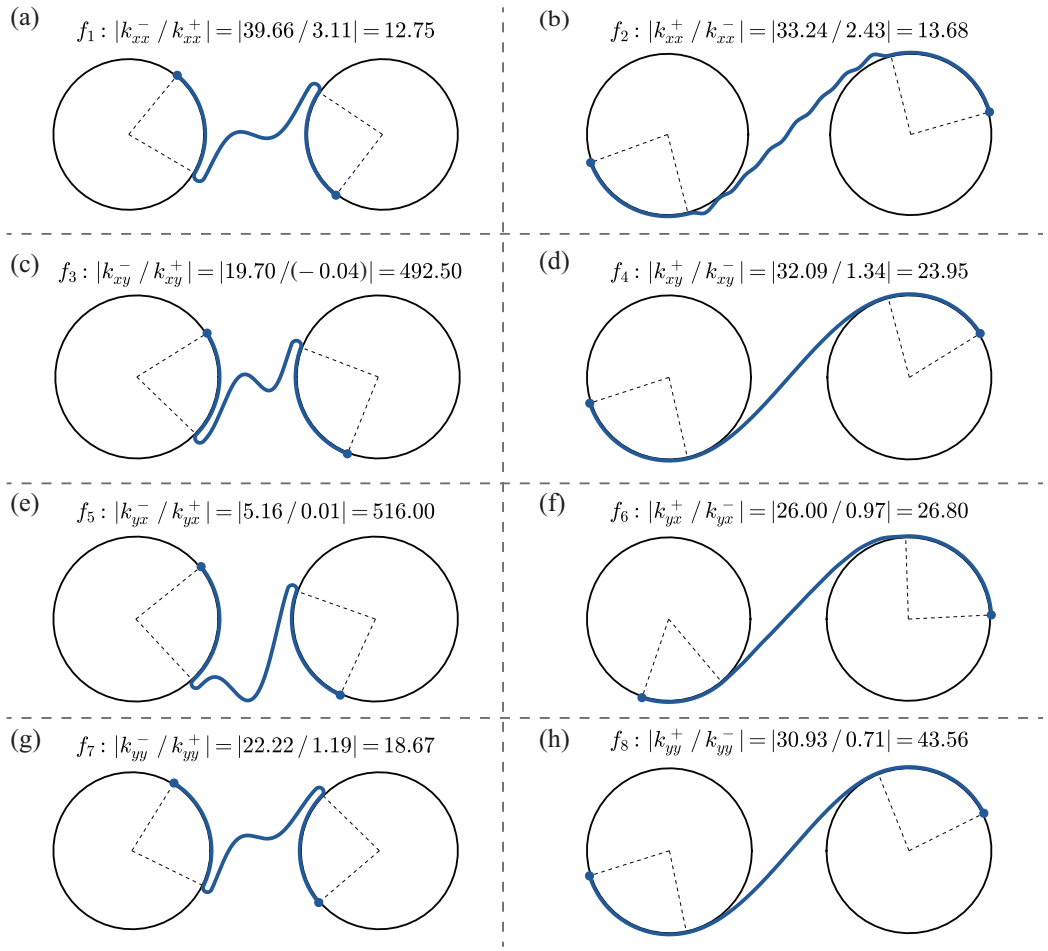


Figure 5·4: Optimal designs for maximizing eight non-reciprocity objectives after 10 iterations of Bayesian Optimization. Each figure shows the optimal design for the (a) objective f_1, (b) objective f_2, (c) objective f_3, (d) objective f_4, (e) objective f_5, (f) objective f_6, (g) objective f_7 and (h) objective f_8. The title above each structure indicates the corresponding stiffness values.

designs for objectives f_3, f_5, and f_7 exhibit similar contact modes, which are visualized in Appendix A.4, Fig. A·10.

Essentially, higher stiffness values are typically observed during contact, as illustrated in Fig. 5·5 and Fig. A·10, while lower stiffness values are observed when there is no contact. Additionally, based on the design parameters outlined in Table 4.1, the initial contact angles between the ligament and the circle, denoted as θ_1 and θ_{11}, range from 20 to 90 degrees. Notably, the initial contact angles of the optimal designs depicted in Fig. 5·4 (a)(c)(e)(g) are

all closer to the upper bound, i.e., 90 degrees, which enables the establishment of a larger contact area under loads. Therefore, maximizing the difference between higher and lower stiffness promotes a larger initial contact area between the ligament and the rigid circle. Upon closer inspection of the optimal designs, it is evident that the area where the ligament and the circle connect is large and towards the center of the structure provides stronger resistance during compression loads. Thus, it can be found that the larger initial connecting area between the ligament and the circle tends to enhance the stiffness, as it will increase the contact area under loads from certain directions. Moreover, the shape of the ligament for the designs depicted in Fig. 5·4(a)(c)(e)(g) tends to be curved in the middle, which provides additional resistance against the compression load.

The optimal designs for objectives f_2, f_4, f_6, f_8 in Fig. 5·4(b)(d)(f)(h) represent the reversed versions of those in Fig. 5·4(a)(c)(e)(g). Specifically, for the four objectives f_2, f_4, f_6, and f_8, the goal is to design material such that the stiffness values for extension $(+x)$ are greater than those for compression $(-x)$, and the stiffness values for anti-clockwise rotation $(+y)$ are greater than those for clockwise rotation $(-y)$. Accordingly, the optimal geometries for these objectives in Fig. 5·4(b)(d)(f)(h) are also alike but in contrast with those of the optimal designs in Fig. 5·4(a)(c)(e)(g). Concretely, in this set of optimal designs, the contact areas are situated away from the central region, and the ligaments are straight rather than curved. In addition, contact occurs during extension and anti-clockwise rotation, while no contact is observed during compression and clockwise rotation, which contrasts with the structures in Fig. 5·4(a)(c)(e)(g). The contact modes of the optimal structure for objective $f_4 = |k_{yy}^{-}/k_{yy}^{+}|$ are shown in Fig. 5·5(b). Here, a substantial contact area is observed during anti-clockwise rotation (see Fig. 5·5(b-i)), and no contact area is observed during clockwise rotation (see Fig. 5·5(b-ii)). The high k_{yy}^{-} is attributed to the contact force being in the opposite direction of the applied force, enhancing the structure's resistance to external forces and resulting in increased stiffness. Additionally, the straight ligament facilitates stretching

strain energy when subjected to extension or anti-clockwise rotation. The contact modes of optimal designs for objectives f_2, f_6 and f_8 are visualized in Appendix A.4, Fig. A·10.

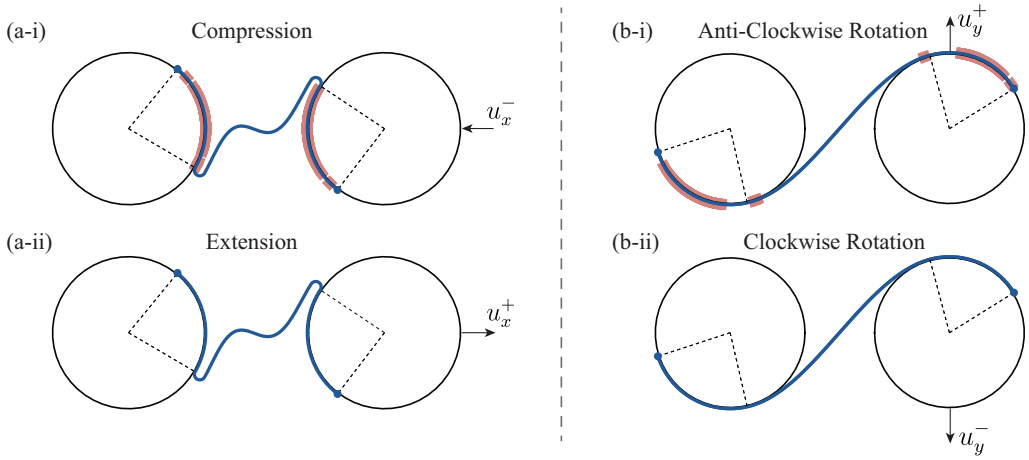


Figure 5·5: Two examples of contact modes under loads from different directions. The highlighted pink color indicates the contact areas after the deformation. (a) The contact modes for the optimal design of objective $f_1 = |k_{xx}^-/k_{xx}^+|$. The stiffness value k_{xx}^- is obtained during the (a-i) compression, and k_{xx}^+ is obtained during the (a-ii) extension. (b) The contact modes optimal design of objective $f_4 = |k_{yy}^-/k_{yy}^+|$. The stiffness value k_{yy}^+ is obtained during the (b-i) anti-clockwise rotation, and k_{yy}^- is obtained during the (b-ii) clockwise rotation.

5.3.2 Elastic Asymmetry Optimization

The optimal designs discovered for the asymmetry objectives $g_{1:8}$ defined in Section 4.4.2 are presented in Fig. 5·6. A summary of the optimal performance for each objective is below

$$g_1 = |\frac{k_{xy}^-}{k_{yx}^-}| = |2.45/1.21| = 2.02, \quad g_2 = |\frac{k_{xy}^-}{k_{yx}^+}| = |19.70/(-0.04)| = 492.50,$$
$$g_3 = |\frac{k_{xy}^+}{k_{yx}^-}| = |29.68/1.08| = 27.48, \quad g_4 = |\frac{k_{xy}^+}{k_{yx}^+}| = |2.39/2.36| = 1.01,$$
$$g_5 = |\frac{k_{yx}^-}{k_{xy}^-}| = |0.71/0.59| = 1.20, \quad g_6 = |\frac{k_{yx}^+}{k_{xy}^-}| = |26.00/1.04| = 25.00,$$
$$g_7 = |\frac{k_{yx}^-}{k_{xy}^+}| = |3.72/2.11^{-3}| = 1763.03, \quad g_8 = |\frac{k_{yx}^+}{k_{xy}^+}| = |1.39/0.62| = 2.24.$$

In contrast to the optimal solutions for non-reciprocity, not all of the optimal designs for asymmetry achieve notably high objective values, as defined by cases where the stiffness of the numerator is larger than the denominator by at least one order of magnitude. Specifically,

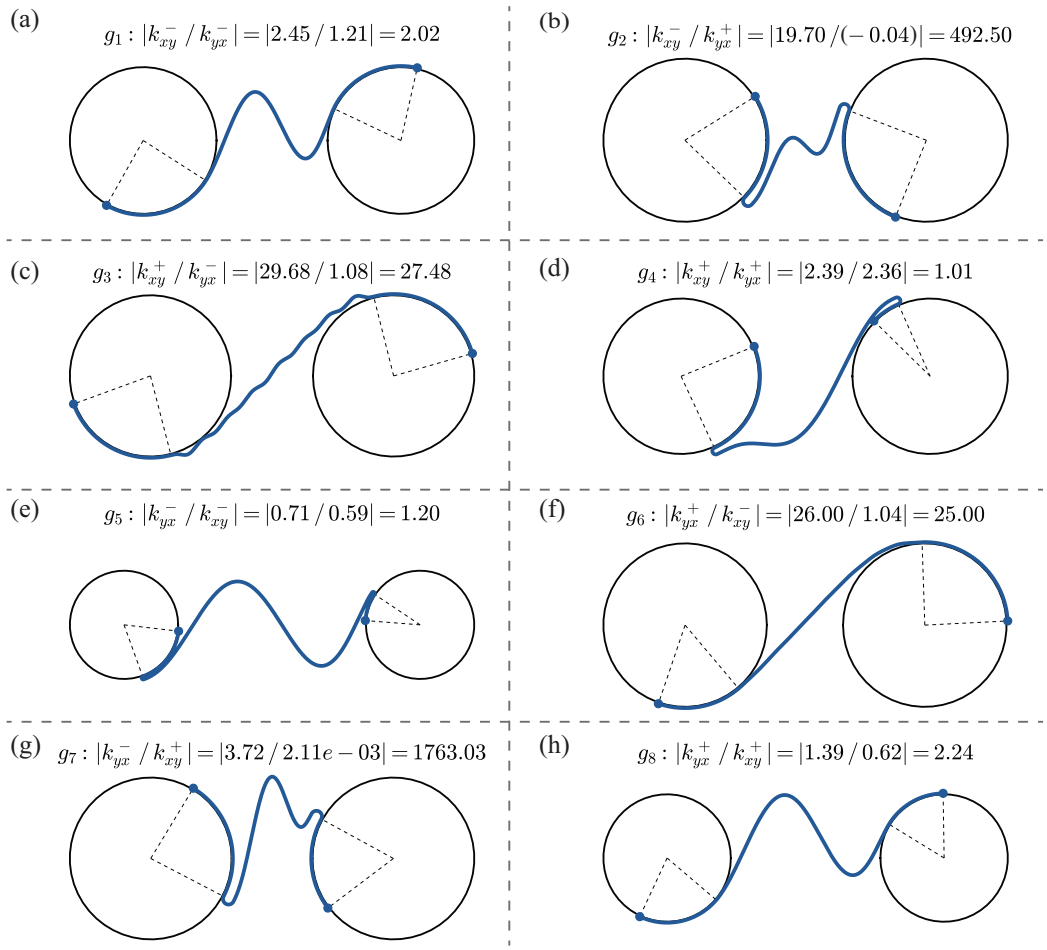


Figure 5·6: Optimal designs for maximizing eight asymmetry objectives after 10 iterations of Bayesian Optimization. Each figure shows the optimal design for the (a) objective g_1, (b) objective g_2, (c) objective g_3, (d) objective g_4, (e) objective g_5, (f) objective g_6, (g) objective g_7 and (h) objective g_8. The title above each structure indicates the corresponding stiffness values.

the optimal designs for g_2, g_3, g_6, and g_7 achieve high objective values, where the stiffness values in the numerator significantly surpass those in the denominator. These optimal designs exhibit a substantial contact area for one direction but no contact for the other. As a result, the stiffness value for the loading direction where contact is established tends to be much higher, as the contact provides increased resistance to the loads. For a detailed visualization of the contact modes under different load directions, please refer to Appendix A.4, Fig. A·11.

In contrast, the objective values for g_1, g_5, and g_8 are relatively small. This is primarily

attributed to the challenge of finding designs capable of exhibiting large contact areas in the desired direction while maintaining no contact in the other. In design spaces 1 and 3, contact is more likely to occur during the extension and anti-clockwise rotation loads, while no contact is expected for loads from the other two directions. Conversely, in design space 2, contact is more probable during compression and clockwise rotation, with no contact expected for loads from the other two directions. In design space 4, contact is likely to occur for loads in all directions. Upon scrutinizing objectives g_1, g_5, and g_8, it can be found that the required contact modes are not commonly found within these design spaces. For instance, achieving a high value of objective $g_1 = |k_{xy}^{-}/k_{yx}^{-}|$ necessitates a large contact area for clockwise rotation loads and no contact for compression, which is not a common scenario within the four design spaces. Nevertheless, the optimal design for objective g_1 was found from design space 1. While contact still occurs during compression loading and there is no contact during clockwise rotation, the contact area is very small, mitigating the stiffness value k_{yx}^{-} during compression. As a result, the final asymmetry results in k_{xy}^{-} being slightly larger than k_{yx}^{-}.

To facilitate the visualization of contact in these optimal structures, Fig. 5·7 presents two examples: the optimal designs for objectives $g_1 = |k_{xy}^{-}/k_{yx}^{-}|$ and $g_8 = |k_{yx}^{+}/k_{xy}^{+}|$. For both structures, contact modes at the two load directions are different, leading to an asymmetry in the stiffness values. However, the contact area is significantly smaller for the loading direction for which the stiffness value is anticipated to be smaller.

Finally, the optimal design for g_4 yields an objective value of nearly 1, indicating the absence of designs meeting the criterion where the stiffness value k_{xy}^{+} exceeds k_{yx}^{+}. To maximize objective $g_4 = |k_{xy}^{+}/k_{yx}^{+}|$, an ideal design should feature a substantial contact area under anti-clockwise rotation while exhibiting no contact under extension. As discussed above, such a contact mode is not prevalent within the existing design spaces. The exploration of alternative design spaces capable of accommodating optimal designs meeting the

requirements of objective g_4 remains a topic for future investigation.

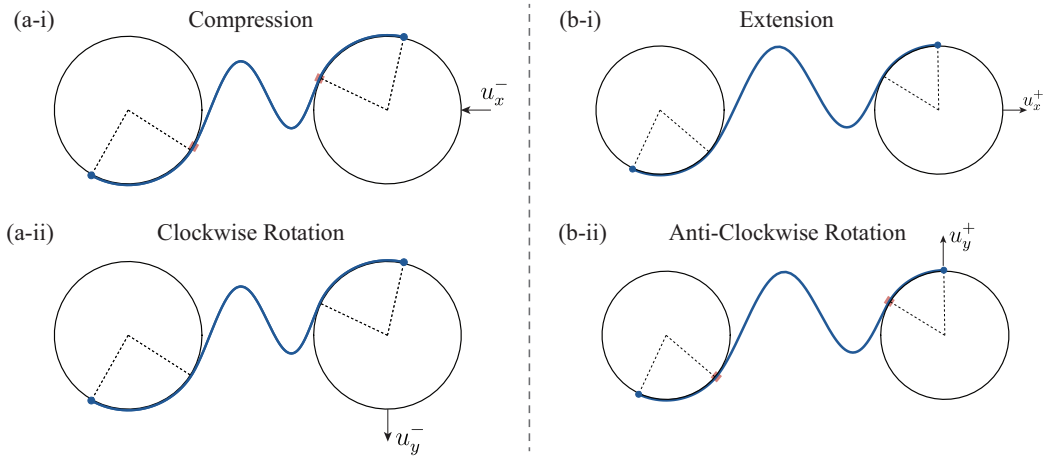


Figure 5·7: Two examples of contact modes under loads from different directions. The highlighted pink color indicates the contact areas. (a) The contact modes for the optimal design of objective $g_1 = |k_{xy}^-/k_{yx}^-|$. The stiffness value k_{yx}^- is obtained during the (a-i) compression, and k_{xy}^- is obtained during the (a-ii) clockwise rotation. (b) The contact modes for objective $g_8 = |k_{yx}^+/k_{xy}^+|$. The stiffness value k_{yx}^+ is obtained during the (b-i) extension, and k_{xy}^+ is obtained during the (b-ii) anti-clockwise rotation.

5.3.3 Multi-Objective Optimization

In this Section, the goal is to identify optimal designs that exhibit both non-reciprocity and asymmetry, as defined in Section 4.4.3. Therefore, there are two objectives to optimize, selecting one from $f_{1:8}$, and another from $g_{1:8}$. Based on the results of single objective optimization in Sections 5.3.1 and 5.3.2, it can be found that the optimal designs for objectives f_1, f_3, f_5, f_7, g_2, and g_7 exhibit similarities. Thus these objectives are designated as Group 1 since they are not contradictory. Similarly, the optimal designs for objectives f_2, f_4, f_6, f_8, g_3, and g_6 share similarities and are classified as Group 2. It is important to note that the objectives in Group 1 and Group 2 are contradictory because their optimal designs for single objective optimization exhibit opposite characteristics. Additionally, objectives g_1, g_5, and g_8 are designated as Group 3 due to their relatively small optimal values discovered in Section 5.3.2. Objective g_4 does not have an optimal design in the

single objective optimization, so it is not considered in the multi-objective optimization scenario. The three groups of objectives are summarized below.

- Group 1 (non-contradictory): f_1, f_3, f_5, f_7, g_2, g_7
- Group 2 (non-contradictory): f_2, f_4, f_6, f_8, g_3, g_6
- Group 3 (challenging): g_1, g_5, g_8

Based on the different groups established above for the objectives, three different cases are explored for multi-objective optimization. In the first case, non-contradictory objectives in the same group are considered, where both objectives are chosen from Group 1 or both chosen from Group 2. As a representative case, the objectives f_1 and g_2 from Group 1 are selected. In the second case, two contradictory objectives are chosen, one from Group 1 and another from Group 2. The representative case adopted is f_1 and g_3. In the third case, one challenging objective from Group 3 and one relatively feasible objective from Group 1 or Group 2 are chosen. The selected representative case is f_1 and g_1. The distribution of the objectives for the three cases can be found in Appendix A.5, Fig. A·12 - A·14. Below is a summary of the three representative multi-objective optimization cases:

- Multi-Objective 1 (non-contradictory): f_1 and g_2
- Multi-Objective 2 (contradictory): f_1 and g_3
- Multi-Objective 3 (challenging): f_1 and g_1

After 10 iterations of Bayesion Optimization, three Pareto optima are identified for Multi-Objective 1, one Pareto optimum for Multi-Objective 2 and four for Multi-Objective 3. The Pareto front of each multi-objective and its corresponding designs can be found in Appendix A.6. For each Multi-Objective, the Pareto optima with the largest modulus (i.e. square root of the sum of all the objective values) is selected and is presented in Fig. 5·8.

In Multi-Objective 1, where the objectives f_1 and g_2 are not contradictory, both objectives achieved relatively high values with magnitudes larger than ten, as depicted in Fig. 5·8(a). However, Multi-Objective 2 presents a challenge as it involves two contradictory objectives, f_1 and g_3, where increasing one objective results in a decrease in the other, as illustrated in Fig. A·13. The optimal design discovered by ML methods shows a trade-off between asymmetry and non-reciprocal properties, albeit with relatively smaller values for both objectives, as shown in Fig. 5·8(b). Similarly, Multi-Objective 3 includes a challenging objective, g_1, and an easily attainable objective, f_1. As shown in Fig. A·14, the high value of g_1 is observed only when f_1 is small, indicating a contradiction between the two objectives. The optimal design after balancing these conflicting objectives is presented in Fig. 5·8(c). Notably, the optimal designs for the contradictory multi-objectives 2 and 3 are both found in Design Space 4 (refer to Fig. 4·6), while all the optimal designs are obtained from Design Space 1-3 during single objective optimization. This highlights that optimizing a single objective does not necessarily ensure high performance across all dimensions. Leveraging the Pareto Front and Bayesian Optimization facilitates the discovery of designs that exhibit multiple desirable properties simultaneously.

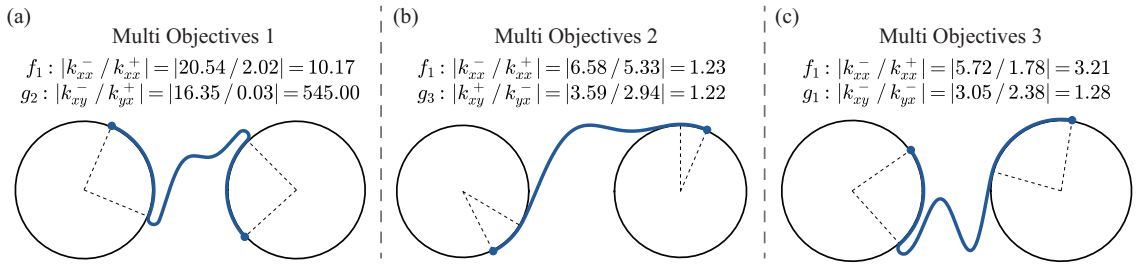


Figure 5·8: The Pareto front and the corresponding designs for optimizing (a) Multi-Objective 1: f_1 and g_1, (b) Multi-Objective 2: f_1 and g_3, (c) Multi-Objective 3: f_1 and g_1. The title above each structure indicates the corresponding objective values for the structure.

Upon scrutinizing the contact modes of these optimal designs, a notable distinction is observed for the optimal design depicted in Fig. 5·8(c) for Multi-Objective 3 compared to

the others. Specifically, while the majority of chiral structures typically exhibit only two distinct contact states under loads from four different directions, as illustrated in the example case in Fig. 5·2, the structure in Fig. 5·8(c) displays four different types of contact states under loads from four different directions. The four distinct contact statuses are illustrated in Fig. 5·9, where the highlighted pink areas denote the contact regions under each load. Notably, the contact areas vary depending on the applied loads. Since each contact mode corresponds to beam models with different boundary conditions, as discussed in Section 4.2, the stiffness values vary in all directions, resulting in $[k_{xx}^{-}, k_{xy}^{-}, k_{yx}^{-}, k_{yy}^{-}, k_{xx}^{+}, k_{xy}^{+}, k_{yx}^{+}, k_{yy}^{+}] = [5.72, 3.05, 2.38, 3.51, 1.78, 0.42, 1.13, 2.11]$. Consequently, this structure can satisfy the most objectives in terms of achieving both non-reciprocity and asymmetry, summarized below:

$$f_1 = |\frac{k_{xx}^{-}}{k_{xx}^{+}}| = 5.72/1.78 = 3.21, \quad f_3 = |\frac{k_{xy}^{-}}{k_{xy}^{+}}| = 3.05/0.42 = 7.26,$$
$$f_5 = |\frac{k_{yx}^{-}}{k_{yx}^{+}}| = 2.38/1.13 = 2.10, \quad f_7 = |\frac{k_{yy}^{-}}{k_{yy}^{+}}| = 3.51/2.11 = 1.66,$$
$$g_1 = |\frac{k_{xy}^{-}}{k_{yx}^{-}}| = 3.05/2.38 = 1.28, \quad g_2 = |\frac{k_{xy}^{-}}{k_{yx}^{+}}| = 3.05/1.13 = 2.69,$$
$$g_7 = |\frac{k_{yx}^{-}}{k_{xy}^{+}}| = 2.38/0.42 = 5.66, \quad g_8 = |\frac{k_{yx}^{+}}{k_{xy}^{+}}| = 1.13/0.42 = 2.69.$$

5.4 Conclusion

In Chapter 4 and Chapter 5, machine learning (ML) techniques are utilized in conjunction with Finite Element Simulation to engineer chiral metamaterials exhibiting significant directional non-reciprocity and stiffness asymmetry. The mechanisms underlying non-reciprocity and asymmetry are elucidated by qualitatively analyzing the contact behavior under various loads through equivalent beam models. Additionally, through quantitative analysis of strain energy changes during chiral metamaterial deformation, it was uncovered that stretching deformation of the elastic ligament in the chiral structures leads to higher stiffness, while the bending structures result in lower stiffness values. This insight explains the heuristic behind programming the chiral metamaterial geometry to achieve desired

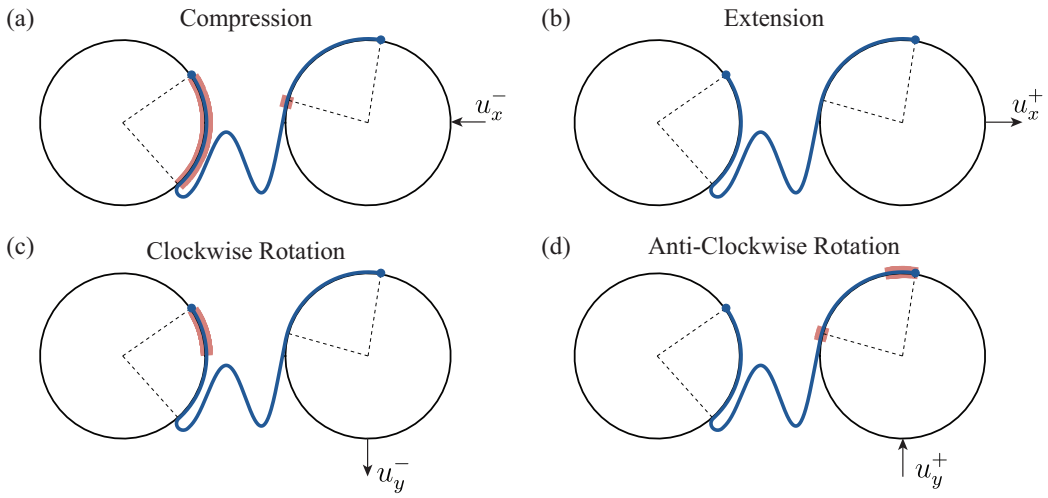


Figure 5·9: The contact modes for the optimal design shown in Fig. 5·8(c). The highlighted pink color indicates the different contact areas between the rigid circle and the elastic ligament under (a) Compression, (b) Extension, (c) Clockwise Rotation, and (d) Anti-Clockwise Rotation. The structure satisfies the most objectives in terms of achieving both non-reciprocity and asymmetry that are summarized in Section 5.3.3.

non-reciprocity and asymmetry properties.

This study encompassed the optimization of eight non-reciprocity objectives, eight asymmetry objectives, and three multi-objectives incorporating both non-reciprocity and asymmetry. To formalize the problem under the framework of ML, the design freedom of chiral structures was defined and each structure was characterized by its geometric features. Four different design spaces were proposed, with each being governed by a different contact mechanism. To reduce the computational cost, the Bayesian Optimization algorithm facilitated with ensemble learning was implemented to efficiently search the optimal structures for each objective by balancing between the exploration and exploitation. Particularly, multi-objective optimization was achieved by searching the Pareto Front of the performance spaces, and the corresponding optimal structures were obtained to exhibit both non-reciprocity and asymmetry. Following the completion of the optimization, the discovered optimal designs were presented for each single objective and an analysis was

performed on their similarities and disparities, highlighting the role of contact in enhancing chiral structure stiffness. Leveraging insights from single-objective optimization, the focus was placed on three distinct types of multi-objectives, including both contradictory and non-contradictory combinations of multi-objectives. Among the discovered optimal designs for multi-objectives, there is a structure that demonstrates the maximum diversity in non-reciprocity and asymmetry, which is attributed to the various contact behaviors of the structure under different loads.

In contrast to the majority of the current ML and mechanics literature focusing on testing the applicability of machine learning (ML) methods in solving conventional mechanics problems, where their effectiveness can be validated against known solutions (Brunton et al., 2016; Chen and Gu, 2021; Jiao et al., 2024), this study employed ML to a novel domain with limited prior information. With the aid of ML, this study uncovers a compelling relationship between contact behaviors and the non-reciprocity and asymmetry of chiral metamaterials through qualitative, quantitative, and case-based analyses. A noteworthy observation from the results is the presence of extreme non-reciprocity and asymmetry in the discovered optimal designs. For instance, as illustrated in Fig. 5·6(g), the optimal designs for the objective g_7 have a stiffness k_{yx}^{-} that is 1763 times larger than k_{xy}^{+}. The extreme and diverse non-reciprocity and asymmetry observed in the optimized chiral structures pave the way for metamaterials capable of novel wave propagation characteristics, including unidirectional wave propagation and non-Hermitian wave phenonena (Wang et al., 2024), which offers promising prospects for applications in non-reciprocal wave propagation.

While optimizing the chiral metamaterial to achieve various levels of non-reciprocity and asymmetry, it was observed that some of the asymmetry objectives did not achieve notably high objective values, and the objective g_4 resulted in no qualified optimal design within the current scope of design spaces. Additionally, although the optimal design for the multi-objective optimization displayed diverse non-reciprocity and asymmetry characteristics, as

depicted in Fig. 5·9, the absolute magnitudes of most objective values remained relatively small. This is attributed to the current confined design spaces, i.e., the rigid body of the structure can only be circles, and the shape of partially connected elastic ligaments is well-defined by certain base functions, as detailed in Section 4.5. Looking ahead, the aim is to broaden the design scope to explore a wider range of possibilities and search for designs capable of exhibiting extreme multi-objective performance. Furthermore, by leveraging deep generative models such as GANs (Generative Adversarial Networks) (Wang et al., 2022; Kobeissi et al., 2022), Transformers (Buehler, 2024, 2023a), and Diffusion Models (Ni et al., 2024; Luu et al., 2023), it is anticipated that advanced AI-designed structures with non-conventional properties can provide insights not only into the field of odd elastic materials but also into other emerging novel domains.

Chapter 6

Summary

6.1 Summary

This thesis focused on exploring the limitations and potential of machine learning techniques on rarely examined challenges in metamaterial prediction and design. Firstly, this thesis investigated the robustness of machine learning methods in addressing real-world mechanics problems characterized by data distribution shifts. Chapter 2 examined the prediction of heterogeneous metamaterials under three distinct data distribution shift scenarios: covariate shift, mechanism shift, and sampling bias. For each data distribution shift scenario, two benchmark datasets were created, one sourced from Mechanical MNIST and another from Mechanical EMNIST-Letters. Subsequently, in Chapter 3, three OOD algorithms (IRM, REx, IGA) alongside the traditional ML method (ERM) were implemented on two ML models (MLE and modified LeNet) to conduct a comparative analysis of the performance of these methods on the benchmark datasets created in Chapter 2. Meanwhile, the discussion regarding their limitations, and the insights into improving these OOD algorithms' efficacy in metamaterial prediction, are also provided.

The second challenge lies in applying ML methods to the intricate design and optimization of chiral metamaterials where the prior knowledge about the material is unknown. Chapter 4 investigated the underlying heuristics behind chiral metamaterials' non-reciprocity and asymmetry properties from the perspective of contact behaviors and energy components under the deformation. After defining the optimization objectives, the design space was broadened compared to the originally-used chiral metamaterials to facilitate the search for

chiral metamaterials that can exhibit extreme non-reciprocity and asymmetry properties. In Chapter 5, the Bayesian Optimization method was utilized to search for optimal chiral metamaterial designs and identify the Pareto front of the chiral metamaterial design space, obtaining chiral structures that exhibit both extreme non-reciprocity and asymmetry.

The main contributions of this thesis are: 1) The benchmark dataset collection comprising three different types of OOD generalization problems specifically tailored for mechanics problems in metamaterials is created and made publicly accessible, serving as a valuable and foundational resource for future research endeavors focused on OOD regression problems within this domain. 2) The OOD algorithms improve the OOD generalization in most data distribution shift scenarios for metamaterial prediction. This evaluation is essential for providing insights to metamaterial design, which often involves exploring design spaces with unknown data distributions. 3) The heuristics behind contact behaviors governing the non-reciprocity property of chiral metamaterials was elucidated. 4) Machine learning-guided optimization techniques were leveraged to identify chiral metamaterials that exhibit both non-reciprocity and asymmetry, offering promising prospects for applications in non-reciprocal wave propagation.

6.2 Future Work

6.2.1 Challenges of OOD Generalization on Shuffled Data

In Chapter 3, three popular OOD algorithms have been introduced. These methods aim to address OOD generalization problems in mechanics by encouraging ML models to learn causal relationships within the data. The causal relationship is assumed to be invariant across different environments where the data are collected. In this study, the data are created in a way that they are divided into different environments, and there are at least two different training environments for ML models to learn from. However, the information about data collecting sources may not always be readily accessible for all mechanics problems. For

example, in the third data distribution shift setting, i.e., sampling bias, the hidden sampling probability of data may not known for most problems. When we get a mixed and shuffled dataset, how can we learn the causal relationship within it? There are two solutions towards addressing this potential challenge.

The first solution still involves utilizing the OOD generalization algorithms introduced in this thesis. However, one extra step needs to be taken to divide the dataset into different environments such that the data from the same environment exhibit similar distributions while the data from different environments are in different distributions. This step can be achieved through leveraging unsupervised learning techniques - data clustering (Oyewole and Thopil, 2023; Sinaga and Yang, 2020; Ren et al., 2022). After using the clustering method to divide data into different groups, the OOD algorithms can then be used to learn invariant relationships across different groups, i.e., training environments, and thus improve the OOD generalization performance of ML models.

The second solution aims to learn the causal relationship by integrating physics information into machine learning models. This solution is more suitable for domain-specific OOD generalization problems where expert knowledge about the domain is available. The heuristic behind this solution is that physics laws are usually invariant across different environments. Thus it is expected that the OOD generalization can be improved by embedding the physics information into neural networks and encouraging ML models to learn the invariant physics within the data. For example, in Cranmer et al. (2020) and Greydanus et al. (2019), integrating Lagrangian and Hamiltonian mechanics into baseline neural networks improves the performance of ML models on testing datasets dramatically compared to baseline models. Considering that most work about physics-informed ML models primarily concentrates on in-distribution generalization (Willard et al., 2020; Karpatne et al., 2017; Cranmer et al., 2020; Greydanus et al., 2019), exploring how physics-informed ML can aid out-of-distribution (OOD) generalization has significant potential for expediting the

application of ML in metamaterials design and other fields in mechanics.

6.2.2 Deep Learning for Metamaterials Design

In Chapter 5, Bayesian optimization methods were utilized to identify optimal chiral metamaterials with desired properties. This was achieved by searching from a large design pool and the best candidates were selected based on the predicted property given by surrogate ML models. There are two main advantages of this approach. Firstly, the method is effective because it is not data hungry, i.e., the demand for large observed data. Secondly, training surrogate ML models do not require huge computational resources. However, the constraint is that the optimal designs will only be found within the pre-defined design pools, such that the exploring of other potential novel designs is not enabled. To discover designs that exhibit novel functionality beyond the training data, employing deep generative models such as GANs (Generative Adversarial Networks) (Wang et al., 2022; Kobeissi et al., 2022), Transformers (Buehler, 2024, 2023a), and Diffusion Models (Ni et al., 2024; Luu et al., 2023) for metamaterial generation is promising for future research. The challenge for this future direction is that deep generative models usually require large computational resources and the quality of the generated designs is not guaranteed. While there is a plethora of research focusing on generating metamaterials using supervised deep generative models (Chen and Gu, 2020; Zheng et al., 2023; Liu et al., 2024), the validation for the generalized materials is not a big challenge for these design problems. However, this may not apply to chiral metamaterial design because there are several constraints that must be satisfied for a generalized chiral structure to be physically valid. For example, the ligament shape cannot be arbitrary and should be smooth, while the shape of the elastic ligament and the rigid circle at the connecting area must be aligned. Despite the challenges to overcome, it is anticipated that advanced AI-designed metamaterials with non-conventional properties can provide insights not only into the field of odd elastic materials but also into other emerging novel domains.

6.3 Data and Code Availability

The extensions of Mechanical MNIST data are available through the OpenBU Institutional Repository at https://open.bu.edu/handle/2144/39371. The access to all the OOD datasets is at https://open.bu.edu/handle/2144/44485. The code to reproduce equibiaxial simulation on FEniCS is at https://github.com/elejeune11/Mechanical-MNIST/blob/master/generate_dataset/Equibiaxial_Extension_FEA_test_FEniCS.py The code for implementing the four algorithms introduced in Section 3.1 and for creating the datasets is available at https://github.com/lingxiaoyuan/ood_mechanics

The data and the code to optimize chiral metamaterial designs and reproduce simulation on ABAQUS are openly available from https://github.com/lingxiaoyuan/chiral.

Appendix A

Appendix

A.1 Details of Hyperparameters

For all ML model training in Chaper 3, the learning rate is fixed at 0.001, the total number of training epochs is fixed at 50001, and for each epoch, each model was trained with one single batch with all training data, this information is listed in Table A.1. The final selection of penalty weight λ and anneal step t (count from zero) for the two ML models using the three OOD generalization algorithms(see Section 3.1) is based on the approach for hyperparameters tuning introduced in Section 3.3.1. These values are listed in Table A.2.

A.2 Additional Metrics and Visualization for OOD Evaluation

Violin Plots

Here, supporting information for the evaluation results shown in Section 3.4 is provided. In Section 3.4, only the aggregated mean prediction defined by eqn. 3.10 is shown in Figure 3·4 - 3·6 to ensure a clear comparison between the different environments and approaches. Here, the results are presented in a more detailed form using violin plots. In each violin

Table A.1: Training hyperparameters

	Learning Rate	Epochs	Batch Size
MLP	0.001	50001	training data size
LeNet	0.001	50001	training data size

Table A.2: Hyperparameter selection results where λ is the penalty weight for each OOD generalization method (IRM,REx, IGA), and t is the anneal step (or epoch) after which the penalty weight will be introduced during training.

Source Data		Mechanical MNIST				Mechanical MNIST - EMNIST Letters			
OOD problems		Covariate Shift Mechanism Shift		Sampling Bias		Covariate Shift Mechanism Shift		Sampling Bias	
Hyperparameters		λ	t	λ	t	λ	t	λ	t
MLP	IRM	1e-5	15000	1e-6	10000	1e-6	15000	1e-5	10000
MLP	REx	1e-1	15000	1e-1	10000	1e-2	15000	1e-1	10000
MLP	IGA	1e-2	15000	1e-1	10000	1e-2	15000	1e0	10000
LeNet	IRM	1e-6	15000	1e-5	15000	1e-5	15000	1e-6	15000
LeNet	REx	1e0	15000	1e-1	15000	1e-2	15000	1e-1	15000
LeNet	IGA	1e-4	15000	1e-1	15000	1e-5	15000	1e-3	15000

plot, there are 15 white points which represent the RMSE performance of the MLP model (or the modified LeNet model) with 15 different initialization, and one color-filled point which represents the aggregated mean prediction calculated through eqn. 3.10 based on the predicting quantity of interest (i.e., the change in strain energy after an equibiaxial extension) value given by these 15 models. Specifically, Fig. A·1 shows the performance of all algorithms (ERM, IRM, REx and IGA) on the training, validation, and test environments from the covariate shift dataset created with the methods described in Section 2.2 on both Mechanical MNIST (Fig. A·1a) and Mechanical EMNIST-Letters (Fig. A·1b). In Fig. A·1a shows a violin plot for each of the four algorithms implemented on the MLP model (Fig. A·1a-i), and the modified LeNet model (Fig. A·1a-ii). Similarly, in Fig. A·1b shows the evaluation results of all the algorithms on the covariate shift dataset from Mechanical EMNIST-Letters through violin plots. Following the same format, Fig. A·2 shows the performance of all algorithms on the mechanism shift dataset described in Section 2.3 for both Mechanical MNIST (Fig. A·2a) and Mechanical EMNIST-Letters (Fig. A·2b). And, Fig. A·3 shows the performance of all algorithms on the sampling bias dataset described in Section 2.4 for both Mechanical MNIST (Fig. A·2a) and Mechanical EMNIST-Letters (Fig. A·2b).

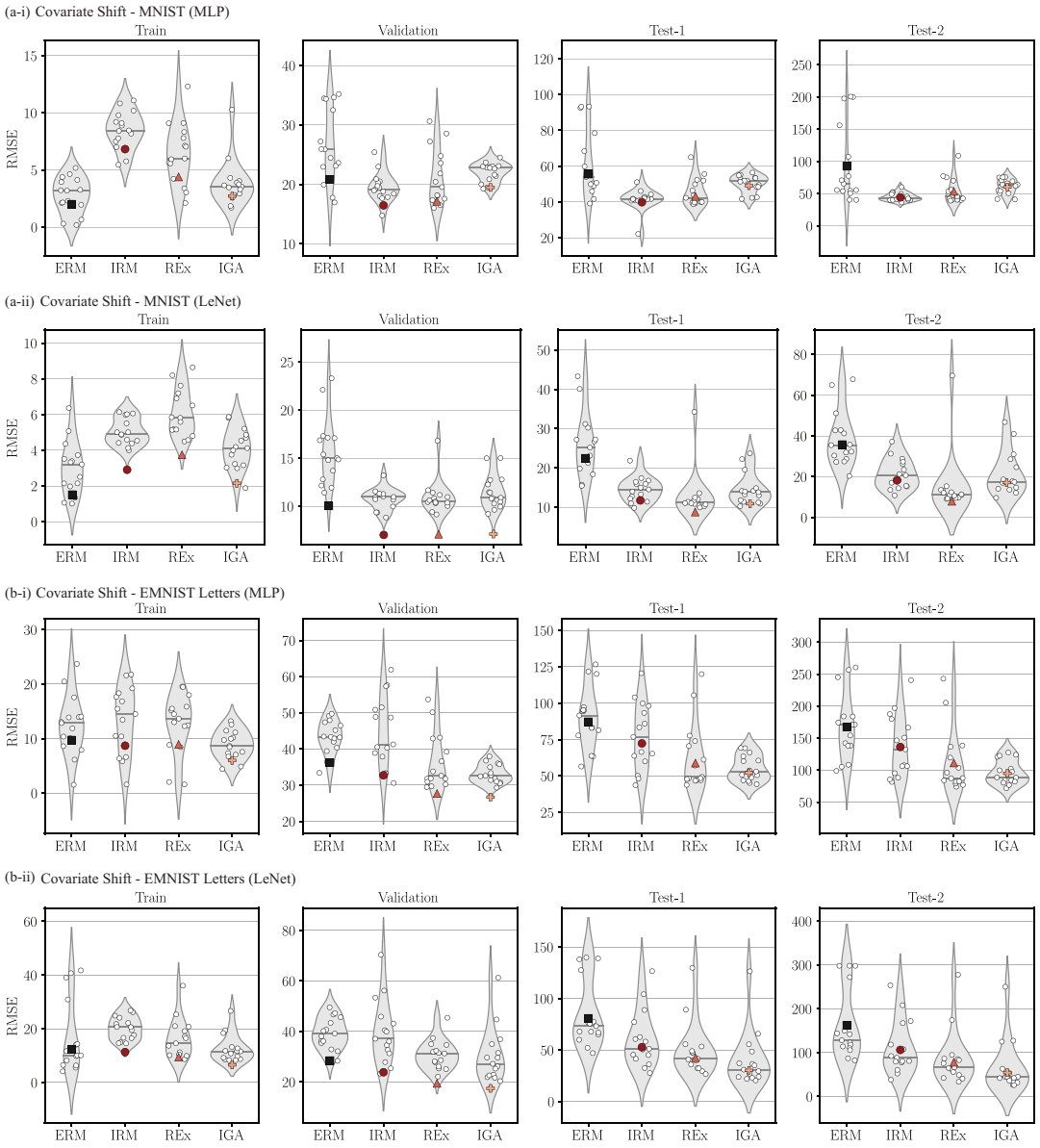


Figure A·1: The performance of the four algorithms (ERM, IRM, REx, IGA) on the covariate shift data defined in Section 2.2. Every white point represents the RMSE given by a single model initialized with different seeds. The color-filled points show the RMSE of the aggregated mean prediction calculated by eqn. 3.10. (a) The performance of a MLP model (a-i) and the modified LeNet model (a-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST Collection. (b) The performance of a MLP model (b-i) and a modified LeNet model (b-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST - EMNIST Letters Collection.

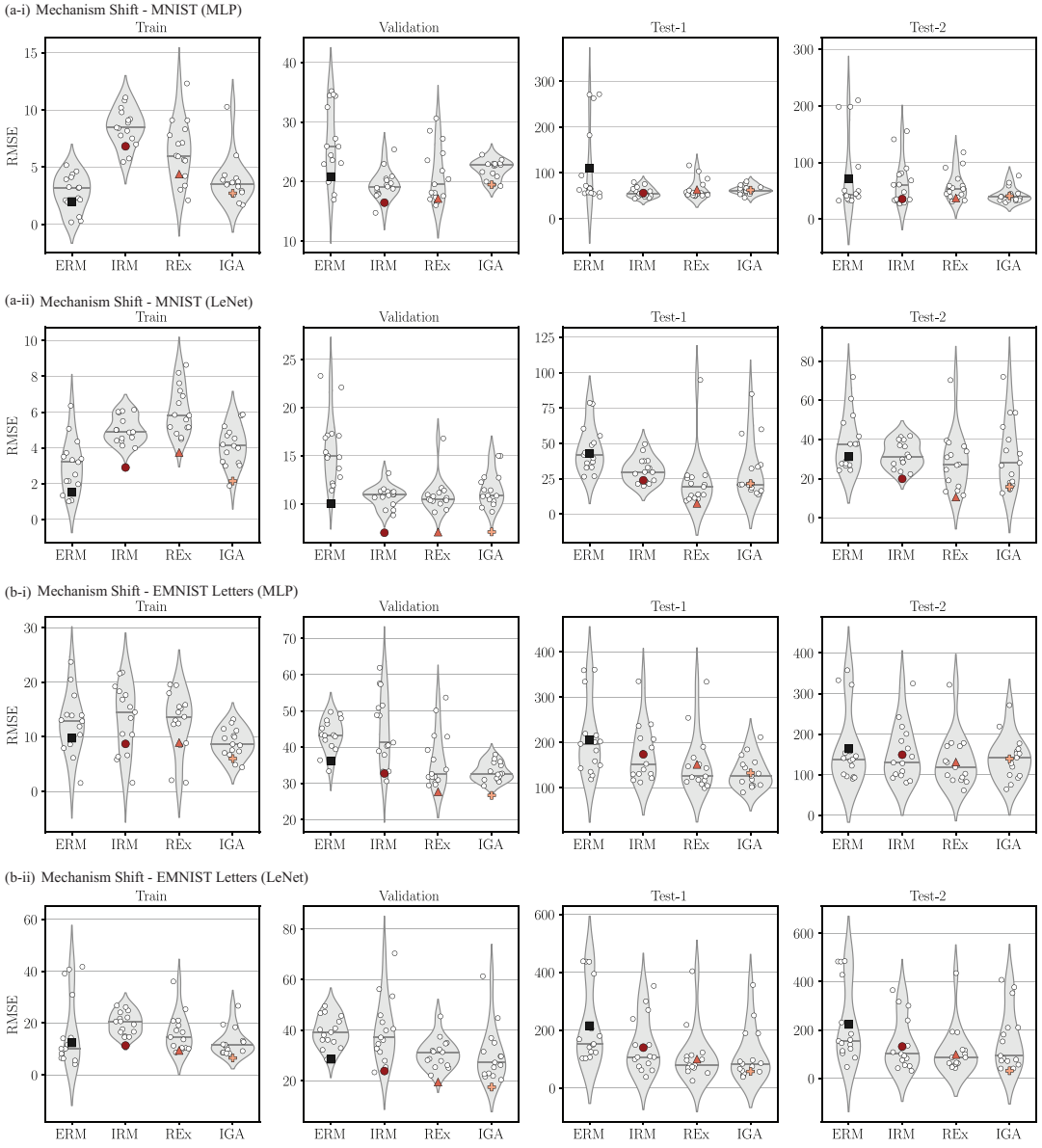


Figure A·2: The performance of the four algorithms (ERM, IRM, REx, IGA) on the mechanism shift data defined in Section 2.3. Every white point represents the RMSE given by a single model initialized with different seeds. The color-filled points show the RMSE of the aggregated mean prediction calculated by eqn. 3.10. (a) The performance of a MLP model (a-i) and the modified LeNet model (a-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST Collection. (b) The performance of a MLP model (b-i) and a modified LeNet model (b-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST - EMNIST Letters Collection.

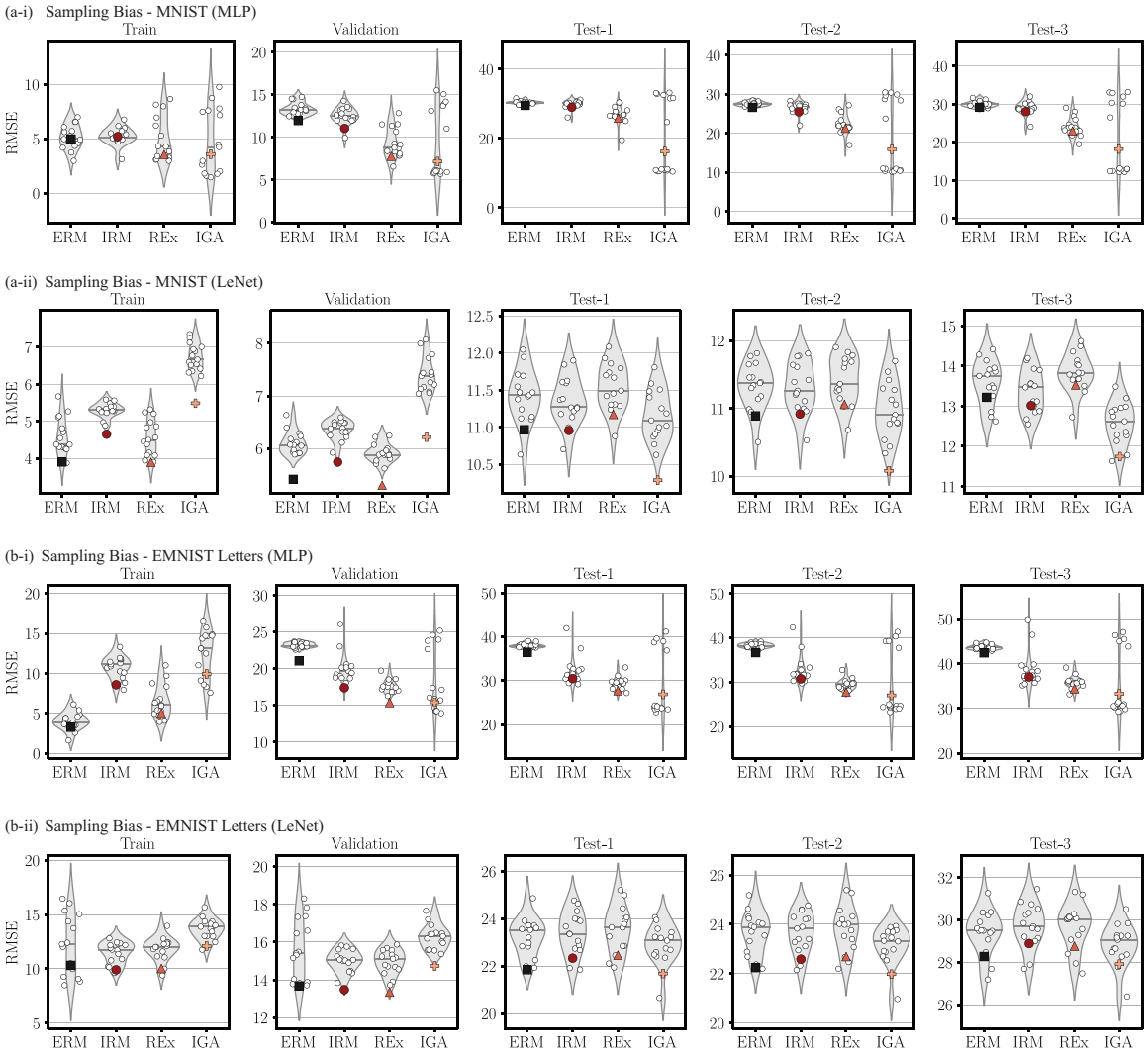


Figure A·3: The performance of the four algorithms (ERM, IRM, REx, IGA) on the sampling bias data defined in Section 2.4. Every white point represents the RMSE given by a single model initialized with different seeds. The color-filled points show the RMSE of the aggregated mean prediction calculated by eqn. 3.10. (a) The performance of a MLP model (a-i) and the modified LeNet model (a-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST Collection. (b) The performance of a MLP model (b-i) and a modified LeNet model (b-ii) trained by the four algorithms on training, validation, and testing data from the Mechanical MNIST - EMNIST Letters Collection.

Tables

Table A.3-A.8 present the RMSE performance calculated by both eqn. 3.10 and eqn. 3.11 for all methods introduced in Section 3.1 on all OOD datasets introduced in Chapter 2. Note that in these tables, the RMSE is calculated by eqn. 3.10 corresponds to the RMSE performance presented and discussed in Section 3.4. In addition, for each table, the mean and the standard deviation of the change in strain energy are shown for each group of data: training, validation, and testing datasets. Note that typically datasets with larger standard deviation of strain energy tend to correspond to a larger RMSE which makes the RMSE from Mechanical MNIST-EMNIST Letters generally larger than the RMSE from Mechanical MNIST. Table A.3 and Table A.4 show the RMSE performance for covariate shift datasets based on Mechanical MNIST and Mechanical MNIST-EMNIST Letters respectively. Table A.5 and Table A.6 show the RMSE performance for mechanism shift datasets based on Mechanical MNIST and Mechanical MNIST-EMNIST Letters respectively. Table A.7 and Table A.8 show the RMSE performance for sampling bias datasets based on Mechanical MNIST and Mechanical MNIST-EMNIST Letters respectively.

Table A.3: The performance of four algorithms (ERM, IRM, REx, IGA) on the **covariate shift** data from the **Mechanical MNIST Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2	
Statistics of Strain Energy		y_mean = 559.12 y_std = 45.99		y_mean = 559.35 y_std = 45.74		y_mean = 575.96 y_std = 48.17		y_mean = 562.21 y_std = 44.90	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	2.01	2.77	20.85	26.60	55.53	61.50	92.52	95.04
MLP	IRM	6.82	8.51	16.48	19.54	39.85	41.40	44.23	45.07
MLP	REx	4.39	6.45	17.17	21.16	43.11	45.60	53.44	54.55
MLP	IGA	2.72	3.87	19.50	22.08	49.11	50.34	60.33	60.98
LeNet	ERM	1.51	3.01	10.04	15.72	22.46	25.85	35.75	38.69
LeNet	IRM	2.91	5.04	7.03	10.78	11.74	14.55	18.21	21.40
LeNet	REx	3.75	6.11	7.10	10.94	8.77	12.82	7.94	15.13
LeNet	IGA	2.15	4.09	7.11	11.52	10.89	14.55	17.12	21.61

Table A.4: The performance of four algorithms (ERM, IRM, REx, IGA) on the **covariate shift** data from the **Mechanical MNIST - EMNIST Letters Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2	
Statistics of Strain Energy		y_mean = 703.01 y_std = 86.26		y_mean = 706.98 y_std = 88.45		y_mean = 695.56 y_std = 86.30		y_mean = 671.97 y_std = 79.58	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	9.72	12.40	36.21	43.34	87.24	90.10	167.57	168.80
MLP	IRM	8.70	13.35	32.72	44.84	72.41	77.36	136.24	138.15
MLP	REx	8.94	12.70	27.67	36.51	58.89	63.02	111.22	113.05
MLP	IGA	6.01	8.89	26.68	33.14	52.24	55.32	95.04	96.73
LeNet	ERM	12.33	16.89	28.47	39.44	80.77	84.93	162.48	164.70
LeNet	IRM	11.32	20.10	23.84	39.84	52.78	60.54	106.13	111.77
LeNet	REx	9.41	16.34	19.45	30.85	42.58	50.08	77.88	85.92
LeNet	IGA	6.54	12.60	17.47	30.45	30.27	40.41	54.33	65.97

Table A.5: The performance of four algorithms (ERM, IRM, REx, IGA) on the **mechanism shift** data from the **Mechanical MNIST Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2	
Statistics of Strain Energy		y_mean = 559.12 y_std = 45.99		y_mean = 559.35 y_std = 45.74		y_mean = 543.16 y_std = 37.94		y_mean = 504.45 y_std = 31.37	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	2.01	2.77	20.85	26.60	110.19	112.06	71.55	76.02
MLP	IRM	6.82	8.51	16.48	19.54	56.31	58.13	35.43	64.90
MLP	REx	4.39	6.45	17.17	21.16	63.50	65.08	37.13	58.78
MLP	IGA	2.72	3.87	19.50	22.08	62.55	63.36	40.64	42.85
LeNet	ERM	1.51	3.01	10.04	15.72	42.88	45.93	31.15	38.28
LeNet	IRM	2.91	5.04	7.03	10.78	23.96	31.38	19.89	32.28
LeNet	REx	3.75	6.11	7.10	10.94	7.70	23.12	10.59	28.94
LeNet	IGA	2.15	4.09	7.11	11.52	21.70	31.23	15.79	32.66

Table A.6: The performance of four algorithms (ERM, IRM, REx, IGA) on the **mechanism shift** data from the **Mechanical MNIST - EMNIST Letters Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2	
Statistics of Strain Energy		y_mean = 703.01 y_std = 86.26		y_mean = 706.98 y_std = 88.45		y_mean = 629.56 y_std = 65.91		y_mean = 567.98 y_std = 48.67	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	9.72	12.40	36.21	43.34	206.02	207.21	163.66	165.31
MLP	IRM	8.70	13.35	32.72	44.84	174.03	175.42	149.32	153.51
MLP	REx	8.94	12.70	27.67	36.51	151.76	153.21	131.10	134.32
MLP	IGA	6.01	8.89	26.68	33.14	133.05	134.75	139.39	140.81
LeNet	ERM	12.33	16.89	28.47	39.44	215.98	217.78	222.95	224.47
LeNet	IRM	11.32	20.10	23.84	39.84	140.05	147.29	131.84	141.74
LeNet	REx	9.41	16.34	19.45	30.85	100.36	111.68	98.68	114.34
LeNet	IGA	6.54	12.60	17.47	30.45	57.96	115.96	31.49	161.78

Table A.7: The performance of four algorithms (ERM, IRM, REx, IGA) on the **sampling bias** data from the **Mechanical MNIST Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2		Test-3	
Statistics of Strain Energy		y_mean = 545.99 y_std = 42.55		y_mean = 545.99 y_std = 41.23		y_mean = 565.68 y_std = 22.45		y_mean = 564.82 y_std = 22.27		y_mean = 567.02 y_std = 47.57	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	4.99	5.16	11.96	13.36	29.54	30.38	26.63	27.53	29.21	30.05
MLP	IRM	5.20	5.28	11.05	12.56	29.00	29.92	25.51	26.51	28.08	29.05
MLP	REx	3.55	5.00	7.77	9.39	25.85	26.90	21.29	22.38	22.99	24.17
MLP	IGA	3.58	4.77	7.15	9.59	16.25	20.39	15.91	19.05	18.24	21.31
LeNet	ERM	3.90	4.59	5.41	6.10	10.97	11.42	10.89	11.33	13.22	13.60
LeNet	IRM	4.65	5.30	5.74	6.36	10.96	11.35	10.92	11.32	13.02	13.37
LeNet	REx	3.89	4.56	5.30	5.91	11.17	11.54	11.06	11.42	13.53	13.85
LeNet	IGA	5.49	6.73	6.21	7.41	10.28	11.16	10.08	10.99	11.74	12.59

Table A.8: The performance of four algorithms (ERM, IRM, REx, IGA) on the **sampling bias data** from the **Mechanical MNIST - EMNIST Letters Collection**. The RMSE on each dataset is calculated using both eqn. 3.10 and 3.11 based on 15 predictions given by the corresponding ML model trained with 15 different initialization. In addition, the mean and standard deviation for the change in strain energy from the train, validation, and test datasets are also given.

Data		Train		Valid		Test-1		Test-2		Test-3	
Statistics of Strain Energy		y_mean = 686.22 y_std = 80.05		y_mean = 686.08 y_std = 80.41		y_mean = 704.81 y_std = 37.24		y_mean = 705.21 y_std = 37.75		y_mean = 713.32 y_std = 87.12	
Evaluation Method		RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)	RMSE (eqn. 3.10)	RMSE (eqn. 3.11)
MLP	ERM	3.25	3.99	21.05	23.21	36.48	38.01	36.69	38.24	42.42	43.70
MLP	IRM	8.58	10.83	17.40	20.15	30.49	32.43	30.78	32.77	37.05	38.83
MLP	REx	5.03	6.51	15.39	17.65	27.73	29.35	27.83	29.56	34.42	35.85
MLP	IGA	9.92	12.59	15.42	18.24	26.91	28.88	27.08	29.46	33.28	35.40
LeNet	ERM	10.32	12.26	13.68	15.64	21.84	23.24	22.25	23.67	28.27	29.43
LeNet	IRM	9.91	11.51	13.48	15.03	22.34	23.41	22.59	23.67	28.89	29.72
LeNet	REx	10.03	11.82	13.36	14.96	22.46	23.54	22.69	23.79	28.76	29.67
LeNet	IGA	12.10	13.70	14.74	16.28	21.69	22.91	21.98	23.17	27.91	28.85

ML Model Prediction vs. Ground Truth Visualization

Here an additional supplement is provided for Fig. 3·4-3·6 from Section 3.4. Fig. A·4-A·5 visualize example comparisons between the ground truth and the aggregated mean prediction of the change in strain energy through implementing OOD generalization algorithms (IRM, REx, IGA) and ERM on LeNet model and predicting on test environment 1 from each OOD dataset drawn from Mechanical MNIST and Mechanical MNIST - EMNIST Letters. In addition, the RMSE between the aggregated mean prediction and the ground truth for each method is shown in the legend of each plot. These values of RMSE can also be found in Table A.3-A.8. As shown in Fig. A·4ab and Fig. A·5ab, the prediction given by ERM underestimates the ground truth for the test data from the covariate and mechanism shift datasets. While the prediction given by OOD generalization methods (IRM, REx and IGA) sometimes also underestimates the ground truth, it is closer to the ground truth (i.e., the change in strain energy calculated by FEM) and sometimes has no underestimation (e.g., prediction given by REx for covariate shift dataset from Mechanical MNIST dataset). For the results from the sampling bias datasets shown in Fig. A·4c and Fig. A·5c, the prediction accuracy on the test environment is similar for ERM and OOD generalization methods, which is consistent with the discussion in Section 3.4.3.

Best and Worst Cases Pattern Visualization

To further support the results shown in Fig. 3·4-3·6 from Section 3.4 and the prediction vs. ground truth results shown in Fig. A·4-A·5. Here the visualizations of representative samples from the Mechanical MNIST dataset that lead to different error levels are presented. Fig. A·6, which contains visualizations of the data shown in Fig. A·4, presents randomly selected samples from test environment 1 for each OOD dataset from Mechanical MNIST. Specifically, for each algorithm (ERM, IRM, REx and IGA) on each test environment 1 from covariate shift (Fig. A·6a), mechanism shift (Fig. A·6b) and sampling bias dataset (Fig. A·6c),

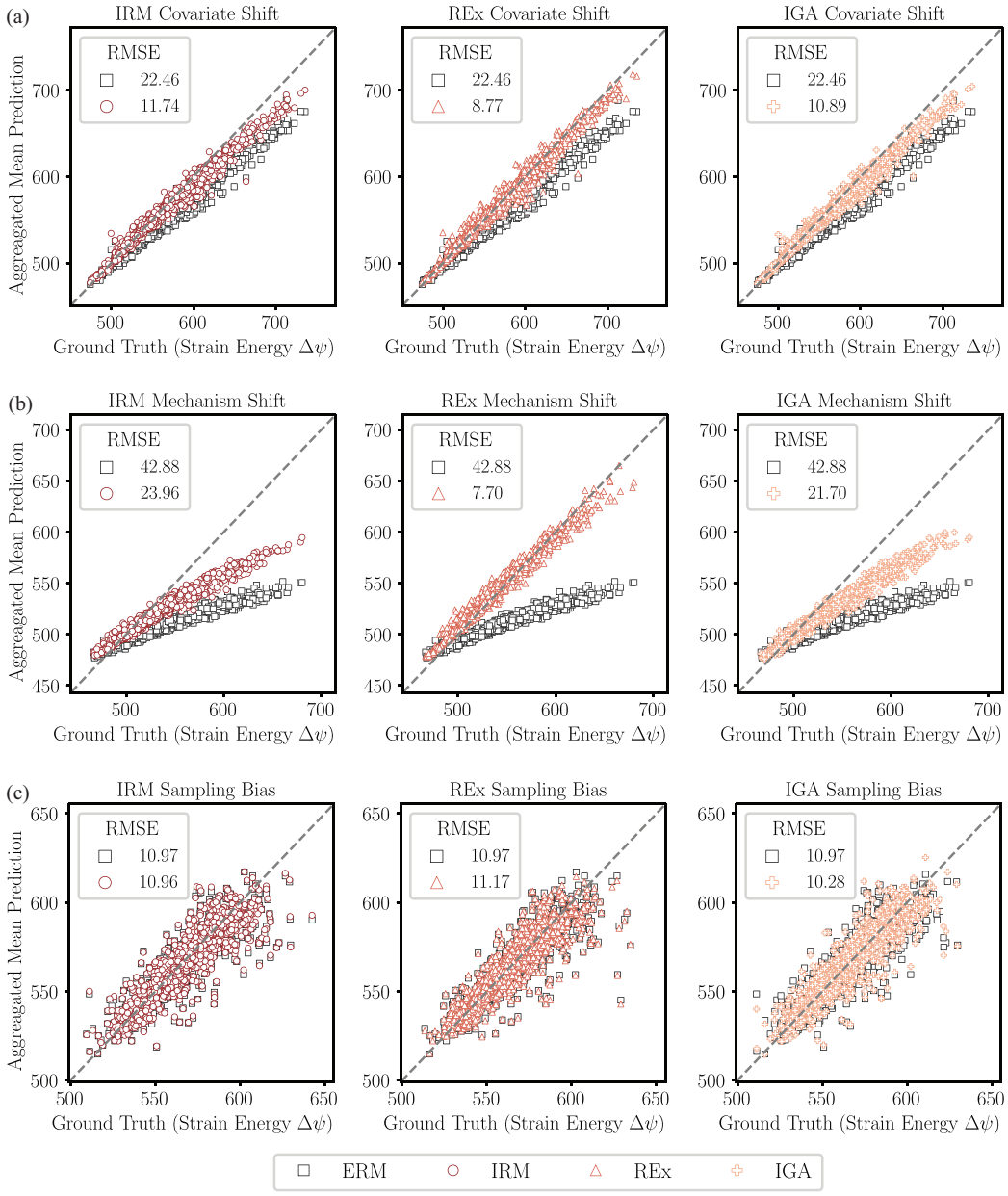


Figure A·4: The aggregated mean prediction versus the ground truth of the change in strain energy for OOD generalization algorithms (IRM, REx, IGA) and ERM on the test environment 1 of each OOD dataset on Mechanical MNIST.

examples at different levels of RMSE are shown. To achieve this, sorted the data based on RMSE and sampled three cases from three groups: lowest error (bottom 10%), median error ($45\% - 55\%$), and highest error (top 10%). For each example, the figure reports the

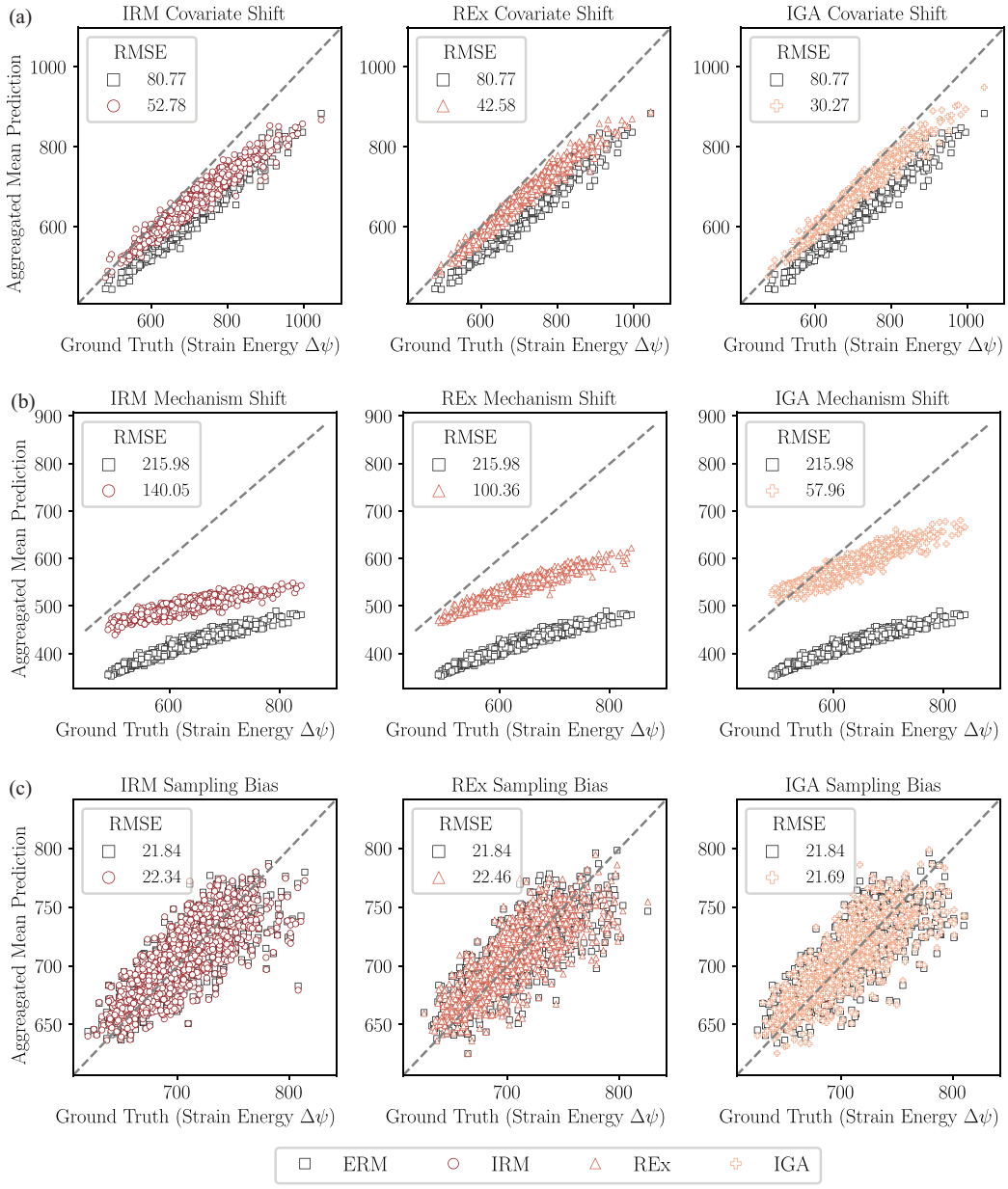


Figure A·5: The aggregated mean prediction versus the ground truth of the change in strain energy for OOD generalization algorithms (IRM, REx, IGA) and ERM on the test environment 1 of each OOD dataset on Mechanical MNIST - EMNIST Letters.

ground truth of the change in strain energy, the predicted change in strain energy, and the corresponding RMSE calculated through eqn. 3.10. Note that for the covariate shift and mechanism shift dataset, both the best (lowest error) and worst (highest error) cases of ERM

lead to higher RMSE than other three OOD algorithms. The sample cases for Mechanical MNIST - EMNIST Letters are presented on Fig. A·7 with an identical format.

A.3 Sensitivity Analysis for Finite Element Simulation

The properties of the chiral metamaterial were acquired through the Finite Element Method (FEM). This section provides detailed information on the FEM settings. The two rigid circles are set as discrete rigid surfaces, while the elastic ligament of each chiral structure is modeled as linear elastic Bernoulli–Euler beam. The cross section of the ligament is rectangular with a fixed width of 30 and a fixed thickness 1.5. The ligament material is linear elastic with a fixed Elastic modulus $E = 70$ and Poisson ratio $\nu = 0.3$. The left circle is fixed, while the right rigid circle is only allowed to move under the applied displacement load with the same magnitude 0.08. Both the circle and the ligament are meshed in a graded fashion, where the area closer to the connecting region has a finer mesh compared to areas farther away. The finest mesh size is set as 0.02 and the coarsest mesh size is 20 times the finest mesh size. On average, there are approximately 270 mesh elements on each of the rigid circles and approximately 714 mesh elements on the elastic ligament, as calculated from 12 randomly generated chiral structure finite element models. Fig. A·8, exported from the software ABAQUS, illustrates the distribution of mesh size on the rigid circles and the elastic ligament for a representative example. Each circle has 254 R2D2 (two node 2D linear discrete rigid element) elements and the ligament has 920 B23 (two node cubic Euler-Bernoulli beam element) elements (AbaqusAnalysis, 2016). The mesh size at the ligament-circle contact regions is smaller than other regions. The finest mesh size 0.02 was determined by conducting mesh sensitivity analysis. Fig. A·9 demonstrates how stiffness values change depending on the mesh size. Choosing an appropriate mesh size involves balancing computational cost and FEM accuracy. It is worth noting that excessively fine mesh sizes, such as 0.01, can lead to more simulation failures due to convergence

Figure A·6: The sampled cases of three groups: lowest error (bottom 10% RMSE), median error (45% – 55% RMSE) and highest error (top 10% RMSE) from the Mechanical MNIST, and the corresponding ground truth and predicted change in strain energy given by LeNet.

Figure A·7: The sampled cases of three groups: lowest error (bottom 10% RMSE), median error (45% – 55% RMSE) and highest error (top 10% RMSE) from the Mechanical MNIST - EMNIST Letters, and the corresponding ground truth and predicted change in strain energy given by LeNet.

difficulties. Therefore, a mesh size of 0.02, where simulation results become converged and nearly independent of mesh size, was adopted for all chiral metamaterial simulations in this work. Nonetheless, note that a few simulation cases still fail to converge for certain chiral structures with a mesh size of 0.02. These failed examples were disregarded during the Bayesian Optimization data acquisition process. For more details on the finite element modeling, readers can refer to the open-source code for this work, as detailed in Section 6.3.

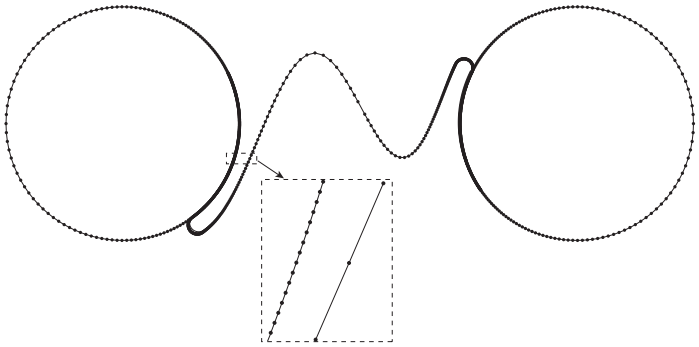


Figure A·8: Visualization of the mesh on a representative chiral structure. The finest mesh size is 0.02 and the coarsest mesh size is 0.4. Graded meshing was performed using the commercial software ABAQUS.

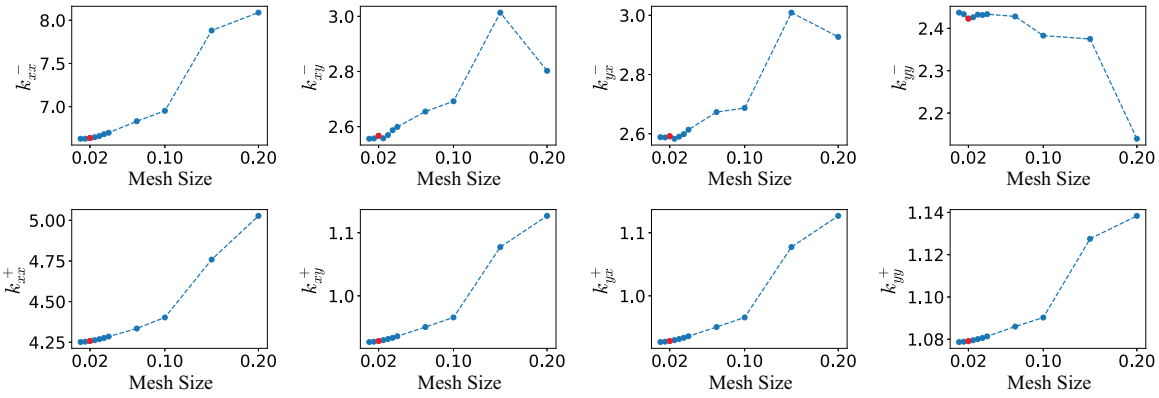


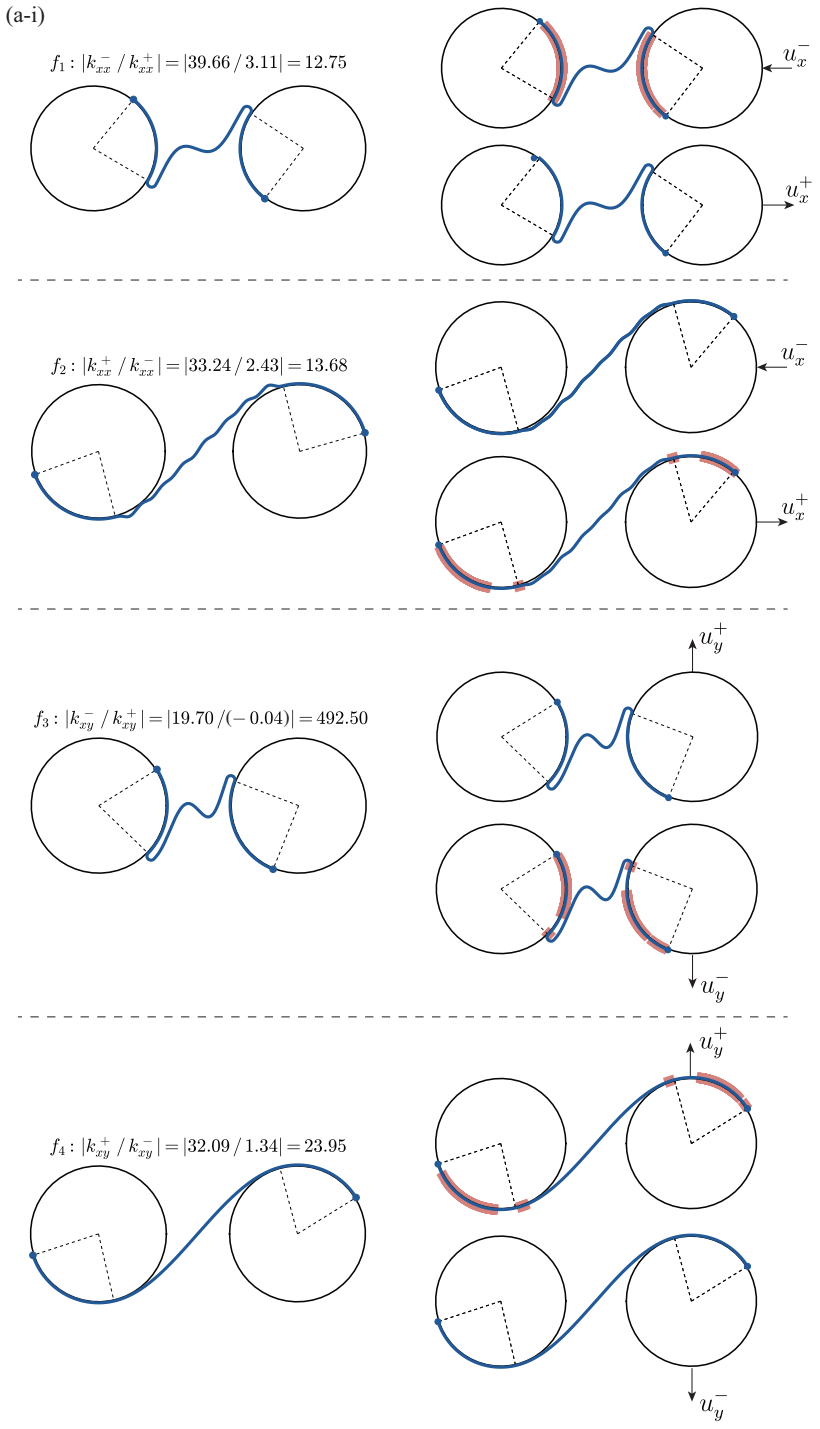
Figure A·9: The relationship between calculated stiffness values and the selected finest mesh size for the chiral structure depicted in Fig. A·8. The final adopted finest mesh size is 0.02, highlighted in red.

A.4 Contact Modes for Optimal Designs

In Section 5.3.1 and Section 5.3.2, the optimal designs for the eight objectives of non-reciprocity and eight objectives of asymmetry are presented. Here, this section presents the contact status for each design under different loads. Specifically, Fig. A·10 shows the optimal chiral structures and the corresponding contact status for objectives f_1 to f_8, Fig. A·11 shows the optimal chiral structures and the corresponding contact status for objectives g_1 to g_8.

A.5 Data Distribution for Multi-objectives

In Section 4.4.3, Fig. 4·5 illustrated the trade-off between asymmetry and non-reciprocity in multi-objective optimization, where the aim is to optimize two objectives with values exceeding 1 while maintaining good performance for both. With eight non-reciprocity and eight asymmetry objectives defined in Sections 4.4.1 and Section 4.4.2 respectively, there are 64 comprehensive combinations of non-reciprocity and asymmetry multi-objective optimization. In Section 5.3.3, the multi-objective optimization was categorized into three cases: non-contradictory, contradictory, and challenging. To provide insight into this categorization, 500 chiral structures obtained in the single-objective optimization process were randomly sampled, and their distribution of non-reciprocity and asymmetry properties is plotted in Fig. A·12-A·14, where the red dashed lines represent thresholds of $x = 1$ and $y = 1$ for valid designs. Any design having a non-reciprocity or asymmetry property below 1 was not considered further. Fig. A·12 demonstrates 16 pairs of non-reciprocity and asymmetry where the properties are not contradictory, allowing for simultaneous improvement of both. Conversely, Fig. A·13 exhibits 16 contradictory pairs, where high performance in one property corresponds to low performance in the other. Fig. A·14 illustrates the data distribution of 24 pairs of multi-objectives, introducing a challenging asymmetry property that makes it difficult to achieve good performance in non-reciprocity


(a-i)
$f_1 : |k_{xx}^- / k_{xx}^+| = |39.66 / 3.11| = 12.75$
u_x^-
u_x^+
$f_2 : |k_{xx}^+ / k_{xx}^-| = |33.24 / 2.43| = 13.68$
u_x^-
u_x^+
$f_3 : |k_{xy}^- / k_{xy}^+| = |19.70 / (-0.04)| = 492.50$
u_y^+
u_y^-
$f_4 : |k_{xy}^+ / k_{xy}^-| = |32.09 / 1.34| = 23.95$
u_y^+
u_y^-

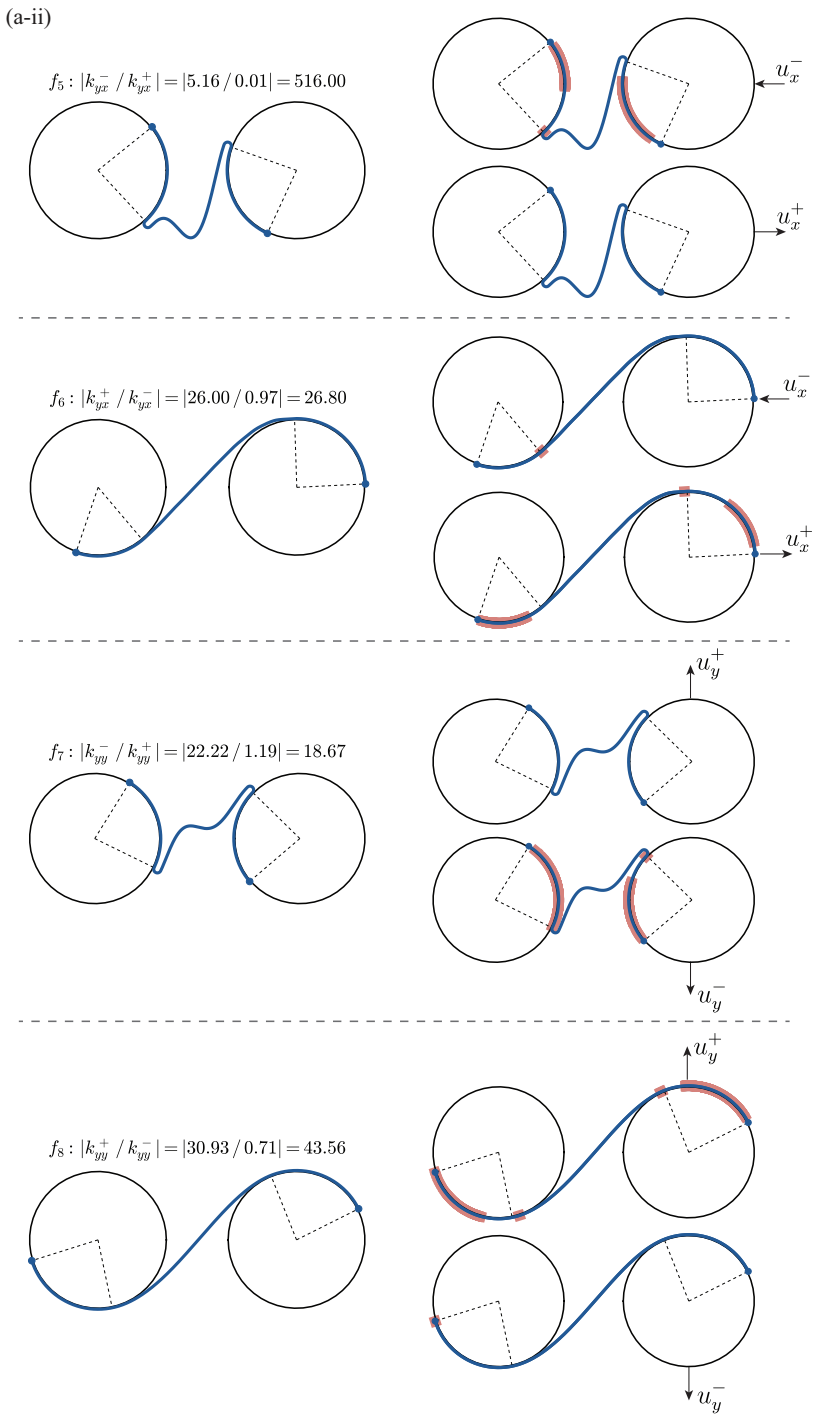
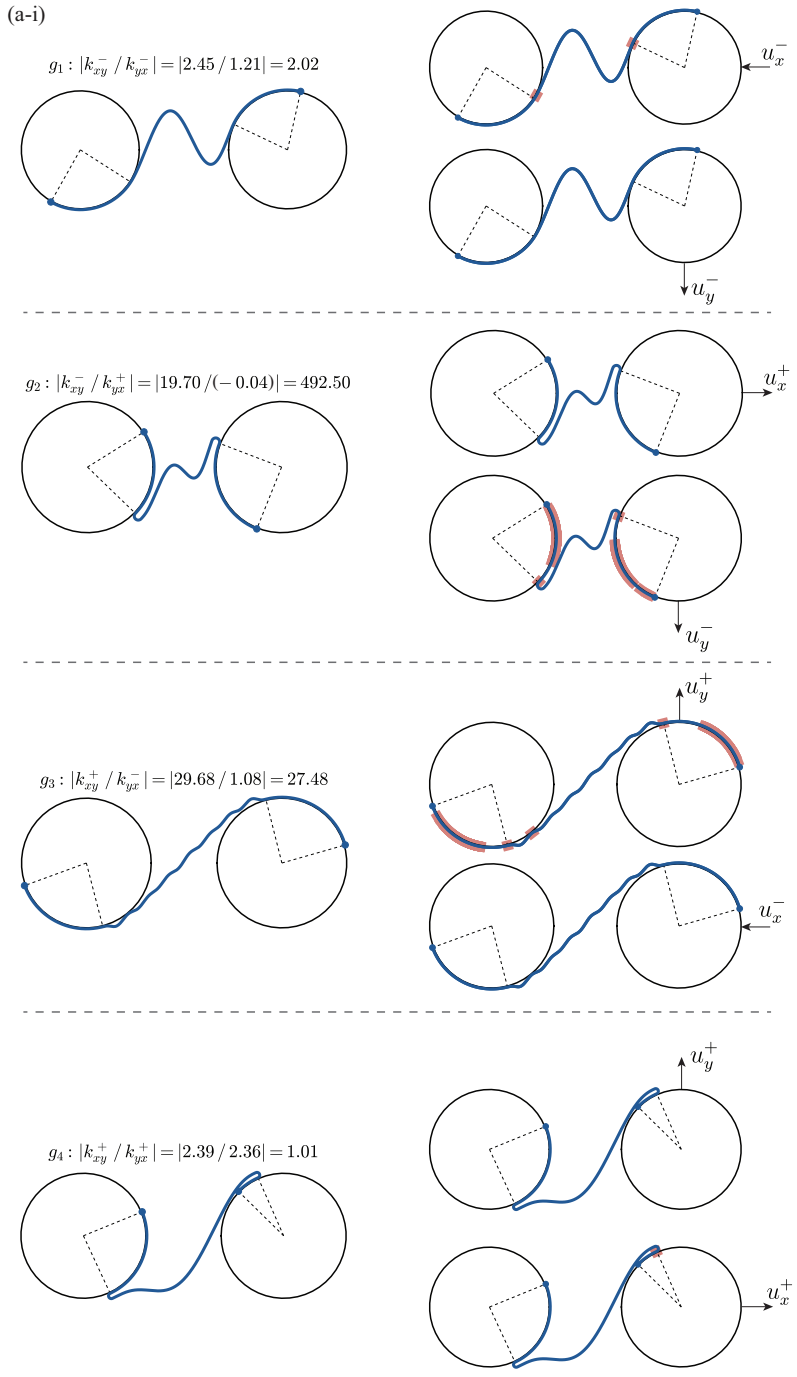


Figure A·10: The contact modes for the optimal designs of the eight non-reciprocity objectives (a-i) f_1, f_2, f_3, f_4 and (a-ii) f_5, f_6, f_7, f_8. For each objective, the optimal design and the contact modes during the loading in which the stiffness values are obtained are depicted.

.


(a-i)
$g_1 : |k_{xy}^{-} / k_{yx}^{-}| = |2.45 / 1.21| = 2.02$
u_x^{-}
u_y^{-}
$g_2 : |k_{xy}^{-} / k_{yx}^{+}| = |19.70 / (-0.04)| = 492.50$
u_x^{+}
u_y^{-}
$g_3 : |k_{xy}^{+} / k_{yx}^{-}| = |29.68 / 1.08| = 27.48$
u_y^{+}
u_x^{-}
$g_4 : |k_{xy}^{+} / k_{yx}^{+}| = |2.39 / 2.36| = 1.01$
u_y^{+}
u_x^{+}

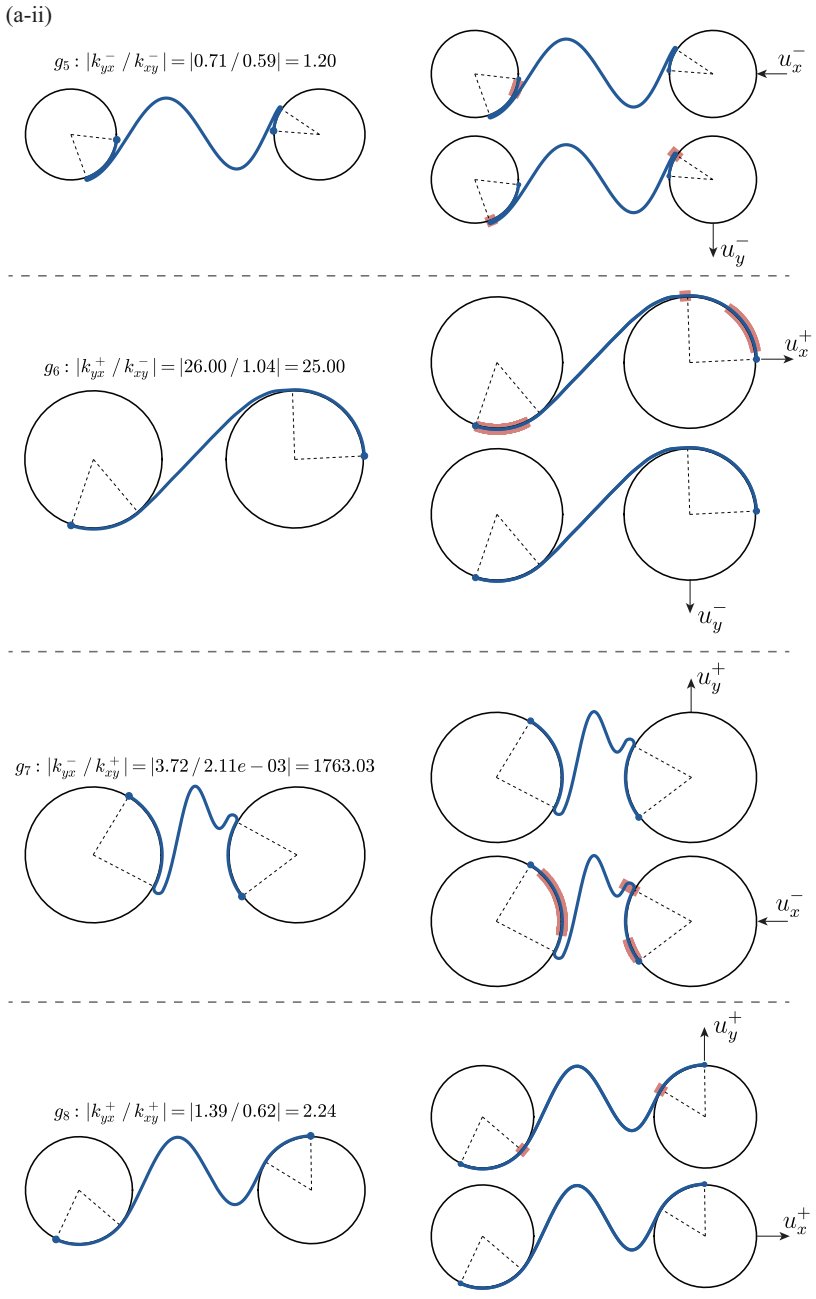


Figure A·11: The contact modes for the optimal designs of the eight asymmetry objectives (a-i) g_1, g_2, g_3, g_4 and (a-ii) g_5, g_6, g_7, g_8. For each objective, the optimal design and the contact modes during the loading in which the stiffness values are obtained are depicted.

.

simultaneously. Notably, the non-reciprocity of all materials generally falls below 3. Note that the the total number of pairs displayed in Fig. A·12-A·14 is less than 64 due to the exclusion of objective $g_4 = k_{xy}^{+}/k_{yx}^{+}$, for which no optimal design was found during the single-objective optimization process, as discussed in Section 5.3.2 and Section 5.3.3.

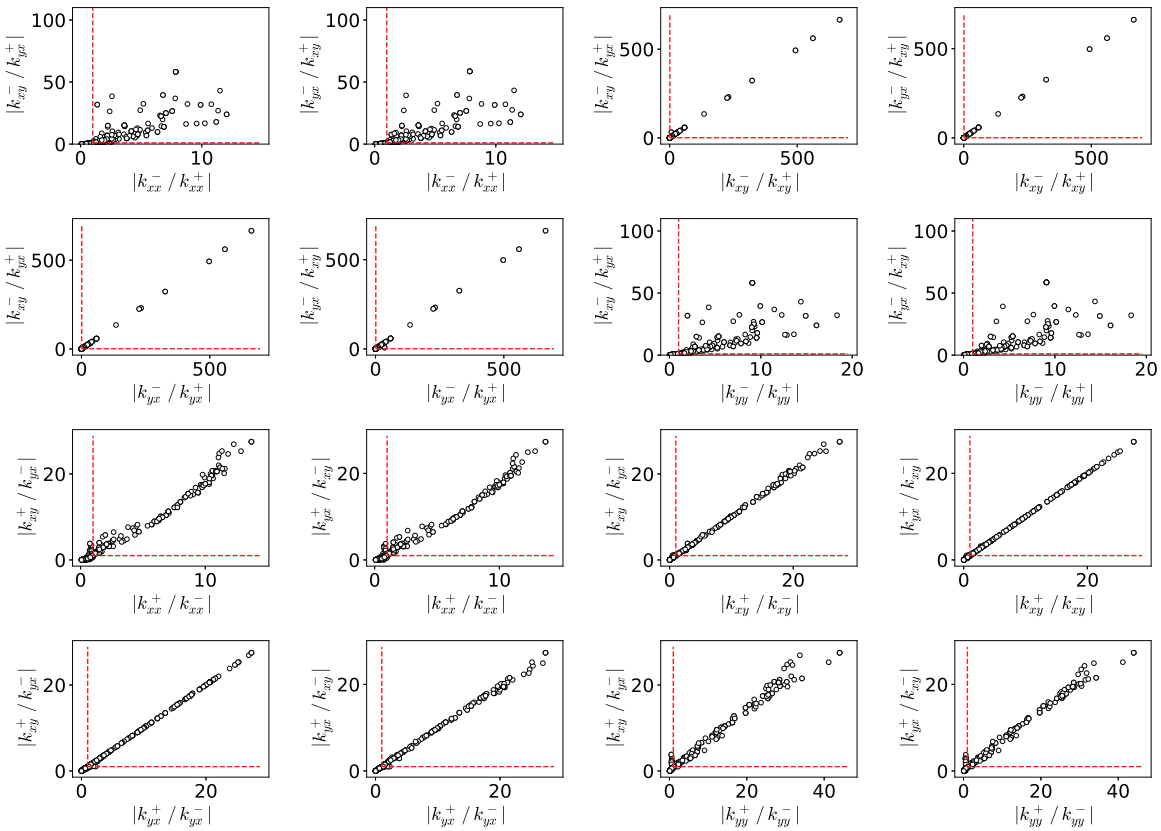


Figure A·12: The data distribution of 16 pairs of non-contradictory multi-objectives for non-reciprocity and asymmetry multi-objectives. The red dashed lines indicate thresholds of $x = 1$ and $y = 1$ for valid designs. The x-axis denotes one non-reciprocity objective and the y-axis denotes one asymmetry objective, as defined in Sections 4.4.1 and Section 4.4.2 respectively.

A.6 Pareto Front for Multi-objective Optimization

Section 5.3.3 investigated three Multi-Objectives and discussed about the Pareto front for each. Although there can be more than one Pareto Optimum for each multi-objective, only

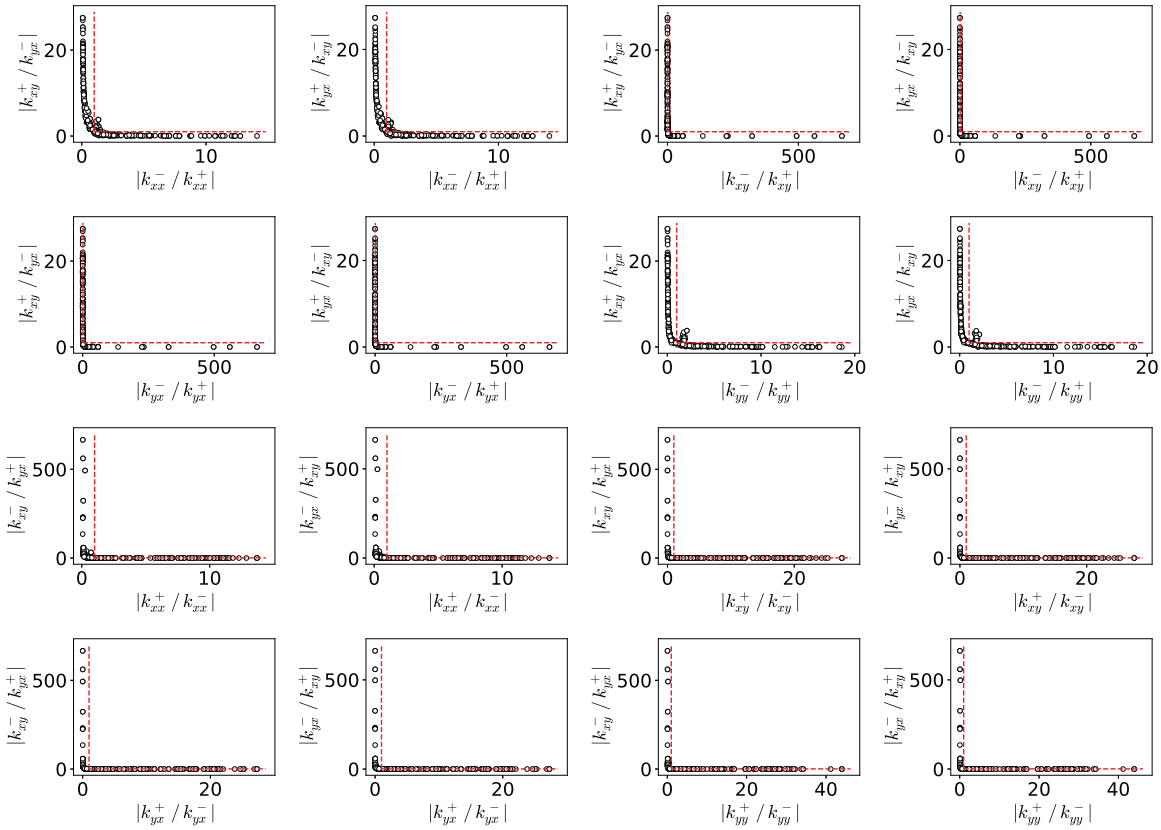


Figure A·13: The data distribution of 16 pairs of contradictory multi-objectives for non-reciprocity and asymmetry multi-objectives. The red dashed lines indicate thresholds of $x = 1$ and $y = 1$ for valid designs. The x-axis denotes one non-reciprocity objective and the y-axis denotes one asymmetry objective, as defined in Sections 4.4.1 and Section 4.4.2 respectively.

the optimal design with the largest modulus was analyzed specifically. Here Fig. A·15 presents all the Pareto optima for multi-objectives optimization. From the figure it is evident that the optimal designs for the same multi-objectives share common features, i.e., the shape of the ligament and the connecting area of the circles and ligament appear similar.

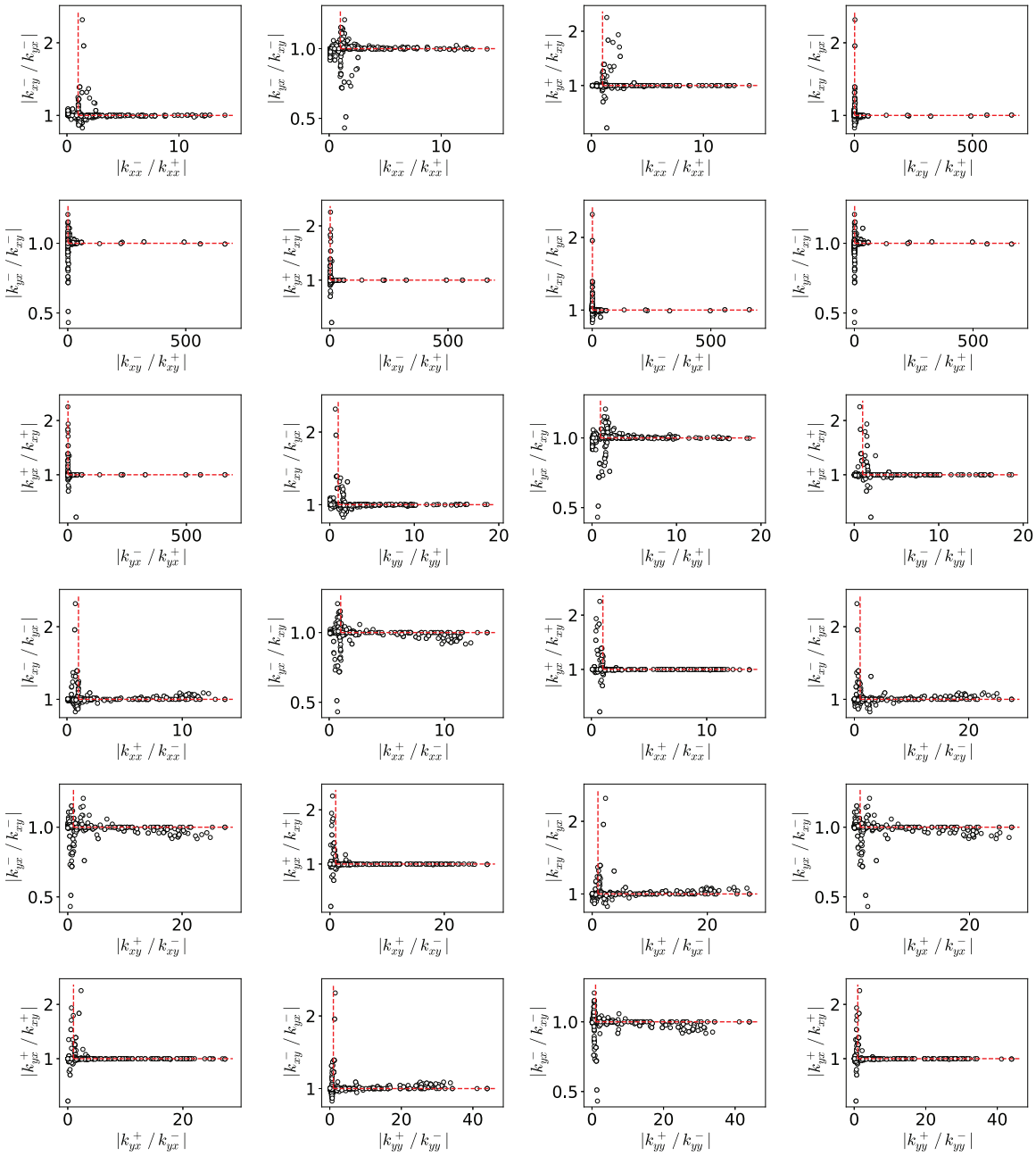


Figure A·14: The data distribution of 24 pairs of challenging multi-objectives for non-reciprocity and asymmetry multi-objectives. The red dashed lines indicate thresholds of $x = 1$ and $y = 1$ for valid designs. The x-axis denotes one non-reciprocity objective and the y-axis denotes one asymmetry objective, which is the objective that is challenging to optimize. The definition of non-reciprocity and asymmetry can be found in Sections 4.4.1 and Section 4.4.2.

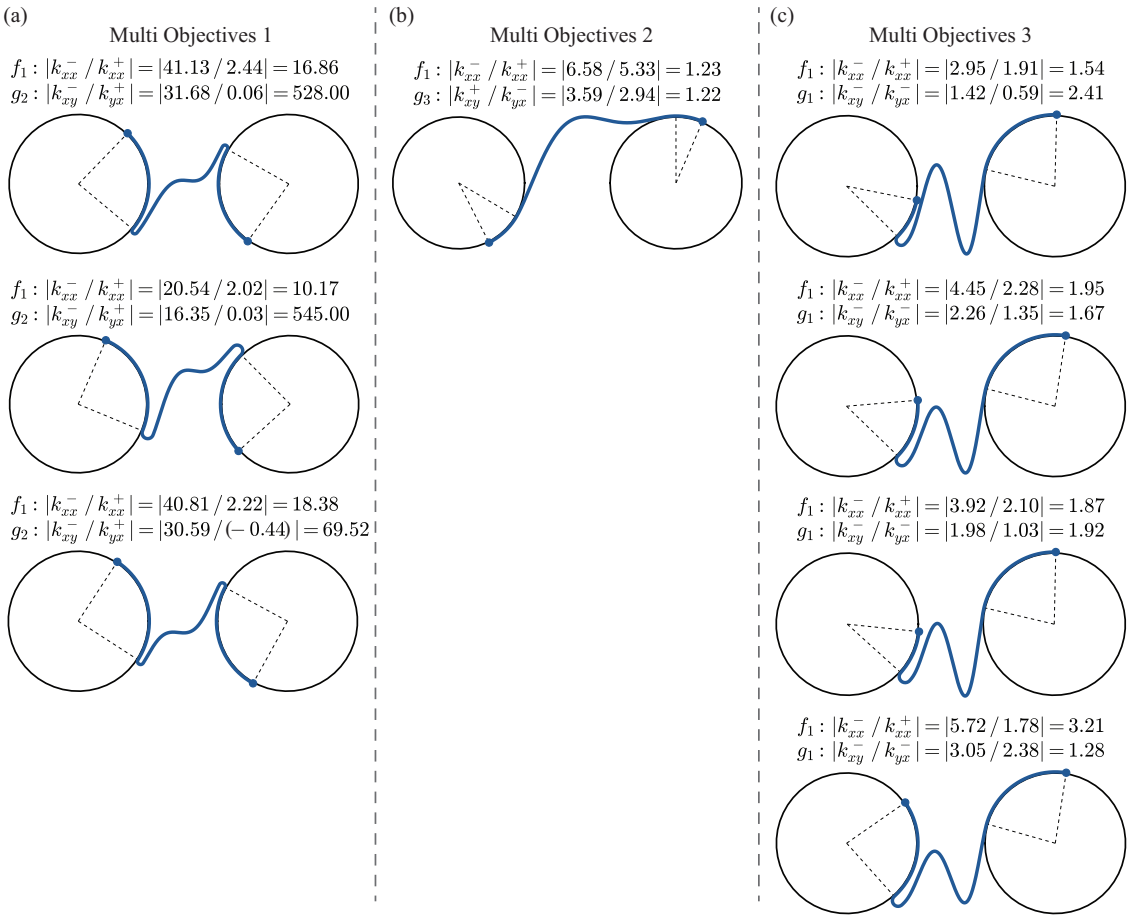


Figure A·15: Optimal designs for the three multi-objectives outlined In Section 5.3.3. (a) Three Pareto optimal designs for Multi-Objective 1, discovered from design space 2. (b) One Pareto optimal design for Multi-Objective 2, discovered from design space 1. (c) Four Pareto optimal designs for Multi-Objective 3, discovered from design space 4.

References

AbaqusAnalysis (2016). Abaqus analysis user's guide. http://130.149.89.49:2080/v2016/books/usb/default.htm.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297.

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. (2020). Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR.

Alber, M., Buganza Tepole, A., Cannon, W. R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W. W., Perdikaris, P., Petzold, L., et al. (2019). Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ digital medicine*, 2(1):1–11.

An, S., Lee, M., Park, S., Yang, H., and So, J. (2020). An ensemble of simple convolutional neural network models for mnist digit recognition. *arXiv preprint arXiv:2008.10400*.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Arzani, A., Yuan, L., Newell, P., and Wang, B. (2023). Interpreting and generalizing deep learning in physics-based problems with functional linear models. *arXiv preprint arXiv:2307.04569*.

Attarzadeh, M. A., Callanan, J., and Nouh, M. (2020). Experimental observation of nonreciprocal waves in a resonant metamaterial beam. *Physical Review Applied*, 13:021001.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473.

Berger, C., Paschali, M., Glocker, B., and Kamnitsas, K. (2021). Confidence-based out-of-distribution detection: a comparative study and analysis. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*, pages 122–132. Springer.

Bian, J. and Nicola, L. (2021). On the lubrication of rough copper surfaces with graphene. *Tribology International*, 156:106837.

Binois, M. and Wycoff, N. (2022). A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R., and Klusemann, B. (2019). A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, 6:110.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937.

Buehler, M. J. (2023a). Generative pretrained autoregressive transformer graph neural network applied to the analysis and discovery of novel proteins. *Journal of Applied Physics*, 134(8).

Buehler, M. J. (2023b). Melm, a generative pretrained language modeling framework that solves forward and inverse mechanics problems. *Journal of the Mechanics and Physics of Solids*, 181:105454.

Buehler, M. J. (2024). Mechgpt, a language-based strategy for mechanics and materials modeling that connects knowledge across scales, disciplines, and modalities. *Applied Mechanics Reviews*, 76(2):021001.

Bühlmann, P. (2020). Invariance, causality and robustness. *Statistical Science*, 35(3):404–426.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., and Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6).

Carpentier, A., Lazaric, A., Ghavamzadeh, M., Munos, R., and Auer, P. (2011). Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *International Conference on Algorithmic Learning Theory*, pages 189–203. Springer.

Challapalli, A., Patel, D., and Li, G. (2021). Inverse machine learning framework for optimizing lightweight metamaterials. *Materials & Design*, 208:109937.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. (2020). Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR.

Chen, C.-T. and Gu, G. X. (2020). Generative deep neural networks for inverse materials design using backpropagation and active learning. *Advanced Science*, 7(5):1902607.

Chen, C.-T. and Gu, G. X. (2021). Learning hidden elasticity with deep neural networks. *Proceedings of the National Academy of Sciences*, 118(31):e2102721118.

Chen, G., Li, T., Chen, Q., Ren, S., Wang, C., and Li, S. (2019). Application of deep learning neural network to identify collision load conditions based on permanent plastic deformation of shell structures. *Computational Mechanics*, 64(2):435–449.

Chen, Q., Xie, Y., Ao, Y., Li, T., Chen, G., Ren, S., Wang, C., and Li, S. (2021a). A deep neural network inverse solution to recover pre-crash impact data of car collisions. *Transportation research part C: emerging technologies*, 126:103009.

Chen, Y., Li, X., Scheibner, C., Vitelli, V., and Huang, G. (2021b). Realization of active metamaterials with odd micropolar elasticity. *Nature Communications*, 12:5935.

Chen, Y. J., Scarpa, F., Liu, Y. J., and Leng, J. S. (2013). Elasticity of anti-tetrachiral anisotropic lattices. *International Journal of Solids and Structures*, 50:996–1004.

Cireşan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE.

Cohen, T. S., Geiger, M., and Weiler, M. (2019). A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32.

Coulais, C., Sounas, D., and Alu, A. (2017). Static non-reciprocity in mechanical metamaterials. *Nature*, 542:461–464.

Cox, D. D. and John, S. (1992). A statistical method for global optimization. In *[Proceedings] 1992 IEEE international conference on systems, man, and cybernetics*, pages 1241–1246. IEEE.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020). Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.

Creager, E., Jacobsen, J.-H., and Zemel, R. (2021). Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR.

Csurka, G. (2017). In Csurka, G., editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 1–35. Springer, Cham.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *arXiv preprint arXiv:2201.05624*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE.

Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., and Csányi, G. (2021). Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141.

DeVries, T. and Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Duchi, J. and Namkoong, H. (2018). Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*.

Fang, L., Mojahed, A., Darabi, A., Vakakis, A. F., and Leamy, M. J. (2021). Passive nonreciprocity in a system of asymmetrical rotational oscillators. *Physical Review Applied*, 15:014005.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2021). A brief review of domain adaptation. In Stahlbock, R., Weiss, G. M., Abou-Nasr, M., Yang, C.-Y., Arabnia, H. R., and Deligiannidis, L., editors, *Advances in Data Science and Information Engineering*, pages 877–894. Springer, Cham.

Fernandez-Corbaton, I., Rockstuhl, C., Ziemke, P., Gumbsch, P., Albiez, A., Schwaiger, R., Frenzel, T., Kadic, M., and Wegener, M. (2019). New twists of 3d chiral metamaterials. *Advanced Materials*, 31:1807742.

Forte, A. E., Hanakata, P. Z., Jin, L., Zari, E., Zareei, A., Fernandes, M. C., Sumner, L., Alvarez, J., and Bertoldi, K. (2022). Inverse design of inflatable soft membranes through machine learning. *Advanced Functional Materials*, 32(16):2111610.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. (2018). An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*.

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. (2020). Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981.

Fuhg, J. N., van Wees, L., Obstalecki, M., Shade, P., Bouklas, N., and Kasemer, M. (2022). Machine-learning convex and texture-dependent macroscopic yield from crystal plasticity simulations. *Materialia*, 23:101446.

Fung, Y. C. (2013). *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media.

Ganaie, M. A., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Goldsberry, B. M., Wallen, S. P., and Haberman, M. R. (2019). Non-reciprocal wave propagation in mechanically-modulated continuous elastic metamaterials. *Journal of the Acoustic Society of America*, 146:782–788.

Gongora, A. E., Xu, B., Perry, W., Okoye, C., Riley, P., Reyes, K. G., Morgan, E. F., and Brown, K. A. (2020). A bayesian experimental autonomous researcher for mechanical design. *Science advances*, 6(15):eaaz1708.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goswami, S., Kontolati, K., Shields, M. D., and Karniadakis, G. E. (2022). Deep transfer learning for partial differential equations under conditional shift with deeponet. *arXiv preprint arXiv:2204.09810*.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. *Advances in neural information processing systems*, 32.

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

Guo, K., Yang, Z., Yu, C.-H., and Buehler, M. J. (2021). Artificial intelligence and machine learning in design of mechanical materials. *Materials Horizons*, 8(4):1153–1172.

Guo, R., Cheng, L., Li, J., Hahn, P. R., and Liu, H. (2020). A survey of learning causality with data: Problems and methods. *ACM Computing Surveys (CSUR)*, 53(4):1–37.

Ha, C. S., Yao, D., Xu, Z., Liu, C., Liu, H., Elkins, D., Kile, M., Deshpande, V., Kong, Z., Bauchy, M., et al. (2023). Rapid inverse design of metamaterials based on prescribed mechanical behavior through machine learning. *Nature Communications*, 14(1):5765.

Hadash, G., Kermany, E., Carmeli, B., Lavi, O., Kour, G., and Jacovi, A. (2018). Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*.

Hanakata, P. Z., Cubuk, E. D., Campbell, D. K., and Park, H. S. (2020). Forward and inverse design of kirigami via supervised autoencoder. *Physical Review Research*, 2(4):042006.

Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6).

Henning, C., D'Angelo, F., and Grewe, B. F. (2021). Are bayesian neural networks intrinsically good at out-of-distribution detection? *arXiv preprint arXiv:2107.12248*.

Hu, R., Sang, J., Wang, J., and Jiang, C. (2021). Understanding and testing generalization of deep networks on out-of-distribution data. *arXiv preprint arXiv:2111.09190*.

Huang, J., Liew, J., Ademiloye, A., and Liew, K. M. (2021). Artificial intelligence in materials modeling and design. *Archives of Computational Methods in Engineering*, 28(5):3399–3413.

Izmailov, P., Nicholson, P., Lotfi, S., and Wilson, A. G. (2021). Dangers of bayesian model averaging under covariate shift. *Advances in Neural Information Processing Systems*, 34.

Jiao, Q., Chen, Y., Kim, J.-h., Han, C.-F., Chang, C.-H., and Vlassak, J. J. (2024). A machine learning perspective on the inverse indentation problem: uniqueness, surrogate modeling, and learning elasto-plastic properties from pile-up. *Journal of the Mechanics and Physics of Solids*, 185:105557.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492.

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., and Kumar, V. (2017). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331.

Kim, Y., Gu, G. H., Asghari-Rad, P., Noh, J., Rho, J., Seo, M. H., and Kim, H. S. (2022). Novel deep learning approach for practical applications of indentation. *Materials Today Advances*, 13:100207.

Kim, Y., Kim, Y., Yang, C., Park, K., Gu, G. X., and Ryu, S. (2021). Deep learning framework for material design space exploration using active transfer learning and data augmentation. *npj Computational Materials*, 7(1):1–7.

Kobeissi, H., Mohammadzadeh, S., and Lejeune, E. (2022). Enhancing mechanical meta-models with a generative model-based augmented training dataset. *arXiv preprint arXiv:2203.04183*.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Kollmann, H. T., Abueidda, D. W., Koric, S., Guleryuz, E., and Sobh, N. A. (2020). Deep learning for topology optimization of 2d metamaterials. *Materials & Design*, 196:109098.

Kotthoff, L., Wahab, H., and Johnson, P. (2021). Bayesian optimization in materials science: a survey. *arXiv preprint arXiv:2108.00002*.

Koyama, M. and Yamaguchi, S. (2020). Out-of-distribution generalization with maximal invariant predictor. https://www.researchgate.net/publication/343470695_Out-of-Distribution_Generalization_with_Maximal_Invariant_Predictor.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR.

Kuang, K., Xiong, R., Cui, P., Athey, S., and Li, B. (2020). Stable prediction with model misspecification and agnostic distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4485–4492.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Kussul, E. and Baidyk, T. (2004). Improved method of handwritten digit recognition tested on mnist database. *Image and Vision Computing*, 22(12):971–981.

Kuznetsova, M. S., Pasternak, E., and Dyskin, A. V. (2017). Analysis of wave propagation in a discrete chain of bilinear oscillators. *Nonlinear Processes in Geophysics*, 24:455–460.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, D., Chen, W., Wang, L., Chan, Y.-C., and Chen, W. (2023). Data-driven design for metamaterials and multiscale systems: A review. *Advanced Materials*, 36(8):2305254.

Lee, J.-H., Park, H. S., and Holmes, D. P. (2021). Elastic instabilities govern the morphogenesis of the optic cup. *Physical Review Letters*, 127(13):138102.

Lejeune, E. (2020). Mechanical mnist: A benchmark dataset for mechanical metamodels. *Extreme Mechanics Letters*, 36:100659.

Lejeune, E. and Zhao, B. (2021). Exploring the potential of transfer learning for metamodels of heterogeneous material deformation. *Journal of the Mechanical Behavior of Biomedical Materials*, 117:104276.

Li, B., Deng, B., Shou, W., Oh, T.-H., Hu, Y., Luo, Y., Shi, L., and Matusik, W. (2024). Computational discovery of microstructured composites with optimal stiffness-toughness trade-offs. *Science Advances*, 10(5):eadk4284.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550.

Liang, Q., Gongora, A. E., Ren, Z., Tiihonen, A., Liu, Z., Sun, S., Deneault, J. R., Bash, D., Mekki-Berrada, F., Khan, S. A., et al. (2021). Benchmarking the performance of bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):1–10.

Liu, B., Xu, L., Wang, Y., and Huang, J. (2024). Diffusion model-based inverse design for thermal transparency. *Journal of Applied Physics*, 135(12).

Liu, F., Jiang, X., Wang, X., and Wang, L. (2020). Machine learning-based design and optimization of curved beams for multistable structures and metamaterials. *Extreme Mechanics Letters*, 41:101002.

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. (2021a). Heterogeneous risk minimization. In *International Conference on Machine Learning*, pages 6804–6814. PMLR.

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. (2021b). Kernelized heterogeneous risk minimization. *arXiv preprint arXiv:2110.12425*.

Liu, W. K., Li, S., and Park, H. (2021c). Eighty years of the finite element method: Birth, evolution, and future. *arXiv preprint arXiv:2107.04960*.

Liu, X., Athanasiou, C. E., Padture, N. P., Sheldon, B. W., and Gao, H. (2021d). Knowledge extraction and transfer in data-driven fracture mechanics. *Proceedings of the National Academy of Sciences*, 118(23).

Liu, X. and Hu, G. (2016). Elastic metamaterials making use of chirality: a review. *Journal of Mechanical Engineering*, 62:403–418.

Liu, X. N., Huang, G. L., and Hu, G. K. (2012). Chiral effect in plane isotropic micropolar elasticity and its application to chiral lattices. *Journal of the Mechanics and Physics of Solids*, 60:1907–1921.

Logg, A., Mardal, K.-A., and Wells, G. (2012). *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media.

Logg, A. and Wells, G. N. (2010). Dolfin: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37(2):1–28.

Lookman, T., Balachandran, P. V., Xue, D., and Yuan, R. (2019). Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):1–17.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

Lu, L., Dao, M., Kumar, P., Ramamurty, U., Karniadakis, G. E., and Suresh, S. (2020). Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proceedings of the National Academy of Sciences*, 117(13):7052–7062.

Lu, Z. and Norris, A. N. (2021). Unilateral and nonreciprocal transmission through bilinear spring systems. *Extreme Mechanics Letters*, 42:101087.

Lu, Z. and Norris, A. N. (2022). Passive nonreciprocity-induced directional wave scattering. *Extreme Mechanics Letters*, 51:101600.

Luu, R. K., Wysokowski, M., and Buehler, M. J. (2023). Generative discovery of de novo chemical designs using diffusion modeling and transformer deep neural networks with application to deep eutectic solvents. *Applied Physics Letters*, 122(23).

Mahajan, D., Tople, S., and Sharma, A. (2021). Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR.

Marrel, A. and Iooss, B. (2024). Probabilistic surrogate modeling by gaussian process: A review on recent insights in estimation and validation. *Reliability Engineering & System Safety*, page 110094.

Mianroodi, J. R., H Siboni, N., and Raabe, D. (2021). Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials. *npj Computational Materials*, 7(1):1–10.

Mienye, I. D. and Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10:99129–99149.

Ming, Y., Yin, H., and Li, Y. (2022). On the impact of spurious correlation for out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10051–10059.

Mohammadzadeh, S. and Lejeune, E. (2022). Predicting mechanically driven full-field quantities of interest with deep learning-based metamodels. *Extreme Mechanics Letters*, 50:101566.

Mohammadzadeh, S., Prachaseree, P., and Lejeune, E. (2023). Investigating deep learning model calibration for classification problems in mechanics. *Mechanics of Materials*, 184:104749.

Morgan, D. and Jacobs, R. (2020). Opportunities and challenges for machine learning in materials science. *Annual Review of Materials Research*, 50:71–103.

Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J., and Bessa, M. (2019). Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences*, 116(52):26414–26420.

Nagarajan, V., Andreassen, A., and Neyshabur, B. (2020). Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*.

Nassar, H., Chen, Y. Y., and Huang, G. L. (2020a). Polar metamaterials: a new outlook on resonance for cloaking applications. *Physical Review Letters*, 124:084301.

Nassar, H., Xu, X. C., Norris, A. N., and Huang, G. L. (2017). Modulated phononic crystals: non-reciprocal wave propagation and willis materials. *Journal of the Mechanics and Physics of Solids*, 101:10–29.

Nassar, H., Yousefzadeh, B., Fleury, R., Ruzzene, M., Alu, A., Daraio, C., Norris, A. N., Huang, G., and Haberman, M. R. (2020b). Non-reciprocity in acoustic and elastic materials. *Nature Reviews Materials*, 5:667–685.

Nguyen, Q. and Lejeune, E. (2024). Segmenting mechanically heterogeneous domains via unsupervised learning. *Biomechanics and Modeling in Mechanobiology*, pages 1–24.

Ni, B. and Gao, H. (2021). A deep learning approach to the inverse problem of modulus identification in elasticity. *MRS Bulletin*, 46(1):19–25.

Ni, B., Kaplan, D. L., and Buehler, M. J. (2024). Forcegen: End-to-end de novo protein generation based on nonlinear mechanical unfolding responses using a language diffusion model. *Science Advances*, 10(6):eadl4000.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632.

Nix, D. A. and Weigend, A. S. (1994). Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE.

Oyewole, G. J. and Thopil, G. A. (2023). Data clustering: application and trends. *Artificial Intelligence Review*, 56(7):6439–6475.

Park, H. S., Gall, K., and Zimmerman, J. A. (2006). Deformation of fcc nanowires by twinning and slip. *Journal of the Mechanics and Physics of Solids*, 54(9):1862–1881.

Patil, G. U., Cui, S., and Matlack, K. H. (2022). Leveraging nonlinear wave mixing in rough contacts-based phononic diodes for tunable nonreciprocal waves. *Extreme Mechanics Letters*, 55:101821.

Peng, G. C., Alber, M., Buganza Tepole, A., Cannon, W. R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W. W., Perdikaris, P., et al. (2021). Multiscale modeling meets machine learning: What can we learn? *Archives of Computational Methods in Engineering*, 28(3):1017–1037.

Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012.

Pilania, G. (2021). Machine learning in materials science: From explainable predictions to autonomous design. *Computational Materials Science*, 193:110360.

Piratla, V., Netrapalli, P., and Sarawagi, S. (2020). Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, pages 7728–7738. PMLR.

Polyzos, K. D., Lu, Q., and Giannakis, G. B. (2023). Bayesian optimization with ensemble learning models and adaptive expected improvement. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Prachaseree, P. and Lejeune, E. (2022). Learning mechanically driven emergent behavior with message passing neural networks. *arXiv preprint arXiv:2202.01380*.

Qin, C., Klabjan, D., and Russo, D. (2017). Improving the expected improvement algorithm. *Advances in Neural Information Processing Systems*, 30.

Quade, M., Abel, M., Nathan Kutz, J., and Brunton, S. L. (2018). Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):063116.

QUVA-Lab, U. o. A. (2017). Artemis. https://github.com/QUVA-Lab/artemis.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707.

Raissi, M., Yazdani, A., and Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. A. (2021). Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948.

Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., Yu, P. S., and He, L. (2022). Deep clustering: A comprehensive survey. *arXiv preprint arXiv:2210.04142*.

Riquelme, N., Von Lücken, C., and Baran, B. (2015). Performance metrics in multi-objective optimization. In *2015 Latin American computing conference (CLEI)*, pages 1–11. IEEE.

Ruan, X., Jiang, P., Zhou, Q., Hu, J., and Shu, L. (2020). Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. *Structural and Multidisciplinary Optimization*, 62:3021–3052.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. (2020). An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR.

Saha, S., Gan, Z., Cheng, L., Gao, J., Kafka, O. L., Xie, X., Li, H., Tajdari, M., Kim, H. A., and Liu, W. K. (2021). Hierarchical deep learning neural network (hidenn): An artificial intelligence (ai) framework for computational science and engineering. *Computer Methods in Applied Mechanics and Engineering*, 373:113452.

Sanchez-Lengeling, B. and Aspuru-Guzik, A. (2018). Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365.

Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR.

Scheibner, C., Souslov, A., Banerjee, D., Surowka, P., Irvine, W. T. M., and Vitelli, V. (2020). Odd elasticity. *Nature Physics*, 16:475–480.

Schölkopf, B. (2022). Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 765–804.

Schulz, A., Wang, H., Grinspun, E., Solomon, J., and Matusik, W. (2018a). Interactive exploration of design trade-offs. *ACM Transactions on Graphics (TOG)*, 37(4):1–14.

Schulz, E., Speekenbrink, M., and Krause, A. (2018b). A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85:1–16.

Shaat, M. (2020). Nonreciprocal elasticity and the realization of static and dynamic nonreciprocity. *Scientific Reports*, 10:21676.

Shaat, M. and Park, H. S. (2023). Chiral nonreciprocal elasticity and mechanical activity. *Journal of the Mechanics and Physics of Solids*, 171:105163.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. (2021). Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Simchi-Levi, D. and Wang, C. (2023). Multi-armed bandit experimental design: Online decision-making and adaptive inference. In *International Conference on Artificial Intelligence and Statistics*, pages 3086–3097. PMLR.

Sinaga, K. P. and Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.

Smidt, T. E., Geiger, M., and Miller, B. K. (2021). Finding symmetry breaking order parameters with euclidean neural networks. *Physical Review Research*, 3(1):L012002.

Smith, M. (2009). *ABAQUS/standard user's manual, version 6.9*. Dassault Systèmes Simulia Corp, Providence, RI.

Smits, A. J., McKeon, B. J., and Marusic, I. (2011). High–reynolds number wall turbulence. *Annual Review of Fluid Mechanics*, 43:353–375.

Su, M., Zhong, Q., Peng, H., and Li, S. (2021). Selected machine learning approaches for predicting the interfacial bond strength between frps and concrete. *Construction and Building Materials*, 270:121456.

Sui, F., Guo, R., Zhang, Z., Gu, G. X., and Lin, L. (2021). Deep reinforcement learning for digital materials design. *ACS Materials Letters*, 3(10):1433–1439.

Süli, E. (2012). Lecture notes on finite element methods for partial differential equations. https://people.maths.ox.ac.uk/suli/fem.pdf.

Svolos, L., Mourad, H. M., Manzini, G., and Garikipati, K. (2022). A fourth-order phase-field fracture model: Formulation and numerical solution using a continuous/discontinuous galerkin method. *Journal of the Mechanics and Physics of Solids*, page 104910.

Szabó, B. and Babuška, I. (2021). *Finite element analysis: Method, verification and validation, 2nd Edition*. Newark: John Wiley & Sons, Incorporated.

Tan, T. H., Mietke, A., Li, J., Chen, Y., Higinbotham, H., Foster, P. J., Gokhale, S., Dunkel, J., and Fakhri, N. (2022). Odd dynamics of living chiral crystals. *Nature*, 607:287–293.

Trainiti, G. and Ruzzene, M. (2016). Non-reciprocal elastic wave propagation in spatiotemporal periodic structures. *New Journal of Physics*, 18:083047.

Tripuraneni, N., Adlam, B., and Pennington, J. (2021). Overparameterization improves robustness to covariate shift in high dimensions. *Advances in Neural Information Processing Systems*, 34.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Tsymbal, A. (2004). The problem of concept drift: definitions and related work. https://publications.scss.tcd.ie/tech-reports/reports.04/TCD-CS-2004-15.pdf.

Viesca, R. C. and Rice, J. R. (2011). Elastic reciprocity and symmetry constraints on the stress field due to a surface-parallel distribution of dislocations. *Journal of the Mechanics and Physics of Solids*, 59(4):753–757.

Wallen, S. P., Haberman, M. R., Lu, Z., Norris, A., Wiest, T., and Seepersad, C. C. (2018). Static and dynamic non-reciprocity in bi-linear structures. *Proceedings of Meetings on Acoustics*, 34:065002.

Wang, C., Li, S., Zeng, D., and Zhu, X. (2021). Quantification and compensation of thermal distortion in additive manufacturing: a computational statistics approach. *Computer Methods in Applied Mechanics and Engineering*, 375:113611.

Wang, J. (2023). An intuitive tutorial to gaussian processes regression. *Computing in Science & Engineering*, 25(4):4–11.

Wang, J., Chen, W. W., Da, D., Fuge, M., and Rai, R. (2022). Ih-gan: A conditional generative model for implicit surface-based inverse design of cellular structures. *Computer Methods in Applied Mechanics and Engineering*, 396:115060.

Wang, K. and Sun, W. (2019). Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 346:216–241.

Wang, K., Sun, W., and Du, Q. (2019). A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with ai-guided experimentation. *Computational Mechanics*, 64(2):467–499.

Wang, L., Martinez, J. A. I., Ulliac, G., Wang, B., Laude, V., and Kadic, M. (2023). Non-reciprocal and non-newtonian mechanical metamaterials. *Nature Communications*, 14:4778.

Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

Wang, X. and Aitchison, L. (2021). Bayesian ood detection with aleatoric uncertainty and outlier exposure. In *Fourth Symposium on Advances in Approximate Bayesian Inference*.

Wang, Y., Wu, Q., Tian, Y., and Huang, G. (2024). Non-hermitian wave dynamics of odd plates: microstructure design and theoretical modelling. *Journal of the Mechanics and Physics of Solids*, 182:105462.

Wang, Y., Yousefzadeh, B., Chen, H., Nassar, H., Huang, G., and Daraio, C. (2018). Observation of nonreciprocal wave propagation in a dynamic phononic lattice. *Physical Review Letters*, 121:194301.

Wang, Z. and Jegelka, S. (2017). Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR.

Weichwald, S. and Peters, J. (2021). Causality in cognitive neuroscience: concepts, challenges, and distributional robustness. *Journal of Cognitive Neuroscience*, 33(2):226–247.

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1):1–34.

Winship, C. and Mare, R. D. (1992). Models for sample selection bias. *Annual review of sociology*, 18(1):327–350.

Winship, C. and Morgan, S. L. (1999). The estimation of causal effects from observational data. *Annual review of sociology*, 25(1):659–706.

Wu, W., Hu, W., Qian, G., Liao, H., Xu, X., and Berto, F. (2019). Mechanical design and multifunctional applications of chiral mechanical metamaterials: a review. *Materials and Design*, 180:107950.

Xie, Y., Li, S., Wu, C., Lyu, D., Wang, C., and Zeng, D. (2022a). A generalized bayesian regularization network approach on characterization of geometric defects in lattice structures for topology optimization in preliminary design of 3d printing. *Computational Mechanics*, 69(5):1191–1212.

Xie, Y., Wu, C., Li, B., Hu, X., and Li, S. (2022b). A feed-forwarded neural network-based variational bayesian learning approach for forensic analysis of traffic accident. *Computer Methods in Applied Mechanics and Engineering*, 397:115148.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021). Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Yang, Y., Guo, H., Du, Z., Hong, W., Lu, T., and Wang, T. (2022). Rate-dependent fracture of hydrogels due to water migration. *Journal of the Mechanics and Physics of Solids*, 167:105007.

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. (2021). Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34.

Yin, H. and Liu, C. (2023). Anisotropy and asymmetry of the elastic tensor of lattice materials. *Journal of Elasticity*, 154(5):659–691.

Yuan, L., Park, H. S., and Lejeune, E. (2022). Towards out of distribution generalization for problems in mechanics. *Computer Methods in Applied Mechanics and Engineering*, 400:115569.

Yuan, L., Yuan, W., and Wang, G. (2019). A notification of local minimum pressure at deep elastic–plastic indentation and its application. *Extreme Mechanics Letters*, 29:100446.

Yuan, L., Yuan, W., and Wang, G. (2020). Effects of residual stress on the hardness of elastoplastic material under spherical indentation. *Journal of Applied Mechanics*, 87(5):051004.

Zhan, D. and Xing, H. (2020). Expected improvement for expensive optimization: a review. *Journal of Global Optimization*, 78(3):507–544.

Zhang, H. K., Chen, Y., Liu, X. N., and Hu, G. K. (2020a). An asymmetric elastic metamaterial model for elastic wave propagation. *Journal of the Mechanics and Physics of Solids*, 135:103796.

Zhang, X. and Garikipati, K. (2020). Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures. *Computer Methods in Applied Mechanics and Engineering*, 372:113362.

Zhang, Y., Apley, D. W., and Chen, W. (2020b). Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific reports*, 10(1):4924.

Zheng, L., Karapiperis, K., Kumar, S., and Kochmann, D. M. (2023). Unifying the design space and optimizing linear and nonlinear truss metamaterials by generative modeling. *Nature Communications*, 14(1):7563.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. (2021). Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*.

CURRICULUM VITAE