8-2024

Physics-Informed Machine Learning Methods for Inverse Design of Multi-Phase Materials with Targeted Mechanical Properties

Yunpeng Wu
yunpeng@g.clemson.edu

PHYSICS-INFORMED MACHINE LEARNING METHODS FOR INVERSE DESIGN OF MULTI-PHASE MATERIALS WITH TARGETED MECHANICAL PROPERTIES

A Dissertation

Presented to

the Graduate School of

Clemson University

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

Mechanical Engineering

By

Yunpeng Wu

August 2024

Accepted by:

Dr. Gang Li, Committee Chair

Dr. Feng Luo

Dr. Huijuan Zhao

Dr. Oliver Myers

ABSTRACT

Advances in machine learning algorithms and applications have significantly enhanced engineering inverse design capabilities. This work focuses on the machine learning-based inverse design of material microstructures with targeted linear and nonlinear mechanical properties. It involves developing and applying predictive and generative physics-informed neural networks for both 2D and 3D multiphase materials.

The first investigation aims to develop a machine learning method for the inverse design of 2D multiphase materials, particularly porous materials. We first develop machine learning methods to understand the implicit relationship between a material's microstructure and its mechanical behavior. Specifically, we use ResNet-based models to predict the elastic modulus and stress-strain curves of linear and nonlinear porous materials from their microstructure images. To generate microstructures of porous materials with targeted mechanical behavior, we create variational autoencoder (VAE) based neural networks. These networks generate the microstructure of porous materials from a prescribed elastic modulus or stress-strain curve. In both property prediction and microstructure generation, the stress-strain curves are approximated using cubic polynomials and characterized by their coefficients. To explicitly enforce the mechanics of materials in the generative machine learning models, we devise and incorporate a new condition fusion layer into the traditional VAE architecture. Additionally, a pretrained regression model is introduced to constrain the decoder, ensuring the production of physically meaningful images. The results show that this machine learning approach is capable of ultra-fast prediction of material properties directly from microstructure images, as well as the inverse design of material microstructures to achieve desirable mechanical behaviors.

The second investigation focuses on the inverse design of 3D multiphase materials, specifically considering fiber-reinforced polymer composites (FRPC) as the model system. This research aims to develop physics-informed neural networks for inverse design of such a material system. Compared to 2D porous materials, 3D FRPC involve complex 3D microstructure geometries and require physically feasible topologies, making the inverse design significantly more challenging. To address these challenges, we develop a novel diffusion model for 3D fiber reconstruction and generation. This model includes a forward diffusion process that adds noise to the fiber distribution and a reverse process that denoises to generate desirable fiber distributions. The diffusion model is defined via a stochastic differential equation (SDE), and both the diffusion and reverse processes are modeled as solutions to this SDE. To ensure feasible topology for the generated fiber distributions, non-collision constraints are incorporated into the generative neural networks. The results demonstrate that these new models can generate high-quality 3D FRPC designs with tailored mechanical behaviors while ensuring compliance with physical constraints.

Contents

List of Tables

List of Figures

CHAPTER ONE

INTRODUCTION AND MOTIVATION

In recent years, machine learning methods have found more applications in engineering analysis and design [1-4]. One of such applications that have attracted much attention is the analysis and prediction of material properties, including the thermal, mechanical, electrical properties of metals, ceramics, polymers and composites [1, 3]. Among the material properties, mechanical properties such as stiffness and stress-strain behavior are particularly important for structural materials. Typically, mechanical properties are determined through experimental characterization which requires careful material sample preparation and expensive experimental equipment (e.g. Instron machine and strain gauges). Alternatively, the mechanical properties can be determined through physics based computational analysis. While computational analysis can be more efficient than experimental characterization, the construction and validation of high-fidelity models can still be time consuming. For example, as engineering materials' mechanical properties largely depend on their microstructural configurations, the construction of high-fidelity computational models requires accurate digital representation of the actual microstructures, which is often through careful segmentation of scanned micro-CT images. This process is both expensive and time consuming, especially when the microstructure is complex and highly inhomogeneous. In addition, the computational cost of simulating nonlinear mechanical behavior using high-fidelity models with very large degrees of freedom (DOF) is typically high. Given the rapid advancement of machine learning methods, it has been envisioned that the material characterization process can be largely accelerated using machine learning methods if the relationship between material microstructure and mechanical properties can be learned [5]. Unfortunately, currently there is not much work in

the literature related to predicting mechanical properties of engineering materials using machine learning methods, due to, among other factors, the scarcity of experimental data.

A more challenging problem is the design of engineering materials from given material property requirements. Methods for solving material design problems can be broadly categorized into direct and inverse design methods. The direct method of experimental or analytical screening is largely a trial-and-error process that can be extremely inefficient. The design space engineering materials is enormous as it includes different combinations of material components/phases, their arrangements, volume fractions, dimensions, and their morphology, which is all together referred to as the material microstructure. Due to the large number of design variables, even with large computational resources, a global strategy for exploring completely the design space of material microstructure is intractable. Inverse design starts with the desired performance and searches for an ideal material microstructure. In essence, any inverse design process is an optimization process. Once again, the large material design space makes the application of standard optimization methods, such as evolutionary structural optimization (ESO) [6] and the solid isotropic material with penalization (SIMP) [7] methods, difficult. Data-based inverse design approaches such as machine learning, however, do not seek an explicit description of the mapping between the material design space and the functional space. The mapping is constructed via machine learning and stored in neural networks. Similarly, the search of the optimal solution is also implicitly done through the trained neural networks. There are three main approaches for solving inverse problems: variational autoencoders (VAEs) [8], reinforcement learning (RL) [9], and generative adversarial networks (GANs) [10]. In materials design, attempts have been made by using all the three approaches. While encouraging results were reported in molecular design [11], drug design [12], crystal physics [13, 14], thermoelectrics [15], structural topology optimization [16] etc., however,

to the best of authors' knowledge, inverse design for mechanical properties, especially nonlinear stress-stain behavior has not yet been investigated using machine learning methods.

1.1 Material Inverse Design

In the realm of materials science, the intrinsic and distinctive qualities of composite materials lie in their heterogeneity and anisotropy, which afford them a complex array of physical and chemical properties. The multitude of material composition and structural possibilities results in an expansive material design space that has yet to be fully explored. This is particularly because the traditional direct method of addressing material design challenges is predominantly a trial-and-error process that can be remarkably inefficient. Inverse design, as the term implies, reverses the conventional approach by beginning with the desired material properties and seeking the ideal composite material structure. Any inverse design process is an optimization endeavor where the input is the material properties, and the output is the composite material structure. It is important to note that composite properties do not necessarily correspond to a singular material structure but to a range of possible structures. Inverse design has (1) analytical/mathematical modeling combined with optimization, or (2) data-driven search methods to navigate the functional space. In the analytical approach, a forward mathematical model is explicitly defined, criteria for finding a solution are established, and a method for arriving at a solution is chosen. The optimization process leverages the mathematical model to explore the functional space. However, analytical models are generally limited to the simplest composite structures, such as orthotropic laminae. For more complex material structures, incorporating the effects of manufacturing processes makes constructing a comprehensive analytical model to describe the composite's functional behavior virtually unfeasible. Another framework for inverse material design falls under data-driven

approaches, which do not seek an explicit mapping between material structural space and functional space. Here, machine learning emerges as a promising technique due to its rapid advancements.

On the manufacturing process design front, traditionally, composite manufacturing processes were developed with the aim of fulfilling economic and technical requirements. Optimizing these processes involved setting manufacturing parameters and then experimentally validating these parameters through extensive testing to develop a set of "B-basis Allowables" and predict failures. An equivalency test matrix would be completed for any changes in material constituents. Both processes depend heavily on existing domain knowledge, are costly, time-consuming, hinder agility, and pose significant barriers to exploring the design spaces and transitioning new composite materials and processes. Moreover, a key characteristic of composite material structures is that both the material and the structure are formed during the manufacturing process. Strictly speaking, classical fiber-reinforced composites are not merely materials; they are akin to micro-scale structures produced by the interplay of material and manufacturing processes. The quality and properties of the final manufactured part are influenced significantly by the production method, sometimes as much as the constituent materials themselves. Thus, manufacturing plays a critical role in the development of products made from polymer composites. In current practices, material design and manufacturing process development are mostly conducted in isolation, preventing the exploration of synergistic effects that could arise from their integration.

As mentioned, in the realm of material design, the prevailing methodology inverts the traditional paradigm by initiating the design process with predefined desirable material properties and subsequently seeking an optimal composite material architecture. This methodology is operationalized through three primary approaches: (1) Direct Screening: This approach entails

empirical experimentation with various material compositions and configurations, leveraging prior knowledge to test different combinations. Despite its straightforward application, this method is primarily characterized by a trial-and-error process, rendering it significantly inefficient due to its empirical nature. (2) Analytical/Mathematical Modeling and Optimization: Here, the design criteria and solution methodology are predetermined, facilitating the use of a mathematical model to navigate the functional space. This model serves to systematically explore possible solutions. However, this approach is inherently constrained to simplistic representations of composite architectures, limiting its applicability to more complex material systems. (3) Data-Driven Search Methods: These methods employ computational algorithms to traverse the functional space manifold without seeking an explicit representation of the relationship between material architecture and its functional properties. This approach is hindered by its inability to directly elucidate the underlying mapping between the architectural configuration of materials and their resultant functionalities.

Given these limitations, the integration of machine learning techniques presents a promising path for surmounting these challenges. Specifically, machine learning can facilitate the discovery of an explicit mapping between material architectures and functional spaces, thereby enhancing the efficacy and efficiency of material design processes.

1.2 Machine Learning and Inverse Design

The idea of machines that can learn and adapt originated in the mid-20th century. A pioneering British computer scientist, Alan Turing [17], discussed the possibility of machines learning in 1950. This set the stage for the formal inception of AI and machine learning. The first practical applications of machine learning were developed during this time. Arthur Samuel [18] wrote a

checkers-playing program that learned from its experiences, effectively becoming more skilled the more it played. The program's ability to improve itself by learning from past games was a landmark achievement in machine learning. Innovations in network architectures and algorithms, along with the increasing power of computers, revived interest in machine learning. The development of the backpropagation algorithm in 1986 by David Rumelhart [19], which improved the training of neural networks. The last decades have seen explosive growth in machine learning, driven by vast improvements in hardware (like GPUs), big data availability, and further algorithmic advancements. Major breakthroughs include the success of deep learning, which has dramatically improved the performance of tasks like image recognition, natural language processing, and predictive analytics. Today, deep learning—a subset of machine learning involving deep neural networks—dominates the field. Innovations like convolutional neural networks (CNNs) [20] and recurrent neural networks (RNNs) [21] have enabled incredible advances in areas such as autonomous vehicles [22], speech recognition [23], and real-time translation [24].

As the field of machine learning continues to evolve, it has become increasingly integral to various branches of engineering, particularly in the realm of material design. This area has witnessed a shift from traditional methodologies towards a more innovative, data-driven approach. In the conventional paradigm, the process typically began with the selection of composite material architectures based on existing knowledge and iterative testing to meet desired properties. However, machine learning inverts this paradigm by adopting a goal-oriented strategy where the starting point is the specification of desired material properties. This approach leverages advanced computational models to systematically explore the vast design space of potential composite materials. The input to these models consists of detailed descriptions of desired properties such as mechanical strength [25], thermal conductivity [26], electrical conductivity [27], corrosion

resistance [28], and other relevant physical [29] or chemical [30] characteristics. The output, generated through complex algorithms and data analysis techniques, is a set of optimal composite material architectures that are predicted to exhibit these specified properties. The underlying machine learning algorithms employ sophisticated techniques such as regression analysis, pattern recognition, and optimization algorithms to identify correlations and interactions between various material components and their configurations. This not only accelerates the process of material discovery but also enables the identification of novel materials with enhanced or previously unattainable properties. Furthermore, this data-based approach facilitates a more efficient exploration of the compositional and structural parameters that influence material performance. By harnessing the power of machine learning, engineers can predict the outcomes of material combinations and configurations without the need for extensive physical prototyping and inverse design, thereby reducing development costs and time to market.

In essence, the integration of machine learning into material design signifies a transformative advancement in the engineering discipline. It empowers researchers and engineers to transcend traditional boundaries by enabling a predictive design framework. This framework not only streamlines the discovery of innovative materials but also paves the way for the development of material properties analysis and inverse design technologies that meet the increasingly complex demands of modern applications.

1.3 Research Questions and Approach

The ultimate objective of this project is to develop and apply physics-informed machine learning (ML) techniques for the inverse design of multi-phase materials, with a particular focus on achieving specific mechanical properties. This endeavor will transition from addressing four key

research questions representing the design challenges of two-dimensional (2D) materials to three-dimensional (3D) materials in a sequential manner. The research questions are:

1. Can machine learning methods learn the implicit relationship between multiphase materials' microstructures and their mechanical behavior?
2. Can machine learning methods generate 2-D microstructures of multiphase materials based on targeted mechanical properties and nonlinear deformation behavior?
3. Can machine learning methods generate 3-D geometry and topology of multiphase materials based on targeted mechanical behavior?
4. How can physics and manufacturability requirements be incorporated into the machine learning models for improved prediction and generation?

The completion of this project is structured around addressing these four research questions and progressively establishing the feasibility and effectiveness of ML models in facilitating inverse design processes.

1.3.1 Establishing the Relationship between Material Microstructures and Mechanical Properties

The first research question focuses on uncovering the implicit relationship between the microstructures of materials and their resultant behaviors. This step is crucial in understanding how the arrangement, size, and composition of microscopic features within a material influence its physical and mechanical properties, such as strength, ductility, hardness, and elasticity. This exploration is foundational for materials science, as it informs the development of new materials with tailored properties for specific applications, enhancing performance, durability, and efficiency in various engineering and technological contexts.

To addressing the first question of whether machine learning methods can discern the implicit relationship between a material's microstructures and its behavior, it is necessary to first specify a material whose behavior can be clearly defined. This initial step involves selecting a material with well-understood properties and behaviors in response to various conditions, thereby providing a solid foundation for exploring the capabilities of machine learning in identifying and learning these complex relationships.

Given that material challenges can manifest in both two-dimensional (2D) and three-dimensional (3D) contexts, and that 2D microstructures tend to be simpler to understand, our focus narrows to 2D configurations. Within this domain, various types of microstructures exist, including grain structures [31], layered structures [32], porous structures [33], and fiber-reinforced structures [34]. Among these, grain structures and fiber-reinforced structures present complexities that are more accurately represented and exhibit significant performance distinctions in 3D models. On the other hand, layered structures, while simpler in 2D formats, lack the complexity that necessitates advanced analytical methods, such as machine learning, due to their straightforward, uncomplicated nature. Consequently, porous materials are selected for the 2D approach due to their suitable balance of structural complexity and analytical tractability, making them an optimal candidate for employing machine learning techniques in the investigation of their properties and behaviors.

It is evident that the field of materials science and mechanical engineering includes a wide spectrum of characteristics, which are essential for understanding and optimizing the behavior of materials under various conditions. While each of these properties plays a critical role in determining how materials respond to external stresses, environmental conditions, and long-term usage, the starting point for studying mechanical material properties is often elasticity. This is

because elasticity is a fundamental property that describes how materials deform and then return to their original shape when subjected to forces. This characteristic is crucial for understanding material behavior under various stress conditions, informing the design and selection of materials for engineering applications. Structural analysis, for example, relies heavily on understanding the elastic properties of materials to predict deflections, stresses, and strains. The elasticity concepts of elastic modulus and the stress-strain relationship are critical for more advanced topics such as plasticity [35], fracture mechanics [36], and fatigue [37]. In essence, elasticity is a cornerstone of materials science and mechanical engineering, providing the necessary tools and concepts to understand, utilize, and innovate with materials across a wide range of applications. Therefore, we choose to focus on elasticity as the topic, addressing it as the targeted material property to solve our research questions.

To elucidate the relationship between elastic properties and material microstructure, we aim to adopt a general and accessible ML approach. Recognizing the wealth of existing methodologies and successful case studies in image recognition, along with the availability of potent analytical tools, we are motivated to leverage these advancements as a foundational basis for our research. Within the expansive landscape of technologies that deploy image recognition as a foundational machine learning technique, deep learning stands at the forefront, eclipsing other methods such as support vector machines (SVMs) [38], decision trees [39], random forests [40], K-nearest neighbors (KNN) [41], principal component analysis (PCA) [42], and linear discriminant analysis (LDA) [43]. Consequently, deep learning [44] has been selected as our primary methodology due to its superior performance and versatility across a broad spectrum of applications, including object detection [45], image segmentation [46], and video analysis [47]. Deep learning's dominance is further evidenced by its widespread implementation by leveraging specialized architectures such

as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to address complex challenges.

In summary, to address Research Question 1, we develop a deep learning based predictive ML model to correlate the elastic material properties with material microstructures represented by images. Our testing demonstrates that this recognition capability can accurately predict the behavior of 2D-based materials. This advancement has the potential to be applied across a broad spectrum of mechanical materials, serving as preliminary groundwork for material inverse design. This approach not only enhances our understanding of material properties through image analysis but also lays the foundation for designing materials with desired mechanical characteristics from the outset.

1.3.2. ML Models for Generating 2D Microstructures of Multi-Phase Materials

The second research question aims to address a more complex issue regarding the design of materials' microstructures for targeted mechanical behavior. Building on the insights from Research Question 1, where we establish how the microstructures of materials lead to material properties, Research Question 2 seeks to explore the inverse path. Specifically, we aim to identify the implicit route from the material properties back to the microstructure of the material. In this context, our focus shifts towards determining how specific properties of porous materials can inform and guide the generation of their microstructures, thereby providing a comprehensive understanding of the bidirectional relationship between a material's microstructure and its properties.

Inverse design methods in materials science and engineering involve a process where the desired outcomes or properties of a material or system are specified first, and then the necessary structure or configuration to achieve those properties is determined. This approach contrasts with the more

common forward design process, where materials or components are designed based on trial-and-error or heuristic principles, and their properties are then tested and analyzed. Existing inverse design methods typically rely on a combination of computational modeling, optimization algorithms, and sometimes experimental validation to iteratively refine a design. Computational optimization based inverse design methods have been applied in various fields, including optical and photonic design [48], material science [49], and structural engineering [50]. However, these inverse design methods often face challenges due to the complexity of the design space, the computational cost of simulations, and the limitations of the models used.

The emergence of machine learning and artificial intelligence techniques has begun to complement and, in some cases, revolutionize these traditional approaches by enabling more efficient exploration of design spaces and by offering predictive capabilities that were previously unattainable. The progression in ML based inverse design methods begins with image recognition techniques. Initially, this domain's development was somewhat constrained, with Generative Adversarial Networks (GANs) [51] and Variational Autoencoders (VAEs) [52] being the predominant methods for image generation. Over time, advancements in machine learning have broadened the spectrum of image generation methodologies to include various forms of GANs such as standard GANs [53], conditional GANs (cGANs) [54], StyleGANs [55], and CycleGANs [56], alongside VAEs, auto-regressive models [57], diffusion models [58], and transformer-based models [59]. These image-based recognition techniques have proven to be particularly effective for analyzing 2D microstructures. In the context of porous materials, where specific material properties such as elastic modulus and Young’s modulus are of interest, leveraging these advanced image generation and recognition methods facilitates the inverse design process. To address Research Question 2, we develop an approach that employs a sophisticated blend of computational

models to both analyze existing microstructural images and generate new microstructural configurations that align with targeted mechanical behaviors. By integrating generative modeling and image analysis, our ML model achieves a high degree of precision in material design, optimizing for specific properties like elasticity and structural integrity. The inverse design process involves using the proposed ML model to iteratively refine and generate microstructures that meet predetermined mechanical property criteria. This approach exemplifies the cutting-edge intersection of materials science and machine learning, offering a powerful toolkit for the design and discovery of new materials with tailored properties.

By addressing this research question, we establish a link between image generation technologies and mechanical inverse design. Furthermore, this method can be extended to include inverse design for materials with 2D-like properties, which holds significant potential. This approach not only broadens the applicability of image generation in the context of material science but also enhances the utility of inverse design methods across both 2D and 3D material systems, offering valuable tools for material innovation and development.

1.3.3 3D Microstructure Generation

In addressing the first two research questions, attention is predominantly directed towards the development of machine learning models for 2D microstructure recognition and generation. However, this effort has a notable limitation: the application of 2D microstructure inverse design methodologies is often constrained when translated into practical scenarios. This limitation motivated us to transition from 2D analyses and generation to a 3D framework.

In this effort, while our focus remains on dual-phase materials, from a practical engineering perspective, our model material system shifts to fiber-reinforced composite materials. This selection is driven by the unique mechanical properties and application potentials that fiber-

reinforced composites offer, aligning with our objective to identify and analyze optimal 3D microstructural materials. For material properties, our focus remains on the elastic modulus and stress-strain behavior, of fiber composites.

Designing 3D fiber composite structures presents a significantly more complex challenge compared to 2D microstructures. In 2D microstructure design, datasets can be augmented by utilizing randomly cut sections to ensure coverage of a wide range of possible scenarios. This approach allows for the effective use of image recognition and generation techniques to analyze and replicate the microstructures. However, the complexity of 3D fiber composites, characterized by their random orientation, varying diameters, positions, number of fibers, and rotation angles, cannot be directly captured or analyzed using conventional image-based methods. These attributes extend beyond the capabilities of standard 2D imaging, making it difficult to directly apply image recognition and generation models for 3D fiber composite design. Therefore, there is a need to develop alternative methods that can accommodate the multidimensional and variable nature of fiber composites, addressing the challenges posed by their 3D characteristics.

We present an advanced 3D diffusion-based machine learning model designed for precise 3D prediction and generation tasks. This model specifically targets the accurate determination of fiber positions and orientations, which are critical for various applications. By leveraging the principles of diffusion, our approach ensures high fidelity in the spatial configuration and alignment of fibers. In addition to the core model, we have integrated an innovative collision-free methodology that addresses common physical simulation challenges in the generation process. This novel technique not only prevents fiber overlap and intersection but also enhances the stability and reliability of the generated structures. By incorporating these improvements, our model achieves superior physics-based performance, making it highly effective for practical implementations in fields such

as materials science, bioengineering, and computational design. The integration of these advanced features demonstrates our commitment to pushing the boundaries of machine learning in inverse design, providing a robust and efficient solution for complex 3D modeling tasks.

Addressing this challenge opens the door to generating 3D structures, such as fibers, using machine learning techniques. Moreover, the approach is not limited to dual-phase materials; it can also be applied to materials with more phases, paving the way for the development of a comprehensive mechanical material inverse design system, thereby broadening the scope of materials inverse design through advanced ML models.

1.3.4. Incorporation of Physics and Manufacturability Requirements into ML Models

Incorporating physics and manufacturability requirements into machine learning models is crucial. Integrating physical laws into machine learning models helps ensure that predictions and solutions are physically plausible. This leads to more accurate and reliable outcomes, especially in complex systems where purely data-driven approaches might fail to capture underlying principles. Models that incorporate physics-based constraints are better at generalizing to new, unseen scenarios. This is because they are not solely reliant on the training data but also adhere to the fundamental laws governing the system. This can be particularly important in fields like fluid dynamics [60], material science [61], and climate modeling [62]. Physics-informed machine learning models often require less data to achieve high performance. By embedding known physical laws, these models can make better use of available data, reducing the need for large datasets that are often difficult and expensive to obtain. Incorporating prior knowledge about physics can guide the learning process, leading to faster convergence and reduced training times. This is because the model is already biased towards feasible solutions, minimizing the search space it needs to explore. Physics-informed models are typically more interpretable because their predictions can be directly linked

to known physical laws and principles. By incorporating physics, models become more robust to uncertainties and variations in the data. This robustness is essential for real-world applications where conditions can change, and data may be noisy or incomplete. Inverse design problems, where the goal is to find a system configuration that meets specific performance criteria, benefit significantly from physics-informed machine learning. This approach can lead to innovative designs and optimized solutions that might not be achievable through traditional methods. Physics-informed models can be more computationally efficient. By leveraging known physical laws, these models can avoid unnecessary computations and focus on feasible solutions, leading to cost savings in computational resources. Overall, incorporating physics and design requirements into machine learning models bridges the gap between data-driven methods and domain-specific knowledge, resulting in more robust, efficient, and reliable solutions that are essential for advancing both scientific research and practical applications.

In the inverse design of 3D fiber composites, a significant challenge arises from the potential for fiber collisions, where individual fibers may intersect with each other. This intersection presents a critical physical issue in the design of fiber composites. Based on the hypothesis that incorporating physics-based knowledge into the ML model can effectively resolve fiber collisions, we redesign the classical diffusion models to make it suitable for generating 3D fiber reinforced composite materials.

In addition, we further enhance our machine learning models so that the generated material designs are consistent with practical manufacturing methods. We will employ 3D printing technology to manufacture the ML-designed 3D fiber reinforced composite materials and then experimentally evaluate the elastic properties of the produced materials. To ensure practical applicability, our redesigned model will feature layered material configurations that are feasible for 3D printing. In

addition, we introduce a Voronoi-based strategy for generating fiber data sets that are close to the actual fiber distribution produced by the fuse deposition modeling (FDM) 3D printing process. This updated model will then be subjected to experimental testing to validate its effectiveness.

In summary, we propose a systematic ML-based inverse material design approach beginning with problem definition, followed by the selection of the type of material system and microstructure, data preparation, ML model development, and culminating in the accomplishment of the inverse design objective. This structured framework aims to rigorously validate the potential of machine learning models for advancing the inverse design process of multi-phase materials by bridging the gap between desired mechanical properties and material microstructures.

By completing this research, we establish a robust connection between machine learning and inverse design of 2D and 3D multiphase materials. Furthermore, we extend these methods to enforce physical principles and manufacturing requirements. This research not only bridges the gap between theoretical design and practical implementation but also paves the way for future innovations in the field of composite materials and their manufacturing processes.

1.4 Manuscript Organization

The remaining chapters of this manuscript are presented as follows. Chapter 2 presents a literature review. Chapter 3 illustrates the general framework of machine learning for inverse design and emphasizes the implementation of machine learning methodologies in inverse design processes. Chapter 4 describes the machine learning methods generate 2D microstructure based on targeted mechanical properties and nonlinear deformation behavior. Chapter 5 presents machine learning generate 3D geometry and topology of multiphase materials based on targeted mechanical behavior and implement physics and manufacturability requirements be incorporated into the

machine learning methods. In the literature review (Chapter 3), we start by examining the conventional approaches to mechanical material inverse design, highlighting the current traditional methods along with their advantages and drawbacks. Following this, the review explores the integration of machine learning into mechanical material inverse design, covering both its predictive and generative capacities. The benefits of applying machine learning in this field are then detailed, illustrating how these advanced computational techniques offer significant improvements over traditional methods in terms of accuracy, efficiency, and the ability to handle complex material behaviors and structures.

Separate framework and implementation chapters are devoted to solving each research question. In the first chapter, we outline a machine learning-based methodology that delineates an implicit connection between material microstructures and their targeted properties. We detail the creation of a dataset composed of randomly generated 2D porous materials, including specifics on how this data was generated. Addressing the initial research question, we introduce novel machine learning models specifically developed to predict the targeted material properties from their microstructures. These models are meticulously dissected to explain their components, operational mechanisms, and how they process microstructural data to make predictions. Finally, we explore potential extensions of these models and discuss prospects.

In the subsequent chapter, building upon the methods outlined in the previous chapter, we proceed to develop a generative machine learning model. This model is designed to create material microstructures that meet specific targeted material properties. We then provide a detailed breakdown of these models, shedding light on their operational intricacies and the innovative approaches they incorporate. The discussion also covers the novel aspects of our models, setting

them apart from existing methods. We outline the advantages of these new models, emphasizing their potential impact on the field.

In the following chapter, we present a dataset centered around 3D fiber structures, detailing the methodology behind its creation and the characteristics of its distribution. This introduction sets the foundation for unveiling a specialized machine learning model designed for generating 3D fiber structures. The model is thoroughly examined, with a deep dive into its architecture, operational mechanics, and the principles that facilitate its functionality. Subsequently, we explore the benefits of this innovative model, highlighting how it advances the field of material science through its ability to accurately simulate and generate complex 3D fiber structures. The discussion culminates with considerations on the future implications of this model, including potential applications, anticipated enhancements, and its integration into practical work. And introduce an improved machine learning method to solve the fiber collision problem, shows how the physics and design requirements be incorporated in the machine learning models. This chapter aims to show the intersection of machine learning and material science, illustrating the transformative potential of these models in crafting the next generation of materials and facilitating groundbreaking research in the domain.

For the future work chapter, we will integrate 3-D printing with our current methods, detailing a feasible approach to implementing new inverse design methods in practical applications. We will outline specific steps and considerations for this integration, ensuring optimal performance. Additionally, we will discuss potential challenges and solutions, providing a guide for practitioners. The chapter will conclude with an outlook on future work and prospects, highlighting key areas for further research and the long-term impact of these technologies on various practical industries.

CHAPTER TWO

LITERATURE REVIEW

2.1 Traditional Mechanical Material Inverse Design

Inverse design in the context of mechanical materials entails defining desired outcomes or performance metrics, and thereafter deducing the material composition and architecture that would fulfill those objectives. Contrary to forward design, which derives properties from predetermined materials, inverse design initiates with an end goal, rendering it a potent methodology for pioneering material innovation. Traditionally, this procedure was predominantly dependent on empirical insights, iterative experimentation, and deterministic optimization techniques. The notion of inverse design in mechanical materials is historically established, originating across diverse engineering fields with the objective to surmount the constraints posed by forward design methodologies. In its early stages, these strategies were limited by the computational resources at hand, centering on rudimentary models and heuristic methods to investigate potential material configurations.

In the realm of mechanical material inverse design, traditional methodologies primarily focused on optimization techniques and empirical modeling. Key tools for navigating the material design space included optimization algorithms like gradient-based methods [63], genetic algorithms [64], and simulated annealing [65]. These methods were augmented with empirical [66] and semi-empirical [67] models, which were instrumental in forecasting material responses under a spectrum of conditions.

Inverse design has found application across multiple domains within mechanical materials, including: (1) composite materials [68], where the focus is on devising composites with ideal fiber orientations and volume fractions to attain specific mechanical properties. (2) Metamaterials [69],

aimed at fabricating materials with custom microstructures to realize characteristics unattainable in natural substances, such as negative Poisson's ratio or customized thermal expansion coefficients. (3) Structural components [70], involving the creation of components with optimized geometries and materials to meet precise requirements for load bearing, vibration mitigation, or thermal management.

Key challenges encountered in the traditional approach to mechanical material inverse design include computational complexity, characterized by the significant computational expense required for simulating and optimizing intricate material systems. Limited search space exploration, where conventional techniques may not thoroughly probe the extensive design landscape, thereby possibly overlooking optimal configurations. Dependency on material models, indicating that the precision of inverse design results is greatly reliant on the veracity of the employed empirical and semi-empirical models.

2.2 Machine Learning for Material Design

Initial efforts in applying ML to material design were concentrated on the straightforward prediction of properties using datasets of previously characterized materials. These early studies employed regression models [71] and decision trees to predict various properties, such as melting points [72], stability [73], and electronic configurations [74], demonstrating ML's capability to significantly influence material discovery and design.

Subsequently, the field has witnessed rapid advancements with the adoption of more complex ML models, including as DL, GNNs, and generative models. Deep learning has revolutionized the processing of complex, hierarchical data prevalent in materials science. GNNs have gained popularity for their efficacy in modeling materials that can be depicted as graphs, like molecules

[75] and crystals [76]. Generative models have been utilized to design new molecules and materials by learning the distribution of existing materials and generating novel instances with desired characteristics.

A pivotal development in this domain is the application of active learning and Bayesian optimization techniques, which iteratively refine models and experimental designs. These methods have optimized the exploration of extensive material spaces, efficiently pinpointing promising candidates for synthesis and testing.

Machine learning's application extends across various materials science domains, including energy storage [77], catalysis [78], photovoltaics [79], and pharmaceuticals [80], showcasing its versatility in designing high-capacity battery materials [81], discovering efficient catalysts [82], developing new solar cell materials [83], and accelerating drug discovery [84].

Despite considerable advancements, challenges such as data scarcity, dataset quality, and the interpretability of ML models persist. The materials science field often grapples with the absence of large, high-quality datasets, hampering the efficacy of data-driven approaches. Furthermore, the opaque nature of certain ML models, especially deep neural networks, presents hurdles in comprehending and trusting their predictions. Future research will likely concentrate on crafting more interpretable ML models, fostering collaborations with computational chemistry and physics to generate data efficiently, and enhancing unsupervised and semi-supervised learning techniques to better utilize unlabeled data. Another promising avenue involves amalgamating simulation and experimental data to forge hybrid models that more accurately predict material properties and inform the design process.

2.3 Machine Learning and Mechanical Material Inverse Design

This interdisciplinary approach aims to enhance the precision, efficiency, and innovation in designing materials with desired mechanical properties. The review will cover the evolution, key contributions, methodologies, applications, challenges, and prospects of integrating ML into mechanical material inverse design.

The incorporation of machine learning into the inverse design of mechanical materials represents a significant paradigm shift. Traditional inverse design methods, while powerful, often face limitations in terms of computational efficiency and the ability to navigate complex, high-dimensional design spaces. Machine learning, with its ability to learn from data and make predictions, offers a promising solution to these challenges, facilitating a more refined search for optimal material configurations.

The integration of ML into this field has evolved rapidly in recent years, fueled by advancements in computational power, algorithmic development, and the availability of large datasets. Early contributions focused on leveraging regression models and neural networks to predict material properties from compositional and structural parameters. Recent advancements have seen the adoption of more sophisticated models, as we mentioned, CNNs have been developed for image recognition, while generative models such as GANs and VAEs are employed in generative tasks.

Machine learning models, particularly supervised learning algorithms, have been extensively used to predict the properties of materials based on their composition and structure. These models can significantly reduce the need for exhaustive experimental or computational property evaluations.

Generative models have been applied to generate new material designs. These approaches can explore the design space more effectively, proposing novel materials that fulfill specified performance criteria. Applied into topology and optimization of mechanical materials [85] by using reinforcement learning and VAEs.

Combining ML models with optimization algorithms and active learning strategies enables efficient navigation of the design space. This synergy focuses experimental and computational resources on the most promising areas, accelerating the discovery process.

The integration of machine learning into the inverse design of mechanical materials offers a promising avenue for accelerating the discovery and optimization of novel materials. By enhancing traditional design methodologies with the predictive and generative capabilities of ML, researchers and engineers can tackle complex design challenges more efficiently and innovatively. As the field continues to evolve, interdisciplinary collaboration will be key to unlocking the full potential of this approach, leading to breakthroughs in material science and engineering.

And machine learning based mechanical inverse design have challenges. (1) Complexity of mechanical systems, mechanical systems often involve intricate dynamics, non-linear behaviors, and multi-physics interactions can be challenging due to the high-dimensional and complex of the data. (2) Data availability and quality, machine learning models require large amounts of high-quality data for training. In mechanical inverse design, obtaining sufficient and relevant data can be difficult, especially for complex applications. (3) Interpretable, understanding the features and representations that are meaningful for mechanical design can be important, but have difficult in machine learning based process. (4) Physical constraints and validity, mechanical designs must adhere to physical laws and constraints. Ensuring that machine learning models produce designs that are physically valid and feasible requires careful integration of domain knowledge into the learning process. (5) Transferability, machine learning models trained on specific datasets may struggle to generalize to new scenarios or transfer to different applications. This limits their adaptability and scalability in mechanical inverse design tasks. (6) Computational efficiency, design optimization often requires evaluating many design configurations. Machine learning

models used in inverse design must be computationally efficient to handle such evaluations within time challenge.

CHAPTER THREE

ESTABLISHING THE RELATIONSHIP BETWEEN MATERIAL MICROSTRUCTURES AND MECHANICAL PROPERTIES

3.1 Data Preparation

The goal of material property prediction is to predict porous material's mechanical properties using the materials' microstructural images. Such images are routinely obtained by using micro-imaging techniques such as optical microscopy (OM) [86], scanning electron microscopy (SEM) [87], and Electron Backscatter Diffraction (EBSD) [88] etc. For microstructure generation, given the target material properties as the input, the objective is to generate microstructural configurations with which the porous material exhibits the required material properties. In both cases, it is sufficient to represent the 3-D microstructure of the porous materials by using multi-layer binary images similar to CT scans. Due to the lack of large scale scanned images necessary for the development of machine learning models, in this work, we generate a large set of computer images of porous materials with various porosity and microstructure morphology by cutting out pores from a solid material. In addition, without losing generality, we focus on 2-D microstructures in this study. The extension of the work to 3-D is straightforward. The development of the neural networks involves three major steps: generation of the input data, construction of the neural network architecture, and training and verification. In this section, the steps for constructing the neural networks to achieve the functions described above are presented as follows.

3.1.1 Data Generation

Three input data sets are necessary for neural network development and training. The first data set contains microstructure binary images of porous materials. Associated with each of the

microstructure images are the porosities of the materials which are stored as elements of the second data set. The third data set contains the mechanical properties such as elastic stiffness or stress-strain curves. All the data sets are generated computationally as described below.

3.1.2 Microstructure of Porous Material

The microstructure data is generated by creating computer images in a representative domain. The initial domain is set to be a solid square domain. The domain is discretized into 100×100 square elements. The pores of the material are generated by cutting out blocks of elements at random locations of the domain. The elements that are cut out are considered as belonging to pore regions and remaining elements are solid materials. The elements of pore regions are assigned value "0", and those occupied by the solid material are assigned value "1". Thus the microstructure of the material is represented by the 100×100 binary image. To control the porosity, the volume fraction of the pores, i.e. the porosity, is calculated after each cut as $\varphi = V_p/V_b$ where Vp is the volume of the pores and Vb is the volume of the solid domain. The cutting stops when the desired porosity is reached. To control the size of the pores as well as the feature size of the material microstructure, different sizes of the cutting block are used. In this work, the set of cutting block sizes used are 2×2, 3×3, 4×4, 5×5, and 6×6. It should be noted that, to maintain the integrity of the domain and avoid material "breaking away", a boundary region measured 5% of the length/width of the domain is kept intact in the cutting process. After the pore cutting process, the porosity of the material φ is recorded. Through the cutting process, two data sets, including a set of binary matrices representing the microstructures of the porous materials and the corresponding porosity set, are produced. Examples of the generated porous material domains are shown in Fig. 1. In the preparation of the microstructure image sets, we produce a uniform distribution of the porosity

from 1% to 60%. As shown in Fig. 1, when the porosity is larger than 60% pore regions become dominant, and the center part of the domain becomes mostly a large pore. In that case, the effect of microstructural morphology diminishes, and the mechanical properties of the material are mostly determined by the porosity. For this reason, we only focus on the porosities less than 60%. To ensure a uniform distribution, the range of the porosity under consideration is divided into 60 equal-size bins, each contains the same number of microstructure samples. In addition, each bin of porosity contains the same number of microstructure images with each of the four cut sizes. In this work, a total of 120,000 of microstructure images are generated for the development of the neural networks (i.e. 2000 samples in each porosity bin), among which 80% of the images are used for training, 20% are used for testing.

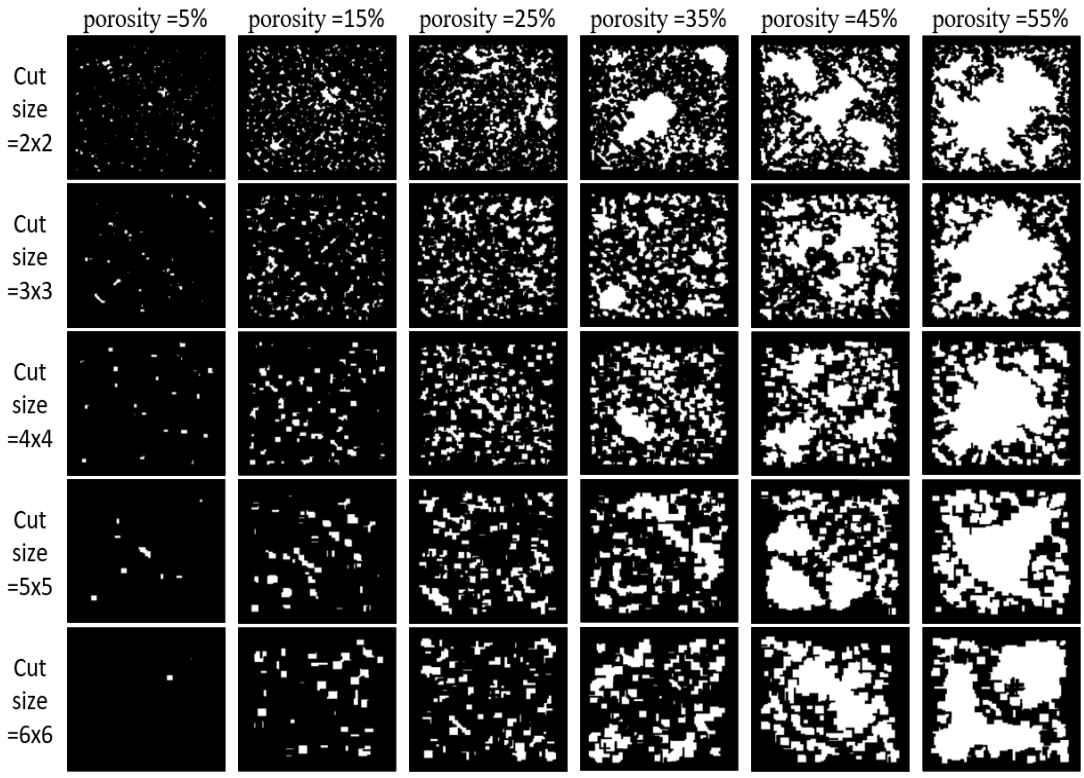


Figure 1 Computer generated porous material microstructures.

3.1.3 Elastic Moduli and Nonlinear Stress-Strain Curves

Once the representative microstructure images are produced, the porous materials' mechanical properties are obtained by using finite element analysis (FEA). In this work, we consider two different scenarios: linear and nonlinear porous materials subjected to linear small elastic deformation and nonlinear large elastic deformation, respectively. In the linear scenario, the constituent (or base) material is linear elastic and deformation is small. We focus on the tensile elastic modulus or Young's modulus of the porous material. The elastic modulus Ep of the porous material is obtained as $E_p = \sigma/\epsilon$, where σ and ε are the normal stress and strain, respectively. The finite element model setup is shown in Fig. 2 (a) where a unit thickness 2-D representative material is roller-supported at the left and bottom edges and a horizontal displacement is applied to the right edge. Note that the left and bottom sides of the representative material is fixed in x- and y-directions, respectively. The top edge is kept free. The stress σ is calculated as the reaction force on the left edge divided by the cross-section area of the material at the left edge. The normal strain ε is obtained as the applied horizontal displacement divided by the original length of the material. To make the machine learning models independent from the constituent material, the elastic modulus of the solid constituent material (porosity=0) is scaled to 100. The obtained porous materials' elastic modulus values are normalized against the solid material's modulus, which is referred to as the normalized elastic modulus. For example, elastic modulus of 50 obtained from a given porous material means it is 50% of the solid constituent material's modulus. The normalized Young's modulus is also denoted as Ep for convenience. For the nonlinear case, the constituent material is assumed to be hyperelastic and subjected to large deformation (strain >100%). While there are many hyperelastic material models, without losing generality, the five parameter

Mooney-Rivlin model [126] is employed in this study to describe the material behavior. The strain energy is given by

$$W = c_{10}(\bar{I}_1 - 3) + c_{01}(\bar{I}_2 - 3) + c_{20}(\bar{I}_1 - 3)^2 + c_{11}(\bar{I}_1 - 3)(\bar{I}_2 - 3) + c_{02}(\bar{I}_2 - 3)^2 \quad (1)$$

where $\bar{I}_1, \bar{I}_2$ are the first and second invariants of the deviatoric part of the right Cauchy-Green deformation tensor, respectively. In this work, a representative set of the five parameters is used for the constituent material: $c_{10} = -1.4516, c_{01} = 1.8669, c_{20} = 1.5083, c_{11} = -4.3864, c_{02} = 3.8062$, all in terms of unit stress. Note that, for generality the unit stress is simply defined as (unit force)/(unit length)2 to nondimensionalize the analysis. The actual units can be chosen depending on the specific materials. For the nonlinear large deformation case, the FEA is set up the same way as shown in Fig. 2(a). The target material property is the stress-strain curve obtained by calculating a set of average stresses at different strains. In this study, the maximum strain is set to be 120%, as depicted in Fig. 2(b).

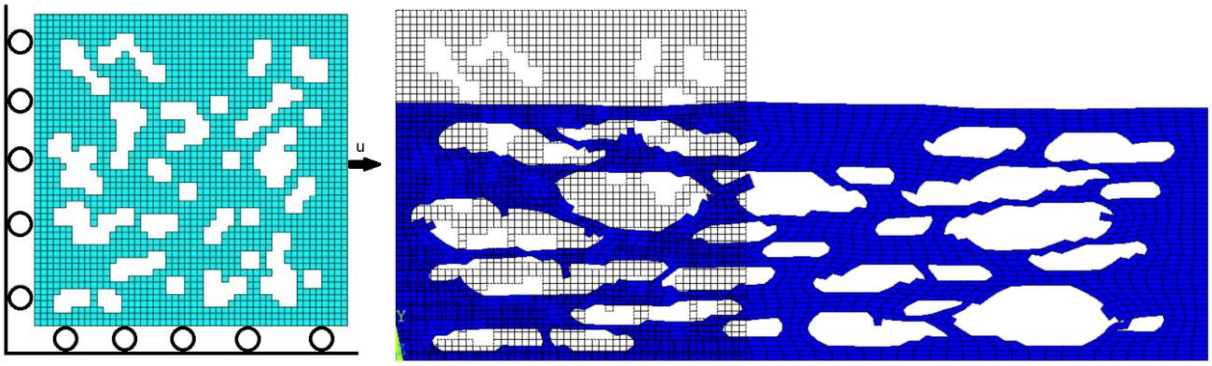


Figure 2 (a) Setup of tensile deformation FEA; (b) Large deformation of a hyperelastic porous material.

The FEA results show that, while the tensile stress-strain curves can be very different depending on the microstructure of the hyperelastic porous material, they can be closely approximated by using a cubic function as,

$$\sigma = b_0 + b_1\epsilon + b_2\epsilon^2 + b_3\epsilon^3 \qquad (2)$$

where σ is the axial stress, ϵ is the strain, and b0, b1, b2, b3 are the coefficients. Note that, since the curves all go through the origin point, b0 is always zero. Therefore, each stress-strain curve is represented by a 3×1 vector containing the coefficients b1, b2 and b3. For implementation convenience, as discussed in Section 2.3, it is necessary to normalize the coefficients such that their values have a range spanning tens to hundreds of integers. Many normalization schemes can be used for this purpose. In this work, we implement a simple normalization scheme as follows. First, a normalized displacement is defined as $\overline{u} = u/(sL)$, where u is the actual displacement, L is the length and height of the representative material, and s is a scaling factor for adjusting the value range of the normalized coefficients. In addition, a dimensionless normalized force $\overline{F}$ is defined as $\overline{F} = F/F_0$, where F_0 is the unit force. Next, by using the stress and strain definitions, $\sigma = F/(Lt)$ and $\varepsilon = u/L$, Eq. (2) can be rewritten as

$$\overline{F} = \frac{b_1 Lts}{F_0}(\overline{u}) + \frac{b_2 < s^2}{F_0}(\overline{u})^2 + \frac{b_3 < s^3}{F_0}(\overline{u})^3 = a_1\overline{u} + a_2\overline{u}^2 + a_3\overline{u}^3 \qquad (3)$$

where is $\overline{F}$ the normalized tensile force, t is the thickness of the material. Thus, the stress-strain function shown in Eq. (2) is transformed to the normalized force-displacement equation shown in Eq. (3) in which the dimensionless coefficients are $a_1 = b_1 Lts/F_0$, $a_2 = b_2 < s^2/F_0$, and $a_3 = b_3 < s^3/F_0$. With this transformation, the new coefficients are dimensionless with a proper range for the machine learning model. For the porous material under consideration, s is set to be 0.4 and t is unit length for all the calculations. In summary, for each microstructure configuration and loading

condition, three data sets are generated and stored: (1) an L×L binary image representing the material microstructure, (2) the porosity, and (3) the material property of the material, which is the elastic modulus for the linear case and the coefficient vector for the nonlinear case.

3.2 Neural Network for Prediction

Among the existing neural networks suitable for image classification, convolution neural networks (CNN) are mostly formulated as a variant of feedforward neural networks [89]. Typically, a CNN turns an image into a layer with a filter or detector, along with pooling and normalization layers allowing for data simplification. They are suitable for image processing and recognition patterns in visual data [90, 91], and have been applied to material texture recognition [92]. ResNet is a deep residual learning framework that combines residual connection and inception architecture to overcome CNN's accuracy degradation problem [93]. Since the problem under consideration is essentially an image recognition and characteristic prediction problem, ResNet based methods are suitable choices. In this study, for predicting the elastic modulus of the linear elastic porous material, a ResNet-18 neural network is constructed and trained.

The plain network structure is based on VGG nets. The convolution layers typically have 3×3 filters, in our model, we continuously take 3x3 as our filter size, the limited sized filters kind of consistently catch and focus the local details and relationships of microstructure. For the same output feature map, the layers have the same number of filters. If the size of the feature map is halved the number of filters is doubled to preserve the time complexity of each layer. Based on the above plain network, a shortcut connection is inserted which turns the network into its counterpart residual version as shown in Fig. 3. The identity shortcuts $F(x(W) + x)$ can be directly used when the input and output are of the same dimensions. The attributes of VGG net and residual shortcuts

constitute the Resnet help the model remember and understand the local detailed of porous microstructure and not ignore the global attributes of the morphology. The material property of the material is global porous material properties based on the local details of the porous material structure, the model based on Resnet matches the goal we pursuit well.

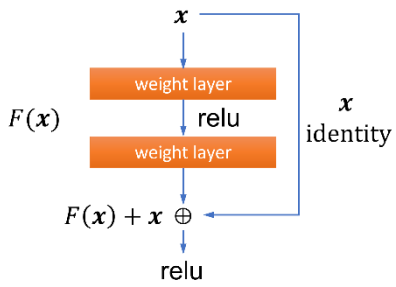


Figure 3 Basic ResNet architecture.

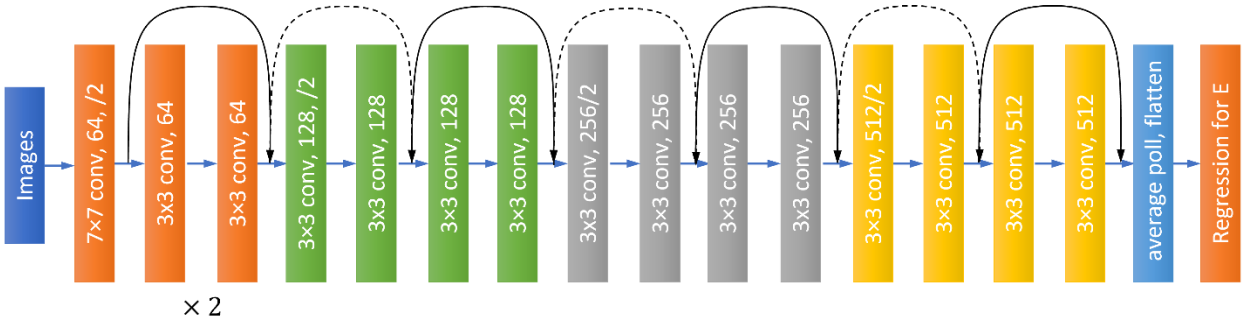


Figure 4 ResNet-18 architecture for predicting Young's modulus Ep .

3.2.1 Neural Network for Elastic Modulus Prediction

For Young's modulus prediction, we follow the ResNet-18 architecture as shown in Fig. 4, which is composed of 18 layers of residual neural networks, including 5 layers of 64 filters convolution layers, 4 layers of 128 filters convolution layers, 4 layers of 256 convolution layers, 4 layers of 512 convolution layers, 1 layer of average pooling, and 1 layer of regression. As discussed in Section 2.1, the microstructures of the porous material are generated by randomly cutting square

shaped pores in the solid domain using five different cutting sizes (2×2, 3×3, 4×4, 5×5, 6x6). For the ResNet-18 network's training and testing (ratio = 80%/20%), a total of 120,000 microstructures are generated uniformly over the porosity range of 1% - 60%. It is found that for this porosity range, the normalized Young's modulus varies from 12 to 100. For Young's modulus prediction, we use the smooth L1 loss function as it is less sensitive to outliers compared to the mean square error and, in some cases, can prevent exploding gradients. The loss function is given by

$$loss(x, y) = \frac{1}{n}\sum_{i=1}^{n} z_i(x_i, y_i) \tag{4}$$

and

$$z_i = \begin{cases} 0.5(x_i - y_i)^2/\beta, \wedge\, if|x_i - y_i| < \beta \\ |x_i - y_i| - \beta/2, \wedge\, otherwise \end{cases} \tag{5}$$

where β is set to be 1.0. In the regression step, the regression vector V is initialized as a 1-D vector containing the values of Young's modulus ranging from 12 to 100. Therefore, the length of the regression vector is 89, representing the integer values in the range. The regression value is computed as V·x, where x is the result from the last step of the ResNet-18 network. The regression layer is refined to output Young's modulus from 12 to 100. It is treated as an extension of classification in regression mode, putting the different Young's modulus into the appropriate integer bin.

3.2.2 Neural Network for Nonlinear Stress-Strain Curve Prediction

For the nonlinear case, the goal is to predict the stress-strain curve for a given microstructure of the hyperelastic porous material. Different from the linear material case where the output is one scalar material property (Young's modulus Ep), the nonlinear stress-strain curve is represented by a cubic polynomial. The output curve is described by three nonzero coefficients of the polynomial.

In this case, with a modification of the regression step to accommodate the three scalar output quantities, the same ResNet-18 is used to predict the stress-strain curves. The network is shown in Fig. 5. The same microstructure image set is used for the nonlinear material case. After the stress-strain curves are computed by using FEA, each microstructure image is associated with three coefficients characterizing the stress-strain curve. Once again, 80% of the microstructure data sets are used for training, and the rest for testing.

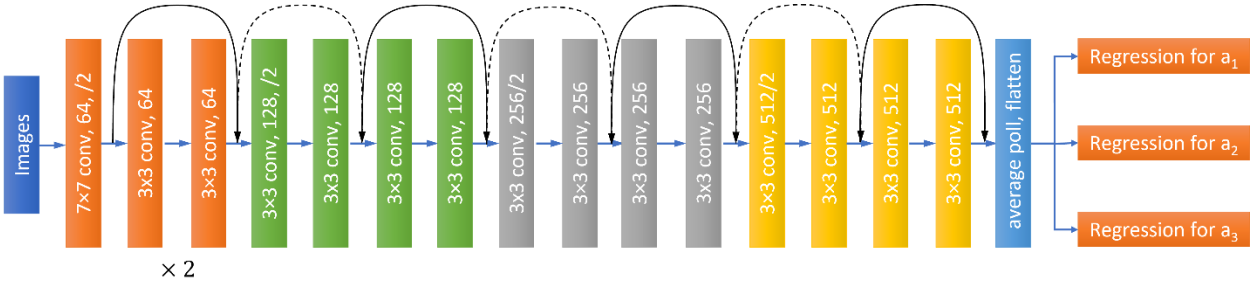


Figure 5 ResNet-18 for predicting stress-strain curves.

In the selection of the loss function, it is observed that, compared to the L1 loss function, ResNet-18 achieves a better performance for predicting nonlinear stress-strain curves if more weight is given to the outliers. Therefore, we adopt the MSE loss function for the nonlinear case:

$$MSE(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{6}$$

In the regression step, three regression vectors, V1, V2, V3, are set up using the integer values of the three stress-strain curve coefficients. From the polynomial fitting results, the ranges of the coefficients a1, a2, a3 are observed to be [5, 35], [-40, -5] and [0, 15], respectively, and the lengths of the three regression vectors are then 16, 36 and 31, respectively. The regression values are Vi·x, i=1,2,3, and x is the result from the last step of the ResNet-18 network.

3.3 Results

3.3.1 Elastic Modulus Prediction

We first demonstrate the performance of the ResNet-18 network for Young's modulus prediction. For the 120,000 microstructure samples generated and uniformly distributed in the porosity range of [1%, 60%], the variation of the normalized Young's modulus as a function of the porosity is shown in Fig. 6. It is shown that the range of Young's modulus is between 100 and 12. At the intermediate porosities, the elastic modulus of the porous materials with the same porosity exhibits large variations. The variation decreases when the porosity becomes smaller or larger. That is, the variation becomes quite small when the porosity is either very small (nearly a dense solid) or very large (nearly a hollow structure). This implies that the influence of porosity on the elastic modulus is more than that of the microstructure configuration (morphology of the pores) at small and large porosities. However, the influence of the microstructure becomes more significant at the intermediate porosities and reaches its peak at porosity around porosity = 18%. As shown in Fig. 6, for the same porosity of 18%, depending on the morphology of the pores in the material, the elastic modulus can vary from 19 to 49. Figure 7 shows Young's moduli of 24,000 random image samples predicted by using the ResNet-18 shown in Fig. 4 in comparison with the FEA results which are considered as the true values of Young's modulus. The average MSE of the predictions is 0.00386. It is evident that the Young's modulus variation due to the combination of the porosity and microstructure morphology is captured by the ML model. Note that the total training epochs is 250.

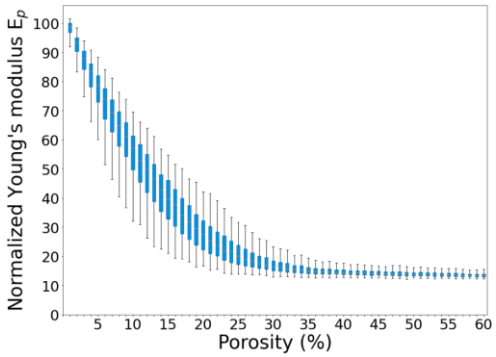


Figure 6 Elastic modulus variation as a function of porosity.

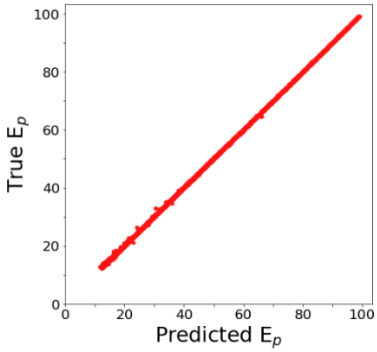
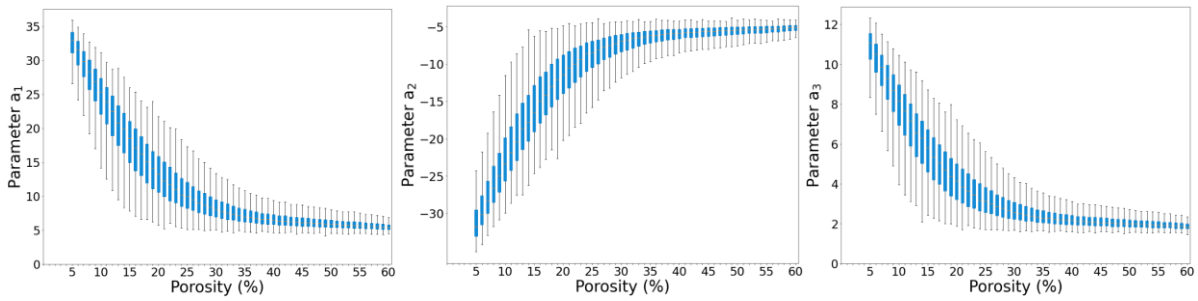


Figure 7 Predicted vs. true Young's modulus (24,000 samples).

3.3.2 Nonlinear Stress-Strain Curve Prediction

The variation and distribution of the three coefficients of the cubic stress-strain curves as functions of the porosity of the hyperelastic porous material are shown in Fig. 8. Similar to the elastic modulus, the stress-strain curve coefficients exhibit smaller variations at the two ends of the porosity axis, although their trends are different. The prediction results obtained from the ResNet-18 network shown in Fig. 5 in comparison with the FEA results are shown in Fig. 9. While the results of the predicted coefficients are slightly less accurate compared to the Young's modulus predictions, the resultant nonlinear stress-strain curves of the material match quite well with the FEA results. This is because the stress-strain curve is determined by the cubic polynomial as a whole and is less sensitive to the value of individual coefficients. Figure 12 shows an example with the binary microstructure image on the left and the stress-strain curve comparison on the right. The average MSEs of a_1, a_2 and a_3 are 0.0222, 0.0176, 0.0208 respectively. The total training epochs for this case is 250.



(a) (b) (c)

Figure 8 Variation and distribution of the stress-strain curve coefficients as functions of porosity. From left to right: a1, a2 and a3.

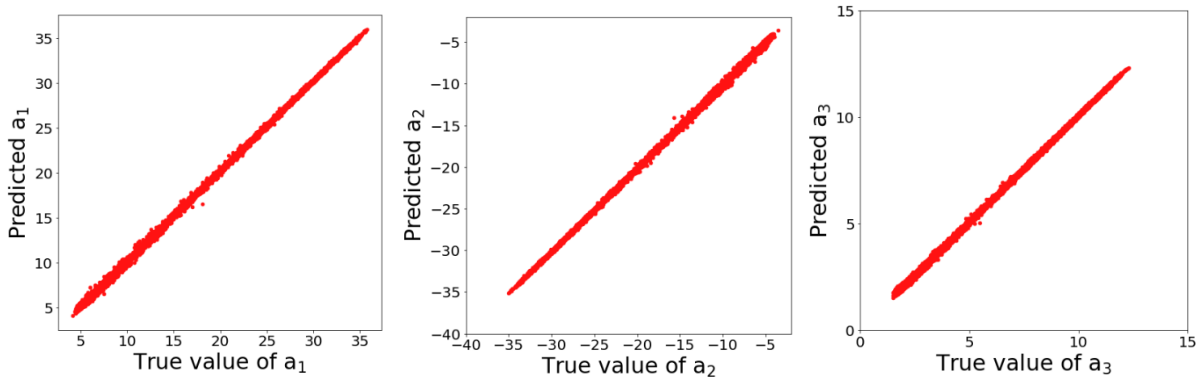
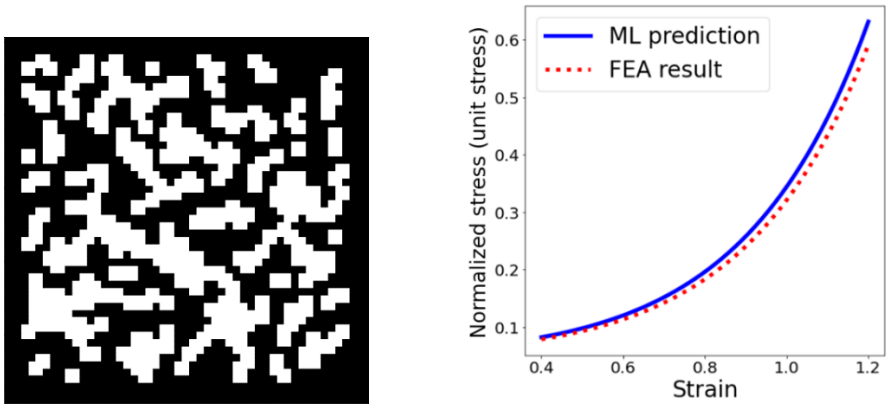


Figure 9 Predicted vs. true values of the coefficients of the cubic stress-strain curves (24,000 samples). From left to right: a1, a2 and a3.



(a) (b)

Figure 10 An example of predicted stress-strain curve for a given hyperelastic porous microstructure in comparison with the FEA result. Left: given porous microstructure. Right: stress-strain curve comparison.

3.4 Discussion

In this section, we propose a machine learning approach to predict Young's modulus and stress-strain curves for linear and nonlinear porous materials, respectively, and to generate porous microstructures based on given linear and nonlinear material properties. In our neural networks, the microstructure of the porous materials is represented using binary images. Pores are generated by randomly removing element blocks of different sizes from the material domain. For predicting material properties, we develop ResNet-18 neural networks to predict the normalized Young's modulus of porous materials. These ResNet-18 networks are trained on 120,000 image samples, and optimal hyperparameters are determined. Our results demonstrate that the predictions of Young's modulus are highly accurate, capturing the variations due to the combination of porosity and microstructure morphology effectively.

CHAPTER FOUR

2D MICROSTRUCTURE GENERATION FOR LINEAR AND NONLINEAR TARGET MATERIAL PROPERTIES

4.1 Generative Machine Learning Models

4.1.1 Microstructure Generation Model for Given Elastic Modulus

While the machine learning models discussed above enable a fast mechanical property evaluation of a given porous material based on its microstructure, in this section, we develop machine learning models that will enable efficient design of the microstructure to achieve given desirable mechanical properties. For such inverse design problems, generative neural networks are good candidates. As discussed in Section 1, there are three main frameworks for generative models: variational autoencoders (VAE), reinforcement learning (RL), and generative adversarial networks (GAN). The major challenge in microstructure generation through GAN is the mismatch between the generator's continuous output and the actual binary image, making the adversarial training difficult to succeed. For RL, it is difficult to design and implement an appropriate reward function for the complex binary microstructures. Another challenge is maintaining consistency among the microstructures generated by the trained RL model. In comparison, VAE models encode the global distribution of the microstructures, making it more suitable to match the attributions of the microstructure data. For this reason, in this work we choose the framework of VAE for microstructure generation.

VAEs simultaneously train a generative model $p_\theta(X, z) = p_\theta(X \vee z)p_\theta(z)$ for the input x using latent variable z, and an inference model $q_\phi(z|X$ by optimizing the evidence lower bound (ELBO). For microstructure generation based on a given elastic modulus (linear case), we construct a conditional VAE (cVAE) with the conditional distribution $p(X \vee z, y)$, where X is the

microstructure image, y is the elastic modulus and z is a latent variable. We can use deep neural networks (parametrized by θ) to approximate $p(X \vee z, y)$. The ELBO for the conditional VAE can be written as:

$$L(\theta, \phi; X, y)_{VAE} = -KL\left(q_\phi(z|X) \parallel p_\theta(z)\right) + E_{q_\phi(z|X)}[logp_\theta(X \vee z, y)] \qquad (7)$$

where the Kullback-Leibler divergence $KL(q_\phi \parallel p_\theta)$ is used to quantify the distance between two distributions. The encoder (parametrized by ϕ) is to encode the input *x* into a latent variable *z*, and the decoder (parametrized by θ) is used to reconstruct the original input *x* using sample from the latent distribution and the corresponding elastic modulus *y*.

There are two main challenges for the conditional generation of microstructure images. First, it is difficult to integrate the elastic modulus, which is a scalar, into the generator as a condition to control the generation of the corresponding microstructure. Second, the Encoder-Decoder pair (from *X* to *z*, then *y* and *z* back to $X^{'}$) tends to focus on local details at the pixel level to minimize the reconstruction loss and ignores the global semantic information and some key pixels, such as areas of high stress. For the binary microstructure images, it is necessary to construct an efficient model to capture the specific geometrical patterns and ensure the generated images have a reasonable physical meaning. In this work, we extend the cVAE model shown in Eq. (8) and construct a property-to-microstructure generation model to address the above challenges. First, we introduce a fusion layer to integrate conditional information *y* with the latent variable *z*. While this effectively alleviates the problem of condition integration, the resultant Young's modulus of the generated images does not match well with the given conditions. To achieve better conditional generation, we impose a stronger constraint ($\beta > 1$) on the latent variable *z* to limit the representation capacity of *z* [94]. When the latent variable *z* and the condition *y* are provided together to the decoder for image reconstruction, a higher β enforces *z* to be more concerned with

capturing the microstructural information, such as the size and distribution of the pores. Therefore, the encoder projects the image to the latent variable z that is largely independent from the scalar value of Young's modulus, which is more conducive to the conditional generation of the image. To address the second challenge, we introduce a pretrained regression model to form the decoder-regression pair (from y and z to X', then back to y'). Similar to the discriminator in GAN [95], the pretrained regression model constrains the decoder so that it generates more physically meaningful images. Meanwhile, the regression model also allows the decoder to generate images that better match the conditions. The new conditional generation model with the scalar Young's modulus value as the condition is shown in Fig. 11. The objective function of our framework can be written as:

$$L(\theta, \phi; X, y)_{VAE} = \beta KL\left(q_\phi(z|X) \parallel p_\theta(z)\right) - E_{q_\phi(z|X)}[logp_\theta(X|z, y)] + \alpha L_R \quad (8)$$

where the regularization coefficients β and α are used to balance the weights of KL loss and regression loss L_R. The framework is successfully applied to the generation of binary microstructure images with Young's modulus as the condition. As shown in Fig. 11, the encoder maps the input binary images X to the latent variables z. The fusion layer is used to integrate the information from the condition y and latent variable z. The decoder decodes the output of fusion layer to binary microstructure images. Then the pretrained regression model predicts Young's modulus given the reconstructed image. During the model training, the pretrained regression model is frozen and kept the same.

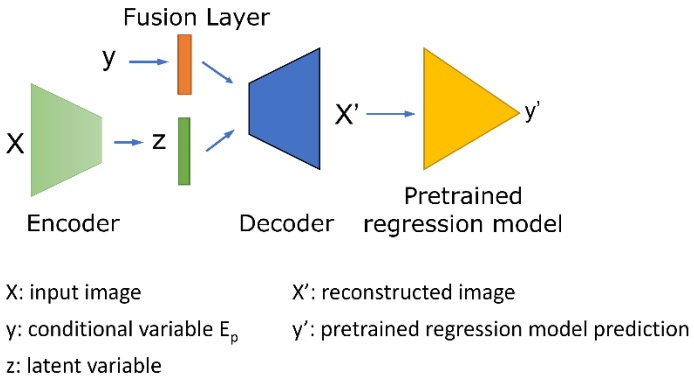


Figure 11 Neural network architecture of the property-microstructure generation model.

Once again, the dataset is split into training (80%) and testing (20%) datasets. Each sample is a $1 \times 100 \times 100$ binary image and is paired with a ground-truth elastic modulus. We implement the conditional property-microstructure generation model based on the Ladder VAE [96, 97], which allows interaction between the bottom-up and top-down layers and provides better log-likelihood lower bound compared to the conventional VAE. These two VAE models have the identical generative models, the main difference is in the inference models. Since the input image is binary, the Binary Cross Entropy is chosen for the reconstruction loss. We apply the L2 loss as the regression loss L_R. We also employ the batch-normalization and dropout scheme (dropout probability: 0.2) which are important for training models with many stochastic layers. The coefficient β is set to 3 to place a stronger constraint on the latent bottleneck. α is set to 30,000 to balance the regression loss term. After training, we randomly sample each pixel from a uniform distribution to generate the binary image. Finally, a post-processing step is performed to remove the isolated material islands in the generated binary images by using the breadth-first search (BFS). In the microstructure generation model, the encoder and the decoder are composed of basic residual blocks shown in Table 1. To be specific, the encoder includes 5 ResidualBlock blocks and

the decoder includes 8 ResidualBlock blocks. The total number of trainable parameters is 1,417,537. We use the Sigmoid activation function in the final output layer of the decoder. For the fusion layer, a trainable linear layer is used to map Young's modulus (scalar value) to the same dimension as the latent variable, which is similar to the word embedding layer. The difference is that word embedding uses indices to retrieve the corresponding embeddings. We obtain desirable results by directly summing the transformed Young's modulus and the latent variable. The regression model is the ResNet-18 shown in Fig. 4. To train the generative model, we choose the Adamax optimizer, and set the batch size to 64. The learning rate is set to 3e-4. The total training epochs is 500.

Table 1 Modules and their parameters for each ResidualBlock

Modules	Parameter
BatchNorm2d	num_features=64, eps=1e-05, momentum=0.1, affine=True
ELU (activation)	alpha=1.0
Conv2d	in_channels=64, out_channels=64, kernel_size=(3,3), stride=(1,1), padding=(1,1)
Dropout2d	p=0.2, inplace=False
BatchNorm2d	num_features=64, eps=1e-05, momentum=0.1, affine=True
ELU (activation)	alpha=1.0
Conv2d	in_channels=64, out_channels=64, kernel_size=(3,3), stride=(1,1), padding=(1,1)

Dropout2d	p=0.2, inplace=False

4.1.2 Microstructure Generation Model for Given Stress-Strain Curves.

In this section, we construct a machine learning model for generating microstructures of a hyperelastic porous material to exhibit a given stress-strain curve. Different from the linear porous material characterized by a single scalar Young's modulus, the stress-strain curves of the hyperelastic porous materials can be described by using a cubic function with three coefficients, as discussed in Section 2.2. While the descriptor of mechanical deformation behavior is changed from a scalar to a vector, the microstructure generation considerations are the same as the generation for Young's modulus. Therefore, the microstructure generation framework described in Section 2.4 is adopted for the nonlinear stress-strain relationship and the cVAE based model shown in Fig. 11 is extended to include the stress-strain curve coefficient vector as the condition in the fusion layer and the Resnet-18 shown in Fig. 5 as the pretrained regression model. The architecture of the extended model is shown in Fig. 12. The objective function becomes:

$$L(\theta, \phi; X, a)_{VAE} = \beta KL\left(q_\phi(z|X) \parallel p_\theta(z)\right) - E_{q_\phi(z|X)}[log p_\theta(X|z, a)] + \alpha L_R \quad (9)$$

where a=$[a_1, a_2, a_3]^T$. In Eq. (9), the regression loss L_R employs the relative absolute error between two vectors. It is observed that β=3 and α=100,000 give a better balance of the loss terms in the nonlinear deformation generation case. The encoder and decoder blocks, activation function, learning rate, batch size, and all the other parameters of the models are the same as the Young's modulus based microstructure generation model. All the hyperparameters and training strategies are also the same as given in Section 2.4.

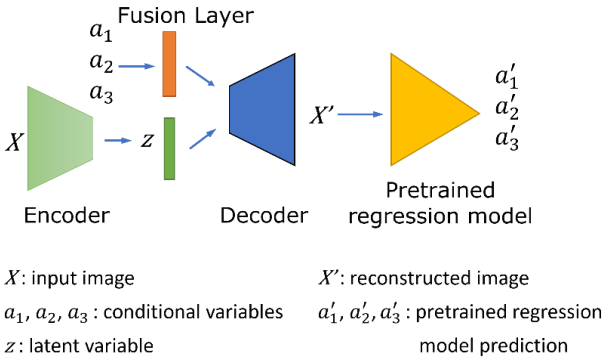


Figure 12 The microstructure generation model for generating hyperelastic material microstructures for given stress-strain curves.

4.2 Results

4.2.1 Microstructure Generation Based on Elastic Modulus

To evaluate the performance of microstructure generation, we choose eight different Young's modulus conditions ranging from 15 to 80 and generate 128 samples for each condition to calculate mean and standard deviation of predicted Young's modulus. The results are shown in Table 2. Figure 18 shows examples of generated microstructures: each column contains eight different generated microstructures corresponding to the same Young's modulus denoted at the top. The results demonstrate that the model is capable of producing microstructures that satisfy the mechanical property requirement and providing microstructural "designs" with considerable morphology variations.

Table 2 Mean and standard deviation of predicted elastic modulus from 15 to 80

Condi-tion	15	20	30	40	50	60	70	80

Mean	15.55	18.88	28.93	39.94	50.57	60.97	70.86	80.98
Std Dev	1.13	1.43	1.75	1.48	1.41	1.31	1.18	1.05

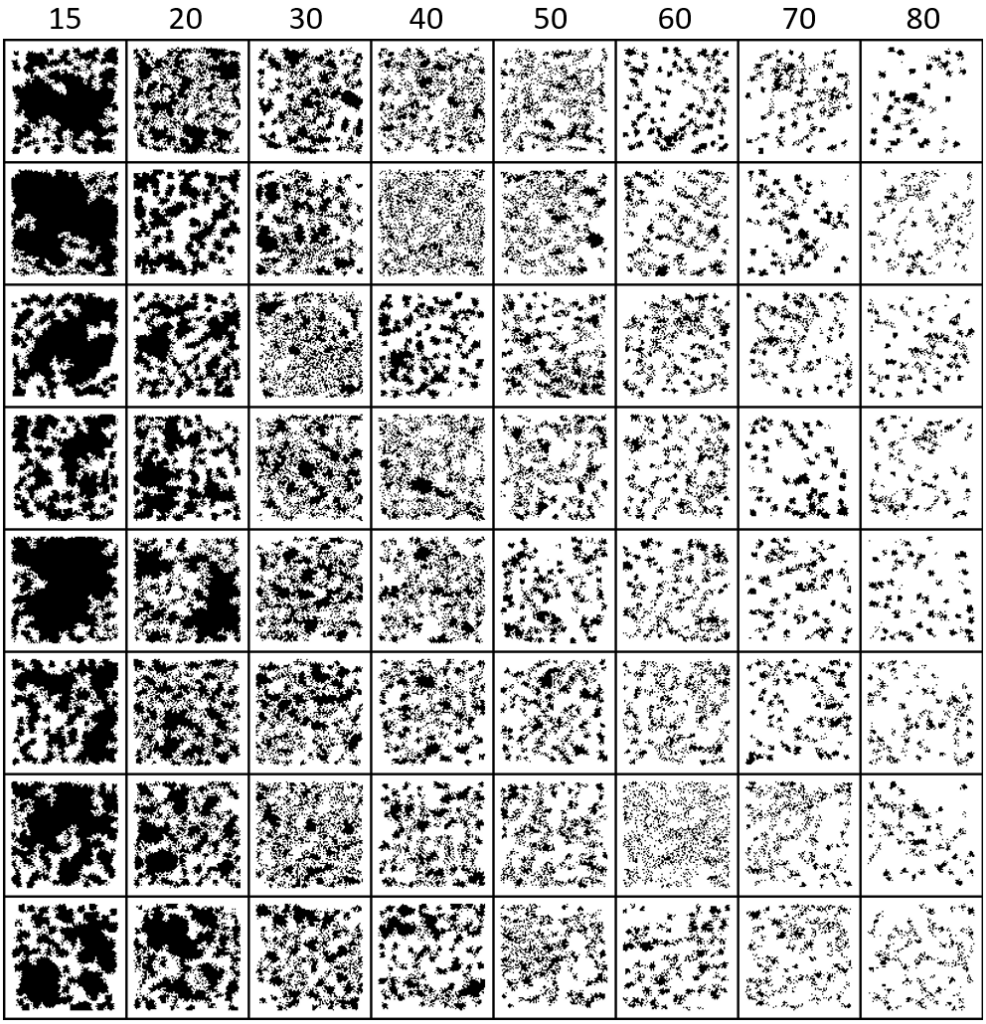


Figure 13 Examples of generated images for different Young's modulus conditions from 15 to 80.

Each column is corresponding to the Young's modulus condition denoted at the top.

4.2.2 Microstructure Generation Based on Stress-Strain Curve.

To evaluate the performance of conditional microstructure generation from stress-strain curves, we choose eight different sets of a_1, a_2 and a_3 coefficients as input conditions and generate 128 microstructures for each input condition. The mean and standard deviation of the stress-strain curve coefficients of the generated microstructures are then calculated. The results are shown in Table 3. Note that the first four conditions are manually chosen based on dataset patterns and the last four conditions are randomly chosen from testing dataset. Examples of the generated microstructures are shown in Fig. 19. From left to right, the figure's columns are corresponding to the coefficient sets 1 to 8 in Table 3. It is shown that, for most of the coefficient results, the error in comparison with the original input conditions is less than 10%. Furthermore, the results show that resultant stress-strain curves match quite well with the input curves, as the stress-strain curves are determined by the three coefficients as a whole and less sensitive to the individual coefficients. The results demonstrate that the cVAE based microstructure generation model produces physically meaningful porous material microstructures to achieve prescribed nonlinear material deformation behavior.

Table 3 Input and output of stress-strain curve coefficients of the generated microstructures.

Set	Coefficients			Mean ± Std Dev		
	a_1	a_2	a_3	a_1	a_2	a_3
1	20.0	-18.0	5.0	19.6418 ±1.0653	-16.7710 ± 1.1119	5.8215 ± 0.3727
2	10.0	-7.0	2.5	8.0511 ±1.3590	-6.0975 ± 1.0805	2.2886 ± 0.3469

3	5.0	-4.5	1.5	4.8452 ± 0.1915	-4.4102 ± 0.2032	1.6642 ± 0.0695
4	12.0	-10.0	3.5	11.0227 ± 1.0821	-9.2037 ± 0.8950	3.3253 ± 0.3020
5	7.3643	-6.5799	2.4055	7.0598 ± 0.3363	-6.5115 ± 0.3281	2.3446 ± 0.1099
6	26.8913	-24.9718	8.6733	29.2309 ± 0.8360	-27.3082 ± 0.8990	9.4483 ± 0.3066
7	14.3594	-12.4054	4.4816	14.5218 ± 1.3014	-12.2906 ± 1.1959	4.4225 ± 0.4013
8	10.5609	-8.7976	3.2363	9.2894 ± 0.8528	-8.0024 ± 0.6880	2.9180 ± 0.2376

Coefficient set 1 2 3 4 5 6 7 8

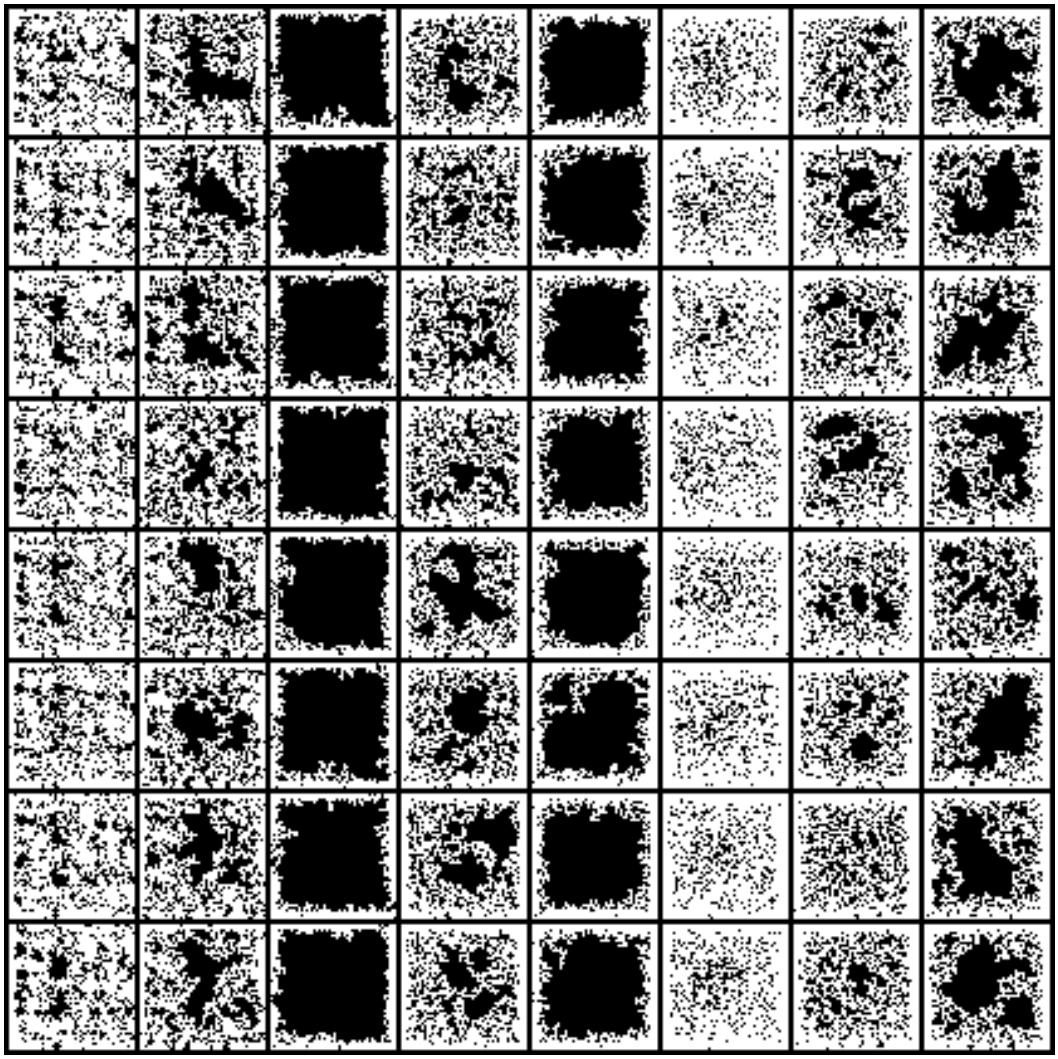


Figure 14 Examples of generated microstructures for given stress-strain curve coefficients. The columns from left to right are corresponding to the eight stress-strain curve coefficient sets listed in Table 3.

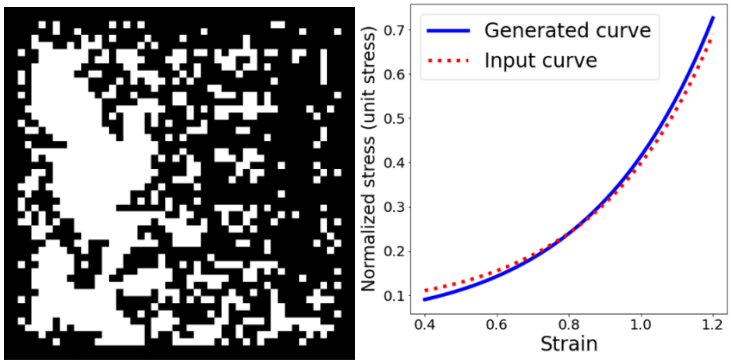


Figure 15 Generated microstructures and their stress-strain curves. Left: an example of generated porous microstructures from stress-strain coefficient set 2 in Table 3. Right: stress-strain curve of the generated microstructure in comparison with the input targe

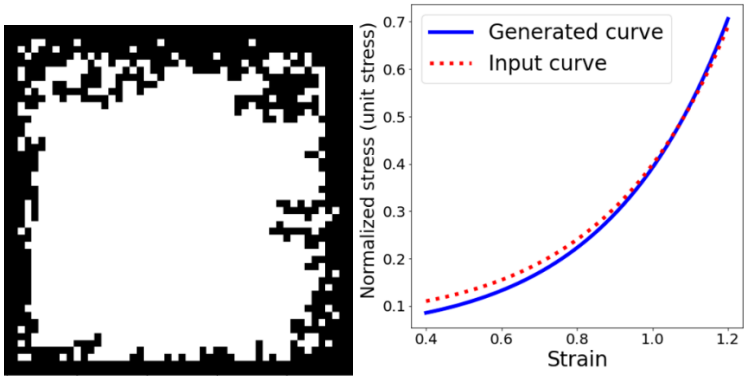


Figure 16 Generated microstructures and their stress-strain curves. Left: an example of generated porous microstructures from stress-strain coefficient set 5 in Table 3. Right: stress-strain curve of the generated microstructure in comparison with the input targe

4.3 Discussion

For microstructure generation, the stress-strain curves of the nonlinear porous materials are approximated using cubic polynomials and characterized by three coefficients. Microstructure generation models are developed based on conditional VAE. In the generative models, a condition

fusion layer, a stronger latent variable constraint ($\beta = 3$), and a pretrained regression model are incorporated into the traditional architecture of VAE to reduce error and ensure physically meaningful images. The results obtained show that the generative models are capable of producing microstructures that satisfy the mechanical property requirements and providing microstructural "designs" with considerable morphology variations.

CHAPTER FIVE

INVERSE DESIGN OF 3D FIBER-REINFORCED POLYMER COMPOSITE MATERIALS

In this chapter, we develop ML methods to design 3D microstructures of materials. Specifically, we develop a diffusion model designed for the inverse design of fiber-reinforced polymer composites (FRPCs) with a tailored nonlinear stress-strain response. In this section, we extend from solving 2D problems to addressing 3D problems. For this purpose, We also introduce a novel, collision-free fiber generation method utilizing machine learning techniques.

5.1 3D Data Generation

To train our diffusion model for inverse design, we generate a large-scale dataset of fiber-reinforced composites covering 52 fiber configurations (number of fibers per unit volume n in range from 10 to 50, fiber length $l \in \{30,50\}$, fiber diameter $d \in \{4,6,8,10\}$, and unidirectional/heterogenous fiber orientations) and obtain their stress-strain response curves through finite element analysis (FEA) using ANSYS. The FEA results show that the stress-strain curves could vary largely depending on the microstructure of the hyper-elastic fiber-reinforced polymer composites. Fig. 23 exhibits the ranges of the stress-strain curves of different fiber configurations. For each configuration, we generate 25,000 samples and build a dataset with 1.3 million samples in total covering a wide range of stress-strain curves. This provides us an insight into the valid range of the stress-strain response of each fiber configuration. We take 24,000 samples from each fiber configuration as the training data and keep the remaining 1,000 samples for testing purposes.

Given a fiber configuration, we generate samples by randomly placing fibers into the REV one by one with collision check to avoid fibers intersecting or overlapping with others. Collision-free placement is the prerequisite for FEA and also the physical constraint that must be met during inverse design. All fibers are modeled as homogeneous cylinders and considered as linear materials, while the polymer matrix is modeled as hyperelastic materials for large nonlinear elastic deformation. Each fiber cylinder is discretized as an octagonal prism to generate tetrahedral meshes for FEA with external parts outside the REV cut off along the six sides of the REV cube. Stress-strain responses are obtained through simulation by applying displacements of 10 mm, 20 mm and 30 mm along the x-direction of the REV, which corresponds to 10%, 20% and 30% strains respectively. The stress-strain curve is fitted as a cubic function.

$$\sigma(\eta) = a_1 + a_2\eta^2 + a_3\eta^3 \tag{9}$$

where σ is the nominal stress, η is the strain, and a_1, $a_2$and $a_3$are coefficients fitted based on the given simulated stress results $\sigma(10\%)$, $\sigma(20\%)$ and $\sigma(30\%)$.

Given a specific configuration of fiber length l and diameter d in the unit of mm and amount n, fibers with random positions $\{p^i\}_i$ and orientations $\{R^i\}_i$ are generated inside a cubic representative volume (REV) measuring $100 \times 100 \times 100$. Each fiber is modeled as a homogeneous cylinder. To replicate the experimental tensile test conditions, during the generation process, fibers must reside inside the cubic RVE without any collision between each other. Any part of the fiber extending beyond the RVE's boundaries is truncated by the RVE's faces.

In this work, we generated data of 52 combinations of fiber configurations:

$$n \in [10,11,\cdots,49,50], l = 50, d = 10 with\ various\ orientations$$

$$n = 30, l = 30, d \in [4,6,8,10] with\ heterogeneous\ orientations$$

$$n = 30, l = 50, d \in [4,6,8,10] with\ heterogeneous\ orientations$$

$$n = 30, l = 50, d \in [4,6,8,10]\ with\ unidirectional\ orientations$$

$$n = 30, l = \infty, d \in [4,6,8,10] with\ unidirection\ orientations$$

with volume fraction varying from around 1% to 25%. For the configurations with unidirectional orientations, a unified orientation is randomly sampled and applied to all fibers in one sample, while with heterogeneous orientations each fiber has its orientation sampled randomly and independently. For the scheme with $l = \infty$, fibers are considered long enough to completely penetrate the cubic REV. To do so, we set $l = 230$ during data generation. For each configuration, we generated 26,000 samples, 25,000 of which are used as the training data and the remaining are kept for testing. Totally, we have 1.3 million samples for training. We refer to Algorithm 1 for the details of our fiber distribution generation approach, where we denote the center position of each fiber using p^i and the orientation using the rotation representation R^i. Besides, we further assume that the default orientation of fibers with a zero rotation is the positive direction of z-axis, i.e. $z_{ref} \coloneqq [0,0,1]^T$.

Algorithm 1 Random Fiber Distribution Generation for Data Collection

function GENERATE($\mathbf{p}^i, \mathbf{R}^i$; d, l, n)
  $a = 0; b = 100$ ▷ Boundary constraints
  $r = 0.5d$ ▷ Fiber radius
  $i = 0$
  while $i < n$ **do**
    do
      $\mathbf{p}_i \sim$ UNIFORM$[a + \sqrt{2}r, b - \sqrt{2}r]$ ▷ Fiber position
      Draw $\mathbf{R}_i$ in 3D space uniformly. ▷ Fiber orientation
      $\mathbf{p}_i \leftarrow$ SHRINK$(\mathbf{p}_i, \mathbf{R}_i; r, l, a, b)$ ▷ Apply boundary constraint
      $d_{\min} = \min\{$DIST$(\mathbf{p}^i, \mathbf{R}^i; \mathbf{p}^j, \mathbf{R}^j; d, l)\}_{j<i}$ ▷ Collision Check
    while $d_{\min} < d + 0.02$
    $i \leftarrow i + 1$
  end while
end function

function DIST($\mathbf{p}^i, \mathbf{R}^i$; $\mathbf{p}^j, \mathbf{R}^j$; d, l) ▷ Shortest distance between two fibers
  $\mathbf{d}^i = \mathbf{R}^i \mathbf{z}_{\text{ref}}$ ▷ Direction vector of fiber i
  $\mathbf{u} = l\mathbf{d}^i$; $\mathbf{e}^i = \mathbf{p}^i - 0.5\mathbf{u}$ ▷ End points of fiber i
  $\mathbf{d}^j = \mathbf{R}^j \mathbf{z}_{\text{ref}}$; $\mathbf{v} = l\mathbf{d}^j$; $\mathbf{e}^j = \mathbf{p}^j - 0.5\mathbf{v}$
  $\mathbf{r} = \mathbf{e}^j - \mathbf{e}^i$
  $\Delta = ||\mathbf{u}||^2||\mathbf{v}||^2 - ||\mathbf{uv}||^2$
  if $\Delta < 10^{-6}$ **then**
    $s = \mathbf{ru}/||\mathbf{u}||^2; t = s(\mathbf{uv} - \mathbf{rv})/||\mathbf{u}||^2$
  else
    $s = \left(\text{CLIP}\left(((\mathbf{ru})(\mathbf{uv}) - (\mathbf{rv})||\mathbf{u}||^2)/\Delta, 0, 1\right)(\mathbf{uv}) + \mathbf{ru}\right)/l^2$
    $t = \left(\text{CLIP}\left(((\mathbf{ru})||\mathbf{v}||^2 - (\mathbf{rv})(\mathbf{uv})/\Delta, 0, 1\right)(\mathbf{uv}) - \mathbf{rv}\right)/l^2$
  end if
  return $||(\mathbf{e}^i + s\mathbf{u}) - (\mathbf{e}^j + t\mathbf{v})||$
end function
function SHRINK($\mathbf{p}^i, \mathbf{R}^i$; r, l, a, b) ▷ Shrink a fiber into bounded range
  $\mathbf{d}^i = \mathbf{R}^i \mathbf{z}_{\text{ref}}$ ▷ Direction vector of fiber i
  $x, y, z \leftarrow \mathbf{d}^i$
  $\mathbf{d}_a = [r\sqrt{x^2 + y^2} + a, r\sqrt{y^2 + z^2} + a, r\sqrt{z^2 + x^2} + a]^T - \mathbf{p}^i)/\mathbf{d}^i$
  ▷ Distance to the boundary plane along each axis
  $\mathbf{d}_b = [b - r\sqrt{x^2 + y^2}, b - r\sqrt{y^2 + z^2}, b - r\sqrt{z^2 + x^2}]^T - \mathbf{p}^i)/\mathbf{d}^i$
  return $\mathbf{p}^i + 0.5\left(\max\{-0.5l, \min\{\mathbf{d}_a\}\} + \min\{0.5l, \min\{\mathbf{d}_b\}\}\right)\mathbf{d}^i$
end function

Upon completion of the composite microstructure generation, we employ Gmsh to construct a progressively refined tetrahedron mesh for each generated RVE. Specifically, the top and bottom faces of a fiber cylinder are discretized into identical octagons with 8 vertices for meshing. This leads to an octagonal prism representation of each fiber after discretization and meshing. This technique is exemplified with both the composite RVE, and its associated boundary conditions illustrated in Fig 22. This approach highlights the application of advanced meshing methods in the modeling of fiber-reinforced polymer composites for computing stress-strain curves.

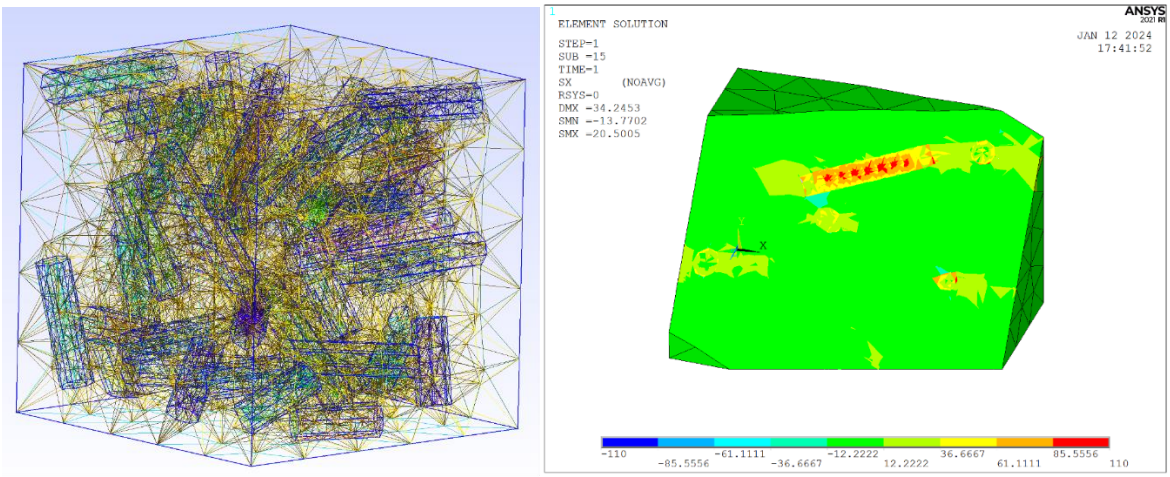


Figure 17 Demonstration of structural analysis on generated data. Left: meshing results by discretizing each fiber as an octagonal prism. Right: stress response obtained by simulation using finite element analysis.

Once the meshes of 3D fiber-reinforced polymer composite RVEs are produced, the mechanical properties are obtained by using finite element analysis (FEA) with applied displacements of $10mm$, $20mm$ and $30mm$, which corresponds to 10%, 20% and 30% strains, respectively. In this work, we consider the fibers as linear materials and the matrix of composites as hyperelastic

materials for large elastic deformation. The polymer matrix material is described by the 3-parameter Ogden hyperelastic model. The strain energy is given by.

$$W = \sum_{i=1}^{3} \frac{\mu_i}{a_i} \left(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3 \right) + \sum_{k=1}^{3} \frac{1}{D_k} (J-1)^{2k} \tag{10}$$

where μ_i and a_i are material constants in the unit of pressure and a dimensionless quantity respectively, D_k indicates volume change with The initial bulk modulus $K = \frac{2}{D_1}$, λ_1, λ_2 and $\lambda_3$are principal stretches in three directions of the strain tensor, and $J = (\lambda_1 \lambda_2 \lambda_3)^{\frac{1}{2}}$. Material properties used for simulation are listed in Table 4. Simulation is done on machines equipped with a 16-core E5-2665 CPU and 62GB memory. The time needed to generate and simulate one sample is around 15 seconds to 1 minute mainly depending on the number of fibers.

Table 4 Material properties used during simulation of finite element analysis.

Fibers		RVE (3-parameter Ogden Hyperelastic Model)					
Elastic Modulus (E)	1000MPa	a_1	2.74	μ_1	9.19MPa	D_1	0.00001
Poisson's Ratio (ν)	0.3	a_2	-5.55	μ_2	8.61MPa		
		a_3	1.31	μ_3	6.92MPa		

We fit stress-strain curves using a cubic function with coefficients of a_1, $a_2$and a_3 in Equation 9. Given $\sigma(10\%)$, $\sigma(20\%)$ and $\sigma(30\%)$. obtained through simulation in the unit of MPa.

The FEA results show that the stress-strain curves could vary largely depending on the microstructure of the hyperelastic fiber-reinforced polymer composites. Fig 23. exhibits the ranges of the stress-strain curves of different fiber configurations.

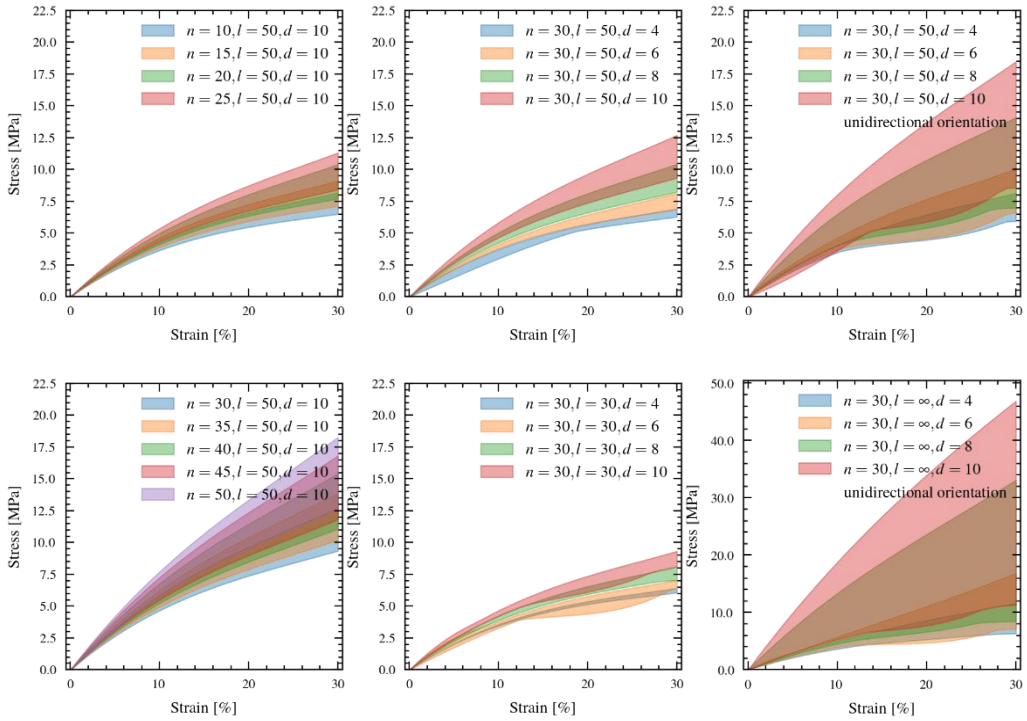


Figure 18 Ranges of the stress-strain curves of different fiber configurations

5.2 3D Spatial Diffusion Model

Our inverse design process takes a target stress-strain curve as input and generates the microstructures of fiber-reinforced composites that satisfy the stress-strain curve as output (Fig. 24 a). We consider the inverse design process of fiber-reinforced composites as a conditional generative task of deciding the position and orientation of each fiber based on the combination of the given stress-strain curve ($s \coloneqq [a_1, a_2, a_3] \in R^3$) and the fiber configuration c (including the fiber diameter ($d \in R^1$), length ($l \in R^1$),) and fiber amount (n)), where d and l are unified for all fibers in a cubic representative volume element (RVE). While d and l are given as extra conditional input to the model network, the fiber amount condition n is taken into account implicitly, which is decided by the model input dimension. Each fiber i therefore has two attributes to be modeled: $p_i \in R^3$ is the position coordinate and $R_i \in SO(2)$ is the orientation. The orientation R_i is defined

in the $SO(2)$ space, because the symmetry of the cylindric geometric model of fibers makes the rotation around the pole axis of each fiber itself inconsequential.

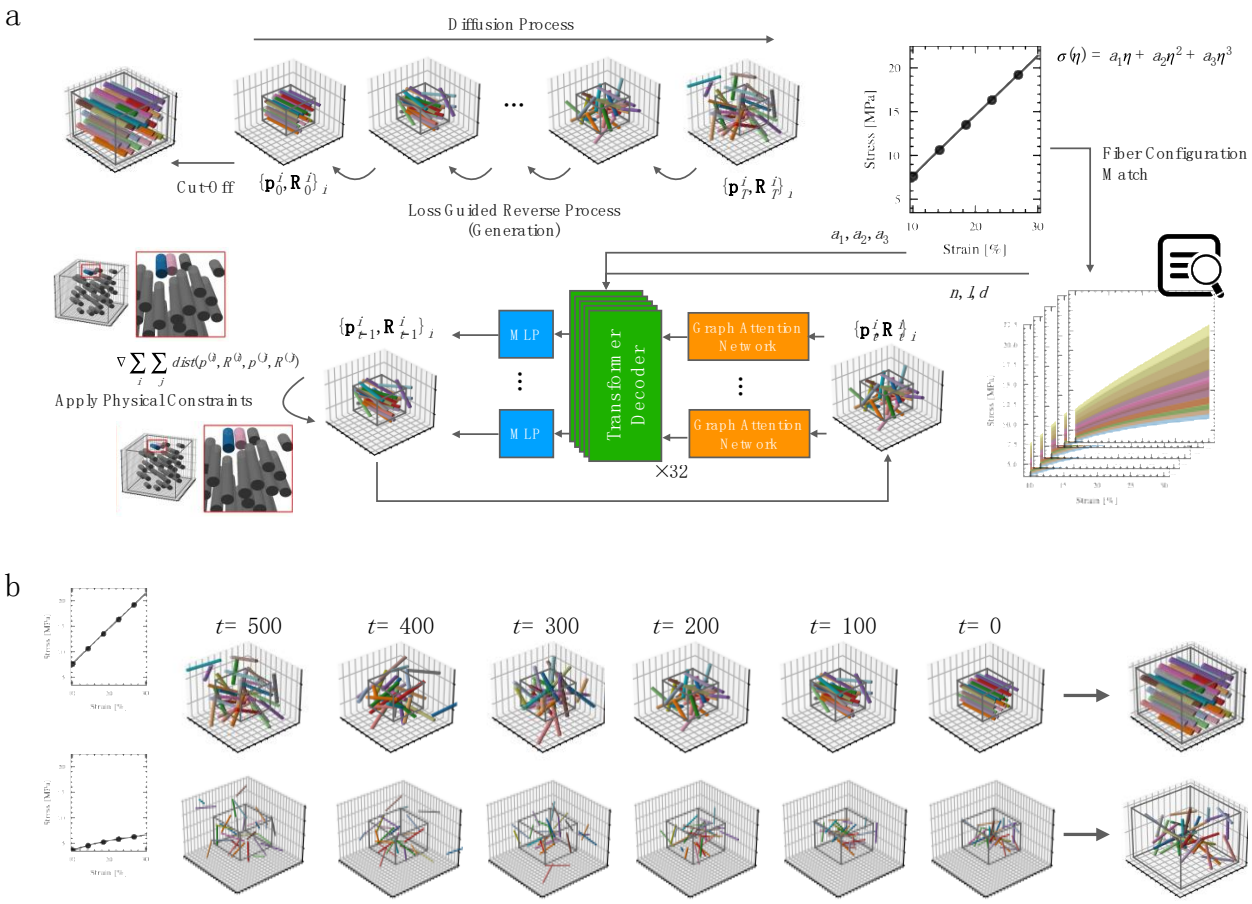


Figure 19 (a) Overview of our proposed inverse design system of fiber-reinforced composites. During training, we take the diffusion process and add noise to fiber positions p_0^i and orientations R_0^i by randomly moving and rotating the fibers. A network is trained to predict the noise added to get p_t^i from p_0^i. By utilizing the denoising neural network, during inverse design, we estimate p_{t-1}^i, R_{t-1}^i from p_t^i, R_t^i step by step starting from a randomly drawn fiber spatial distribution of $\{p_T^i, R_T^i\}_i$. The fiber configuration (amount n, length l and diameter d) is chosen through an automatic matching mechanism that compares the input (expected) stress-strain curve and our collected dataset of the stress-strain response. We employ a stack of 32

transformer decoders as the backbone architecture for noise estimation, coupled with a graph attention network for fiber feature embedding. Additionally, a loss-based guidance is employed during the generation process to apply physical constraints on the generated fiber distributions and to ensure that the generated fiber distributions are free of collisions. (b) Demonstrations of the denoising process where arranged fiber distributions ($t = 0$) are achieved from randomly drawn distributions ($t = T$ where $T = 500$ in our implementation) step by step given the expected stress-strain curves shown in the left plots. Top row: a distribution of unidirectional, long fibers penetrating the cubic representative volume element (RVE) with $n = 30$ and $d = 10$. Bottom row: a distribution of short fibers of $n = 20, l = 50, d = 4$ with heterogeneous orientations. In the final step, all fibers are cut along the RVE cube to achieve the spatial distribution where fibers are all inside the RVE.

We first find the proper fiber configurations for the expected stress-strain curve *s* by either matching the given stress-strain curve with our collected stress-strain response dataset if the given stress-strain curve is in the range of our dataset (see Methods) or guessing some fiber configurations if the given stress-strain curve is out of the range of our dataset. Then, we will feed the stress-strain curve and the chosen fiber configuration into our PC3D_Diffusion to generate the desired fiber-reinforced composite (Fig. 24 b). We perform conditional generation in a guidance-free way [98], where the condition is applied as a condition input fed into the diffusion neural network.

Given that the impact of one fiber on the overall stress-strain response is not only decided by the fiber's position and orientation in the RVE but also by the interactions among multiple fibers, we employ a stack of 32 transformer decoders as the backbone coupled with a graph attention network

(GAT) [99] to synthesize the spatial representation of each fiber. Transformer architecture has been demonstrated as an effective way to capture the relations between multiple parallel inputs [100]. When feeding the transformer network, we enhance the spatial representation of each fiber not only by taking into account the position and orientation of the fiber itself but also the relative positions and orientations of all neighbor fibers. To do so, we model fibers in a RVE as nodes in a graph with directed edges. The neighbor fibers are synthesized through the graph attention mechanism of GAT with learnable edge weights. The spatial representation of each fiber is thus enhanced as $GAT\left(p^i, R^i, \{p^j, R^j\}_{j\neq i}\right)$. Projecting through a multilayer perceptron (MLP), the output of the neural network is the estimated noise added to $\{p_0^i, R_0^i\}$ to get $\{p_t^i, R_t^i\}$. Another characteristic of the transformer architecture is the input permutation invariant. The transformer decoders weigh each parallel input through an attention mechanism by taking into account the features of each input which are unrelated to the input order. For language processing tasks, this characteristic requires additional positional embeddings added to each input language token (words, characters, or phrases) to reflect the positional relations between the input tokens. However, this characteristic indeed is consistent with our setup, where the input order of fibers should not impact the output. Therefore, in our implementation, we directly apply the transformer decoder architecture to the enhanced fiber spatial representations without positional embedding.

As the state-of-the-art technique of generative artificial intelligence, the denoising diffusion models include a stochastic forward process and a reverse process. The forward process gradually adds noise through a Markov chain to convert a data point x_0 (i.e. image) to a prior Gaussian distribution $N(0, I)$ in T steps. There are two methods to add noise, one is variance-preserving [101] and the other is variance-exploding [102]. The reverse process takes a randomly drawn sample $x_t N(0, I)$ as the initial value where $t = T$, and iteratively denoise the result from x_t to x_{t-1} until the

result x_0 is achieved. A neural network ϵ_θ is employed to estimate the noise added to x_0 given x_t. Through the estimated noise, we can further estimate the x_0 based on given x_t, and then compute $q_\theta(x_{t-1} \vee x_t)$ from $q(x_{t-1} \vee x_t, x_0)$ by parameterizing x_0 using the estimated noise. Here, we add noise to the fiber positions $\{p_i\}$and orientations $\{R_i\}$ by randomly moving and rotating each fiber and performing denoising by rearranging the fiber distribution. Due to the different characteristics of position and orientation, we use different methods to add noise to them.

In this section, we propose a 3d spatial diffusion model for the inverse design of fiber-reinforced composites. Specifically, given a certain fiber configuration (diameter, length, amount, and orientation consistency) and an expected stress-strain curve, our approach can generate the distribution of fibers, including fiber positions and orientations distributed in a cubic REV, conditionally to meet the given stress-strain curve. By introducing a guidance term during the generation process, our approach is guaranteed to generate collision-free results, where fibers comply with physical constraints and do not intersect with others.

In this section, we first give a brief introduction to diffusion models, and then provide details of our implementation of the 3D spatial diffusion model including the training and loss-guided generation processes.

Denoising diffusion models [103] have been widely used for generative modeling in various domains, such as images [104], 3D objects [105] and meshes [106], natural language [107], motion planning [108], proteins [109] and material science for porous material design [110]. When applied on generative tasks, diffusion models involve a forward noising process and a reverse denoising process.

In the literature of vanilla denoising diffusion probabilistic models (DDPMs), the forward process is the process where noise is added to the original data $x_0 \sim q(x)$ gradually based on a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t x_{t-1}}, \beta_{tI}\right). \tag{11}$$

This process can be rewritten through a Markov chain:

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I\right) \tag{12}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha_t} = \prod_{i=1}^{t} \alpha_i$. A noising data sample x_t can be obtained via

$$x_t = \sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\epsilon_t \tag{13}$$

where $\epsilon_t \sim N(0, I)$ is the added noise.

In the reverse process, based on the variational lower bound, we can replace x_0 with x_t and the noise ϵ_t, and thus have:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}(x_t, x_0), \widetilde{\beta_{tI}}\right) \tag{14}$$

where $\widetilde{\beta_t} = (1-\overline{\alpha_{t-1}})\beta_t/(1-\overline{\alpha_t})$ and

$$\tilde{\mu}(\mathrm{x_t}, \mathrm{x_0}) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t = \frac{1}{\sqrt{\bar{\alpha}_\mathrm{t}}}\left(\mathrm{x_t} - \frac{\beta_\mathrm{t}}{\sqrt{1-\bar{\alpha}_\mathrm{t}}}\epsilon_\mathrm{t}\right) \tag{15}$$

We refer to the literature for the inference of the above equation.

Typically, β_t increases with t and reaches near 1 when $t = T$. Approximately, we will have $q(x_T) \sim N(0, I)$. Therefore, during generation, we can initially draw x_T from a normal distribution randomly and utilize Equation 14 to iteratively estimate x_{t-1} based on x_t by employing a neural network to predict the noise ϵ_t given x_t and t. The neural network ϵ_θ for noise estimation can be trained by minimizing.

$$\mathcal{L} = E\left[\left|\left|\epsilon_t - \epsilon_\theta(x_t, t)\right|\right|^2\right] \tag{16}$$

where x_t is sampled by the diffusion process defined by Equation 13.

The vanilla DDPM is designed for image generation, where $x \in R^n$ is the collection of all pixel values. It can be extended directly for other kinds of scalar features, e.g. position or displacement from the origin. However, some kinds of features, e.g. rotation, cannot support meaningful scaling operations. In contrast to variance preserving form of DDPM, in the work of Noise

Conditional Score Networks (NCSNs) [111], the author proposed another noising method called variance exploding:

$$x_t = x_0 + \sigma_t \epsilon_t \tag{17}$$

where σ_t is a scheduled variance similar to β_t. Rather than performing scaling on x_0, this method uses noise to cover the original data. However, when $t \to T$, it requires σ_t to be large enough to eliminate the bias brought by x_0.

In [112], the author proved that DDPM and NCSN are equivalent and can be unified in the formation of stochastic differential equations. By projecting t into the range of [0,1] as a continuous, differentiable variable, the reverse process then can be solved through score-based generative modeling.

$$\mathrm{d}x_t \sim \mathcal{N}\left(f(x_t, t)\mathrm{d}t + g(t)^2 \nabla \log p_t\,(x_t), g(t)^{2\mathrm{d}} tI\right) \tag{18}$$

where $\nabla log p_t(x_t)$ is the score (log-likelihood) of x_t, $f(x_t, t) = \beta_{tx_t}/2$ for DDPM and 0 for NCSN, and $g(t)^2 = \beta_t$ for DDPM and $\mathrm{d}\sigma_t^2/\mathrm{d}t$ for NCSN. After getting the value of $\mathrm{d}x_t$, x_{t-1} can be obtained by $x_t - \mathrm{d}x_t$.

In Fiber Diffusion, we employ both Equations 14 and 18 in our reverse process for the generation of fiber positions and orientations respectively. Specially, by putting a negative sign to the mean value in Equation 18, we can obtain x_t using an addition operation. In such a way, we can extend the equation to rotation operations achieved through matrix or quaternion multiplication. The

"wrappable" nature of rotation angles decides that we do not need a very large value of σ_t to make x_t close to pure random variables when $t \to T$.

In our implementation, the fiber diameter ($d \in R^1$), length ($l \in R^1$) and amount (n) are specified externally through an automatic matching mechanism given the desired stress-strain curve,

where d and l are unified for all fibers in a cubic REV. Our model generates fiber distribution by providing the positions and orientation of each fiber to meet the given stress-strain curve ($s \coloneqq [a_1, a_2, a_3] \in R^3$). We consider the fiber distribution generation as a conditional generation task where the condition is defined as $c = [d, l, s] \in R^5$. The condition n is taken into account implicitly, which is decided by the model input dimension. Each fiber i therefore has two attributes to be modeled: $p^i \in R^3$ is the position coordinate and $R^i \in SO(2)$ is the orientation.

The orientation R^i is defined in the $SO(2)$ space because the symmetry of the cylindric geometric model of fibers makes the rotation around the pole axis of each fiber itself ineffective.

Following previous literature [113], we perform conditional generation in a guidance-free way, where the condition c is applied as a condition input fed into the noise estimation neural network.

5.2.1 Diffusion on Fiber Positions:

For the position property of fibers, we add Gaussian noise in a variance-preserving form [114]:

$$q\left(p_t^i | p_{t-1}^i\right) = N\left(p_t^i; \sqrt{1-\beta_t} p_{t-1}^i, \beta_t I\right) \tag{19}$$

given the variance schedule $\{\beta_t \in (0,1)\}_{t=1}^T$ where β_t increases with t. Through the Markov chain, we have

$$q\left(p_t^i | p_0^i\right) = N\left(p_t^i; \sqrt{\overline{\alpha}_t} p_0^i, (1-\overline{\alpha}_t) I\right) \tag{20}$$

where $\overline{\alpha} = \prod_{\tau \le t} \alpha_\tau$, and $\alpha_\tau = 1 - \beta_\tau$. As t increases to T, $1 - \overline{\alpha}_t$ also increases and reaches near 1 when t = T. Correspondingly, $\sqrt{\overline{\alpha}_t}$ decreases while t increases and reaches near 0 when t = T.

This setup leads to $q(p_T^i) \sim N(0, I)$, where the bias of p_0^i is roughly scaled down to zero and the variance is preserved with the noise ϵ. During the reverse process for generation, given $p_{\epsilon_t}^i$as the noise added to p_0^i, we can draw p_{t-1}^i from $q(p_{t-1}^i \vee p_t^i)$ by replacing p_0^i with $p_0^i = \left(p_t^i - \sqrt{1-\overline{\alpha}_t} p_{\epsilon_t}^i\right) / \sqrt{\overline{\alpha}_t}$ in $q(p_{t-1}^i \vee p_t^i, p_0^i)$. Following DDPM, we will train the position diffusion using the loss function:

$$L_p = E\left[\frac{1}{n}\sum_i \left\| p_{\epsilon_t}^i - \epsilon_p\left(p_t^i, R_t^i, \{p_t^i, R_t^i\}_{j\neq i}, c, t|\theta\right)\right\|^2\right] \tag{21}$$

where $p_t^2 = \sqrt{\overline{\alpha}_t} p^i + \sqrt{1-\overline{\alpha}_t} p_{\epsilon_t}^i$, $p_{\epsilon_t}^i \sim N(0, I) \in R^3$ and the original data p is normalized such that the cubic RVE measuring 100 mm×100 mm×100 mm is projected to the range of [−1, 1] along all three dimensions.

5.2.2 Diffusion on Fiber Orientations

The variance preserved form uses a scaling operation to eliminate the bias brought by the ground-truth value (e.g. p_0^i in Eq. 21) and perform manipulation in the range of standard Gaussian distributions. However, it does not work well for properties that cannot support scaling operations. Therefore, for orientation, we adopt a variance exploding form [115] to apply rotation noise. Given the symmetry when defining the fiber position using the center of the cylinder, the valid rotation range is [−π/2, π/2] for each rotation dimension. We approximate Gaussian-like sampling for rotation based on isotropic Gaussian distribution in SO space [116]. The isotropic Gaussian parameterizes a rotation in an axis-angle form, where the axis u is sampled uniformly and rotation angle ω has a probabilistic density function defined in a discrete form:

$$f_{IGSO}(\omega \vee \sigma^2) = \frac{1-cos\omega}{\pi}\sum_{l=0}^{L}(2l+1)\, exp(-l(l+1)\sigma^2)\frac{sin\left((l+1/2)\omega\right)}{sin(\omega/2)} \tag{22}$$

where $\omega \in [0, \pi]$ and σ^2 is the variance. We pick $L = 50{,}000$ in our implementation and pre-compute $f_{IGSO}(\omega|\sigma_t^2)$ for 5000 values with ω distributed uniformly in $[0, \pi]$. We further scale ω by 1/2 after the computation of probabilistic density. As such, the distribution of the rotation angle ω is still Gaussian-like but in the range of $[0, \pi/2]$ after scaling. We further scale ω by 1/2 after the computation of probabilistic density. As such, the distribution of the rotation angle ω is still Gaussian-like but in the range of [0, π/2] after scaling. The direction of the rotation angle is decided by the rotation axis and we thus can always assume ω ≥ 0. For each fiber i, a rotation noise $R_{\epsilon_t}^i$ is obtained through the rotation axis $u_t^i = \mathfrak{u}_t^i / \|\mathfrak{u}_t^i\|$, where $\mathfrak{u}_t^i \sim N(0, I) \in R^2$, and rotation angle $\omega_t^i \sim IGSO(\sigma_t^2)$. The rotation with added noise is defined as

$$R_t^i = R_0^i \left(R_{\epsilon_t}^i\right)^T \tag{23}$$

The vanilla variance exploding form takes $x_t \sim N(x_t; x_0, \sigma_t^2 I)$. It requires σ_t to be quite large in order to suppress the bias brought by x_0. Due to the "wrappable" nature of rotation angles, however, when we apply the formula for rotations by replacing the add operation with rotation multiplication, the bias of R_0^i can be eliminated as long as the cumulative distribution function of $IGSO(\sigma_t^2)$ is approximately linear in the range of $[0, 2\pi]$. We adopt a quadratic schedule with $\sigma_t \in (0.05, 5)$. This results in a near-linear cumulative distribution function of $IGSO(5^2)$ given $\sigma_t = 5$ when t = T. We train the neural network to predict the noise rotation in the form of exponential mapping:

$$L_R = E\left[\frac{1}{n}\sum_i \left\| \omega_t^i u_t^i - \epsilon_R\left(p_t^i, d_t^i, \left\{p_{t'}^i, d_t^i\right\}_{j \neq i'}, c, t \vee \theta\right)\right\|^2\right] \tag{24}$$

where $d^i = R^i z_{ref}$ and $d^j = R^j z_{ref}$ are the directional vectors given the reference unit vector $z_{ref} = [0,0,1]^T$. Given $\hat{\omega}_t^i \hat{u}_t^i = \epsilon_R\left(R_t^i, \backslash\backslash \{R_t^j \backslash\backslash\}_{j \neq i}, c, t \middle| \theta\right)$ is the predicted rotation noise from R_t^i to R^i and based on Equation 18, we estimate the noise from R_t^i to R_{t-1}^i through sampling

$$\omega^i_{\epsilon_{t \to t-1} u^i_{\epsilon_{t \to t-1}}} \sim \mathcal{N}\left(-\widehat{u_t^i} g(\bar{t})^2 \mathrm{d}\bar{t} \nabla \log f_{IGSO}\left(\widehat{\omega_t^i} \middle| \sigma_{\bar{t}}^2\right), g(\bar{t})^2 \mathrm{d}\bar{t} I\right) \tag{25}$$

where $\bar{t} = t/T \in (0,1]$. We use a discrete, integral representation of $t \in [1, T]$ following DDPM, and thus have $\mathrm{d}\bar{t} = 1/T$. Here, we put a negative symbol as the sign of the mean value in Equation 25 such that we can avoid minus operation and obtain R_{t-1}^i by stacking rotations:

$$R_{t-1}^i = R_t^i R_{\epsilon_{t\to t-1}}^i \tag{26}$$

where $R_{\epsilon_{t\to t-1}}^i$ is obtained from the exponential map of $exp(\hat{\omega}_t^i \hat{u}_t^i)$ directly through the noise estimation neural network if $t = 1$ or $exp\left(\omega_{\epsilon_{t\to t-1}}^i u_{\epsilon_{t\to t-1}}^i\right)$ using Equation 25 otherwise.

We use a quadratic variance schedule of σ_t defined in the range of $(\sigma_0, \sigma_1]$:

$$\sigma_{\bar{t}} = \sigma_0 + \bar{t} b_0 + \bar{t}^2(\sigma_1 - \sigma_0 - b_0) \tag{27}$$

Based on the notation in [150] for NCSN in variance exploding form, we have

$$g(\bar{t})^2 = \frac{\mathrm{d}\sigma_{\bar{t}}^2}{\mathrm{d}\bar{t}} = 2\sigma_t\big(b_0 + 2\bar{t}(\sigma_1 - \sigma_0 - b_0)\big) \tag{28}$$

5.2.3 Reverse Process for Composite Generation

The reverse process starts with t = T. By drawing the position and orientation from Gaussian distributions as the initial fiber distribution $\{p_T^i, R_T^i\}_{i=1}^n$, inverse design is done through the reverse process where $\{p_T^i, R_T^i\}_i$ is denoised into $\{p_0^i, R_0^i\}_i$ one step by step by replacing $\{p_0^i, R_0^i\}_i$ with $\{p_T^i, R_T^i\}_i$ and the estimated noise. In this process, the fiber configuration (length *l* and diameter *d*) and the coefficients of the given polynomial stress-strain curve are fed into the noise estimation network as the condition input, while the number of fibers in the RVE *n* is reflected implicitly by the number of input features |*i*| given that the transformer architecture has the same output dimensions with the input. Fig. 1b shows the reverse process for fiber distribution generation. We employ the vanilla denoising diffusion probabilistic model to denoise the position property of fibers. The reverse process of fiber position is done iteratively by

$$\begin{cases} p_{t-1}^{i} = \epsilon_p\left(p_t^i, R_t^i, \{p_t^i, R_t^i\}_{j\neq i}, c, t \vee \theta\right) for\, t = 1 \\ p_{t-1}^{i} \sim N\left(\frac{1}{\sqrt{\overline{\alpha}_t}}\left(p_t^i - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_p\left(p_t^i, R_t^i, \{p_t^i, R_t^i\}_{j\neq i}, c, t \vee \theta\right)\right), \beta_t I\right) otherwise \end{cases} \quad (29)$$

where $t \in [1, T]$ and $p_T^i \sim N(0, I)$ is sampled randomly. We employ a quadratic variance schedule in the range of $[\beta_1, \beta_T]$:$\beta_t = \left(\frac{T-(t-1)}{T}\sqrt{\beta_1} + \frac{t}{T}\sqrt{\beta_T}\right)^2$. For the reverse of fiber orientation, we have

$$R_{t-1}^i = R_t^i R_{\epsilon_{t\to t-1}}^i \quad (30)$$

We use a score-based approach to perform the reverse process of generation and obtain the noise rotation $R_{\epsilon_{t\to t-1}}^i$ through the exponential map representation $\omega_{\epsilon_{t\to t-1}}^i u_{\epsilon_{t\to t-1}}^i$ by

$$\begin{cases} \omega_{\epsilon_{t\to t-1}}^i u_{\epsilon_{t\to t-1}}^i = \hat{\omega}_t^i \hat{u}_t^i \wedge for\, t = 1 \\ \omega_{\epsilon_{t\to t-1}}^i u_{\epsilon_{t\to t-1}}^i \sim N\left(-\hat{u}_t^i g(\overline{t})^2 \mathrm{d}\overline{t} \nabla log f_{IGSO}(\hat{\omega}_t^i \vee \sigma_{\overline{t}}^2), g(\overline{t})^2 \mathrm{d}\overline{t} I\right) otherwise \end{cases} \quad (31)$$

where $\hat{\omega}_t^i \hat{u}_t^i = \hat{\epsilon}_{p^i}\left(\{p_t^i, R_t^i\}_i \vee \theta\right)$ is the estimated noise Ri ϵt written in the exponential map with the magnitude of $\hat{\omega}_t^i$ and the direction vector $\hat{u}_t^i$, $t = t/T$ is the normalized time step, and $f_{IGSO}(\cdot \vee \sigma^2)$ is the probabilistic density function of the distribution $IGSO(\sigma_t^2)$. We adopt a quadratic variance schedule: $\sigma_{\overline{t}} = \sigma_0 + \overline{t} b_0 + \overline{t^2}(\sigma_1 - \sigma_0 - b_0)$, where the variance $\sigma_{\overline{t}} = \sigma_1$when t = T, and thus the derivative of $\sigma_{\overline{t}}^2$ with respect to $g(\overline{t})^2 = \frac{\mathrm{d}\sigma_{\overline{t}}^2}{\mathrm{d}\overline{t}} = 2\sigma_t\left(b_0 + 2\overline{t}(\sigma_1 - \sigma_0 - b_0)\right)$.

5.2.4 Model Architecture

Our network for noise estimation is composed of two modules: the fiber representation module employs a graph attention network to embed each fiber while taking into account the spatial relationship between the fiber and its neighbors; the backbone of our network is a decoder-only transformer architecture taking the embedded representation of fibers as input and outputting the prediction of noise added to fiber positions and orientations.

To enhance the spatial representation of fibers within a REV, we perform representation learning for each fiber taking into account the local states of neighbor fibers. For simplicity, we ignore the subscript t here. Then, the j-th fiber's state observed by the i-th fiber locally is defined as

$$\mathrm{n}^{\mathrm{j}|\mathrm{i}} \coloneqq \left[\boldsymbol{p}^{j|i}, \left|\left|\boldsymbol{p}^{j|i}\right|\right|, \cos(\boldsymbol{d}^{\boldsymbol{j}}, \boldsymbol{d}^{\boldsymbol{i}})\right] \tag{32}$$

where $j \neq i$, $p^{j\vee i=(R^i)^T(p^j-p^i)\in R^3}$ is the relative position of the j-th fiber in the i-th fiber's local system, $p^{j\vee i}$ is the Euclidean distance between fiber j and i, $cos(d^j, d^i) = (d^j)^T d^i$ is the angle between two fibers' orientation directions given the directional vectors $d^i = R^i z_{ref}$ and $d^j = R^j z_{ref}$. This leads to $n^{j\vee i\in R^5}$ as the neighbor feature of each fiber j related to fiber i.

We employ a graph attention network (GAT) [117] to synthesize neighbors. To do so, we regard each fiber as a node in a directional graph linked with edges having weights $e^{j\vee i}$. The neighbor states then can be synthesized by weights:

$$w^{j|i} = \frac{exp(e^{j|i})}{\sum_{k\neq 1} exp(e^{k|i})} \tag{33}$$

Following [117], we obtain the edge weights through an attention mechanism using the cosine similarity between the self-state of the fiber i, i.e. p^i and R^i, and the local neighbor state $n^{j\vee i}$:

$$e^{j|i} = LeakyReLU\left(\mathrm{f_q}(\mathrm{p^i}, \mathrm{d^i}) \cdot \mathrm{f_k}(\mathrm{n}^{j|i})\right) \tag{34}$$

where the negative slope of $LeakyReLU$ is 0.2. The synthesized neighbor state is obtained by

$$\sum_j w^{j|i f_v(\mathbf{p^j}, \mathbf{d^j})} \tag{35}$$

where f_q, f_k and f_v are three embedding neural networks. The corresponding attention mechanism can be written as

$$\mathbf{n^i} = \mathrm{ATTN}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathrm{SOFTMAX}\left(\mathrm{f_q}(\mathrm{p^i}, \mathrm{d^i})\mathrm{f_k}\left(\{\mathrm{n}^{j|i}\}_\mathrm{j}\right)^\mathrm{T} \cdot \mathrm{f_v}\left(\{\mathrm{p^j}, \mathrm{d^j}\}_\mathrm{j}\right)\right) \tag{36}$$

where $[p^i, d^i]$, $\{n^{j\vee i}\}_j$ and $\{p^j, d^j\}_j$ correspond to the query, key and value vectors, respectively, in the vanilla attention mechanism [118]. The i-th fiber's spatial representation is denoted by

$$\mathbf{s^i} \coloneqq [\mathbf{p^i}, \mathbf{d^i}, \mathbf{n^i}] \tag{37}$$

The backbone architecture of our noise estimation neural network ϵ_θ is a transformer [118]. Instead of the encoder-decoder architecture [157] in the vanilla transformer, we employ a decoder-only architecture. Such a structure has been widely used in large language models [119-122] and been demonstrated as an effective architecture to capture the complex relationship between multiple inputs (word tokens in language).

In large language models, the neural network is employed to predict the token for the next words recurrently. In our implementation, we utilize the decoder-only architecture to perform noise estimation given multiple inputs (the fiber representations s^i) parallelly at the same time, while the condition c, including fiber diameter d and length l, and the time step t are fed into the network as the memory input. Since the transformer architecture adopts attention mechanisms and provides an output having the same dimension as the input, we do not need to explicitly indicate the fiber amount n in the condition vector c_t.

Another advantage of transformer architecture is that it is an architecture invariant to the order of input. When applying to natural language processing tasks, the transformer gets the word order information through positional encodings added to each input word token. However, our task is input permutation invariant. Namely, the input order of fibers should not impact the output of the network for each fiber. This characteristic is naturally supported by the transformer architecture without using positional encoding. Instead, we apply the position encoding technique to encode the time step variable t. Following the previous notation $\bar{t} = t/T$, we define the embedded time step t through sinusoidal positional encoding:

$$PE(t) = [\sin \bar{t}, \cos \bar{t}]. \tag{38}$$

This leads to a condition vector $c \coloneqq [d, l]$ and a timestep vector $t \coloneqq \left[sin\bar{t}, cos\bar{t}\right]$.

Each output of the transformer is finally passed through a shared two-layer neural network to get the predicted noise $\left[p^i_{\epsilon_t}, \omega^i_{\epsilon_t u^i_{\epsilon_t}}\right] \in R^5$ for each input s^i conditioned on $\backslash\backslash\{s^j\backslash\backslash\}_{j \neq i}$ and c. The architecture hyperparameters of the backbone transformer are listed in Table 5.

Table 5 Architecture of the backbone transformer

Number of attention heads	16	Number of stacked decoders	32
Input dimension	512	Nonlinear activator	ReLU
Latent dimension of feedforward network	2048	Total parameters	150M

5.2.5 Training Protocol

Our noise estimation network is trained using the loss functions defined in Equations 21 and 24. To balance the two loss terms adaptively during training, following previous literature of multi-objective learning [123], we introduce two additional learnable variables w_p and w_R as the coefficients to automatically adjust the learning weights of two terms related to position and rotation noise respectively. The final loss function is:

$$\mathcal{L} = \frac{1}{w_{\mathbf{p}^2}} \mathcal{L}_{\boldsymbol{p}} + \frac{1}{w_{\mathbf{R}^2}} \mathcal{L}_{\boldsymbol{\mathcal{R}}} + 2 \log w_{\mathbf{p}}\, w_{\mathbf{R}} \tag{39}$$

Our condition vector c does not include the configuration of unidirectional orientation or not. Therefore, we train two models to handle the configuration with and without unidirectional orientations respectively.

We list the hyperparameters that we used during model training and diffusion process in Table 6. For the rotation noise variance σ_t, when $\sigma_1 = 5$, the probability density function f_{IGSO} will lead to a linearly increasing cumulative distribution function, and results in a distribution where the angle ω is distributed almost uniformly within the range of $[0, \pi]$. Because of the cyclical nature of angles in the range of $[-\pi, \pi]$, variance exploding form represented in Equation of diffusion process then will not lead to any basis caused by the ground truth original value. Therefore, in the reverse process, we can draw R_T^i directly from $[0, \pi]$ uniformly as the initial value for data generation.

Table 6 Hyperparameters

Model training		Diffusion process					
Learning rate	0.0003	T	500	σ_0	0.05	ε	0.002
Optimizer	Adam W [124]	β_1	0.0001	σ_1	5	α	0.001
Batch size	256	β_T	0.02	b_0	1		

All our models are trained distributed on 4 machines, each of which is equipped with 2×A100 Nvidia GPUs. The whole training consuming all the training data takes around 1 week.

5.3 Physics Design in Collision Free

In actual composite materials, fibers do not penetrate or intersect with each other. Intersection of the fibers is referred to as “collision” in this work. Although the vanilla 3D spatial diffusion model can generate visually good results, its reverse process is performed by random sampling and it is

difficult to directly obtain a collision-free fiber distribution through the reverse process, particularly in scenarios where fibers are meant to be densely distributed or when they are distributed less densely overall but nearly uniformly. (Fig. 26). Even with only 10 fibers (d=10, l=50, random orientation) in the RVE, there are less than 10% of generated composites are collision-free (Fig. 26). For heterogeneous fiber orientations, the percentage of collision-free generation is lower than 4% for (n=30, d=10, l=50) and 0.2% for (n=30, d=10, l=∞) (Fig. 26). On the other hand, we observed that the total number of collisions between fibers are typically small in the results generated by the vanilla 3D spatial diffusion model. This implies that we can correct the generated fiber distributions to achieve collision-free results by a small number of modifications to the generated fiber positions and/or orientations.

Here, we introduce a guidance loss term during the reverse processing to apply physical constraints and enforce collision-free generation. We use a differentiable constraint loss function measuring the distance between each pair of fibers with a boundary constraint:

$$L_{\text{cons}}\left(\left\{p_t^i, R_t^i\right\}_i\right) = \frac{1}{n}\sum_{ij} max\left\{0,1 - \frac{DIST\left(SHRINK(p_t^i), R_t^i; SHRINK\left(p_t^j\right), R_t^j\right)}{d+\varepsilon}\right\} \tag{42}$$

where d is the diameter of fibers, ε is the minimal gap allowed between two fibers, $DIST(\cdot;\cdot)$ is a differentiable distance function and $SHRINK(\cdot)$ function shrinks fiber length to ensure the boundary constraints. $L_{\text{cons}} = 0$ when there is no collision, i.e. $DIST\left(p_t^i, R_t^i; p_t^j, R_t^j\right) \geq d + \varepsilon$ for all fiber pairs. Rather than maximizing the loglikelihood, at each step t during the reverse process, we minimize the loss guidance term by

$$\left\{p_t^i, R_t^i\right\}_i \leftarrow \left\{p_t^i, R_t^i\right\}_i - \alpha\nabla_{\left\{p_t^i, R_t^i\right\}_i} L_{\text{cons}}\left(\left\{p_t^i, R_t^i\right\}_i\right) \tag{43}$$

Applying L_{cons} to update fiber positions and orientations is a process of gradient descent, however, without needing any learning. In our implementation of the reverse process for data generation,

we update $\{p_t^i, R_t^i\}$ once at each time step t and keep performing updates until $L_{\text{cons}}\left(\{p_t^i, R_t^i\}_i\right) = 0$ at the last step. Since the vanilla 3D spatial diffusion model can directly provide results where fibers would not largely intersect or go through other fibers, typically we can achieve collision-free results within 10 iterations, while the whole reverse process takes around 8 seconds given $T = 500$ in our implementation.

Theoretically, when generating unidirectional fiber distributions, we can add an extra loss guidance term measuring the orientation divergence of fibers. However, during experiments, we find that the 3D spatial diffusion model is able to directly provide unified orientations with very small errors (see the quantitative evaluation section). Therefore, for fiber configurations with unidirectional orientations, we perform updating only on fiber positions $\{p_t^i\}_i$.

By applying the loss guidance during generation, our approach can generate collision-free results even in very dense scenarios. From the examples shown in Fig 26 and Fig 32, is observed that the overall distributions of fibers with and without the loss guidance are almost identical. The introduction of loss guidance only slightly changes the positions and/or orientations of collided fibers without significant modification to the overall distributions of fibers. Although the guided loss is applied at each step during generation without considering the change in the stress-strain response when modifying the fiber positions and orientations, the stress-strain response is still guaranteed by the following generation steps. At the last step of generation, the guided loss is applied iteratively without further revision. It is observed that the generated results are already near collision-free, as we achieve completely collision-free results with less than 10 iterations during our testing. The high-quality results obtained demonstrate the effectiveness of our approach of applying physical constraints through guided loss.

In Fig. 34a, we visually compare the results generated with and without the guided loss. As can be seen, by slightly moving the fibers without the change overall fiber distribution largely, the guided loss effectively avoid the collision between fibers. To show the necessity of using the guided loss, in Fig. 34b, we show the collision rate when no guided loss is employed during generation. The collision rate increases as the volume fraction of fibers increases. In the challenge cases where fibers are distributed densely, e.g. $n = 30 \vee 50, l = 50, d = 10$, the collision-free rates are about zero. This result highlights the necessity of using the proposed guidance loss to perform physically constrained fiber distribution generation.

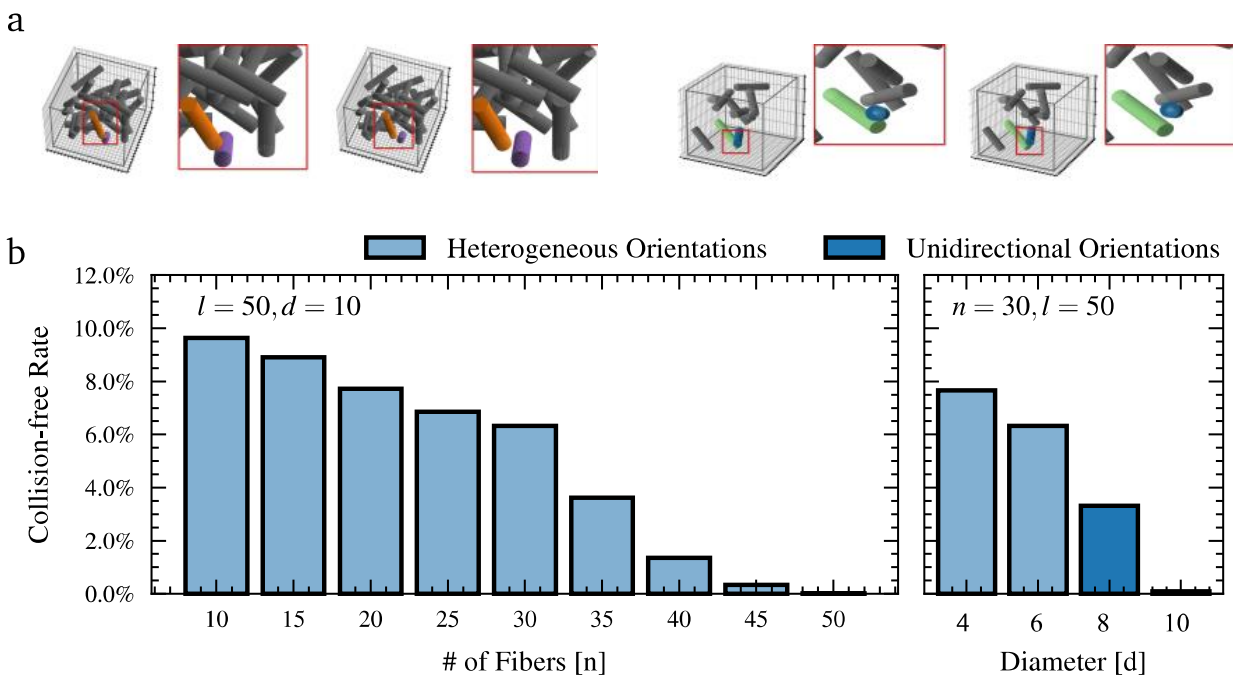


Figure 20 (a) Comparison of generated results with (right) and without (left) using the proposed loss guidance. (b) Collision-free rate when no loss guidance is employed. The proposed loss guidance can effectively ensure that physical constraints can be applied during the process of

generation and help generate collision-free fiber distributions by slightly modifying the position and/or orientation of the fibers.

Due to the uncertainty of artificial neural networks, it is impossible for the neural network to precisely predict noise without any errors. Plus, the reverse process is performed by random sampling, it is difficult to directly get a collision-free fiber distribution through the reverse process when fibers are supposed to be distributed densely. On the other hand, our neural network, which will be detailed in Section 5.2, is able to provide high-quality results, where the collisions between fibers, though existing, are very small. This implies that we can correct the generated fiber distributions to achieve collision-free results by slightly modifying the generated fiber positions and/or orientations.

Inspired by the previous literature employing a classifier to perform conditional generation [125, 126]. we introduce a guidance loss term during the reverse processing to apply physical constraints and perform collision-free generation. Similarly to the score-based generation modeling introduced above, the previous literature employs the gradients of log-likelihood to guide generation during the reverse process, i.e. replacing Equation 44. with

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}(x_t, x_0) + s\nabla_{x_t} \log p\,(y|x_t), \widetilde{\beta_{tI}}\right) \tag{44}$$

where $p(y|x_t)$ is a classifier model and s is a scalar coefficient. This approach guides the reverse process to generate x_{t-1} which has a higher likelihood $p(y|x_t)$. In our implementation, we do not use any learnable classifier here. Instead, we perform collision-free guidance through a differentiable constraint loss function measuring the distance between each pair of fibers with a boundary constraint:

$$\mathcal{L}_{cons}\left(\{p_t^i, R_t^i\}_i\right) = \frac{1}{n}\sum_{ij} max\{\,0,1 - \frac{DIST\left(SHRINK\left(\boldsymbol{p_t^i}\right), \boldsymbol{R_t^i}; SHRINK\left(\boldsymbol{p_t^j}\right), \boldsymbol{R_t^j}\right)}{d+\varepsilon}\} \qquad (45)$$

where d is the diameter of fibers, ε is the minimal gap allowed between two fibers, $Dist(\cdot;\cdot)$ is a differentiable distance function and $Shrink(\cdot)$ function shrinks fiber length to ensure the boundary constraints. $L_{cons} = 0$ when there is no collision, i.e. $Dist\left(p_t^i, R_t^i; p_t^j, R_t^j\right) \geq d + \varepsilon$ for all fiber pairs. Rather than maximizing the log-likelihood, at each step t during the reverse process, we minimize the loss guidance term by

$$\{p_t^i, R_t^i\}_i \leftarrow \{p_t^i, R_t^i\}_i - \alpha\nabla_{\{p_t^i, R_t^i\}_i}\mathcal{L}_{cons}\left(\{p_t^i, R_t^i\}_i\right). \qquad (46)$$

We refer to Algorithm 1 for the differentiable implementation of the functions $Dist$ and $Shrink$.

Applying Equation 46 to update fiber positions and orientations is a process of gradient descent, however, without needing any learning. In our implementation of the reverse process for data generation, we update $\{p_t^i, R_t^i\}_i$ once at each time step t and keep performing update until $L_{cons}\left(\{p_t^i, R_t^i\}_i\right) = 0$ at the last step. Since our neural network can directly provide results where fibers would not largely intersect or go through other fibers, typically we can achieve collision-free results within 10 iterations, while the whole reverse process takes around 8 seconds given $T = 500$ in our implementation.

Theoretically, when generating results where fibers have unidirectional orientations, we can add an extra loss guidance term measuring the orientation divergence of fibers. However, during experiments, we find that our neural network is able to directly provide unified orientations with very small errors.

Therefore, for fiber configurations with unidirectional orientations, we perform updating only on fiber positions $\{p_t^i\}_i$.

5.4 Results

In this section, we provide a detailed analysis quantitatively and qualitatively to evaluate the performance of our 3d spatial diffusion model for the inverse design of fiber-reinforced composites. We also perform a sensitive analysis of the proposed model architecture.

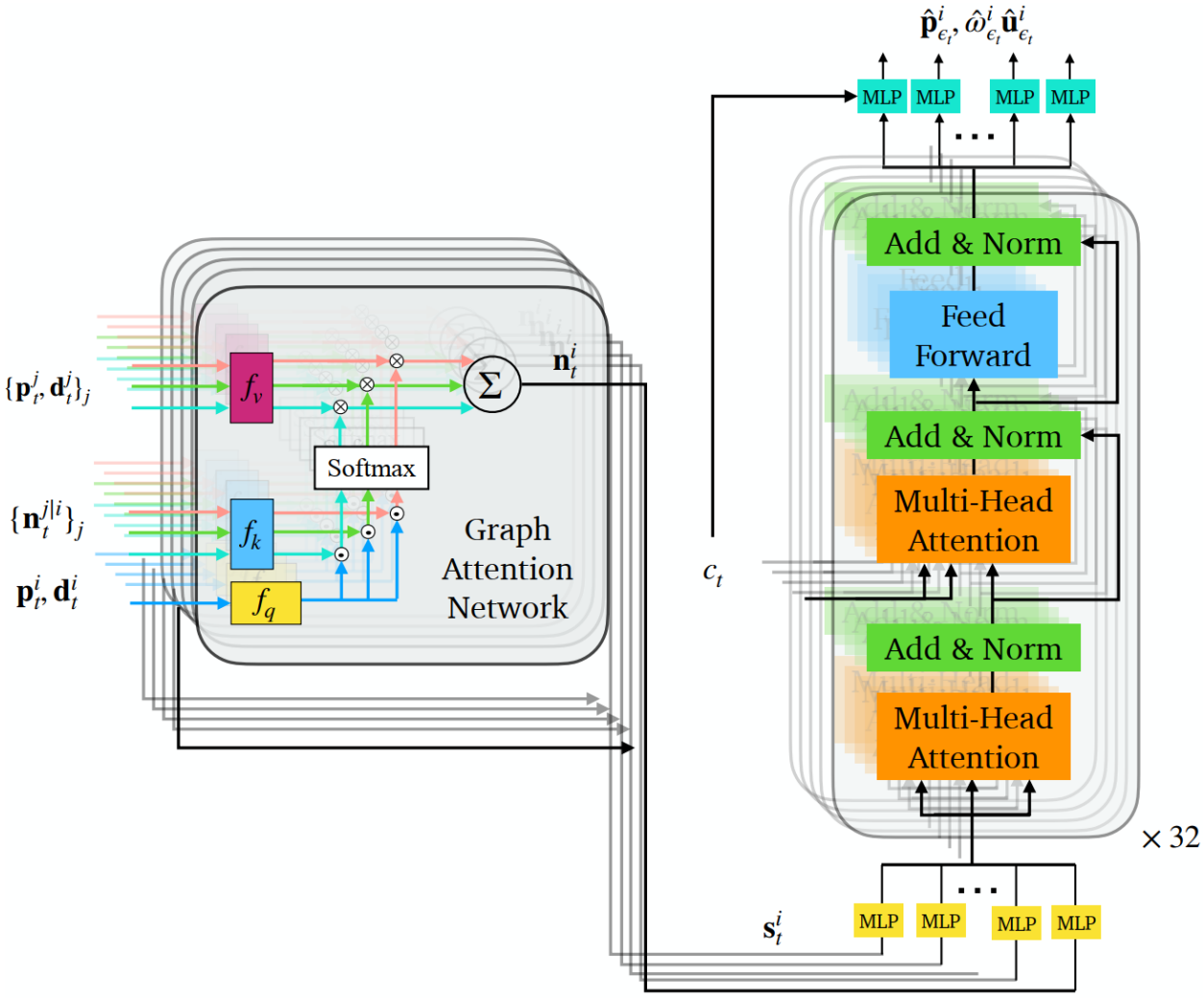


Figure 21 Network architectures ϵ_θ for noise estimation. Left: graph attention network to enhance the spatial representation of each fiber i by introducing the states of neighbor fibers $\{p^j_t, d^j_t\}_j$ through their local spatial states $\{n^{j|i}\}_j$ relative to the fiber i. Right: transformer decoders used to predict added noise given the condition state c_t and fiber representations s^i_t.

The synthesized fiber state s_t^i is projected to the space of R^{512} through a multilayer perception (yellow), and the final output of the decoders is converted to the prediction $\left[\hat{p}_{\epsilon_t}^i, \omega_{\epsilon_t}^{\hat{i}\hat{u}_{\epsilon_t}^i}\right] \in R^5$ corresponding to each input fiber state s_t^i through an additional multilayer perception (cyan).

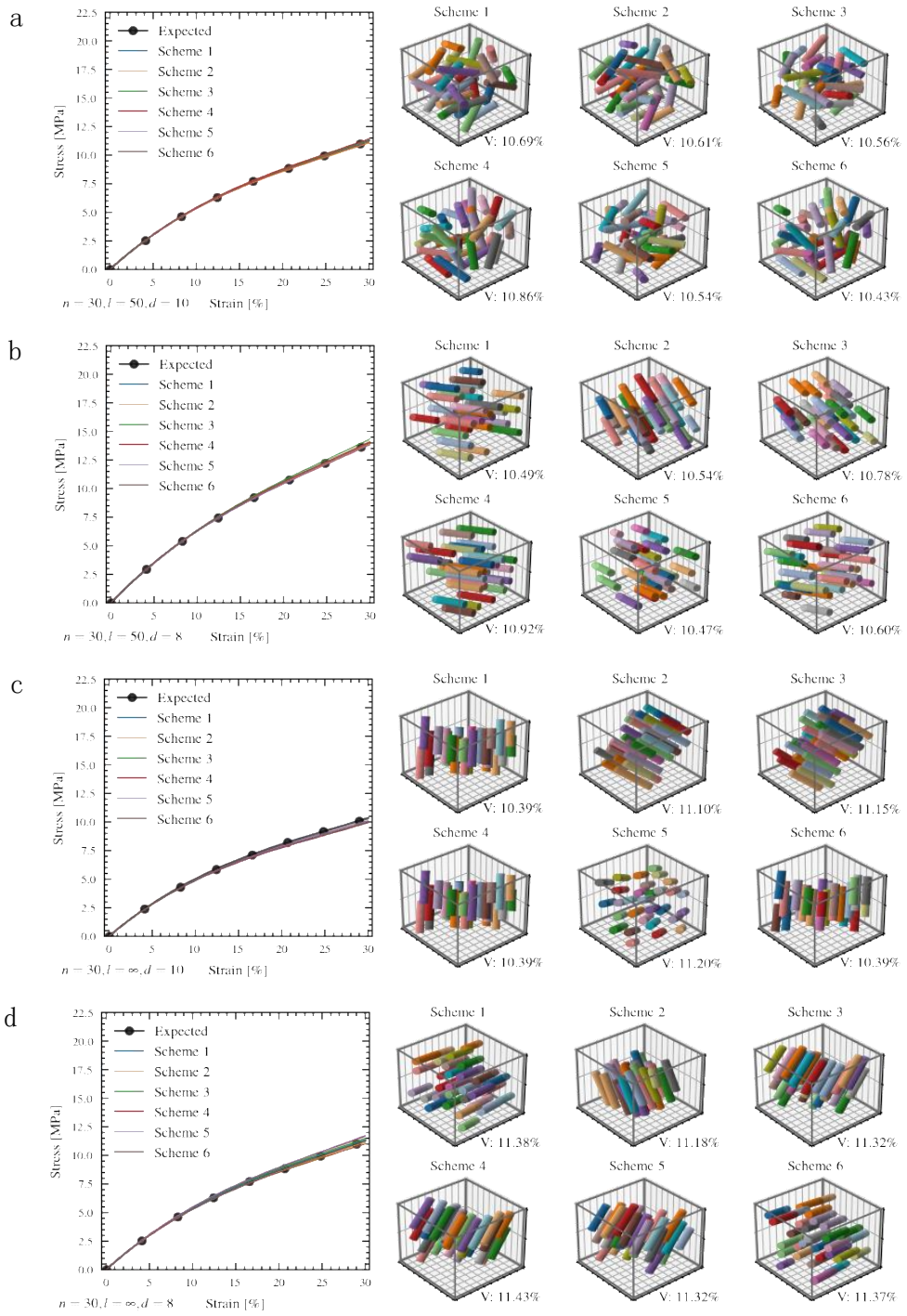
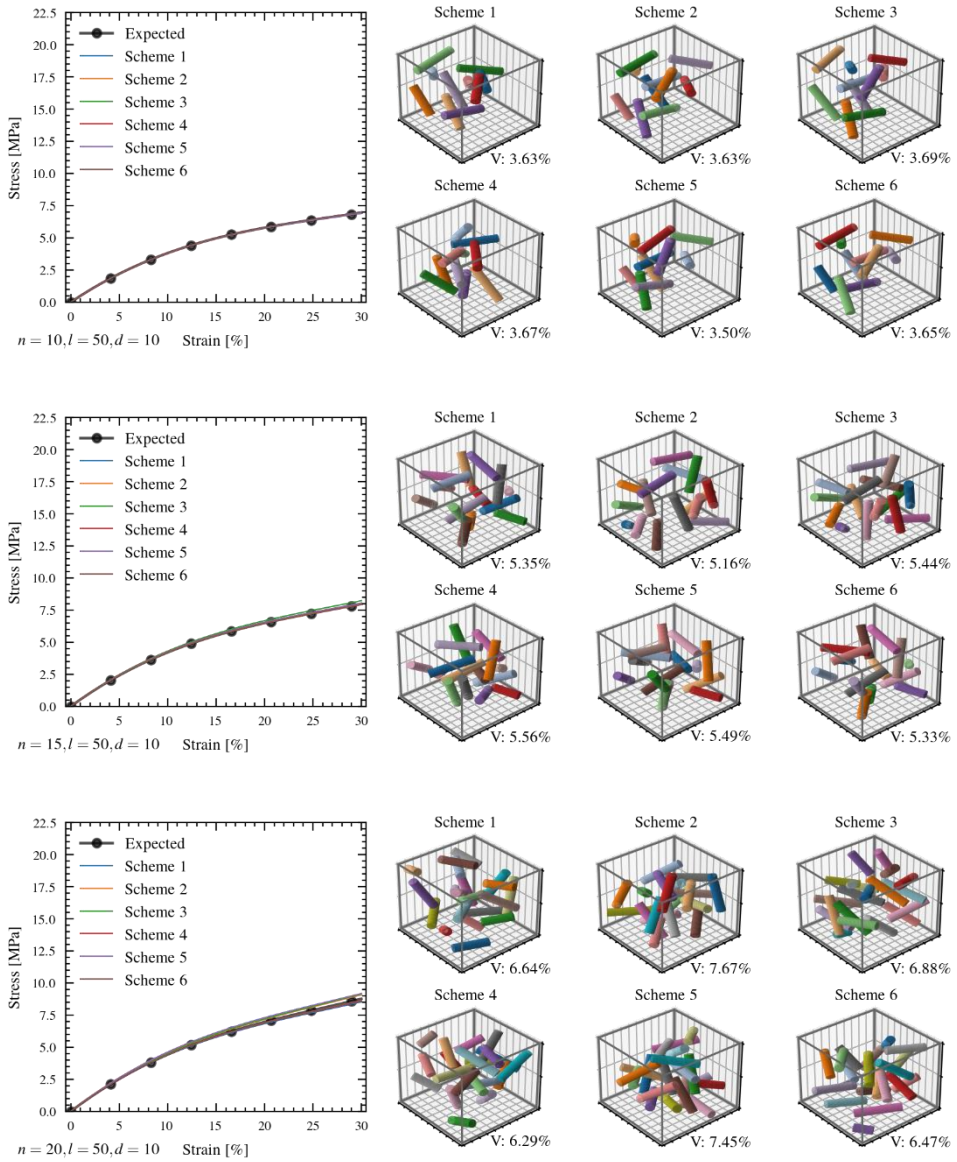


Figure 22 Generated fiber distribution schemes using different candidate fiber configurations given the same input stress-strain curve. The four fiber configurations with divergent lengths,

diameters, orientation constraints and, thereby, different resulting volume fractions are picked through our automatic matching mechanism. For each candidate fiber configuration, we generate six distinct schemes. Along each scheme, we show the volume fractions (V) after cutting off external fiber parts outside the cubic RVE. Our system can find proper fiber configurations and generate various fiber distributions with stress-strain curves close to the given one.



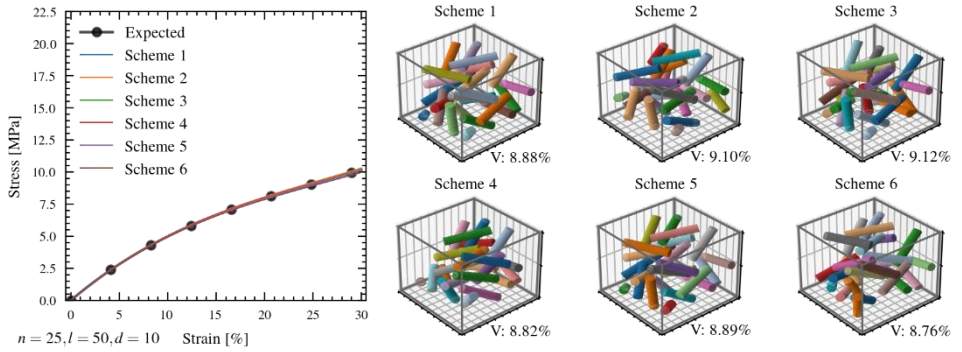
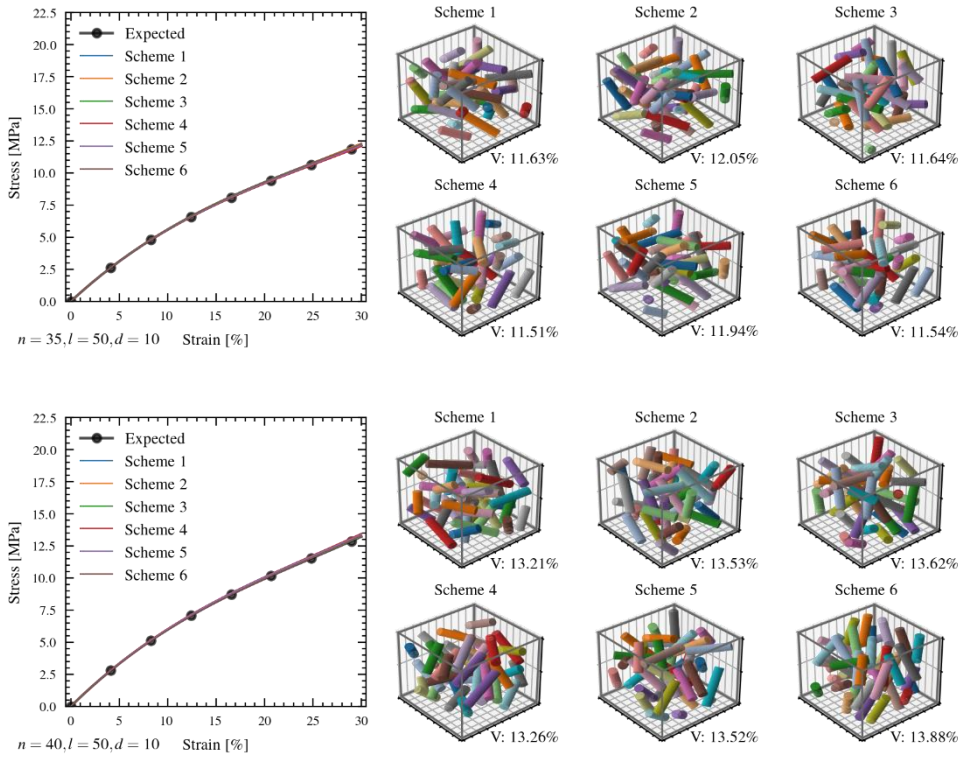


Figure 23 Generated short fiber distribution schemes with different numbers of fibers (n = 10, 15, 20, 25). All fibers distributed partially outside the cubic RVE are cut off along each side of the RVE. Volume fractions (V) are computed after cut-off.



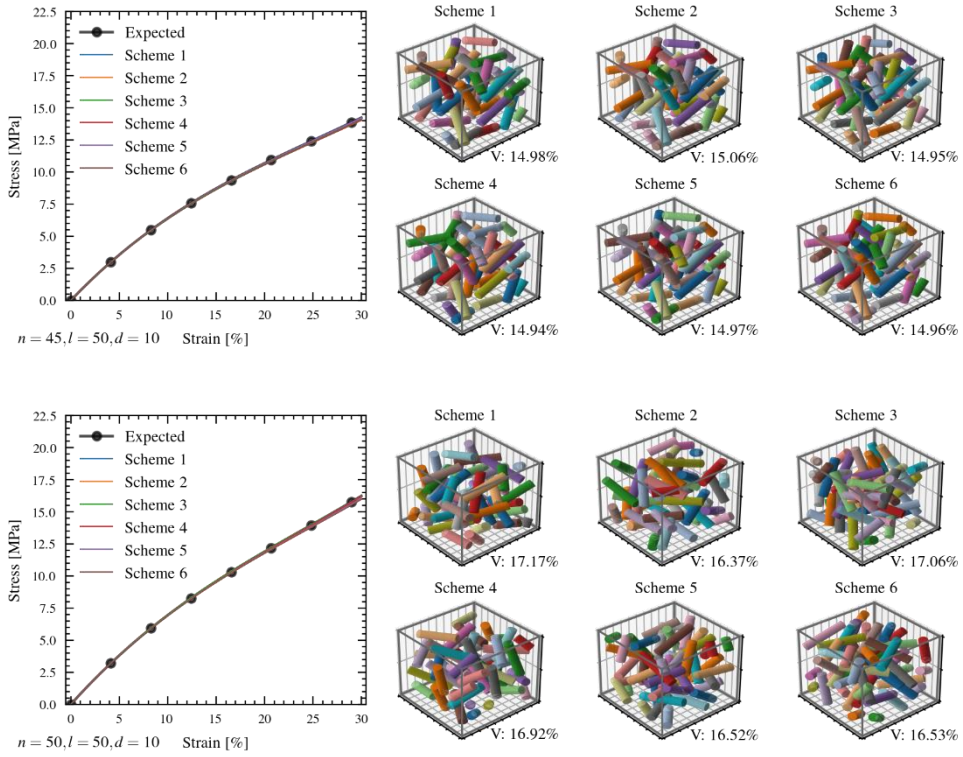
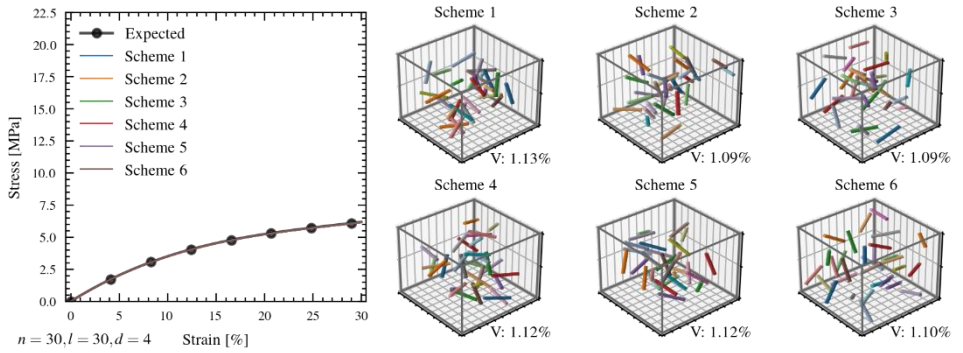


Figure 24 Generated short fiber distribution schemes with different numbers of fibers (n = 35, 40, 45, 50)



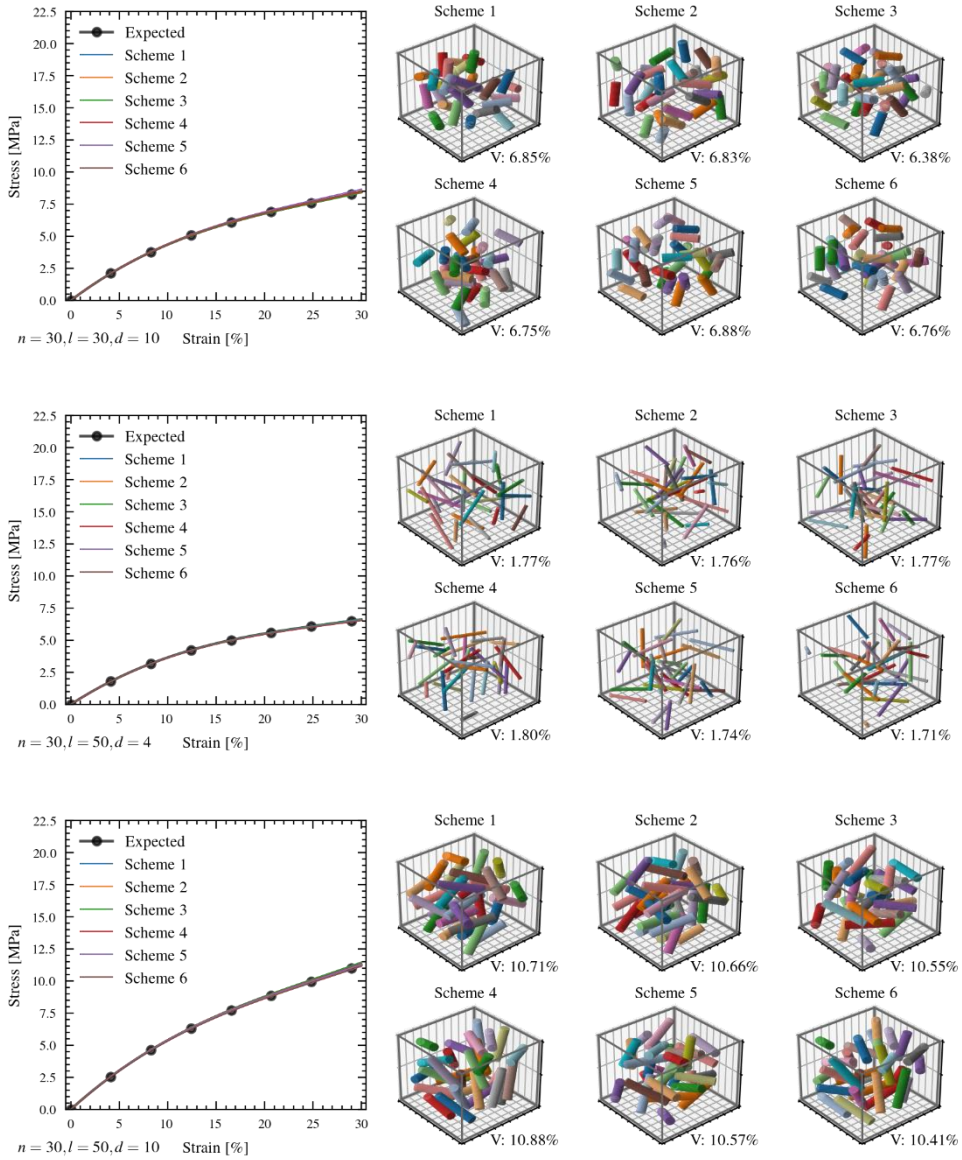


Figure 25 Generated unidirectional, short fiber distribution schemes given n = 30, l = 50 with different fiber diameters.

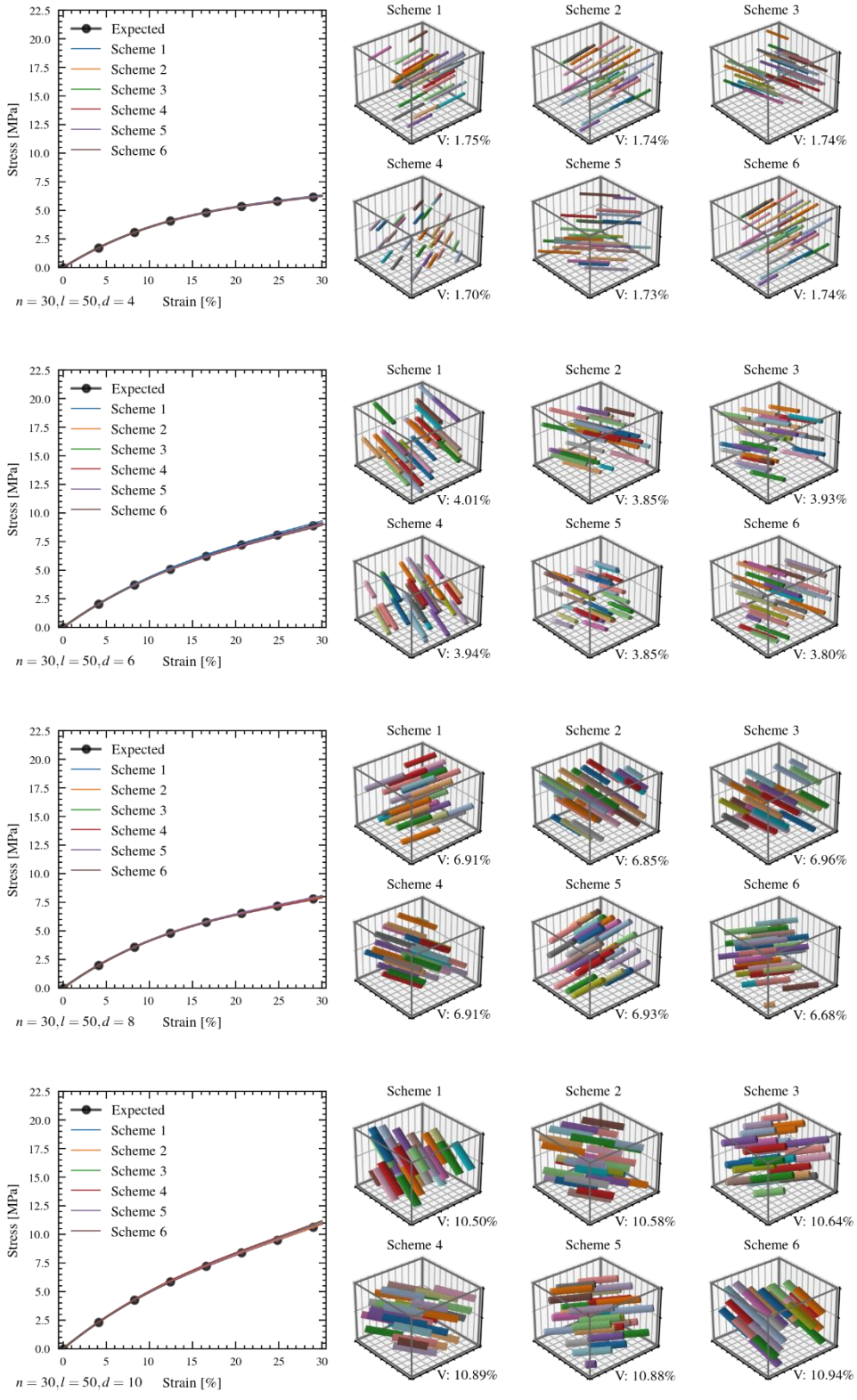
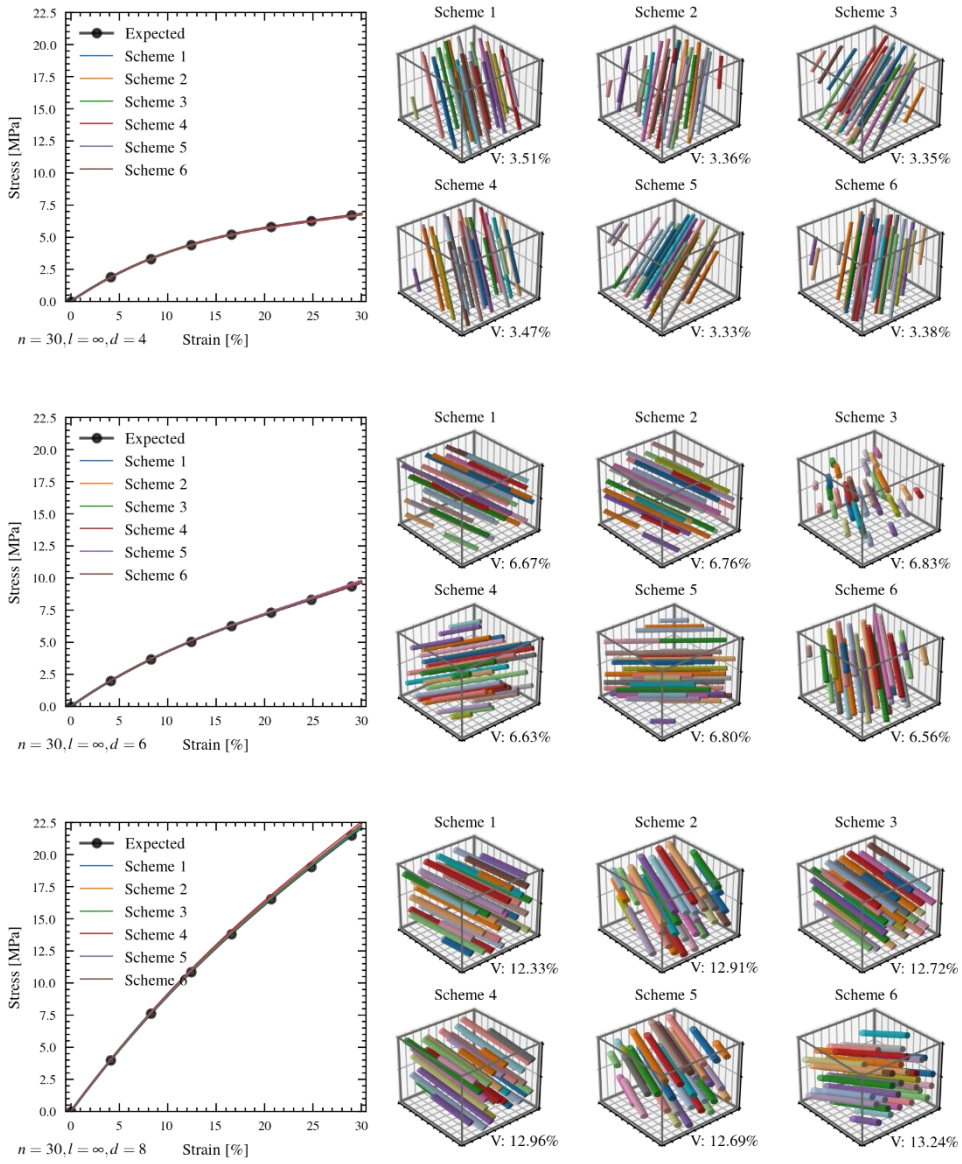

Expected
Scheme 1
Scheme 2
Scheme 3
Scheme 4
Scheme 5
Scheme 6
Stress [MPa]
Strain [%]
$n = 30, l = 50, d = 4$
$n = 30, l = 50, d = 6$
$n = 30, l = 50, d = 8$
$n = 30, l = 50, d = 10$
V: 1.75%
V: 1.74%
V: 1.74%
V: 1.70%
V: 1.73%
V: 1.74%
V: 4.01%
V: 3.85%
V: 3.93%
V: 3.94%
V: 3.85%
V: 3.80%
V: 6.91%
V: 6.85%
V: 6.96%
V: 6.91%
V: 6.93%
V: 6.68%
V: 10.50%
V: 10.58%
V: 10.64%
V: 10.89%
V: 10.88%
V: 10.94%

Figure 26 Generated unidirectional, short fiber distribution schemes given n = 30, l = 50 with different fiber diameters.



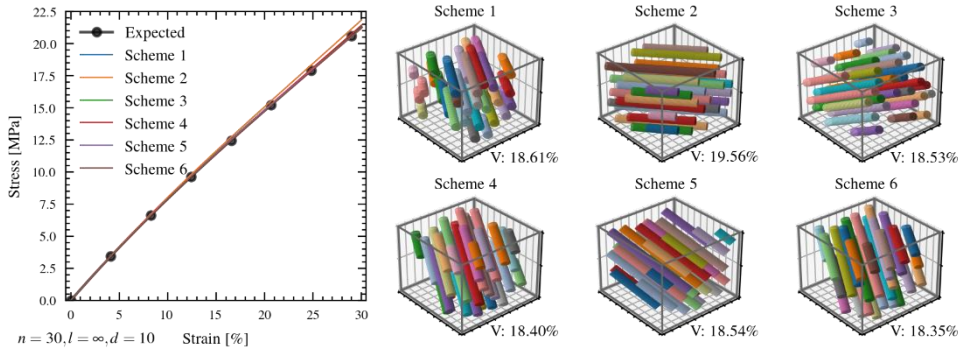


Figure 27 Generated unidirectional, long fiber distribution schemes with different fiber diameters.

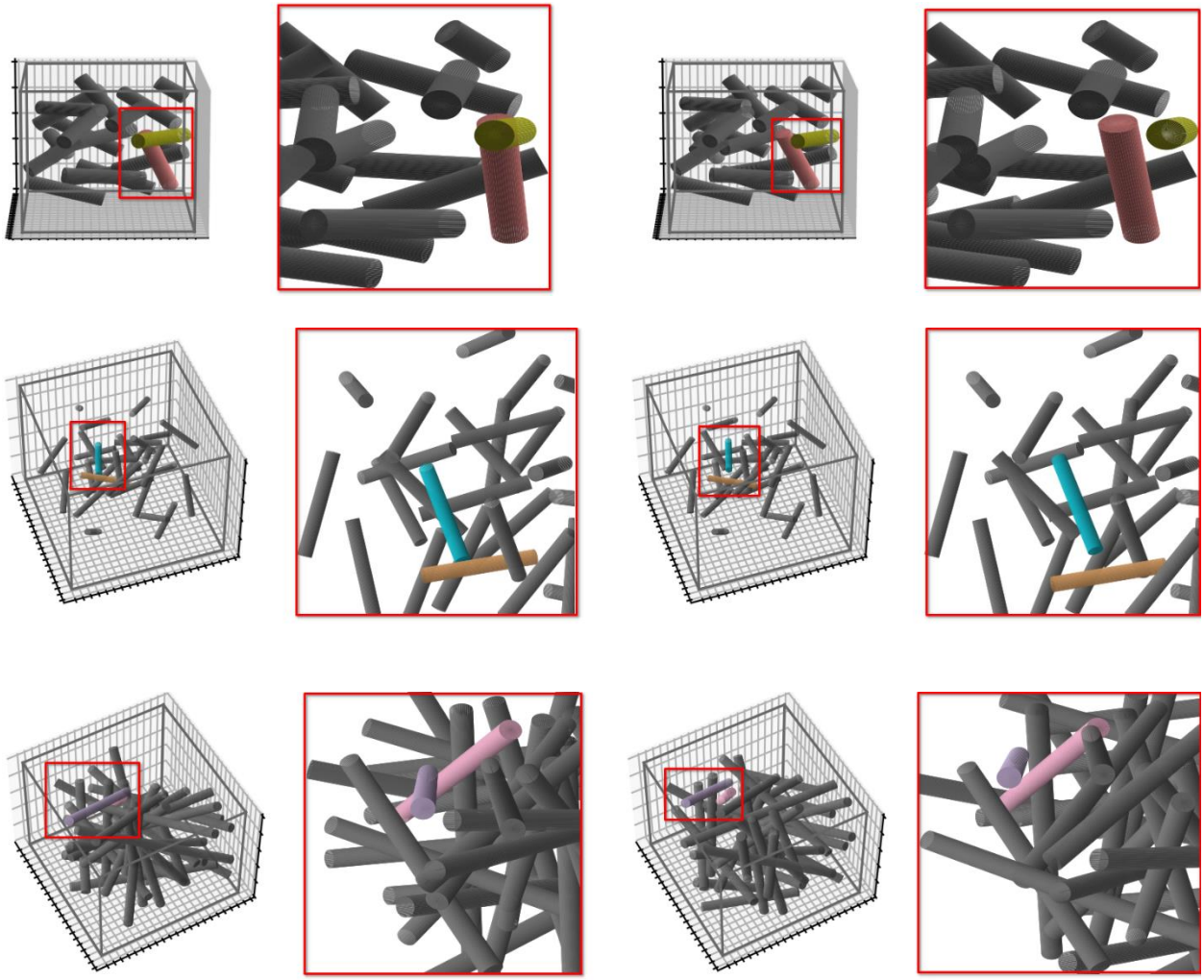


Figure 28 Comparison between the generated results without applying the loss guidance during generation (left) and those using the loss guidance to ensure the physical constraints (right).

We first demonstrate our model's capability to generate fiber-reinforced composites for stress-strain curves and configurations within the range of our training dataset. We randomly generate stress-strain curves within the range of our training dataset, as shown in Fig. 18, without them being present in the training set. For a given stress-strain curve, our system provides multiple candidate fiber configurations (Fig. 18). The stochastic nature of the reverse process of data generation using diffusion models makes it very easy for our approach to generate multiple results simultaneously. Here, for each candidate fiber configuration (n, l, d, fiber orientations), we generate six distinct material designs with volume fractions varying from 10% to 18%. We verify the generated results by FEA. All generated results of the five fiber configurations have stress-strain responses very close to the input curve. More examples of generation are shown in Fig. 22-27. It is shown that our method can generate high-quality results with diverse fiber configurations and volume fractions (from 1.70% to 19.56%), all producing stress-strain curves that closely match the input data. In cases of unidirectional fibers, the variation in fiber orientation is less than 1 degree in each of the generated composite materials. Moreover, the generated results are consistent with the underlying mechanics of composite materials. For short fibers with heterogeneous orientations, given the same fiber length and diameters, our model generates more fibers, leading to an increased volume fraction, in response to an increase in the required composite stiffness (i.e., the slope of the input stress-strain curve), as shown in Fig. 22-25. In cases of unidirectional fibers, our model increases the number of fibers along the loading direction to produce a stiffer stress-

strain curve (Fig. 26-27). Such behavior demonstrates that the physical principles of the composite mechanics have been learned and understood by our model.

Next, we evaluate our method quantitatively using the testing dataset. We take the coefficients of the fitted stress-strain curves of the samples in the test set as the condition input to generate fiber distributions and verify the generated results against the high-fidelity FEA results. First, we measure the upper-bound performance of our methods. We select the best-of-10 generated results for each testing sample based on the relative error area e_A(Method). As shown in Table 7, the average mean-absolute-errors (MAEs) between expected curve coefficients (a_1, a_2 and a_3) and the ones of generated results ($\hat{a}_1$, $\hat{a}_2$ and $\hat{a}_3$) are all less than 0.5%, and their e_A are less than 0.2%. For composites with unidirectional fibers, the fiber orientation divergence has a standard deviation less than 0.08°.

We further evaluate the average performance of 10 generated composites. For generated composites with heterogeneous orientations, the average MAEs are all less than 1% and e_A are less than 0.45%. The generated composites with unidirectional fibers have higher MAEs (≤2.42%) and e_A (≤0.96%). The maximal error of 2.42% occurs at $\hat{a}_3$for the configuration of n = 30, l = ∞, d = 10 with unidirectional fiber orientations. The orientation divergence of composites with unidirectional fibers only has a standard deviation of less than 0.31°. We can, thereby, simply regard that fibers all have the same orientations in the generated results. Note that the impact of the error of the 3rd order coefficient a_3 is much smaller than that of the 1st-order coefficient a_1. As shown in the Table 7 and 8, e_A is identical with e_{a_1} for almost all configurations, which means that the error between the generated and the expected stress-strain curves mostly comes from the error of e_{a_1}. Overall, the small value of e_A indicates that our model is able to consistently provide composite designs with stress-strain curves that closely match the expected ones.

	e_{a_1} [%]	e_{a_2} [%]	e_{a_3} [%]	e_A [%]	
Heterogeneous Orientations					
n = 10, l = 50, d = 10	0.04 ±0.08	0.05 ±0.24	0.07 ±0.41	0.04 ±0.08	
n = 15, l = 50, d = 10	0.03 ±0.04	0.04 ±0.05	0.05 ±0.07	0.03 ±0.04	
n = 20, l = 50, d = 10	0.06 ±0.06	0.15 ±0.20	0.19 ±0.27	0.06 ±0.06	
n = 25, l = 50, d = 10	0.05 ±0.07	0.06 ±0.06	0.06 ±0.07	0.05 ±0.07	
n = 30, l = 50, d = 10	0.05 ±0.09	0.06 ±0.09	0.12 ±0.27	0.06 ±0.09	
n = 35, l = 50, d = 10	0.05 ±0.06	0.09 ±0.15	0.19 ±0.48	0.05 ±0.07	
n = 40, l = 50, d = 10	0.09 ±0.11	0.11 ±0.32	0.24 ±0.62	0.09 ±0.11	
n = 45, l = 50, d = 10	0.07 ±0.07	0.17 ±0.37	0.47 ±0.28	0.07 ±0.08	
n = 50, l = 50, d = 10	0.05 ±0.05	0.07 ±0.07	0.14 ±0.16	0.05 ±0.05	
n = 30, l = 50, d = 8	0.03 ±0.04	0.04 ±0.05	0.07 ±0.17	0.03 ±0.04	
n = 30, l = 50, d = 6	0.04 ±0.05	0.10 ±0.26	0.19 ±0.67	0.04 ±0.05	
n = 30, l = 50, d = 4	0.09 ±0.09	0.04 ±0.04	0.02 ±0.04	0.09 ±0.09	
n = 30, l = 30, d = 10	0.02 ±0.02	0.03 ±0.03	0.03 ±0.02	0.02 ±0.02	
n = 30, l = 30, d = 8	0.04 ±0.03	0.06 ±0.08	0.07 ±0.08	0.03 ±0.03	
n = 30, l = 30, d = 6	0.02 ±0.02	0.05 ±0.06	0.07 ±0.08	0.02 ±0.02	
n = 30, l = 30, d = 4	0.04 ±0.04	0.02 ±0.03	0.03 ±0.12	0.04 ±0.04	
Unidirectional Orientations					e_{ori} [°]
n = 30, l = 50, d = 10	0.06 ±0.06	0.13 ±0.11	0.20 ±0.58	0.06 ±0.06	0.02
n = 30, l = 50, d = 8	0.05 ±0.06	0.07 ±0.07	0.09 ±0.09	0.05 ±0.06	0.03
n = 30, l = 50, d = 6	0.04 ±0.06	0.11 ±0.48	0.17 ±0.60	0.04 ±0.06	0.02
n = 30, l = 50, d = 4	0.16 ±0.31	0.40 ±0.34	0.40 ±0.27	0.16 ±0.31	0.07
n = 30, l = 1, d = 10	0.16 ±0.16	0.31 ±0.33	0.36 ±0.38	0.16 ±0.16	0.04
n = 30, l = 1, d = 8	0.09 ±0.09	0.16 ±0.19	0.33 ±0.17	0.09 ±0.09	0.04
n = 30, l = 1, d = 6	0.12 ±0.23	0.20 ±0.33	0.32 ±0.37	0.11 ±0.22	0.04
n = 30, l = 1, d = 4	0.16 ±0.38	0.11 ±0.32	0.27 ±0.52	0.16 ±0.38	0.04

Table 7 Quantitative evaluation using the best of 10 generated results for each testing sample. Results are reported as relative MAEs in the format of mean ± std. deviation. For configurations with unidirectional orientations, fiber orientation divergence e_{ori} is reported through the standard deviation in the unit of degrees.

	e_{a_1} [%]	e_{a_2} [%]	e_{a_3} [%]	e_A [%]	
Heterogeneous Orientations					
$n = 10, l = 50, d = 10$	0.21 ± 0.26	0.21 ± 0.33	0.27 ± 0.51	0.21 ± 0.26	
$n = 15, l = 50, d = 10$	0.20 ± 0.22	0.19 ± 0.18	0.24 ± 0.24	0.20 ± 0.22	
$n = 20, l = 50, d = 10$	0.36 ± 0.26	0.78 ± 0.60	0.91 ± 0.76	0.36 ± 0.26	
$n = 25, l = 50, d = 10$	0.28 ± 0.30	0.30 ± 0.27	0.40 ± 0.38	0.28 ± 0.30	
$n = 30, l = 50, d = 10$	0.37 ± 0.43	0.39 ± 0.41	0.55 ± 0.64	0.37 ± 0.43	
$n = 35, l = 50, d = 10$	0.32 ± 0.33	0.45 ± 0.45	0.66 ± 0.75	0.32 ± 0.33	
$n = 40, l = 50, d = 10$	0.36 ± 0.37	0.49 ± 0.58	0.82 ± 0.78	0.36 ± 0.37	
$n = 45, l = 50, d = 10$	0.44 ± 0.38	0.72 ± 0.86	0.93 ± 0.70	0.44 ± 0.38	
$n = 50, l = 50, d = 10$	0.28 ± 0.20	0.44 ± 0.32	0.79 ± 0.63	0.28 ± 0.20	
$n = 30, l = 50, d = 8$	0.18 ± 0.17	0.20 ± 0.18	0.29 ± 0.30	0.18 ± 0.17	
$n = 30, l = 50, d = 6$	0.22 ± 0.20	0.40 ± 0.50	0.53 ± 0.91	0.22 ± 0.19	
$n = 30, l = 50, d = 4$	0.27 ± 0.15	0.17 ± 0.13	0.14 ± 0.14	0.27 ± 0.15	
$n = 30, l = 30, d = 10$	0.13 ± 0.11	0.14 ± 0.10	0.19 ± 0.19	0.13 ± 0.11	
$n = 30, l = 30, d = 8$	0.26 ± 0.17	0.32 ± 0.25	0.40 ± 0.33	0.26 ± 0.17	
$n = 30, l = 30, d = 6$	0.12 ± 0.09	0.28 ± 0.21	0.34 ± 0.26	0.12 ± 0.09	
$n = 30, l = 30, d = 4$	0.14 ± 0.09	0.11 ± 0.08	0.11 ± 0.15	0.15 ± 0.09	
Unidirectional Orientations					e_{ori}
$n = 30, l = 50, d = 10$	0.38 ± 0.31	0.85 ± 0.64	1.07 ± 1.04	0.38 ± 0.31	0.04
$n = 30, l = 50, d = 8$	0.28 ± 0.23	0.39 ± 0.33	0.53 ± 0.47	0.28 ± 0.23	0.04
$n = 30, l = 50, d = 6$	0.20 ± 0.20	0.38 ± 0.70	0.50 ± 1.19	0.20 ± 0.19	0.04
$n = 30, l = 50, d = 4$	0.73 ± 0.73	1.39 ± 1.13	1.48 ± 1.00	0.74 ± 0.73	0.31
$n = 30, l = \infty, d = 10$	0.96 ± 0.79	1.71 ± 1.46	2.42 ± 1.42	0.96 ± 0.79	0.27
$n = 30, l = \infty, d = 8$	0.55 ± 0.48	0.97 ± 0.91	1.47 ± 1.69	0.55 ± 0.48	0.12
$n = 30, l = \infty, d = 6$	0.55 ± 1.27	0.73 ± 1.80	0.82 ± 1.46	0.54 ± 1.27	0.10
$n = 30, l = \infty, d = 4$	0.37 ± 0.57	0.33 ± 0.65	0.65 ± 0.66	0.37 ± 0.57	0.10

Table 8 Quantitative evaluation averaged over 10 generated results for each testing sample. Results are reported as relative MAEs. For configurations with unidirectional orientations, fiber orientation divergence e_{ori} is reported through the standard deviation in the unit of degrees.

5.4.1 Quantitative Evaluation

To perform evaluation quantitatively, we run our model to generate results using the fitted strain-stress curve coefficients from the test sets as the conditional input. Given that we have 1000 testing samples for each fiber configuration, there are 52,000 samples in the test set.

With the stochastic nature of diffusion models, our approach can easily generate a batch of random results. To evaluate the performance of our approach more comprehensively, for each testing sample, we run our model to generate 10 results for evaluation.

All the generated fiber distributions are meshed using Gmsh and simulated by FEA, as we described in Data generation. We compare the given, expected curve coefficients (a_1, a_2 and a_3) to the ones obtained by simulation on the generated results ($\hat{a}_1$, $\hat{a}_2$ and $\hat{a}_3$) through relative mean absolute error (MAE):

$$e_{a_i} = \left|\frac{a_i - \hat{a}_i}{a_i}\right| \times 100\% \tag{47}$$

where $i = 1,2,$ and 3 given the stress-strain curve in the form of Equation 9. Additionally, to compare the expected, input curve and the generated curve more intuitively, we compute the relative error area between the two curves:

$$e_A = \frac{\int_{0.1}^{0.3} |(a_1\eta + a_2\eta^2 + a_3\eta^3) - (\hat{a}_1\eta + \hat{a}_2\eta^2 + \hat{a}_3\eta^3)| d\eta}{\int_{0.1}^{0.3} (a_1\eta + a_2\eta^2 + a_3\eta^3) d\eta} \tag{48}$$

given the valid range of η is [0.1,0.3]. The smaller e_A is, the closer the expected and generated curves are.

In the main text, we already evaluate our approach using the best of 10 metric. Here, in Table 8, we show the evaluation results averaged over the 10 results generated for each testing sample. As can be seen in the table, our model can provide high-quality results with most errors less than 1%. Only the configurations with unidirectional orientations have errors bigger than 1%. This may be caused by the larger valid stress-strain response range when fibers have unified orientations, as shown Figure 26. The maximal error is only 2.42%, occurring at a_3 for the configuration of $n = 30, l = \infty, d = 10$ with unidirectional orientations. Given that the valid range of strain in our implementation is [0.1,0.3], the influence caused by the error of the 3rd order coefficient a_3 is much smaller than that of the 1st order coefficient a_1. This can be demonstrated by the error area e_A. As shown in the table, e_A is identical with e_{a_1} for almost all configurations, which means that the error areas between the generated and the expected stress-strain curves mostly come from the

error of e_{a_1}. The small value of e_A, whose mean value is less than 1% in all configurations, also indicates that our model is always able to provide results having the stress-strain curves close to the expected ones.

Note that we rely on our loss guidance term to adjust the fiber position and orientation and obtain collision-free fiber distributions. However, for the configuration of fibers with unidirectional orientations, we only update the fiber positions through the guidance loss; the orientations are gotten directly through the reverse process based on the predicted rotation noise. To evaluate our model performance to generate fibers with unidirectional orientations, we introduce e_{ori} to measure the standard deviation of the orientation error of fibers in the unit of degrees. As we can see, our unidirectional model can provide results with very small standard deviations of the orientation divergence less than 0.33°. We can, thereby, simply regard that fibers all have the same orientations in the generated results.

5.4.2 Qualitative Evaluation

In the main text, we show the result of our approach to generate multiple distribution schemes using distinct fiber configurations given the same stress-strain curve as input. To further exhibit our approach's ability to generate various results given different fiber configurations, in Fig 22 to 25, we perform evaluations using randomly generated stress-strain curves as the input to generate results. All stress-strain curves are randomly picked within the valid range as shown in Fig 18 but do not exist in the training set. For each stress-strain curve input with a candidate fiber configuration, we generate six fiber distribution schemes by random sampling (cf. Equations 31 and 25) with the initial position $p_T^i \sim N(0, I) \in R^3$ and the initial orientation obtained through the directional vector $d_T^i \sim N(0, I) \in R^3$ where $T = 500$ in our implementation. Along with each

generated scheme, we also show the volume fractions counting only the effective fiber volume inside the REV.

Figs 23 and 24 show the results of generated composites reinforced with short fibers but different numbers of fibers. Fig 25 shows the results of various fiber length and diameter combinations given the same number of fibers. In particular, Fig 26 exhibits the results of short fiber reinforced composites with unidirectional orientations, and Fig 27 exhibits the results using long fibers. As can be seen, our approach is capable of generating high-quality fiber distributions, the stress-strain properties of which are very close to the expected (input) one in all the tested cases,

with volume fraction varying from 1.70% to 19.56%. For unidirectional cases, our model can generate results with orientation errors of less than 1 degree. The random nature of the generation process enables our approach to easily generate multiple candidate schemes simultaneously.

For short fibers with heterogeneous orientations, the generated results indicate an evident correlation between the stress-strain properties and the volume fractions for the composites reinforced by short fibers with heterogeneous orientations. Illustrated in Fig 23 and 24, an increase in the number of fibers corresponds to heightened composite stiffness given the same fiber length and diameters. Fig 25 further indicates the divergence of the stress-strain curves when the fiber length and diameter change, which results in distinct volume fractions. From the results of unidirectional cases, we can see that an increase in the number of fibers along the x-direction correlates with a stiffer stress-strain curve. This behavior is attributed to the fact that when force is applied along the x-direction, fibers oriented in the same direction exhibit increased stiffness. Additionally, larger diameter fibers exhibit a stiffer stress-strain curve. The prevalence of fibers along the x-direction contributes to the composite stress-strain curve resembling that of a linear material.

In Fig 28, we provide additional results to highlight the subtle difference with and without using the proposed loss guidance during generation. Our loss guidance can effectively help generate collision-free results by applying physical constraints during the reverse (generation) process of the diffusion model.

As stated in the main text, our system employs an automatic matching mechanism to select candidate fiber configurations given an expected stress-strain curve.

This mechanism ensures that our system can always generate results using reasonable fiber configurations, and users can further pick the schemes from the candidates according to their preferences. In Fig 29, we show two failing cases, where we skip the automatic matching mechanism, and force the model to generate results using unmatched fiber configurations where the input stress-strain curves fall outside the valid range that the specific fiber configuration can support. As we can see, when given using unmatched fiber configurations, the system fails with most fibers being distributed outside the REV and overlapping with others. Though the loss guidance can help ensure the physical constraints, the large error in the generated results obviously cannot be made up only by the guided loss. The proposed automatic matching mechanism is critical to ensure generation using proper fiber configurations matched with the given stress-strain curve.

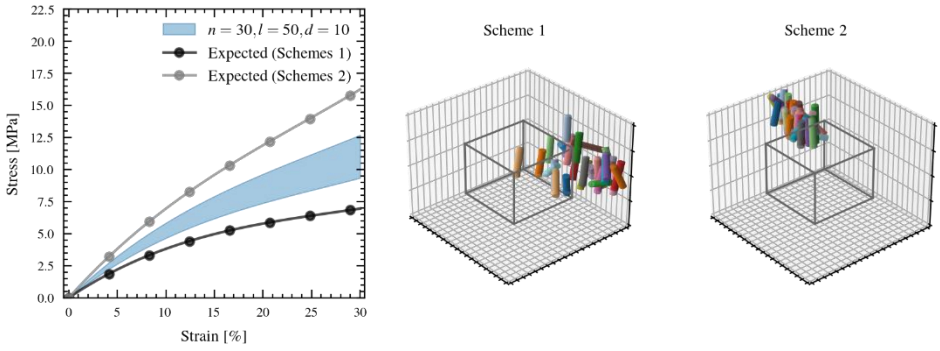


Figure 29 Failure cases where the model is asked to generate fiber distributions with an expected stress-strain curve outside the valid range. The blue range indicates the valid range of

the stress- strain curves given the target fiber configuration n = 30, l = 50, d = 10. The two marked lines are the input curves fed to the model. The two small plots show the generated results.

5.4.3 Sensitivity Evaluation

We use the configuration of $n = 30, l = 50, d = 10$ for the study of sensitive analysis.

We focus on the analysis of three key points concerning (1) the transformer architecture, (2) the usage of fiber spatial representation learning through GAT, and (3) the need for loss guidance to generate collision-free fiber distributions. Identically with the quantitative evaluation in quantitative evaluation, for each testing sample, we generate 10 results and use the average performance to evaluate the models.

Table 9 Sensitive analysis of different transformer architectures and the analysis using and not using GAT for fiber spatial representation learning. All results are studied using the configuration of n = 30, l = 50, d = 10 and reported using relative MAEs averaged over 10 generated results for each testing sample.

Heads	Layers	GAT	e_{a1}[%]	e_{a2}[%]	e_{a3}[%]	e_{a4}[%]
1	8	√	0.13	0.18	0.27	0.13
1	16	√	0.10	0.13	0.21	0.10
4	16	√	0.09	0.12	0.19	0.09
8	16	√	0.09	0.10	0.18	0.09
8	32	√	0.07	0.07	0.14	0.07
16	32	√	0.05	0.06	0.12	0.06
16	32	−	0.08	0.08	0.16	0.08

Table 9 lists the performance of our models with different transformer architectures (number of attention heads and number of decoder layers) and also the performance of the model without using GAT for fiber spatial representation learning. For the latter one, we simply pass each fiber state $[p^i, R^i] \in R^5$ through a shared network with two fully connected layers to get a 512-dimension vector as the embedding of each fiber fed into the transformer network. As shown in the table, the model performance increases consistently as the model scale, i.e. the number of attention heads and the number of decoders layers, increases. Though the GAT module has a similar size (number of parameters) to the shared embedding neural network, it can provide an overall improvement of around 25%, where e_A improves to 0.06% from 0.08%, compared to the case without using GAT.

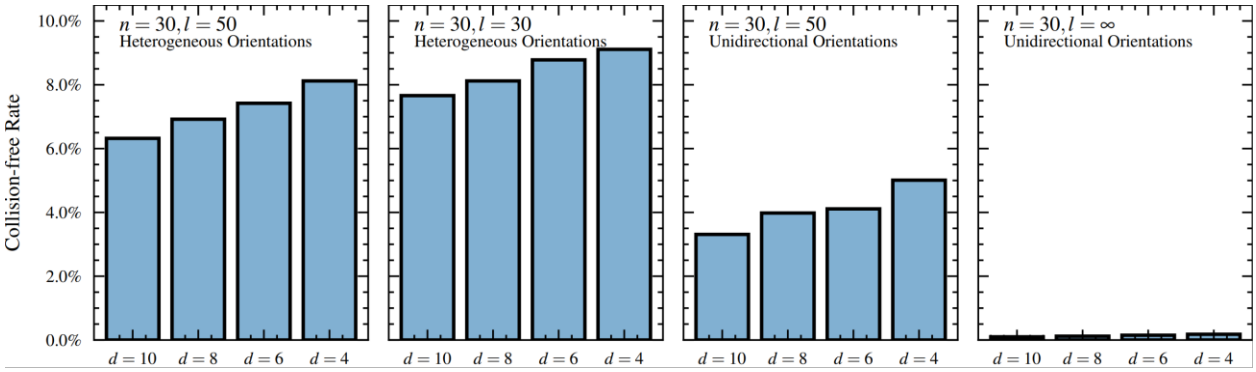


Figure 30 collision-free rate when generating fiber distributions without using the proposed loss guidance.

We also studied the success rate of generating collision-free fiber distributions with and without using the guided loss. All the results are achieved on the test set where we generate 10 results for each testing sample. Because the guided loss can ensure to obtain collision-free results,

here we only show the collision-free rate when generating fiber distributions without using the proposed loss guidance. As we can see from Fig. 30, generating through the vanilla reverse process of the diffusion model without loss guidance can only achieve collision-free results lower than 9% in all the studied cases. The collision-free rate will decrease while the volume fraction increases. In the challenge cases where fibers are distributed densely, e.g. $n = 30, l = \infty$ with unidirectional orientation, the collision-free rates are about zero. This result highlights the necessity of using the proposed guidance loss to perform physically constrained fiber distribution generation.

5.5 Generation using Out-of-the-Range Conditions

To further evaluate the generalization capability of the PC3D_Diffusion model, we examine its performance on stress-strain curves and configurations out of the range of the training datasets. First, as shown in Fig. 18, we consider configurations that are outside of the training data for a stress-strain curve that falls within the range of the training dataset but is not present in the training set. As shown in Fig. 31a-31d, composites with configuration combination out of the training data are generated. It is shown that the four generated fiber configurations all give stress-strain response curves that are very close to the expected response.

Next, we consider a more challenging task where both the configuration and stress-strain curves fall outside the range of our training data. We select a stress-strain curve that is below the envelope of stress-strain curves for heterogeneous fiber orientations, and another one below the envelope of stress-strain curves for unidirectional fibers. For both stress-strain curves, we select the configuration n=30, l=30, and d=2 (d is out of range of the training data). As shown in Fig. 31e-31f, the composites generated by the PC3D_Diffusion model can produce stress-strain response curves that closely match the expected curves that have not been seen in the training data. These

examples show the generalization capability of our model. It is important to note that this generalization capability allows for the discovery of new composite material designs with superior material responses that have not been seen before.

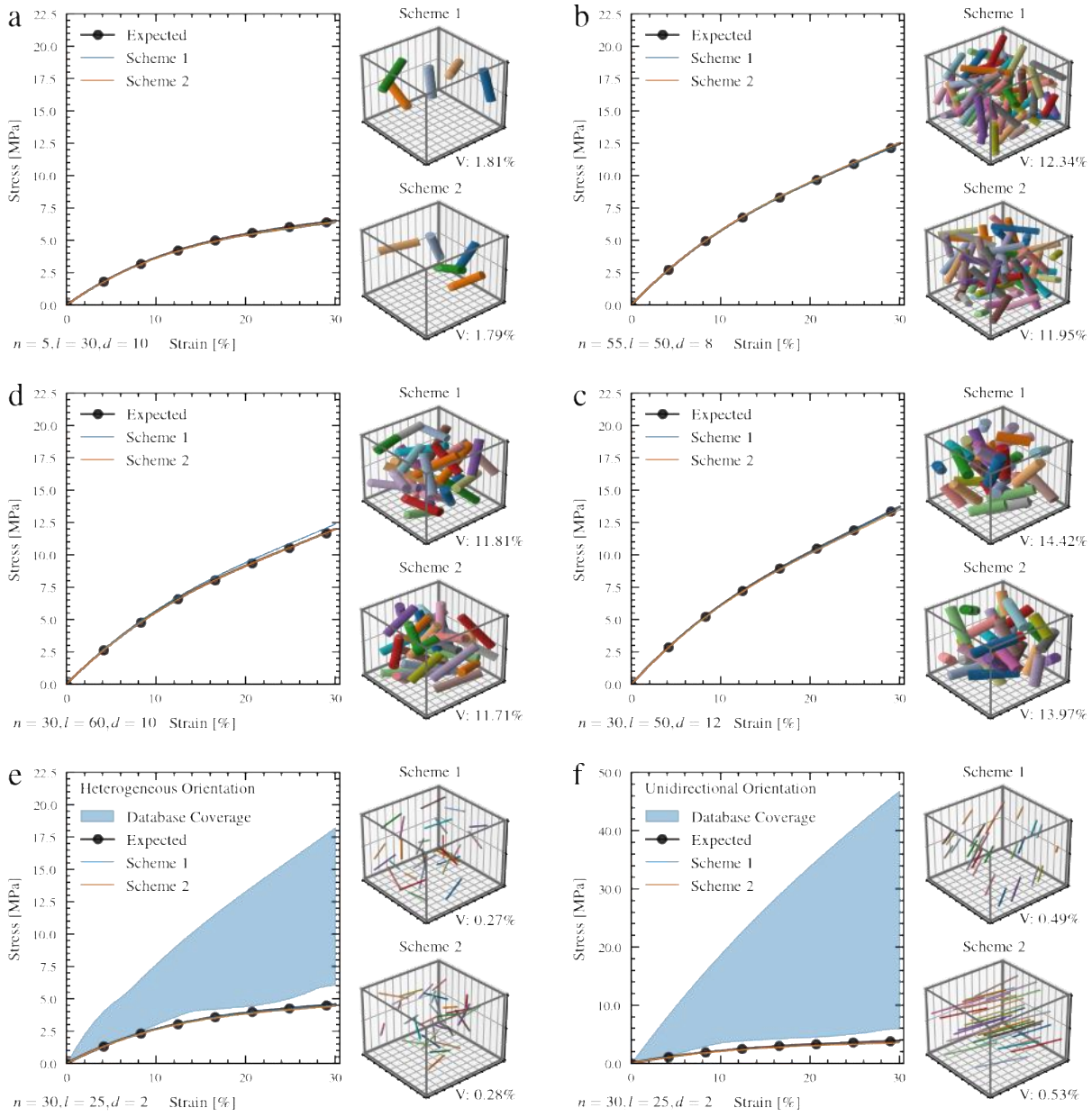


Figure 31 Examples of generated results using fiber configurations out of the training set. (a) - (c): Results generated using fiber configurations respectively with fiber amount (n), length (l)

and diameter (d) not appearing in the training set. (e) - (f): Results generated with the target stress-strain curve out of the range covered by the collected database with heterogeneous and unidirectional orientation constraints respectively. This leads to the results where the needed fiber configuration with $l = 25$ *and* $d = 2$ *falls out of the training set.*

5.6 Discussion

We have examined how transformer architecture and fiber representation impact the performance of our model. We use the configuration of n = 30, l = 50, d = 10 for the study. For each testing sample, we generate 10 results and use the average performance to evaluate the models. As shown in Table 9, the model performance increases consistently as the model scale, i.e. the number of attention heads and the number of decoders layers, increases. To evaluate the performance of the model without using GAT for fiber spatial representation learning, we simply pass each fiber state $[p^i, R^i \in R^5$ through a shared network with two fully connected layers to get a 512-dimension vector as the embedding of each fiber fed into the transformer network. The models with or without GAT have a similar size (number of parameters). However, the GAT can reduce the average errors e_{a_1}, e_{a_2}, e_{a_3} and e_A from 0.08, 0.08, 0.16, 0.08 to 0.05, 0.06, 0.12, 0.06, respectively.

Designing fiber-reinforced composites in 3D is a challenge for traditional optimization methods due to the large design space of material microstructure. It becomes even more difficult when requesting that the material design satisfies the entire nonlinear stress-strain curve. In this work, we regard the inverse design problem of fiber-reinforced composites as a generative task to decide the distribution of fiber positions and orientations. Our 3D diffusion model considers different characteristics of position and orientation properties and is particularly suitable for 3D fiber-

reinforced composites under a non-linear mechanical deformation setting. The 3D diffusion model offers the following benefits:

- By collecting 1.3 million samples for training, our data-driven model can accurately capture and reproduce the entire non-linear stress-strain response under large deformation, which is difficult, if not impossible, to achieve with traditional optimization methods.
- While the composite microstructure configuration for a given stress-strain response is inherently not unique, the stochastic nature of the diffusion model allows us to easily generate a variety of valid composite designs, which offers composite engineers multiple options to consider, along with other factors like manufacturability and cost. In comparison, traditional optimization methods typically stop searching when one design is found.
- As demonstrated by the results, the generalization capability of the 3D diffusion model enables generation of composite designs to previously unseen stress-strain curves, which facilitates discovery of new composite materials exhibiting unprecedented properties.
- The current study is designed to generate fiber reinforced polymer composites for given stress-strain curves. Our approach is generally applicable for any spatial generation task. It can be easily expanded to other types of composites such as particulate, flake or laminar composites, and to target other material properties such as viscoelastic, damage, and/or fatigue properties, albeit with additional training data, probably a larger network, and more training time.
- The designs generated by the vanilla 3D spatial diffusion model have only a low percentage that is collision-free since there is no mechanism in the vanilla 3D diffusion model to enforce physical constraints. To guarantee the generation of physically allowed fiber distributions, we propose distance-guided generation to avoid collision between fibers.

This loss guidance approach can be further extended by introducing other differentiable losses to apply additional physical constraints, e.g., fiber packing density limits imposed by manufacturing processes.

Although our model shows high-quality results in generating fiber distribution in 3D space, it has its limitations:

- The generated results are not always realizable due to the difficulty to precisely control individual fiber's position and orientation in manufacturing process. As the direction for further research, we are exploring approaches to take into account manufacturability in generating fiber distributions. Experimental validations will be carried out for the generated "manufacturing-aware" composite designs.
- For an arbitrarily given stress-strain curve, there may not be a corresponding composite design solution. In this case, the current model would generate designs that best approximate the target curve. However, it is desirable to determine a priori the feasibility of an input stress-strain curve to ensure the existence of a material design solution. To this end, an auxiliary machine learning model may be necessary to help decide what is a feasible stress-strain curve.

In addition to these limitations, it should be noted that the training data used by the 3D diffusion model is produced by the deterministic finite element analysis (FEA) model. In practice, composite materials often display small variability in properties due to variations in the matrix properties, the size and geometry of the fibers, as well as the conditions under which they are manufactured. To capture such uncertainties, it is necessary to employ a probabilistic approach (e.g. stochastic FEA models) to generate training data, which is beyond the scope of this work.

CHAPTER SIX

CONCLUSIONS

This dissertation aims to provide valuable insights into inverse design across various domains through the application of machine learning techniques. Inverse design, a method that starts with the desired outcome and works backward to determine the necessary input parameters, is significantly enhanced by machine learning. This approach allows for the development of high-performing design solutions by leveraging advanced models capable of handling complex data and predicting optimal structures.

The first research question addresses the relationship between materials' microstructures and their behaviors. By utilizing data generated from porous materials and advanced ResNet (Residual Networks) models, this study aims to establish a clear understanding of how different microstructures influence material behaviors. ResNet models, known for their deep learning capabilities, are particularly effective in capturing intricate patterns and relationships within the data. This part of the research focuses on mapping these patterns to predict the mechanical properties of materials based on their microstructural characteristics.

The second research question focuses on generating 2-D microstructures of multiphase materials based on targeted mechanical behaviors. This involves using the data and insights gained from the first research question. By employing Variational Autoencoder (VAE) models, the study successfully generates 2D microstructures that meet specific mechanical criteria. VAEs, with their ability to learn and generate new data points, are ideal for creating diverse and functional microstructures tailored to desired properties. This approach not only facilitates the design of new materials but also optimizes existing ones for improved performance.

The third research question explores the generation of 3D geometry and topology of multiphase materials based on targeted mechanical behaviors. This involves using data generated from 3D fiber composite materials and advanced Denoising Diffusion Probabilistic Models (DDPM). DDPMs are powerful generative models that excel in creating complex, high-dimensional data structures. In this context, they are used to develop detailed 3D geometries that meet specific mechanical performance criteria. This part of the research highlights the transition from 2D to 3D modeling, demonstrating the scalability and applicability of machine learning techniques in designing advanced materials.

The fourth research question examines the integration of physics and design requirements into machine learning models. Incorporating physics into machine learning models ensures that the generated designs are not only theoretically optimal but also practically feasible. By using an improved DDPM model, the study achieves collision-free fiber generation, addressing a common issue in the design of composite materials. Furthermore, this research question explores the combination of machine learning-generated designs with practical 3D printing techniques. This integration enhances the real-world applicability of the designs, ensuring that they can be manufactured efficiently and effectively. Additionally, outline future research work and the anticipated developments in this field.

Overall, this dissertation presents a comprehensive framework for utilizing machine learning in the inverse design of materials. By addressing key challenges and leveraging advanced models, it offers a pathway to designing materials with tailored properties, optimized performance, and practical applicability in various engineering domains.

CHAPTER SEVEN

FUTURE WORK: INCORPORATING MANUFACTURING REQUIREMENTS INTO ML MODELS

7.1 Manufacturing Requirements

In our recent material manufacturing design, we encountered a significant issue with the fibers generated in previous sections. These fibers exhibited an excessively high density, causing them impossible in manufacturing. This high fiber density created substantial challenges during the 3D printing process, ultimately preventing the successful completion of the manufacturing process. This issue not only impeded the production efficiency but also compromised the quality of the final product.

To address this critical problem, we are implementing a new fiber generation process based on the Voronoi diagram [127]. The Voronoi-based method offers a sophisticated approach to controlling the spatial distribution and density of the fibers. Voronoi diagrams partition space into regions based on the distance to a specific set of points, which allows for a more even and controlled distribution of fibers. By leveraging the properties of Voronoi diagrams, we can create a more uniform and manageable fiber structure that is conducive to 3D printing. This approach involves generating a set of seed points within the desired material volume and using the Voronoi algorithm to define the boundaries of the regions around each seed point. The fibers are then distributed according to these regions, ensuring that they do not clump together and maintain an appropriate density. This not only resolves the issue of fiber sticking but also enhances the overall mechanical properties and performance of the manufactured materials.

Furthermore, the Voronoi-based fiber generation process allows for greater customization and optimization of the fiber properties. By adjusting the parameters of the Voronoi diagram, such as

the number and placement of seed points, we can tailor the fiber distribution to meet specific requirements for different applications. This level of control and precision ensures that the final products are not only manufacturable but also exhibit the desired characteristics and performance. By transitioning to this advanced fiber generation technique, we aim to significantly improve the practicality and efficiency of our manufacturing process. This change will enable us to produce high-quality materials that meet stringent standards and specifications, ultimately enhancing the reliability and performance of our products.

We aim to ensure that the fibers are positioned randomly and distributed evenly among the sections. To achieve this, we will use a Voronoi diagram-based method for distributing fiber positions in each layer. This method allows for a controlled yet random spatial arrangement of fibers, ensuring uniform distribution which enhances the structural properties of the material.

During the physical manufacturing process, we encountered a significant issue related to bending. When the layers of fiber composites are not symmetrical, the product tends to bend and deform, altering its shape during the manufacturing process. This deformation negatively impacts the structural integrity and functionality of the final product.

To address this critical issue, we have decided to design the fiber composites to form a symmetric cubic structure, as illustrated in Figure 32. By ensuring symmetry in the composite layers, we can mitigate the bending problem. The symmetric arrangement of fibers balances the internal stresses and strains, preventing the product from bending and maintaining its intended shape throughout the manufacturing process. The implementation of the Voronoi diagram-based method for fiber distribution, combined with the symmetrical cubic design, offers several advantages: The random yet uniform distribution of fibers prevents clumping and ensures a consistent density, improving the overall strength and durability of the material. Symmetrical layers balance internal forces,

reducing the risk of bending and deformation during manufacturing. A more predictable and stable material behavior leads to fewer production issues and higher-quality final products. The Voronoi diagram-based method allows for precise control over fiber placement, enabling customization to meet specific performance requirements.

To achieve the desired fiber distribution, we will generate a set of seed points within each layer of the material. The Voronoi algorithm will then be used to define the boundaries around these seed points, creating regions that dictate where the fibers will be placed. This approach ensures that the fibers are evenly distributed, reducing the likelihood of high-density areas that could cause manufacturing issues. Additionally, the symmetric cubic design will involve carefully arranging the fibers in a manner that mirrors across the central axis of the composite. This symmetry ensures that any forces acting on the material are evenly distributed, preventing the development of stress concentrations that could lead to bending. By adopting these advanced techniques, we aim to significantly improve the practicality and reliability of our manufacturing process. The resulting materials will not only meet stringent quality standards but also exhibit superior performance characteristics, ensuring their suitability for a wide range of applications. This approach underscores our commitment to innovation and excellence in material manufacturing.

In summary, the combination of a Voronoi diagram-based fiber distribution method and a symmetric cubic design will enable us to produce high-quality, structurally sound materials. This strategy addresses the challenges of fiber density and bending, ensuring that our manufacturing process yields products that maintain their shape and integrity throughout production and into their final application.

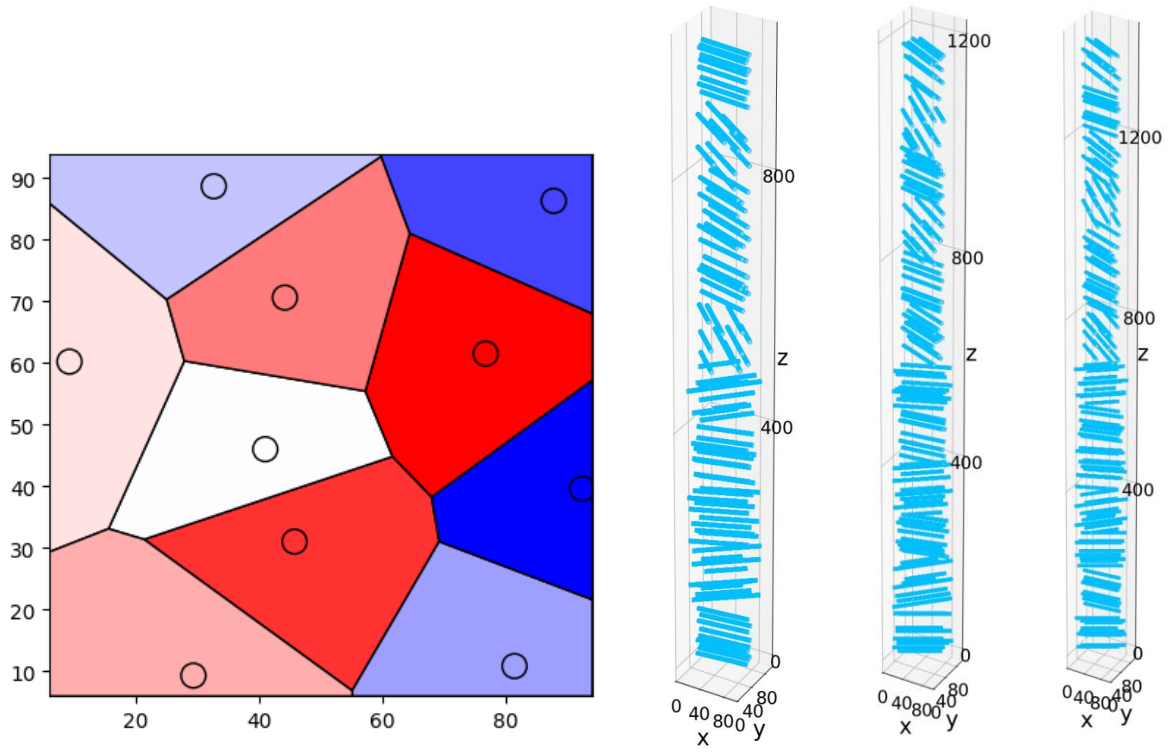


Figure 32 Left, the Voronoi diagram section of each layer, right, the figure of fiber composites during manufacturing of layer 10, 12, 14.

To enhance the microstructure of fiber composites using Voronoi diagrams, we employed Gmsh to generate a detailed mesh diagram. As depicted in Figure 33, this illustration showcases the representative volume element (RVE) of the composite material. The RVE provides a comprehensive view of the microstructural arrangement, enabling an in-depth analysis of the material properties. Additionally, the figure includes a zoomed-in section highlighting the intricate details and distribution of fibers within the composite. This meticulous representation aids in understanding the interaction between fibers and the matrix, ultimately contributing to optimizing the composite's mechanical performance.

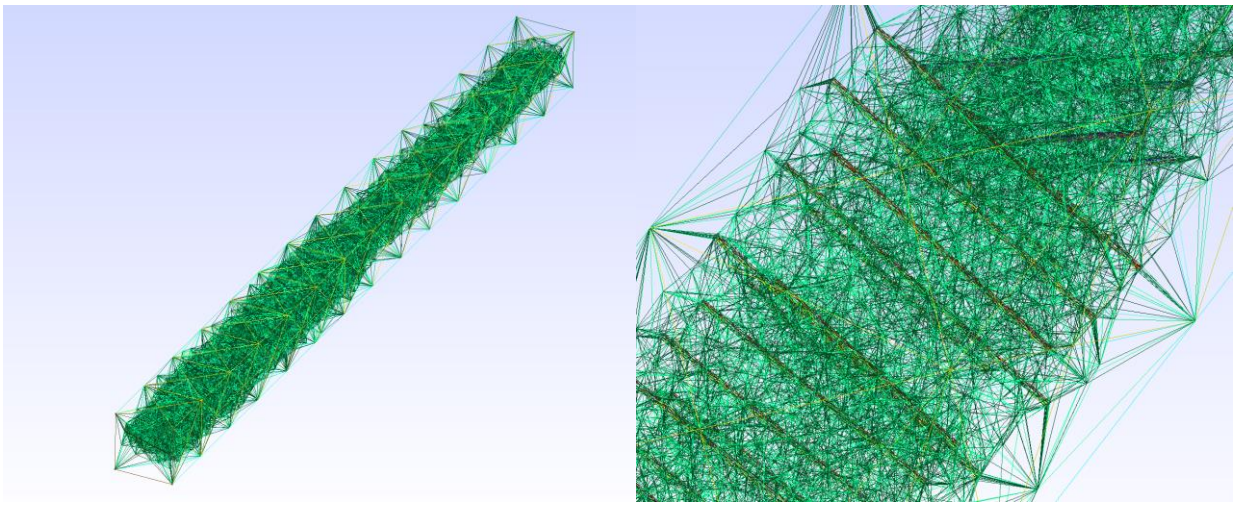


Figure 33 Left: Representative volume element of 14 layers Right: Zoom in details of fiber representative volume element.

7.2 Future Work

The next steps of the future work are as follows:

Step 1: data generation

Generate the datasets as previously described. This involves creating a comprehensive and diverse set of data points, approximately 1 million in total, to ensure the machine learning models have sufficient information to learn from. These datasets will capture various aspects of the material properties and microstructures, providing a robust foundation for model training.

Step 2: machine learning model development

Develop an efficient and suitable machine learning model tailored to our specific requirements. This step involves selecting the appropriate algorithms and techniques that can effectively handle the complexity of our data. The model should be designed to achieve high accuracy in predicting and optimizing material properties based on the generated microstructures.

Step 3: model training

Train the developed machine learning model using the generated dataset. This process includes multiple iterations of training, validation, and tuning to refine the model's performance. The aim is to achieve a mature and robust model that can accurately predict the desired material properties. This step may also involve techniques such as cross-validation, hyperparameter tuning, and performance evaluation metrics to ensure the model's reliability and accuracy.

Step 4: microstructure generation and validation

Utilize the trained machine learning model to generate material microstructures that meet the targeted material properties. This involves running simulations and optimizations to produce microstructures that align with the desired specifications. The generated microstructures will then undergo rigorous validation to ensure they meet the necessary performance criteria. This validation process includes comparing the predicted properties with experimental or known values to confirm the model's accuracy.

Step 5: 3D printing and final validation

Employ 3D printing technology to fabricate the product with the designed material microstructure. This step involves translating the digital microstructure designs into physical prototypes using advanced 3D printing techniques. Once the product is fabricated, it will undergo thorough testing and validation to ensure its accuracy in terms of material properties and performance. The final validation ensures that the 3D printed product meets all the targeted specifications and performs as expected in real-world applications.

REFERENCES

1. Wei, H., H. Bao, and X.L. Ruan, *Machine learning prediction of thermal transport in porous media with physics-based descriptors.* International Journal of Heat and Mass Transfer, 2020. **160**.
2. Kim, S.Y., S.I. Kim, and Y.S. Bae, *Machine-Learning-Based Prediction of Methane Adsorption Isotherms at Varied Temperatures for Experimental Adsorbents.* Journal of Physical Chemistry C, 2020. **124**(36): p. 19538-19547.
3. Karakoc, A. and O. Keles, *A predictive failure framework for brittle porous materials via machine learning and geometric matching methods.* Journal of Materials Science, 2020. **55**(11): p. 4734-4747.
4. Anderson, G., et al., *Attainable Volumetric Targets for Adsorption-Based Hydrogen Storage in Porous Crystals: Molecular Simulation and Machine Learning.* Journal of Physical Chemistry C, 2019. **123**(1): p. 120-130.
5. Stoll, A. and P. Benner, *Machine learning for material characterization with an application for predicting mechanical properties.* GAMM-Mitteilungen, 2021. **44**(1): p. e202100003.
6. Xie, Y.M. and G.P. Steven, *Evolutionary structural optimization for dynamic problems.* Computers & Structures, 1996. **58**(6): p. 1067-1073.
7. Bendsoe, M.P. and N. Kikuchi, *Generating Optimal Topologies in Structural Design Using a Homogenization Method.* Computer Methods in Applied Mechanics and Engineering, 1988. **71**(2): p. 197-224.
8. Kingma, D.P. and M. Welling, *Auto-encoding variational bayes.* arXiv preprint arXiv:1312.6114, 2013.

9. Arora, S. and P. Doshi, *A survey of inverse reinforcement learning: Challenges, methods and progress.* arXiv preprint arXiv:1806.06877, 2018.
10. Dan, Y., et al., *Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials.* npj Computational Materials, 2020. **6**(1): p. 1-7.
11. Gómez-Bombarelli, R., et al., *Automatic chemical design using a data-driven continuous representation of molecules.* ACS central science, 2018. **4**(2): p. 268-276.
12. Zhang, L., et al., *From machine learning to deep learning: progress in machine intelligence for rational drug discovery.* Drug discovery today, 2017. **22**(11): p. 1680-1685.
13. Pilozzi, L., et al., *Machine learning inverse problem for topological photonics.* Communications Physics, 2018. **1**(1): p. 1-7.
14. Schmidt, J., et al., *Recent advances and applications of machine learning in solid-state materials science.* npj Computational Materials, 2019. **5**(1): p. 1-36.
15. Iwasaki, Y., et al., *Machine-learning guided discovery of a new thermoelectric material.* Scientific reports, 2019. **9**(1): p. 1-7.
16. Brown, N.K., et al., *?Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains?* Materials & Design, 2022. **218**.
17. Alan Turing: Computing Machinery and Intelligence. 1950
18. Gio W, John M, : Arthur Samuel: Pioneer in Machine Learning. 1992
19. David R, Geoffrey H, Ronald W: learning representations by back-propagation errors. 1986
20. Zewen Li; Fan Liu; Wenjie Yang; Shouheng Peng; Jun Zhou;: A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. 2021

21. Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, Shahrokh Valaee;: Recent Advances in Recurrent Neural Networks. 2017
22. Sina M, Mandar P, Vasu S, Zhangyang W: Practical Solutions for Machine Learning Safety in Autonomous Vehicles. 2019
23. Li D, Xiao L. : Machine Learning Paradigms for Speech Recognition: An Overview. 2013
24. Jiatao G, Graham N, Kyunghyun C, Victor L. : Learning to Translate in Real-time with Neural Machine Translation. 2017
25. ÁngelaG, Ofélia A, Carla I, Helena P, Javier M, Javier T. : Learning to Translate in Real-time with Neural Machine Translation. 2017
26. Yufeng L, Mengke L, Hongmei Y, Huijun L, Ying F. : Predicting lattice thermal conductivity via machine learning: a mini review. 2023
27. Theodoros K,Filippos S, Christos T. : The Electrical Conductivity of Ionic Liquids: Numerical and Analytical Machine Learning Approaches. 2022
28. Christopher T, Brett T. : High temperature oxidation of corrosion resistant alloys from machine learning. 2021
29. Giuseppe C, Ignacio C, Kyle C, Laurent D, Maria S, Naftali T, Leslie V, Lenka Z,: Machine learning and the physical sciences. 2019
30. Andreas B, Nadine S, Marwin S, W. Patrick W, Ola E, Tiago R;: Machine learning and the physical sciences. 2019
31. M. Rappaz, Ch A. Gandin, J. L. Desbiolles & Ph. Thévoz;: Prediction of grain structures in various solidification processes. 1996
32. P.K. Jena, K. Ramanjeneyulu, K. Siva Kumar, T. Balakrishna Bhat;: Ballistic studies on layered structures. 2009

33. Junkil P, Aseem P, Seyed Mohamad M and Jihan K;: Inverse design of porous materials: a diffusion model approach. 2023
34. Taehoon Jung a, Jaewook Lee a, Tsuyoshi Nomura b c, Ercan M. Dede c;: Inverse design of three-dimensional fiber reinforced composites with spatially-varying fiber size and orientation using multiscale topology optimization. 2022
35. Randolph J. Nudo;: Plasticity. 2012
36. C. K. H. Dharan;: Fracture Mechanics of Composite Materials. 1978
37. J.C. Pang, S.X. Li, Z.G. Wang, Z.F. Zhang;: General relation between tensile strength and fatigue strength of metallic materials. 2013
38. M.A. Hearst; S.T. Dumais; E. Osuna; J. Platt; B. Scholkopf;: Support vector machines. 1998
39. Barry de Ville;: Decision trees. 2013
40. Adele Cutler, D. Richard Cutler & John R. Stevens;: Random Forests. 2012
41. J. Laaksonen; E. Oja;: Classification with learning k-nearest neighbors. 2002
42. Hervé Abdi, Lynne J. Williams;: Principal component analysis. 2010
43. Petros Xanthopoulos, Panos M. Pardalos & Theodore B. Trafalis;: Linear Discriminant Analysis. 2012
44. Yann LeCun, Yoshua Bengio & Geoffrey Hinton;: Deep learning. 2015
45. Dominik Maximilián Ramík, Christophe Sabourin, Ramon Moreno & Kurosh Madani;: A machine learning based intelligent vision system for autonomous object detection and recognition. 2013
46. Hyunseok Seo, Masoud Badiei Khuzani, Varun Vasudevan, Charles Huang, Hongyi Ren, Ruoxiu Xiao, Xiao Jia, Lei Xing;: Machine learning techniques for biomedical image

segmentation: An overview of technical aspects and introduction to state-of-art applications. 2020

47. Vladimir Pavlov; Vladimir Khryashchev; Evgeny Pavlov; Lev Shmaglit;: Application for video analysis based on machine learning and computer vision algorithms. 2013
48. Zhaocheng Liu, Dayu Zhu, Lakshmi Raju, Wenshan Cai;: Tackling Photonic Inverse Design with Machine Learning. 2021
49. by Jia Wang, Yingxue Wang, Yanan Chen;: Inverse Design of Materials by Machine Learning. 2022
50. Peter Dørffler Ladegaard Jensen, Tim Felle Olsen, J. Andreas Bærentzen, Niels Aage, Ole Sigmund;: Efficient inverse-designed structural infill for complex engineering structures. 2023
51. Antonia Creswell; Tom White; Vincent Dumoulin; Kai Arulkumaran; Biswa Sengupta; Anil A. Bharath;: Generative Adversarial Networks: An Overview. 2018
52. Diederik P Kingma, Max Welling;: Auto-Encoding Variational Bayes. 2013
53. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio;: Generative Adversarial Networks. 2014
54. Mehdi Mirza, Simon Osindero;: Conditional Generative Adversarial Nets. 2014
55. Tero Karras, Samuli Laine, Timo Aila;: A Style-Based Generator Architecture for Generative Adversarial Networks. 2019
56. Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros;: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. 2017
57. Murtaza Dalal, Alexander C. Li, Rohan Taori;: Autoregressive Models: What Are They Good For? 2019

58. Jonathan Ho, Ajay Jain, Pieter Abbeel;: Denoising Diffusion Probabilistic Models. 2020
59. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin;: Attention Is All You Need. 2017
60. Ali Usman; Muhammad Rafiq; Muhammad Saeed; Ali Nauman; Andreas Almqvist; Marcus Liwicki;: Machine Learning Computational Fluid Dynamics. 2021
61. Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, Ming Lei;: Machine learning in materials science. 2019
62. Peter D. Dueben and Peter Bauer;: Challenges and design choices for global weather and climate models based on machine learning. 2018
63. D.S. Weile; E. Michielssen, Genetic algorithm optimization applied to electromagnetics: a review, IEEE Transactions on Antennas and Propagation, 1997
64. R.A. Rutenbar, Simulated annealing algorithms: an overview, IEEE Circuits and Devices Magazine, 1989
65. Jia Wang ,Yingxue Wang , andYanan Chen, Inverse Design of Materials by Machine Learning, materials, 2022
66. Beibei Yan , Sheng Zhao, Jian Li, Guanyi Chen, Junyu Tao, A conceptual framework for biomass gasifier design using a semi-empirical model and heuristic algorithm, Chemical Engineering Journal, 2022
67. Nathan K. Brown, Anthony P. Garland, Georges M. Fadel, Gang Li, Deep reinforcement learning for the rapid on-demand design of mechanical metamaterials with targeted nonlinear deformation responses, Engineering Applications of Artificial Intelligence Volume 126, Part C, November 2023, 106998

68. Avni Jain, Jonathan A. Bollinger, Thomas M. Truskett;: Perspective: Inverse methods for material design. 2014

69. Wei Ma, Feng Cheng, Yihao Xu, Qinlong Wen, Yongmin Liu;: Probabilistic Representation and Inverse Design of Metamaterials Based on a Deep Generative Model with Semi-Supervised Learning Strategy. 2019

70. Yannick Augenstein and Carsten Rockstuhl;: RETURN TO ISSUEPREVARTICLENEXT Inverse Design of Nanophotonic Devices with Structural Integrity. 2020

71. Tri Doan; Jugal Kalita;: Selecting Machine Learning Algorithms Using Regression Models. 2015

72. Vishwesh Venkatraman, Sigvart Evjen, Hanna K. Knuutila, Anne Fiksdahl, Bjørn Kåre Alsberg;: Predicting ionic liquid melting points using machine learning. 2018

73. Oyeniyi Akeem Alimi; Khmaies Ouahada; Adnan M. Abu-Mahfouz;: A Review of Machine Learning Approaches to Power System Security and Stability. 2020

74. Anand Chandrasekaran, Deepak Kamal, Rohit Batra, Chiho Kim, Lihua Chen & Rampi Ramprasad;: Solving the electronic structure problem with machine learning. 2019

75. Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, Peter Battaglia;: Simple GNN Regularisation for 3D Molecular Property Prediction & Beyond. 2021

76. Zhihao Dong, Jie Feng, Yujin Ji, and Youyong Li;: SLI-GNN: A Self-Learning-Input Graph Neural Network for Predicting Crystal and Molecular Properties. 2023

77. Zhong-Hui Shen, Han-Xing Liu, Yang Shen, Jia-Mian Hu, Long-Qing Chen, Ce-Wen Nan;: Machine learning in energy storage materials. 2020

78. John R. Kitchin;: Machine learning in catalysis. 2018

79. Jorge Felipe Gaviria, Gabriel Narváez, Camilo Guillen, Luis Felipe Giraldo, Michael Bressan;: Machine learning in photovoltaic systems: A review. 2022
80. Sheela Kolluri, Jianchang Lin, Rachael Liu, Yanwei Zhang Wenwen Zhang;: Machine Learning and Artificial Intelligence in Pharmaceutical Research and Development: a Review. 2022
81. Zheng Wei, Qiu He, Yan Zhao;: Machine learning for battery research. 2022
82. Eric Musa, Francis Doherty, Bryan R Goldsmith;: Accelerating the structure search of catalysts with machine learning. 2021
83. Asif Mahmood, Jin-Liang Wang;: Machine learning for high performance organic solar cells: current scenario and future prospects. 2022
84. Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer Shanrong Zhao;: Applications of machine learning in drug discovery and development. 2019
85. Nathan K. Brown, Anthony P. Garland, Georges M. Fadel, Gang Li, Deep reinforcement learning for the rapid on-demand design of mechanical metamaterials with targeted nonlinear deformation responses, Engineering Applications of Artificial Intelligence Volume 126, Part C, November 2023, 106998
86. Michael W. Davidson and Mortimer Abramowitz;: OPTICAL MICROSCOPY. 2002
87. H. J. Leamy;: Charge collection scanning electron microscopy. 1982
88. Stuart I Wright, Matthew M Nowell, David P Field;: A Review of Strain Analysis Using Electron Backscatter Diffraction. 2011
89. O'Shea, K. and R. Nash, *An introduction to convolutional neural networks.* arXiv preprint arXiv:1511.08458, 2015.

90. Chauhan, R., K.K. Ghanshala, and R.C. Joshi, *Convolutional Neural Network (CNN) for Image Detection and Recognition.* 2018 First International Conference on Secure Cyber Computing and Communications (Icsccc 2018), 2018: p. 278-282.

91. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.

92. Cang, R. and M.Y. Ren. *Deep network-based feature extraction and reconstruction of complex material microstructures*. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2016. American Society of Mechanical Engineers.

93. He, K.M., et al., *Deep Residual Learning for Image Recognition.* 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2016: p. 770-778.

94. Higgins, I., et al., *beta-vae: Learning basic visual concepts with a constrained variational framework.* 2016.

95. Goodfellow, I., et al., *Generative adversarial networks.* Communications of the ACM, 2020. **63**(11): p. 139-144.

96. Sønderby, C.K., et al., *Ladder variational autoencoders.* Advances in neural information processing systems, 2016. **29**.

97. Maaløe, L., et al., *Biva: A very deep hierarchy of latent variables for generative modeling.* Advances in neural information processing systems, 2019. **32**.

98. J. Ho and T. Salimans, Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.

99. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.

100. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need. Advances in neural information processing systems, 2017. 30.
101. J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models. Advances in neural information processing systems, 2020. 33: p. 6840-6851.
102. Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.
103. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems 33, 6840–6851 (2020)
104. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
105. Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al.: Lion: Latent point diffusion models for 3d shape generation. Advances in Neural Information Processing Systems 35, 10021–10039 (2022)
106. iu, Z., Feng, Y., Black, M.J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdif- fusion: Score-based generative 3d mesh modeling. In: The Eleventh International Conference on Learning Representations (2023)
107. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text- to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems 35, 36479–36494 (2022)

108. anner, M., Du, Y., Tenenbaum, J., Levine, S.: Planning with diffusion for flexible behavior synthesis. In: International Conference on Machine Learning (2022)
109. Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., et al.: De novo design of protein structure and function with rfdiffusion. Nature 620(7976), 1089–1100 (2023)
110. Herron, E., Lee, X.Y., Balu, A., Pokuri, B.S.S., Ganapathysubramanian, B., Sarkar, S., Krishnamurthy, A.: Generative design of material microstructures for organic solar cells using diffusion models. In: AI for Accelerated Materials Design NeurIPS 2022 Workshop (2022)
111. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems 32 (2019)
112. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021)
113. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Work- shop on Deep Generative Models and Downstream Applications (2021)
114. J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models.* Advances in neural information processing systems, 2020. **33**: p. 6840-6851.
115. Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution.* Advances in neural information processing systems, 2019. **32**.
116. T. Savjolova, *Preface to novye metody issledovanija tekstury polikristalliceskich materialov.* Metallurgija, Moscow, 1985. **4**(2): p. 6.2.

117. Veli˘ckovi´c, P., Cucurull, G., Casanova, A., Romero, A., Li`o, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
118. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)
119. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019)
120. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.- T., Jin, A., Bos, T., Baker, L., Du, Y., et al.: Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239 (2022)
121. OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
122. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozi`ere, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
123. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7482–7491 (2018)
124. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Interna- tional Conference on Learning Representations (2019)
125. Dhariwal, P., Nichol, A.Q.: Diffusion models beat GANs on image synthesis. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021)

126. Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.-Y., Kautz, J., Chen, Y., Vah- dat, A.: Loss-guided diffusion models for plug-and-play controllable generation. In: International Conference on Machine Learning (2023)
127. Franz Aurenhammer. : Voronoi diagrams—a survey of a fundamental geometric data structure. 1991