

Modulus tunability in hierarchical architectures: A machine learning-enabled approach

Liuchao Jin^a, Kang Zhang^a, Sicong Zhou^a, Guoquan Xie^a, and Wei-Hsin Liao^{a,b}

^aDepartment of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China

^bInstitute of Intelligent Design and Manufacturing, The Chinese University of Hong Kong, Hong Kong, China

ABSTRACT

This paper presents a machine learning-enabled approach for achieving modulus tunability in hierarchical architectures, with a specific focus on the inverse design of modulus properties in both the x - and y -directions. By leveraging finite element analysis, a comprehensive dataset was generated, capturing the relationships between material allocation and directional modulus properties across isotropic and anisotropic configurations. A machine learning model was trained to predict the x - and y -directional modulus with remarkable accuracy (R-squared can reach 0.9997), enabling precise forward predictions of mechanical behavior. To achieve targeted modulus properties, an evolutionary algorithm was employed for inverse design, optimizing material allocations to match desired modulus magnitudes with high precision. This dual-direction modulus design methodology offers a powerful framework for tailoring hierarchical architectures to specific mechanical requirements, advancing applications in fields such as soft robotics, aerospace, and advanced manufacturing. The integration of machine learning and optimization highlights the potential for data-driven strategies in the design of customizable material systems.

Keywords: hierarchical architectures, modulus tunability, machine learning-enabled design, directional modulus, inverse design, evolutionary algorithm, data-driven materials design

1. INTRODUCTION

The design and optimization of materials with tunable mechanical properties have garnered significant attention in recent years, driven by the growing demand for advanced materials in applications such as aerospace,^{1–5} soft robots,^{6–10} and biomedical devices.^{11–13} Hierarchical architectures, characterized by their multi-scale structural organization, offer a promising solution for achieving such tunability. By strategically controlling material composition and distribution, hierarchical architectures enable the customization of mechanical responses, including modulus properties,¹⁴ stress-strain distribution,^{15,16} and even 4D printing effect,^{17–21} to meet specific functional requirements.

Modulus tunability is essential for engineering materials with specific directional properties. Some applications require anisotropic behavior, while others need isotropic responses. Hierarchical architectures offer great potential, but designing them is challenging. The relationship between material distribution and mechanical properties is complex. Traditional trial-and-error methods are inefficient and often overlook better design possibilities.

Recent advancements in computational and machine learning techniques have provided effective tools to address these challenges.²² Hierarchical architecture design research has primarily focused on two directions: optimizing mechanical properties, such as stiffness and toughness, and predicting or designing deformed shapes under stimuli. Numerous studies have employed machine learning to enhance forward prediction and inverse design in hierarchical architectures. For optimizing mechanical properties, Gu et al. utilized machine learning models to predict and optimize mechanical properties, achieving configurations with superior toughness and strength compared to baseline designs.^{23,24} Similarly, generative design approaches have been explored, such

Further author information: (Send correspondence to Wei-Hsin Liao)
Wei-Hsin Liao: whliao@cuhk.edu.hk

as those by Chen and Gu, who developed a generative inverse design network to optimize hierarchical architectures for maximal toughness.²⁵ Other researchers have focused on using neural networks and deep reinforcement learning to optimize architectures for specific mechanical properties, including stiffness and stress-strain behavior.^{26–31} The second research direction focuses on predicting or designing the deformed shapes of hierarchical architectures when subjected to external forces or stimuli, such as stretching or heating. Zhang and Gu employed a physics-informed neural network (PINN) to predict deformed shapes without relying on ground truth data, using the principle of energy minimization as the optimization criterion.³² Zhao et al. integrated machine learning with evolutionary algorithms to optimize grayscale material distributions in 3D-printed components, achieving desired deformed shapes.³³ Additionally, Jin et al. proposed a residual neural network combined with evolutionary optimization to design hierarchical architectures with non-rectangular deformed configurations.^{34,35} Sun et al. extended these approaches by using machine learning and evolutionary algorithms to design 4D printing effects for 2D rods and 3D plates, showcasing the versatility of such frameworks in tailoring deformation responses.^{36–38}

Despite these advances, the inverse design of directional modulus properties in hierarchical architectures has received limited attention. Designing for tunable modulus in x - and y -directions, whether isotropic or anisotropic, is critical for applications requiring precise control of material behavior. Addressing this gap, we present a machine learning-enabled framework for inverse design of modulus properties in hierarchical architectures.

In this paper, we present a machine learning-enabled framework for the inverse design of modulus properties in hierarchical architectures, focusing on tunability in the x - and y -directions. Using Finite Element Analysis (FEA), we generate a dataset of material allocations and their corresponding modulus properties, which serves as the foundation for training a highly accurate predictive model ($R^2 = 0.9997$). We further employ an evolutionary algorithm to achieve precise inverse design, optimizing material distributions to match desired modulus magnitudes. Our approach enables both isotropic and anisotropic designs, offering good flexibility in tailoring hierarchical architectures.

The remainder of this paper is organized as follows. Section 2 describes the methodology, including FEA-based dataset generation, machine learning model development, and inverse design using evolutionary algorithms. Section 3 presents results and discussions, highlighting the performance of the predictive model and the effectiveness of the inverse design framework. Finally, Section 4 concludes the paper with a discussion of potential applications and future directions.

2. METHODOLOGY

2.1 Overview

This study presents a computational framework for designing hierarchical materials with tunable mechanical properties. It combines FEA, machine learning, and an evolutionary algorithm. The workflow begins with the generation of a dataset using FEA simulations, which captures the relationship between material allocation and resulting properties, specifically the modulus in the x - and y -directions.

Next, a machine learning model is trained to perform forward prediction. This model estimates mechanical properties based on material configurations, establishing a robust relationship between input structural configurations and output properties.

Finally, an evolutionary algorithm is employed for inverse optimization. This step identifies the optimal material allocation that achieves the desired mechanical properties in the x - and y -directions. The combination of these methods ensures a systematic and efficient approach to material design.

2.2 Dataset Generation with Finite Element Analysis

The dataset generation process in this study relies on FEA simulations and requires significant computational resources. We utilize a Python-based automation process to generate the necessary datasets. This method runs Abaqus simulations using Python scripts in a multi-process framework, enabling efficient and parallel computation across 32 separate processes. The task involves generating 80,000 unique datasets, which takes four hours to complete using this approach.

The computer we use for this paper is equipped with a 13th Gen Intel® Core™ i9-13900K processor, which operates at a base frequency of 3.00 GHz. The system has 64.0 GB of installed RAM. The computer features an NVIDIA® GeForce RTX™ 4090 GPU.

In this study, the dataset generation process involves simulating the mechanical response of a 16 mm by 16 mm part that is divided into a grid of 40 by 40 voxels. Each voxel has a size of 0.4 mm by 0.4 mm. The analysis aims to investigate the behavior of two different materials in the hierarchical architecture: a soft material and a hard material. The soft material is assigned a Young's Modulus of 1.27 MPa, while the hard material has a Young's Modulus of 400 MPa. The mesh size in the simulation is 0.05 mm. To perform the analysis, a reference point is created at the boundary for displacement-based evaluation. This reference point connects to the right boundary's edges, allowing the application of constraints and loads. The left edge remains fixed, while a 5.0 mm displacement is applied to the right edge to simulate deformation. After the simulation, a Python script automatically extracts Young's modulus from the .odb (output database) file in Abaqus.

The results of the dataset generation process are visualized in Figure 1. Figure 1(a) illustrates the distribution of the ratio of soft material across the dataset, which indicates the variety in material composition. Figure 1(b) shows the relationship between the ratio of soft material and the corresponding Young's modulus. Finally, Figure 1(c) presents a comparative analysis of Young's modulus in the x -direction (E_x) and y -direction (E_y).

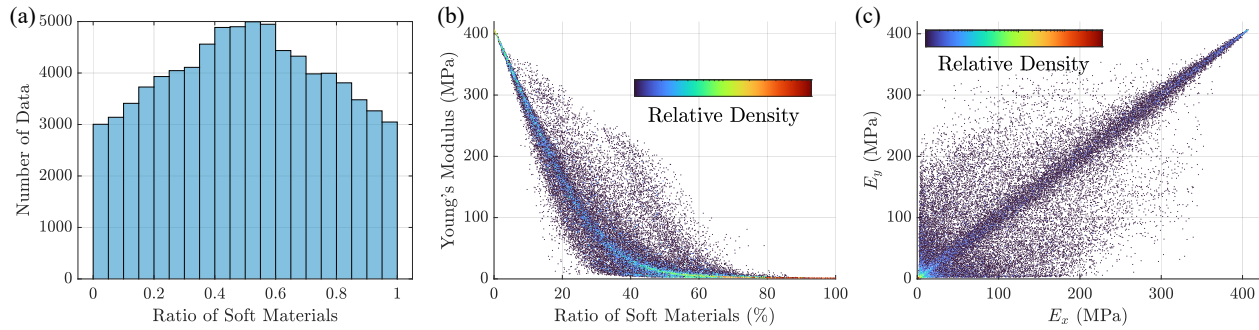


Figure 1: Visualization of dataset characteristics. (a) Distribution of the ratio of soft materials across the dataset. (b) Relationship between the ratio of soft materials and Young's modulus. (c) Comparison of Young's modulus in the x -direction (E_x) and y -direction (E_y).

2.3 Machine Learning for Forward Prediction

The machine learning framework employed for forward prediction in this study is based on a convolutional neural network (CNN) architecture, specifically designed to handle the complex relationships between input structural configurations and output material properties. The workflow integrates data preparation, model building, and training into a cohesive pipeline, ensuring accurate predictions of material behavior under varying conditions.

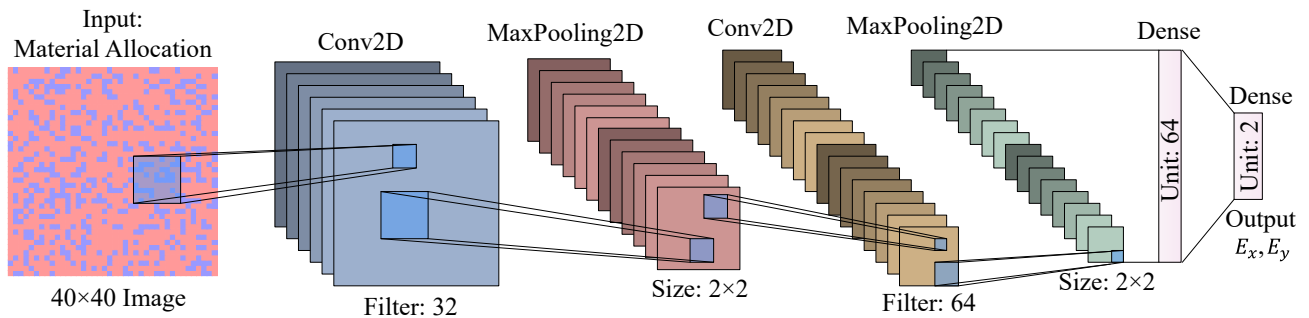


Figure 2: CNN architecture for forward prediction.

The CNN architecture used for forward prediction is shown in Figure 2. The input to the model is a matrix representing the material allocation in the hierarchical structure. This input undergoes multiple convolutional layers, including Conv2D-32 and Conv2D-64, followed by a dense layer, and finally produces the output prediction for the material properties. This design captures spatial relationships and effectively maps the input structure to its corresponding mechanical behavior.

The process begins with the preparation of a substantial dataset comprising 80,000 samples. The dataset includes input matrices and corresponding outputs for Young's modulus, both preprocessed and normalized to enhance model convergence. The input data, originally represented as 0 for hard material and 1 for soft material, is normalized to -1 and 1 . The output data normalization is based on the calculated mean and standard deviation. The data is then split into training and validation subsets, with 80% allocated for training and 20% for validation to monitor and evaluate model performance during training.

2.4 Evolutionary Algorithm for Inverse Design

The evolutionary algorithm implemented for inverse design leverages a framework combining genetic operations, fitness evaluation, and population management to optimize material configurations for specific mechanical properties. This section outlines the algorithm's methodology and mechanisms in detail.

2.4.1 Initialization and Population Generation

The algorithm begins with the generation of an initial population consisting of 1,000 individuals, each represented as a binary chromosome of length 1,600, corresponding to a 40×40 material matrix. These chromosomes encode material configurations with binary values indicating the allocation of either a soft or hard material. The initial population is created randomly to ensure diverse solutions, which forms the basis for evolutionary exploration and exploitation.

2.4.2 Fitness Evaluation and Normalization

The fitness evaluation process is designed to assess each individual in the population based on their predicted Young's modulus in the x - and y -directions. The process begins by reshaping each individual, represented as a one-dimensional array, into a $40 \times 40 \times 1$ tensor to match the input format required by the CNN model. To ensure the data is appropriately scaled for the model, the reshaped input undergoes normalization through a predefined normalization function (0 to -1 and 1 to 1). This step standardizes the values, preparing the input for accurate predictions.

After normalization, a batch dimension is added to the input data, making it compatible with the requirements of the CNN model for prediction. The model then processes the input and predicts the Young's modulus values for the x - and y -directions. These predictions are initially output as normalized values and are subsequently denormalized using predefined mean and standard deviation values. This denormalization step recovers the actual Young's modulus values, ensuring that the predictions are expressed in meaningful physical terms.

The fitness evaluation follows two objectives. The first measures the absolute difference between the predicted Young's modulus in the x -direction and the target value. This captures the deviation along the x -axis. The second measures the same difference in the y -direction and the target modulus, assessing accuracy along the y -axis. These objectives are defined as

$$F_x = |\text{Predicted modulus}_x - \text{Target modulus}_x|$$

and

$$F_y = |\text{Predicted modulus}_y - \text{Target modulus}_y|,$$

respectively.

The fitness function then returns a pair of scores, (F_x, F_y) . These scores represent the individual's performance in both directions. This approach ensures a comprehensive evaluation, guiding the optimization process toward the desired material properties.

2.4.3 Selection of Elite Individuals

The selection process prioritizes individuals with optimal Young's modulus in both directions. First, the fitness of each individual is evaluated, producing two scores: F_x and F_y . These values determine the fitness in the x - and y -directions, respectively.

The population is then sorted separately based on F_x and F_y . From these sorted lists, the top 30% of individuals with the highest F_x scores and the top 30% of individuals with the highest F_y scores are selected. These groups are referred to as e_x and e_y , representing elite individuals in the x - and y -directions, respectively.

The intersection of e_x and e_y is computed, identifying the set of overlapping individuals e_o who exhibit high fitness in both directions. These overlapping individuals are preserved to form part of the next generation, as they represent solutions with strong anisotropic properties.

Individuals in $e_x \setminus e_o$ and $e_y \setminus e_o$ (those who are not overlapping elites) are then selected as parents for crossover operations, contributing to the generation of offspring for the next population. This ensures diversity while retaining individuals with high fitness properties.

2.4.4 Genetic Operators: Crossover and Mutation

Genetic operators, specifically crossover and mutation, play a pivotal role in the evolutionary process by facilitating the generation of offspring with potentially enhanced properties. These mechanisms are instrumental in maintaining diversity within the population and enabling a thorough exploration of the solution space to identify optimal configurations.

In the proposed evolutionary algorithm, the crossover operation is designed to combine genetic material from selected parents, generating offspring with a blend of advantageous traits. The process begins with crossover among elite individuals. For these elites, crossover is performed separately along the x -direction and y -direction, with two methods applied: row exchange and column exchange. Specifically, 50% of the crossover operations are carried out via row exchange, and the remaining 50% via column exchange. During row exchange, a random number of rows is selected for crossover, with both the starting and ending indices determined randomly. Similarly, in column exchange, random column ranges are swapped between the parent individuals.

Following the crossover of elite individuals, an additional 30% of the general population, referred to as normal individuals, are randomly selected to undergo crossover. The same division of operations—50% row exchange and 50% column exchange—is applied to these individuals, ensuring the generation of diverse offspring.

The mutation operation is introduced to preserve genetic diversity and explore the search space. Individuals are selected for mutation from the entire population, with their number calculated as:

$$\text{Population size} - \text{Number of overlapping elites} - \text{Number of offspring from crossover} - 2.$$

Mutation is performed in both row and column dimensions. Each dimension is further divided into two equal parts. For the first part of row mutation, specific rows are randomly selected (with start and end indices determined randomly), and their values are flipped (i.e., $0 \rightarrow 1$ and $1 \rightarrow 0$). For the second part, values within the selected rows are replaced with randomly generated binary numbers (0 or 1). The column mutation follows an analogous procedure.

To ensure specific anisotropic performance properties, two deterministic individuals—one composed entirely of 0s and the other of 1s—are added to the offspring pool. The new population is then constructed by combining the overlapping elites, the offspring generated by crossover, the offspring generated by mutation, and the two deterministic individuals. The total population size is maintained at 1000.

2.4.5 Population Filtering and Replenishment

The evaluation process identifies the best individual in each iteration using the Root Mean Square Error (RMSE) of fitness scores. This helps ensure precise convergence toward the desired material properties.

First, overlapping elites are identified by selecting top-performing individuals in both the x - and y -directions. These elites are then evaluated using their fitness scores. The RMSE is calculated for each elite to measure their overall deviation from target properties. This produces a single scalar value, simplifying the comparison process.

The overlapping elites are then sorted based on their RMSE values in ascending order, with the individual exhibiting the lowest RMSE considered the best candidate. This sorting process ranks the elites, ensuring that the optimization prioritizes solutions with the smallest overall error.

From the sorted list, the best individual (the one with the lowest RMSE) is selected as the top-performing candidate for the current generation. This individual is stored for later analysis and reporting. Additionally, the corresponding fitness scores for the x - and y -directions are extracted and recorded for tracking the optimization progress.

2.4.6 Iterative Evolution and Optimization

The evolutionary loop iterates for 1,000 generations, continually refining the population towards configurations that minimize the RMSE between predicted and target mechanical properties. The dynamic selection, crossover, and mutation mechanisms ensure that the algorithm balances exploration of the solution space with the exploitation of high-quality solutions.

Algorithm 1: Evolutionary Algorithm for Inverse Design

Input:

- Target Young's modulus: ($\text{Modulus}_x^{\text{target}}, \text{Modulus}_y^{\text{target}}$)
- Population size: 1000
- Chromosome length: 1600
- Matrix dimensions: 40×40
- Number of generations: 1000

Output: Best material configuration with minimized RMSE.

1. Initialization:

Generate an initial population of 1000 individuals;
Each individual is represented as a binary chromosome of length 1600;

2. Iterative Evolution (for each generation):

a. Fitness Evaluation:

Reshape each chromosome to a $40 \times 40 \times 1$ tensor;
Normalize the input data;
Predict Young's modulus in the x - and y -directions using the CNN model;
Denormalize predictions to recover actual values;
Calculate fitness scores:

$$F_x = |\text{Predicted modulus}_x - \text{Target modulus}_x|, \quad F_y = |\text{Predicted modulus}_y - \text{Target modulus}_y|$$

b. Selection of Elite Individuals:

Sort population by F_x and F_y ;
Select top 30% based on F_x as e_x and F_y as e_y ;
Compute overlapping elites, $e_o = e_x \cap e_y$;
Retain e_o for the next generation;
Select $e_x \setminus e_o$ and $e_y \setminus e_o$ as parents for crossover;

Algorithm 1: Evolutionary Algorithm for Inverse Design (continued)

c. Genetic Operators:**Crossover:**

Perform row exchange for 50% of parents;
Perform column exchange for the other 50%;
Apply the same to 30% of normal individuals;

Mutation:

Calculate the number of individuals for mutation:

$$\text{Mutation_count} = \text{Population size} - |e_o| - \text{Crossover offspring} - 2$$

Perform row and column mutations:

- Flip random values in one-half of selected rows/columns.
- Replace random values in the other half.

Add two deterministic individuals (all 0s and all 1s);

Construct the new population by combining e_o , crossover offspring, mutated individuals, and deterministic individuals;

d. Population Filtering:

Evaluate RMSE for overlapping elites in e_o :

$$\text{RMSE} = \sqrt{\frac{F_x^2 + F_y^2}{2}}$$

Sort e_o by RMSE in ascending order;

Select the individual with the lowest RMSE as the best candidate;

3. Output Results:

After 1000 generations, return the best material configuration and its fitness scores.

3. RESULTS AND DISCUSSION

3.1 Forward Prediction Results

To validate the performance of our forward prediction model, we generated an additional dataset comprising 80,000 samples, each containing input material configurations and corresponding output data. The output data, representing the ground truth, was obtained through simulation.

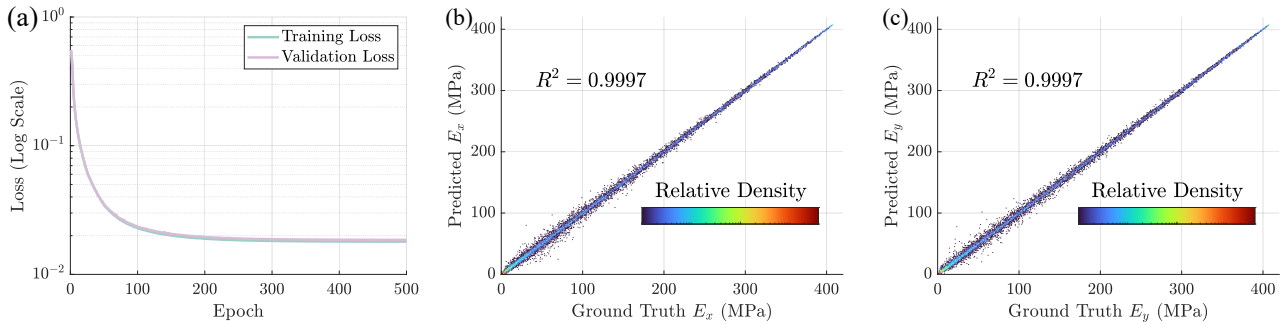


Figure 3: Performance of the forward prediction model. (a) Training loss and validation loss for the forward prediction model. (b) Comparison of ground truth E_x values with predicted E_x values. (c) Comparison of ground truth E_y values with predicted E_y values.

Using the forward prediction model, we predicted the Young's modulus values (E_x and E_y) for the 80,000 material allocations. To visually and quantitatively assess the model's performance, the training loss and the validation loss were shown in Figure 3(a), which illustrates that for both training loss and validation loss, the value will decrease to a small amount (less than 10^{-1}). We also plotted the ground truth values (x -axis) against the predicted values (y -axis) for both E_x (Figure 3(b)) and E_y (Figure 3(c)). The resulting scatter plots exhibit points that are closely aligned with the $y = x$ line, signifying an excellent agreement between the predicted and simulated values.

Moreover, the coefficient of determination (R^2) was calculated for both E_x and E_y , yielding values of 0.9997 in each case. This high R^2 score demonstrates the good predictive accuracy of our model and confirms its capability to generalize effectively to unseen data. The clustering of data points along the $y = x$ line further corroborates the robustness of the model, highlighting its reliability in forward predictions.

3.2 Inverse Design Results

To evaluate the performance of the evolutionary algorithm we designed, we selected nine target points with specified Young's modulus (E_x , E_y) values as follows:

(100, 100) MPa, (200, 100) MPa, (300, 100) MPa,
 (100, 200) MPa, (200, 200) MPa, (300, 200) MPa,
 (100, 300) MPa, (200, 300) MPa, (300, 300) MPa.

For each target point, the evolutionary algorithm was employed to determine the near-optimal material allocation. Subsequently, the machine learning model was used to predict the E_x and E_y values for these optimized material allocations. These predicted values were then compared with the ground truth E_x and E_y , which were obtained through simulation of the optimized material allocations.

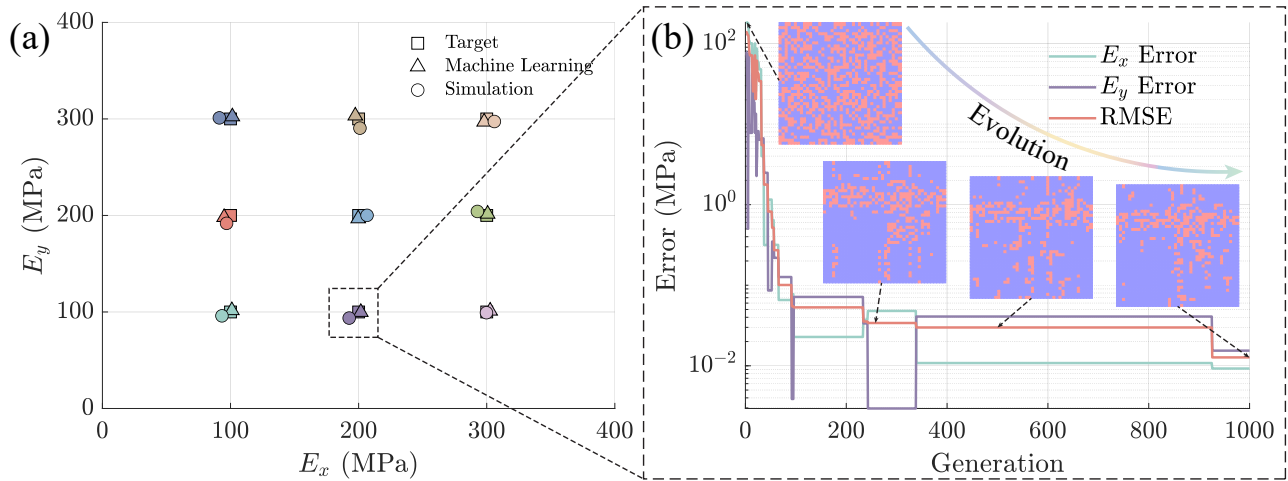


Figure 4: Performance of the inverse optimization framework. (a) Comparison of the target, machine learning-predicted results for the optimized material allocation, and simulation results for the optimized material allocation. (b) Evolution process during optimization with a target of $E_x = 200$ MPa and $E_y = 100$ MPa, illustrating the iterative improvement toward the target properties.

The results for all cases are presented in Figure 4(a). In this figure:

- Squares represent the target modulus values (E_x , E_y).
- Triangles represent the machine learning-predicted performance of the optimized material allocations.

- Circles represent the simulated (ground truth) performance of the optimized material allocations.

From Figure 4(a), it can be observed that the three types of data—target values, machine learning predictions, and simulated results—are closely aligned. This alignment demonstrates the accuracy and effectiveness of the proposed inverse optimization algorithm in achieving material allocations that satisfy the target modulus requirements.

To further illustrate the optimization process, we selected one representative case with a target modulus of (200,100) MPa. The evolution of the root mean square error over generations for this case is depicted in Figure 4(b). The RMSE decreases rapidly as the number of generations increases, eventually converging to a minimal value. This rapid convergence highlights the efficiency of the evolutionary algorithm in optimizing material configurations. The velocity of error reduction, particularly in the early generations, further underscores the robustness of our approach. These results validate the excellent performance of the evolutionary algorithm, making it a reliable tool for inverse optimization of material properties.

4. CONCLUSIONS & FUTURE WORKS

This study presented a machine learning-enabled framework for designing hierarchical architectures with tunable modulus in the x - and y -directions. The approach combined finite element analysis, convolutional neural networks, and evolutionary algorithms. It achieved high accuracy ($R^2 = 0.9997$) in predicting mechanical properties. The framework also optimized material configurations for both isotropic and anisotropic designs.

While effective, this work leaves room for further exploration. Future research could enhance material complexity by using multi-phase materials with gradient properties. This approach may improve tunability and expand the design space, leading to more versatile applications. Another promising direction is extending the framework to three-dimensional hierarchical structures. Moving beyond two-dimensional designs could unlock new possibilities and address existing limitations. Testing designed architectures under dynamic or multi-axial loading is also essential. Real-world conditions involve complex forces. Investigating these scenarios would improve durability and functionality. Experimental validation remains a key challenge. Physical prototyping with multi-material 3D printing would help bridge the gap between theory and practice. Testing real models is necessary to confirm predictions and refine designs. Finally, integrating additional properties such as thermal, electrical, or acoustic characteristics could open doors to multifunctional applications. Addressing these factors would enhance the relevance of hierarchical architectures across various industries.

ACKNOWLEDGMENTS

This work is supported by Research Grants Council (C4074-22G and STG5/E-103/24-R), Hong Kong Special Administrative Region, China, The Chinese University of Hong Kong (Project ID: 3110174). The support for Mr. Liuchao Jin provided by Research Grants Council under Hong Kong PhD Fellowship Scheme is also acknowledged.

REFERENCES

- [1] Yi, J., Lyu, R., Li, Y., Wan, C., and Zhang, K., “Stability-enhanced variable stiffness metamaterial with controllable force-transferring path,” *Advanced Functional Materials* **35**(4), 2413789 (2025).
- [2] Tiwary, A., Kumar, R., and Chohan, J. S., “A review on characteristics of composite and advanced materials used for aerospace applications,” *Materials Today: Proceedings* **51**, 865–870 (2022).
- [3] Zhou, S., Zhang, K., Jin, L., Gao, Q., and Liao, W.-H., “Efficient data driven optimization framework for designing B-spline honeycombs with excellent energy absorption,” *Thin-Walled Structures* **209**, 112941 (2025).
- [4] Li, Y., Xiao, Y., Yu, L., Ji, K., and Li, D., “A review on the tooling technologies for composites manufacturing of aerospace structures: materials, structures and processes,” *Composites Part A: Applied Science and Manufacturing* **154**, 106762 (2022).
- [5] Das, M., Sahu, S., and Parhi, D., “Composite materials and their damage detection using AI techniques for aerospace application: A brief review,” *Materials Today: Proceedings* **44**, 955–960 (2021).

- [6] Sanchez, V., Walsh, C. J., and Wood, R. J., "Textile technology for soft robotic and autonomous garments," *Advanced Functional Materials* **31**(6), 2008278 (2021).
- [7] Yang, L. and Wang, H., "High-performance electrically responsive artificial muscle materials for soft robot actuation," *Acta Biomaterialia* (2024).
- [8] Jones, T. J., Jambon-Puillet, E., Marthelot, J., and Brun, P.-T., "Bubble casting soft robotics," *Nature* **599**(7884), 229–233 (2021).
- [9] Zhang, M., Fan, X., Dong, L., Jiang, C., Weeger, O., Zhou, K., and Wang, D., "Voxel design of grayscale DLP 3D-printed soft robots," *Advanced Science* , 2309932 (2024).
- [10] Jin, L., Zhai, X., Xue, W., Zhang, K., Jiang, J., Bodaghi, M., and Liao, W.-H., "Finite element analysis, machine learning, and digital twins for soft robots: state-of-arts and perspectives," *Smart Materials and Structures* **34**(3) (2025).
- [11] Zheng, M., Sheng, T., Yu, J., Gu, Z., and Xu, C., "Microneedle biomedical devices," *Nature Reviews Bioengineering* **2**(4), 324–342 (2024).
- [12] Yao, M., Wei, Z., Li, J., Guo, Z., Yan, Z., Sun, X., Yu, Q., Wu, X., Yu, C., Yao, F., et al., "Microgel reinforced zwitterionic hydrogel coating for blood-contacting biomedical devices," *Nature Communications* **13**(1), 5339 (2022).
- [13] Hasan, M. N., Sahlan, S., Osman, K., and Mohamed Ali, M. S., "Energy harvesters for wearable electronics and biomedical devices," *Advanced Materials Technologies* **6**(3), 2000771 (2021).
- [14] Li, B., Deng, B., Shou, W., Oh, T.-H., Hu, Y., Luo, Y., Shi, L., and Matusik, W., "Computational discovery of microstructured composites with optimal stiffness-toughness trade-offs," *Science Advances* **10**(5), eadk4284 (2024).
- [15] Rashid, M. M., Pittie, T., Chakraborty, S., and Krishnan, N. A., "Learning the stress-strain fields in digital composites using Fourier neural operator," *iScience* **25**(11) (2022).
- [16] Xu, W., Zhang, M., Xu, H., Yu, M., Jin, L., Zhai, X., and Jiang, J., "INPR-Connector: Interlocking negative Poisson's ratio connectors design for deployable energy absorption structures," *Composites Part B: Engineering* **297**, 112243 (2025).
- [17] Sun, X., Zhou, K., Demoly, F., Zhao, R. R., and Qi, H. J., "Perspective: Machine learning in design for 3D/4D printing," *Journal of Applied Mechanics* **91**(3), 030801 (2024).
- [18] Wang, M., Liu, Z., Furukawa, H., Li, Z., Ge, Y., Xu, Y., Qiu, Z., Tian, Y., Wang, Z., Xu, R., et al., "Fast reverse design of 4D-printed voxelized composite structures using deep learning and evolutionary algorithm," *Advanced Science* , 2407825 (2024).
- [19] Bougzime, O., Cruz, C., André, J.-C., Zhou, K., Qi, H. J., and Demoly, F., "Neuro-symbolic artificial intelligence in accelerated design for 4D printing: Status, challenges, and perspectives," *Materials & Design* , 113737 (2025).
- [20] Athinarayanarao, D., Prod'hon, R., Chamoret, D., Qi, H. J., Bodaghi, M., André, J.-C., and Demoly, F., "Computational design for 4D printing of topology optimized multi-material active composites," *npj Computational Materials* **9**(1), 1 (2023).
- [21] Hamel, C. M., Roach, D. J., Long, K. N., Demoly, F., Dunn, M. L., and Qi, H. J., "Machine-learning based design of active composite structures for 4D printing," *Smart Materials and Structures* **28**(6), 065005 (2019).
- [22] Jin, L., Zhai, X., Wang, K., Zhang, K., Wu, D., Nazir, A., Jiang, J., and Liao, W.-H., "Big data, machine learning, and digital twin assisted additive manufacturing: A review," *Materials & Design* **244**, 113086 (2024).
- [23] Gu, G. X., Chen, C.-T., Richmond, D. J., and Buehler, M. J., "Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment," *Materials Horizons* **5**(5), 939–945 (2018).
- [24] Gu, G. X., Chen, C.-T., and Buehler, M. J., "De novo composite design based on machine learning algorithm," *Extreme Mechanics Letters* **18**, 19–28 (2018).
- [25] Chen, C.-T. and Gu, G. X., "Generative deep neural networks for inverse materials design using backpropagation and active learning," *Advanced Science* **7**(5), 1902607 (2020).

- [26] Kim, Y., Kim, Y., Yang, C., Park, K., Gu, G. X., and Ryu, S., “Deep learning framework for material design space exploration using active transfer learning and data augmentation,” *npj Computational Materials* **7**(1), 140 (2021).
- [27] Qian, C., Tan, R. K., and Ye, W., “Design of architected composite materials with an efficient, adaptive artificial neural network-based generative design method,” *Acta Materialia* **225**, 117548 (2022).
- [28] Sui, F., Guo, R., Zhang, Z., Gu, G. X., and Lin, L., “Deep reinforcement learning for digital materials design,” *ACS Materials Letters* **3**(10), 1433–1439 (2021).
- [29] Yang, Z., Yu, C.-H., and Buehler, M. J., “Deep learning model to predict complex stress and strain fields in hierarchical composites,” *Science Advances* **7**(15), eabd7416 (2021).
- [30] Yang, Z., Yu, C.-H., Guo, K., and Buehler, M. J., “End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures,” *Journal of the Mechanics and Physics of Solids* **154**, 104506 (2021).
- [31] Yang, C., Kim, Y., Ryu, S., and Gu, G. X., “Prediction of composite microstructure stress-strain curves using convolutional neural networks,” *Materials & Design* **189**, 108509 (2020).
- [32] Zhang, Z. and Gu, G. X., “Physics-informed deep learning for digital materials,” *Theoretical and Applied Mechanics Letters* **11**(1), 100220 (2021).
- [33] Zhao, B., Zhang, M., Dong, L., and Wang, D., “Design of grayscale digital light processing 3D printing block by machine learning and evolutionary algorithm,” *Composites Communications* **36**, 101395 (2022).
- [34] Jin, L., Yu, S., Cheng, J., Ye, H., Zhai, X., Jiang, J., Zhang, K., Jian, B., Bodaghi, M., Ge, Q., and Liao, W.-H., “Machine learning-driven forward prediction and inverse design for 4D printed hierarchical architecture with arbitrary shapes,” *Applied Materials Today* **40**, 102373 (2024).
- [35] Jin, L., Zhai, X., Jiang, J., Zhang, K., and Liao, W.-H., “Optimizing stimuli-based 4D printed structures: a paradigm shift in programmable material response,” in [*Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2024*], **12949**, 321–332, SPIE (2024).
- [36] Sun, X., Yue, L., Yu, L., Shao, H., Peng, X., Zhou, K., Demoly, F., Zhao, R., and Qi, H. J., “Machine learning-evolutionary algorithm enabled design for 4D-printed active composite structures,” *Advanced Functional Materials* **32**(10), 2109805 (2022).
- [37] Sun, X., Yue, L., Yu, L., Forte, C. T., Armstrong, C. D., Zhou, K., Demoly, F., Zhao, R. R., and Qi, H. J., “Machine learning-enabled forward prediction and inverse design of 4D-printed active plates,” *Nature Communications* **15**(1), 5509 (2024).
- [38] Sun, X., Yu, L., Yue, L., Zhou, K., Demoly, F., Zhao, R. R., and Qi, H. J., “Machine learning and sequential subdomain optimization for ultrafast inverse design of 4D-printed active composite structures,” *Journal of the Mechanics and Physics of Solids* , 105561 (2024).