Machine learning powered inverse design for strain fields of hierarchical architectures

Liuchao Jin [a,b] , Shouyi Yu [b], Jianxiang Cheng [b], Zhigang Liu [c,d], Kang Zhang [a], Sicong Zhou [a] , Xiangnan He [b], Guoquan Xie [a], Mahdi Bodaghi [e], Qi Ge [b,*], Wei-Hsin Liao [a,f] ,**

[a] Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China
[b] Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, 518055, China
[c] Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[d] Guangdong Provincial Key Laboratory of Functional Oxide Materials and Devices, Southern University of Science and Technology, Shenzhen 518055, China
[e] Department of Engineering, School of Science and Technology, Nottingham Trent University, Nottingham, NG11 8NS, UK
[f] Institute of Intelligent Design and Manufacturing, The Chinese University of Hong Kong, Hong Kong, China

ARTICLE INFO

ABSTRACT

Hierarchical architectures are complex structures composed of multiple materials arranged at a microstructural level to achieve specific macroscopic properties. Despite the advantages offered by hierarchical architectures which are offering broad design freedom, this extensive design space also poses significant challenges for inverse designing hierarchical architectures. This paper addresses the inverse design of strain fields for hierarchical architectures by integrating efficient forward prediction with precise inverse optimization. Forward prediction models are developed to accurately predict the physical properties and performance metrics of these materials, while inverse optimization algorithms determine the optimal material distribution to achieve desired outcomes. We propose a machine learning approach that utilizes a recurrent neural network (RNN)-based forward prediction model trained on finite element analysis data, achieving over 99% accuracy. An evolutionary algorithm-based inverse optimization model is then used to identify the optimal material configuration to reach the desired strain fields. The results, validated through simulation and experimental testing, demonstrate the potential of machine learning to accelerate the design and optimization of strain fields in hierarchical architectures, paving the way for advanced material applications in the fields of aerospace engineering, biomedical devices, robotics, structural engineering, and energy storage systems.

1. Introduction

Hierarchical architectures [1] are heterogeneous structures composed of multiple distinct material phases arranged across different length scales to achieve synergistic properties that cannot be realized by single-component materials. These architectures encompass a broad range of engineered materials, including digital materials and metamaterials. Digital materials [2] represent a subset of hierarchical architectures, where material compositions are discretized into binary or multi-phase voxel-based units, typically denoted as 0 and 1 [3], enabling precise microstructural control. The precise control over material placement in the hierarchical architecture can be enabled by advanced manufacturing techniques like 3D printing [4–7], allowing for the creation of complex, multi-material systems with tailored mechanical [8–13], thermal [14–18], and other properties.

Hierarchical architectures have a wide range of applications, including aerospace [19–25], biomedical devices [26–31], renewable energy systems [32–35], and automotive industries [36–41], where the tailored properties of these materials can lead to significant advancements in performance and efficiency.

The design of hierarchical architecture involves addressing two fundamental research questions: efficient forward prediction and precise inverse optimization. Forward prediction involves determining how a given design will behave and perform under certain stimuli. These stimuli include force and displacement for the mechanical behavior, heat, humidity, and magnetic fields for the 4D printing effect, and so on. The forward question focuses on developing models and methods to accurately predict the physical properties and performance metrics of hierarchical materials based on their structural configuration.

---

* Corresponding author.
** Correspondence to: Room 308, William M.W. Mong Engineering Building, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, 999077, China.
E-mail addresses: geq@sustech.edu.cn (Q. Ge), whliao@cuhk.edu.hk (W.-H. Liao).
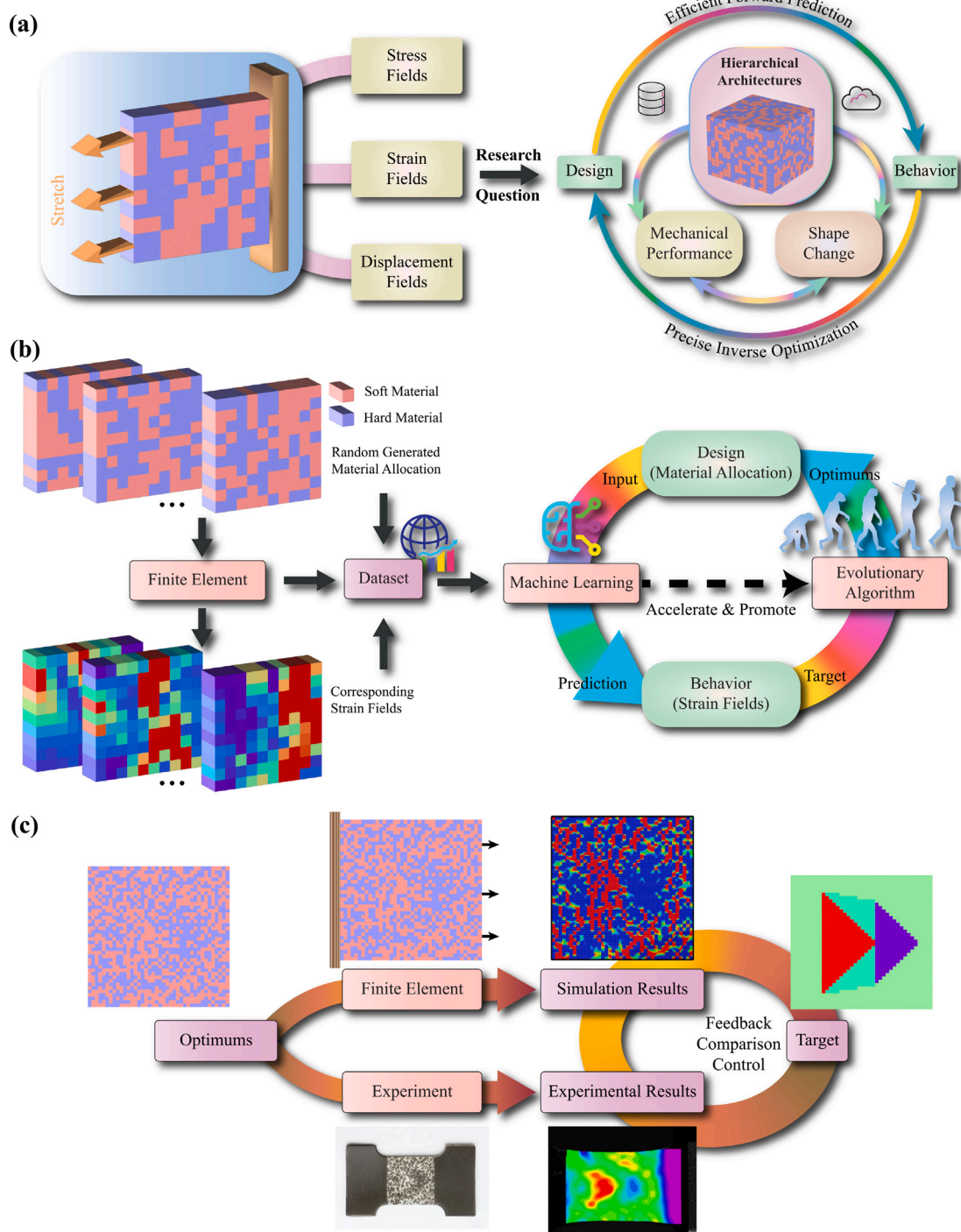
Fig. 1. Schematic diagram for proposed methods. (a). Research questions for the design of hierarchical architecture. (b). Workflows for forward prediction and inverse optimization of strain fields in hierarchical architecture. (c). Interaction among target, simulation results, and experiment results.

Inverse optimization, on the other hand, tackles the problem of designing hierarchical architectures to achieve specific desired behaviors or performance characteristics. This involves developing algorithms and frameworks that can backtrack from the desired outcomes – such as optimal mechanical behavior [42–46] and 4D printing effects [47–53] – to the structural parameters and material compositions that would produce these results. By integrating efficient forward prediction with precise inverse optimization, a robust framework for the design of hierarchical architectures can be created, as illustrated in Fig. 1(a). Forward prediction models help in understanding and simulating the behavior of potential designs, providing critical feedback for the optimization process. Inverse optimization algorithms, informed by these

predictive models, guide the exploration of the design space, ensuring that the search is both comprehensive and targeted. This integrated approach accelerates the discovery and development of advanced materials with superior properties, applicable in various fields such as sensor design [54,55], actuator design [56,57], metamaterial design [58,59], energy harvester design [60,61], and beyond.

The design of hierarchical architecture is a complex and pressing problem that attracts a lot of attention. This challenge arises from the huge design space inherent in hierarchical systems and interdependencies between different components. First, designing an effective and efficient hierarchical architecture involves navigating a vast array of potential configurations. For example, if a hierarchical architecture

contains two materials and has 1000 voxels, then its design space is $2^{1000} \approx 1.07 \times 10^{301}$, which is overwhelming. Each component must be carefully designed to ensure overall system coherence and optimal performance, making it difficult to identify the most suitable design that meets all required criteria. Furthermore, the interdependencies between different components add another aspect of complexity. Decisions made on one component may have significant influences on the entire structure, necessitating a holistic approach to design. Balancing performance, scalability, reliability, and cost-effectiveness within such a broad and complex design space requires advanced methodologies and innovative solutions.

Given these challenges, developing effective strategies and tools for hierarchical architecture design is crucial [62]. The design of hierarchical architectures can be pursued through two primary research directions: optimizing mechanical properties and predicting and inverse designing the deformed shape of the architecture when subjected to stimuli. Some researchers aim to design hierarchical architectures to achieve optimal configurations with maximal mechanical performance, including enhanced stiffness, toughness, and stress–strain behavior. Gu et al. [63,64] utilized machine learning in a hierarchical system, accurately predicting mechanical properties such as toughness and strength. The method optimized designs with significantly better properties than the mean of the input training data. The results demonstrated the ability of machine learning to efficiently search for optimal designs with limited training data, making it a promising tool for hierarchical architecture design. Exploiting more advanced machine learning algorithms, Chen & Gu [65] also presented a general-purpose inverse design approach using generative inverse design networks to obtain the optimal design of hierarchical architecture with maximized toughness. The method employed backpropagation to calculate gradients of an objective function with respect to design variables and used active learning to enhance performance and reduce training data. Kim et al. [66] proposed a deep neural network-based forward design approach, which used active transfer learning and data augmentation to search for superior materials beyond the initial training set to get the optimal design of hierarchical architecture with maximal stiffness or strength. Qian et al. [67] developed an adaptive artificial neural network-based method combined with a generative adversarial network for designing architectured composite materials with optimal toughness or stiffness. Sui et al. [68] used a deep Q network (DQN) to design hierarchical architectures for maximal equivalent modulus. Yang et al. [69] employed principal component analysis (PCA) and convolutional neural networks (CNN) to predict stress–strain curves of binary hierarchical architecture. The model achieved a mean absolute error of less than 10% using only 10%–27% of possible microstructure configurations. Li et al. [70] introduced a pipeline integrating physical experiments, numerical simulations, and artificial neural networks to discover microstructured hierarchical architecture with optimal stiffness-toughness trade-offs. This method efficiently bridged the gap between simulation and reality, and identified toughness enhancement mechanisms automatically, demonstrating a robust approach to computational design. Some other studies research on the design for deformed shape of hierarchical architecture after stretching or heating. Zhang & Gu [71] introduced a physics-informed neural network (PINN) for analyzing the deformed shape of hierarchical architecture without ground truth data, using the minimum energy criterion as the loss function. Zhao et al. [72] integrated machine learning with an evolutionary algorithm to obtain the hierarchical architectures that can follow with the desired deformed shape by optimizing grayscale distributions in digital light processing 3D printed blocks. Jin et al. [73,74] proposed the residual neural network-based-forward prediction method and evolutionary algorithm-based inverse optimization method for inverse design of 4D printed hierarchical architecture with non-rectangular shape. Sun et al. [75,76] developed machine learning methods for inverse design of 4D printed active beam and plate to achieve desired behaviors.

Few studies have focused on the inverse design for stress–strain fields hierarchical architectures. However, stress and strain fields are very important for in-depth study of the performance of hierarchical architecture, because stiffness, toughness, stress–strain curve, and deformed shape are all based on stress and strain fields. Some research focuses on efficient forward prediction for stress and strain fields of hierarchical architecture. Yang et al. [77] reported a deep learning method to predict residual strain and stress tensors from input hierarchical architecture. They also implemented similar model to predict complete strain and stress tensors [78]. The model adhered to continuum mechanical principles and improved efficiency in predicting mechanical behaviors of composites, enhancing design capabilities for multifunctional composites and hierarchical structures. Rashid et al. [79] employed a Fourier neural operator (FNO) to predict stress and strain tensor fields in 2D composites. The FNO model demonstrated high-fidelity predictions, zero-shot generalization on unseen geometries, and zero-shot super-resolution capabilities. These studies have proposed some good and efficient forward prediction methods. However, previous studies have primarily focused on optimizing mechanical properties (e.g., stiffness, toughness) or predicting stress–strain behavior. Our work uniquely advances the field by pioneering the inverse design of strain fields as functional targets, enabling applications such as strain-activated anti-counterfeiting patterns and adaptive optical modulation. Unlike conventional optimization approaches, our method does not solely aim for optimal mechanical properties but introduces a paradigm shift toward strain-driven functionalities.

In this paper, we will propose a machine learning approach to inversely design the strain fields of hierarchical architectures as demonstrated in Fig. 1(b). We will develop a recurrent neural network (RNN)-based forward prediction model that leverages finite element analysis data to rapidly and accurately predict the strain field of a component given its material allocation, achieving an accuracy of over 99%. Building on this, we will introduce an inverse optimization model powered by evolutionary algorithms to determine the optimal material distribution that aligns the strain field with preset performance goals, demonstrating the potential of machine learning to accelerate material design and optimization of strain fields for hierarchical architectures. The optimal material allocation determined by the inverse optimization model will be input into finite element analysis to obtain simulation results of the optimal solution as described in Fig. 1(c). This optimal material configuration will also be fabricated using multi-material digital light processing (DLP). The resulting physical object will be subjected to tensile testing, and the experimental strain field will be measured using digital image correlation (DIC). By comparing the experimental results, simulation data, and target strain fields, we will validate and refine our design approach.

2. Materials and methods

2.1. Specimen manufacturing

In this paper, we use centrifugal multi-material digital light processing 3D printing technology to manufacture the designed hierarchical architecture [80–83], inspired by the principle of non-contact resin removal via centrifugal force. As shown in Fig. 2(a), the system employs a bottom projection approach where UV light patterns are projected upward through a transparent glass plate to cure liquid resin in sequential layers. Two resin tanks — containing Agilus (soft material) and Vero (hard material) — are positioned beneath the printing platform.

This centrifugal multi-material digital light processing 3D printing technology represents a significant advancement over conventional multi-material 3D printing methods. Traditional approaches rely on direct contact with wipers [84–86] or fluid flows [87–89] to remove residual resin during material switching, which often contaminates the printed part or limits the maximum build area. In contrast, our method employs centrifugal force (6000 rpm for 60 s) to remove uncured resin
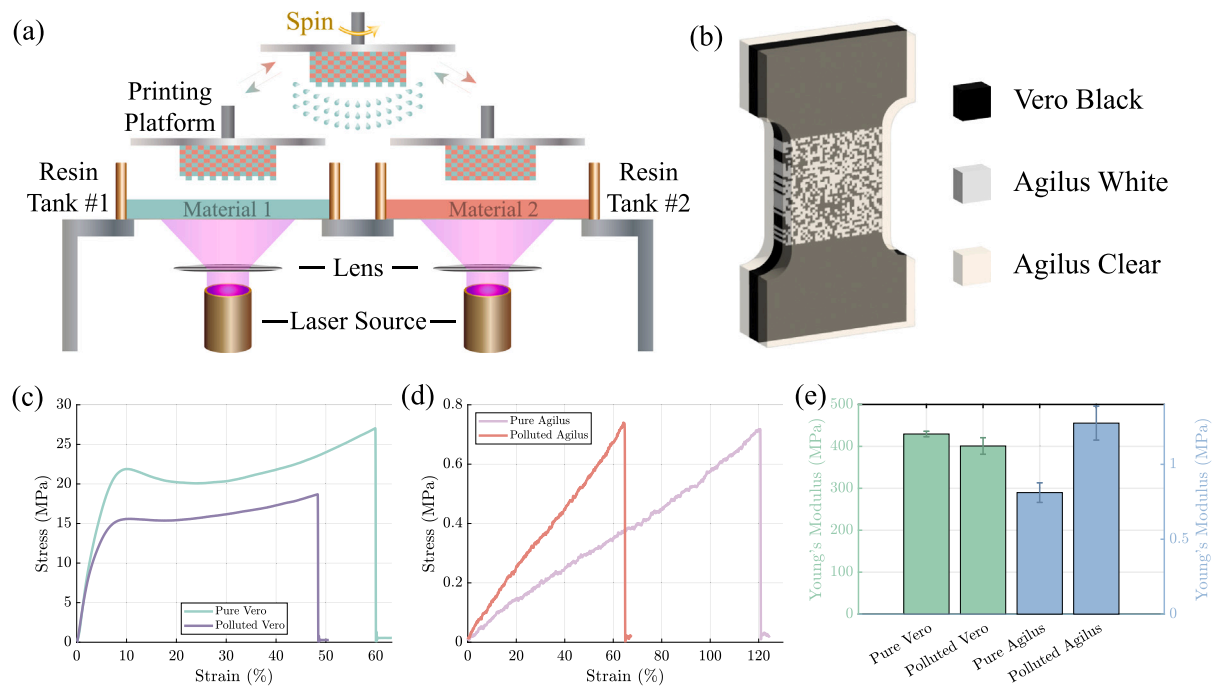
Fig. 2. Specimen manufacturing, part design, and material characterization. (a). Schematic diagram for the multi-material digital light process technique used in this paper. (b). Design of dog-bone structure for tensile test. (c). Tensile test results for Vero (hard material). (d). Tensile test results for Agilus (soft material). (e). Comparison of the modulus for hard and soft materials.

Table 1
3D printing parameters for hierarchical architectures.

Parameter	Agilus white (Soft)	Vero black (Hard)
UV intensity (mW/cm^2)	12.16	16.22
Exposure time (s/layer)	10	12
Layer thickness (μm)	100	
Centrifugation speed (rpm)	6000	
Centrifugation time (s/layer)	60	
Post-curing time (min)	10	

without physical contact, inspired by the efficiency of mammalian body-shaking mechanisms [80]. The system's ability to handle resins with viscosities spanning 10^{-3} to 10^1 Pa s allows the simultaneous use of soft Agilus White and stiff Vero Black without compromising resolution.

The printing process begins by lowering the printing platform into the first resin tank (Agilus White) to a layer thickness of 100 μm. UV light with an intensity of 12.16 mW/cm^2 is projected for 10 s to cure the soft material. After solidification of each material, the platform is lifted and rotated at 6000 rpm for 60 s to centrifugally remove uncured resin from both the printing platform and the printed part, ensuring minimal cross-contamination between materials. The printing platform is then reset to its original configuration, and then the process is repeated for the second resin tank (Vero Black), where UV intensity is set to 16.22 mW/cm^2 with a 12 s exposure time to solidify the stiff material. After printing the second material, centrifuge again to remove the adhered uncured resin. This cycle of printing and centrifugation repeats layer by layer until the printing is completed. Post-processing included rinsing in ethanol for 5 min to remove residual resin and UV post-curing at 365 nm for 10 min to achieve full material crosslinking. The key parameters for 3D printing are summarized in Table 1.

The resolution of the UV light source is 1920 × 1080 dpi in the *x* and *y* directions, with each dpi corresponding to 20 μm. In the *z* direction, the minimum accuracy of our printer can reach 10 μm. For our printing, we set the size of each voxel to 400 μm by 400 μm, with a layer thickness of 100 μm. To achieve optimal results, we designed the structure shown in Fig. 2(b). The black material represents the

hard material, the white indicates the soft material, and the light yellow signifies the transparent soft material. The part corresponding to the optimized material distribution is positioned in the middle of the entire structure, flanked by two hard dog bones at both ends to facilitate clamping by the tensile test machine. Additionally, the entire structure is wrapped with a layer of transparent soft material above and below the hierarchical architecture layer. This ensures the structure maintains its integrity during tensile testing and prevents premature breakage due to insufficient bonding between adjacent voxels in the hierarchical architecture. During the printing, we first print four layers of transparent soft material, followed by four layers of dog bones with optimized material distribution, and finally, four layers of transparent material, resulting in a total thickness of 1.2 mm. The procedure for the hierarchical architecture manufacturing using multi-material digital light processing is demonstrated in Figure S1.

2.2. Material selection and characterization

To achieve optimal performance in the hierarchical architecture, it was essential to select two materials with significantly different stiffness properties, designated as the soft and hard materials. For this study, we selected Agilus as the soft material, available in two colors (white and clear), and Vero Black as the hard material. As depicted in Fig. 5(b), the optimized material allocation within the central structure, along with the dog bones at both ends, was fabricated using Agilus White and Vero Black. The outermost layers, consisting of a transparent soft material, were printed using Agilus Clear, effectively encapsulating the core structure.

In addition to material selection, it was imperative to thoroughly characterize the mechanical properties of both the soft and hard materials. To this end, we fabricated standard dog bone specimens of each material and subjected them to tensile testing using a universal testing machine (Model E45, MTS Systems Corporation, USA). The resulting stress–strain data are presented in Figs. 2(c) and (d). During this characterization process, we observed notable differences between the mechanical properties of pure specimens and those of specimens impregnated with another material following multi-material

printing. Despite the centrifugation process used to remove uncured resin, residual traces of the other material were inevitably retained, leading to modifications in the mechanical behavior of the impregnated specimens. Specifically, the hard material exhibited a reduction in stiffness, while the soft material became slightly more rigid. Additionally, the fracture strain of both materials decreased as a result of the impregnation.

To clearly differentiate between the two conditions, we refer to the unimpregnated material as "pure material" and the impregnated material as "polluted material." To characterize material contamination during multi-material printing, two sets of dog-bone specimens were fabricated: pure and polluted. Pure specimens were printed entirely within a single resin tank. For polluted specimens, after printing each layer of the target material (e.g., Agilus), the part was immersed in the second material tank (e.g., Vero) and centrifuged (6000 rpm, 60 s) to replicate residual contamination. This process ensured that pollution levels mirrored those in actual multi-material printing. Centrifugation parameters and material viscosity were held constant, resulting in uniform contamination across polluted specimens. The tensile test results for Vero are shown in Fig. 2(c), while the corresponding results for Agilus are displayed in Fig. 2(d). We conducted eight tensile tests for each material and compiled Young's modulus data for both pure and polluted Vero and Agilus, as summarized in Fig. 2(e). From the results, it is evident that the stiffness of Vero is 134 times greater than that of Agilus, which satisfies our design requirements for creating a hierarchical architecture with distinct material properties.

2.3. Simulation

The simulation process involves the utilization of Dassault Systèmes' Abaqus (Dassault Systèmes SIMULIA Established Products 2022, Johnston, United States) finite element simulator to automatically generate the data necessary for training a forward prediction model. The simulator predicts the strain field based on the distribution of soft and hard materials. Python scripts are employed to automatically create Abaqus input files (.inp) tailored to our specific requirements. These input files encapsulate the parameters and conditions needed to simulate the behavior of the hierarchical architecture. The Neo-Hookean material model is applied to two materials in a 2D setting, specifically Agilus (soft material) and Vero (hard material).

To improve the accuracy of the simulation, each voxel is subdivided into a 10×10 mesh, resulting in 100 elements per voxel. The hierarchical architecture is discretized using CPE8H elements (eight-node hybrid quadratic plane strain elements) to mitigate volumetric locking effects. Displacement boundary conditions in the tensile direction (x-axis) are enforced with one end fixed (u_x, $u_z = 0$ at the whole edge, $u_y = 0$ at the center vertex of the edge), while a displacement of 5 mm is applied to the other end. A fixed displacement is used instead of a fixed force to account for variability in Young's modulus due to different material distributions, preventing inconsistent deformations across different structures. An implicit solver is chosen for time integration to enhance numerical stability.

To ensure diversity in the simulation data, the proportion of soft material in the overall structure is uniformly distributed, as shown in Fig. 3(a). The simulation and forward prediction model training employ binary numbers (0s and 1s) to represent the hard and soft materials, respectively, as illustrated in Fig. 3(b). The strain field distribution is extracted from the Green–Lagrange strain tensor, facilitating direct comparison with experimental digital image correlation results to validate the numerical predictions.

To maximize throughput, 48 different material allocations are batched and simulated concurrently, achieving optimal CPU utilization. The finite element simulations are performed on a workstation equipped with a 13th Gen Intel® Core™ i9-13900K processor (3.00 GHz) and 64 GB RAM. This method significantly reduces the time required to obtain simulation data, enabling the acquisition of 48 data sets in just 2 min, compared to the 2 min typically required for a single data set.

2.4. Forward prediction model

2.4.1. Model architecture

In this paper, we employ RNN as the forward prediction model for the strain fields of hierarchical architectures. RNN offers significant advantages in the forward prediction of strain fields for hierarchical architectures. One of the primary benefits is its ability to effectively capture and utilize sequential dependencies within the data. In the context of predicting strain fields, the spatial relationships and interactions between soft and hard materials and regions within the hierarchical structure can be inherently sequential. RNN, with its unique architecture designed to process sequences of data, is adept at learning these dependencies, leading to more accurate and reliable predictions of strain fields across complex material distributions. Moreover, RNN is good at handling time-series data and can be particularly beneficial when dealing with the dynamic response of materials under varying conditions. By leveraging their internal memory, RNNs can remember previous states and use this information to inform future predictions. This capability is important for modeling the behavior of hierarchical materials, where the current strain state can be influenced by the material's previous states and interactions. Consequently, RNN can provide a more precise and detailed understanding of how hierarchical materials will respond under different stress and strain conditions [90–92].

The RNN model we build employs a Bidirectional Long Short-Term Memory (LSTM) architecture implemented in TensorFlow. Machine learning models were implemented in Python 3.7.16, leveraging TensorFlow 2.5.0 with GPU acceleration. The hyperparameters for this model are listed in Table 2. The RNN model is composed of three types of layers — input, hidden, and output layers — as illustrated in Fig. 3(c). The input layer is 40×40 binary array representing the material allocation. The hidden layer is designed to sequentially handle the input data, capturing the intricate dependencies and interactions between the voxels. As the RNN processes the input through its hidden layers, it leverages its recurrent structure to maintain and update information about previous voxel states, thereby understanding the material's spatial relationships and mechanical properties over the entire grid. The core of the model is a Bidirectional LSTM layer with 25 units in each direction (forward and backward), totaling 50 hidden units. This layer processes the input sequence bidirectionally to capture spatial dependencies across the hierarchical architecture. The LSTM uses hyperbolic tangent (tanh) activation for cell state updates and sigmoid activation for input/forget/output gating. A recurrent dropout rate of 0.2 is applied during training to prevent overfitting. Finally, the output layer of our RNN model translates the processed information into a 40 by 40 array representing the elongation of each voxel in the x direction. The output layer is a fully connected dense layer with 40 neurons and linear activation, which reconstructs the predicted strain field into a 40×40 matrix. This output layer mirrors the input layer's structure, providing a direct and interpretable mapping from material distribution to mechanical response.

2.4.2. Data preprocessing and training

After obtaining the data from the Abaqus simulation, it undergoes a processing phase to prepare it for model training. This preparation involves normalizing both the input and output data to ensure they possess suitable characteristics for the training process. For the input data, which consists of matrices indicating hard (0) and soft (1) material allocations, we apply a specific normalization equation: $x_n = 2x - 1$, where x represents the original input data and x_n is the normalized version. This transformation shifts the original values of 0 and 1 to -1 and 1, respectively, centering the mean near 0 and the standard deviation near 1. This adjustment facilitates better convergence during the training phase. For the output data, normalization is performed using the mean and standard deviation of the dataset. We use the formula $y_n = (y - y_{mean})/y_{std}$, where y represents the original output data, y_n is the normalized output data, y_{mean} is the mean, and y_{std} is
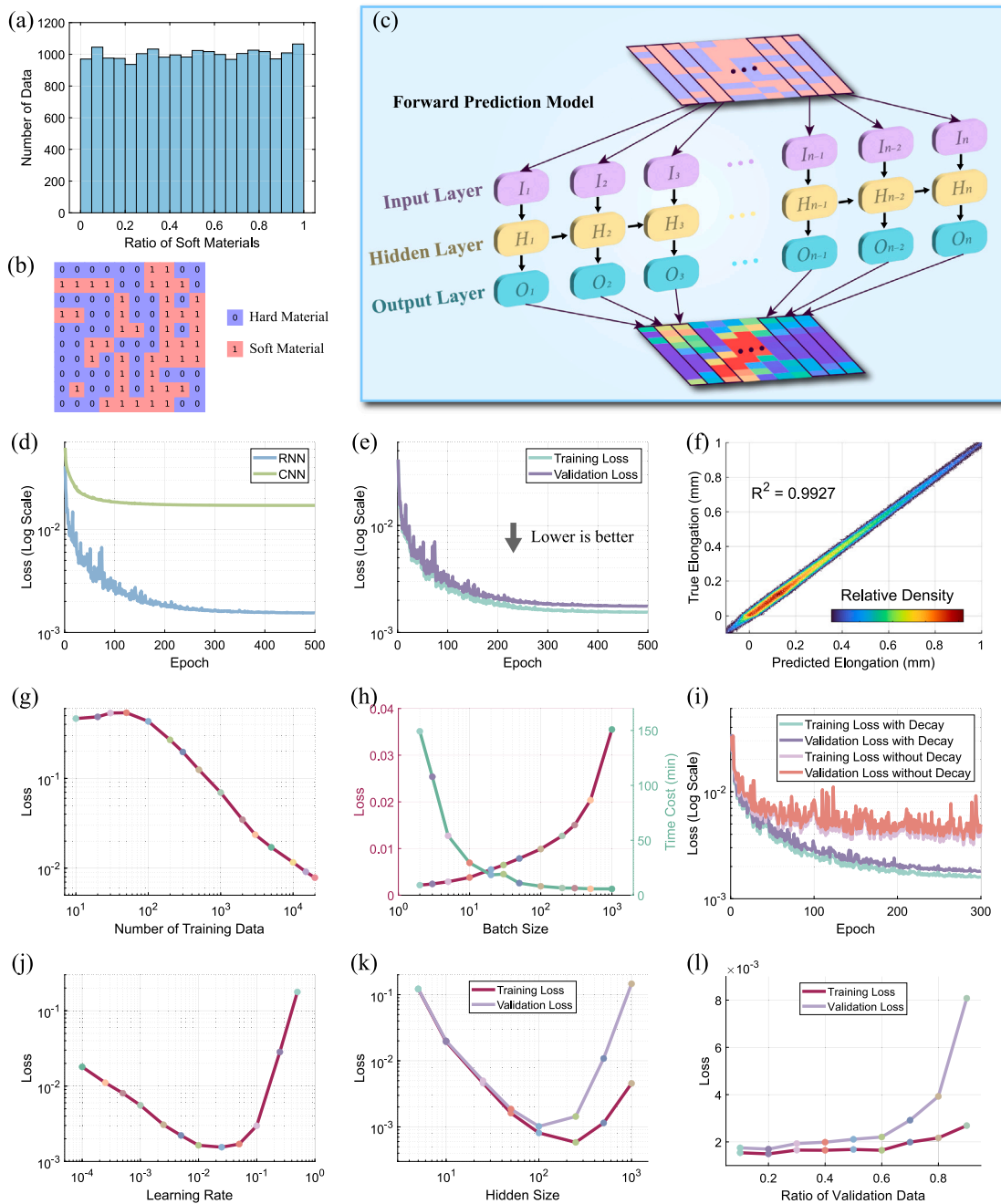
Fig. 3. Forward prediction model. (a). The distribution of 20,000 data in the dataset considering the ratio of soft materials. (b). Digital representation of soft material and hard material. (c). The schematic diagram for the RNN-based forward prediction model. (d). Loss comparison between RNN and CNN model for forward prediction. (e). Training loss and validation loss with optimal hyperparameters. (f). Comparison of machine learning predicted and finite element analysis calculated strain fields. (g-l). The influence of (g) the number of training data, (h) batch size, (i) learning rate decay, (j) learning rate, (k) hidden size, and (l) ratio of validation data on loss.

the standard deviation of all output data. This process ensures that the output data is transformed to have a mean of 0 and a standard deviation of 1, thereby promoting consistency and stability during training.

Following the normalization process, the data is split into training and validation sets. This split is crucial for evaluating the model's performance on unseen data and mitigating overfitting. Typically, 80% of the normalized data is allocated to the training set, while the remaining 20% is reserved for the validation set. This allocation strategy ensures that the model is trained on a substantial amount of data while retaining a separate portion for evaluation, thereby providing a robust framework for assessing the model's generalization capabilities.

Once the data and model architecture are prepared, the training process starts. The machine learning training is conducted on a workstation equipped with a GPU–an NVIDIA GeForce RTX 4090, which

accelerates the training of the deep learning models. The model is compiled using the Adam optimizer, a popular choice due to its efficiency and adaptability in handling sparse gradients and noisy data. The mean squared error (MSE) is selected as the loss function, which quantifies the discrepancy between the predicted outputs and the actual outputs by averaging the squares of the errors. This choice of loss function helps ensure that the model accurately captures the underlying patterns in the data by minimizing large errors. To enhance the training process and prevent overfitting, early stopping is implemented, which involves continuously monitoring the validation loss during training. If the validation loss does not show improvement for a predetermined number of consecutive epochs, the training process is halted. Early stopping ensures that the model does not overfit the training data, thereby maintaining its generalization capabilities on unseen data.

Table 2
Hyperparameters of the bidirectional LSTM model.

Parameter	Value
Input dimensions	40×40
Output dimensions	40×40
Hidden layer type	Bidirectional LSTM
Units per direction	25
Total hidden units	50
Learning rate	0.025
Learning rate decay	$1/\sqrt{2}$ every 25 epochs
Batch size	50
Maximum epochs	500
Validation split	20%
Early stopping patience	25 epochs
Loss function	Mean Squared Error (MSE)
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)

This approach strikes a balance between underfitting and overfitting, leading to a more robust and reliable model.

2.4.3. Model validation and comparative analysis

To explore the efficacy of RNN and CNN as forward prediction models, we analyzed their performance on the task of minimizing prediction errors. Fig. 3(d) illustrates the comparative results, demonstrating the advantage of using RNN over CNN. Specifically, RNN was able to achieve lower prediction errors across multiple test cases, confirming its suitability for our model. The recurrent nature of RNN allows it to effectively capture temporal dependencies and sequential patterns within the data, which is crucial for accurate predictions in our application. In contrast, CNN, despite its strong capability in handling spatial data, falls short in this context due to its limited ability to model sequential dependencies.

The performance of the forward prediction model for strain fields of hierarchical architecture is good. As depicted in Fig. 3(e), with an increasing number of epochs, the model's loss converges to a remarkably low value, nearing 10^{-3}. Notably, the model exhibits no signs of overfitting, maintaining generalization across varying datasets. The forward prediction results for the strain fields in the y direction, as illustrated in Figure S2(a), demonstrate similar trends to those observed in the x direction. As the training process progresses, with the number of epochs increasing, both the training loss and validation loss exhibit a significant exponential decrease, ultimately converging to a minimal value.

To rigorously assess the model's predictive accuracy, we employed it to generate strain field predictions for 10,000 new datasets and compared these predictions against actual simulation results. Fig. 3(f) demonstrates that the majority of the predicted strain values align closely with the true values, clustering around the $y = x$ line. The concentration of points around 0.125 mm in the red area indicates the distribution of strain when the sample is uniformly stretched by 5 mm, translating to 0.125 mm per voxel in a 40-voxel sample. This high degree of correlation is quantitatively supported by an R-squared value of 0.9927, underscoring the model's precision and reliability in forward prediction tasks. Figure S2(b) provides a comparison between the machine learning-based predictions for the strain fields in the y direction and the corresponding simulation results obtained through finite element analysis. The scatter plot illustrates that the predicted strain values closely align with the simulation results, as evidenced by the majority of points clustering around the line defined by the function $y = x$ with the R-squared value of 0.9988. This alignment confirms the machine learning model's capability to predict strain fields across all directions with high accuracy. We can also notice that the red points (high density) are basically distributed near 0 mm, indicating that the strain fields in the y direction are minimal when the part is stretched in the x direction. That is why only the results in the x direction are analyzed in the main text.

2.4.4. Parameter selection criteria

To refine the performance of our forward prediction model, we systematically explored the influence of various machine learning parameters on model accuracy and efficiency. The key parameters considered were the number of training data sets, batch size, learning rate and decay, hidden size, and the ratio of validation data.

The number of training data sets plays a crucial role in reducing model loss. As depicted in Fig. 3(g), when the dataset size is below 100, the model exhibits high loss values and minimal improvement with additional data, indicating insufficient training samples for effective learning. However, once the dataset size exceeds 1000, we observe a sharp reduction in loss, signifying that the model begins to capture meaningful patterns in the data. As the dataset size further increases to 5000 and 10,000, the loss continues to decline, but at a slower rate. Beyond 20,000 samples, the loss reaches an asymptotic value, suggesting that additional data yields diminishing returns in performance improvement.

Batch size also significantly impacts the model's performance and computational efficiency. Fig. 3(h) illustrates that increasing the batch size reduces the loss but at the expense of longer training times. This trade-off necessitates a balanced approach, leading us to select a batch size of 50. This choice strikes an optimal balance, ensuring sufficient gradient updates per epoch while maintaining reasonable training durations.

The learnning rate and its decay are critical for the convergence of the model. Fig. 3(i) shows that without learning rate decay, both training and validation losses exhibit high volatility, indicating instability and poor convergence. Applying a learning rate decay smoothens the loss curves and results in lower final losses, highlighting its role in stabilizing training. Moreover, Fig. 3(j) reveals that the loss initially decreases with increasing learning rates but starts to rise beyond a certain point. An optimal learning rate of approximately 0.025 was identified, ensuring rapid yet stable convergence.

The hidden size parameter, which dictates the number of neurons in hidden layers, influences the model's capacity and risk of overfitting. Fig. 3(k) shows that while increasing the hidden size initially reduces both training and validation losses, a hidden size above 100 leads to overfitting, evident from the widening gap between the two losses. Therefore, a hidden size of 50 was chosen to balance model complexity and generalization ability.

The ratio of validation data to total data is another vital factor. As shown in Fig. 3(l), an excessively high proportion of validation data can lead to overfitting, as the model may not have enough training samples to learn from. Conversely, too little validation data fails to provide an adequate assessment of model performance. We found that allocating 20% of the data for validation and 80% for training provided a robust evaluation framework, ensuring the model was both well-trained and appropriately validated.

By elaborately selecting these parameters, we have optimized our forward prediction model to achieve high accuracy and efficiency, ensuring robust performance across various hierarchical architecture scenarios.

2.5. Inverse optimization model

2.5.1. Evolutionary algorithm mechanism

In this paper, an evolutionary algorithm is developed to perform inverse optimization on strain fields with hierarchical architecture. Evolutionary algorithms, unlike gradient-based methods like topology optimization [93–96], leverage principles of natural selection and evolutionary processes to explore the solution space more comprehensively. This non-gradient-based approach significantly reduces the risk of getting trapped in local optima, a common challenge in high-dimensional and complex optimization problems.

Our evolutionary algorithm operates on a population of candidate solutions, each represented as an individual or chromosome. In this
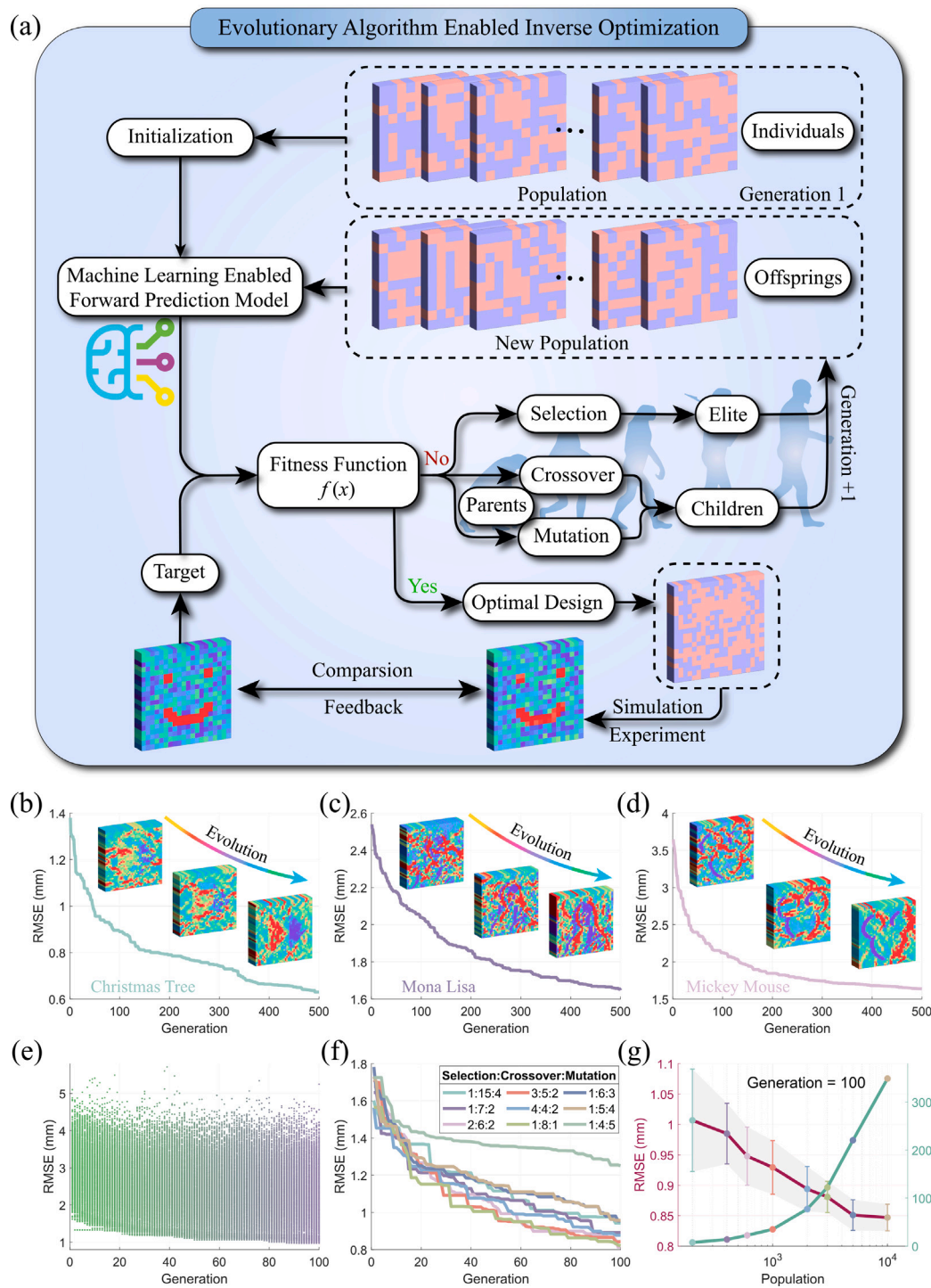
Fig. 4. Inverse optimization model. (a). The schematic diagram for the evolutionary algorithm from inverse optimization. (b-d). The fitness evolution when the generation increases for (b) Christmas tree, (c) Mona Lisa, and (d) Mickey Mouse cases. (e). Evolution of all individuals in the population during the inverse optimization. (f). The influence of the selection, crossover, and mutation ratio on optimization results. (g). The influence of number of population on the optimization performance and time cost.

context, individuals encode various configurations of hierarchical architectures, combining soft and hard materials in different arrangements. The initial population is generated randomly, ensuring a diverse set of potential solutions to start the optimization process.

The algorithm iteratively evolves this population through three main operations: selection, crossover, and mutation. During selection, individuals with higher fitness, measured by their performance in the given optimization criteria, are chosen to pass their genes to the next generation. Crossover, or recombination, mixes the genetic information

of selected individuals to create new offspring, promoting the exchange of beneficial traits. Mutation introduces random changes to individuals, enhancing diversity and enabling the exploration of new regions in the solution space.

2.5.2. Evolutionary algorithm design

The evolutionary algorithm we designed is illustrated in Fig. 4(a). The process begins with the random generation of 2000 individuals to form the initial population. Each individual is represented as a

40×40 matrix composed of 0s and 1s, corresponding to soft and hard materials, respectively. These individuals are then normalized using the same normalization method employed for the forward prediction model, ensuring consistency in data preprocessing.

Subsequently, the normalized population is input into our previously established machine learning-based forward prediction model to predict the strain fields for each individual configuration, and the machine learning output is subjected to a denormalization operation. These predicted strain fields are then compared to the target strain fields using a fitness function, which outputs a value analogous to error, thereby evaluating each individual's performance. The fitness function for the designed evolutionary algorithm is the root mean squared error.

The individual with the highest fitness score is identified as the best candidate. If this candidate meets the predefined performance criteria, the algorithm terminates, having successfully optimized the material allocation for the desired strain field. If the best candidate does not meet the performance requirements, the algorithm proceeds with selection, crossover, and mutation to generate a new population.

During the selection process, the population is sorted based on their fitness scores. The top 5% of individuals with the highest scores are designated as elite individuals. Additionally, 75% of the population is randomly selected for crossover operations. The crossover method is to randomly select a random number of columns from two parents and exchange them with each other, as illustrated in Figure S3. The material allocation after the exchange forms two children.

Mutation is performed on random-choose 20% of the population, where a random number of columns in selected individuals undergo random mutations as demonstrated in Figure S3. Each voxel in the chosen columns has a 50% probability of being assigned as either soft or hard material. The mutated individuals, along with the elite and offspring, constitute the new generation.

This iterative process of fitness evaluation, elite selection, crossover, and mutation continues until the algorithm reaches a satisfactory level of convergence or the required performance is achieved. The evolutionary algorithm's ability to explore the solution space comprehensively through these genetic operations ensures a robust search for the optimal hierarchical material configuration. This method not only leverages the principles of natural evolution but also incorporates advanced computational techniques to efficiently navigate complex optimization landscapes.

To refine the algorithm's efficiency and effectiveness, we incorporated several enhancements. The selection process includes a tournament selection mechanism, ensuring that only the most promising individuals contribute to the next generation. Crossover operations are customized to preserve the hierarchical structure of the material configurations, maintaining the integrity of the encoded solutions.

Moreover, to prevent redundant evaluations and enhance computational efficiency, a hash-table-based deduplication mechanism was employed [97]. This mechanism ensures that identical configurations are not re-evaluated, significantly reducing unnecessary computations and speeding up the optimization process.

Despite being evaluated separately, the modular implementation of our evolutionary algorithm allows seamless integration into the broader computational discovery pipeline. This flexibility facilitates easy switching between optimization algorithms and enhances the algorithm's applicability to a wide range of hierarchical architecture design problems.

2.5.3. Optimization performance

The performance of the inverse optimization model is comprehensively illustrated in Figs. 4(b), (c), and (d). These figures depict the evolution of three cases presented in Section 3.2. It is evident that as the number of generations increases, the errors for all three cases progressively decrease. The optimized best candidate becomes increasingly similar to the target over successive generations. Notably, the errors for the Mona Lisa and Mickey Mouse cases are larger than those

for the Christmas Tree case. This discrepancy arises due to specific requirements that must be met when setting the target: (1) the sum of strain fields in each row should be the same, that is the total displacement, and (2) the displacement fields of cell $c_{i,j}$ should not exceed those of $c_{i+1,j+1}$, $c_{i+1,j}$, and $c_{i+1,j-1}$. The first number after c represents the column in which the cell is located, and the second number after c represents the row in which the cell is located. While the Christmas Tree design strictly adheres to these criteria, the Mona Lisa and Mickey Mouse designs do not fully comply, leading to patterns that cannot entirely match the target and causing some cells in these two patterns to never reach the target. Consequently, these designs exhibit larger errors during evolution. However, despite these larger errors, the resulting designs remain close to the target. In addition, we plotted the fitness of all individuals over the first 100 generations of the Christmas Tree pattern's evolution algorithm optimization in Fig. 4(e). This plot reveals that with increasing generations, the performance of most individuals within the population improves. A small number of mutant and crossover individuals exhibit poor performance, which, although suboptimal, contributes to the overall development of the population by introducing diversity.

Fig. 4(f) examines the impact of different proportions of selection, crossover, and mutation on the population's evolution. The proportions of selection : crossover : mutation = 5% : 75% : 20% were found to deliver near-optimal performance. We also observed that crossover plays a more significant role in group evolution than mutation, as higher crossover proportions accelerate evolutionary progress.

Moreover, Fig. 4(g) demonstrates the effect of population size on optimization results, specifically for the Christmas Tree case. For a constant number of generations (100), smaller population sizes result in larger errors and higher standard deviations, which is a bad performance. However, increasing the population size also extends optimization time, indicating a trade-off between accuracy and computational efficiency. The optimization time correlates with the number of individuals undergoing forward prediction, necessitating a balance between population size and time constraints.

These findings highlight the robustness and effectiveness of our evolutionary algorithm in achieving optimal hierarchical material configurations. The iterative refinement process, driven by natural selection principles, enables comprehensive exploration and ensures that the final solutions are both robust and optimal, efficiently navigating complex optimization landscapes.

2.6. Physical validation

After fabricating the specimens, we utilized digital image correlation to measure the strain field of the hierarchical architecture with optimized material allocation during stretching [98–101]. This technique allowed for the capture of high-resolution strain maps, providing a detailed understanding of the mechanical behavior under load.

First, the printed specimens were preprocessed to ensure accurate strain measurement . The surface of each specimen was sprayed with matte black and white paint (Sanhe Automotive Paint from Guangzhou Sanhe Coating Technology Co., Ltd., Guangzhou, China) alternately at a 60 : 40 ratio, creating a random speckle pattern essential for DIC. The speckle pattern is illustrated in Figure S5(a) in the Supplementary Materials. The speckle pattern has an average speckle size of 5.2 pixels, as quantified in Supplementary Figure S5(b). This size falls near the optimal range of 3–5 pixels, minimizing DIC measurement errors while preserving spatial resolution [102]. This speckle pattern served as a reference grid, enabling the software to track displacements and deformations during the testing process.

The prepared specimens were then clamped securely at both ends and mounted in a tensile testing machine (PURI Materials, Shenzhen, China). The specimens were stretched at a controlled speed of 0.1 mm/s, and a high-resolution camera (Basler acA4112-30um) recorded the deformation process. The camera was preset with a

resolution of 3400 × 2110 pixels, a frame rate of 10 frames per second, a shutter speed of 1/1000 s, and an aperture of f/5.6. Uniform LED illumination minimized shadows and reflections. The camera continuously recorded images until the specimens were elongated by 5 mm, resulting in a total of 500 images captured during the experiment.

The collected images were then processed to obtain the strain field of the hierarchical architecture. The images were imported into the DIC software (VIC-2D, Correlated Solutions), where initial preprocessing ensured correct alignment, compensating for minor shifts or rotations during the experiment. The software analyzed the speckle pattern on each image, identifying and tracking the movement of the speckles between successive frames. This tracking allowed for the calculation of displacement fields across the specimen surface.

Specific parameters for DIC analysis were set to optimize accuracy and efficiency. The subset size was configured to 80 × 80 pixels, and the step size was set to 8 pixels, balancing resolution and processing speed. The correlation criterion used was the zero-normalized sum of squared differences (ZNSSD), chosen for its robustness and resistance to noise.

Using the displacement fields obtained from speckle tracking, the software computed the strain fields through the finite difference method, deriving local strain values from the displacement gradients. To enhance the clarity of the strain maps, noise reduction techniques such as Gaussian filtering were applied, eliminating high-frequency noise and ensuring smooth strain contours. The final strain fields were visualized as contour plots, clearly depicting the strain distribution across the specimen.

3. Results and discussion

3.1. Forward prediction results

3.1.1. Model development and training

In our research, we developed an RNN-based forward prediction model to forward estimate the strain fields of hierarchical architectures. The studied architecture consists of 40 × 40 voxels, resulting in a design space of 2^{1600} configurations, a remarkably vast and complex configuration. Each voxel has the size of 400 μm by 400 μm, resulting in a total structure size of 16 mm by 16 mm. To evaluate model performance, we examined the mechanical behavior of these hierarchical architectures under tensile loading, as illustrated in Fig. 5(a), with one side fixed and a load applied on the opposite side along the x direction, or horizontal axis. And the final displacement is set at 5 mm. For accurate modeling and simulation, we aimed to keep most voxels deformations within the elastic range of the materials, ensuring linear behavior as shown in Fig. 5(b).

The RNN model we build is specifically designed to handle the complexity of predicting strain fields in hierarchical architectures. RNNs are well-suited for sequence data and can capture temporal dependencies, which, in our case, translates to understanding the spatial relationships between voxels in the material grid. The input of the forward prediction model is the distribution of soft and hard materials, as shown in Fig. 5(a). The red voxel represents the soft material, and the blue voxel represents the hard material. In machine learning, we use 0 to represent hard materials and 1 to represent soft materials, so the input of machine learning is a 40 by 40 array consisting of binary values–0 and 1. The output is the amount of elongation of each voxel. Here, we mainly discuss the strain in the x direction, which is the stretching direction, because, in the stretching scenario, the deformation mainly occurs in the stretching direction. We will also show the results in the y direction, which is perpendicular to the stretching direction, in the supplementary materials. Therefore, the output is the 40 by 40 elongation value, each value represents the elongated length of the corresponding voxel. The RNN model was trained with hyperparameters, including a learning rate of 0.025, which was reduced to $1/\sqrt{2}$ of its original value every 25 epochs, and the batch size is set to 50. We used finite element analysis to generate 20,000 data (input and output) in eight hours to train our forward prediction model.

3.1.2. Evaluation of model performance

The results of our RNN model training are shown in Fig. 5(c), where we present four random cases. The first column displays randomly generated material allocations. These material allocations are then input into finite element analysis to obtain the simulation results shown in the second column. The same material allocations are input into an RNN-based machine learning model to predict the strain fields, with the results shown in the third column. The comparison indicates that our machine learning model predictions are highly consistent with the finite element analysis simulations, demonstrating the model's accuracy and performance. Moreover, our machine learning based forward prediction model reduces the forward prediction time from the simulation time of finite element analysis of two or three minutes to less than 0.01 s. This efficiency lays a strong foundation for subsequent inverse optimization.

Figure S4 presents the comparison for forward prediction results obtained by finite element analysis and proposed machine learning methods to predict the strain fields along the y direction. The material distributions for these cases are identical to those utilized in the x direction. The model's ability to accurately predict strain fields in the y direction is evident, as the results show a remarkable consistency with those in the x direction. The machine learning-based forward predictions for the y direction strain fields exhibit the same high level of accuracy and reliability, perfectly aligning with the results obtained through finite element analysis.

This consistency highlights the robustness and versatility of the machine learning model in handling complex hierarchical architectures. By applying the same training methodology across different directional strains, the model demonstrates its capability to generalize well, ensuring that the predictions remain precise irrespective of the strain direction.

3.2. Inverse optimization results

In this paper, we employ an inverse optimization approach utilizing evolutionary algorithms. Traditional gradient-based optimization techniques, like gradient descent, become computationally intensive when optimizing complex, high-dimensional loss functions with numerous model parameters (such as neural network weights) due to the need for repeated derivative evaluations of the loss function. In contrast, evolutionary algorithms, which do not require derivative evaluations, are more efficient and better suited for these scenarios.

Once our forward prediction model, based on machine learning, is established, it provides a highly efficient foundation for our inverse optimization process. The evolutionary algorithm we employ requires performing a vast number (typically over 100,000) of forward predictions to evaluate the performance of each individual. This evaluation is crucial for executing the selection, crossover, and mutation steps inherent in evolutionary algorithms. If we were to use finite element analysis to simulate the behavior of the same design, each simulation would take approximately two minutes. Consequently, 100,000 forward predictions would demand around 200,000 min, or more than 3000 h. However, our machine learning model significantly reduces the time required for forward predictions to about 0.01 s per prediction, bringing the total time for 100,000 predictions down to around 16 min, with only a marginal accuracy trade-off (i.e., more than 99% R-squared accuracy vs. FEM).

The results of our inverse optimization using the evolutionary algorithm are illustrated in Fig. 6. We examined three cases: Christmas tree, Mona Lisa, and Mickey Mouse. The first row displays the target strain fields we aim to achieve in the samples after stretching. The second row shows the optimized material allocations designed to produce these target strain fields. The third row presents the simulation results obtained from finite element analysis of these optimized allocations. The close resemblance between the simulation results and the targets
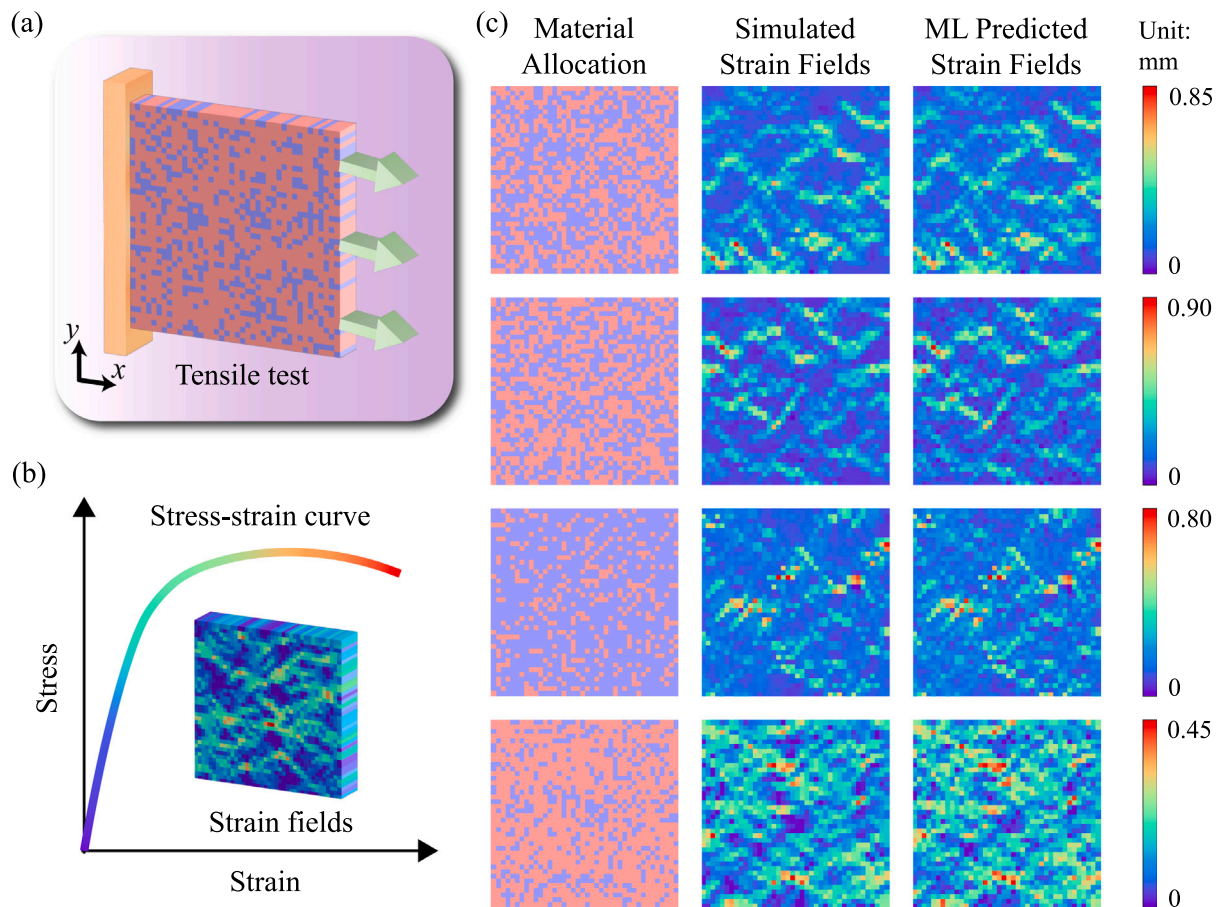
Fig. 5. Boundary conditions and comparison between finite element ground truth and machine learning predicted result. (a). Boundary conditions of tensile test. (b). Demonstration diagram for the stress–strain curve and the strain field. (c). Comparison of the strain field results (in *x* direction) represented by elongation from the finite element analysis and the machine learning prediction with the same material allocation.

demonstrates the effectiveness of our optimization process. Minor discrepancies in the blank areas, which do not form part of the patterns, are inevitable due to the interweaving of soft and hard materials.

In addition, we also fabricated samples of the optimized material allocations using digital light processing. These printed samples are shown in the fourth row, with their dimensions specified in Section 3.1.1. The fifth row features the strain fields of the experimental results, obtained by conducting tensile tests on the printed samples and detecting the strain fields using digital image correlation. The contours of the experimental results closely match those of the simulation results. However, some discrepancies are reasonable because of the inherent errors in additive manufacturing, the contamination challenges of multi-material printing, and the small scale of the problem under study. These results validate the accuracy and efficiency of our inverse optimization approach using evolutionary algorithms.

3.3. Discussion

3.3.1. Observation

The results of this study highlight the potential of machine learning and evolutionary algorithms in advancing the design and optimization of hierarchical architectures. Beyond the high accuracy of our RNN-based forward prediction model, a significant finding is the model's ability to generalize across different directional strain fields. This generalization is crucial for developing versatile predictive tools that can be applied to a wide range of material configurations and loading conditions, which is particularly important as the complexity of the hierarchical structures increases.

Moreover, the application of evolutionary algorithms for inverse optimization demonstrates a promising approach to solving high-dimensional design problems. Traditional optimization methods, such as gradient descent, often struggle with the computational demands of large-scale problems, particularly those involving non-linear material behaviors or complex boundary conditions. Evolutionary algorithms, however, offer a robust alternative by effectively exploring the design space through selection, crossover, and mutation processes. This capability allows for the identification of optimal material distributions that achieve desired strain fields with greater efficiency and less computational overhead.

Additionally, the research opens up new avenues for exploring the impact of different material properties and geometries on the performance of hierarchical architectures. Future studies could investigate the integration of more complex material models, such as those accounting for viscoelasticity or plasticity, to further enhance the predictive capabilities of the machine learning models. Moreover, expanding the scope of the optimization to include dynamic loading conditions or multi-objective criteria could lead to the development of even more advanced and resilient material systems.

3.3.2. Limitation

Although this study demonstrates the efficacy of machine learning and evolutionary algorithms for strain field design in hierarchical architectures, several limitations exist.

First, the forward prediction model relies on FEA-generated training data, which inherently assumes idealized material behavior (e.g., Neo-Hookean elasticity) and boundary conditions. Discrepancies between simulation assumptions and real-world material nonlinearities, such
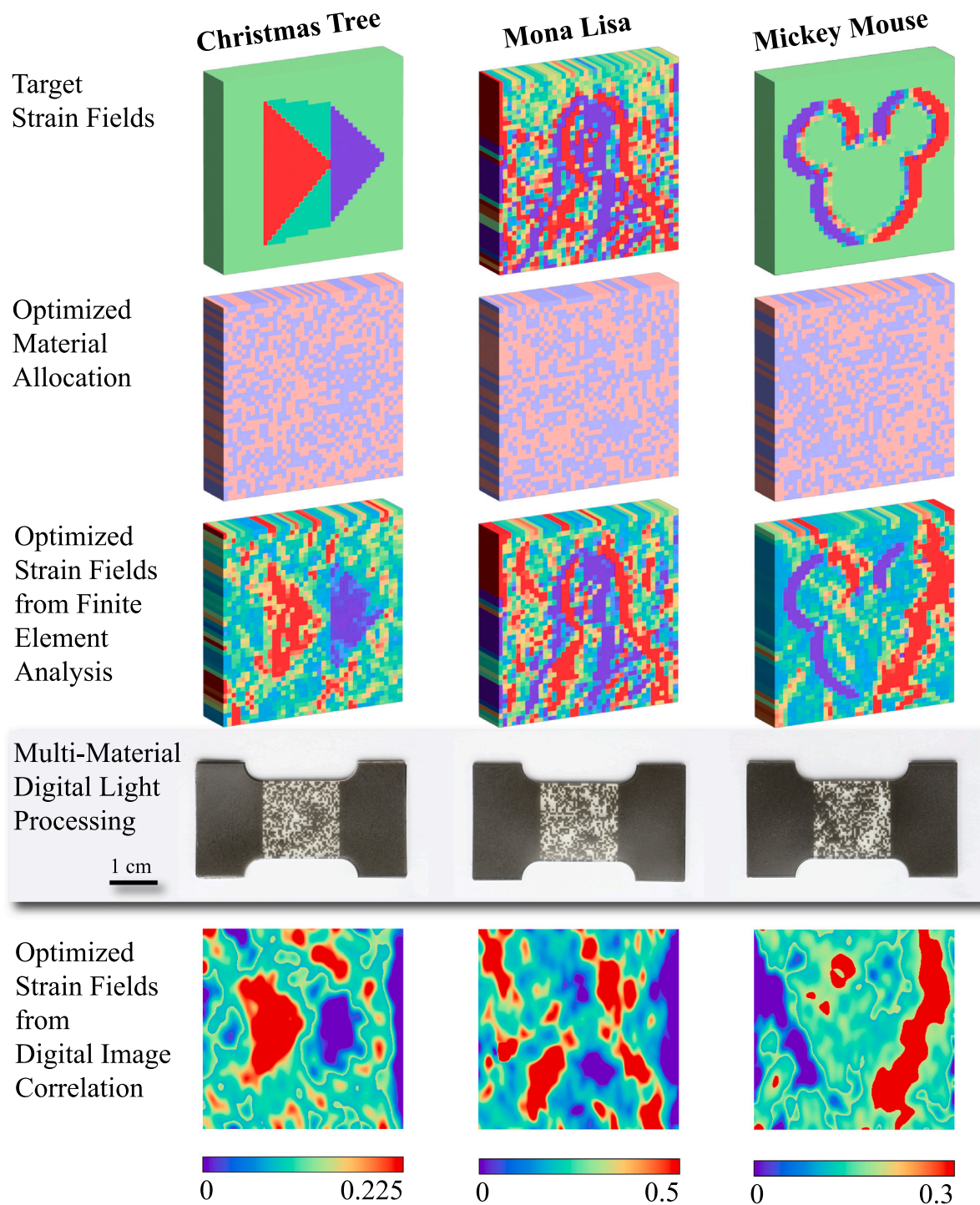
Fig. 6. Inverse optimization performance and case studies.

as viscoelasticity or plasticity, may introduce biases in the model's predictions. Additionally, the dataset, though extensive, may not fully represent the entire design space of 2^{1600} configurations, potentially limiting generalization to highly irregular or unseen material distributions. Normalization procedures applied to input/output data could further amplify biases if extreme strain values or rare configurations are underrepresented in the training set.

Second, the experimental tests conducted in this study aim to characterize the mechanical response of hierarchical architectures, providing valuable insights into their material behavior and validating the optimized designs. However, directly correlating experimental results

with machine learning predictions poses challenges due to fabrication-induced variations, material inconsistencies, and the inherent complexity of the response. In practical manufacturing processes, factors such as layer adhesion, multi-material contamination, and microstructural heterogeneities introduce discrepancies between experimental measurements and computational predictions, making it difficult to establish a precise one-to-one numerical comparison. Additionally, the machine learning model is trained under idealized conditions, predicting strain fields based on predefined input parameters, whereas real-world experimental settings involve additional complexities such as boundary effects, localized stress concentrations, and variations in loading conditions. Furthermore, the DIC technique assumes planar deformation

under quasi-static loading, neglecting out-of-plane displacements and dynamic effects. The small-scale specimens (16 mm × 16 mm) may also exhibit size-dependent behaviors not captured in bulk material characterization, potentially limiting extrapolation to macroscale applications. Despite these challenges, qualitative validation remains a viable approach, as the experimental strain fields exhibit strong agreement with the deformation patterns predicted by the model, reinforcing the reliability of our framework. Nevertheless, the observed discrepancies highlight the challenges of translating computational designs into physical prototypes, particularly in additive manufacturing. Addressing these issues in future research, such as through high-resolution digital image correlation methods for precise strain measurements and improved alignment of computational simulations with experimental boundary conditions, will enhance model validation and improve the fidelity of printed architectures to their designed counterparts. By tackling these aspects, this study contributes to bridging the gap between data-driven predictions and experimental verification in hierarchical material design.

Furthermore, the computational framework is currently optimized for 40 × 40 voxel grids. Scaling to larger architectures would exponentially increase the design space, raising computational costs for both FEA data generation and evolutionary optimization. While the ML model reduces prediction time, training on larger grids would require architectural modifications (e.g., memory-augmented networks) to handle increased dimensionality. Manufacturing scalability is similarly constrained by the resolution limits of DLP printing (400 μm voxels) and the difficulty of maintaining interlayer adhesion in multi-material systems at larger scales. Additionally, the evolutionary algorithm's population size and mutation rates may require recalibration for high-dimensional problems to avoid premature convergence or excessive computation time.

3.3.3. Perspective

Although the current study demonstrates the effectiveness of machine learning-driven design for hierarchical architectures, several avenues remain open for further exploration. Below, we outline key directions for future work.

First, although the current RNN-based model achieves high accuracy, future studies could investigate alternative architectures such as transformers, graph neural networks (GNNs), or physics-informed neural networks (PINNs). Transformers may better capture long-range spatial dependencies in hierarchical architectures, while GNNs could explicitly model voxel connectivity. PINNs, which embed governing physical laws (e.g., equilibrium equations) directly into the loss function, will reduce reliance on large FEA datasets and improve extrapolation to unseen boundary conditions.

Besides, although our current implementation considers a fixed set of loading conditions, the deep learning model fundamentally possesses the capacity to generalize across varying loading scenarios. This generalization capability stems from the neural network's ability to map material property distributions to strain field responses within high-dimensional design spaces.

Moreover, this study focuses on binary material systems (soft/hard). Expanding to ternary or functionally graded materials would enable richer property tailoring but requires redefining the design space and training datasets. Incorporating nonlinear material behaviors — such as viscoelasticity, plasticity, or strain-rate dependence — would enhance the model's applicability to dynamic loading scenarios. Experimental validation could further explore anisotropic or temperature-sensitive resins to broaden practical relevance.

Finally, although this work focuses on 2D architectures for computational tractability, the methodology is extensible to 3D. 3D neural networks (such as the Conv3D-based model) can replace the RNN for forward prediction of 3D strain fields, and the evolutionary algorithm can optimize 3D voxel distributions. However, computational costs grow exponentially in 3D—e.g., a 100 × 100 × 100 architecture involves 10^6 design variables versus 10^4 in 2D. Future work will integrate parallel computing and adaptive sampling to mitigate these challenges. The multi-material DLP technique used here has been validated for 3D printing [80], suggesting strong potential for hierarchical 3D design with optimized computational frameworks.

The ability to precisely design and control strain fields in voxel heterogeneous microstructures opens up various advanced engineering and biomedical applications. Below, we highlight several promising directions:

- Anti-Counterfeiting Tags: Microstructures with hidden strain-activated patterns (e.g., logos, QR codes) that become visible only under specific tensile loads, as demonstrated in Fig. 6. Such features are nearly impossible to replicate without knowledge of the encoded material distribution.
- Adaptive Optical Devices: Stretch-tunable lenses or diffraction gratings where strain-induced deformations modulate light transmission or focal length. For example, a Fresnel lens pattern could emerge when stretched, enabling dynamic optical systems.
- Bio-Inspired Sensors: Mimicking mechanoreceptors in skin, hierarchical architectures could map strain fields to electrical signals for high-resolution tactile sensing in robotics or prosthetics.
- Programmable Metamaterials: Structures designed to exhibit negative Poisson's ratios or strain-dependent acoustic bandgaps for vibration isolation, energy harvesting, or acoustic cloaking.
- Smart Biomedical Implants: Microstructures that generate strain gradients promoting controlled tissue regeneration or drug release in response to physiological loads.

4. Conclusions

This paper presents a comprehensive approach to the design of hierarchical architectures using advanced machine learning techniques, specifically focusing on strain fields. We developed a recurrent neural network-based forward prediction model that demonstrated exceptional accuracy in predicting strain fields of hierarchical materials, achieving over 99% accuracy in significantly reduced computation times. This model serves as a foundational tool for inverse optimization, facilitating the efficient exploration of vast design spaces inherent in hierarchical systems.

Our inverse optimization framework employs evolutionary algorithms to determine optimal material distributions, aligning strain fields with desired performance targets. The effectiveness of this approach was validated through finite element analysis and experimental results, showcasing the ability to accurately replicate target strain fields in complex patterns such as the Christmas tree, Mona Lisa, and Mickey Mouse.

The integration of efficient forward prediction and precise inverse optimization offers a robust methodology for the design of hierarchical architectures. By drastically reducing computation times and improving prediction accuracy, our approach accelerates the material design process, enabling the discovery of advanced materials with superior properties. This method holds significant potential for various applications.

The experimental validation, using digital light processing and digital image correlation, further substantiates the practicality of our approach. Despite minor discrepancies due to manufacturing limitations, the strong correlation between simulation and experimental results underscores the reliability of our models.

CRediT authorship contribution statement

Liuchao Jin: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shouyi Yu:** Writing – original

draft, Visualization, Validation, Formal analysis, Data curation. **Jianxiang Cheng:** Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Formal analysis. **Zhigang Liu:** Writing – original draft, Investigation, Formal analysis, Data curation. **Kang Zhang:** Writing – original draft, Validation, Formal analysis, Data curation. **Sicong Zhou:** Writing – original draft, Methodology, Investigation. **Xiangnan He:** Validation, Formal analysis. **Guoquan Xie:** Writing – original draft, Visualization, Formal analysis. **Mahdi Bodaghi:** Writing – review & editing, Investigation, Formal analysis. **Qi Ge:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation. **Wei-Hsin Liao:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compositesb.2025.112372.

Data availability

Data will be made available on request.

References

[1] Tsang MY, Sinelshchikova A, Zaremba O, Schöfbeck F, Balsa AD, Reithofer MR, Wuttke S, Chin JM. Multilength scale hierarchy in metal–organic frameworks: Synthesis, characterization and the impact on applications. Adv Funct Mater 2023;2308376.

[2] Hiller J, Lipson H. Tunable digital material properties for 3D voxel printers. Rapid Prototyp J 2010;16(4):241–7.

[3] Chen C-T, Gu GX. Machine learning for composite materials. MRs Commun 2019;9(2):556–66.

[4] Jin L, Zhai X, Wang K, Zhang K, Wu D, Nazir A, Jiang J, Liao W-H. Big data, machine learning, and digital twin assisted additive manufacturing: A review. Mater & Des 2024;244:113086.

[5] Ahn S-J, Lee H, Cho K-J. 3D printing with a 3D printed digital material filament for programming functional gradients. Nat Commun 2024;15(1):3605.

[6] Zhai X, Jin L, Jiang J. A survey of additive manufacturing reviews. Mater Sci Addit Manuf 2022;1(4):21.

[7] Li X, Wang X, Mei D, Xu C, Wang Y. Acoustic-assisted DLP 3D printing process for carbon nanofiber reinforced honeycomb structures. J Manuf Process 2024;121:374–81.

[8] Jiang J, Zhai X, Zhang K, Jin L, Lu Q, Shen Z, Liao W-H. Low-melting-point alloys integrated extrusion additive manufacturing. Addit Manuf 2023;103633.

[9] Pei Z, Rozman KA, Doğan ÖN, Wen Y, Gao N, Holm EA, Hawk JA, Alman DE, Gao MC. Machine-learning microstructure for inverse material design. Adv Sci 2021;8(23):2101207.

[10] Jiang J, Zhai X, Jin L, Zhang K, Chen J, Lu Q, Liao W-H. Design for reversed additive manufacturing low-melting-point alloys. J Eng Des 2023;1–14.

[11] Liu Q, Hong L, Dong X, Zhai W. 3D printed hierarchical interpenetrating phase composites with multi-scale mechanical energy absorption mechanisms. Compos Part B: Eng 2023;264:110911.

[12] Patadiya J, Wang X, Joshi G, Kandasubramanian B, Naebe M. 3D-printed biomimetic hierarchical nacre architecture: Fracture behavior and analysis. ACS Omega 2023;8(21):18449–61.

[13] Choi GP. Computational design of art-inspired metamaterials. Nat Comput Sci 2024;4(8):549–52.

[14] Dutto A, Zanini M, Jeoffroy E, Tervoort E, Mhatre SA, Seibold ZB, Bechthold M, Studart AR. 3D printing of hierarchical porous ceramics for thermal insulation and evaporative cooling. Adv Mater Technol 2023;8(4):2201109.

[15] Gao F, Tong Z, Xiao W, Liu Q, Lu J, Hou Y, He Q, Gao X, Cheng D, Zhan X, Ma Y, Zhang Q. Structural engineering of hierarchical aerogels hybrid networks for efficient thermal comfort management and versatile protection. Small 2023;19(25):2301164.

[16] Xu D, Chen Z, Liu Y, Ge C, Jiao L, Guo W, Zhang Q, Fang J, Xu W. Hump-inspired hierarchical fabric for personal thermal protection and thermal comfort management. Adv Funct Mater 2023;33(10):2212626.

[17] Yue X, Zhang R, Jin X, Zhang X, Bao G, Qin D. Bamboo-derived phase change material with hierarchical structure for thermal energy storage of building. J Energy Storage 2023;62:106911.

[18] Tan C, Li Q, Yao X, Chen L, Su J, Ng FL, Liu Y, Yang T, Chew Y, Liu CT, et al. Machine learning customized novel material for energy-efficient 4D printing. Adv Sci 2023;10(10):2206607.

[19] Valorosi F, De Meo E, Blanco-Varela T, Martorana B, Veca A, Pugno N, Kinloch IA, Anagnostopoulos G, Galiotis C, Bertocchi F, et al. Graphene and related materials in hierarchical fiber composites: Production techniques and key industrial benefits. Compos Sci Technol 2020;185:107848.

[20] Asyraf M, Ilyas R, Sapuan S, Harussani M, Hariz H, Aiman J, Baitaba DM, Sanjay M, Ishak M, Norkhairunnisa M, et al. Advanced composite in aerospace applications: opportunities, challenges, and future perspective. Adv Compos Aerosp Eng Appl 2022;471–98.

[21] Ni X, Acauan L, Wardle BL. Coherent nanofiber array buckling-enabled synthesis of hierarchical layered composites with enhanced strength. Extrem Mech Lett 2020;39:100773.

[22] Zhang Z, Cai S, Li Y, Wang Z, Long Y, Yu T, Shen Y. High performances of plant fiber reinforced composites—a new insight from hierarchical microstructures. Compos Sci Technol 2020;194:108151.

[23] Xin X, Wang Z, Zeng C, Lin C, Liu L, Liu Y, Leng J. 4D printing bio-inspired chiral metamaterials for flexible sensors. Compos Part B: Eng 2024;286:111761.

[24] Li W, Pei R, Xu F, et al. Composite metamaterial antenna with super mechanical and electromagnetic performances integrated by three-dimensional weaving technique. Compos Part B: Eng 2024;273:111265.

[25] Jia Y, Wu X, Ren B, Ti J, Deng Y, Wang Q, Li H. Electromagnetic wave absorption of polymer derived ceramic composites tuned by multi-component oxide solid solution. Compos Part B: Eng 2024;278:111431.

[26] Wei J, Pan F, Ping H, Yang K, Wang Y, Wang Q, Fu Z. Bioinspired additive manufacturing of hierarchical materials: From biostructures to functions. Research 2023;6:0164.

[27] Nepal D, Kang S, Adstedt KM, Kanhaiya K, Bockstaller MR, Brinson LC, Buehler MJ, Coveney PV, Dayal K, El-Awady JA, et al. Hierarchically structured bioinspired nanocomposites. Nat Mater 2023;22(1):18–35.

[28] Li W, Jiang L, Wu S, Yang S, Ren L, Cheng B, Xia J. A shape-programmable hierarchical fibrous membrane composite system to promote wound healing in diabetic patients. Small 2022;18(11):2107544.

[29] Dudte LH, Choi GP, Becker KP, Mahadevan L. An additive framework for kirigami design. Nat Comput Sci 2023;3(5):443–54.

[30] Corigliano A. Discovering kirigami patterns. Nat Comput Sci 2023;3(5):368–9.

[31] He Q, Ferracin S, Raney JR. Programmable responsive metamaterials for mechanical computing and robotics. Nat Comput Sci 2024;4(8):567–73.

[32] Hu H. Recent advances of polymeric phase change composites for flexible electronics and thermal energy storage system. Compos Part B: Eng 2020;195:108094.

[33] Xu Z, Deng W, Wang X. 3D hierarchical carbon-rich micro-/nanomaterials for energy storage and catalysis. Electrochem Energy Rev 2021;4:269–335.

[34] Tao K, Chen Z, Yi H, Zhang R, Shen Q, Wu J, Tang L, Fan K, Fu Y, Miao J, et al. Hierarchical honeycomb-structured electret/triboelectric nanogenerator for biomechanical and morphing wing energy harvesting. Nano-Micro Lett 2021;13(1):123.

[35] Pei H, Chen Y, Lv Q, Peng Z, Wang X, Chen N, Zhang H. A novel microwave assisted multi-material 3D printing strategy to architect lamellar piezoelectric generators for intelligent sensing. Compos Part B: Eng 2024;280:111529.

[36] Ilyas R, Sapuan S, Nurazzi NM, Norrrahim MNF, Ibrahim R, Atikah M, Huzaifah M, Radzi A, Izwan S, Azammi AN, et al. Macro to nanoscale natural fiber composites for automotive components: Research, development, and application. Biocomposite Synth Compos Automot Appl 2021;51–105.

[37] Salifu S, Desai D, Ogunbiyi O, Mwale K. Recent development in the additive manufacturing of polymer-based composites for automotive structures—a review. Int J Adv Manuf Technol 2022;119(11):6877–91.

[38] Zhang H, Peijs T. Multifunctional composites based on hierarchical micro-nano structures: design, manufacturing, properties, and applications. In: Fiber-reinforced nanocomposites: fundamentals and applications. Elsevier; 2020, p. 183–98.

[39] Meng Z, Yan H, Liu M, Qin W, Genin GM, Chen CQ. Encoding and storage of information in mechanical metamaterials. Adv Sci 2023;10(20):2301581.

[40] Wu L, Lu Y, Li P, Wang Y, Xue J, Tian X, Ge S, Li X, Zhai Z, Lu J, et al. Mechanical metamaterials for handwritten digits recognition. Adv Sci 2024;11(10):2308137.

[41] Deshpande V, Myers O, Li S. Large-ratio stiffness switching via harnessing the in-plane buckling and bi-stability of high-load capacity composite laminates. Compos Part B: Eng 2024;279:111440.

[42] Pereira JLJ, Oliver GA, Francisco MB, Cunha Jr. SS, Gomes GF. A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. Arch Comput Methods Eng 2022;29(4):2285–308.

[43] Zheng Y, Wang Y, Lu X, Liao Z, Qu J. Evolutionary topology optimization for mechanical metamaterials with auxetic property. Int J Mech Sci 2020;179:105638.

[44] Lu Y, Kong D, Yang G, Wang R, Pang G, Luo H, Yang H, Xu K. Machine learning-enabled tactile sensor design for dynamic touch decoding. Adv Sci 2023;10(32):2303949.

[45] Lu B, Xia Y, Ren Y, Xie M, Zhou L, Vinai G, Morton SA, Wee AT, van der Wiel WG, Zhang W, et al. When machine learning meets 2D materials: A review. Adv Sci 2024;11(13):2305277.

[46] Wei L, Li Q, Song Y, Stefanov S, Dong R, Fu N, Siriwardane EM, Chen F, Hu J. Crystal composition transformer: Self-learning neural language model for generative and tinkering design of materials. Adv Sci 2024;2304305.

[47] Sun X, Yu L, Yue L, Zhou K, Demoly F, Zhao RR, Qi HJ. Machine learning and sequential subdomain optimization for ultrafast inverse design of 4D-printed active composite structures. J Mech Phys Solids 2024;105561.

[48] Sun X, Zhou K, Demoly F, Zhao RR, Qi HJ. Perspective: Machine learning in design for 3D/4D printing. J Appl Mech 2024;91(3):030801.

[49] 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. Adv Sci 2024;2407825.

[50] Ren S, Zhang Z, He R, Fan J, Wang G, Wang H, Han B, Zhang Y-L, Ma Z-C. Autonomous inverse encoding guides 4D nanoprinting for highly programmable shape morphing. Int J Extrem Manuf 2025;7(3):035502.

[51] Athinarayanarao D, Prod'hon R, Chamoret D, Qi HJ, Bodaghi M, André J-C, Demoly F. Computational design for 4D printing of topology optimized multi-material active composites. Npj Comput Mater 2023;9(1):1.

[52] Hamel CM, Roach DJ, Long KN, Demoly F, Dunn ML, Qi HJ. Machine-learning based design of active composite structures for 4D printing. Smart Mater Struct 2019;28(6):065005.

[53] Li H, Tang Z, Gong X, Zou N, Duan W, Xu Y. Deep-learning electronic-structure calculation of magnetic superstructures. Nat Comput Sci 2023;3(4):321–7.

[54] Ballard Z, Brown C, Madni AM, Ozcan A. Machine learning and computation-enabled intelligent sensor design. Nat Mach Intell 2021;3(7):556–65.

[55] Liu Z, Cai M, Hong S, Shi J, Xie S, Liu C, Du H, Morin JD, Li G, Wang L, et al. Data-driven inverse design of flexible pressure sensors. Proc Natl Acad Sci 2024;121(28):e2320222121.

[56] Bonfanti S, Guerra R, Font-Clos F, Rayneau-Kirkhope D, Zapperi S. Automatic design of mechanical metamaterial actuators. Nat Commun 2020;11(1):4162.

[57] Jin L, Zhai X, Zhang K, Jiang J, Liao W-H. 3D printing soft robots integrated with low-melting-point alloys. Mater Sci Addit Manuf 2024;3(3):4144.

[58] Lee S, Zhang Z, Gu GX. Generative machine learning algorithm for lattice structures with superior mechanical properties. Mater Horizons 2022;9(3):952–60.

[59] Dai T, Jin L, Shang C, Zhai X, Fu X-M, Liu L. Advances in intelligent design of metamaterials. J Computer- Aided Des Comput Graph 2025.

[60] Lee S, Choi W, Park JW, Kim D-S, Nahm S, Jeon W, Gu GX, Kim M, Ryu S. Machine learning-enabled development of high performance gradient-index phononic crystals for energy focusing and harvesting. Nano Energy 2022;103:107846.

[61] Chen K, Fang S, Yan T, Chuai X, Li X, Lai Z, Liang J, Liao W-H. Design and optimization of an auxetic piezoelectric energy harvester with tapered thickness for IoT applications. IEEE Internet Things J 2024.

[62] Bonfanti S, Hiemer S, Zulkarnain R, Guerra R, Zaiser M, Zapperi S. Computational design of mechanical metamaterials. Nat Comput Sci 2024;4(8):574–83.

[63] Gu GX, Chen C-T, Richmond DJ, Buehler MJ. Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment. Mater Horizons 2018;5(5):939–45.

[64] Gu GX, Chen C-T, Buehler MJ. De novo composite design based on machine learning algorithm. Extrem Mech Lett 2018;18:19–28.

[65] Chen C-T, Gu GX. Generative deep neural networks for inverse materials design using backpropagation and active learning. Adv Sci 2020;7(5):1902607.

[66] Kim Y, Kim Y, Yang C, Park K, Gu GX, Ryu S. Deep learning framework for material design space exploration using active transfer learning and data augmentation. Npj Comput Mater 2021;7(1):140.

[67] Qian C, Tan RK, Ye W. Design of architectured composite materials with an efficient, adaptive artificial neural network-based generative design method. Acta Mater 2022;225:117548.

[68] Sui F, Guo R, Zhang Z, Gu GX, Lin L. Deep reinforcement learning for digital materials design. ACS Mater Lett 2021;3(10):1433–9.

[69] Yang C, Kim Y, Ryu S, Gu GX. Prediction of composite microstructure stress–strain curves using convolutional neural networks. Mater Des 2020;189:108509.

[70] Li B, Deng B, Shou W, Oh T-H, Hu Y, Luo Y, Shi L, Matusik W. Computational discovery of microstructured composites with optimal stiffness-toughness trade-offs. Sci Adv 2024;10(5):eadk4284.

[71] Zhang Z, Gu GX. Physics-informed deep learning for digital materials. Theor Appl Mech Lett 2021;11(1):100220.

[72] Zhao B, Zhang M, Dong L, Wang D. Design of grayscale digital light processing 3D printing block by machine learning and evolutionary algorithm. Compos Commun 2022;36:101395.

[73] Jin L, Yu S, Cheng J, Ye H, Zhai X, Jiang J, Zhang K, Jian B, Bodaghi M, Ge Q, Liao W-H. Machine learning-driven forward prediction and inverse design for 4D printed hierarchical architecture with arbitrary shapes. Appl Mater Today 2024;40:102373.

[74] Jin L, Zhai X, Jiang J, Zhang K, 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. SPIE; 2024, p. 321–32.

[75] Sun X, Yue L, Yu L, Shao H, Peng X, Zhou K, Demoly F, Zhao R, Qi HJ. Machine learning-evolutionary algorithm enabled design for 4D-printed active composite structures. Adv Funct Mater 2022;32(10):2109805.

[76] Sun X, Yue L, Yu L, Forte CT, Armstrong CD, Zhou K, Demoly F, Zhao RR, Qi HJ. Machine learning-enabled forward prediction and inverse design of 4D-printed active plates. Nat Commun 2024;15(1):5509.

[77] Yang Z, Yu C-H, Buehler MJ. Deep learning model to predict complex stress and strain fields in hierarchical composites. Sci Adv 2021;7(15):eabd7416.

[78] Yang Z, Yu C-H, Guo K, Buehler MJ. End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures. J Mech Phys Solids 2021;154:104506.

[79] Rashid MM, Pittie T, Chakraborty S, Krishnan NA. Learning the stress–strain fields in digital composites using Fourier neural operator. Iscience 2022;25(11).

[80] Cheng J, Wang R, Sun Z, Liu Q, He X, Li H, Ye H, Yang X, Wei X, Li Z, et al. Centrifugal multimaterial 3D printing of multifunctional heterogeneous objects. Nat Commun 2022;13(1):7931.

[81] Rafiee M, Farahani RD, Therriault D. Multi-material 3D and 4D printing: a survey. Adv Sci 2020;7(12):1902307.

[82] Dominguez-Alfaro A, Mitoudi-Vagourdi E, Dimov I, Picchio ML, Lopez-Larrea N, de Lacalle JL, Tao X, Serrano RR-M, Gallastegui A, Vassardanis N, et al. Light-based 3D multi-material printing of micro-structured bio-shaped, conducting and dry adhesive electrodes for bioelectronics. Adv Sci 2024;2306424.

[83] Zhang M, Fan X, Dong L, Jiang C, Weeger O, Zhou K, Wang D. Voxel design of grayscale DLP 3D-printed soft robots. Adv Sci 2024;2309932.

[84] Zhou C, Chen Y, Yang Z, Khoshnevis B. Digital material fabrication using mask-image-projection-based stereolithography. Rapid Prototyp J 2013;19(3):153–65.

[85] Wang Q, Jackson JA, Ge Q, Hopkins JB, Spadaccini CM, Fang NX. Lightweight mechanical metamaterials with tunable negative thermal expansion. Phys Rev Lett 2016;117(17):175901.

[86] Chen D, Zheng X. Multi-material additive manufacturing of metamaterials with giant, tailorable negative Poisson's ratios. Sci Rep 2018;8(1):1–8.

[87] Han D, Yang C, Fang NX, Lee H. Rapid multi-material 3D printing with projection micro-stereolithography using dynamic fluidic control. Addit Manuf 2019;27:606–15.

[88] Kowsari K, Akbari S, Wang D, Fang NX, Ge Q. High-efficiency high-resolution multimaterial fabrication for digital light processing-based three-dimensional printing. 3D Print Addit Manuf 2018;5(3):185–93.

[89] Wang M, Li W, Mille LS, Ching T, Luo Z, Tang G, Garciamendez CE, Lesha A, Hashimoto M, Zhang YS. Digital light processing based bioprinting with composable gradients. Adv Mater 2022;34(1):2107038.

[90] Maqsood A, Chen C, Jacobsson TJ. The future of material scientists in an age of artificial intelligence. Adv Sci 2024;11(19):2401401.

[91] Zeng Q, Duan S, Zhao Z, Wang P, Lei H. Inverse design of energy-absorbing metamaterials by topology optimization. Adv Sci 2023;10(4):2204977.

[92] Liu H, Li L, Wei Z, Smedskjaer MM, Zheng XR, Bauchy M. De novo atomistic discovery of disordered mechanical metamaterials by machine learning. Adv Sci 2024;11(13):2304834.

[93] Zhai X, Gai Y, Jin L, Liao W-H, Chen F, Hu P. Isogeometric topology optimization of auxetic materials based on moving morphable components method. Adv Top Mech Mater Struct Construction: AToMech1-2023 2023;172.

[94] Zheng N, Zhai X, Chen F. Topology optimization of self-supporting porous structures based on triply periodic minimal surfaces. Computer- Aided Des 2023;161:103542.

[95] Zhai X, Wang W, Chen F, Wu J. Topology optimization of differentiable microstructures. Comput Methods Appl Mech Engrg 2024;418:116530.

[96] Zheng N, Zhai X, Jiang J, Chen F. Topology optimization of self-supporting structures for additive manufacturing via implicit b-spline representations. Computer- Aided Des 2024;103745.

[97] Ahmed ST, George LE. Lightweight hash-based de-duplication system using the self detection of most repeated patterns as chunks divisors. J King Saud University- Comput Inf Sci 2022;34(7):4669–78.

[98] Liu Z, Fan X, Liu K, Zhang F, Huang B, Wang Y, Li J. Visualizing multi-physical spatiotemporal evolution of dielectric breakdown in high-performance polymeric capacitors. Appl Phys Lett 2023;123(16).

[99] Yang R, Li Y, Zeng D, Guo P. Deep DIC: Deep learning-based digital image correlation for end-to-end displacement and strain measurement. J Mater Process Technol 2022;302:117474.

[100] Janeliukstis R, Chen X. Review of digital image correlation application to large-scale composite structure testing. Compos Struct 2021;271:114143.

[101] Mei Y, Deng J, Zhao D, Xiao C, Wang T, Dong L, Zhu X. Toward improved accuracy in quasi-static elastography using deep learning. CMES Comput Model Eng Sci 2024;139(1).

[102] Lionello G, Cristofolini L. A practical approach to optimizing the preparation of speckle patterns for digital-image correlation. Meas Sci Technol 2014;25(10):107001.