Proceedings of the ASME 2024
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE 2024
August 25-28, 2024, Washington, DC

IDETC2024-143200

A GENERATIVE GRAPH NEURAL NETWORK-BASED FRAMEWORK FOR DESIGNING CONNECTIVITY-GUARANTEED POROUS METAMATERIAL UNITS

Zihan Wang, Austin Bray, Kiarash Naghavi Khanghah, Hongyi Xu*

School of Mechanical, Aerospace, and Manufacturing Engineering
University of Connecticut, Storrs, CT 06269
* Email: hongyi.3.xu@uconn.edu

ABSTRACT

Porous metamaterial units filled with fluid have been used in engineering systems due to their ability to achieve desired properties (e.g., effective thermal conductvity). Designing 3D porous metamaterial units while ensuring complete connectivity of both solid and pore phases presents a significant challenge. In this study, we propose a generative graph neural network-based framework for designing the porous metamaterial units infilled with liquid. Firstly, we propose a graph-based metamaterial unit generation approach to generate porous metamaterial samples with complete connectivity in both solid and pore phases. Secondly, we establish and evaluate three distinct variational graph autoencoder (VGAE)-based generative models to assess their effectiveness in generating an accurate latent space representation of metamaterial structures. By choosing the model with the highest reconstruction accuracy, the property-driven design search is conducted to obtain novel metamaterial unit designs with the targeted properties. A case study on designing liquid-filled metamaterials for thermal conductivity properties is carried out. The effectiveness of the proposed graph neural network-based design approach is evaluated by comparing the performances of the obtained designs with those of existing designs in the training database. Merits and shortcomings of the proposed framework are also discussed.

Keywords: Porous metamaterial; Graph neural network; Variational graph autoencoder; Design optimization; Generative design; Connectivity.

1. INTRODUCTION

Metamaterials are artificially engineered architectured materials that exhibit extraordinary mechanical properties, such as stiffness-to-weight ratio [1], acoustic damping [2], vibration damping [3-5], wave trapping [6-9], and energy absorption properties [10-12]. These properties are derived from the topological characteristics of the metamaterials. Although a considerable amount of research has been devoted to metamaterial design, there has been relatively limited focus on porous metamaterials that enables applications envolving fluid-filled conditions [13-15]. In the field of fluid-filled porous metamaterials [1-5], designs have largely relied on traditional design approaches including parametric design and analytical modeling, which offer limited design flexibility. Therefore, there is a need for new methodologies that enable freeform design of porous metamaterials. It is crucial to emphasize the fundamental requirement of achieving *complete connectivity in both the solid and pore phases*. This means ensuring there are no disconnected materials floating in 3D space or enclosed cavities where fluid cannot reach. Previous methods of identifying and addressing disconnections, such as texture synthesis [16] and virtual temperature method [17], either cannot ensure complete connectivity or are computationally expensive.

Deep learning (DL) is promising in freeform metamaterial design [18-26], particularly in cases where analytical gradients of properties are unavailable. The design representation of metamaterial units is critical in selecting appropriate deep learning models and also determines their effectiveness in exploring the design space. When using parametric representation, the metamaterial units are represented by feature vectors [27-32]. Feedforward neural networks [33] have been used in conjunction with the parametric design representation. It is easy to explore the low dimensional parametric design space, but the design degree of freedom is limited. Another way of representing metamaterial units is to use their surface geometries [24]. Since different surfaces can be described by distinct mathematical equations, this method allows for a more nuanced representation of metamaterial units. However, the diversity in surface geometry equations presents a challenge in creating a generalized design space that can accommodate all possible design variations. Pixel/voxel-based representation offers great flexibility in the freeform design of metamaterial units. Convolutional neural networks (CNNs), and neural networks with convolutional layers (e.g., variational autoencoder, generative adverseral neural network, etc.), are the most used models when dealing with pixel/voxel-based design space [34]. Pixel/voxel-based representation can capture highly complex geometrical features with a sufficient resolution [22]. However, this approach also demands higher computational resources and complexity in model training and design exploration. Metamaterial topologies can also be described using graphs, comprising a set of nodes and edges connecting them. Graph-based representations are well-suited for truss-lattice [35], origami-based [36], and shell-based [24] structures. Graph neural network (GNN) [37, 38] is an effective approach for capturing the relationships between nodes within a graph. GNN also demonstrate significant potential in the field of metamaterial modeling and design. Guo et al. [39] employed GNNs to address boundary value problems in the design of architected materials. Dold et al. [40] introduced a differentiable graph-structured model for designing lattice materials, which uses a GNN as the surrogate model for structural analysis. Meyer et al. [24] proposed a graph representation method for shell-lattices, utilizing a GNN to establish the structure-property relationship, thereby enabling the design of novel metamaterial structures. The above-mentioned works only focus on lattice-based or shell-based graph structures. Zhang et al. [41] introduced a variant design generation method to create pixelated auxetic metamaterials, utilizing a multiscale geometry-informed Graph U-net capable of predicting complex nonlinear deformations. However, this method does not ensure complete connectivity of the solid and void phases, making it unsuitable for generalized porous metamaterial unit design.

Previous research works on porous metamaterial unit design either fail to ensure complete connectivity in both solid and pore phases or restrict the design space to simple structures that can easily verify connectivity [42-46]. To resolve this issue, the objective of this paper is to establish a graph neural network-based design approach for the freeform design of porous metamaterial units, while ensuring complete connectivity in both solid and pore/fluid phases. To achieve this, we first propose a novel method for generating highly diverse connectivity-guaranteed porous metamaterial samples for model training. Then three different variational graph autoencoder (VGAE)-based generative models are established and compared in terms of their reconstruction accuracies. Finally, a VGAE-based design approach is proposed for the inverse design of porous metamaterial units infilled with liquid, which is demonstrated through an engineering case study focusing on thermal conductivities.

In Section 2, we propose a graph-based approach for generating metamaterial unit samples and simulating their thermal conductivity properties with liquid infill. In Section 3, we propose and compare three variational graph autoencoder (VGAE)-based generative models. In section 4, we present a VGAE-based design approach for the inverse design of porous metamaterial units. In Section 5, we demonstrate the application of the proposed framework to obtain novel metamaterial unit designs with desired thermal conductivity values. Section 6 concludes this study.

2. GENERATION OF POROUS METAMATERIAL UNITS WITH COMPLETE CONNECTIVITY IN BOTH SOLD AND PORE PHASES

A graph-based approach is proposed for creating complex porous microstructures in which both the solid phase and the pore phase are fully connected. Furthermore, a finite element simulation model is presented to obtain the heat conductivity properties of the microstructure samples to support the following machine learning studies.

2.1 Generation of Porous Metamaterial Units

Graphs have previously been employed in designing metamaterials [24, 36, 47] and microstructures [48]. The techniques introduced in previous studies harness graph theory to facilitate the computationally efficient creation of fully connected structures and the detection of any disconnections or isolated parts. The central idea of the proposed microstructure generation approach is to construct the porous metamaterial unit with two "interwoven" graphs, each representing a separate phase. As shown in Figure 1, the proposed approach consists of the following four steps.

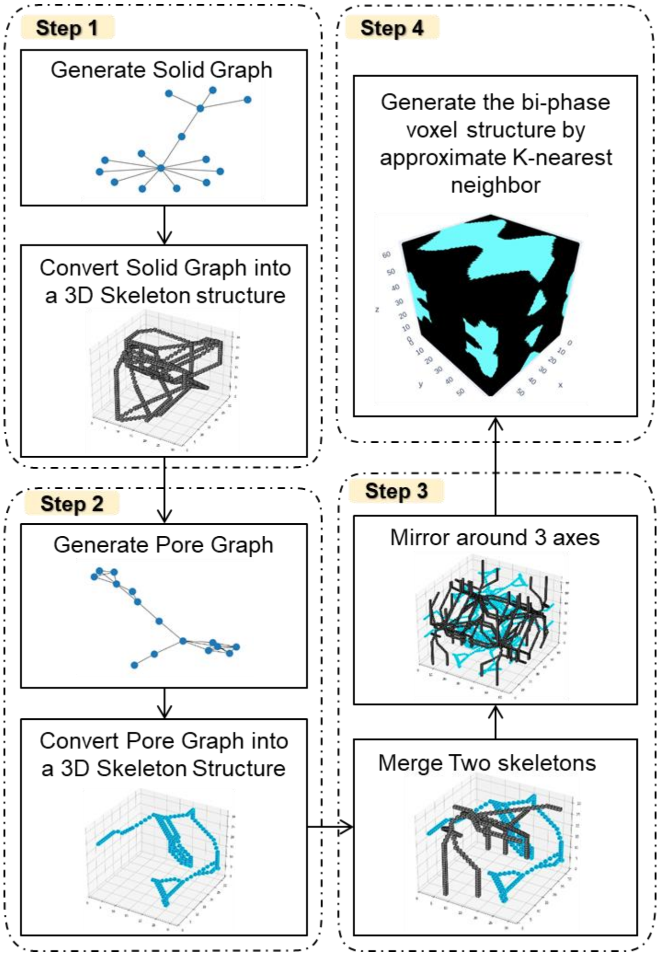

Figure 1: The proposed approach for generating complex porous metamaterial unit samples with complete connectivity in both solid and pore phases.

Step 1: Generate the solid phase skeleton structures as graph edges

In the first step, the goal is to create the graph that represent the skeleton of the solid phase ("solid graph"). Nodes in the solid graph are chosen randomly and distance-based connecting approach is used because by considering a lattice structure as a graph, the joints (nodes) should always be connected to other nearby joints.

As illustrated in Figure 1 and Figure 2a, the spatial distance between nodes is first calculated to form an edge. This edge is established if this distance is less than the set radius and if the nodes were not previously connected. The radius, which starts at 1 voxel, will incrementally increase by 1 voxel until all nodes are connected. To ensure a fully connected graph, the remaining isolated clusters are connected with each other by selecting and connecting a pair of nodes within each cluster based on their distances (Figure 2b).

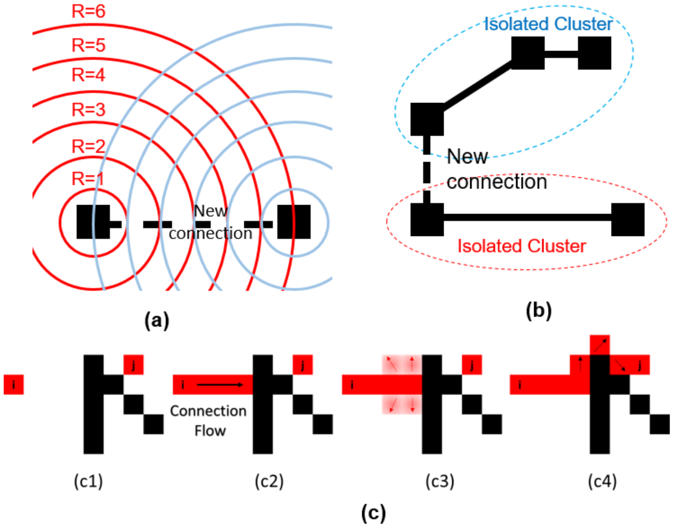

Figure 2: (a) The process of increasing the radius of search from nodes until they get connected. (b) Remaining isolated clusters will get connected based on their distances. (c) The process of connecting the pore phase's nodes i and j using Manhattan distance and rerouting: (1) Nodes i and j represent two voxels in the pore phase skeleton, while the solid phase skeleton is represented by black pixels. (2) The edges will be constructed using the Manhattan method until an obstruction is reached. (3) Exploring the 26 neighbor voxels to find the closest node to node j based on Euclidean distance, which is not part of the solid or existing pore phases. (4) After avoiding the intersection with the solid phase, the Manhattan method links nodes i and j through the node (i,j).

The nodes and edges of the graph are mapped to a voxel grid to create the voxel skeleton of the solid phase. In order to create edges, the basis Manhattan distance [49], which is the sum of the absolute differences of node's Cartesian coordinates, has been utilized. From the start node, the pathway (edge) is incrementally constructed, proceeding one voxel at a time along any one of the three axes. This approach ensures that each step moves closer to the end node in terms of the Manhattan distance, which in this article is called the "Manhattan method".

Step 2: Generate the pore phase skeleton structure as graph edges

The second step is to create the graph that represents the skeleton of the pore phase ("pore graph"). The pore phase, also referred to as the dual phase [24], is crucial for the movement of air and other fluids, necessitating a fully connected network to prevent isolated holes and blind pores, the same as the solid phase.

When creating the voxel skeleton of the pore phase, intersections with the voxel skeleton of the solid phase should be avoided. The path from one node to another may encounter obstacles presented by the edge of the solid phase graph, preventing the two nodes from connecting. A rerouting strategy based on the Manhattan method is developed to address this issue (Figure 2c). If the path between a pair of nodes is blocked by a voxel in the solid skeleton, the algorithm will select an edge voxel from the neighbor voxels that is not part of the solid skeleton. Then it will proceed with the Manhattan method to search the remaining voxels on the edge.

Step 3: Integrate the solid and pore skeletons

After merging the two skeletons into one voxel image, the voxel image is mirrored along all three axes to create a symmetric metamaterial unit structure. This mirroring operation ensures the periodic boundary condition.

Step 4: Assign remaining voxels to each of the two phases

The remaining unlabeled voxels in the 3D image are assigned to either the solid phase or the pore phase by the approximate K-nearest neighbor-based clustering using K-Dimensional tree (K-D tree) [50].

The major advantage of this approach is that both phases are *inherently fully connected*. No additional post-processing is required to eliminate "enclosed voids" or "disconnected materials". In this work, we represent the solid phase and the pore phase with 15 nodes each. By randomizing the locations of the input nodes within a $32 \times 32 \times 32$ domain, we generate a highly diverse metamaterial unit database with 15,000 samples, each with a voxel size of $64 \times 64 \times 64$ (Figure 3).

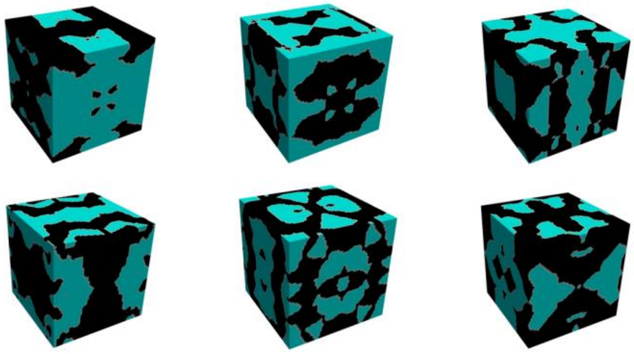

Figure 3: Diversity of the training samples: several examples of metamaterial unit samples in the created database.

2.2 Heat Transfer Simulation

The thermal conductivities of 3D fluid-solid metamaterial units are simulated through a user defined steady state thermal analysis subroutine in ABAQUS 2022. Each voxel is converted to a cubic element, so the mesh has $64 \times 64 \times 64$ elements. The constant temperatures at the two opposing sides of the sample are set as $\mathrm{T_1} = 20°\mathrm{C}$ and $\mathrm{T_2} = 120°\mathrm{C}$. Conduction is the sole source of heat transfer within the volume of the samples. Heat transfer throughout each point under steady state conditions in the sample geometry can be modeled using the Fourier equation shown in Eq. 1:

$$\mathrm{Q} = -\mathrm{kA}\frac{\Delta\mathrm{T}}{\Delta\mathrm{x}} \tag{1}$$

where Q represents the rate of heat transfer, k represents thermal conductivity of the material, A represents the cross-sectional area through which heat is being transferred, $\Delta\mathrm{T}$ represents the temperature difference and $\Delta\mathrm{x}$ represents the length of the heat transfer area. The fluid phase is water with heat conductivity $\mathrm{k_{water}} = 0.598\frac{\mathrm{W}}{mK}$, and the solid phase is aluminum with heat conductivity $\mathrm{k_{Al}} = 237\frac{\mathrm{W}}{mK}$. The fluid component of the metamaterial was modeled as stationary with zero velocity throughout the sample. Constant thermal conductivities were defined for the water and aluminum elements within the simulation. A temperature gradient was modeled by defining temperature boundary conditions along the axis of interest, with additional simulations completed for each sample modeling heat transfer in the x, y, and z directions.

After simulating the heat transfer through the material, the temperature and heat flux can be calculated at each node in the material. The average conductivity $\mathrm{k_{avg}}$ for each sample is calculated from the heat fluxes recorded on the surfaces using Eq. 2.

$$\mathrm{k_{avg}} = -\frac{1}{\mathrm{n}}\left(\frac{\mathrm{q_1}}{\frac{\mathrm{dT}}{\mathrm{dx_1}}} + \frac{\mathrm{q_2}}{\frac{\mathrm{dT}}{\mathrm{dx_2}}} + \cdots + \frac{\mathrm{q_n}}{\frac{\mathrm{dT}}{\mathrm{dx_n}}}\right) \tag{2}$$

where n represents the number of nodes where heat flux and temperature gradients are recorded. q_i represents the heat flux at the i^{th} node. $\frac{\mathrm{dT}}{\mathrm{dx}_i}$ represents the temperature gradient at the i^{th} node.

The heat conduction simulations are conducted on all 15,000 samples in each cartesian direction. Histograms of all samples properties are shown in Figure 4b.

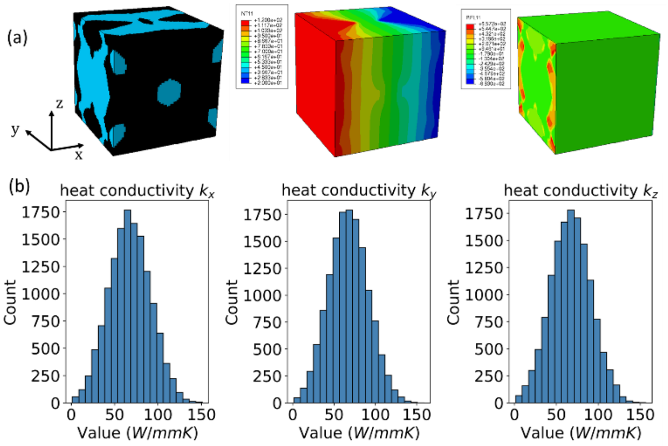

Figure 4: Heat transfer simulation in ABAQUS. (a) The input porous metamaterial unit, the temperature gradient, and the heat flus on the surfaces of the sample. (b) Histogram of the simulated heat conductivity values $\mathrm{k_x}$, $\mathrm{k_y}$, and $\mathrm{k_z}$ from X, Y, and Z directions of all samples.

3. VARIATIONAL GRAPH AUTOENCODER-BASED DEEP GENERATIVE MODELS

3.1 Variational Graph Autoencoder (VGAE)

GNNs have received increasing attention in recent years for their capability to analyze graph-structured data. They represent a distinct class of neural networks, distinguished by their capability to process graph-represented data (referred to as graphs) rather than vectorized or image data [38]. The variational graph autoencoder (VGAE) was firstly introduced by [51], representing a unsupervised learning framework for graph-structured data. VGAE is effective in generating new undirected graph samples by sampling the latent feature space. VGAE consists two parts: an encoder that takes the adjacency matrix $\mathbf{A}$ and feature matrix $\mathbf{X}$ as inputs and generates the latent vector $\mathbf{Z}$, and a decoder that decodes the latent vector $\mathbf{Z}$ to reconstruct the original graph. The encoder (inference model) can be expressed as:

$$q(\mathbf{z}|\mathbf{X},\mathbf{A}) = \Pi_{i=1}^{N} q(\mathbf{z}_i|\mathbf{X},\mathbf{A}) \tag{3}$$

$$q(\mathbf{z}_i|\mathbf{X},\mathbf{A}) = N(\mathbf{z}_i|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i{}^2)) \tag{4}$$

where we define an undirected, unweighted graph $G = (v, \varepsilon)$ with $N = |v|$ nodes. $\mathbf{A}$ is the adjacency matrix of graph G, and $\mathbf{D}$ is the feature matrix of G. $\mathbf{z}_i$ represents a latent variable, and the latent vector $\mathbf{z}$ is an $N \times F$ matrix. F is the dimension of the latent vector to which each node is mapped. $\mathbf{X}$ represents the node features matirx with the shape of $N \times D$. The edge and node attributes are input into the convolution layer, which is based on the classical graph convolutional network (GCN). Therefore, $\boldsymbol{\mu} = \mathrm{GCN}_{\boldsymbol{\mu}}(\mathbf{X},\mathbf{A})$ is the matrix of mean vectors $\boldsymbol{\mu}_i$, $\log\boldsymbol{\sigma} = \mathrm{GCN}_{\sigma}(\mathbf{X},\mathbf{A})$. During the covolution, for each graph, given the node feature matrix $\mathbf{X}$ and the edge feature matrix $\mathbf{A}$, we then have $\mathbf{H} = \mathbf{A'XW}$, where $\mathbf{W}$ is the trainable weight matrix in the VGAE, and $\mathbf{A'} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}$. Here

$\mathbf{D}$ refers to the normalized degree matrix of the graph. More details can be found in literature [51].

The decoder (generative model), constructed with fully connected layers, takes the latent vector $\mathbf{z}$ as input to reconstruct the original graph G. When reconstructing edges, the reconstructed adjacency matrix $\widetilde{\mathbf{A}}$ is:

$$\widetilde{\mathbf{A}} = \sigma(\mathbf{z}\mathbf{z}^{\mathrm{T}}) \tag{5}$$

To reconstruct the nodes' coordinates ($\widetilde{\mathbf{X}}$), linear layer can be used in the decoder, followd by a node-wise softmax operation to obtain $\widetilde{\mathbf{X}}$.

The VGAE model is trained to optimize the variational lower bound $\mathcal{L}$:

$$\mathcal{L}_{VGAE} = \mathrm{E}_{q(\mathbf{z}|\mathbf{X},\mathbf{A})}[\log p(\mathbf{A}|\mathbf{z})] - \mathrm{KL}[q(\mathbf{z}|\mathbf{X},\mathbf{A})||p(\mathbf{z})] \tag{6}$$

where $\mathrm{KL}[q(\cdot)||p(\cdot)]$ is the Kullback-Leibler divergence between $q(\cdot)$ and $p(\cdot)$. We use Gaussian prior $p(\mathbf{z}) = \Pi_i p(\mathbf{z}_i) = \Pi_i N(\mathbf{z}_i|0,\mathbf{I})$. To optimize the parameters of the Gaussian distribution, we perform mini-batch gradient descent and leverage the reparameterization trick [52].

The VGAE architecture creates an information bottleneck within the latent representation, retaining only essential and meaningful information necessary for reconstructing the original graphs. Within this latent space, graphs with similar topological and geometrical characteristics are positioned closely together. Graphs that are farther apart in the latent space can be smoothly transitioned from one to another by traversing along a continuous path in the space.

3.2 VGAE with Regressor

To predict the structure properties from latent variables, one strategy is to integrate the regressors with the deep generative model [7, 22, 53-57]. There are three strategies for establishing a learning framework with a VGAE and a regressor (Figure 5):

- *Parallel processes* (Figure 5a), where the variational VGAE and the regressor model are trained independently without any interplay. This strategy offers flexibility in model training, but one major limitation is that it does not allow structure design optimization as the regressor is not linked to the structure design variables (latent variables).
- *Sequential integration* (Figure 5b) connects the regressor to the output of the decoder. The structure's properties are predicted based on the reconstructed image.
- *Latent space mediated integration* (Figure 5c) connects the regressor to the latent space. The structure's properties are predicted from the latent variables. The VGAE and the regressor are trained simultaneously. The regressor serves as regularization in learning the latent space.

Literatures [54, 55] have shown that both *sequential integration* and *latent space mediated integration* are capable of embedding the structure property information into the latent space, therefore leads to improved property prediction accuracy. In this work, we adopt the *sequential integration* and *latent space mediated integration* strategies in developing the VGAE-based design approach.

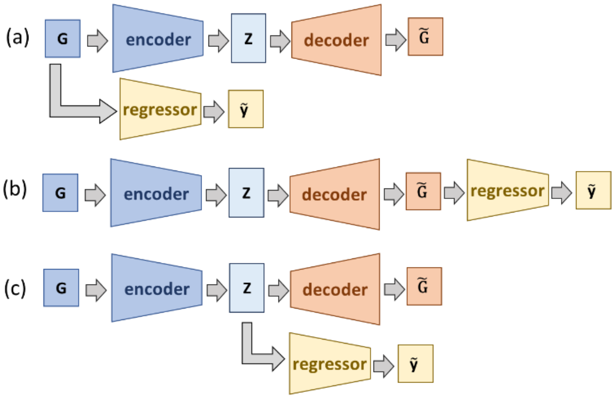

Figure 5: Three strategies of integrating a regressor with a deep generative model: (a) Parallel processes. (b) Sequential integration. (c) Latent space mediated integration.

3.3 VGAE-based Generative Models for Designing Connectivity-guarateed Porous Metamaterial Units

To improve the edge prediction accuracy and to leverage the characteristics of our generated dataset, we proposed three VGAE-based generative models which based on the *sequential integration* and *latent space mediated integration* strategies (Figure 6b~d).

The graph representation G of the porous metamaterial structure consists two parts: node coordinates $\mathbf{X}$, and the adjacency matrix $\mathbf{A}$ which represents the edge features. $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{A}}$ are the reconstructed node coordinates and adjacency matrix based on the latent variables learned by the deep generative model.

In addition to the three proposed models, a baseline VGAE model (Figure 6a) is created. The baseline model encodes $\mathbf{X}$ and $\mathbf{A}$ into the latent space, and both $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{A}}$ are reconstructed by the decoder. In Section 3.4, we will demonstrate that VGAE fails to reconstruct $\widetilde{\mathbf{A}}$ accurately. Therefore, in the following three proposed models, only the node coordinates $\widetilde{\mathbf{X}}$ are reconstructed by the deep generative model, while the edges (adjacency matrix $\widetilde{\mathbf{A}}$) are created using the Manhattan method introduced in Section 2.

The first model, VGAE-(b), is based on the sequential integration strategy. An input graph, which includes node coordinates and adjacency matrix, is compressed into the latent space by an encoder. In the reconstructed graph, the node coordinates are generated by decoder and the edges are generated by the Manhattan method. A feedforward neural network regressor takes the reconstructed graph to predict the property of interest, which is the heat conductivity of the fluid-infilled metamaterial units in this study.

The second model, VGAE-(c), is based on the latent space mediated integration strategy. The regressor, which is also a feed forward neural network, is linked to the latent space. The VGAE only reconstructs the node features, and the edges are reconstructed by the Manhattan method.

The third model, VGAE-(d), is based on the sequential integration strategy. This model differentiate itself from VGAE-(b) by utilizing the reconstructed voxel image, instead of the reconstructed graph, to predict the properties. As the voxel image is used in property prediction, the regressor is created with a CNN architecture.

To ensure a fair comparison, we use identical architectures for the encoders and decoders in the three VGAE-based generative models, as shown in Appendix A1. The encoder utilizes the GraphSAGE model [58], a framework that has been proven effective in generating node embeddings. The decoder employs dense layers and takes only the latent feature $\mathbf{z}$ as its input. For the regression, the regressors in VGAE-(b) and VGAE-(c) models utilize the GraphSAGE model, whereas the regressor in VGAE-(d) is a CNN. The detailed architecture of the regressors of VGAE-(b), VGAE-(c) and VGAE-(d) are shown in Appendix A1.

When training the abovementioned models, the loss for the VGAE model and the loss for the regressor are considered simultaneously. Thus, the loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{VGAE} + \mathcal{L}_{reg} \tag{7}$$

The regressor loss is measured by the mean squared error, $\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_i - \widetilde{\mathbf{Y}}_i)^2$, where $\mathbf{Y}_i$ represents the true response of the i^{th} sample, $\widetilde{\mathbf{Y}}_i$ represents the predicted response of the i^{th} sample.

For model implementation, we use the PyTorch Geometric library. The models are all trained on a Nvidia RTX8000 GPU. The porous metamaterial dataset is divided into two sets: 13500 (90%) for training and 1500 (10%) for testing. Adam is used as the optimizer for parameter optimization. The number of epochs is set to 400 for all the models. We also implement an early stopping criterion to halt training when the test loss begins to increase, thereby preventing overfitting.

3.4 Comparison and Validation of VGAE-based Generative Models

The baseline model and the three proposed VGAE-based generative models are compared and validated in two aspects:

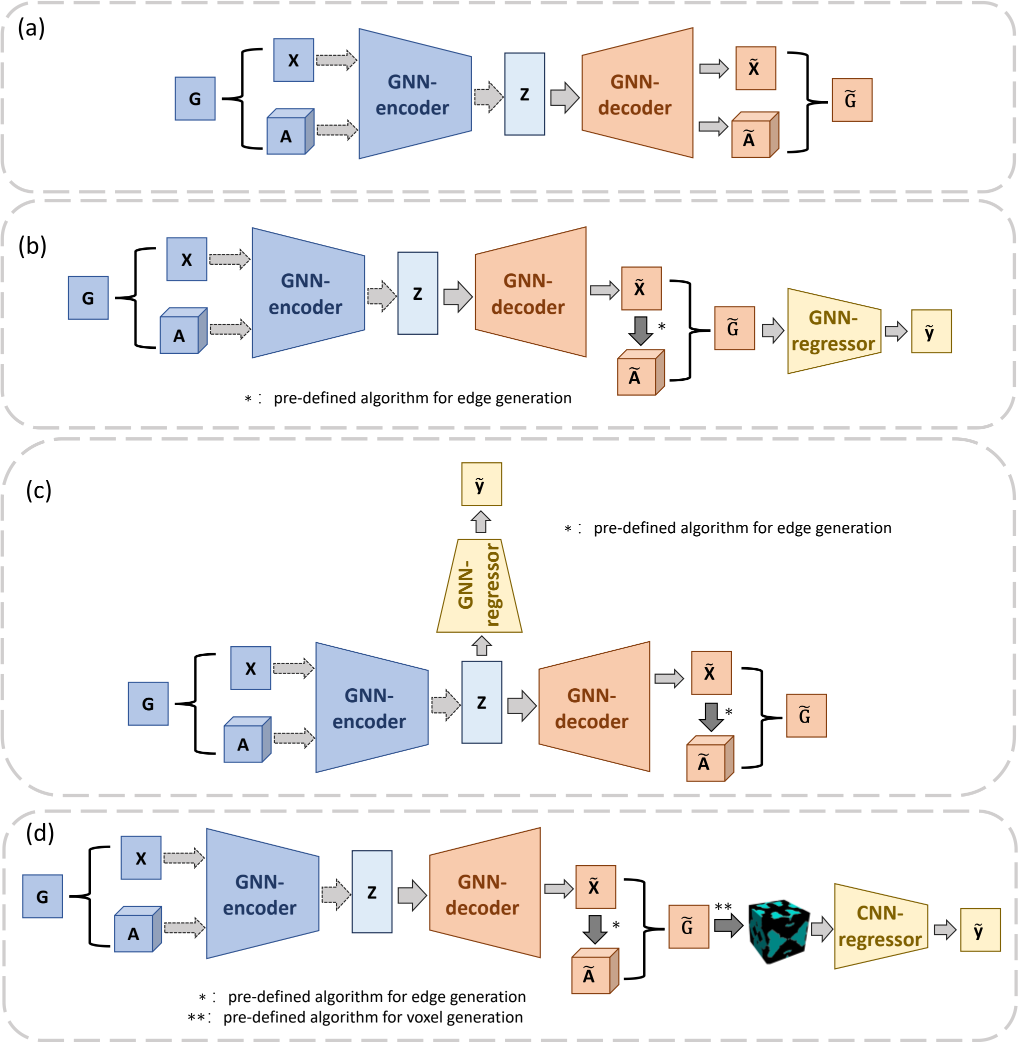

Figure 6: (a) Baseline VGAE model. (b)~(d) Three VGAE-based generative models based on sequential integration or latent space mediated integration strategy.

the reconstruction accuracy of the VGAE and the prediction accuracy of the regressor.

The accuracies of reconstructing node features $\widetilde{\mathbf{X}}$ and final voxelated metamaterial structures are measured by the differences between the original sample and the reconstructed sample. The differences between original node coordinates $\mathbf{X}$ and reconstructed node coordinates $\widetilde{\mathbf{X}}$ measured by the coefficient of determination (R^2), which quantifies the deviation of data from their true mean value:

$$\mathrm{R}^2 = 1 - \frac{\sum(\mathbf{Y}_i - \widetilde{\mathbf{Y}}_i)^2}{\sum(\mathbf{Y}_i - \overline{\mathbf{Y}}_i)^2} \tag{8}$$

where $\mathbf{Y}_i$ represents the true response of the i^{th} sample, $\widetilde{\mathbf{Y}}_i$ represents the predicted response of the i^{th} sample, and n_{sample} represents the total number of sample points. $\overline{\mathbf{Y}}_i$ is the averaged value of $\mathbf{Y}_{i(true)}$ and $\overline{\mathbf{Y}}_i = \frac{1}{n_{sample}} \sum \mathbf{Y}_i$. A higher R^2 value means a more accurate model. We also perform a voxel-to-voxel comparison of the original and reconstructed voxel structures, which are defined as:

$$\alpha_{\text{voxel}} = 1 - \frac{1}{\mathrm{N}} \frac{1}{l^3} \sum_{i=1}^{l} \sum_{j=1}^{l} \sum_{k=1}^{l} \left| \mathrm{O}_{ijk} - \mathrm{R}_{ijk} \right| \tag{9}$$

where N represents the total amount of structures used, l represents the side length of the structures in voxel, and $l = 64$ in this study. For the baseline model, the accuracy of reconstructing edge $\widetilde{\mathbf{A}}$ is also validated by R^2. It is to be noted that in the three VGAE-based generative models, edge reconstruction is carried out using the same Manhattan method as the one used in training sample generation. Therefore, there is no need to evaluate the accuracy of edge reconstruction. The accuracy of heat conductivity prediction $\tilde{\boldsymbol{y}}$ in the VGAE-based generative models is evaluated by R^2. The validation results are summarized in Table 1.

Table 1: Reconstruction/prediction accuracies of the baseline model and the VGAE-based generative models. $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{A}}$ and $\tilde{\boldsymbol{y}}$ are measured by R^2.

model		$\widetilde{\mathbf{X}}$	$\widetilde{\mathbf{A}}$	α_{voxel}	$\tilde{\boldsymbol{y}}$
Baseline	Training set	0.999	-7946	/	/
	Test set	0.999	-8966	/	/
VGAE-(b)	Training set	0.999	/	0.9165	0.235
	Test set	0.999	/	0.9163	0.206
VGAE-(c)	Training set	0.996	/	0.9166	0.078
	Test set	0.993	/	0.9139	0.056
VGAE-(d)	Training set	0.999	/	0.9302	0.995
	Test set	0.999	/	0.9291	0.974

The results indicate that VGAE fails to reconstruct $\widetilde{\mathbf{A}}$ accurately. All three VGAE-based generative models exhibit satisfactory performance in reconstructing node features, thereby enabling them to reconstruct voxel-based structures with small error. However, the VGAE-(b) and VGAE-(c) cannot accurately predict the structure-property relationship. This indicates that relying solely on the graph representation may not fully capture all geometric features crucial to thermal conductivity properties. VGAE-(d) outperforms the others in terms of prediction accuracy. VGAE-(d) utilizes the voxel image, instead of a reduced graph representation, to predict properties. Consequently, we select VGAE-(d) for the following task of VGAE-based design.

4. VGAE-BASED DESIGN APPROACH

A design approach is proposed based on VGAE-(d). The design approach utilize the pre-trained VGAE-based generative model and perform design search in the latent space (Figure 7). The design approach follows the following steps:

(1) Use the pretrained decoder to reconstruct node coordinates $\widetilde{\mathbf{X}}$ based on the latent variables, which are used as the structure design variables. Then the adjacency matrix $\widetilde{\mathbf{A}}$ is reconstructed using the pre-defined Manhattan method. The reconstructed graphs represent the skeletons of the two phases of the metamaterial unit.

(2) Create the voxel image of the metamaterial unit following Step 4 in Section 2, based on the skeletons. With the voxel images, the thermal conductivity values are obtained by the pretrained regressor model.

(3) Conduct design search approach in the latent space. The design problem can be formulated as:

$$\min_{\mathbf{z}} \quad [\mathrm{f}_1(\mathbf{z}), \mathrm{f}_2(\mathbf{z}) \dots, \mathrm{f}_{\mathrm{n_f}}(\mathbf{z})] \qquad (10)$$
$$\mathrm{s.t.} \qquad \mathrm{c_j}(\mathbf{z}) \leq 0$$

where $\mathbf{z}$ is the latent vector, $\mathrm{f_i}$ $(\mathrm{i} = 1,2, \dots, \mathrm{n_f})$ are the objective functions, $\mathrm{c_j}$ $(\mathrm{j} = 1,2, \dots, \mathrm{n_c})$ are the constraint functions. $\mathrm{f_i}$ and $\mathrm{c_j}$ are both functions of $\mathbf{z}$. The obtained optimal design is in the format of $\mathbf{z}$, and then the corresponding metamaterial unit image is reconstructed.

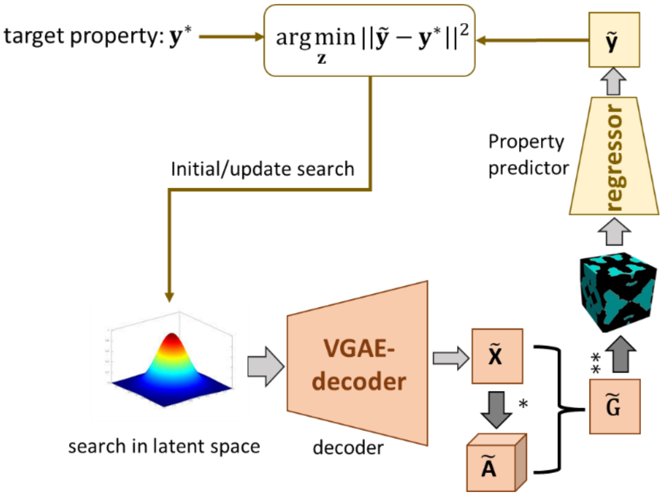

Figure 7: VGAE-based design approach.

5. CASE STUDY: DESIGN POROUS METAMATERIAL UNIT FOR THERMAL CONDUCTIVITY PROPERTIES UNDER VOLUME FRACTION CONSTRAINT

In this section, we demonstrate the capability of the proposed VGAE-based design approach for designing porous metamaterial units infilled with liquid for thermal conductivity properties. The design objective is to maximize thermal conductivity in both the x and y directions, while meeting two design constraints: the volume fraction of the solid phase is 0.7, within a tolerance of 0.01; and the thermal conductivity along the z direction is half that of the x direction, within a tolerance of 5. The design problem is formulated as follows:

$$
\begin{cases} \max\limits_{\mathbf{z}} & \mathrm{k_x}(\mathbf{z}) \\ \max\limits_{\mathbf{z}} & \mathrm{k_y}(\mathbf{z}) \end{cases} \tag{11}
$$

$$
\begin{aligned}
\text{s.t.} \quad & |\,\mathrm{VF_{solid}}(\mathbf{z}) - 0.7| \le 0.01 \\
& |2\mathrm{k_z}(\mathbf{z}) - \mathrm{k_x}(\mathbf{z})| \le 5 \\
& \min(\mathbf{z}) \le \mathbf{z} \le \max(\mathbf{z})
\end{aligned}
$$

Non-dominated Sorting Genetic Algorithm (NSGA-II) [59] is applied to explore the latent space to search designs that yield the highest thermal conductivities along the x and y directions. Subsequently, the optimial designs in latent space are decoded to reconstruct the corresponding metamaterial units in the form of graphs and voxel images. As illustrated in Figure 8a, the proposed design approach successfully generates designs that satisfy the design constraints, as well as ensuring the designs' connectivities of both solid component and component phase.

The true properties of these optimal designs are confirmed by simulations. The predicted and actual values of both objective functions and constraint functions are shown in Table 2. Out of 9 optimal design candidates, 6 strictly meet the design constraints. We also find that the optimal design candidates outperform all existing designs in the database. No design in the original database meets the design constraints that thermal conductivity in the z direction is half that of the x direction.

Last but not the least, we demonstrate the superiority of the VGAE-based approach over the convolutional variational autoencoder (cVAE)-based approach through a comparative study using the same design objective (Eq. 11). Details of training and validating the cVAE model are provided in Appendix A2. As shown in Figure 8b, designs produced by the cVAE-based approach fail to ensure connectivity within the pore/fluid phase, underscoring the VGAE-based approach superiority in maintaining connectivity in both solid and pore phases. The failure of cVAE is inherent in its utilization of the voxel representation of the structures. For example, a few extra

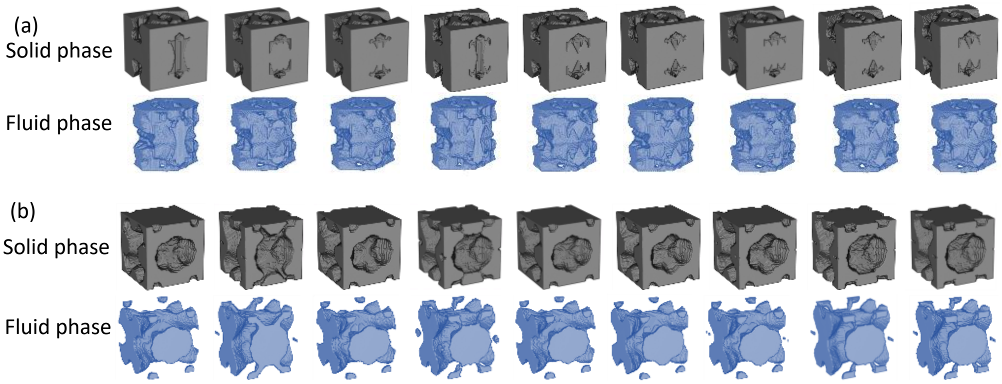

Figure 8: Optimal design candidates obtained by (a) VGAE-based design approach, (b) cVAE-based design approach.

Table 2: Ground truth and predited objectives and constraints values of the optimal design candidates using VGAE-based design approach. P represents the predicted values, T represents ground truth values. The design candidates satisfying all constraints are highlighted in grey.

Design Candidates	Objective				Constraint		
	$\mathrm{k_x}$ (W/mmK)		$\mathrm{k_y}$ (W/mmK)		$\mathrm{k_z}$ (W/mmK)		$\mathrm{VF_{solid}}$
	P	T	P	T	P	T	P/T
1	138.418	139.990	113.009	113.261	69.169	62.773	0.696
2	139.563	140.822	111.785	110.373	70.052	69.275	0.701
3	138.144	141.095	113.075	113.794	69.938	73.021	0.690
4	138.240	140.502	113.150	105.916	70.060	65.060	0.698
5	138.871	140.917	112.587	113.348	69.771	71.872	0.700
6	134.751	141.274	115.023	116.795	70.241	65.084	0.696
7	137.568	141.048	113.557	113.795	70.820	71.795	0.700
8	141.353	141.826	112.863	113.107	71.708	71.268	0.700
9	137.841	141.352	114.025	113.050	71.012	71.461	0.700

voxels in the reconstruction may obstruct a conduit in the pore network, resulting in the formation of a fully enclosed pore. The ground truth and predicted objective and constraint values of the optimal designs obtained by the cVAE-based approach are presented in Appendix A3.

6. CONCLUSION

This paper presents a graph-based method for generating porous metamaterial units with complete connectivity in both solid and pore phases, as well as a VGAE-based approach for the property-driven generative design of connectivity-guaranteed metamaterial units. Our major conclusions are summarized as follows:

1. We established a novel method for generating voxelated, connectivity-guaranteed metamaterial units using graph data.
2. We established and compared three VGAE-based generative models based on sequential integration or latent space mediated integration and identify the best model structure for design applications.
3. The VGAE-based design approach proves to be effective in the generative design of porous metamaterial units with desired properties and guaranteed connectivity.

However, we also acknowledge some shortcomings in our proposed design approach:

1. Despite the VGAE-based model's high accuracy in reconstructing node coordinates, slight variations in node coordinates can result in significant discrepancies in voxelated structures.
2. The proposed VGAE-based generative models and VGAE-based design approach rely on the pre-defined edge connection algorithm (the Manhattan method), limiting their generalizability to other dataset.

For future work, we intend to further enhance the edge reconstruction process and broaden the generalizability of our design methodology.

ACKNOWLEDGEMENT

We gratefully acknowledge the financial support from the United States Army DEVCOM Ground Vehicle Systems Center, National Science Foundation (CMMI-2142290), and U.S. Department of Education, Graduate Assistance in Areas of National Need (GAANN) program (P200A210089). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. OPSEC8378

APPENDIX

A.1 Hyperparameters in the three VGAE-based generative models

Table A1: The dimensionality of each layer in the encoder, decoder and regressor.

Encoder	
Block	Specifications
GraphSAGE-1	GraphSAGE32+ReLU
GraphSAGE-2	GraphSAGE32+ReLU
Encoder FC	32+ReLU→3
Mean, Variance, Latent vector	100
Decoder	
Block	Specifications
Decoder FC	3+ReLU→32+ReLU→32 →3
Regressor-VGAE(b)	
Block	Specifications
Reg-GraphSAGE-1	GraphSAGE32+ReLU
Reg FC	32+ReLU→3
Regressor-VGAE(c)	
Block	Specifications
Reg-GraphSAGE-1	GraphSAGE32+ReLU
Reg FC	32+ReLU→3
Regressor-VGAE(d)	
Block	Specifications
Reg-Conv3d-1	(Conv32+ReLU) ×3+ MaxPooling
Reg-Conv3d-2	(Conv64+ReLU) ×3+ MaxPooling
Reg-Conv3d-3	(Conv96+ReLU) ×3+ MaxPooling
Reg FC	2592+ReLU→256+ReLU→3

A.2 cVAE-based generative model and cVAE-based design approach

We also established a cVAE-based generative model (Figure A1a), which comprises an encoder and a decoder constructed with 3D convolutional layers and a regressor. To ensure a fair comparison with the VGAE-(d) generative model, the cVAE-based generative model employs the sequential integration strategy as well. We utilze the same training and test set split and the same training process as indicates in section 3.5.

Validations of the cVAE's accuracy includes voxel-wise comparisons between the original and reconstructed structures, as well as assessing the regressor's performance in predicting thermal conductivity using Eq. 9 and Eq. 8, respectively. The accuracy of the cVAE-based generative model is presented in Table A1. Additionally, we applied the cVAE to reconstruct the training and test samples to evaluate its capability of capturing the connectivity characteristics in the pore and solid phases. It is important to note that our training and test sets only include structure samples with complete connectivity in both solid and pore phases. However, the reconstructions by cVAE cannot fully maintain the connectivity characteristics. Among the reconstructions of the training samples, 80.2% have isolated voxel clusters in either the solid or the pore phases, while among the reconstructions of the test samples, 83.8% have isolated voxel clusters in either the solid or the pore phases. This outcome underscores a limitation of the cVAE-based generative model: small deviations in the reconstructed structures may potentially result in disconnections in both solid and pore phases. For example, a few extra voxels in the reconstruction may obstruct a conduit in the pore network, resulting in the formation of a fully enclosed pore.

Then a cVAE-based design approach is proposed based on the cVAE-based generative model (Figure A1b). The design search method is the same as the one in Section 4. The difference lies in the decoder, which directly decodes the latent space parameters into the voxel structure, thereby combining steps (2) and (3) from Section 4.

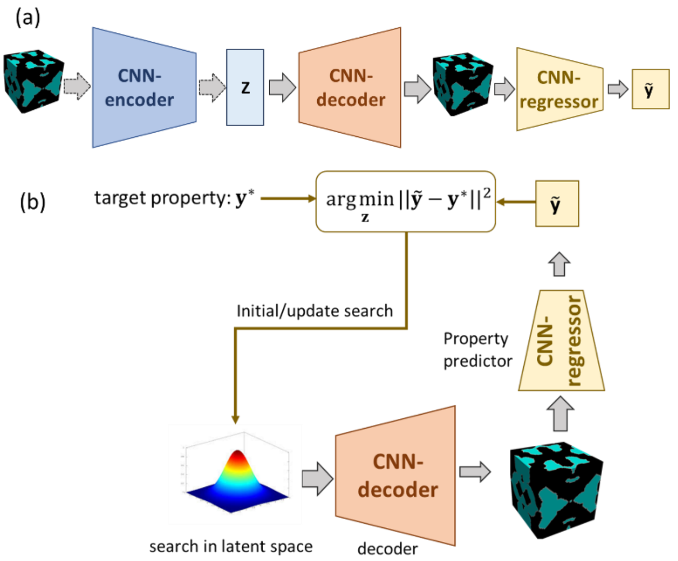

Figure A1: (a) The proposed cVAE-based generative model. (b) The cVAE-based design approach.

Table A2: Reconstruction accuracy of the voxelated structures and prediction accuracy of regressor in the cVAE-based generative models.

model		α_{voxel}	$\tilde{y}$
cVAE	Training set	0.8938	0.935
	Test set	0.8760	0.928

A.3 Table A3: Ground truth and predited objectives and constraints values of the optimal design candidates using cVAE-based design framework. P represents the predicted values, T represents ground truth values. The design candidates satisfying all constraints are highlighted in grey.

	Objective				Constraint		
	k_x (W/mmK)		k_y (W/mmK)		k_z (W/mmK)		VF_{solid}
	P	T	P	T	P	T	P/T
1	131.68	123.33	142.65	139.45	66.21	66.57	0.71
2	130.32	129.57	144.21	131.85	63.33	61.78	0.70
3	134.25	123.28	140.22	136.92	67.22	67.07	0.70
4	137.66	125.92	136.58	133.62	66.34	62.31	0.70
5	135.65	123.13	139.01	139.10	68.12	63.56	0.70
6	133.25	123.67	140.82	138.26	67.36	62.92	0.70
7	134.23	124.79	140.21	138.08	67.25	63.69	0.70
8	134.72	124.76	139.61	138.06	67.75	62.81	0.71
9	133.80	126.76	139.89	133.72	67.88	62.91	0.70

REFERENCES

1. Zheng, X., et al., *Ultralight, ultrastiff mechanical metamaterials.* Science, 2014. **344**(6190): p. 1373-1377.
2. Chen, H. and C.T. Chan, *Acoustic cloaking in three dimensions using acoustic metamaterials.* Applied physics letters, 2007. **91**(18): p. 183518.
3. Garland, A.P., et al., *Coulombic friction in metamaterials to dissipate mechanical energy.* Extreme Mechanics Letters, 2020. **40**: p. 100847.
4. Claeys, C., et al., *Design and validation of metamaterials for multiple structural stop bands in waveguides.* Extreme Mechanics Letters, 2017. **12**: p. 7-22.
5. Qian, J., et al., *Optimization design of metamaterial vibration isolator with honeycomb structure based on multi-fidelity surrogate model.* Structural and Multidisciplinary Optimization, 2021. **64**: p. 423-439.
6. Wang, Z., et al., *Design of Phononic Bandgap Metamaterials based on Gaussian Mixture Beta Variational Autoencoder and Iterative Model Updating.* Journal of Mechanical Design, 2022: p. 1-35.
7. Wang, Z., et al. *A Gaussian Mixture Variational Autoencoder-Based Approach for Designing Phononic Bandgap Metamaterials*. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2021. American Society of Mechanical Engineers.
8. Wang, Z., et al. *Phononic Metamaterial Design via Transfer Learning-Based Topology Optimization Framework*. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2022. American Society of Mechanical Engineers.
9. Gurbuz, C., et al., *Generative adversarial networks for the design of acoustic metamaterials.* J Acoust Soc Am, 2021. **149**(2): p. 1162.
10. Alberdi, R., et al., *Multi-morphology lattices lead to improved plastic energy absorption.* Materials & Design, 2020. **194**: p. 108883.
11. Xu, H. and Z. Liu, *Control variate multifidelity estimators for the variance and sensitivity analysis of mesostructure–structure systems.* ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, 2019. **5**(2): p. 020907.
12. Liu, Z., H. Xu, and P. Zhu, *An adaptive multi-fidelity approach for design optimization of mesostructure-*

structure systems. Structural and Multidisciplinary Optimization, 2020. **62**: p. 375-386.

13. Zhang, Q., K. Zhang, and G. Hu, *Tunable fluid-solid metamaterials for manipulation of elastic wave propagation in broad frequency range.* Applied Physics Letters, 2018. **112**(22).
14. He, Z.-H., Y.-Z. Wang, and Y.-S. Wang, *Active feedback control of sound radiation in elastic wave metamaterials immersed in water with fluid–solid coupling.* Acta Mechanica Sinica, 2021. **37**: p. 803-825.
15. Song, Y. and Y. Shen, *Highly morphing and reconfigurable fluid–solid interactive metamaterials for tunable ultrasonic guided wave control.* Applied Physics Letters, 2022. **121**(26).
16. Gao, D., et al., *Connectivity-guaranteed porous synthesis in free form model by persistent homology.* Computers & Graphics, 2022. **106**: p. 33-44.
17. Swartz, K.E., et al., *Manufacturing and stiffness constraints for topology optimized periodic structures.* Structural and Multidisciplinary Optimization, 2022. **65**(4): p. 129.
18. Yang, K.V., et al., *Porosity formation mechanisms and fatigue response in Al-Si-Mg alloys made by selective laser melting.* Materials Science and Engineering: A, 2018. **712**: p. 166-174.
19. Liu, R., et al. *Materials discovery: Understanding polycrystals from large-scale electron patterns*. in *2016 IEEE International Conference on Big Data (Big Data)*. 2016. IEEE.
20. Jha, D., et al., *Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks.* Microscopy and Microanalysis, 2018. **24**(5): p. 497-502.
21. Cang, R., et al., *Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design.* Journal of Mechanical Design, 2017. **139**(7).
22. Wang, Z., et al., *Design of Phononic Bandgap Metamaterials Based on Gaussian Mixture Beta Variational Autoencoder and Iterative Model Updating.* Journal of Mechanical Design, 2022. **144**(4): p. 041705.
23. Wang, L., et al., *Data-driven metamaterial design with Laplace-Beltrami spectrum as "shape-DNA".* Structural and multidisciplinary optimization, 2020. **61**: p. 2613-2628.
24. Meyer, P.P., et al., *Graph-based metamaterials: Deep learning of structure-property relations.* Materials & Design, 2022. **223**: p. 111175.
25. Bastek, J.-H., et al., *Inverting the structure–property map of truss metamaterials by deep learning.* Proceedings of the National Academy of Sciences, 2022. **119**(1): p. e2111505119.
26. Kumar, S., et al., *Inverse-designed spinodoid metamaterials.* npj Computational Materials, 2020. **6**(1): p. 73.
27. Gongora, A.E., et al., *Using simulation to accelerate autonomous experimentation: A case study using mechanics.* Iscience, 2021. **24**(4).
28. Gongora, A.E., et al., *A Bayesian experimental autonomous researcher for mechanical design.* Science advances, 2020. **6**(15): p. eaaz1708.
29. Shin, D., et al., *Spiderweb nanomechanical resonators via bayesian optimization: inspired by nature and guided by machine learning.* Advanced Materials, 2022. **34**(3): p. 2106248.
30. Ma, J., et al., *Inverse design of broadband metasurface absorber based on convolutional autoencoder network and inverse design network.* Journal of Physics D: Applied Physics, 2020. **53**(46): p. 464002.
31. Huang, L., et al., *Impact of resonator geometry and its coupling with ground plane on ultrathin metamaterial perfect absorbers.* Applied Physics Letters, 2012. **101**(10).
32. Wilt, J.K., C. Yang, and G.X. Gu, *Accelerating auxetic metamaterial design with deep learning.* Advanced Engineering Materials, 2020. **22**(5): p. 1901266.
33. Bebis, G. and M. Georgiopoulos, *Feed-forward neural networks.* Ieee Potentials, 1994. **13**(4): p. 27-31.
34. Gu, J., et al., *Recent advances in convolutional neural networks.* Pattern recognition, 2018. **77**: p. 354-377.
35. Indurkar, P.P., et al., *Predicting deformation mechanisms in architected metamaterials using GNN.* arXiv preprint arXiv:2202.09427, 2022.
36. Yamaguchi, K., et al., *Graph-theoretic estimation of reconfigurability in origami-based metamaterials.* Materials & Design, 2022. **213**: p. 110343.
37. Scarselli, F., et al., *The graph neural network model.* IEEE transactions on neural networks, 2008. **20**(1): p. 61-80.
38. Wu, Z., et al., *A comprehensive survey on graph neural networks.* IEEE transactions on neural networks and learning systems, 2020. **32**(1): p. 4-24.
39. Guo, K. and M.J. Buehler, *A semi-supervised approach to architected materials design using graph neural networks.* Extreme Mechanics Letters, 2020. **41**: p. 101029.
40. Dold, D. and D.A. van Egmond, *Differentiable graph-structured models for inverse design of lattice materials.* arXiv preprint arXiv:2304.05422, 2023.
41. Zhang, C., et al., *Variant design generation and machine learning aided deformation prediction for auxetic metamaterials.* Mechanics of Materials, 2023. **181**: p. 104642.
42. Holdstein, Y., et al. *Volumetric texture synthesis of bone micro-structure as a base for scaffold design*. in *2009 IEEE international conference on shape modeling and applications*. 2009. IEEE.
43. Men, H., et al., *Robust topology optimization of three-dimensional photonic-crystal band-gap structures.* Optics express, 2014. **22**(19): p. 22632-22648.

44. Kench, S. and S.J. Cooper, *Generating 3D structures from a 2D slice with GAN-based dimensionality expansion.* arXiv preprint arXiv:2102.07708, 2021.
45. Zheng, X., et al., *Structure-dependent analysis of nanoporous metals: clues from mechanical, conduction, and flow properties.* The Journal of Physical Chemistry C, 2018. **122**(29): p. 16803-16809.
46. Xu, H., et al., *Descriptor-based methodology for statistical characterization and 3D reconstruction of microstructural materials.* Computational Materials Science, 2014. **85**: p. 206-216.
47. Makatura, L., et al., *Procedural metamaterials: A unified procedural graph for metamaterial design.* ACM Transactions on Graphics, 2023. **42**(5): p. 1-19.
48. Du, P., et al., *Microstructure design using graphs.* npj Computational Materials, 2018. **4**(1): p. 50.
49. Szabo, F., *The linear algebra survival guide: illustrated with Mathematica*. 2015: Academic Press.
50. Otair, D.M., *Approximate k-nearest neighbour based spatial clustering using kd tree.* arXiv preprint arXiv:1303.1951, 2013.
51. Kipf, T.N. and M. Welling, *Variational graph auto-encoders.* arXiv preprint arXiv:1611.07308, 2016.
52. Kingma, D.P. and M. Welling, *Auto-encoding variational bayes.* arXiv preprint arXiv:1312.6114, 2013.
53. Xu, L., et al., *Harnessing structural stochasticity in the computational discovery and design of microstructures.* Materials & Design, 2022. **223**: p. 111223.
54. Wang, L., et al., *Deep generative modeling for mechanistic-based learning and design of metamaterial systems.* Computer Methods in Applied Mechanics and Engineering, 2020. **372**: p. 113377.
55. Wang, Z. and H. Xu, *Manufacturability-aware deep generative design of 3D metamaterial units for additive manufacturing.* Structural and Multidisciplinary Optimization, 2024. **67**(2): p. 22.
56. Xu, L., K. Naghavi Khanghah, and H. Xu, *Designing Mixed-Category Stochastic Microstructures by Deep Generative Model-Based and Curvature Functional-Based Methods.* Journal of Mechanical Design, 2024. **146**(4).
57. Xu, L., K. Naghavi Khanghah, and H. Xu. *Design of Mixed-Category Stochastic Microstructures: A Comparison of Curvature Functional-Based and Deep Generative Model-Based Methods*. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2023. American Society of Mechanical Engineers.
58. Hamilton, W., Z. Ying, and J. Leskovec, *Inductive representation learning on large graphs.* Advances in neural information processing systems, 2017. **30**.
59. Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II.* IEEE transactions on evolutionary computation, 2002. **6**(2): p. 182-197.