

Supplementary Material

From RVE Data to Auxetic Design Rules: Interpretable Feature Analysis and Machine Learning-Based Modeling of Microstructured Materials

Alexander Hüls¹, Benjamin Alheit^{2,3} and Swantje Bargmann^{4,5,*}

- ¹ Faculty of Informatics and Natural Sciences, University of Applied Sciences South Westfalen, 58636 Iserlohn, Germany; alexander.huels@stud.fh-swf.de
² Department of BioMechanical Engineering, Faculty of Mechanical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands; b.h.alheit@tudelft.nl
³ Department of Materials Science and Engineering, Faculty of Mechanical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands
⁴ Chair of Solid Mechanics, School of Mechanical Engineering and Safety Engineering, University of Wuppertal, 42119 Wuppertal, Germany
⁵ Wuppertal Center for Smart Materials & Systems (CM@S), University of Wuppertal, 42119 Wuppertal, Germany
* Correspondence: bargmann@uni-wuppertal.de

1. Computation of effective properties

The macroscopic response of each RVE is characterized within the framework of small-strain, linear elasticity. The microscopic displacement field $\mathbf{u}(\mathbf{x})$ gives rise to the strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$ and the Cauchy stress tensor $\boldsymbol{\sigma}$ is related to the strain by homogenized Hooke's law $\boldsymbol{\sigma} = \mathbf{C}^* : \boldsymbol{\varepsilon}$ where \mathbf{C}^* is the fourth-order effective stiffness tensor of the homogenized, plane-stress medium. In 2D, this relation can be expressed in Voigt notation as

$$\begin{bmatrix} \sigma_{M11} \\ \sigma_{M22} \\ \sigma_{M12} \end{bmatrix} = \mathbf{C}^* \cdot \begin{bmatrix} \varepsilon_{M11} \\ \varepsilon_{M22} \\ \gamma_{12} \end{bmatrix} \quad (1)$$

where $\gamma_{12} = 2\varepsilon_{M12}$ denotes the engineering shear strain and “M” indicates volume-averaged (macroscopic) quantities. The macroscopic stress $\boldsymbol{\sigma}_M = \frac{1}{V} \sum_e \boldsymbol{\sigma}^{(e)} V^{(e)}$ and strain $\boldsymbol{\varepsilon}_M = \frac{1}{V} \sum_e \boldsymbol{\varepsilon}^{(e)} V^{(e)}$ are computed by volume-averaging over the RVE, using the integration point volumes as weights, where $V^{(e)}$ denotes the volume associated with element e and $V = \sum_e V^{(e)}$ is the total RVE volume. Under periodic boundary conditions, these volume averages coincide with the macroscopic stress and stress tensors of the homogenized material.

The prescribed displacements produce a nominal unit strain in x -direction $\varepsilon_{M11}^{(1)} \approx 1$, $\varepsilon_{M22}^{(1)} \approx 0$, $\gamma_{12}^{(1)} \approx 0$ and the resulting macroscopic stress tensor $\boldsymbol{\sigma}_M^{(1)}$ provides the stress response. A nominal unit strain in y -direction is generated via $\varepsilon_{M11}^{(2)} \approx 0$, $\varepsilon_{M22}^{(2)} \approx 1$, $\gamma_{12}^{(2)} \approx 0$. In the shear step, the control-node displacements are chosen such that the RVE undergoes simple shear with engineering shear strain $\gamma_{12}^{(3)}$. In the post-processing, the volume-averaged shear stress $\sigma_{M12}^{(3)}$ is divided by two before assembly of \mathbf{C}^* , which is equivalent to using $\gamma_{12}^{(3)} = 2\varepsilon_{M12}^{(3)}$ as the third generalized strain component. In compact form, the three macroscopic stress and strain states can be collected in the matrices $\boldsymbol{\Sigma} = [\boldsymbol{\sigma}_M^{(1)} \boldsymbol{\sigma}_M^{(2)} \boldsymbol{\sigma}_M^{(3)}]$ and $\mathbf{E} = [\boldsymbol{\varepsilon}_M^{(1)} \boldsymbol{\varepsilon}_M^{(2)} \boldsymbol{\varepsilon}_M^{(3)}]$, with

$$\boldsymbol{\sigma}_M^{(k)} = \begin{bmatrix} \sigma_{M11}^{(k)} \\ \sigma_{M22}^{(k)} \\ \sigma_{M12}^{(k)} \end{bmatrix}, \quad \boldsymbol{\varepsilon}_M^{(k)} = \begin{bmatrix} \varepsilon_{M11}^{(k)} \\ \varepsilon_{M22}^{(k)} \\ \gamma_{12}^{(k)} \end{bmatrix}. \quad (2)$$

At the macroscopic level this implies $\Sigma = \mathbf{C}^* : \mathbf{E} \Rightarrow \mathbf{C}^* = \Sigma : \mathbf{E}^{-1}$. Because the loading steps are chosen such that $\varepsilon_M^{(1)} \approx (1,0,0)^T$, $\varepsilon_M^{(2)} \approx (0,1,0)^T$ and $\varepsilon_M^{(3)} \approx (0,0,\gamma_{12}^{(3)})^T$, the matrix \mathbf{E} is close to the identity and its inverse is well approximated by a simple normalization of the shear response. Using the engineering shear strain and normalizing the shear step, the effective stiffness matrix is assembled directly from the macroscopic stresses as

$$\mathbf{C}^* = \begin{bmatrix} \sigma_{M11}^{(1)} & \sigma_{M11}^{(2)} & \sigma_{M11}^{(3)}/2 \\ \sigma_{M22}^{(1)} & \sigma_{M22}^{(2)} & \sigma_{M22}^{(3)}/2 \\ \sigma_{M12}^{(1)} & \sigma_{M12}^{(2)} & \sigma_{M12}^{(3)}/2 \end{bmatrix} \quad (3)$$

which corresponds to the effective plane-stress stiffness relating $(\varepsilon_{M11}, \varepsilon_{M22}, \gamma_{12})^T$ to $(\sigma_{11}, \sigma_{22}, \sigma_{12})^T$. From \mathbf{C}^* , we extract the scalar effective properties: the directional Young's moduli $E_{22}^{\text{eff}} = \frac{\sigma_{M22}^{(2)}}{\varepsilon_{M22}^{(2)}}$, Poisson's ratios $\nu_{12}^{\text{eff}} = -\frac{\varepsilon_{M11}^{(2)}}{\varepsilon_{M22}^{(2)}}$ and the effective shear modulus $G_{12}^{\text{eff}} = \frac{\sigma_{M12}^{(3)}}{\gamma_{12}^{(3)}}$. The displacements are chosen such that $\varepsilon_{M11}^{(1)}$ and $\varepsilon_{M22}^{(2)}$ are close to unity in the corresponding loading steps, so that the columns of \mathbf{C}^* directly approximate the stress response to unit macroscopic strains. The resulting plane-stress stiffness tensor is additionally stored in contracted Voigt form: \mathbf{C}^* is represented as a 3×3 matrix $C_{\alpha\beta}$ with $\alpha, \beta \in \{11, 22, 12\}$, consistent with the symmetric in-plane strain tensor. The corresponding compliance tensor $\mathbf{S}^* = (\mathbf{C}^*)^{-1}$ is used to evaluate directional effective properties beyond the principal axes.

The microstructures feature strongly oriented nanorod arrangements and exhibit pronounced direction-dependent stiffnesses (Fig. 6). To characterize this behavior, we consider uniaxial macroscopic loading along an axis that is rotated by an angle θ with respect to the global x -axis. Rather than performing additional finite element simulations for each orientation, we exploit the tensorial transformation properties of \mathbf{S}^* . A 3×3 rotation matrix $\mathbf{R}(\theta)$ is constructed from the standard 2×2 in-plane rotation, acting on the index pairs (11), (22) and (12). The compliance tensor in the rotated coordinate system then follows as $\mathbf{S}^*(\theta) = \mathbf{R}(\theta)^T \cdot \mathbf{S}^* \cdot \mathbf{R}(\theta)$.

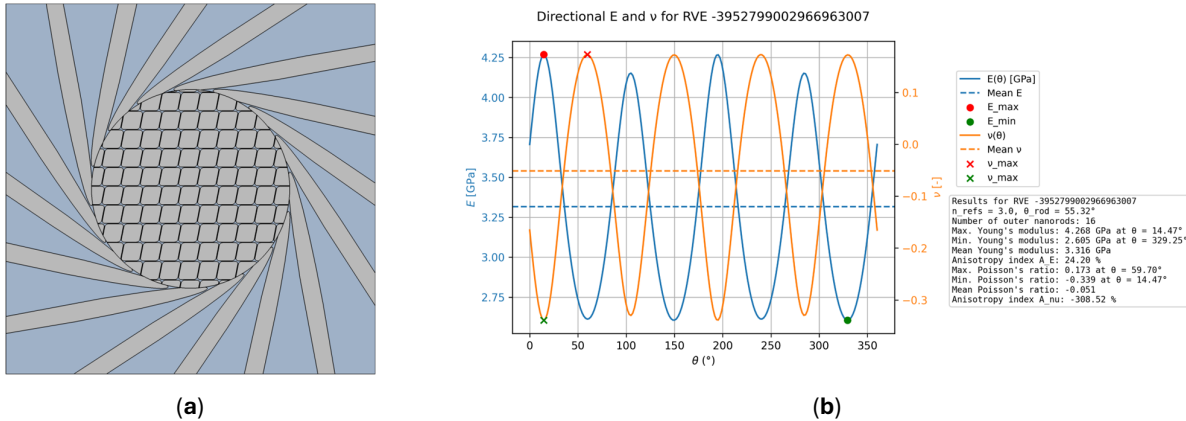


Figure S1. Example RVE. (a) Sketch of one example RVE; **(b)** Directional effective Young's modulus $E(\theta)$ and Poisson's ratio $\nu(\theta)$ for that RVE

For a uniaxial macroscopic stress state aligned with the rotated x_1 -axis, the corresponding directional Young's modulus $E(\theta) = \frac{1}{S_{11}^*(\theta)}$ and Poisson's ratio $\nu(\theta) = -\frac{S_{21}^*(\theta)}{S_{11}^*(\theta)}$ are obtained from $\mathbf{S}^*(\theta)$.

The orientation-dependent effective Young's modulus $E(\theta)$ and Poisson's ratio $\nu(\theta)$ are computed numerically over a dense set of in-plane loading directions $\theta \in [0, 2\pi]$. For each RVE, these directional responses are condensed into a small set of scalar descriptors: the minimum, maximum, and mean values of $E(\theta)$ and $\nu(\theta)$.

In-plane behavior is modeled under plane-stress conditions using four-node and three-node reduced-integration elements.

2. Feature representation and ML-specific preprocessing

The starting point is the table described in Section 3.3, in which each row represents one specific RVE and each column a scalar descriptor. These columns are assigned clear roles: The meta columns “id” and “campaign” are kept only for bookkeeping and are never used as inputs. The binary auxetic indicator “is_aux” serve as classification targets; “is_aux” is defined as 1 if $v_{min} < 0$ and 0 otherwise. Effective elastic quantities - E_{min} , E_{mean} , E_{max} , v_{min} , v_{mean} , v_{max} - as well as inclusion volume-fraction measures ($V_{f,inc}$, $V_{f,inc,centre}$, $V_{f,inc,outer}$) are treated as target variables for the surrogate models.

Angular quantities θ_{rhom} and θ_{rod} are converted from degrees to radians and encoded by their sine and cosine. The original angle columns are kept for information, not for usage in ML modelling, so that the models only see $\cos(\theta_{rhom})$, $\sin(\theta_{rhom})$, $\cos(\theta_{rod})$, $\sin(\theta_{rod})$. This circular encoding respects the periodic nature of the angles and avoids artificial jumps at 0° and 360°. Boolean indicators such as $connected_{outer_centre}$ are mapped to 0/1.

After removing metadata, labels, effective outputs and microstructure outputs from the candidate pool, and after handling angular variables separately, the remaining numerical descriptors form the core input feature set. The continuous input features are C_E , E_{mat} , $fillet_{rhom}$, l_{rod} , r_{centre} , $ref_{position}$, S_{rhom} , v_{inc} and v_{mat} . The refinement level n_{refs} is treated as a categorical feature. Parameters that are constant throughout the dataset, such as rve_size , E_{inc} and the fixed geometric dimensions (l_{rhom} , w_{rod}), are not used for ML model learning.

Before training, the inputs are passed through a feature-wise preprocessing pipeline. For continuous features, missing values are imputed by the median and the resulting columns are standardized to zero mean and unit variance. The categorical feature n_{refs} is processed by a separate pipeline that performs most-frequent imputation followed by one-hot encoding. Both pipelines are combined in a column transformer that drops all remaining columns, producing a processed feature matrix of size 30004×18. The additional four columns correspond to one-hot encodings of the observed refinement levels (proc_n_refs_1 – proc_n_refs_4). The same preprocessing is used for all ML model data.

3. Supervised binary classification of auxetic behavior

A first round of supervised machine learning was conducted using a binary auxetic vs non-auxetic classification setup with a Random Forest (RF) classifier and a fully connected neural network (FCNN) classifier.

To address the class imbalance, imbalance handling was applied via label-derived class weights computed on the training split only. Based on the training labels, the resulting class weights were $w_0 = 0.513$ for the non-auxetic class and $w_1 = 19.567$ for the auxetic class, corresponding to an effective ratio of $w_1/w_0 \approx 38.13$. This reweighting increases sensitivity to the rare auxetic class during optimization.

For the initial binary classification, we used a three-way split (train/validation/test = 60%/20%/20%) following common recommendations to retain an independent test set. The split is stratified with respect to the auxetic indicator to preserve the rare-class fraction. For the subsequent regression surrogates and baseline comparisons, we adopt the split strategy reported in the main paper (Section 5).

1. Random forest classifier:

We train a RF classifier with 400 trees and balanced class weights. Default splitting criteria and unconstrained depth are used to allow expressive decision rules in the heterogeneous descriptor set. Beyond predictive performance, the fitted random forest is used to derive a parameter-importance ranking for auxetic classification. The model provides impurity-based feature-importance scores for the 18 processed input columns. Let $j = \{1, \dots, 18\}$ denote these processed columns and let I_j^{RF} be the importance assigned to column j by the fitted weighted RF classifier. To obtain importances at the level

of the underlying raw design variables, encoded columns belonging to the same physical parameter are consolidated by summation, i.e.,

$$I_f = \sum_{j \in G(f)} I_j^{RF},$$

where $G(f)$ denotes the set of processed columns representing raw variable f . This applies in particular to the sine/cosine encodings of θ_{rod} and θ_{rhomb} , and to the one-hot encoded columns of n_{refs} . For variables represented by a single processed column, $I_f = I_j^{RF}$. The final parameter ranking is then obtained by sorting the consolidated raw-variable importances I_f in descending order. These consolidated scores are subsequently used to interpret which design descriptors are most strongly associated with auxetic classification and to guide the focused Monte Carlo sampling described below.

2. Fully connected neural network classifier:

We train a FCNN that takes the 18 processed inputs and passes them through three dense hidden layers (32–128–32) with ReLU activations. Dropout regularization is applied after each hidden layer, and L2 weight regularization is used to mitigate overfitting. The output layer consists of a single sigmoid neuron producing the auxetic probability. Training uses the Adam optimizer with an exponentially decaying learning-rate schedule. To prevent overfitting and to select a well-generalizing model, early stopping and model checkpointing are applied based on validation loss. Class predictions are obtained by thresholding the predicted probability at 0.5.

Python code: https://github.com/AlexanderHuels/ml_limpet_2d_rve_data.git

4. ML-guided design-space exploration and dataset refinement

Design parameter space

The design vector \mathbf{z} comprises only direct geometric and material inputs specified prior to meshing and FEM (Sections 3.1–3.2). Fixed constants (e.g., RVE size and rhomboid dimensions) are not treated as design variables. For each continuous parameter, lower and upper bounds are derived from the empirical distribution of the cleaned dataset using central quantiles (typically 5th–95th percentiles) with a small safety margin. The discrete refinement level n_{refs} is restricted to the values observed in the original simulations (1–4). Together, this defines a data-informed hyper-rectangular design domain for continuous variables and a finite admissible set for discrete ones.

Importance- and direction-weighted focused sampling

Uniform Monte Carlo sampling over this domain is inefficient because auxetic realizations are rare. We therefore use a focused sampler that increases sampling density in empirically “promising” directions while keeping all candidates inside the FEM-proven bounds.

From the random-forest classification analysis, each raw design variable f is assigned (i) a non-negative importance score I_f (magnitude of contribution to auxetic discrimination), and (ii) a qualitative direction label $d_f \in \{\text{high}, \text{low}, \text{none}\}$. The direction label is inferred from the sign-consistent trend between auxetic vs non-auxetic subgroups (difference of subgroup means) and the correlation sign with the auxetic indicator; if no consistent tendency is observed, $d_f = \text{none}$.

Candidate vectors \mathbf{z} are generated as follows:

1. Select a small set of “focus variables”. For each candidate, we draw $k \in \{1, 2, 3\}$ variables to bias (typically one to three) using sampling probabilities proportional to I_f^α (with $\alpha > 0$ controlling how strongly the sampler concentrates on high-importance variables).
2. Sample focus variables from directional tails. For each selected variable f :
 - if $d_f = \text{high}$, sample f from the upper tail of its admissible interval,
 - if $d_f = \text{low}$, sample f from the lower tail,
 - if $d_f = \text{none}$, sample uniformly.

In practice, “tail” means restricting the draw to a fixed upper/lower fraction of the interval (rather than changing the bounds), so the candidate remains admissible by construction.

3. Sample all remaining variables uniformly over their admissible ranges to preserve broad coverage of the design domain. Discrete variables (e.g., n_{refs}) are handled analogously by favoring larger/smaller admissible values when a direction label exists, and sampling uniformly otherwise.

This mechanism is intentionally simple: it injects the learned importance structure only as a lightweight bias (few variables, tail sampling) rather than as a deterministic rule, so exploration remains diverse while being substantially more “auxetic-aware” than uniform scanning.

Surrogate screening and FEM loop closure

All focused candidates are passed through the fixed preprocessing pipeline (Section 2). They are then screened by the two binary classifiers (RF and FCNN) and ranked using the ensemble auxetic score defined in the main paper (Eq. (1)). Only high-score candidates are retained and exported as Abaqus-ready inputs and simulated. The resulting FEM outputs are merged into the dataset, enabling an iterative refinement loop in which additional simulations densify the data specifically in auxetic-relevant regions (see main paper for thresholds, sample counts, and dataset-level outcomes).

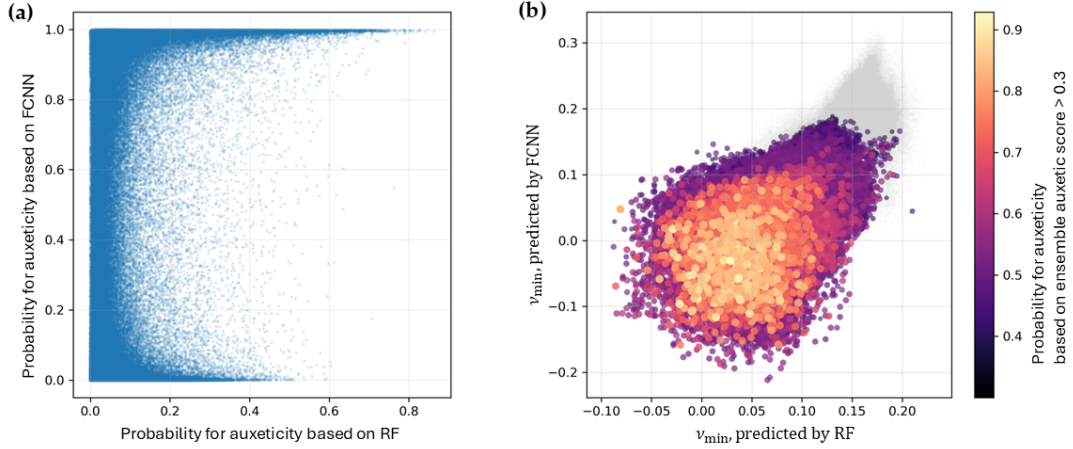


Figure S2. Design domain after Monte Carlo sampling. Monte Carlo-generated candidates screened and scored by RF and FCNN classification. (a) All candidates with probabilities to be auxetic, (b) predicted Poisson's ratio for those candidates

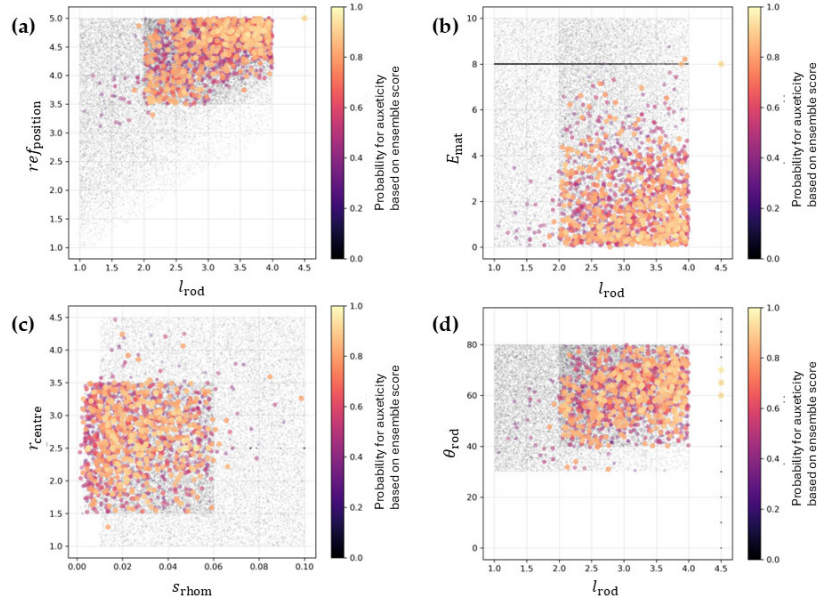


Figure S3. Potential designs after Monte Carlo sampling from the perspective of specific feature combination. Features of Monte Carlo-generated candidates screened (a) Position $ref_{position}$ vs. half-length l_{rod} of outer inclusions, (b) Young's modulus matrix E_{mat} vs. half-length l_{rod} of outer inclusions, (c) Radius vs. Spacing s_{rhom} of central inclusions, (d) Orientation angle θ_{rod} vs. half-length l_{rod} of outer inclusions

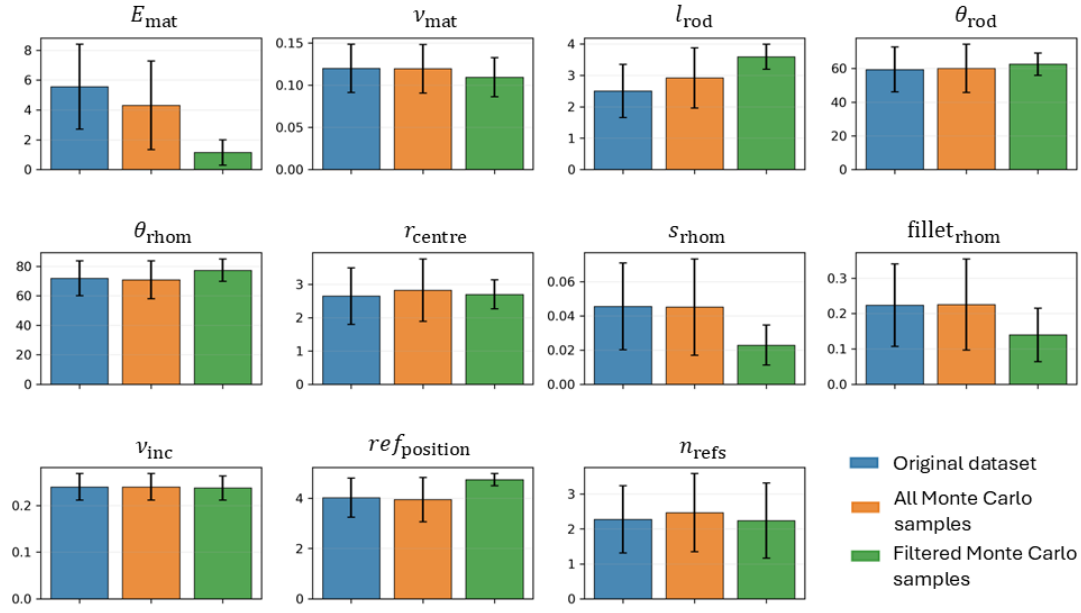


Figure S4. Effect of focused Monte Carlo sampling on design-variable distributions. Mean \pm standard deviation of each design variable for the original simulation dataset, the Monte Carlo-generated candidates, and the retained high-score subset. The figure qualitatively verifies that focused sampling modifies sampling density while remaining within the prescribed admissible bounds.

5. K Means Downsampling

Purpose: Reduce the non-auxetic subset to a target size while preserving coverage of the non-auxetic design space.

Input subset: only the non-auxetic RVEs from tabular database.

Target auxetic fraction $f_{\text{aux,target}} = 40\%$

Target auxetic fraction $n_{\text{target}} = \text{round}\left(n_{\text{aux}} \frac{1-f_{\text{aux,target}}}{f_{\text{aux,target}}}\right)$ where $n_{\text{aux}} = 5487$ (implemented in compute_nonaux_target). Here: $n_{\text{target}} = \text{round}\left(5487 \times \frac{0.6}{0.4}\right) = 8230$.

Feature set:

Table S1. Summary statistics for the original (unstandardized) feature values (all numeric, in total 29929 samples). All features are z-score normalized (zero mean, unit variance) using StandardScaler within the preprocessing pipeline.

FEATURES	MEAN	STD	MIN	25%	50%	75%	MAX
v_{inc}	0.24	0.028	1.90E-01	0.215	0.24	0.264	0.29
E_{mat}	5.514	2.867	1.00E-03	2.983	6.075	8	9.999
v_{mat}	0.12	0.029	7.00E-02	0.095	0.12	0.144	0.17
r_{centre}	2.642	0.848	1.00E+00	1.967	2.585	3.256	4.5
l_{rhom}	0.5	0	5.00E-01	0.5	0.5	0.5	0.5
$\text{fillet}_{\text{rhom}}$	0.224	0.117	0.00E+00	0.134	0.217	0.3	0.5
l_{rod}	2.507	0.844	1.00E+00	1.877	2.48	3.133	4.5
n_{refs}	2.275	0.959	1.00E+00	2	2	3	4
$\text{ref}_{\text{position}}$	4.028	0.776	1.01E+00	3.635	4.185	4.639	5.077
$\text{connected}_{\text{outer_centre}}$	0.864	0.343	0.00E+00	1	1	1	1
$\sin(\theta_{\text{rhom}})$	0.931	0.073	5.00E-01	0.895	0.955	0.989	1
$\cos(\theta_{\text{rhom}})$	0.304	0.19	-8.66E-01	0.146	0.297	0.445	0.866
$\sin(\theta_{\text{rod}})$	0.838	0.127	0.00E+00	0.761	0.871	0.943	1
$\cos(\theta_{\text{rod}})$	0.495	0.191	0.00E+00	0.334	0.492	0.649	1

Clustering algorithm: MiniBatch KMeans with fixed seed (random_state=42).

Number of clusters: $K = \min(K_{\max}, n_{\text{target}}, N_{\text{nonaux}})$, $K_{\max} = 3000$.

Steps:

1. Cluster sizes and weights:

KMeans assigns each non-auxetic RVE to exactly one cluster i . The cluster populations $|C_i|$ are counted and converted to weights $w_i = \frac{|C_i|}{N_{\text{nonaux}}}$, $\sum_i w_i = 1$.

N_{nonaux} : total number of non-auxetic RVEs (here 29,929)

2. Initial proportional quota (floor allocation):

The target number of retained non-auxetic RVEs is n_{target} . Each cluster is assigned an initial integer quota $m_i^{(0)} = \lfloor w_i n_{\text{target}} \rfloor$.

n_{target} : desired number of non-auxetic RVEs after downsampling (here 8230).

m_i : number of representatives selected from cluster i .

3. Remainder distribution (largest remainder method or Hamilton's method):

Deficit caused by flooring $D = n_{\text{target}} - \sum_i m_i^{(0)}$.

Fractional parts $r_i = w_i n_{\text{target}} - \lfloor w_i n_{\text{target}} \rfloor$.

Add one additional slot to the D clusters with the largest r_i .

After adjustment: $\sum_i m_i = n_{\text{target}}$

4. Feasibility cap:

Enforce $m_i \leq |C_i|$, preventing assignment of more representatives than a cluster contains.

5. Representative selection within clusters:

For each cluster i , select the m_i RVEs with the smallest squared Euclidean distance to the cluster centroid in the standardized feature space.

Discard the remaining cluster members.

Python code: https://github.com/AlexanderHuels/ml_limpet_2d_rve_data.git

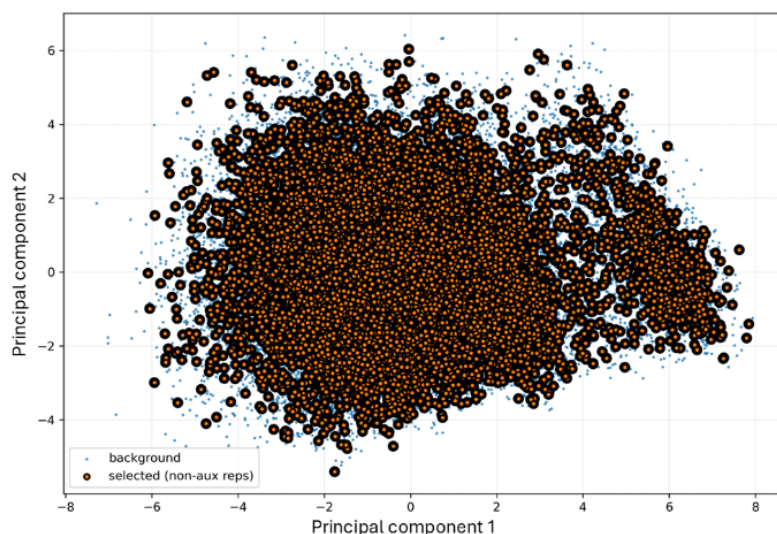


Figure S5. PCA projection of the non-auxetic design space with selected K-means representatives.

6. FCNN regressor

Training objective and reported validation metrics

The FCNN is trained with the mean weighted Huber loss defined in the main manuscript. Let \tilde{y}_{ij} and \hat{y}_{ij} denote the standardized ground-truth and predicted values for validation sample i and target component j , and let $y_{ij} = \sigma_j \tilde{y}_{ij} + \mu_j$, $\hat{y}_{ij} = \sigma_j \hat{\tilde{y}}_{ij} + \mu_j$ be the corresponding values in the original target space, with target-specific mean μ_j and standard deviation σ_j .

For the hyperparameter studies in Sect. 8, model comparison relies primarily on weighted MAE and RMSE evaluated after inverse transformation to the original target space. Using target-wise weights w_j and sample-wise auxetic weights a_i , with

$$a_i = \begin{cases} \alpha_{\text{aux}}, & \text{if sample } i \text{ is auxetic,} \\ 1.0, & \text{otherwise,} \end{cases}$$

the aggregated weighted validation metrics are

$$\text{MAE}_{\text{phys,weighted}} = \frac{\sum_{i=1}^N \sum_{j=1}^9 a_i w_j |y_{ij} - \hat{y}_{ij}|}{\sum_{i=1}^N \sum_{j=1}^9 a_i w_j},$$

$$\text{RMSE}_{\text{phys,weighted}} = \left(\frac{\sum_{i=1}^N \sum_{j=1}^9 a_i w_j (y_{ij} - \hat{y}_{ij})^2}{\sum_{i=1}^N \sum_{j=1}^9 a_i w_j} \right)^{1/2}.$$

Thus, the reported weighted MAE and RMSE are not percentage errors. Rather, they are weighted composite error measures in the original target space, aggregated across targets with different units and scales. Accordingly, within a fixed weighting scheme, lower weighted MAE/RMSE indicates better overall predictive performance.

Python source code: https://github.com/AlexanderHuels/ml_limpet_2d_rve_data.git

7. Data partitioning and split robustness

A fixed stratified hold-out test subset comprising 7.5% of the 13,717 RVEs is set aside once using random seed 42. All model development and model selection steps are then carried out exclusively within the remaining train/validation pool. In the split-sensitivity study, the FCNN training setup is kept fixed and only the train/validation/test ratios and train/validation split seeds are varied. This analysis serves to validate the data partitioning and model selection protocol rather than to tune FCNN hyperparameters.

To assess split robustness, we consider three train/validation/test ratios, (70%, 22.5%, 7.5%), (75%, 17.5%, 7.5%), and (80%, 12.5%, 7.5%), combined with train/validation split seeds 7, 21, 42, 63, 84, 101, 123, 202, 555, 777, and 999. Model comparison is based primarily on weighted validation MAE and RMSE in physical target space, while weighted validation R^2 is used only as a complementary measure. We prioritize weighted validation metrics because model development in this study intentionally emphasizes reliable performance in the auxetic regime rather than minimizing an entirely uniform average error across all samples. As summarized in Fig. S6, the validation metrics vary only moderately across the tested partitionings.

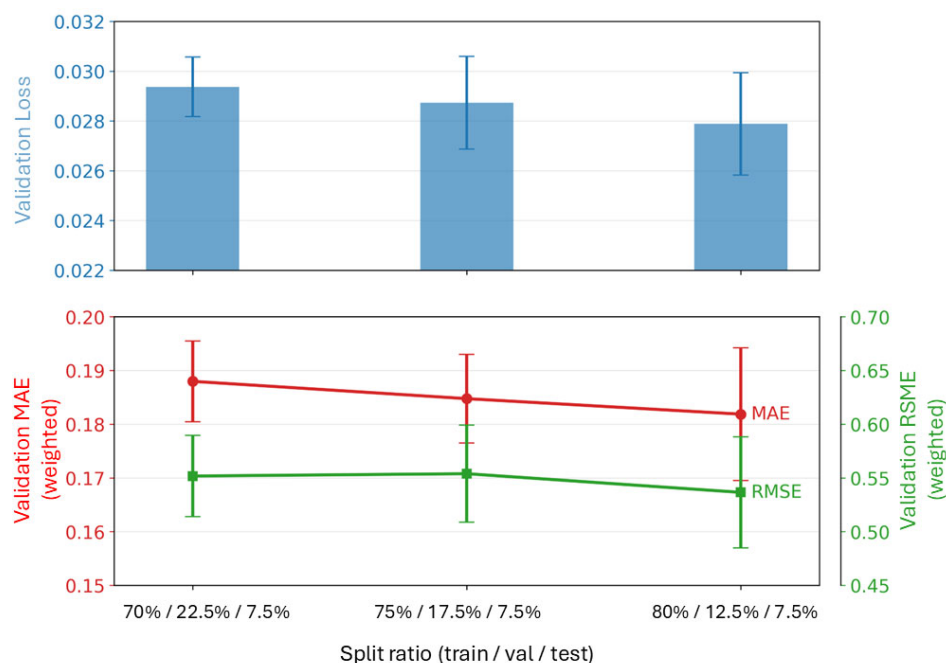


Figure S6. Split-ratio robustness analysis for the retained reference FCNN. Mean validation metrics aggregated over the tested train/validation split seeds; error bars indicate the corresponding variability across seeds. Top: validation loss used for checkpointing and early stopping. Bottom: weighted validation MAE and RMSE in the original target space.

The 80% / 12.5% / 7.5% split shows the most favorable mean validation errors overall and is therefore retained for the subsequent sensitivity studies. Within this protocol, varying the train/validation split seed does not change the qualitative conclusions. Among the examined train/validation seeds, seed 101 is retained for the follow-up studies because it yields representative validation performance within the tested range, rather than corresponding to either a particularly favorable or an unfavorable partition.

8. Hyperparameter studies

- All subsequent FCNN trainings use the 80% / 12.5% / 7.5% split defined in Sect. 7.
- The hyperparameter analysis is performed in a staged, validation-based manner rather than as a single exhaustive grid search.
- Model comparison is based primarily on weighted validation MAE and RMSE in the original target space and secondarily on validation loss.
- Unless stated otherwise, the base configuration uses hidden layers 256-256-128, dropout = 0.08, $L2 = 10^{-4}$, GELU activation, learning rate = $2 \cdot 10^{-4}$, batch size = 64, Huber $\delta = 1.0$, auxetic weighting factor = 1.5, and the v -target weighting 1.0, 1.0, 1.0, 1.3, 1.6, 1.6, 0.8, 0.8, 0.8.

Stage 1: architecture and regularization

The first stage varies hidden-layer architecture, dropout, and L2 regularization.

The screened architectures are 256-256-128, 512-256-128, 512-256-256, 512-512-256, 512-512-512, 512-256-128-64, 512-256-128-64-32, and 1024-512-256-128-64. Dropout rates of 0.02, 0.05, 0.08, 0.12, and 0.2 are combined with L2 of 10^{-5} , $5 \cdot 10^{-5}$, 10^{-4} .

Stage 1 does not identify a single uniformly best architecture across all validation metrics.

The anchor model 256-256-128 remains a relevant reference configuration.

Among the wider three-layer models, 512-256-256 yields the most favorable combination of validation loss, weighted validation MAE, and weighted validation RMSE across the most competitive screened settings (cf. Figs. S7-8).

Architectures with more than three hidden layers perform well in selected cases, but they do not improve the overall validation picture consistently enough to justify their higher complexity in the present follow-up analysis.

The subsequent stage therefore focuses on three candidates:

- the reference model 256-256-128, dropout = 0.08, $L_2 = 10^{-4}$,
- the wider candidate 512-256-128, dropout = 0.12, $L_2 = 10^{-5}$, and
- the wider follow-up candidate 512-256-256, dropout = 0.12, $L_2 = 10^{-5}$.

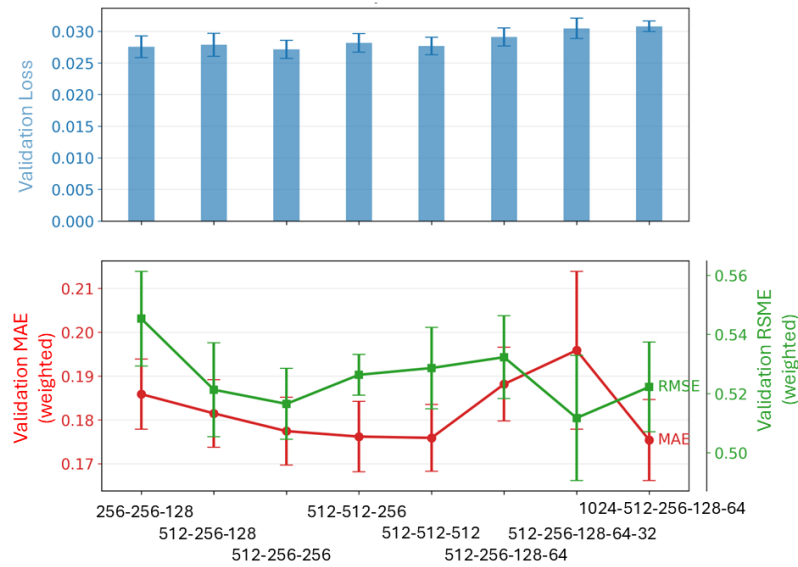


Figure S7. Architecture-and-regularization study. Top: mean validation loss aggregated by hidden-layer architecture. Bottom: mean weighted validation MAE and RMSE in physical target space. Error bars indicate variability across the corresponding regularization settings.

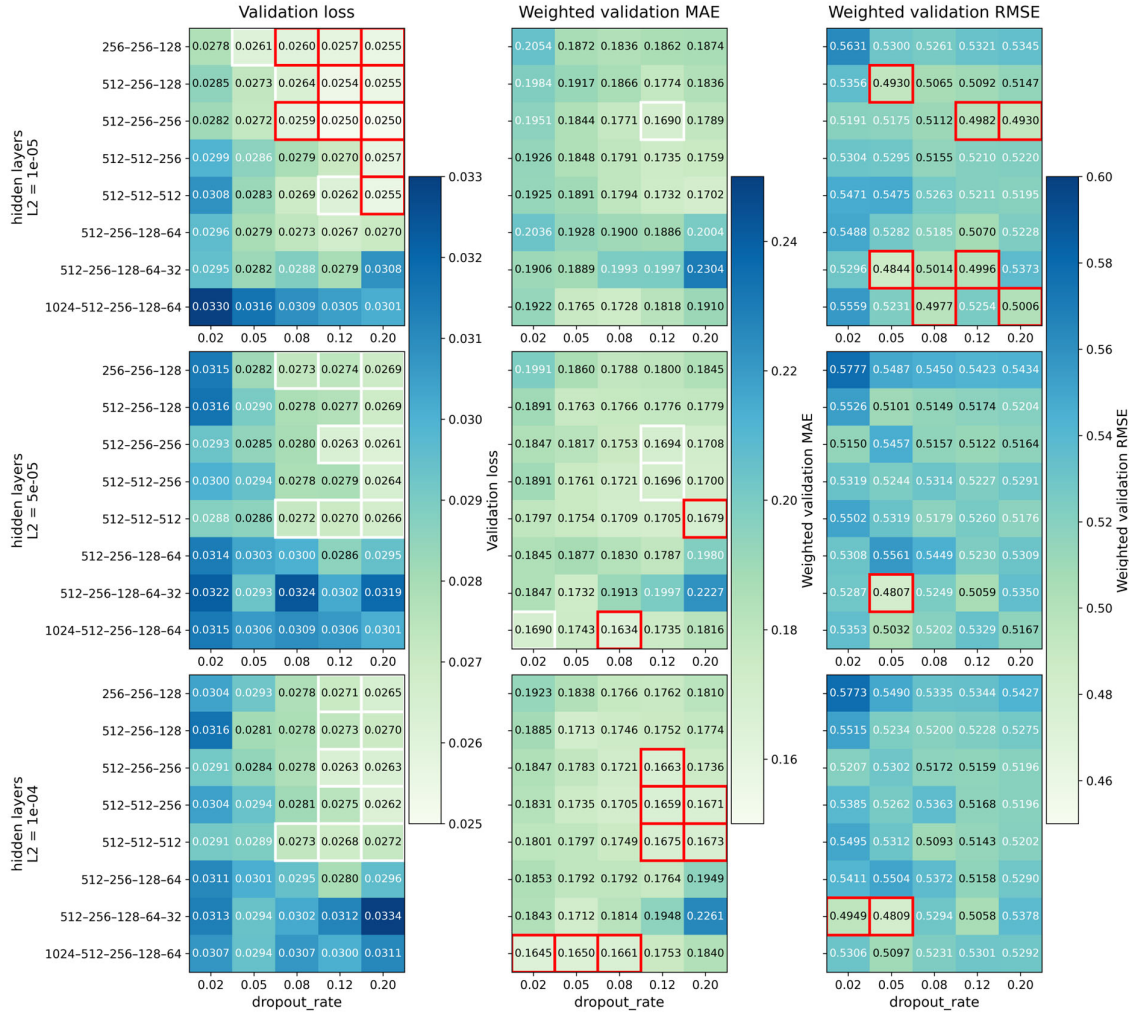


Figure S8. Heatmap summary of the architecture-and-regularization study (Stage 1). Rows denote hidden-layer architectures and columns denote dropout rates. Panels are grouped by L2 regularization level (10^{-4} , $5 \cdot 10^{-5}$, 10^{-5}) and by metric, showing mean validation loss, mean weighted validation MAE, and mean weighted validation RMSE. Red outlines highlight the most favorable parameter combinations by metric.

Stage 2: optimization settings

The second stage varies activation function, learning rate, and batch size for the three shortlisted architectures.

GELU and ReLU are tested with learning rates of 10^{-4} , $2 \cdot 10^{-4}$, and $5 \cdot 10^{-4}$, and batch sizes of 32, 64, and 128.

Across the grid, GELU gives the more favorable overall weighted validation metrics than ReLU, especially for the wider networks, while the most competitive settings cluster at $2 \cdot 10^{-4}$ and $5 \cdot 10^{-4}$.

The reference architecture 256-256-128 improves substantially at the higher learning rates, with its strongest setting at GELU / $5 \cdot 10^{-4}$ / batch size 64, but it remains inferior to the wider candidates. The architecture 512-256-128 reaches the lowest validation loss at GELU / $5 \cdot 10^{-4}$ / batch size 32, yet 512-256-256 with GELU provides the best overall trade-off across validation loss, weighted validation MAE, and weighted validation RMSE.

In particular, 512-256-256 / GELU / 2×10^{-4} / batch size 64 combines near-minimal validation loss and MAE with the lowest weighted validation RMSE (cf. Fig. S9).

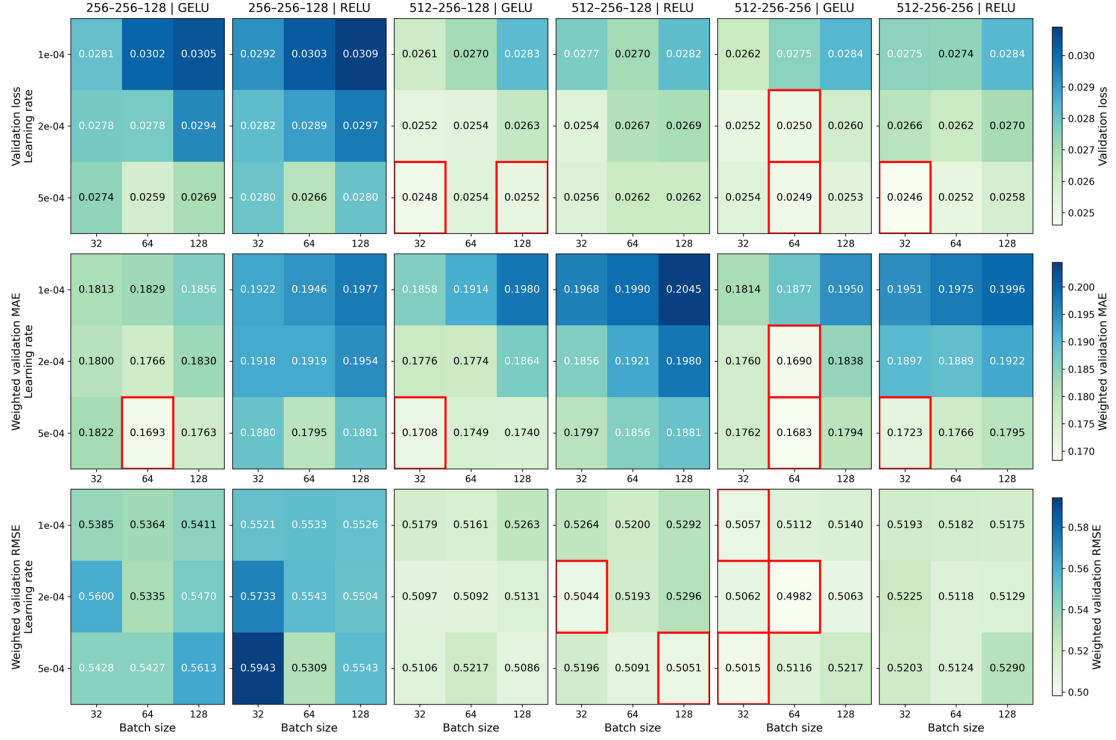


Figure S9. Heatmap summary of the optimization study for the shortlisted FCNN architectures (Stage 2). The two architecture blocks compare GELU and ReLU activation, while the three metric rows report mean validation loss, mean weighted validation MAE, and mean weighted validation RMSE. In each panel, rows denote learning rate and columns denote batch size. Red outlines highlight the most favorable parameter combinations by metric.

Stage 3: loss parameters and sample weighting

The third stage varies the Huber transition threshold $\delta \in \{0.5, 1.0, 1.5\}$ and the auxetic weighting factor $\in \{1.0, 1.5, 2.0\}$ applied to auxetic samples.

To assess whether the retained loss-weighting choice depends materially on the selected architecture, the Stage 3 grid is evaluated for the three shortlisted FCNN architectures 256-256-128, 512-256-128, and 512-256-256, each under its retained Stage 2 optimization settings.

Across all three architectures, lower δ values and stronger auxetic weighting tend to give the most favorable weighted validation metrics.

Among the examined configurations, the architecture 512-256-256 combined with $\delta = 0.5$ and aux factor = 2.0 provides the most favorable overall weighted validation performance (cf. Fig. S10)



Figure S10. Heatmap summary of the loss-parameter and sample-weighting study (Stage 3). Rows denote the three shortlisted FCNN architectures, and columns denote the reported validation metrics: mean validation loss, mean weighted validation MAE, and mean weighted validation RMSE. Within each panel, rows correspond to the Huber transition threshold δ and columns to the auxiliary weighting factor. Red outlines highlight the most favorable parameter combinations by metric.

Stage 4: ν -target weighting sensitivity

The final stage examines sensitivity to alternative ν -target weighting presets while jointly rechecking the auxetic weighting factor. These presets are compared:

ν -low: (1.0, 1.0, 1.0, 1.1, 1.3, 1.3, 0.8, 0.8, 0.8),

ν -base: (1.0, 1.0, 1.0, 1.3, 1.6, 1.6, 0.8, 0.8, 0.8), and

ν -high: (1.0, 1.0, 1.0, 1.5, 1.8, 1.8, 0.8, 0.8, 0.8),

each evaluated for auxiliary weighting factors of 1.5 and 2.0.

Across all settings, the qualitative conclusions from the previous stages remain unchanged. The architecture 512-256-256 consistently provides the most favorable validation behavior, followed by 512-256-128, whereas 256-256-128 remains clearly inferior. The comparison of ν -target weight presets shows only moderate differences within the retained architecture. While ν -low gives the lowest validation loss and weighted validation MAE, ν -base remains highly competitive and yields the lowest weighted validation RMSE. Thus, the tested ν -target weight variations do not alter the overall model ranking but mainly shift the balance between average error and sensitivity to larger deviations.

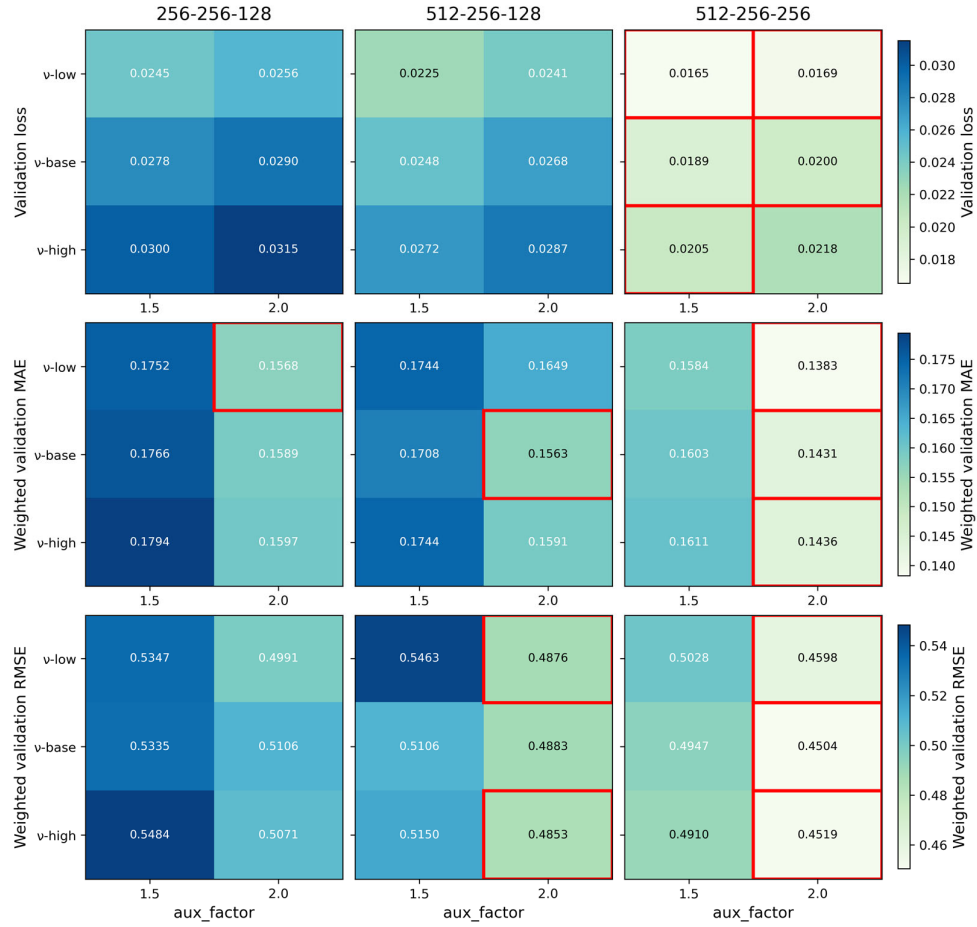


Figure S11. Heatmap summary of v -target weighting sensitivity study (Stage 4). Columns show the hidden-layer architectures, rows show validation loss, weighted validation MAE, and weighted validation RMSE. Within each panel, the horizontal axis indicates the auxetic weighting factor and the vertical axis the v -target weighting preset. Red outlines highlight the most favorable parameter combinations by metric.

Retained final FCNN configuration

The staged study yields the following final FCNN configuration:

- hidden layers: 512-256-256,
- dropout 0.12,
- L2 regularization 10^{-5} ,
- GELU activation,
- learning rate $2 \cdot 10^{-4}$,
- batch size 64,
- Huber $\delta = 0.5$, and
- auxetic weighting factor 2.0,
- the v -base target weighting (1.0, 1.0, 1.0, 1.3, 1.6, 1.6, 0.8, 0.8, 0.8)

as a balanced final choice.

9. Results

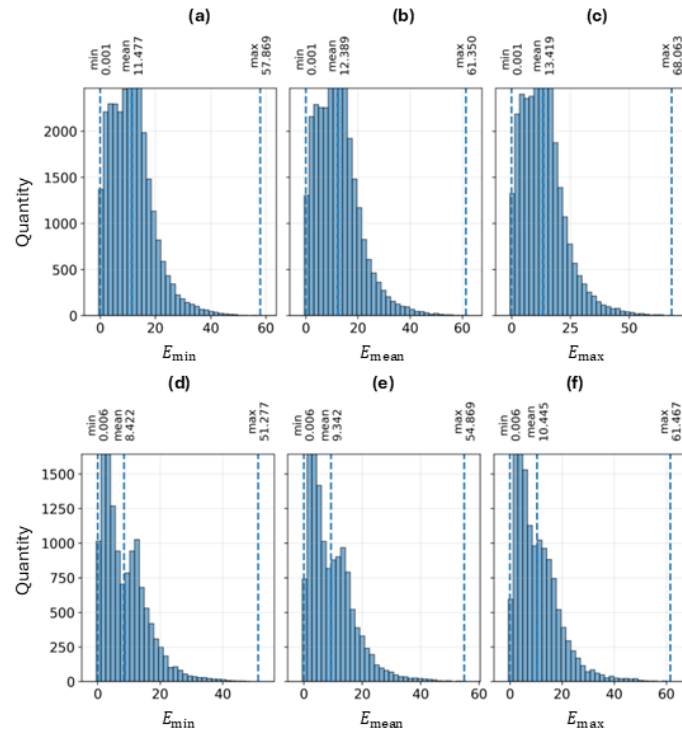


Figure S12. Global distribution of effective elastic moduli over all simulations (a-c) Database before enrichment; (d-f) "Enriched" and "pruned" database

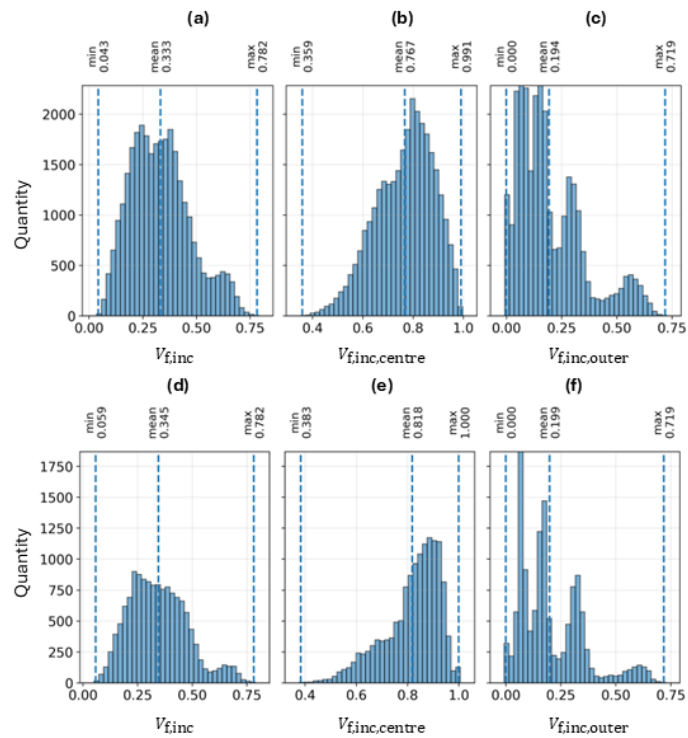


Figure S13. Global distribution of volume fractions over all simulations (a-c) Database before enrichment; (d-f) "Enriched" and "pruned" database

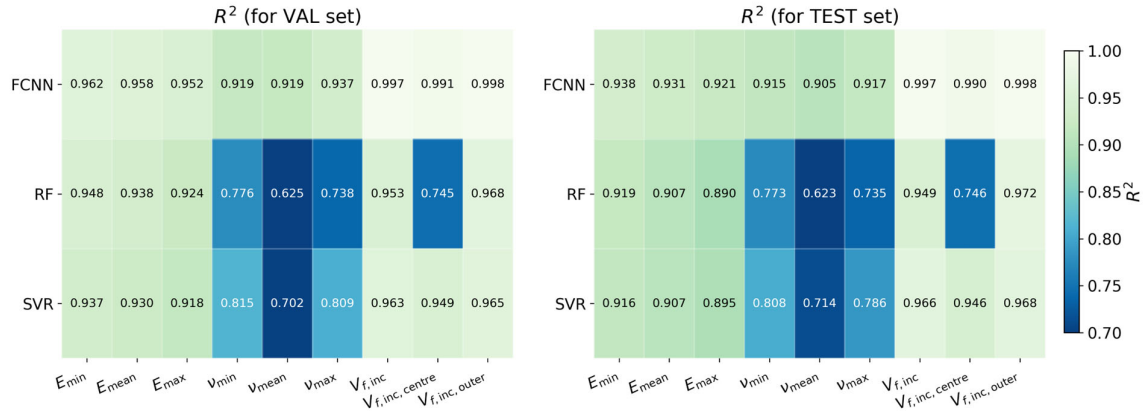


Figure S14. Per-target coefficients of determination (R^2) for FCNN, RF, and SVR on the validation (VAL) and independent test (TEST) sets. Results are shown for all nine outputs, covering elastic moduli, Poisson-ratio descriptors, and volume fractions.

- FCNN achieves the highest R^2 values across all targets on both VAL and TEST.
- The largest deficits of the baseline models occur for the Poisson-ratio descriptors, especially v_{mean} , and for RF also for $V_{f,inc,centre}$.

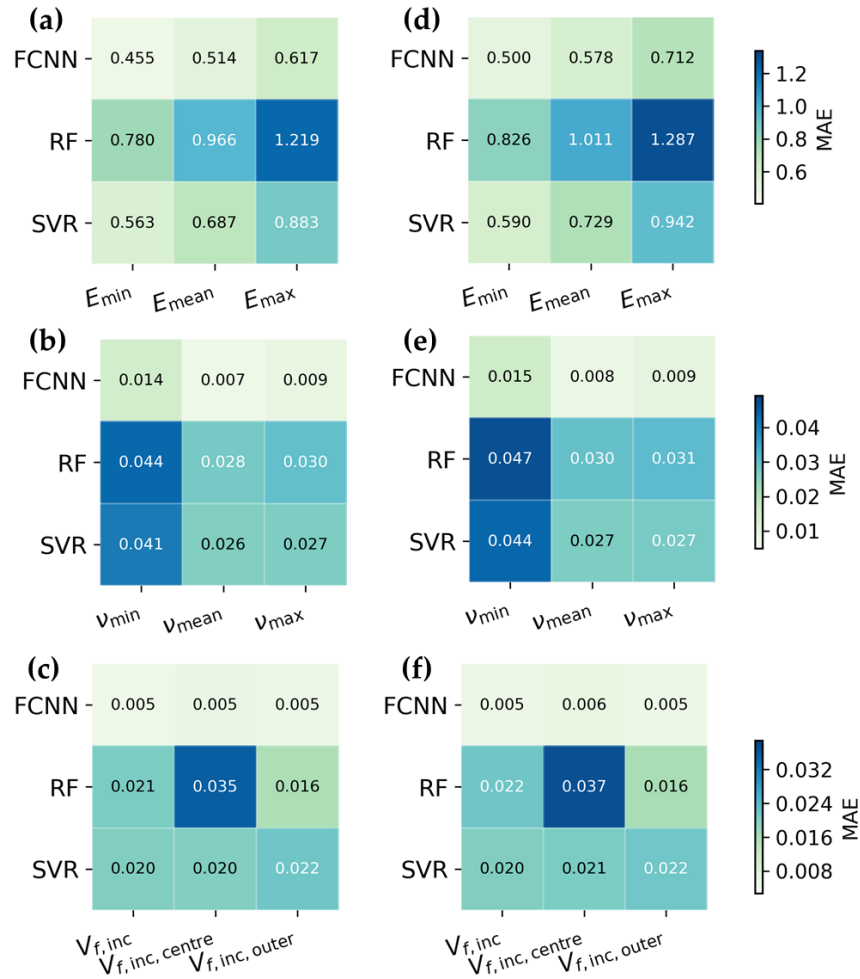


Figure S15. Per-target mean absolute errors (MAE) for FCNN, RF, and SVR on the validation (VAL) (a-c) and independent test (TEST) sets (d-f), shown separately for elastic moduli (a,d), Poisson's ratios (b,e), and volume fractions (c,f).

- FCNN yields the lowest MAE for every target on both data splits.
- Among the baseline models, SVR is generally closer to FCNN than RF, while RF shows particularly large errors for ν_{\min} and $V_{f,\text{inc,centre}}$.

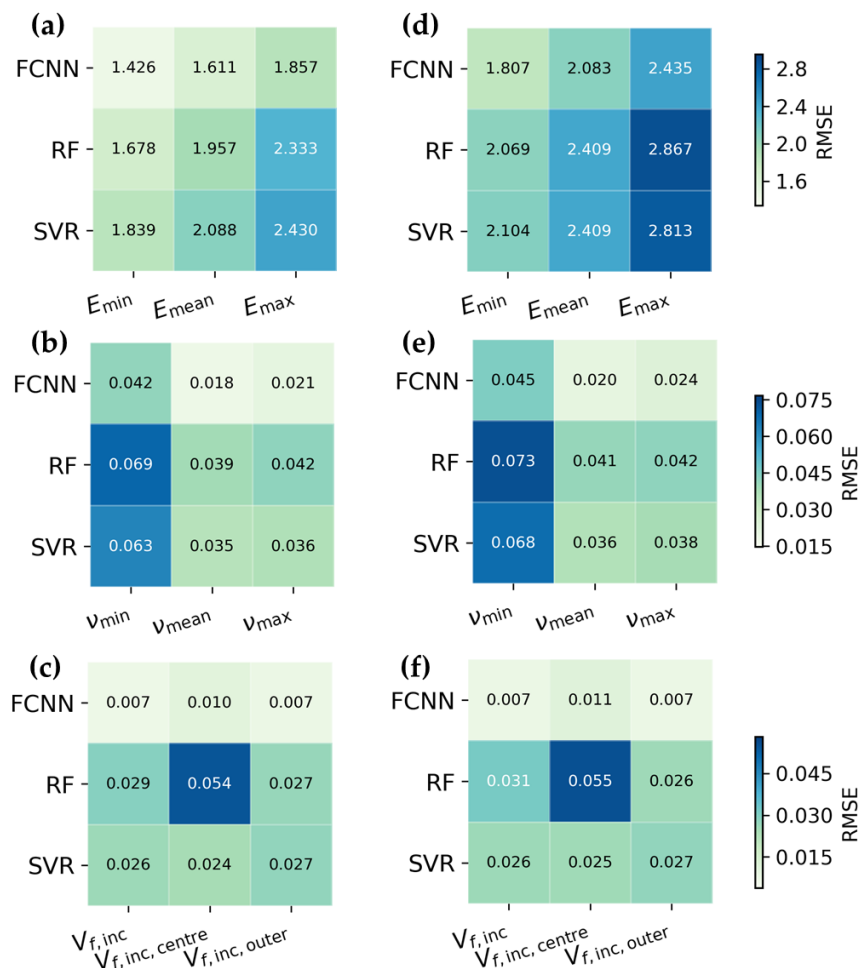


Figure S16. Per-target root mean square errors (RMSE) for FCNN, RF, and SVR on the validation (VAL) (a-c) and independent test (TEST) sets (d-f), shown separately for elastic moduli (a,d), Poisson's ratios (b,e), and volume fractions (c,f).

- The RMSE patterns closely mirror the MAE results and confirm that the model ranking remains stable from validation to test.
- The Poisson's ratios and, in absolute terms, the elastic-modulus targets remain more demanding for the baseline models than the volume-fraction outputs.

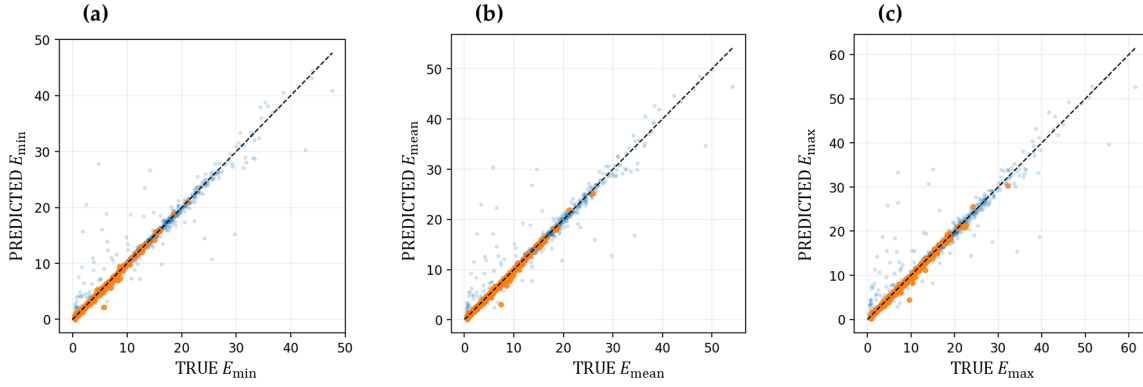


Figure S17. Observed vs. predicted Poisson's ratios on the validation subset (FCNN): (a) ν_{\min} , (b) ν_{mean} , and (c) ν_{\max} . Orange markers: auxetic samples, blue dots: non-auxetic samples.

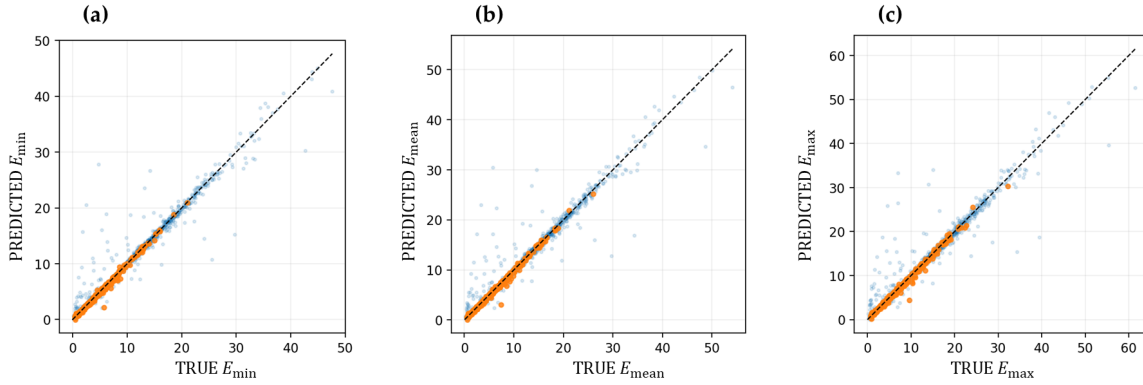


Figure S18. Observed vs. predicted elastic moduli on the validation subset (FCNN): (a) E_{\min} , (b) E_{mean} , (c) E_{\max} . Orange markers: auxetic samples, blue dots: non-auxetic samples.

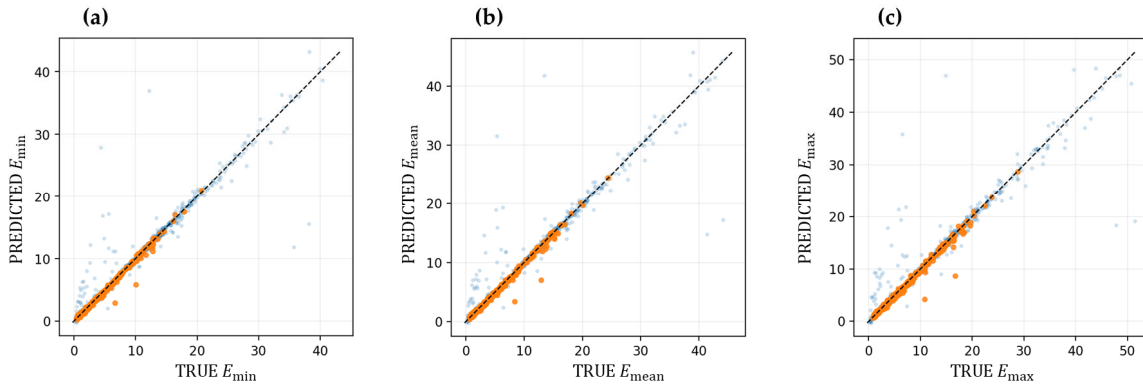


Figure S19. Observed vs. predicted elastic moduli on the hold-out test subset (FCNN): (a) E_{\min} , (b) E_{mean} , (c) E_{\max} . Orange markers: auxetic samples, blue dots: non-auxetic samples.

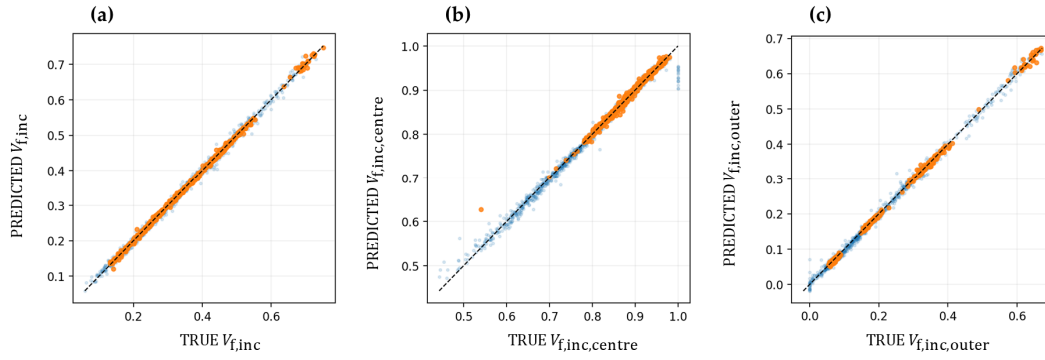


Figure S20. Observed vs. predicted volume fractions on the validation subset (FCNN): **(a)** $V_{f,inc}$, **(b)** $V_{f,inc,centre}$, and **(c)** $V_{f,inc,outer}$. Orange markers: auxetic samples, blue dots: non-auxetic samples.

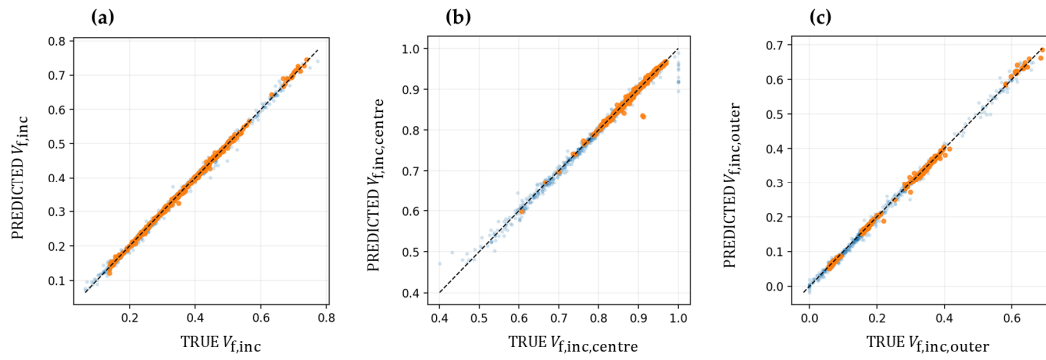


Figure S21. Observed vs. predicted volume fractions on the hold-out test subset (FCNN): **(a)** $V_{f,inc}$, **(b)** $V_{f,inc,centre}$, and **(c)** $V_{f,inc,outer}$. Orange markers: auxetic samples, blue dots: non-auxetic samples.

Table S2. Poisson's ratios (ground truth vs. prediction) with absolute errors for the six leading-regime RVEs (Case Study 1).

RVE	ν_{min}			ν_{mean}			ν_{max}		
	TRUE	PRED	Abs. error	TRUE	PRED	Abso. error	TRUE	PRED	Abs. error
1	-0.053	-0.067	0.014	0.158	0.151	0.007	0.330	0.324	0.006
2	-0.024	0.000	0.023	0.098	0.102	0.003	0.207	0.195	0.012
3	-0.021	-0.022	0.001	0.089	0.098	0.009	0.189	0.207	0.018
4	-0.017	-0.022	0.005	0.132	0.124	0.008	0.263	0.251	0.012
5	-0.005	-0.002	0.003	0.117	0.126	0.009	0.227	0.241	0.014
6	0.000	-0.008	0.009	0.127	0.119	0.008	0.239	0.234	0.005

Table S3. Elastic moduli (ground truth vs. prediction) with absolute errors for the six leading-regime RVEs (Case Study 1).

RVE	E_{\min} [GPa]			E_{mean} [GPa]			E_{\max} [GPa]		
	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error
1	18.30	18.49	0.19	23.04	23.38	0.34	28.94	29.56	0.62
2	23.35	22.25	1.09	26.54	24.99	1.55	30.13	28.05	2.08
3	16.10	16.01	0.10	18.19	18.30	0.10	20.58	20.93	0.35
4	17.43	17.41	0.02	20.73	20.46	0.26	24.62	24.10	0.52
5	18.50	18.79	0.29	21.16	21.90	0.73	24.15	25.52	1.38
6	19.85	20.08	0.23	22.77	23.05	0.28	26.09	26.50	0.41

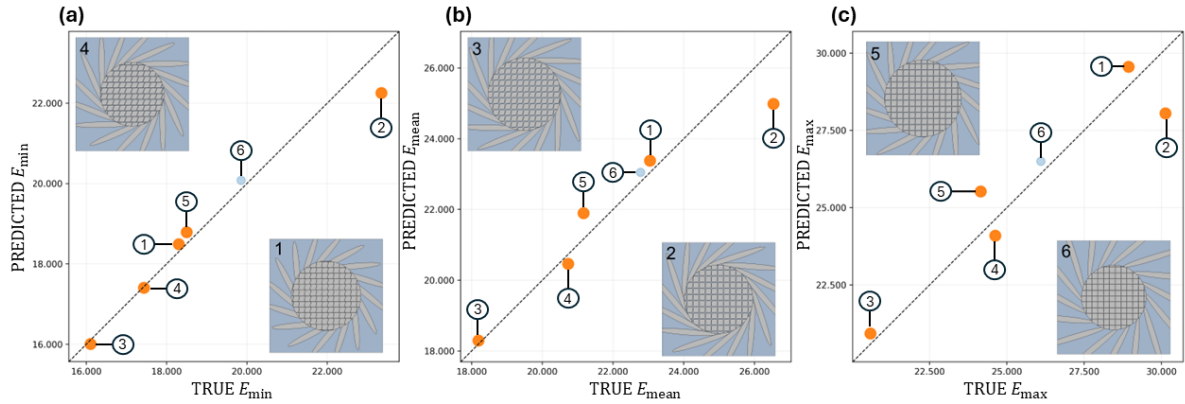


Figure S22. Case “leading part”: Parity plots for Elastic moduli on selected RVEs (predicted based on trained FCNN regressor model): (a) E_{\min} , (b) E_{mean} and (c) E_{\max} in GPa. Orange markers: auxetic samples, blue dots: non-auxetic samples.

Table S4. Volume fractions (ground truth vs. prediction) with absolute errors for the six leading-regime RVEs (Case Study 1).

RVE	$V_{f,inc}$ [%]			$V_{f,inc,centre}$ [%]			$V_{f,inc,outer}$ [%]		
	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error
1	48.2%	49.1%	0.9%	96.0%	95.2%	0.8%	28.2%	30.7%	2.5%
2	53.5%	52.4%	1.1%	83.3%	83.1%	0.2%	41.5%	40.1%	1.5%
3	52.1%	51.1%	1.0%	81.2%	80.9%	0.3%	38.2%	37.6%	0.6%
4	47.8%	47.7%	0.1%	86.4%	86.3%	0.1%	35.0%	34.8%	0.2%
5	51.3%	51.6%	0.3%	86.6%	86.7%	0.2%	31.5%	32.7%	1.2%
6	47.2%	47.4%	0.2%	94.6%	94.9%	0.3%	30.8%	31.3%	0.5%

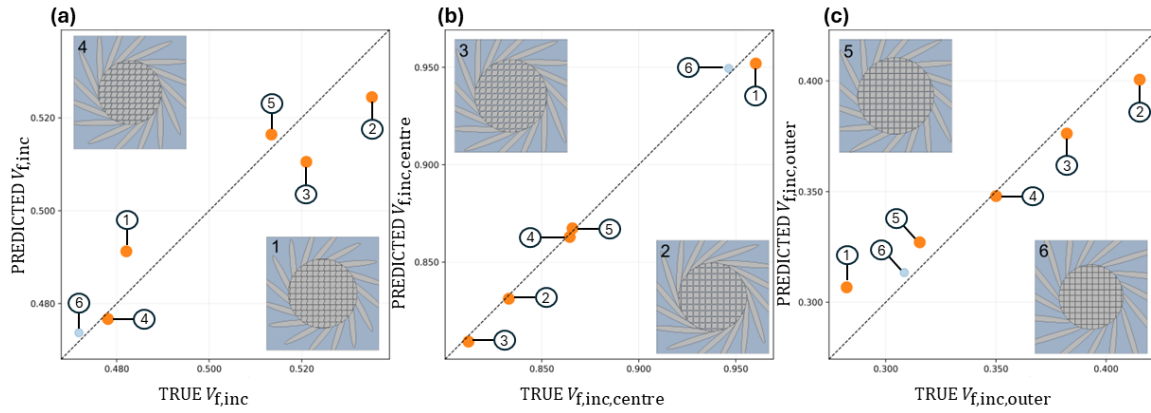


Figure S23. Case “leading part”: Parity plots for the volume fractions on selected RVEs (predicted based on trained FCNN regressor model): **(a)** $V_{f,inc}$, **(b)** $V_{f,inc,centre}$, and **(c)** $V_{f,inc,outer}$. Orange markers: auxetic samples, blue dots: non-auxetic samples.

Table S5. Poisson’s ratios (ground truth vs. prediction) with absolute errors for the eight RVEs (Case Study 2).

RVE	ν_{min}			ν_{mean}			ν_{max}		
	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error
1	-0.510	-0.531	0.020	0.065	0.054	0.011	0.422	0.423	0.001
2	-0.508	-0.495	0.013	0.037	0.041	0.004	0.388	0.388	0.000
3	-0.094	-0.042	0.053	0.122	0.121	0.001	0.297	0.259	0.039
4	-0.004	0.034	0.038	0.147	0.161	0.014	0.277	0.274	0.003
5	0.000	-0.001	0.001	0.145	0.144	0.001	0.271	0.268	0.003
6	0.000	0.002	0.002	0.139	0.140	0.001	0.259	0.259	0.000
7	0.150	-0.202	0.352	0.229	0.122	0.107	0.307	0.358	0.052
8	0.151	-0.075	0.226	0.233	0.136	0.097	0.308	0.307	0.002

Table S6. Elastic moduli (ground truth vs. prediction) with absolute errors for the eight RVEs (Case Study 2).

RVE	E_{min} [GPa]			E_{mean} [GPa]			E_{max} [GPa]		
	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error
1	0.89	0.95	0.06	1.43	1.48	0.05	2.32	2.41	0.09
2	1.41	1.09	0.32	2.22	1.83	0.39	3.49	3.04	0.45
3	7.34	6.71	0.63	9.17	7.95	1.22	11.46	9.48	1.98
4	18.00	17.61	0.39	21.24	20.72	0.51	25.00	24.44	0.56
5	12.18	12.14	0.04	14.35	14.20	0.15	16.83	16.62	0.21
6	12.69	12.52	0.17	14.76	14.44	0.32	17.14	16.71	0.43
7	1.53	6.08	4.55	1.71	8.38	6.67	1.89	11.49	9.60
8	1.80	7.83	6.03	2.00	9.87	7.87	2.24	12.47	10.23

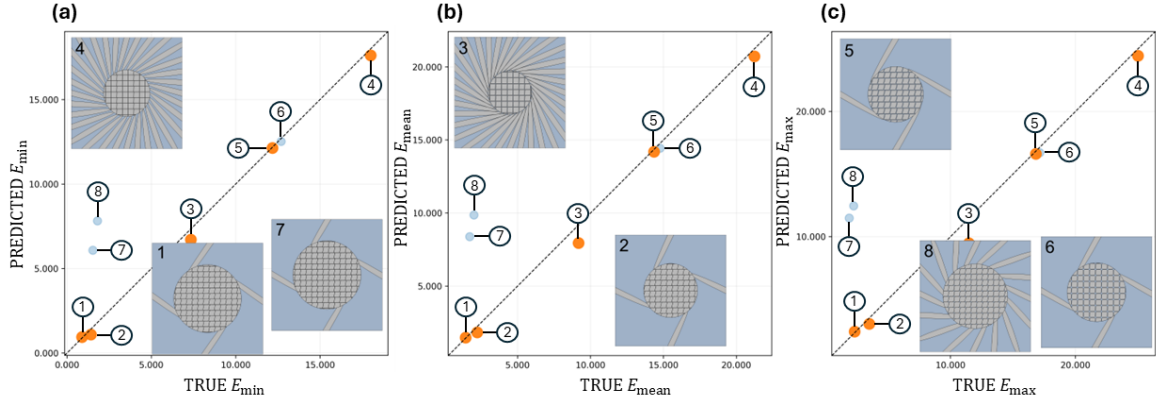


Figure S24. Case Study 2: Parity plots for Elastic moduli on selected RVEs (predicted based on trained FCNN regressor model): (a) E_{\min} , (b) E_{mean} and (c) E_{\max} in GPa. Orange markers: auxetic samples, blue dots: non-auxetic samples.

Table S7. Volume fractions (ground truth vs. prediction) with absolute errors for the eight RVEs (Case Study 2).

RVE	$V_{f,\text{inc}}$ [%]			$V_{f,\text{inc,centre}}$ [%]			$V_{f,\text{inc,outer}}$ [%]		
	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error	TRUE	PRED	Abs. error
1	33.5%	32.8%	0.6%	96.6%	96.2%	0.4%	7.4%	7.6%	0.2%
2	23.5%	23.7%	0.2%	96.4%	95.8%	0.6%	6.9%	6.6%	0.3%
3	69.8%	71.2%	1.3%	89.7%	90.1%	0.4%	67.1%	67.8%	0.7%
4	63.1%	64.5%	1.4%	95.7%	95.6%	0.1%	57.9%	58.5%	0.6%
5	22.8%	22.8%	0.1%	81.3%	81.8%	0.5%	8.5%	8.3%	0.2%
6	23.6%	23.5%	0.1%	81.2%	80.8%	0.4%	7.4%	7.7%	0.3%
7	33.6%	32.9%	0.7%	100.0%	95.8%	4.2%	6.6%	6.4%	0.2%
8	47.7%	47.0%	0.8%	100.0%	92.9%	7.1%	30.2%	30.4%	0.2%

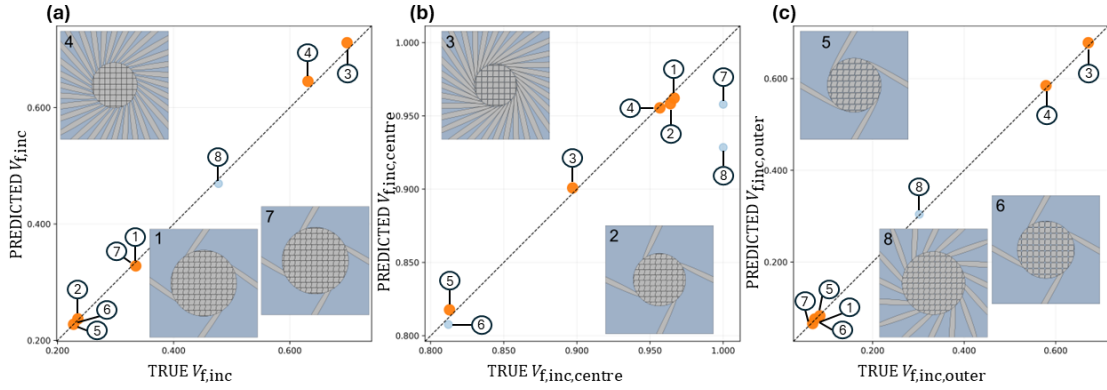


Figure S25. Case Study 2: Parity plots for the volume fractions on selected RVEs (predicted based on trained FCNN regressor model): (a) $V_{f,\text{inc}}$, (b) $V_{f,\text{inc,centre}}$, and (c) $V_{f,\text{inc,outer}}$. Orange markers: auxetic samples, blue dots: non-auxetic samples.