

Supplementary Material for

Inverse Design of Curved Mechanical Metamaterials with Geometric AI: A Generative Diffusion Operates in Compact Latent Space of Cellular Structures

Mohammad Abu-Mualla, Jida Huang¹

Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, 60607, Illinois, United States

1. Dataset Generation

1.1. Topology Creation

Building upon principles of crystallographic tetragonal symmetry, the generation of a truss unit-cell, illustrated in Fig. 1a, leverages symmetrical segmentation to reduce the complexity of the design space while preserving essential structural properties [1,2]. A cube is systematically divided into 16 symmetric prisms, allowing for efficient representation by defining only a subset of the truss topology within a single prism. This segmentation mitigates the computational burden of the exploration search by restricting structural variations to a single representative prism, ensuring both flexibility and the capacity for seamless periodic replication. The truss topology within each prism emerges from predefined interconnections among vertex, edge, and face nodes—each assigned a role based on its positional constraints. The nodal connectivity, represented as discrete variables, outlines the foundational topology, while positional adjustments through controlled offsets of movable nodes (continuous variables) extend the range of achievable mechanical properties. Reflection across symmetry planes then reconstructs a complete structure, inheriting elastic attributes consistent with the original tetragonal symmetry imposed on the truss configuration.

To ensure the connectivity of the truss structures derived from symmetry operations, we impose constraints on both types of variables. Additionally, free-to-move nodes, particularly those on edges and faces, are restricted within specified spatial limits to prevent violations of the intended symmetry. Drawing from shape function principles in finite element methods [3], we define node offsets within a natural coordinate system, representing each node's relative position with respect to the fixed vertex nodes. The spatial location of each edge node is thus expressed in terms of the relative positions of its defining vertex nodes. Let $\text{pos}(e_m)$, $\text{pos}(v_i)$, and $\text{pos}(v_j)$ denote the position vectors in \mathbb{R}^3 corresponding to nodes e_m , v_i , and v_j , respectively. The position of node e_m is thereby determined according to the following definition:

$$\text{pos}(e_m) = w_m \text{pos}(v_i) + (1 - w_m) \text{pos}(v_j) \quad (1)$$

Here, w_m serves as a weight parameter within the interval $0 \leq w_m \leq 1$, modulating the respective contributions of v_i and v_j to the positional determination of e_m . A face node may be positioned on a plane defined by three or four vertex nodes; however, determining the position of a face node within the plane requires only three vertex nodes. Let $\text{pos}(f_m)$, $\text{pos}(v_i)$, $\text{pos}(v_j)$, and $\text{pos}(v_k)$ represent the position vectors in \mathbb{R}^3 for nodes f_m , v_i , v_j , and v_k , respectively. The position of node f_m is then determined according to the following formulation:

$$\text{pos}(f_m) = w_{mi} \text{pos}(v_{mi}) + w_{mj} \text{pos}(v_{mj}) + w_{mk} \text{pos}(v_{mk}) \quad (2)$$

Here, w_{mi} , w_{mj} , and w_{mk} represent barycentric weights constrained within the interval $0 \leq w_{mi}, w_{mj}, w_{mk} \leq 1$. To ensure that node f_m remains positioned within the triangular boundary—the designated surface area—it is required that $\sum_{n=1}^3 w_{mn} = 1$. This weight constraint guarantees that points selected within the defined prism area remain distinct, preventing truss interference after reflection. However, potential interference can still occur

¹ Corresponding author: jida@uic.edu

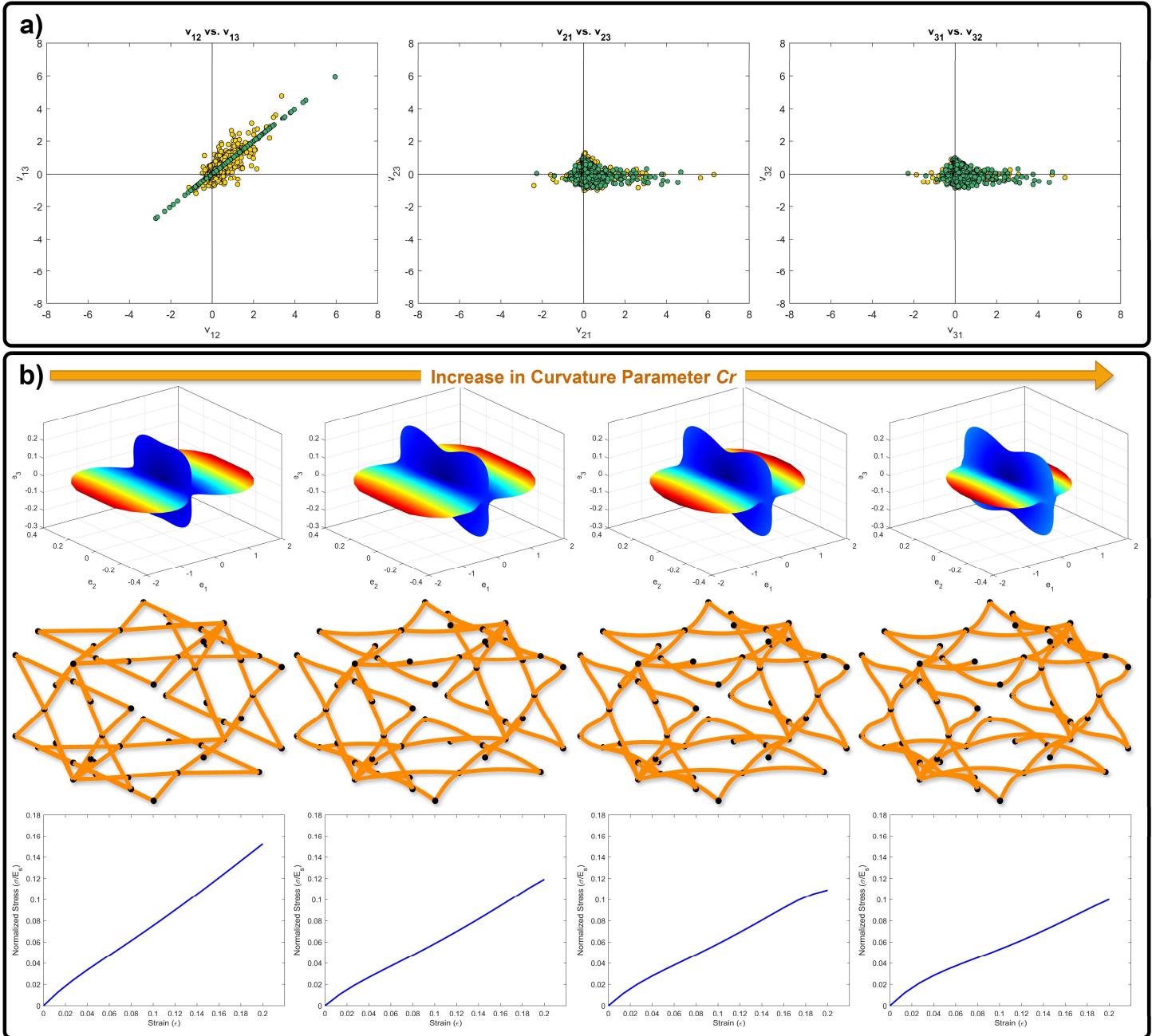
between trusses within the same prism. In such instances, any conflicting points are substituted with a connection node that links the trusses.

While the former approach ensures that nodes are confined to their designated spatial boundaries as defined by vertex nodes, the latter addresses the potential issue of disconnected trusses, where the primary concern lies in the emergence of unconnected topologies. To resolve connectivity in our dataset generation, we represent the entire topology, post-reflection, with a total structure adjacency matrix $A_d \in \{0,1\}^{\{n \times n\}}$, where n denotes the total number of nodes resulting from the unit-cell. In A_d , an entry of 1 signifies an edge between nodes, while an entry of 0 indicates no direct connection. The total adjacency matrix enables a connectivity verification by examining the eigenvalues of the Laplacian matrix derived from A_d [1,4]. This connectivity verification serves as the final refinement step in the topology generation process. The proposed methodology effectively produces datasets tailored for machine learning applications. Supplementary Table 1 shows examples of tetragonal unit-cells, along with the topological representation of each entire structure and its corresponding graph.

Supplementary Table 1: Example of tetragonal structures with their representations.

1.2. Additional Property Space Insights

Supplementary Fig. 1a displays Poisson's ratios (ν_{12} vs. ν_{13} , ν_{21} vs. ν_{23} , and ν_{31} vs. ν_{32}) for the samples utilized in the main text to visualize the design space of our curved dataset. Supplementary Fig. 1b illustrates the effect of varying the curvature parameter Cr on both the linear elastic effective properties (first row) and the compressive stress-strain behavior (third row). It is evident that with increased curvature, both the linear and nonlinear performances exhibit greater compliance.



Supplementary Fig. 1: (a) Poisson's ratio plots from left to right: ν_{12} vs. ν_{13} , ν_{21} vs. ν_{23} , and ν_{31} vs. ν_{32} for a sample of 5,000 unit-cells, illustrating the distribution of anisotropic elastic properties. (b) Influence of increasing the curvature parameter Cr on both linear and nonlinear mechanical responses of the structures. The first row displays the effective elastic property surfaces as a function of the curvature parameter Cr . The second row highlights the geometric transformations of the unit-cell as Cr increases. The third row presents the corresponding normalized compressive stress-strain curves, demonstrating enhanced compliance with increasing curvature.

2. Machine Learning Framework

2.1. Neural Networks Training Protocol

The optimized dimensions and hyperparameters for the variational autoencoder VAE, including the number of hidden layers, the number of nodes per layer, activation functions, and other relevant parameters, are comprehensively detailed in Supplementary Table 2. After evaluating various combinations of latent dimensions, the selected configuration includes $d_A = 50$, $d_x = 50$, and $d_{Ax} = 40$. Additionally, the reconstruction weight factor λ was set to 5 as it produced the optimal performance.

Supplementary Table 2: Optimized dimensions and training hyperparameters for the VAE.

Component	Adjacency Encoder $E_{A\Phi}$	Feature Encoder $E_{X\Phi}$	Adjacency Decoder $D_{A\Theta}$	Feature Decoder $D_{x\Theta}$	Property Predictor P_ω
Input Dimension	210	25	60	60	140
Hidden Dimensions	512, 512, 512, 128	640, 640, 640, 512,	128, 512, 512, 512	640, 640, 512, 256	400, 800, 1000, 400, 400, 200
Output Dimensions	60	60	210	25	37
Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU
Optimizer [5]	Adam	Adam	Adam	Adam	Adam
Batch Size	512	512	512	512	512
Training Epoch	200 epoch or early stop criteria	200 epoch or early stop criteria	200 epoch or early stop criteria	200 epoch or early stop criteria	200 epoch or early stop criteria

The optimized dimensions and hyperparameters for the denoising network, including the number of hidden layers, the number of nodes per layer, activation functions, and other relevant parameters, are comprehensively detailed in Supplementary Table 3.

Supplementary Table 3: Optimized dimensions and training hyperparameters for the diffusion model.

Component	Denoising Network p_η
Input Dimension	140
Denoise Layers	3
Denoise Hidden Dimensions Per Denoising Layer	512, 512
Output Dimensions	140
Activation Function	ReLU
Property Encoder $\tau\psi$ Dimensions.	37, 128, 512
Property Encoder $\tau\psi$ Activation Function.	ReLU
Time Positional Embedding Dimensions	140, 512, 512
Time Positional Embedding Activation Function	GELU
Denoise Hidden Layers Architecture	Each hidden layer receives the previous layer's output concatenated with the Property Encoder $\tau\psi$ output, followed by ReLU activation, batch normalization, and residual connections time positional embeddings.
Time steps	100
Beta Schedule	Linear Beta Scheduler
Optimizer [5]	Adam
Batch Size	512
Training Epoch	1000 or early stop criteria

The computational runtime and the hardware resources needed for each task are detailed in Supplementary Table 4.

Supplementary Table 4: Overview of the computational runtime and the hardware resources required for different tasks. *Computations were performed on a single Nvidia GeForce RTX 3060 GPU with 12 GB GDDR6 memory. ‡Computations were performed on a desktop equipped with a 12th Gen Intel Core i7-12700 processor (2.10 GHz) and 16 GB DDR4 RAM.

Task	Hardware	Runtimes
Joint training of the attributed network embedding VAE and the property predictor P_ω	GPU*	5 hours 15 minutes
Training of the denoising network p_η	GPU*	1 hour 47 minutes
Inverse design using gradient-based optimization – 2,000 target points, 50 samples per target point, and 100 optimization cycles per sample	CPU (10 cores)‡	13 hours 43 minutes
Inverse design using diffusion model (inference) – 2,000 target points, 50 samples per target point, and 100 denoising time steps	GPU*	23 minutes

2.2. Machine Learning Performance

2.2.1. VAE Performance

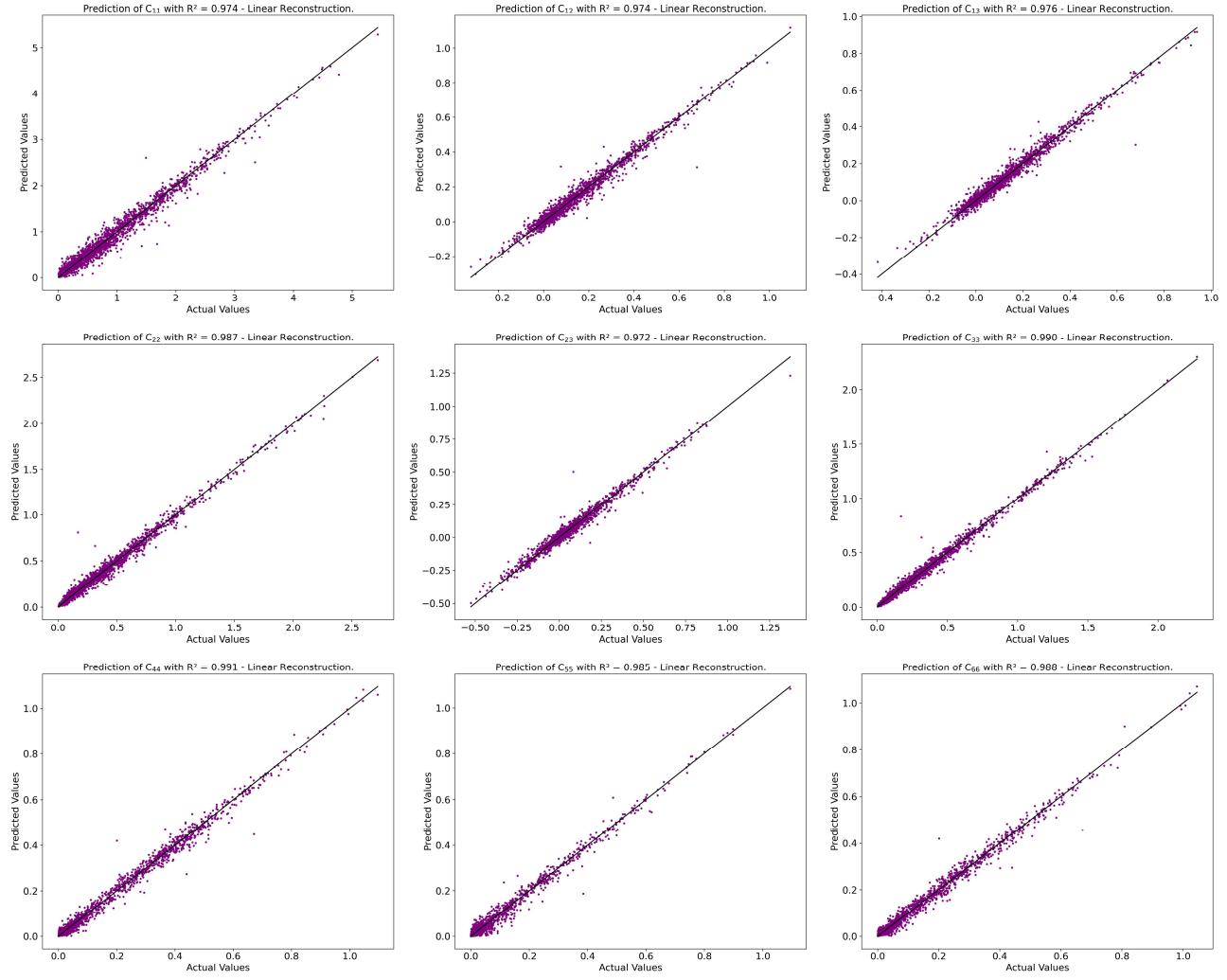
The VAE was trained to accurately reconstruct both the adjacency matrix and the feature vector associated with each structure. To evaluate reconstruction performance, the model was tested on a dataset not seen during training. The reconstruction accuracy for the adjacency matrix exceeded 0.999, indicating near-perfect structural recovery. The reconstruction accuracy for the parameters within the feature vector also indicated near-perfect feature recovery, with detailed results provided in Supplementary Table 5. This distinction highlights the VAE's robustness in preserving both the topological and geometrical information of the original structure.

Supplementary Table 5: VAE reconstruction accuracy of the feature vector.

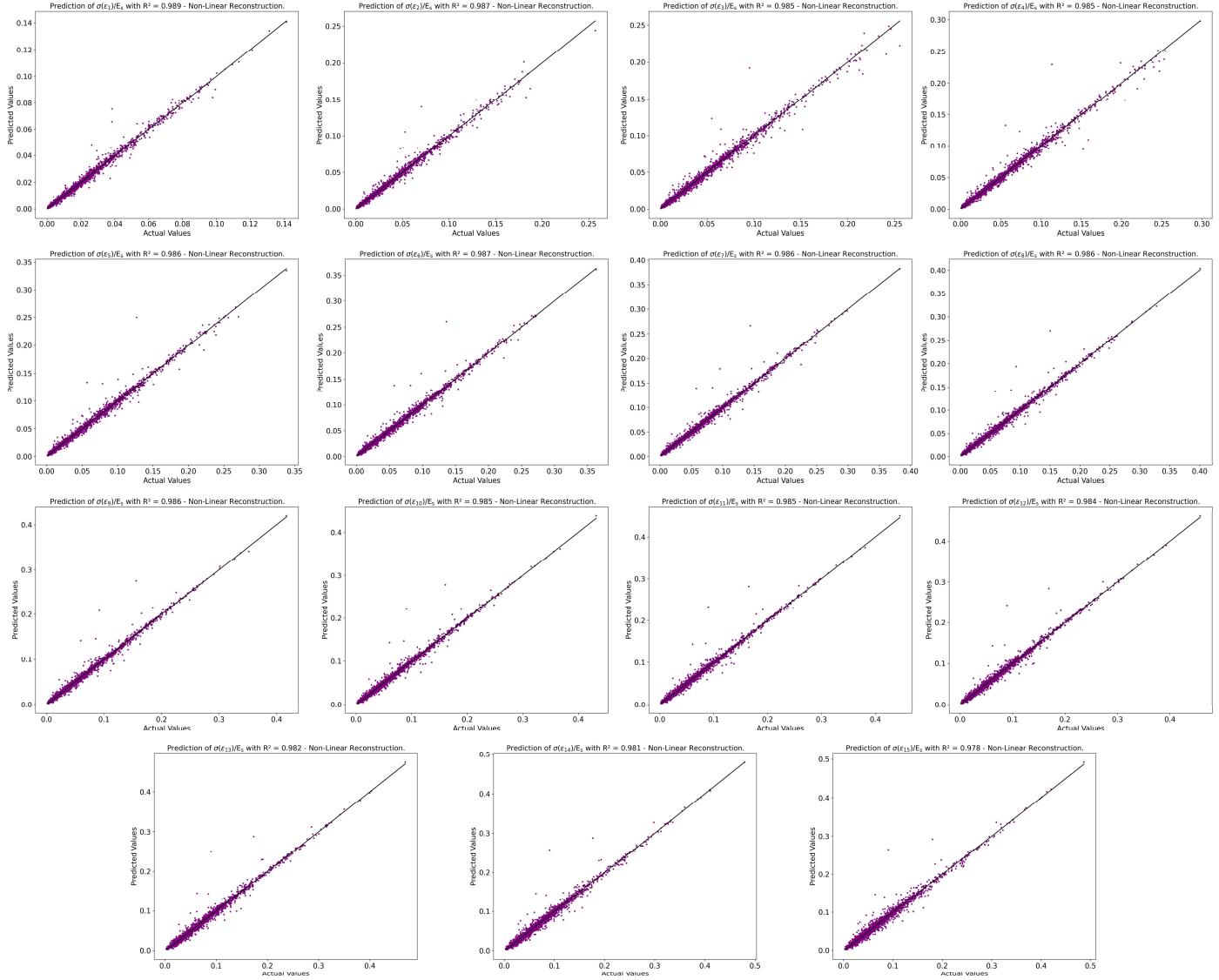
Column	R ²	Column	R ²	Column	R ²	Column	R ²	Column	R ²
w ₀	0.9994	w ₅	0.9988	w ₀₂	0.9968	w ₂₁	1.0000	w ₃₃	1.0000
w ₁	0.9990	w ₆	0.9992	w ₀₃	0.9909	w ₂₂	1.0000	w ₄₁	1.0000
w ₂	0.9869	w ₇	0.9994	w ₁₁	0.9999	w ₂₃	1.0000	w ₄₂	1.0000
w ₃	0.9989	w ₈	0.9990	w ₁₂	1.0000	w ₃₁	1.0000	w ₄₃	1.0000
w ₄	0.9994	w ₀₁	0.9942	w ₁₃	0.9999	w ₃₂	1.0000	Cr	0.9980

2.2.2. Property Predictor P_ω Performance

The truss material was initially designed as a tetragonal straight truss structure, characterized by six independent stiffness components. It was subsequently modified using the curvature parameter Cr , which influenced the structural symmetry. This modification resulted in most structures remaining tetragonally dominant, while some transitioned to being orthotropically dominant, exhibiting nine independent stiffness components. Although certain structures displayed non-zero values for the remaining stiffness components associated with triclinic symmetry (21 independent stiffness components), these values were significantly smaller compared to the orthotropic components, confirming the dominance of the orthotropic behavior. Consequently, for clarity and relevance, all subsequent figures focus exclusively on the prediction accuracy of the orthotropic stiffness components. The prediction performance of the linear stiffness tensor values is shown in Supplementary Fig. 2. And the prediction performance of the nonlinear compressive stress values along the predefined strains ($\epsilon=1.33\%, 2.66\%, \dots, 20\%$) is shown in Supplementary Fig. 3. In Supplementary Figs. 2 and 3, the property predictor model P_ω was trained to predict both linear and nonlinear properties jointly. Supplementary Table 6 further shows that high prediction performance can be achieved by correlating the latent space with only the linear property space (effective stiffness tensor C , as shown in Fig. 3b of the main manuscript) or with only the nonlinear property space (compressive stress values).



Supplementary Fig. 2: Assessment of the prediction accuracy of the property predictor model P_ω in predicting the orthotropic components of the linear stiffness tensor: A comparison between the predicted and true components of the stiffness tensor \mathbf{C} in the test dataset.



Supplementary Fig. 3: Assessment of the prediction accuracy of the property predictor model P_ω in predicting nonlinear compression stress values $\sigma(\epsilon)$: A comparison between the predicted and true components of the compression stress values $\sigma(\epsilon)$ in the test dataset.

Supplementary Table 6: Property predictor model P_ω performance when related to different combinations of properties (both linear and nonlinear, linear only, and nonlinear only).

	Both Linear and Nonlinear Properties	Linear Properties [Effective stiffness tensor C]	Nonlinear Properties [Compressive stress values $\sigma(\epsilon)$]
C_{11}	0.9748	0.9888	—
C_{12}	0.9743	0.9863	—
C_{13}	0.9756	0.9862	—
C_{22}	0.9868	0.9911	—
C_{23}	0.9722	0.9834	—
C_{33}	0.9896	0.9920	—
C_{44}	0.9905	0.9951	—
C_{55}	0.9852	0.9920	—
C_{66}	0.9883	0.9939	—
$\sigma(\epsilon_1)$	0.9889	—	0.9861
$\sigma(\epsilon_2)$	0.9867	—	0.9844
$\sigma(\epsilon_3)$	0.9852	—	0.9833
$\sigma(\epsilon_4)$	0.9844	—	0.9848

$\sigma(\epsilon_5)$	0.9859	—	0.9866
$\sigma(\epsilon_6)$	0.9862	—	0.9876
$\sigma(\epsilon_7)$	0.9862	—	0.9883
$\sigma(\epsilon_8)$	0.9860	—	0.9888
$\sigma(\epsilon_9)$	0.9854	—	0.9890
$\sigma(\epsilon_{10})$	0.9850	—	0.9888
$\sigma(\epsilon_{11})$	0.9844	—	0.9883
$\sigma(\epsilon_{12})$	0.9836	—	0.9873
$\sigma(\epsilon_{13})$	0.9821	—	0.9857
$\sigma(\epsilon_{14})$	0.9801	—	0.9832
$\sigma(\epsilon_{15})$	0.9774	—	0.9798

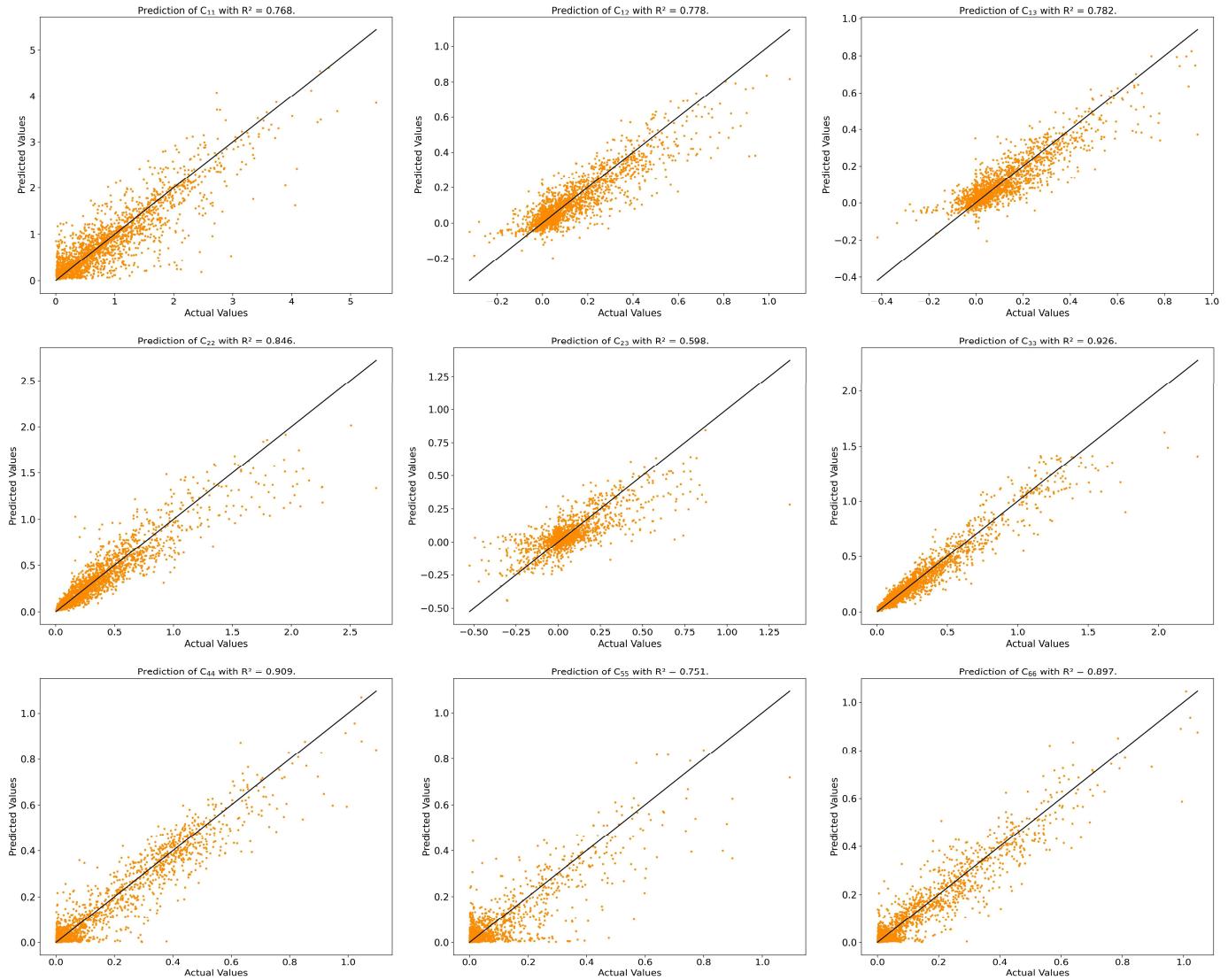
2.2.3. Inverse Design Performance

Building upon the analysis presented in Fig. 5 of the main manuscript, which demonstrated the inverse prediction accuracy for 2,000 unseen testing samples (with 50 samples generated for each testing point), we extend the evaluation to provide a more comprehensive insight into the prediction performance. While only a subset of results was illustrated in the main figure, Supplementary Fig. 4 and Supplementary Fig. 5 depict the prediction accuracy in terms of the sample with the lowest normalized mean square error (NMSE) relative to the target sample, quantified by the coefficient of determination R^2_{best} . These results are presented for both the gradient-based optimization method and the diffusion model, respectively. Further analysis is provided in Supplementary Fig. 6 and Supplementary Fig. 7, which showcase the 95% confidence interval predictions derived from the 50 generated samples for 100 unseen target stiffness tensors C . Additionally, the mean prediction accuracy, represented as R^2_{mean} , is computed using the mean values of the reconstructed samples for each target tensor, with results shown for both the gradient-based optimization and diffusion model.

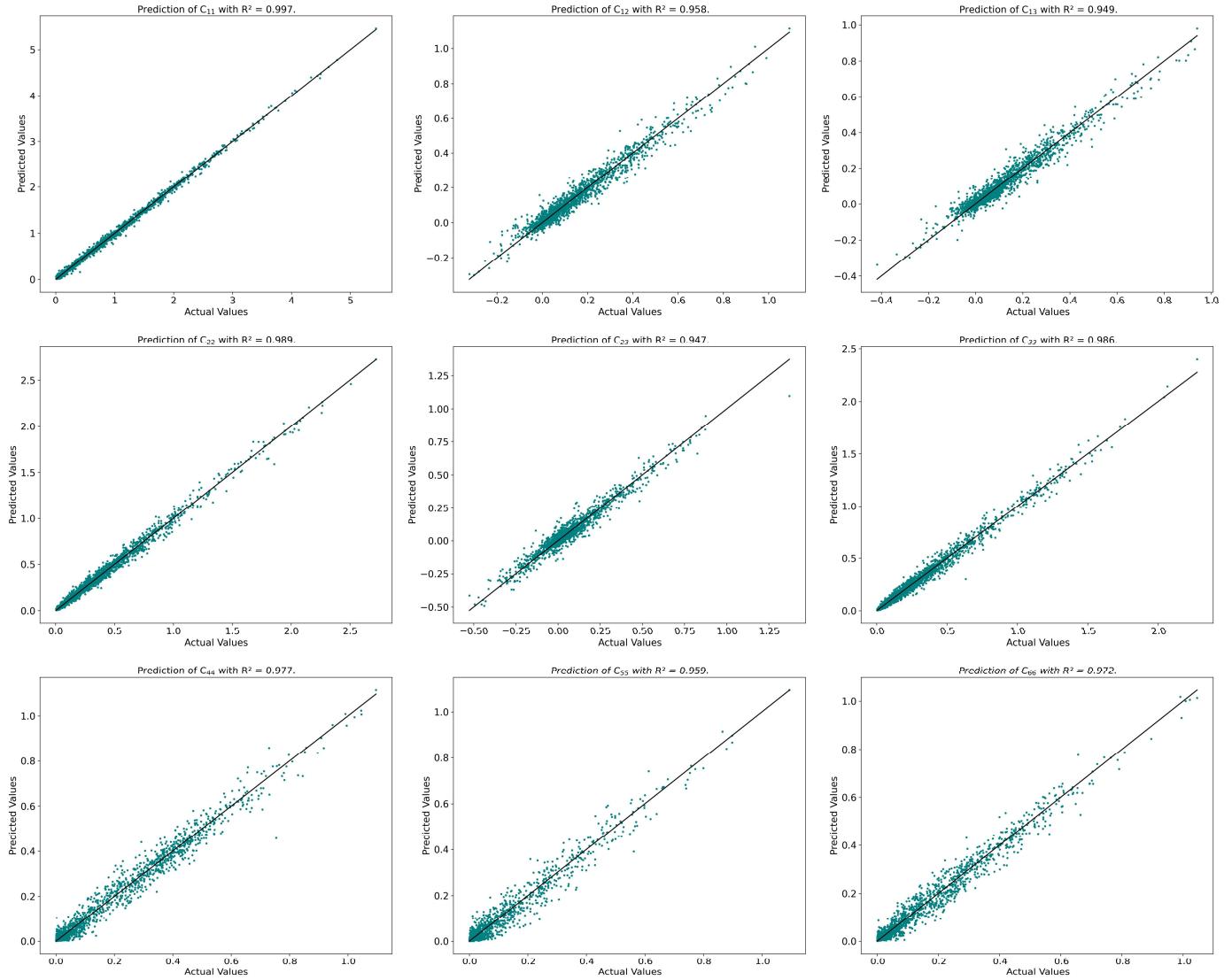
Supplementary Fig. 8 and Supplementary Fig. 9 illustrate the reconstructed samples distribution, evaluated using the Wasserstein distance W relative to the target distribution, highlighting the distributional fidelity of the predictions for both the gradient-based optimization and diffusion model. This comprehensive analysis provides a nuanced understanding of the model's inverse prediction capabilities across various metrics and evaluation frameworks. All the inverse prediction results are summarized in Supplementary Table 7, along with the prediction accuracy of both models, based on comparison with all 50 generated samples for the entire unseen testing points, represented by R^2_{all} .

Supplementary Table 7: Summary of inverse prediction accuracy in terms of R^2_{best} , R^2_{bmean} , R^2_{best} , and W for both Gradient-Based Optimization within the Latent Space and the Latent Diffusion Model.

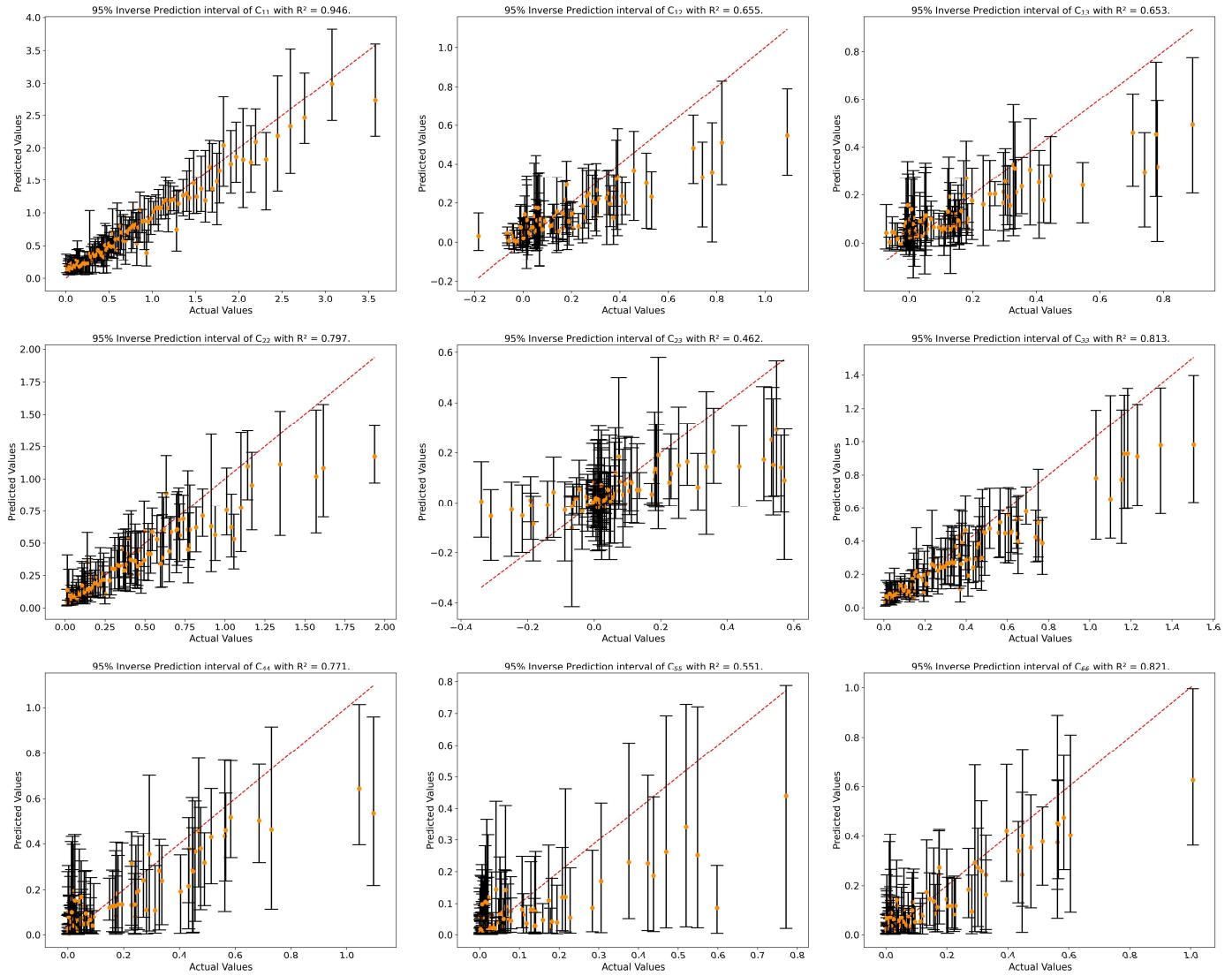
	Gradient-based Optimization Within the Latent Space				Latent Diffusion Model			
	R^2_{best}	R^2_{mean}	R^2_{all}	W	R^2_{best}	R^2_{mean}	R^2_{all}	W
C_{11}	0.7676	0.9461	0.8572	0.0692	0.9972	0.9984	0.9846	0.0062
C_{12}	0.7778	0.6554	0.4582	0.0333	0.9585	0.9373	0.8600	0.0050
C_{13}	0.7824	0.6530	0.4415	0.0300	0.9489	0.9234	0.8419	0.0049
C_{22}	0.8459	0.7972	0.7114	0.0490	0.9892	0.9867	0.9577	0.0036
C_{23}	0.5982	0.4616	0.1739	0.0361	0.9475	0.9310	0.8384	0.0049
C_{33}	0.9258	0.8132	0.7200	0.0542	0.9856	0.9823	0.9558	0.0031
C_{44}	0.9087	0.7712	0.5092	0.0347	0.9769	0.9881	0.9395	0.0031
C_{55}	0.7512	0.5506	0.2098	0.0285	0.9586	0.9623	0.8742	0.0023
C_{66}	0.8972	0.8207	0.5294	0.0254	0.9724	0.9836	0.9294	0.0035



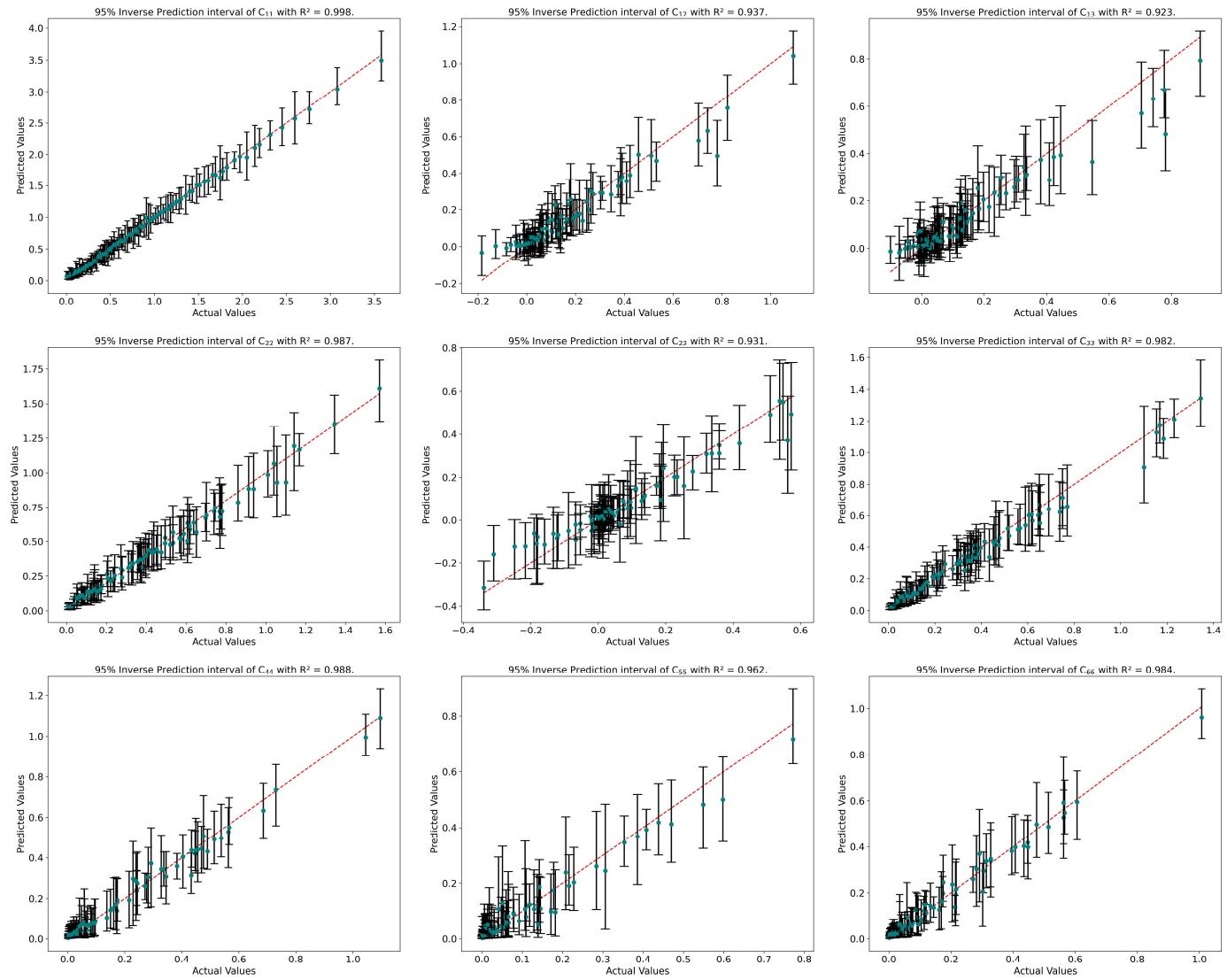
Supplementary Fig. 4: The inverse prediction accuracy, in terms of the sample with the lowest normalized mean square error (NMSE) [out of 50 reconstructed samples] relative to the target stiffness tensor \mathbf{C} , is quantified by the coefficient of determination R^2_{best} for gradient-based optimization within the latent space.



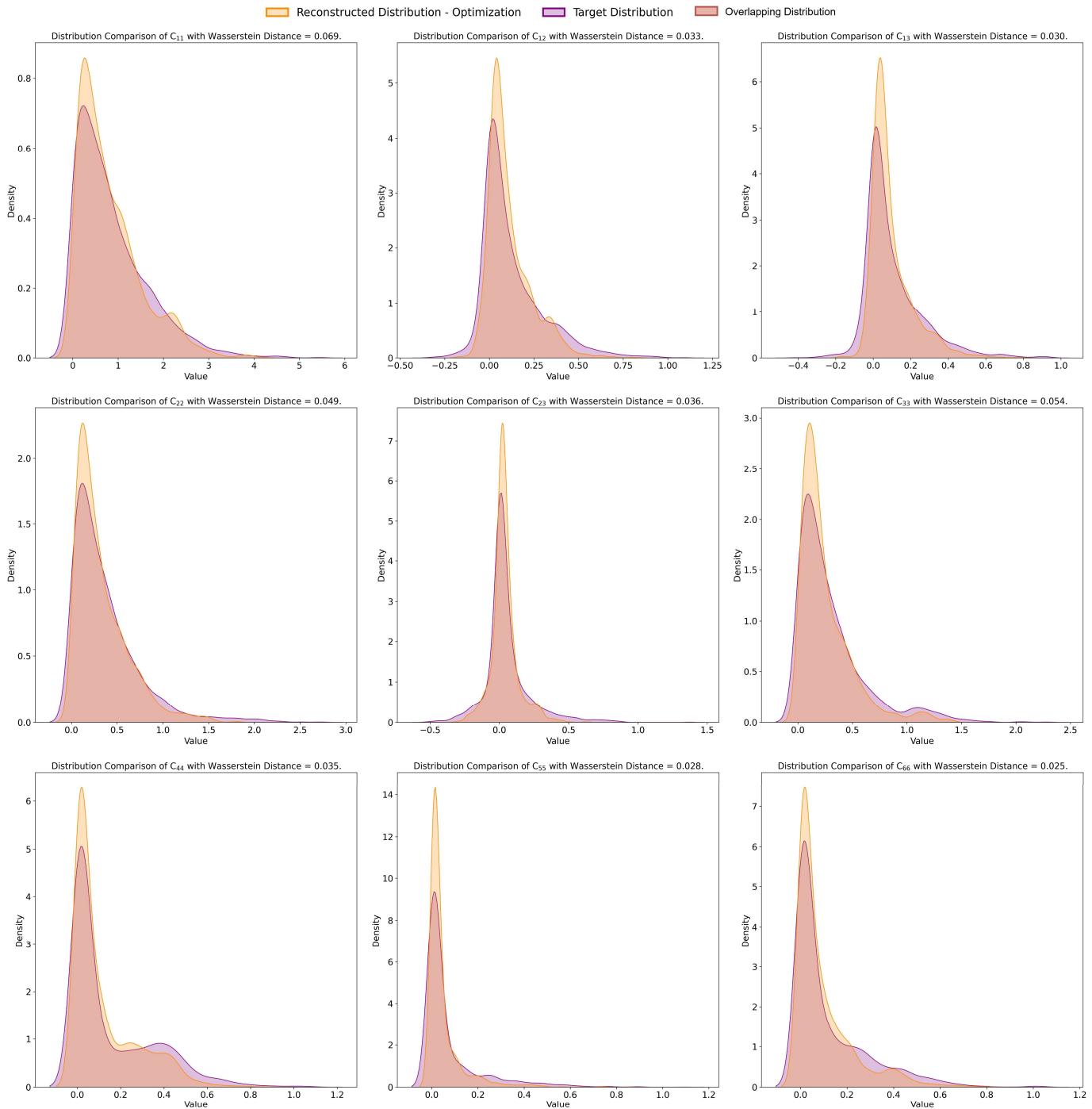
Supplementary Fig. 5: The inverse prediction accuracy, in terms of the sample with the lowest normalized mean square error (NMSE) [out of 50 reconstructed samples] relative to the target stiffness tensor \mathbf{C} , is quantified by the coefficient of determination R^2_{best} for the latent diffusion model.



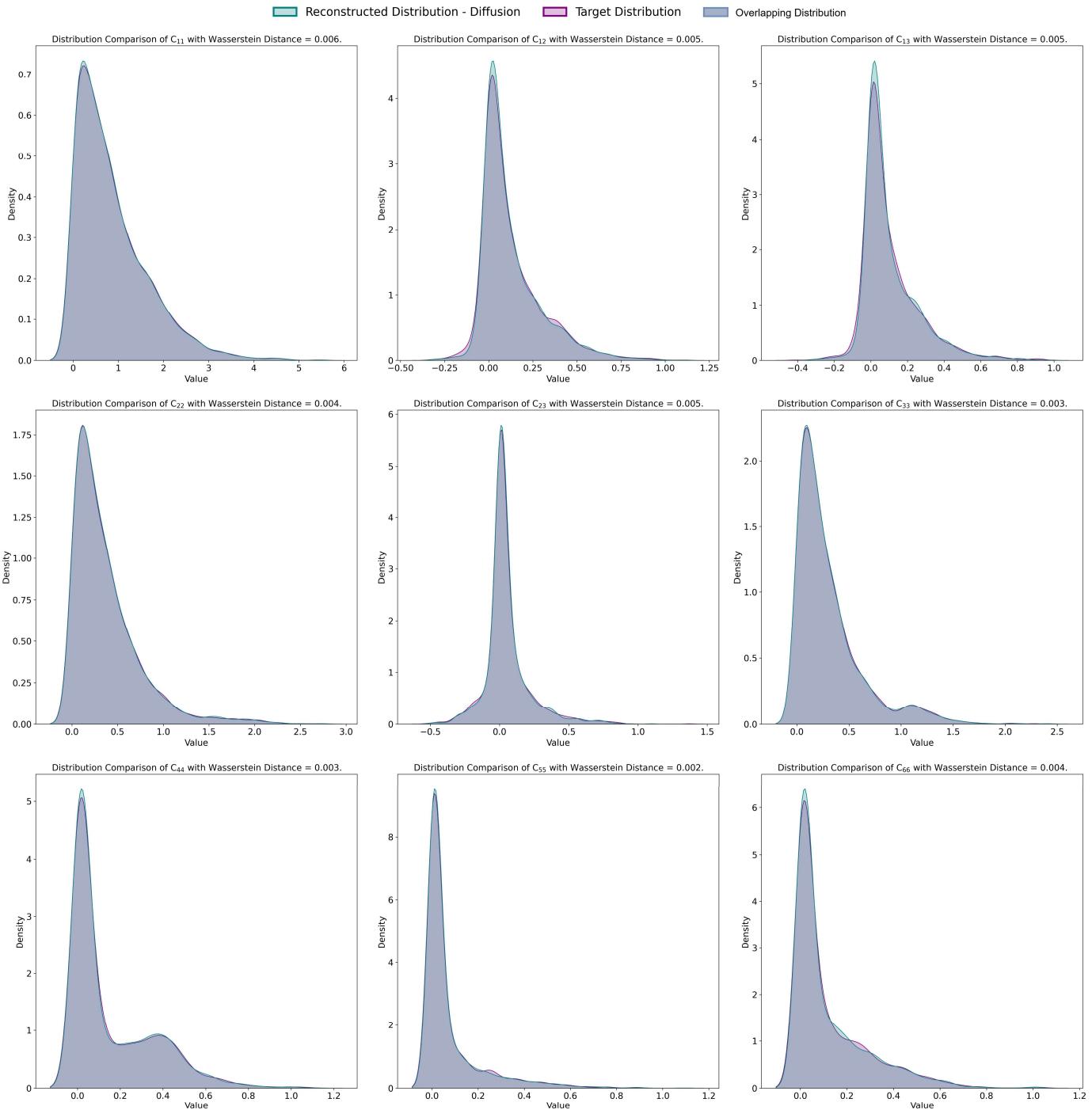
Supplementary Fig. 6: Inverse prediction accuracy with 95% confidence intervals for 100 unseen target stiffness tensors \mathbf{C} , based on 50 reconstructed samples. The mean prediction accuracy (R^2_{mean}) is calculated using the mean values of the reconstructed samples for each target tensor in the inverse design of curved truss materials, utilizing gradient-based optimization within the latent space.



Supplementary Fig. 7: Inverse prediction accuracy with 95% confidence intervals for 100 unseen target stiffness tensors \mathbf{C} , based on 50 reconstructed samples. The mean prediction accuracy (R^2_{mean}) is calculated using the mean values of the reconstructed samples for each target tensor in the inverse design of curved truss materials, utilizing latent diffusion model.



Supplementary Fig. 8: Distributions of reconstructed samples, evaluated using the Wasserstein distance W relative to the target stiffness tensor distributions, during predictions made with gradient-based optimization within the latent space.

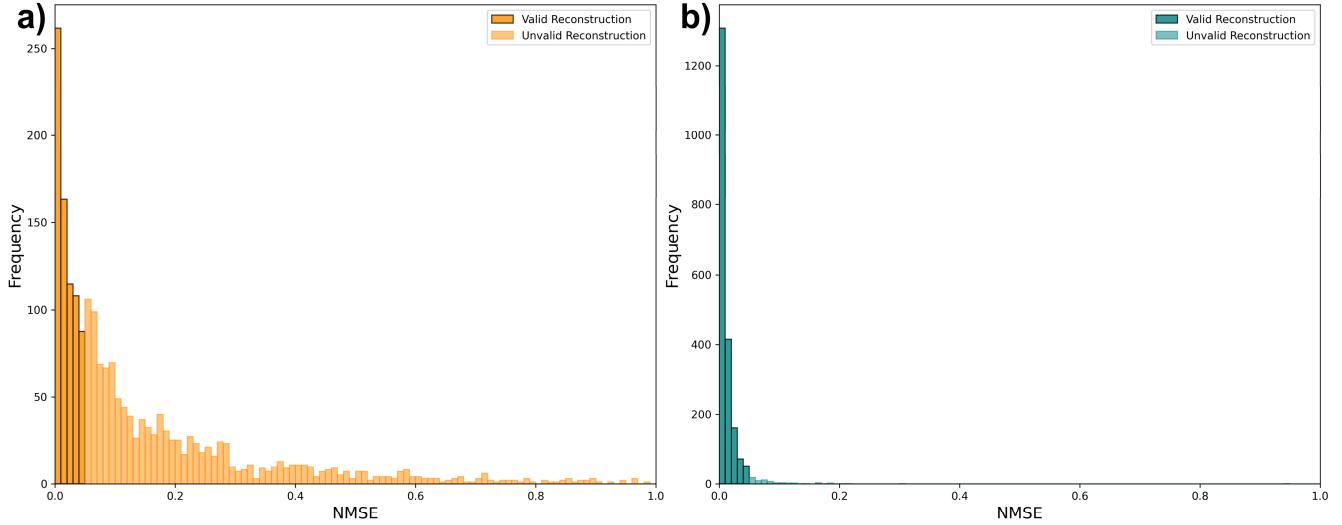


Supplementary Fig. 9: Distributions of reconstructed samples, evaluated using the Wasserstein distance W relative to the target stiffness tensor distributions, during predictions made with latent diffusion model.

We also compared the best reconstructed sample, out of 50, to the target stiffness tensor \mathbf{C} . This comparison was performed for both inverse design methods: gradient-based optimization in the latent space and the latent diffusion model. The normalized mean-squared error (NMSE), as defined in Eq. 3 [6], was used as a quantitative metric to facilitate numerical comparison and evaluate the precision of the reconstructed stiffness tensor relative to the target.

$$\text{NMSE} (\mathbf{C}, \mathbf{C}) = \frac{\sum_{i=1}^6 \sum_{j=1}^6 (C_{target,ij} - C_{pred,ij})}{\sum_{i=1}^6 \sum_{j=1}^6 C_{target,ij}} \quad (3)$$

The comparison is shown in Supplementary Fig. 10. It highlights that 35.45% of the reconstructed structures using gradient-based optimization in the latent space have NMSE values below 0.05. In contrast, approximately 96.68% of reconstructed samples using the latent diffusion model achieve NMSE values below 0.05, demonstrating the superior performance of the latent diffusion inverse model.



Supplementary Fig. 10: Histogram of the NMSEs of the inversely reconstructed unit-cells using: a) gradient-based optimization within the latent space, and b) latent diffusion model.

To demonstrate that the model can also be used to predict structures with a target nonlinear stress-strain response, we conditioned the model on the nonlinear properties and explored the results. These results are illustrated in Supplementary Table 8. It should be noted that the VAE model is trained to relate every point in the latent space to both linear and nonlinear properties. Since the inverse prediction primarily targets nonlinear compressive values, linear stiffness properties are concurrently evaluated throughout the process.

Supplementary Table 8: Gradient-based optimization within the latent space and latent diffusion model performance when conditioned on nonlinear compressive strain values, evaluated in terms of point prediction accuracy (Coefficient of determination R^2_{best}) and distribution reconstruction (Wasserstein distance W).

	Gradient-based Optimization Within the Latent Space		Latent Diffusion Model	
	R^2_{best}	W	R^2_{best}	W
$\sigma(\epsilon_1)$	0.9362	0.0032	0.9822	0.0003
$\sigma(\epsilon_2)$	0.9472	0.0046	0.9811	0.0007
$\sigma(\epsilon_3)$	0.9614	0.0052	0.9817	0.0009
$\sigma(\epsilon_4)$	0.9664	0.0054	0.9853	0.0009
$\sigma(\epsilon_5)$	0.9701	0.0055	0.9871	0.0012
$\sigma(\epsilon_6)$	0.9724	0.0056	0.9879	0.0014
$\sigma(\epsilon_7)$	0.9731	0.0056	0.9889	0.0015
$\sigma(\epsilon_8)$	0.9727	0.0056	0.9896	0.0015
$\sigma(\epsilon_9)$	0.9717	0.0056	0.9897	0.0016
$\sigma(\epsilon_{10})$	0.9694	0.0055	0.9895	0.0017
$\sigma(\epsilon_{11})$	0.9656	0.0053	0.9888	0.0018
$\sigma(\epsilon_{12})$	0.9604	0.0051	0.9872	0.0019
$\sigma(\epsilon_{13})$	0.9527	0.0050	0.9851	0.0020
$\sigma(\epsilon_{14})$	0.9432	0.0049	0.9820	0.0021
$\sigma(\epsilon_{15})$	0.9319	0.0050	0.9776	0.0022

References

- [1] Abu-Mualla, Mohammad, and Jida Huang. "A Dataset Generation Framework for Symmetry-Induced Mechanical Metamaterials." *Journal of Mechanical Design* 147, no. 4 (2025).
- [2] Cowin, Stephen C. *Continuum mechanics of anisotropic materials*. Springer Science & Business Media, 2013.
- [3] Bi, Zhuming. *Finite element analysis applications: a systematic and practical approach*. Academic Press, 2017.
- [4] Fiedler, Miroslav. "Algebraic connectivity of graphs." *Czechoslovak mathematical journal* 23, no. 2 (1973): 298-305.
- [5] Kingma, Diederik P. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
- [6] Abu-Mualla, Mohammad, and Jida Huang. "Inverse design of 3D cellular materials with physics-guided machine learning." *Materials & Design* 232 (2023): 112103.